```
**********************************************************
   47902 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/cmd-inet/lib/ipmgmtd/ipmgmt_persist.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright 2016 Argo Technologie SA.
 * Copyright (c) 2016-2017, Chris Fraire <cfraire@me.com>.
 */

/*
 * Contains DB walker functions, which are of type ‘db_wfunc_t’;
 *
 * typedef boolean_t db_wfunc_t(void *cbarg, nvlist_t *db_nvl, char *buf,
 *                              size_t bufsize, int *errp);
 *
 * ipadm_rw_db() walks through the data store, one line at a time and calls
 * these call back functions with:
 *      ‘cbarg’  - callback argument
 *      ‘db_nvl’ - representing a line from DB in nvlist_t form
 *      ‘buf’    - character buffer to hold modified line
 *      ‘bufsize’- size of the buffer
 *      ‘errp’ - captures any error inside the walker function.
 *
 * All the ’write’ callback functions modify ‘db_nvl’ based on ‘cbarg’ and
 * copy string representation of ‘db_nvl’ (using ipadm_nvlist2str()) into ‘buf’.
 * To delete a line from the DB, buf[0] is set to ‘\0’. Inside ipadm_rw_db(),
 * the modified ‘buf’ is written back into DB.
 *
 * All the ’read’ callback functions, retrieve the information from the DB, by
 * reading ‘db_nvl’ and then populate the ‘cbarg’.
 */

#include <stdlib.h>
#include <strings.h>
#include <errno.h>
#include <assert.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
```



```
#include "ipmgmt_impl.h"

/* SCF related property group names and property names */
#define IPMGMTD_APP_PG          "ipmgmtd"
#define IPMGMTD_PROP_FBD        "first_boot_done"
#define IPMGMTD_PROP_DBVER      "datastore_version"
#define IPMGMTD_TRUESTR         "true"

#define ATYPE   "_atype"                /* name of the address type nvpair */
#define FLAGS   "_flags"                /* name of the flags nvpair */

/*
 * flag used by ipmgmt_persist_aobjmap() to indicate address type is
 * IPADM_ADDR_IPV6_ADDRCONF.
 */
#define IPMGMT_ATYPE_V6ACONF    0x1

extern pthread_rwlock_t ipmgmt_dbconf_lock;

/* signifies whether volatile copy of data store is in use */
static boolean_t ipmgmt_rdonly_root = B_FALSE;

/*
 * Checks if the database nvl, ‘db_nvl’ contains and matches ALL of the passed
 * in private nvpairs ‘proto’, ‘ifname’ & ‘aobjname’.
 */
static boolean_t
ipmgmt_nvlist_match(nvlist_t *db_nvl, const char *proto, const char *ifname,
    const char *aobjname)
{
        char            *db_proto = NULL, *db_ifname = NULL;
        char            *db_aobjname = NULL;
        nvpair_t        *nvp;
        char            *name;

        /* walk through db_nvl and retrieve all its private nvpairs */
        for (nvp = nvlist_next_nvpair(db_nvl, NULL); nvp != NULL;
            nvp = nvlist_next_nvpair(db_nvl, nvp)) {
                name = nvpair_name(nvp);
                if (strcmp(IPADM_NVP_PROTONAME, name) == 0)
                        (void) nvpair_value_string(nvp, &db_proto);
                else if (strcmp(IPADM_NVP_IFNAME, name) == 0)
                        (void) nvpair_value_string(nvp, &db_ifname);
                else if (strcmp(IPADM_NVP_AOBJNAME, name) == 0)
                        (void) nvpair_value_string(nvp, &db_aobjname);
        }

        if (proto != NULL && proto[0] == ’\0’)
                proto = NULL;
        if (ifname != NULL && ifname[0] == ’\0’)
                ifname = NULL;
        if (aobjname != NULL && aobjname[0] == ’\0’)
                aobjname = NULL;

        if ((proto == NULL && db_proto != NULL) ||
            (proto != NULL && db_proto == NULL) ||
            (proto != NULL && db_proto != NULL &&
            strcmp(proto, db_proto) != 0)) {
                /* no intersection - different protocols. */
                return (B_FALSE);
        }
        if ((ifname == NULL && db_ifname != NULL) ||
            (ifname != NULL && db_ifname == NULL) ||
            (ifname != NULL && db_ifname != NULL &&
            strcmp(ifname, db_ifname) != 0)) {
                /* no intersection - different interfaces. */
```

```
 127                 return (B_FALSE);
 128         }
 129         if ((aobjname == NULL && db_aobjname != NULL) ||
 130             (aobjname != NULL && db_aobjname == NULL) ||
 131             (aobjname != NULL && db_aobjname != NULL &&
 132             strcmp(aobjname, db_aobjname) != 0)) {
 133                 /* no intersection - different address objects */
 134                 return (B_FALSE);
 135         }

 137         return (B_TRUE);
 138 }
```

*unchanged_portion_omitted_*

```
 406 /*
 407  * This function takes the appropriate lock, read or write, based on the
 408  * 'db_op' and then calls DB walker ipadm_rw_db(). The code is complicated
 409  * by the fact that we are not always guaranteed to have a writable root
 410  * filesystem since it is possible that we are reading or writing during
 411  * bootime while the root filesystem is still read-only. This is, by far,
 412  * the exception case. Normally, this function will be called when the
 413  * root filesystem is writable. In the unusual case where this is not
 414  * true, the configuration file is copied to the volatile file system
 415  * and is updated there until the root filesystem becomes writable. At
 416  * that time the file will be moved back to its proper location by
 417  * ipmgmt_db_restore_thread().
 418  */
 419 extern int
 420 ipmgmt_db_walk(db_wfunc_t *db_walk_func, void *db_warg, ipadm_db_op_t db_op)
 421 {
 422         int             err;
 423         boolean_t       writeop;
 424         mode_t          mode;
 425         pthread_t       tid;
 426         pthread_attr_t  attr;

 428         writeop = (db_op != IPADM_DB_READ);
 429         if (writeop) {
 430                 (void) pthread_rwlock_wrlock(&ipmgmt_dbconf_lock);
 431                 mode = IPADM_FILE_MODE;
 432         } else {
 433                 (void) pthread_rwlock_rdlock(&ipmgmt_dbconf_lock);
 434                 mode = 0;
 435         }

 437         /*
 438          * Did a previous write attempt fail? If so, don't even try to
 439          * read/write to IPADM_DB_FILE.
 440          */
 441         if (!ipmgmt_rdonly_root) {
 442                 err = ipadm_rw_db(db_walk_func, db_warg, IPADM_DB_FILE,
 443                     mode, db_op);
 444                 if (err != EROFS)
 445                         goto done;
 446         }

 448         /*
 449          * If we haven't already copied the file to the volatile
 450          * file system, do so. This should only happen on a failed
 451          * writeop(i.e., we have acquired the write lock above).
 452          */
 453         if (access(IPADM_VOL_DB_FILE, F_OK) != 0) {
 454                 assert(writeop);
 455                 err = ipmgmt_cpfile(IPADM_DB_FILE, IPADM_VOL_DB_FILE, B_TRUE);
 456                 if (err != 0)
 457                         goto done;
```



```
 458                 (void) pthread_attr_init(&attr);
 459                 (void) pthread_attr_setdetachstate(&attr,
 460                     PTHREAD_CREATE_DETACHED);
 461                 (void) pthread_attr_setname_np(&attr, "db_restore");
 462                 err = pthread_create(&tid, &attr, ipmgmt_db_restore_thread,
 463                     NULL);
 464                 (void) pthread_attr_destroy(&attr);
 465                 if (err != 0) {
 466                         (void) unlink(IPADM_VOL_DB_FILE);
 467                         goto done;
 468                 }
 469                 ipmgmt_rdonly_root = B_TRUE;
 470         }

 472         /*
 473          * Read/write from the volatile copy.
 474          */
 475         err = ipadm_rw_db(db_walk_func, db_warg, IPADM_VOL_DB_FILE,
 476             mode, db_op);
 477 done:
 478         (void) pthread_rwlock_unlock(&ipmgmt_dbconf_lock);
 479         return (err);
 480 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    5437 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/dtrace/test/tst/common/Makefile
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright (c) 2013, Joyent, Inc. All rights reserved.
# Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
# Copyright 2018 Joyent, Inc.
#

#
# WARNING: Do not include Makefile.ctf here. That will cause tests to
# break.
#

include $(SRC)/Makefile.master
include ../Makefile.com

SNOOPDIR = $(SRC)/cmd/cmd-inet/usr.sbin/snoop
SNOOPOBJS = nfs4_xdr.o
SNOOPSRCS = ${SNOOPOBJS:%.o=%.c}
CLOBBERFILES += nfs/$(SNOOPOBJS)

RPCSVCDIR = $(SRC)/head/rpcsvc
RPCSVCOBJS = nfs_prot.o
RPCSVCSRCS = ${RPCSVCOBJS:%o=%c}
CLOBBERFILES += nfs/$(RPCSVCOBJS) $(RPCSVCDIR)/$(RPCSVCSRCS)
CLOBBERFILES += usdt/forker.h usdt/lazyprobe.h

fasttrap/tst.fasttrap.exe := LDLIBS += -ldtrace
fasttrap/tst.stack.exe := LDLIBS += -ldtrace

sysevent/tst.post.exe := LDLIBS += -lsysevent
sysevent/tst.post_chan.exe := LDLIBS += -lsysevent

ustack/tst.bigstack.exe := COPTFLAG += -xO1
```



```
CSTD = $(CSTD_GNU99)

nfs/%.o: $(SNOOPDIR)/%.c
        $(COMPILE.c) -o $@ $< -I$(SNOOPDIR)
        $(POST_PROCESS_O)
nfs/tst.call.exe: nfs/tst.call.o nfs/$(SNOOPOBJS)
        $(LINK.c) -o $@ nfs/tst.call.o nfs/$(SNOOPOBJS) $(LDLIBS) -lnsl
        $(POST_PROCESS) ; $(STRIP_STABS)
$(RPCSVCDIR)/%.c: $(RPCSVCDIR)/%.x
        $(RPCGEN) -Cc $< > $@
nfs/$(RPCSVCOBJS): $(RPCSVCDIR)/$(RPCSVCSRCS)
        $(COMPILE.c) -o $@ $(RPCSVCDIR)/$(RPCSVCSRCS)
        $(POST_PROCESS_O)
nfs/tst.call3.exe: nfs/tst.call3.o nfs/$(RPCSVCOBJS)
        $(LINK.c) -o $@ nfs/tst.call3.o nfs/$(RPCSVCOBJS) \
        $(LDLIBS) -lnsl -lrpcsvc
        $(POST_PROCESS) ; $(STRIP_STABS)

json/tst.usdt.o: json/usdt.h

json/usdt.h: json/usdt.d
        $(DTRACE) -h -s json/usdt.d -o json/usdt.h

CLOBBERFILES += json/usdt.h

json/usdt.o: json/usdt.d json/tst.usdt.o
        $(COMPILE.d) -o json/usdt.o -s json/usdt.d json/tst.usdt.o

json/tst.usdt.exe: json/tst.usdt.o json/usdt.o
        $(LINK.c) -o json/tst.usdt.exe json/tst.usdt.o json/usdt.o $(LDLIBS)
        $(POST_PROCESS) ; $(STRIP_STABS)

#
# Tests that use the next three programs rely on the binaries having
# valid CTF data.
#
uctf/tst.aouttype.exe: uctf/tst.aouttype.c
        $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.aouttype.o uctf/tst.aouttype.c
        $(CTFCONVERT) -i -L VERSION uctf/tst.aouttype.o
        $(LINK.c) -o uctf/tst.aouttype.exe uctf/tst.aouttype.o $(LDLIBS)
        $(CTFMERGE) -L VERSION -o $@ uctf/tst.aouttype.o
        $(POST_PROCESS) ; $(STRIP_STABS)

uctf/tst.chasestrings.exe: uctf/tst.chasestrings.c
        $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.chasestrings.o uctf/tst.chasestrin
        $(CTFCONVERT) -i -L VERSION uctf/tst.chasestrings.o
        $(LINK.c) -o uctf/tst.chasestrings.exe uctf/tst.chasestrings.o $(LDLIBS)
        $(CTFMERGE) -L VERSION -o $@ uctf/tst.chasestrings.o
        $(POST_PROCESS) ; $(STRIP_STABS)

uctf/tst.printtype.exe: uctf/tst.printtype.c
        $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.printtype.o uctf/tst.printtype.c
        $(CTFCONVERT) -i -L VERSION uctf/tst.printtype.o
        $(LINK.c) -o uctf/tst.printtype.exe uctf/tst.printtype.o $(LDLIBS)
        $(CTFMERGE) -L VERSION -o $@ uctf/tst.printtype.o
        $(POST_PROCESS) ; $(STRIP_STABS)

#
# This program should never have any ctf data in it.
#
uctf/tst.libtype.exe:
        $(LINK.c) -o uctf/tst.libtype.exe uctf/tst.libtype.c $(LDLIBS)
        $(POST_PROCESS) ; $(STRIP_STABS)

usdt/tst.args.exe: usdt/tst.args.o usdt/args.o
        $(LINK.c) -o usdt/tst.args.exe usdt/tst.args.o usdt/args.o $(LDLIBS)
```

```
126         $(POST_PROCESS) ; $(STRIP_STABS)

128 usdt/args.o: usdt/args.d usdt/tst.args.o
129         $(COMPILE.d) -o usdt/args.o -s usdt/args.d usdt/tst.args.o

131 usdt/tst.argmap.exe: usdt/tst.argmap.o usdt/argmap.o
132         $(LINK.c) -o usdt/tst.argmap.exe \
133             usdt/tst.argmap.o usdt/argmap.o $(LDLIBS)
134         $(POST_PROCESS) ; $(STRIP_STABS)

136 usdt/argmap.o: usdt/argmap.d usdt/tst.argmap.o
137         $(COMPILE.d) -o usdt/argmap.o -s usdt/argmap.d usdt/tst.argmap.o

139 usdt/tst.forker.exe: usdt/tst.forker.o usdt/forker.o
140         $(LINK.c) -o usdt/tst.forker.exe \
141             usdt/tst.forker.o usdt/forker.o $(LDLIBS)
142         $(POST_PROCESS) ; $(STRIP_STABS)

144 usdt/forker.o: usdt/forker.d usdt/tst.forker.o
145         $(COMPILE.d) -o usdt/forker.o -s usdt/forker.d usdt/tst.forker.o

147 usdt/tst.forker.o: usdt/forker.h

149 usdt/forker.h: usdt/forker.d
150         $(DTRACE) -h -s usdt/forker.d -o usdt/forker.h

152 usdt/tst.lazyprobe.exe: usdt/tst.lazyprobe.o usdt/lazyprobe.o
153         $(LINK.c) -o usdt/tst.lazyprobe.exe \
154             usdt/tst.lazyprobe.o usdt/lazyprobe.o $(LDLIBS)
155         $(POST_PROCESS) ; $(STRIP_STABS)

157 usdt/lazyprobe.o: usdt/lazyprobe.d usdt/tst.lazyprobe.o
158         $(COMPILE.d) -xlazyload -o usdt/lazyprobe.o \
159             -s usdt/lazyprobe.d usdt/tst.lazyprobe.o

161 usdt/tst.lazyprobe.o: usdt/lazyprobe.h

163 usdt/lazyprobe.h: usdt/lazyprobe.d
164         $(DTRACE) -h -s usdt/lazyprobe.d -o usdt/lazyprobe.h

166 SUBDIRS = java_api
167 include ../../Makefile.subdirs
```

```
**********************************************************
    1902 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/dtrace/test/tst/common/threadname/tst.threadname.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2018 Joyent, Inc.
 */

/*
 * All we're doing is constantly modifying a thread name while DTrace is
 * watching us, making sure we don't break.
 */

#include <sys/fcntl.h>
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>

#define NR_THREADS (100)
#define RUNTIME (30) /* seconds */

static void
random_ascii(char *buf, size_t bufsize)
{
        char table[] = "abcdefghijklmnopqrstuvwxyz"
            "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ,.-#'?!";
        size_t len = rand() % bufsize;

        bzero(buf, bufsize);

        for (size_t i = 0; i < len; i++) {
                buf[i] = table[rand() % (sizeof (table) - 1)];
        }
}

static void
busy()
{
        struct timeval tv1;
        struct timeval tv2;

        if (gettimeofday(&tv1, NULL) != 0)
                abort();

        for (;;) {
                static volatile int i;
                for (i = 0; i < 2000000; i++)
                        ;

                if (gettimeofday(&tv2, NULL) != 0)
                        abort();

                /* janky, but we don't care */
```



```
                if (tv2.tv_sec != tv1.tv_sec)
                        return;
        }
}

static void *
thread(void *arg)
{
        char name[PTHREAD_MAX_NAMELEN_NP];

        for (size_t i = 0; ; i++) {
                random_ascii(name, sizeof (name));

                if ((i % 100) == 0) {
                        if (pthread_setname_np(pthread_self(), NULL) != 0)
                                abort();
                } else {
                        (void) pthread_setname_np(pthread_self(), name);
                }

                busy();
        }

        return (NULL);
}

int
main(int argc, char **argv)
{
        pthread_t tids[NR_THREADS];

        for (size_t i = 0; i < NR_THREADS; i++) {
                if (pthread_create(&tids[i], NULL, thread, NULL) != 0)
                        exit(EXIT_FAILURE);
        }

        sleep(RUNTIME);
        exit(EXIT_SUCCESS);
}
```

```
**********************************************************
     600 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/dtrace/test/tst/common/threadname/tst.threadname.d
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2018 Joyent, Inc.
 */

#pragma D option quiet

profile-10ms /pid == $1 && threadname == "unlikely"/
{
        surprising++;
}

syscall::rexit:entry /pid == $1/
{
        exit(arg0);
}
```

```
**********************************************************
    2214 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/halt/Makefile
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2016 Toomas Soome <tsoome@me.com>
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2018 Joyent, Inc.
#

PROG = halt

include ../Makefile.cmd

#
# Currently Fast Reboot is only supported on x86.
# A new propery config/uadmin_boot_archive_sync is added to
# boot-config service. Which needs a support on sparc also.
#
sparc_SUBDIRS = smf.sparc
i386_SUBDIRS = smf.i386
SUBDIRS = $($(MACH)_SUBDIRS)

ROOTLINKS = $(ROOTUSRSBIN)/poweroff $(ROOTUSRSBIN)/reboot
ROOTSYMLINKS= $(ROOTETC)/halt $(ROOTETC)/reboot

FILEMODE = 0755

.KEEP_STATE:

CPPFLAGS += -I../../lib/libzpool/common
CPPFLAGS += -I../../lib/libscf/inc
CPPFLAGS += -I../../uts/common/fs/zfs

CERRWARN += -_gcc=-Wno-unused-label

LDLIBS += -lbsm -lscf -lzfs -lgen
LDLIBS_i386 +=  -lbe
LDLIBS +=       $(LDLIBS_$(MACH))

CLOBBERFILES += $(ROOTLINKS) $(ROOTSYMLINKS)

all     :=      TARGET = all
install :=      TARGET = install
```



```
clean   :=      TARGET = clean
clobber :=      TARGET = clobber
lint    :=      TARGET = lint
lint    :=      LINTFLAGS += -u
lint    :=      LINTFLAGS = -u


all: $(PROG)

install: all $(ROOTUSRSBINPROG) $(ROOTLINKS) $(ROOTSYMLINKS) $(SUBDIRS)

$(ROOTLINKS): $(ROOTUSRSBINPROG)
        $(RM) $@
        $(LN) $(ROOTUSRSBINPROG) $@

$(ROOTSYMLINKS):
        $(RM) $@
        $(SYMLINK) ../usr/sbin/$(PROG) $@

$(SUBDIRS):     FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

clean:

clobber: $(SUBDIRS)

lint:   lint_PROG

check: $(CHKMANIFEST)

FRC:

include ../Makefile.targ
```

```
**********************************************************
   16876 Mon Oct 15 13:24:01 2018
new/usr/src/cmd/ldapcachemgr/cachemgr_change.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  *
  25  * Copyright 2018 Joyent, Inc.
  26  */

  28 #include <strings.h>
  29 #include <stdlib.h>
  30 #include <syslog.h>
  31 #include <errno.h>
  32 #include <libintl.h>
  33 #include <door.h>
  34 #include <sys/types.h>
  35 #include <sys/stat.h>
  36 #include <fcntl.h>
  37 #include <procfs.h>
  38 #include <pthread.h>
  39 #include "cachemgr.h"

  41 extern admin_t  current_admin;

  43 #define CLEANUP_WAIT_TIME 60

  45 typedef enum cleanup_type {
  46         CLEANUP_ALL     = 1,
  47         CLEANUP_BY_PID  = 2
  48 } cleanup_type_t;
_____unchanged_portion_omitted_
 529 /*
 530  * If arg is NULL, it loops forever,
 531  * else it calls cleanup_threads once and exits.
 532  */
 533 void *
 534 chg_cleanup_waiting_threads(void *arg)
 535 {
 536         cleanup_op_t *op = (cleanup_op_t *)arg;
 537         cleanup_type_t type = 0;
 538         pid_t   pid;
 539         int     always = 1, waiting;
```



```
 541         (void) pthread_setname_np(pthread_self(), "chg_cleanup_thr");

 543         if (op == NULL) {
 544                 waiting = 1;
 545                 type = CLEANUP_ALL;
 546                 pid = 0;
 547         } else {
 548                 waiting = 0;
 549                 type = op->type;
 550                 pid = op->pid;
 551         }

 553         while (always) {
 554                 if (waiting)
 555                         (void) sleep(CLEANUP_WAIT_TIME);
 556                 cleanup_threads(&chg, pid, type);
 557                 if (!waiting)
 558                         break;
 559         }

 561         if (op)
 562                 free(op);

 564         thr_exit(NULL);
 565         return (NULL);
 566 }
_____unchanged_portion_omitted_
```

```
**********************************************************
   16253 Mon Oct 15 13:24:05 2018
new/usr/src/cmd/ldapcachemgr/cachemgr_discovery.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 */

#ifdef SLP

/*
 * This file contains all the dynamic server discovery functionality
 * for ldap_cachemgr. SLP is used to query the network for any changes
 * in the set of deployed LDAP servers.
 *
 * The algorithm used is outlined here:
 *
 *   1. Find all naming contexts with SLPFindAttrs. (See
 *      find_all_contexts())
 *   2. For each context, find all servers which serve that context
 *      with SLPFindSrvs. (See foreach_context())
 *   3. For each server, retrieve that server's attributes with
 *      SLPFindAttributes. (See foreach_server())
 *   4. Aggregate the servers' attributes into a config object. There
 *      is one config object associated with each context found in
 *      step 1. (See aggregate_attrs())
 *   5. Update the global config cache for each found context and its
 *      associated servers and attributes. (See update_config())
 *
 * The entry point for ldap_cachemgr is discover(). The actual entry
 * point into the discovery routine is find_all_contexts(); the
 * code thereafter is actually not specific to LDAP, and could also
 * be used to discover YP, or any other server which conforms
 * to the SLP Naming and Directory abstract service type.
 *
 * find_all_attributes() takes as parameters three callback routines
 * which are used to report all information back to the caller. The
 * signatures and synopses of these routines are:
 *
 * void *get_cfghandle(const char *domain);
 *
```



```
 *   Returns an opaque handle to a configuration object specific
 *   to the 'domain' parameter. 'domain' will be a naming context
 *   string, i.e. foo.bar.sun.com ( i.e. a secure-RPC domain-
 *   name).
 *
 * void aggregate(void *handle, const char *tag, const char *value);
 *
 *   Adds this tag / value pair to the set of aggregated attributes
 *   associated with the given handle.
 *
 * void set_cfghandle(void *handle);
 *
 *   Sets and destroys the config object; SLP will no longer attempt
 *   to use this handle after this call. Thus, this call marks the
 *   end of configuration information for this handle.
 */

#include <stdio.h>
#include <slp.h>
#include <stdlib.h>
#include <string.h>
#include <door.h>
#include <unistd.h>
#include "ns_sldap.h"
#include "ns_internal.h"
#include "cachemgr.h"

#define ABSTYPE         "service:naming-directory"
#define CONTEXT_ATTR    "naming-context"
#define LDAP_DOMAIN_ATTR "x-sun-rpcdomain"

/* The configuration cookie passed along through all SLP callbacks. */
struct config_cookie {
        SLPHandle       h;              /* An open SLPHandle */
        const char      *type;          /* The full service type to use */
        char            *scopes;        /* A list of scopes to use */
        const char      *context_attr;  /* Which attr to use for the ctx */
        void            *cache_cfg;     /* caller-supplied config object */
        void *(*get_cfghandle)(const char *);
        void (*aggregate)(void *, const char *, const char *);
        void (*set_cfghandle)(void *);
};
```

*_____unchanged_portion_omitted_*

```
/*
 * This is the ldap_cachemgr entry point into SLP dynamic discovery. The
 * parameter 'r' should be a pointer to an unsigned int containing
 * the requested interval at which the network should be queried.
 */
void
discover(void *r) {
void discover(void *r) {
        unsigned short reqrefresh = *((unsigned int *)r);

        (void) pthread_setname_np(pthread_self(), "discover");

        for (;;) {
            find_all_contexts("ldap",
                                __cache_get_cfghandle,
                                __cache_aggregate_params,
                                __cache_set_cfghandle);

            if (current_admin.debug_level >= DBG_ALL) {
                (void) logit(
                        "dynamic discovery: using refresh interval %d\n",
                        reqrefresh);
```

```
560             }

562             (void) sleep(reqrefresh);
563         }
564 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   78062 Mon Oct 15 13:24:09 2018
new/usr/src/cmd/ldapcachemgr/cachemgr_getldap.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
 *
 * Copyright 2018 Joyent, Inc.
 */

#include <assert.h>
#include <errno.h>
#include <memory.h>
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <libintl.h>
#include <syslog.h>
#include <sys/door.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <synch.h>
#include <pthread.h>
#include <unistd.h>
#include <lber.h>
#include <ldap.h>
#include <ctype.h>      /* tolower */
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <ucred.h>
#include "cachemgr.h"
#include "solaris-priv.h"
#include "ns_connmgmt.h"

static rwlock_t ldap_lock = DEFAULTRWLOCK;
static int      sighup_update = FALSE;
extern admin_t  current_admin;

extern int is_root_or_all_privs(char *dc_str, ucred_t **ucp);
```



```
/* variables used for SIGHUP wakeup on sleep */
static mutex_t                  sighuplock;
static cond_t                   cond;

/* refresh time statistics */
static time_t   prev_refresh_time = 0;

/* variables used for signaling parent process */
static mutex_t  sig_mutex;
static int      signal_done = FALSE;

/* TCP connection timeout (in milliseconds) */
static int tcptimeout = NS_DEFAULT_BIND_TIMEOUT * 1000;

#ifdef SLP
extern int      use_slp;
#endif /* SLP */

/* nis domain information */
#define _NIS_FILTER             "objectclass=nisDomainObject"
#define _NIS_DOMAIN             "nisdomain"

#define CACHESLEEPTIME          600
/*
 * server list refresh delay when in "no server" mode
 * (1 second)
 */
#define REFRESH_DELAY_WHEN_NO_SERVER    1

typedef enum {
        INFO_OP_CREATE          = 0,
        INFO_OP_DELETE          = 1,
        INFO_OP_REFRESH         = 2,
        INFO_OP_REFRESH_WAIT    = 3,
        INFO_OP_GETSERVER       = 4,
        INFO_OP_GETSTAT         = 5,
        INFO_OP_REMOVESERVER    = 6
} info_op_t;
```

unchanged_portion_omitted_

```
static int
getldap_serverInfo_op(info_op_t op, char *input, char **output)
{

        static rwlock_t         info_lock = DEFAULTRWLOCK;
        static rwlock_t         info_lock_old = DEFAULTRWLOCK;
        static mutex_t          info_mutex;
        static cond_t           info_cond;
        static int              creating = FALSE;
        static int              refresh_ttl = 0;
        static int              sec_to_refresh = 0;
        static int              in_no_server_mode = FALSE;

        static server_info_t    *serverInfo = NULL;
        static server_info_t    *serverInfo_old = NULL;
        server_info_t           *serverInfo_1;
        int                     is_creating;
        int                     err, no_server_good = FALSE;
        int                     server_removed = FALSE;
        int                     fall_thru = FALSE;
        static struct timespec  timeout;
        struct timespec         new_timeout;
        struct timeval          tp;
        static time_t           prev_refresh = 0, next_refresh = 0;
        ns_server_status_t              changed = 0;
```

```
1636          (void) pthread_setname_np(pthread_self(), "getldap_serverinfo");

1638          if (current_admin.debug_level >= DBG_ALL) {
1639                  logit("getldap_serverInfo_op()...\n");
1640          }
1641          switch (op) {
1642          case INFO_OP_CREATE:
1643                  if (current_admin.debug_level >= DBG_ALL) {
1644                          logit("operation is INFO_OP_CREATE...\n");
1645                  }

1647                  /*
1648                   * indicate that the server info is being
1649                   * (re)created, so that the refresh thread
1650                   * will not refresh the info list right
1651                   * after the list got (re)created
1652                   */
1653                  (void) mutex_lock(&info_mutex);
1654                  is_creating = creating;
1655                  creating = TRUE;
1656                  (void) mutex_unlock(&info_mutex);

1658                  if (is_creating)
1659                          break;
1660                  /*
1661                   * create an empty info list
1662                   */
1663                  (void) getldap_init_serverInfo(&serverInfo_1);
1664                  /*
1665                   * exit if list not created
1666                   */
1667                  if (serverInfo_1 == NULL) {
1668                          (void) mutex_lock(&info_mutex);
1669                          creating = FALSE;
1670                          (void) mutex_unlock(&info_mutex);
1671                          break;
1672                  }
1673                  /*
1674                   * make the new server info available:
1675                   * use writer lock here, so that the switch
1676                   * is done after all the reader locks have
1677                   * been released.
1678                   */
1679                  (void) rw_wrlock(&info_lock);
1680                  serverInfo = serverInfo_1;
1681                  /*
1682                   * if this is the first time
1683                   * the server list is being created,
1684                   * (i.e., serverInfo_old is NULL)
1685                   * make the old list same as the new
1686                   * so the GETSERVER code can do its work
1687                   */
1688                  if (serverInfo_old == NULL)
1689                          serverInfo_old = serverInfo_1;
1690                  (void) rw_unlock(&info_lock);

1692                  /*
1693                   * fill the new info list
1694                   */
1695                  (void) rw_rdlock(&info_lock);
1696                  /* reset bind time (tcptimeout) */
1697                  (void) getldap_set_serverInfo(serverInfo, 1, INFO_OP_CREATE);

1699                  (void) mutex_lock(&info_mutex);
1700                  /*
1701                   * set cache manager server list TTL,
```



```
1702                   * set refresh_ttl to zero to indicate a fresh one
1703                   */
1704                  refresh_ttl = 0;
1705                  (void) getldap_set_refresh_ttl(serverInfo,
1706                      &refresh_ttl, &no_server_good);
1707                  sec_to_refresh = refresh_ttl;

1709                  /* statistics: previous refresh time */
1710                  if (gettimeofday(&tp, NULL) == 0)
1711                          prev_refresh = tp.tv_sec;

1713                  creating = FALSE;

1715                  /*
1716                   * if no server found or available,
1717                   * tell the server info refresh thread
1718                   * to start the "no-server" refresh loop
1719                   * otherwise reset the in_no_server_mode flag
1720                   */
1721                  if (no_server_good) {
1722                          sec_to_refresh = 0;
1723                          in_no_server_mode = TRUE;
1724                  } else
1725                          in_no_server_mode = FALSE;
1726                  /*
1727                   * awake the sleeping refresh thread
1728                   */
1729                  (void) cond_signal(&info_cond);

1731                  (void) mutex_unlock(&info_mutex);
1732                  (void) rw_unlock(&info_lock);

1734                  /*
1735                   * delete the old server info
1736                   */
1737                  (void) rw_wrlock(&info_lock_old);
1738                  if (serverInfo_old != serverInfo)
1739                          (void) getldap_destroy_serverInfo(serverInfo_old);
1740                  /*
1741                   * serverInfo_old needs to be the same as
1742                   * serverinfo now.
1743                   * it will be used by GETSERVER processing.
1744                   */
1745                  serverInfo_old = serverInfo;
1746                  (void) rw_unlock(&info_lock_old);
1747                  break;
1748          case INFO_OP_DELETE:
1749                  if (current_admin.debug_level >= DBG_ALL) {
1750                          logit("operation is INFO_OP_DELETE...\n");
1751                  }
1752                  /*
1753                   * use writer lock here, so that the delete would
1754                   * not start until all the reader locks have
1755                   * been released.
1756                   */
1757                  (void) rw_wrlock(&info_lock);
1758                  if (serverInfo)
1759                          (void) getldap_destroy_serverInfo(serverInfo);
1760                  serverInfo = NULL;
1761                  (void) rw_unlock(&info_lock);
1762                  break;
1763          case INFO_OP_REFRESH:
1764                  if (current_admin.debug_level >= DBG_SERVER_LIST_REFRESH) {
1765                          logit("operation is INFO_OP_REFRESH...\n");
1766                  }
1767                  /*
```

```
1768                 * if server info is currently being
1769                 * (re)created, do nothing
1770                 */
1771                (void) mutex_lock(&info_mutex);
1772                is_creating = creating;
1773                (void) mutex_unlock(&info_mutex);
1774                if (is_creating)
1775                        break;

1777                (void) rw_rdlock(&info_lock);
1778                if (serverInfo) {
1779                        /* do not reset bind time (tcptimeout) */
1780                        (void) getldap_set_serverInfo(serverInfo, 0,
1781                            INFO_OP_REFRESH);

1783                        (void) mutex_lock(&info_mutex);

1785                        /* statistics: previous refresh time */
1786                        if (gettimeofday(&tp, NULL) == 0)
1787                                prev_refresh = tp.tv_sec;
1788                        /*
1789                         * set cache manager server list TTL
1790                         */
1791                        (void) getldap_set_refresh_ttl(serverInfo,
1792                            &refresh_ttl, &no_server_good);
1793                        /*
1794                         * if no good server found,
1795                         * tell the server info refresh thread
1796                         * to start the "no-server" refresh loop
1797                         * otherwise reset the in_no_server_mode flag
1798                         */
1799                        if (no_server_good) {
1800                                in_no_server_mode = TRUE;
1801                                sec_to_refresh = 0;
1802                        } else {
1803                                in_no_server_mode = FALSE;
1804                                sec_to_refresh = refresh_ttl;
1805                        }
1806                        if (current_admin.debug_level >=
1807                            DBG_SERVER_LIST_REFRESH) {
1808                                logit("getldap_serverInfo_op("
1809                                    "INFO_OP_REFRESH):"
1810                                    " seconds refresh: %d second(s)....\n",
1811                                    sec_to_refresh);
1812                        }
1813                        (void) mutex_unlock(&info_mutex);
1814                }
1815                (void) rw_unlock(&info_lock);

1817                break;
1818        case INFO_OP_REFRESH_WAIT:
1819                if (current_admin.debug_level >= DBG_SERVER_LIST_REFRESH) {
1820                        logit("operation is INFO_OP_REFRESH_WAIT...\n");
1821                }
1822                (void) cond_init(&info_cond, NULL, NULL);
1823                (void) mutex_lock(&info_mutex);
1824                err = 0;
1825                while (err != ETIME) {
1826                        int sleeptime;
1827                        /*
1828                         * if need to go into the "no-server" refresh
1829                         * loop, set timout value to
1830                         * REFRESH_DELAY_WHEN_NO_SERVER
1831                         */
1832                        if (sec_to_refresh == 0) {
1833                                sec_to_refresh = refresh_ttl;
```



```
1834                                timeout.tv_sec = time(NULL) +
1835                                    REFRESH_DELAY_WHEN_NO_SERVER;
1836                                sleeptime = REFRESH_DELAY_WHEN_NO_SERVER;
1837                                if (current_admin.debug_level >=
1838                                    DBG_SERVER_LIST_REFRESH) {
1839                                        logit("getldap_serverInfo_op("
1840                                            "INFO_OP_REFRESH_WAIT):"
1841                                            " entering no-server "
1842                                            "refresh loop...\n");
1843                                }
1844                        } else {
1845                                timeout.tv_sec = time(NULL) + sec_to_refresh;
1846                                sleeptime = sec_to_refresh;
1847                        }
1848                        timeout.tv_nsec = 0;

1850                        /* statistics: next refresh time */
1851                        next_refresh = timeout.tv_sec;

1853                        if (current_admin.debug_level >=
1854                            DBG_SERVER_LIST_REFRESH) {
1855                                logit("getldap_serverInfo_op("
1856                                    "INFO_OP_REFRESH_WAIT):"
1857                                    " about to sleep for %d second(s)...\n",
1858                                    sleeptime);
1859                        }
1860                        err = cond_timedwait(&info_cond,
1861                            &info_mutex, &timeout);
1862                }
1863                (void) cond_destroy(&info_cond);
1864                (void) mutex_unlock(&info_mutex);
1865                break;
1866        case INFO_OP_GETSERVER:
1867                if (current_admin.debug_level >= DBG_ALL) {
1868                        logit("operation is INFO_OP_GETSERVER...\n");
1869                }
1870                *output = NULL;
1871                /*
1872                 * GETSERVER processing always use
1873                 * serverInfo_old to retrieve server infomation.
1874                 * serverInfo_old is equal to serverInfo
1875                 * most of the time, except when a new
1876                 * server list is being created.
1877                 * This is why the check for is_creating
1878                 * is needed below.
1879                 */
1880                (void) rw_rdlock(&info_lock_old);

1882                if (serverInfo_old == NULL) {
1883                        (void) rw_unlock(&info_lock_old);
1884                        break;
1885                } else
1886                        (void) getldap_get_serverInfo(serverInfo_old,
1887                            input, output, &server_removed);
1888                (void) rw_unlock(&info_lock_old);

1890                /*
1891                 * Return here and let remove server thread do its job in
1892                 * another thread. It executes INFO_OP_REMOVESERVER code later.
1893                 */
1894                if (server_removed)
1895                        break;

1897                fall_thru = TRUE;

1899                /* FALL THROUGH */
```

```
1901         case INFO_OP_REMOVESERVER:
1902                 /*
1903                  * INFO_OP_GETSERVER and INFO_OP_REMOVESERVER share the
1904                  * following code except (!fall thru) part.
1905                  */

1907                 /*
1908                  * if server info is currently being
1909                  * (re)created, do nothing
1910                  */

1912                 (void) mutex_lock(&info_mutex);
1913                 is_creating = creating;
1914                 (void) mutex_unlock(&info_mutex);
1915                 if (is_creating)
1916                         break;

1918                 if (!fall_thru) {
1919                         if (current_admin.debug_level >= DBG_ALL)
1920                                 logit("operation is INFO_OP_REMOVESERVER...\n");
1921                         (void) rw_rdlock(&info_lock_old);
1922                         changed = set_server_status(input, serverInfo_old);
1923                         (void) rw_unlock(&info_lock_old);
1924                         if (changed)
1925                                 create_buf_and_notify(input, changed);
1926                         else
1927                                 break;
1928                 }

1930                 /*
1931                  * set cache manager server list TTL if necessary
1932                  */
1933                 if (*output == NULL || changed) {
1934                         (void) rw_rdlock(&info_lock);
1935                         (void) mutex_lock(&info_mutex);

1937                         (void) getldap_set_refresh_ttl(serverInfo,
1938                             &refresh_ttl, &no_server_good);

1940                         /*
1941                          * if no good server found, need to go into
1942                          * the "no-server" refresh loop
1943                          * to find a server as soon as possible
1944                          * otherwise reset the in_no_server_mode flag
1945                          */
1946                         if (no_server_good) {
1947                                 /*
1948                                  * if already in no-server mode,
1949                                  * don't brother
1950                                  */
1951                                 if (in_no_server_mode == FALSE) {
1952                                         sec_to_refresh = 0;
1953                                         in_no_server_mode = TRUE;
1954                                         (void) cond_signal(&info_cond);
1955                                 }
1956                                 (void) mutex_unlock(&info_mutex);
1957                                 (void) rw_unlock(&info_lock);
1958                                 break;
1959                         } else {
1960                                 in_no_server_mode = FALSE;
1961                                 sec_to_refresh = refresh_ttl;
1962                         }
1963                         /*
1964                          * if the refresh thread will be timed out
1965                          * longer than refresh_ttl seconds,
```



```
1966                          * wake it up to make it wait on the new
1967                          * time out value
1968                          */
1969                         new_timeout.tv_sec = time(NULL) + refresh_ttl;
1970                         if (new_timeout.tv_sec < timeout.tv_sec)
1971                                 (void) cond_signal(&info_cond);

1973                         (void) mutex_unlock(&info_mutex);
1974                         (void) rw_unlock(&info_lock);
1975                 }
1976                 break;
1977         case INFO_OP_GETSTAT:
1978                 if (current_admin.debug_level >= DBG_ALL) {
1979                         logit("operation is INFO_OP_GETSTAT...\n");
1980                 }
1981                 *output = NULL;
1982                 (void) rw_rdlock(&info_lock);
1983                 if (serverInfo) {
1984                         (void) getldap_get_server_stat(serverInfo,
1985                             output, &prev_refresh, &next_refresh);
1986                 }
1987                 (void) rw_unlock(&info_lock);
1988                 break;
1989         default:
1990                 logit("getldap_serverInfo_op(): "
1991                     "invalid operation code (%d).\n", op);
1992                 return (-1);
1993                 break;
1994         }
1995         return (NS_LDAP_SUCCESS);
1996 }
______unchanged_portion_omitted_

2535 void
2536 getldap_refresh()
2537 {
2538         struct timespec timeout;
2539         int             sleeptime;
2540         struct timeval  tp;
2541         long            expire = 0;
2542         void            **paramVal = NULL;
2543         ns_ldap_error_t *errorp;
2544         int             always = 1, err;
2545         int             first_time = 1;
2546         int             sig_done = 0;
2547         int             dbg_level;

2549         (void) pthread_setname_np(pthread_self(), "getldap_refresh");

2551         if (current_admin.debug_level >= DBG_ALL) {
2552                 logit("getldap_refresh()...\n");
2553         }

2555         /*
2556          * wait for an available server
2557          */
2558         while (sig_done == 0) {
2559                 (void) mutex_lock(&sig_mutex);
2560                 sig_done = signal_done;
2561                 (void) mutex_unlock(&sig_mutex);
2562         }

2564         (void) __ns_ldap_setServer(TRUE);
2565         while (always) {
2566                 dbg_level = current_admin.debug_level;
2567                 (void) rw_rdlock(&ldap_lock);
```

```
2568                sleeptime = current_admin.ldap_stat.ldap_ttl;
2569                if (dbg_level >= DBG_PROFILE_REFRESH) {
2570                        logit("getldap_refresh: current profile TTL is %d "
2571                        "seconds\n", current_admin.ldap_stat.ldap_ttl);
2572                }
2573                if (gettimeofday(&tp, NULL) == 0) {
2574                        if ((__ns_ldap_getParam(NS_LDAP_EXP_P,
2575                            &paramVal, &errorp) == NS_LDAP_SUCCESS) &&
2576                            paramVal != NULL &&
2577                            (char *)*paramVal != NULL) {
2578                                errno = 0;
2579                                expire = atol((char *)*paramVal);
2580                                (void) __ns_ldap_freeParam(&paramVal);
2581                                if (errno == 0) {
2582                                        if (expire == 0) {
2583                                                first_time = 0;
2584                                                (void) rw_unlock(&ldap_lock);
2585                                                (void) cond_init(&cond,
2586                                                    NULL, NULL);
2587                                                (void) mutex_lock(&sighuplock);
2588                                                timeout.tv_sec =
2589                                                    CACHESLEEPTIME;
2590                                                timeout.tv_nsec = 0;
2591                                                if (dbg_level >=
2592                                                    DBG_PROFILE_REFRESH) {
2593                                                        logit("getldap_refresh:"
2594                                                            "(1)about to sleep"
2595                                                            " for %d seconds\n",
2596                                                            CACHESLEEPTIME);
2597                                                }
2598                                                err = cond_reltimedwait(&cond,
2599                                                    &sighuplock, &timeout);
2600                                                (void) cond_destroy(&cond);
2601                                                (void) mutex_unlock(
2602                                                    &sighuplock);
2603                                                /*
2604                                                 * if woke up by
2605                                                 * getldap_revalidate(),
2606                                                 * do update right away
2607                                                 */
2608                                                if (err == ETIME)
2609                                                        continue;
2610                                                else {
2611                                                        /*
2612                                                         * if load
2613                                                         * configuration failed
2614                                                         * don't do update
2615                                                         */
2616                                                        if (load_config())
2617                                                                perform_update
2618                                                                    ();
2619                                                        continue;
2620                                                }
2621                                        }
2622                                        sleeptime = expire - tp.tv_sec;
2623                                        if (dbg_level >= DBG_PROFILE_REFRESH) {
2624                                                logit("getldap_refresh: expire "
2625                                                    "time = %ld\n", expire);
2626                                        }

2628                                }
2629                        }
2630                }

2632                (void) rw_unlock(&ldap_lock);
```



```
2634                /*
2635                 * if this is the first time downloading
2636                 * the profile or expire time already passed,
2637                 * do not wait, do update
2638                 */
2639                if (first_time == 0 && sleeptime > 0) {
2640                        if (dbg_level >= DBG_PROFILE_REFRESH) {
2641                                logit("getldap_refresh: (2)about to sleep "
2642                                "for %d seconds\n", sleeptime);
2643                        }
2644                        (void) cond_init(&cond, NULL, NULL);
2645                        (void) mutex_lock(&sighuplock);
2646                        timeout.tv_sec = sleeptime;
2647                        timeout.tv_nsec = 0;
2648                        err = cond_reltimedwait(&cond,
2649                            &sighuplock, &timeout);
2650                        (void) cond_destroy(&cond);
2651                        (void) mutex_unlock(&sighuplock);
2652                }
2653                /*
2654                 * if load concfiguration failed
2655                 * don't do update
2656                 */
2657                if (load_config())
2658                        perform_update();
2659                first_time = 0;
2660        }
2661 }
______unchanged_portion_omitted_

2886 /*
2887  * The thread is spawned to do contact_server() so it won't be blocking
2888  * getldap_serverInfo_op(INFO_OP_GETSERVER, ...) case.
2889  * After contact_server() is done, it calls
2890  * getldap_serverInfo_op(INFO_OP_REMOVESERVER, ...) to return to the remaining
2891  * program flow. It's meant to maintain the original program flow yet be
2892  * non-blocking when it's contacting server.
2893  */
2894 static void *
2895 remove_server_thread(void *arg)
2896 {
2897         char *addr = (char *)arg, *out = NULL;
2898         int up;
2899         rm_svr_t rms;

2901         (void) pthread_setname_np(pthread_self(), "remove_server");

2903         up = contact_server(addr);

2905         rms.addr = addr;
2906         rms.up = up;

2908         (void) getldap_serverInfo_op(INFO_OP_REMOVESERVER, (char *)&rms, &out);

2910         free(addr);

2912         thr_exit(NULL);
2913         return (NULL);
2914 }
______unchanged_portion_omitted_
```

```
**********************************************************
  148692 Mon Oct 15 13:24:15 2018
new/usr/src/cmd/mdb/common/mdb/mdb_proc.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
______unchanged_portion_omitted_

1306 static void
1307 pt_thread_name(mdb_tgt_t *t, mdb_tgt_tid_t tid, char *buf, size_t bufsize)
1308 {
1309         char name[THREAD_NAME_MAX];

1311         buf[0] = '\0';

1313         if (t->t_pshandle == NULL ||
1314             Plwp_getname(t->t_pshandle, tid, name, sizeof (name)) != 0 ||
1315             name[0] == '\0') {
1316                 (void) mdb_snprintf(buf, bufsize, "%lu", tid);
1317                 return;
1318         }

1320         (void) mdb_snprintf(buf, bufsize, "%lu [%s]", tid, name);
1321 }

1323 static int
1324 pt_findstack(uintptr_t tid, uint_t flags, int argc, const mdb_arg_t *argv)
1325 {
1326         mdb_tgt_t *t = mdb.m_target;
1327         mdb_tgt_gregset_t gregs;
1328         int showargs = 0;
1329         int count;
1330         uintptr_t pc, sp;
1331         char name[128];

1333         if (!(flags & DCMD_ADDRSPEC))
1334                 return (DCMD_USAGE);

1336         count = mdb_getopts(argc, argv, 'v', MDB_OPT_SETBITS, TRUE, &showargs,
1337             NULL);
1338         argc -= count;
1339         argv += count;

1341         if (argc > 1 || (argc == 1 && argv->a_type != MDB_TYPE_STRING))
1342                 return (DCMD_USAGE);

1344         if (PTL_GETREGS(t, tid, gregs.gregs) != 0) {
1345                 mdb_warn("failed to get register set for thread %p", tid);
1346                 return (DCMD_ERR);
1347         }

1349         pc = gregs.gregs[R_PC];
1350 #if defined(__i386) || defined(__amd64)
1351         sp = gregs.gregs[R_FP];
1352 #else
1353         sp = gregs.gregs[R_SP];
1354 #endif

1356         pt_thread_name(t, tid, name, sizeof (name));

1358         mdb_printf("stack pointer for thread %s: %p\n", name, sp);
1337         mdb_printf("stack pointer for thread %p: %p\n", tid, sp);
1359         if (pc != 0)
1360                 mdb_printf("[ %0?lr %a() ]\n", sp, pc);

1362         (void) mdb_inc_indent(2);
```



```
1363         mdb_set_dot(sp);

1365         if (argc == 1)
1366                 (void) mdb_eval(argv->a_un.a_str);
1367         else if (showargs)
1368                 (void) mdb_eval("<.$C");
1369         else
1370                 (void) mdb_eval("<.$C0");

1372         (void) mdb_dec_indent(2);
1373         return (DCMD_OK);
1374 }
______unchanged_portion_omitted_
```

```
**********************************************************
   21740 Mon Oct 15 13:24:18 2018
new/usr/src/cmd/mdb/common/modules/genunix/findstack.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2013, Josef 'Jeff' Sipek <jeffpc@josefsipek.net>
 * Copyright 2018 Joyent, Inc.
 */

#include <mdb/mdb_modapi.h>
#include <mdb/mdb_ctf.h>

#include <sys/types.h>
#include <sys/regset.h>
#include <sys/stack.h>
#include <sys/thread.h>
#include <sys/modctl.h>
#include <assert.h>

#include "findstack.h"
#include "thread.h"
#include "sobj.h"

/*
 * Parts of this file are shared between targets, but this section is only
 * used for KVM and KMDB.
 */
#ifdef _KERNEL

int findstack_debug_on = 0;

/*
 * "sp" is a kernel VA.
 */
static int
print_stack(uintptr_t sp, uintptr_t pc, uintptr_t addr,
    int argc, const mdb_arg_t *argv, int free_state)
{
        int showargs = 0, count, err;
        char tdesc[128] = "";

        count = mdb_getopts(argc, argv,
```



```
            'v', MDB_OPT_SETBITS, TRUE, &showargs, NULL);
        argc -= count;
        argv += count;

        if (argc > 1 || (argc == 1 && argv->a_type != MDB_TYPE_STRING))
                return (DCMD_USAGE);

        (void) thread_getdesc(addr, B_TRUE, tdesc, sizeof (tdesc));

        mdb_printf("stack pointer for thread %p%s (%s): %p\n",
            addr, (free_state ? " (TS_FREE)" : ""), tdesc, sp);
        mdb_printf("stack pointer for thread %p%s: %p\n",
            addr, (free_state ? " (TS_FREE)" : ""), sp);
        if (pc != 0)
                mdb_printf("[ %0?lr %a() ]\n", sp, pc);

        mdb_inc_indent(2);
        mdb_set_dot(sp);

        if (argc == 1)
                err = mdb_eval(argv->a_un.a_str);
        else if (showargs)
                err = mdb_eval("<.$C");
        else
                err = mdb_eval("<.$C0");

        mdb_dec_indent(2);

        return ((err == -1) ? DCMD_ABORT : DCMD_OK);
}
______unchanged_portion_omitted_

#endif /* _KERNEL */

static void
uppercase(char *p)
{
        for (; *p != '\0'; p++) {
                if (*p >= 'a' && *p <= 'z')
                        *p += 'A' - 'a';
        }
}
______unchanged_portion_omitted_

/* global state cached between invocations */
#define STACKS_STATE_CLEAN      0
#define STACKS_STATE_DIRTY      1
#define STACKS_STATE_DONE       2
static uint_t stacks_state = STACKS_STATE_CLEAN;
static stacks_entry_t **stacks_hash;
static stacks_entry_t **stacks_array;
static size_t stacks_array_size;

static size_t
size_t
stacks_hash_entry(stacks_entry_t *sep)
{
        size_t depth = sep->se_depth;
        uintptr_t *stack = sep->se_stack;

        uint64_t total = depth;

        while (depth > 0) {
                total += *stack;
                stack++; depth--;
        }
```

```
225         return (total % STACKS_HSIZE);
226 }

228 /*
229  * This is used to both compare stacks for equality and to sort the final
230  * list of unique stacks.  forsort specifies the latter behavior, which
231  * additionally:
232  *      compares se_count, and
233  *      sorts the stacks by text function name.
234  *
235  * The equality test is independent of se_count, and doesn't care about
236  * relative ordering, so we don't do the extra work of looking up symbols
237  * for the stack addresses.
238  */
239 static int
227 int
240 stacks_entry_comp_impl(stacks_entry_t *l, stacks_entry_t *r,
241     uint_t forsort)
242 {
243         int idx;

245         int depth = MIN(l->se_depth, r->se_depth);

247         /* no matter what, panic stacks come last. */
248         if (l->se_panic > r->se_panic)
249                 return (1);
250         if (l->se_panic < r->se_panic)
251                 return (-1);

253         if (forsort) {
254                 /* put large counts earlier */
255                 if (l->se_count > r->se_count)
256                         return (-1);
257                 if (l->se_count < r->se_count)
258                         return (1);
259         }

261         if (l->se_tstate > r->se_tstate)
262                 return (1);
263         if (l->se_tstate < r->se_tstate)
264                 return (-1);

266         if (l->se_failed > r->se_failed)
267                 return (1);
268         if (l->se_failed < r->se_failed)
269                 return (-1);

271         for (idx = 0; idx < depth; idx++) {
272                 char lbuf[MDB_SYM_NAMLEN];
273                 char rbuf[MDB_SYM_NAMLEN];

275                 int rval;
276                 uintptr_t laddr = l->se_stack[idx];
277                 uintptr_t raddr = r->se_stack[idx];

279                 if (laddr == raddr)
280                         continue;

282                 if (forsort &&
283                     mdb_lookup_by_addr(laddr, MDB_SYM_FUZZY,
284                     lbuf, sizeof (lbuf), NULL) != -1 &&
285                     mdb_lookup_by_addr(raddr, MDB_SYM_FUZZY,
286                     rbuf, sizeof (rbuf), NULL) != -1 &&
287                     (rval = strcmp(lbuf, rbuf)) != 0)
288                         return (rval);
```



```
290                 if (laddr > raddr)
291                         return (1);
292                 return (-1);
293         }

295         if (l->se_overflow > r->se_overflow)
296                 return (-1);
297         if (l->se_overflow < r->se_overflow)
298                 return (1);

300         if (l->se_depth > r->se_depth)
301                 return (1);
302         if (l->se_depth < r->se_depth)
303                 return (-1);

305         if (l->se_sobj_ops > r->se_sobj_ops)
306                 return (1);
307         if (l->se_sobj_ops < r->se_sobj_ops)
308                 return (-1);

310         return (0);
311 }

313 static int
301 int
314 stacks_entry_comp(const void *l_arg, const void *r_arg)
315 {
316         stacks_entry_t * const *lp = l_arg;
317         stacks_entry_t * const *rp = r_arg;

319         return (stacks_entry_comp_impl(*lp, *rp, 1));
320 }
______unchanged_portion_omitted_

382 /*ARGSUSED*/
383 static int
371 int
384 stacks_thread_cb(uintptr_t addr, const void *ignored, void *cbarg)
385 {
386         stacks_info_t *sip = cbarg;
387         findstack_info_t *fsip = &sip->si_fsi;

389         stacks_entry_t **sepp, *nsep, *sep;
390         int idx;
391         size_t depth;

393         if (stacks_findstack(addr, fsip, 0) != DCMD_OK &&
394             fsip->fsi_failed == FSI_FAIL_BADTHREAD) {
395                 mdb_warn("couldn't read thread at %p\n", addr);
396                 return (WALK_NEXT);
397         }

399         sip->si_count++;

401         depth = fsip->fsi_depth;
402         nsep = mdb_zalloc(STACKS_ENTRY_SIZE(depth), UM_SLEEP);
403         nsep->se_thread = addr;
404         nsep->se_sp = fsip->fsi_sp;
405         nsep->se_sobj_ops = fsip->fsi_sobj_ops;
406         nsep->se_tstate = fsip->fsi_tstate;
407         nsep->se_count = 1;
408         nsep->se_overflow = fsip->fsi_overflow;
409         nsep->se_depth = depth;
410         nsep->se_failed = fsip->fsi_failed;
411         nsep->se_panic = fsip->fsi_panic;
```

```
413         for (idx = 0; idx < depth; idx++)
414                 nsep->se_stack[idx] = fsip->fsi_stack[idx];

416         for (sepp = &sip->si_hash[stacks_hash_entry(nsep)];
417             (sep = *sepp) != NULL;
418             sepp = &sep->se_next) {

420                 if (stacks_entry_comp_impl(sep, nsep, 0) != 0)
421                         continue;

423                 nsep->se_dup = sep->se_dup;
424                 sep->se_dup = nsep;
425                 sep->se_count++;
426                 return (WALK_NEXT);
427         }

429         nsep->se_next = NULL;
430         *sepp = nsep;
431         sip->si_entries++;

433         return (WALK_NEXT);
434 }

436 static int
424 int
437 stacks_run_tlist(mdb_pipe_t *tlist, stacks_info_t *si)
438 {
439         size_t idx;
440         size_t found = 0;
441         int ret;

443         for (idx = 0; idx < tlist->pipe_len; idx++) {
444                 uintptr_t addr = tlist->pipe_data[idx];

446                 found++;

448                 ret = stacks_thread_cb(addr, NULL, si);
449                 if (ret == WALK_DONE)
450                         break;
451                 if (ret != WALK_NEXT)
452                         return (-1);
453         }

455         if (found)
456                 return (0);
457         return (-1);
458 }

460 static int
448 int
461 stacks_run(int verbose, mdb_pipe_t *tlist)
462 {
463         stacks_info_t si;
464         findstack_info_t *fsip = &si.si_fsi;
465         size_t idx;
466         stacks_entry_t **cur;

468         bzero(&si, sizeof (si));

470         stacks_state = STACKS_STATE_DIRTY;

472         stacks_hash = si.si_hash =
473             mdb_zalloc(STACKS_HSIZE * sizeof (*si.si_hash), UM_SLEEP);
474         si.si_entries = 0;
475         si.si_count = 0;
```



```
477         fsip->fsi_max_depth = STACKS_MAX_DEPTH;
478         fsip->fsi_stack =
479             mdb_alloc(fsip->fsi_max_depth * sizeof (*fsip->fsi_stack),
480             UM_SLEEP | UM_GC);

482         if (verbose)
483                 mdb_warn("stacks: processing kernel threads\n");

485         if (tlist != NULL) {
486                 if (stacks_run_tlist(tlist, &si))
487                         return (DCMD_ERR);
488         } else {
489                 if (mdb_walk("thread", stacks_thread_cb, &si) != 0) {
490                         mdb_warn("cannot walk \"thread\"");
491                         return (DCMD_ERR);
492                 }
493         }

495         if (verbose)
496                 mdb_warn("stacks: %d unique stacks / %d threads\n",
497                     si.si_entries, si.si_count);

499         stacks_array_size = si.si_entries;
500         stacks_array =
501             mdb_zalloc(si.si_entries * sizeof (*stacks_array), UM_SLEEP);
502         cur = stacks_array;
503         for (idx = 0; idx < STACKS_HSIZE; idx++) {
504                 stacks_entry_t *sep;
505                 for (sep = si.si_hash[idx]; sep != NULL; sep = sep->se_next)
506                         *(cur++) = sep;
507         }

509         if (cur != stacks_array + si.si_entries) {
510                 mdb_warn("stacks: miscounted array size (%d != size: %d)\n",
511                     (cur - stacks_array), stacks_array_size);
512                 return (DCMD_ERR);
513         }
514         qsort(stacks_array, si.si_entries, sizeof (*stacks_array),
515             stacks_entry_comp);

517         /* Now that we're done, free the hash table */
518         stacks_hash = NULL;
519         mdb_free(si.si_hash, STACKS_HSIZE * sizeof (*si.si_hash));

521         if (tlist == NULL)
522                 stacks_state = STACKS_STATE_DONE;

524         if (verbose)
525                 mdb_warn("stacks: done\n");

527         return (DCMD_OK);
528 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
  134809 Mon Oct 15 13:24:21 2018
new/usr/src/cmd/mdb/common/modules/genunix/genunix.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2018, Joyent, Inc.
 * Copyright 2017 Joyent, Inc.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 */

#include <mdb/mdb_param.h>
#include <mdb/mdb_modapi.h>
#include <mdb/mdb_ks.h>
#include <mdb/mdb_ctf.h>

#include <sys/types.h>
#include <sys/thread.h>
#include <sys/session.h>
#include <sys/user.h>
#include <sys/proc.h>
#include <sys/var.h>
#include <sys/t_lock.h>
#include <sys/callo.h>
#include <sys/priocntl.h>
#include <sys/class.h>
#include <sys/regset.h>
#include <sys/stack.h>
#include <sys/cpuvar.h>
#include <sys/vnode.h>
#include <sys/vfs.h>
#include <sys/flock_impl.h>
#include <sys/kmem_impl.h>
#include <sys/vmem_impl.h>
#include <sys/kstat.h>
#include <sys/dditypes.h>
#include <sys/ddi_impldefs.h>
#include <sys/sysmacros.h>
#include <sys/sysconf.h>
#include <sys/task.h>
#include <sys/project.h>
#include <sys/errorq_impl.h>
#include <sys/cred_impl.h>
```



```
#include <sys/zone.h>
#include <sys/panic.h>
#include <regex.h>
#include <sys/port_impl.h>

#include "avl.h"
#include "bio.h"
#include "bitset.h"
#include "combined.h"
#include "contract.h"
#include "cpupart_mdb.h"
#include "cred.h"
#include "ctxop.h"
#include "cyclic.h"
#include "damap.h"
#include "ddi_periodic.h"
#include "devinfo.h"
#include "dnlc.h"
#include "findstack.h"
#include "fm.h"
#include "gcore.h"
#include "group.h"
#include "irm.h"
#include "kgrep.h"
#include "kmem.h"
#include "ldi.h"
#include "leaky.h"
#include "lgrp.h"
#include "list.h"
#include "log.h"
#include "mdi.h"
#include "memory.h"
#include "mmd.h"
#include "modhash.h"
#include "ndievents.h"
#include "net.h"
#include "netstack.h"
#include "nvpair.h"
#include "pg.h"
#include "rctl.h"
#include "sobj.h"
#include "streams.h"
#include "sysevent.h"
#include "taskq.h"
#include "thread.h"
#include "tsd.h"
#include "tsol.h"
#include "typegraph.h"
#include "vfs.h"
#include "zone.h"
#include "hotplug.h"

/*
 * Surely this is defined somewhere...
 */
#define NINTR           16

#define KILOS           10
#define MEGS            20
#define GIGS            30

#ifndef STACK_BIAS
#define STACK_BIAS      0
#endif

static char
```

```
126 pstat2ch(uchar_t state)
127 {
128         switch (state) {
129                 case SSLEEP: return ('S');
130                 case SRUN: return ('R');
131                 case SZOMB: return ('Z');
132                 case SIDL: return ('I');
133                 case SONPROC: return ('O');
134                 case SSTOP: return ('T');
135                 case SWAIT: return ('W');
136                 default: return ('?');
137         }
138 }

140 #define PS_PRTTHREADS   0x1
141 #define PS_PRTLWPS      0x2
142 #define PS_PSARGS       0x4
143 #define PS_TASKS        0x8
144 #define PS_PROJECTS     0x10
145 #define PS_ZONES        0x20

147 static int
148 ps_threadprint(uintptr_t addr, const void *data, void *private)
149 {
150         const kthread_t *t = (const kthread_t *)data;
151         uint_t prt_flags = *((uint_t *)private);

153         static const mdb_bitmask_t t_state_bits[] = {
154                 { "TS_FREE",    UINT_MAX,       TS_FREE         },
155                 { "TS_SLEEP",   TS_SLEEP,       TS_SLEEP        },
156                 { "TS_RUN",     TS_RUN,         TS_RUN          },
157                 { "TS_ONPROC",  TS_ONPROC,      TS_ONPROC       },
158                 { "TS_ZOMB",    TS_ZOMB,        TS_ZOMB         },
159                 { "TS_STOPPED", TS_STOPPED,     TS_STOPPED      },
160                 { "TS_WAIT",    TS_WAIT,        TS_WAIT         },
161                 { NULL,         0,              0               }
162         };

164         if (prt_flags & PS_PRTTHREADS)
165                 mdb_printf("\tT  %?a <%b>\n", addr, t->t_state, t_state_bits);

167         if (prt_flags & PS_PRTLWPS) {
168                 char desc[128] = "";
167         if (prt_flags & PS_PRTLWPS)
168                 mdb_printf("\tL  %?a ID: %u\n", t->t_lwp, t->t_tid);

170                 (void) thread_getdesc(addr, B_FALSE, desc, sizeof (desc));

172                 mdb_printf("\tL  %?a ID: %s\n", t->t_lwp, desc);
173         }

175         return (WALK_NEXT);
176 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   25785 Mon Oct 15 13:24:25 2018
new/usr/src/cmd/mdb/common/modules/genunix/thread.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 * Copyright (c) 2018, Joyent, Inc.
 */


#include <mdb/mdb_modapi.h>
#include <mdb/mdb_ks.h>
#include <mdb/mdb_ctf.h>
#include <sys/types.h>
#include <sys/thread.h>
#include <sys/lwp.h>
#include <sys/proc.h>
#include <sys/cpuvar.h>
#include <sys/cpupart.h>
#include <sys/disp.h>
#include <sys/taskq_impl.h>
#include <sys/stack.h>
#include "thread.h"

#ifndef STACK_BIAS
#define STACK_BIAS      0
#endif

typedef struct thread_walk {
        kthread_t *tw_thread;
        uintptr_t tw_last;
        uint_t tw_inproc;
        uint_t tw_step;
} thread_walk_t;
_____unchanged_portion_omitted_

/*
 * Return a string description of the thread, including the ID and the thread
 * name.
 *
```



```
 * If ->t_name is NULL, and we're a system thread, we'll do a little more
 * spelunking to find a useful string to return.
 */
int
thread_getdesc(uintptr_t addr, boolean_t include_comm,
    char *buf, size_t bufsize)
{
        char name[THREAD_NAME_MAX] = "";
        kthread_t t;
        proc_t p;

        bzero(buf, bufsize);

        if (mdb_vread(&t, sizeof (kthread_t), addr) == -1) {
                mdb_warn("failed to read kthread_t at %p", addr);
                return (-1);
        }

        if (t.t_tid == 0) {
                taskq_t tq;

                if (mdb_vread(&tq, sizeof (taskq_t),
                    (uintptr_t)t.t_taskq) == -1)
                        tq.tq_name[0] = '\0';

                if (t.t_name != NULL) {
                        if (mdb_readstr(buf, bufsize,
                            (uintptr_t)t.t_name) == -1) {
                                mdb_warn("error reading thread name");
                        }
                } else if (tq.tq_name[0] != '\0') {
                        (void) mdb_snprintf(buf, bufsize, "tq:%s", tq.tq_name);
                } else {
                        mdb_snprintf(buf, bufsize, "%a()", t.t_startpc);
                }

                return (buf[0] == '\0' ? -1 : 0);
        }

        if (include_comm && mdb_vread(&p, sizeof (proc_t),
            (uintptr_t)t.t_procp) == -1) {
                mdb_warn("failed to read proc at %p", t.t_procp);
                return (-1);
        }

        if (t.t_name != NULL) {
                if (mdb_readstr(name, sizeof (name), (uintptr_t)t.t_name) == -1)
                        mdb_warn("error reading thread name");

                /*
                 * Just to be safe -- if mdb_readstr() succeeds, it always NUL
                 * terminates the output, but is unclear what it does on
                 * failure.  In that case we attempt to show any partial content
                 * w/ the warning in case it's useful, but explicitly
                 * NUL-terminate to be safe.
                 */
                buf[bufsize - 1] = '\0';
        }

        if (name[0] != '\0') {
                if (include_comm) {
                        (void) mdb_snprintf(buf, bufsize, "%s/%u [%s]",
                            p.p_user.u_comm, t.t_tid, name);
                } else {
                        (void) mdb_snprintf(buf, bufsize, "%u [%s]",
                            t.t_tid, name);
```

```
636                 }
637         } else {
638                 if (include_comm) {
639                         (void) mdb_snprintf(buf, bufsize, "%s/%u",
640                             p.p_user.u_comm, t.t_tid);
641                 } else {
642                         (void) mdb_snprintf(buf, bufsize, "%u", t.t_tid);
643                 }
644         }

646         return (buf[0] == '\0' ? -1 : 0);
647 }

649 /*
650  * List a combination of kthread_t and proc_t. Add stack traces in verbose mode.
651  */
652 int
653 threadlist(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
654 {
655         int i;
656         uint_t count =  0;
657         uint_t verbose = FALSE;
658         uint_t notaskq = FALSE;
659         kthread_t t;
574         taskq_t tq;
575         proc_t p;
660         char cmd[80];
661         mdb_arg_t cmdarg;

663         if (!(flags & DCMD_ADDRSPEC)) {
664                 if (mdb_walk_dcmd("thread", "threadlist", argc, argv) == -1) {
665                         mdb_warn("can't walk threads");
666                         return (DCMD_ERR);
667                 }
668                 return (DCMD_OK);
669         }

671         i = mdb_getopts(argc, argv,
672             't', MDB_OPT_SETBITS, TRUE, &notaskq,
673             'v', MDB_OPT_SETBITS, TRUE, &verbose, NULL);

675         if (i != argc) {
676                 if (i != argc - 1 || !verbose)
677                         return (DCMD_USAGE);

679                 if (argv[i].a_type == MDB_TYPE_IMMEDIATE)
680                         count = (uint_t)argv[i].a_un.a_val;
681                 else
682                         count = (uint_t)mdb_strtoull(argv[i].a_un.a_str);
683         }

685         if (DCMD_HDRSPEC(flags)) {
686                 if (verbose)
687                         mdb_printf("%<u>%?s %?s %?s %3s %3s %?s%</u>\n",
688                             "ADDR", "PROC", "LWP", "CLS", "PRI", "WCHAN");
689                 else
690                         mdb_printf("%<u>%?s %?s %?s %s/%s%</u>\n",
691                             "ADDR", "PROC", "LWP", "CMD", "LWPID");
692         }

694         if (mdb_vread(&t, sizeof (kthread_t), addr) == -1) {
695                 mdb_warn("failed to read kthread_t at %p", addr);
696                 return (DCMD_ERR);
697         }

699         if (notaskq && t.t_taskq != NULL)
```



```
700                 return (DCMD_OK);

702         if (t.t_state == TS_FREE)
703                 return (DCMD_OK);

705         if (!verbose) {
706                 char desc[128];

708                 if (thread_getdesc(addr, B_TRUE, desc, sizeof (desc)) == -1)
621         if (mdb_vread(&p, sizeof (proc_t), (uintptr_t)t.t_procp) == -1) {
622                 mdb_warn("failed to read proc at %p", t.t_procp);
709                         return (DCMD_ERR);

711                 mdb_printf("%0?p %?p %?p %s\n", addr, t.t_procp, t.t_lwp, desc);
712                 return (DCMD_OK);
713         }

626         if (mdb_vread(&tq, sizeof (taskq_t), (uintptr_t)t.t_taskq) == -1)
627                 tq.tq_name[0] = '\0';

629         if (verbose) {
715         mdb_printf("%0?p %?p %?p %3u %3d %?p\n",
716             addr, t.t_procp, t.t_lwp, t.t_cid, t.t_pri, t.t_wchan);

718         mdb_inc_indent(2);

720         mdb_printf("PC: %a\n", t.t_pc);
635                 mdb_printf("PC: %a", t.t_pc);
636                 if (t.t_tid == 0) {
637                         if (tq.tq_name[0] != '\0')
638                                 mdb_printf("    TASKQ: %s\n", tq.tq_name);
639                         else
640                                 mdb_printf("    THREAD: %a()\n", t.t_startpc);
641                 } else {
642                         mdb_printf("    CMD: %s\n", p.p_user.u_psargs);
643                 }

722         mdb_snprintf(cmd, sizeof (cmd), "<.$c%d", count);
723         cmdarg.a_type = MDB_TYPE_STRING;
724         cmdarg.a_un.a_str = cmd;

726         (void) mdb_call_dcmd("findstack", addr, flags, 1, &cmdarg);

728         mdb_dec_indent(2);

730         mdb_printf("\n");
654         } else {
655                 mdb_printf("%0?p %?p %?p", addr, t.t_procp, t.t_lwp);
656                 if (t.t_tid == 0) {
657                         if (tq.tq_name[0] != '\0')
658                                 mdb_printf(" tq:%s\n", tq.tq_name);
659                         else
660                                 mdb_printf(" %a()\n", t.t_startpc);
661                 } else {
662                         mdb_printf(" %s/%u\n", p.p_user.u_comm, t.t_tid);
663                 }
664         }

732         return (DCMD_OK);
733 }
```

*_____unchanged_portion_omitted_*

```
778 /*
779  * Display kthread stack infos.
780  */
781 int
```

```
782 stackinfo(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
783 {
784         kthread_t t;
719         proc_t p;
785         uint64_t *ptr;  /* pattern pointer */
786         caddr_t start;  /* kernel stack start */
787         caddr_t end;    /* kernel stack end */
788         caddr_t ustack; /* userland copy of kernel stack */
789         size_t usize;   /* userland copy of kernel stack size */
790         caddr_t ustart; /* userland copy of kernel stack, aligned start */
791         caddr_t uend;   /* userland copy of kernel stack, aligned end */
792         size_t percent = 0;
793         uint_t all = FALSE; /* don't show TS_FREE kthread by default */
794         uint_t history = FALSE;
795         int i = 0;
796         unsigned int ukmem_stackinfo;
797         uintptr_t allthreads;
798         char tdesc[128] = "";

800         /* handle options */
801         if (mdb_getopts(argc, argv,
802             'a', MDB_OPT_SETBITS, TRUE, &all,
803             'h', MDB_OPT_SETBITS, TRUE, &history, NULL) != argc) {
804                 return (DCMD_USAGE);
805         }

807         /* walk all kthread if needed */
808         if ((history == FALSE) && !(flags & DCMD_ADDRSPEC)) {
809                 if (mdb_walk_dcmd("thread", "stackinfo", argc, argv) == -1) {
810                         mdb_warn("can't walk threads");
811                         return (DCMD_ERR);
812                 }
813                 return (DCMD_OK);
814         }

816         /* read 'kmem_stackinfo' */
817         if (mdb_readsym(&ukmem_stackinfo, sizeof (ukmem_stackinfo),
818             "kmem_stackinfo") == -1) {
819                 mdb_warn("failed to read 'kmem_stackinfo'\n");
820                 ukmem_stackinfo = 0;
821         }

823         /* read 'allthreads' */
824         if (mdb_readsym(&allthreads, sizeof (kthread_t *),
825             "allthreads") == -1) {
826                 mdb_warn("failed to read 'allthreads'\n");
827                 allthreads = NULL;
828         }

830         if (history == TRUE) {
831                 kmem_stkinfo_t *log;
832                 uintptr_t kaddr;

834                 mdb_printf("Dead kthreads stack usage history:\n");
835                 if (ukmem_stackinfo == 0) {
836                         mdb_printf("Tunable kmem_stackinfo is unset, history ");
837                         mdb_printf("feature is off.\nUse ::help stackinfo ");
838                         mdb_printf("for more details.\n");
839                         return (DCMD_OK);
840                 }

842                 mdb_printf("%<u>%?s%</u>", "THREAD");
843                 mdb_printf(" %<u>%?s%</u>", "STACK");
844                 mdb_printf("%<u>%s%</u>", "   SIZE  MAX LWP");
778                 mdb_printf("%<u>%s%</u>", "   SIZE  MAX CMD/LWPID or STARTPC");
845                 mdb_printf("\n");
```



```
846                 usize = KMEM_STKINFO_LOG_SIZE * sizeof (kmem_stkinfo_t);
847                 log = (kmem_stkinfo_t *)mdb_alloc(usize, UM_SLEEP);
848                 if (mdb_readsym(&kaddr, sizeof (kaddr),
849                     "kmem_stkinfo_log") == -1) {
850                         mdb_free((void *)log, usize);
851                         mdb_warn("failed to read 'kmem_stkinfo_log'\n");
852                         return (DCMD_ERR);
853                 }
854                 if (kaddr == NULL) {
855                         mdb_free((void *)log, usize);
856                         return (DCMD_OK);
857                 }
858                 if (mdb_vread(log, usize, kaddr) == -1) {
859                         mdb_free((void *)log, usize);
860                         mdb_warn("failed to read %p\n", kaddr);
861                         return (DCMD_ERR);
862                 }
863                 for (i = 0; i < KMEM_STKINFO_LOG_SIZE; i++) {
864                         if (log[i].kthread == NULL) {
865                                 continue;
866                         }

868                         (void) thread_getdesc((uintptr_t)log[i].kthread,
869                             B_TRUE, tdesc, sizeof (tdesc));

871                         mdb_printf("%0?p %0?p %6x %3d%% %s\n",
801                         mdb_printf("%0?p %0?p %6x %3d%%",
872                             log[i].kthread,
873                             log[i].start,
874                             (uint_t)log[i].stksz,
875                             (int)log[i].percent, tdesc);
805                             (int)log[i].percent);
806                         if (log[i].t_tid != 0) {
807                                 mdb_printf(" %s/%u\n",
808                                     log[i].cmd, log[i].t_tid);
809                         } else {
810                                 mdb_printf(" %p (%a)\n", log[i].t_startpc,
811                                     log[i].t_startpc);
876                 }
813                 }
877                 mdb_free((void *)log, usize);
878                 return (DCMD_OK);
879         }

881         /* display header */
882         if (DCMD_HDRSPEC(flags)) {
883                 if (ukmem_stackinfo == 0) {
884                         mdb_printf("Tunable kmem_stackinfo is unset, ");
885                         mdb_printf("MAX value is not available.\n");
886                         mdb_printf("Use ::help stackinfo for more details.\n");
887                 }
888                 mdb_printf("%<u>%?s%</u>", "THREAD");
889                 mdb_printf(" %<u>%?s%</u>", "STACK");
890                 mdb_printf("%<u>%s%</u>", "   SIZE  CUR  MAX LWP");
827                 mdb_printf("%<u>%s%</u>", "   SIZE  CUR  MAX CMD/LWPID");
891                 mdb_printf("\n");
892         }

894         /* read kthread */
895         if (mdb_vread(&t, sizeof (kthread_t), addr) == -1) {
896                 mdb_warn("can't read kthread_t at %#lx\n", addr);
897                 return (DCMD_ERR);
898         }

900         if (t.t_state == TS_FREE && all == FALSE) {
901                 return (DCMD_OK);
```

```
902         }

841         /* read proc */
842         if (mdb_vread(&p, sizeof (proc_t), (uintptr_t)t.t_procp) == -1) {
843                 mdb_warn("failed to read proc at %p\n", t.t_procp);
844                 return (DCMD_ERR);
845         }

904         /*
905          * Stack grows up or down, see thread_create(),
906          * compute stack memory aera start and end (start < end).
907          */
908         if (t.t_stk > t.t_stkbase) {
909                 /* stack grows down */
910                 start = t.t_stkbase;
911                 end = t.t_stk;
912         } else {
913                 /* stack grows up */
914                 start = t.t_stk;
915                 end = t.t_stkbase;
916         }

918         /* display stack info */
919         mdb_printf("%0?p %0?p", addr, start);

921         /* (end - start), kernel stack size as found in kthread_t */
922         if ((end <= start) || ((end - start) > (1024 * 1024))) {
923                 /* negative or stack size > 1 meg, assume bogus */
924                 mdb_warn(" t_stk/t_stkbase problem\n");
925                 return (DCMD_ERR);
926         }

928         /* display stack size */
929         mdb_printf(" %6x", end - start);

931         /* display current stack usage */
932         percent = stk_compute_percent(t.t_stk, t.t_stkbase,
933             (caddr_t)t.t_sp + STACK_BIAS);

935         mdb_printf(" %3d%%", percent);
936         percent = 0;

938         (void) thread_getdesc(addr, B_TRUE, tdesc, sizeof (tdesc));

940         if (ukmem_stackinfo == 0) {
941                 mdb_printf("  n/a %s\n", tdesc);
882                 mdb_printf("  n/a");
883                 if (t.t_tid == 0) {
884                         mdb_printf(" %a()", t.t_startpc);
885                 } else {
886                         mdb_printf(" %s/%u", p.p_user.u_comm, t.t_tid);
887                 }
888                 mdb_printf("\n");
942                 return (DCMD_OK);
943         }

945         if ((((uintptr_t)start) & 0x7) != 0) {
946                 start = (caddr_t)((((uintptr_t)start) & (~0x7)) + 8);
947         }
948         end = (caddr_t)(((uintptr_t)end) & (~0x7));
949         /* size to scan in userland copy of kernel stack */
950         usize = end - start; /* is a multiple of 8 bytes */

952         /*
953          * Stackinfo pattern size is 8 bytes. Ensure proper 8 bytes
954          * alignement for ustart and uend, in boundaries.
```



```
955          */
956         ustart = ustack = (caddr_t)mdb_alloc(usize + 8, UM_SLEEP);
957         if ((((uintptr_t)ustart) & 0x7) != 0) {
958                 ustart = (caddr_t)((((uintptr_t)ustart) & (~0x7)) + 8);
959         }
960         uend = ustart + usize;

962         /* read the kernel stack */
963         if (mdb_vread(ustart, usize, (uintptr_t)start) != usize) {
964                 mdb_free((void *)ustack, usize + 8);
965                 mdb_printf("\n");
966                 mdb_warn("couldn't read entire stack\n");
967                 return (DCMD_ERR);
968         }

970         /* scan the stack */
971         if (t.t_stk > t.t_stkbase) {
972                 /* stack grows down */
973 #if defined(__i386) || defined(__amd64)
974                 /*
975                  * 6 longs are pushed on stack, see thread_load(). Skip
976                  * them, so if kthread has never run, percent is zero.
977                  * 8 bytes alignement is preserved for a 32 bit kernel,
978                  * 6 x 4 = 24, 24 is a multiple of 8.
979                  */
980                 uend -= (6 * sizeof (long));
981 #endif
982                 ptr = (uint64_t *)((void *)ustart);
983                 while (ptr < (uint64_t *)((void *)uend)) {
984                         if (*ptr != KMEM_STKINFO_PATTERN) {
985                                 percent = stk_compute_percent(uend,
986                                     ustart, (caddr_t)ptr);
987                                 break;
988                         }
989                         ptr++;
990                 }
991         } else {
992                 /* stack grows up */
993                 ptr = (uint64_t *)((void *)uend);
994                 ptr--;
995                 while (ptr >= (uint64_t *)((void *)ustart)) {
996                         if (*ptr != KMEM_STKINFO_PATTERN) {
997                                 percent = stk_compute_percent(ustart,
998                                     uend, (caddr_t)ptr);
999                                 break;
1000                        }
1001                        ptr--;
1002                }
1003        }

1005        /* thread 't0' stack is not created by thread_create() */
1006        if (addr == allthreads) {
1007                percent = 0;
1008        }
1009        if (percent != 0) {
1010                mdb_printf(" %3d%%", percent);
1011        } else {
1012                mdb_printf("  n/a");
1013        }

1015        mdb_printf(" %s\n", tdesc);

961         if (t.t_tid == 0) {
962                 mdb_printf(" %a()", t.t_startpc);
963         } else {
964                 mdb_printf(" %s/%u", p.p_user.u_comm, t.t_tid);
```

```
 965         }
 966         mdb_printf("\n");
1017         mdb_free((void *)ustack, usize + 8);
1018         return (DCMD_OK);
1019 }

1021 void
1022 stackinfo_help(void)
1023 {
1024         mdb_printf(
1025             "Shows kernel stacks real utilization, if /etc/system "
1026             "kmem_stackinfo tunable\n");
1027         mdb_printf(
1028             "(an unsigned integer) is non zero at kthread creation time. ");
1029         mdb_printf("For example:\n");
1030         mdb_printf(
1031             "          THREAD            STACK   SIZE  CUR  MAX LWP\n");
 981             "          THREAD            STACK   SIZE  CUR  MAX CMD/LWPID\n");
1032         mdb_printf(
1033             "ffffff014f5f2c20 ffffff0004153000   4f00   4%%  43%% init/1\n");
1034         mdb_printf(
1035             "The stack size utilization for this kthread is at 4%%"
1036             " of its maximum size,\n");
1037         mdb_printf(
1038             "but has already used up to 43%%, stack size is 4f00 bytes.\n");
1039         mdb_printf(
1040             "MAX value can be shown as n/a (not available):\n");
1041         mdb_printf(
1042             "  - for the very first kthread (sched/1)\n");
1043         mdb_printf(
1044             "  - kmem_stackinfo was zero at kthread creation time\n");
1045         mdb_printf(
1046             "  - kthread has not yet run\n");
1047         mdb_printf("\n");
1048         mdb_printf("Options:\n");
1049         mdb_printf(
1050             "-a shows also TS_FREE kthreads (interrupt kthreads)\n");
1051         mdb_printf(
1052             "-h shows history, dead kthreads that used their "
1053             "kernel stack the most\n");
1054         mdb_printf(
1055             "\nSee illumos Modular Debugger Guide for detailed usage.\n");
1005             "\nSee Solaris Modular Debugger Guide for detailed usage.\n");
1056         mdb_flush();
1057 }
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    2105 Mon Oct 15 13:24:27 2018
new/usr/src/cmd/mdb/common/modules/genunix/thread.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright (c) 2018, Joyent, Inc.
 */

#ifndef _THREAD_H
#define _THREAD_H


#include <mdb/mdb_modapi.h>

#ifdef  __cplusplus
extern "C" {
#endif

int thread_walk_init(mdb_walk_state_t *);
int thread_walk_step(mdb_walk_state_t *);
void thread_walk_fini(mdb_walk_state_t *);

int deathrow_walk_init(mdb_walk_state_t *);
int deathrow_walk_step(mdb_walk_state_t *);
int thread_deathrow_walk_init(mdb_walk_state_t *);
int lwp_deathrow_walk_init(mdb_walk_state_t *);

int cpu_dispq_walk_init(mdb_walk_state_t *);
int cpupart_dispq_walk_init(mdb_walk_state_t *);
int dispq_walk_step(mdb_walk_state_t *);
void dispq_walk_fini(mdb_walk_state_t *);

int thread(uintptr_t, uint_t, int, const mdb_arg_t *);
void thread_help(void);
int threadlist(uintptr_t, uint_t, int, const mdb_arg_t *);
void threadlist_help(void);
int stackinfo(uintptr_t, uint_t, int, const mdb_arg_t *);
void stackinfo_help(void);

void thread_state_to_text(uint_t, char *, size_t);
int thread_text_to_state(const char *, uint_t *);
```



```
void thread_walk_states(void (*)(uint_t, const char *, void *), void *);

int thread_getdesc(uintptr_t, boolean_t, char *, size_t);

#ifdef  __cplusplus
}
_____unchanged_portion_omitted_
```

```
**********************************************************
   60112 Mon Oct 15 13:24:29 2018
new/usr/src/cmd/nscd/cache.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2012 Milan Jurik. All rights reserved.
 * Copyright (c) 2016 by Delphix. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

/*
 * Cache routines for nscd
 */
#include <assert.h>
#include <errno.h>
#include <memory.h>
#include <signal.h>
#include <stdlib.h>
#include <stddef.h>
#include <stdio.h>
#include <string.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <ucred.h>
#include <nss_common.h>
#include <locale.h>
#include <ctype.h>
#include <strings.h>
#include <string.h>
#include <umem.h>
#include <fcntl.h>
#include "cache.h"
#include "nscd_door.h"
#include "nscd_log.h"
#include "nscd_config.h"
#include "nscd_frontend.h"
#include "nscd_switch.h"

#define SUCCESS         0
#define NOTFOUND        -1
```



```
#define SERVERERROR     -2
#define NOSERVER        -3
#define CONTINUE        -4

static nsc_db_t *nsc_get_db(nsc_ctx_t *, int);
static nscd_rc_t lookup_cache(nsc_lookup_args_t *, nscd_cfg_cache_t *,
                nss_XbyY_args_t *, char *, nsc_entry_t **);
static uint_t reap_cache(nsc_ctx_t *, uint_t, uint_t);
static void delete_entry(nsc_db_t *, nsc_ctx_t *, nsc_entry_t *);
static void print_stats(nscd_cfg_stat_cache_t *);
static void print_cfg(nscd_cfg_cache_t *);
static int lookup_int(nsc_lookup_args_t *, int);

#ifdef  NSCD_DEBUG
static void print_entry(nsc_db_t *, time_t, nsc_entry_t *);
static void avl_dump(nsc_db_t *, time_t);
static void hash_dump(nsc_db_t *, time_t);
#endif  /* NSCD_DEBUG */
static nsc_entry_t *hash_find(nsc_db_t *, nsc_entry_t *, uint_t *, nscd_bool_t);

static void queue_adjust(nsc_db_t *, nsc_entry_t *);
static void queue_remove(nsc_db_t *, nsc_entry_t *);
#ifdef  NSCD_DEBUG
static void queue_dump(nsc_db_t *, time_t);
#endif  /* NSCD_DEBUG */

static int launch_update(nsc_lookup_args_t *);
static void do_update(nsc_lookup_args_t *);
static void getxy_keepalive(nsc_ctx_t *, nsc_db_t *, int, int);

static void ctx_info(nsc_ctx_t *);
static void ctx_info_nolock(nsc_ctx_t *);
static void ctx_invalidate(nsc_ctx_t *);

static void nsc_db_str_key_getlogstr(char *, char *, size_t, nss_XbyY_args_t *);
static void nsc_db_int_key_getlogstr(char *, char *, size_t, nss_XbyY_args_t *);
static void nsc_db_any_key_getlogstr(char *, char *, size_t, nss_XbyY_args_t *);

static int nsc_db_cis_key_compar(const void *, const void *);
static int nsc_db_ces_key_compar(const void *, const void *);
static int nsc_db_int_key_compar(const void *, const void *);

static uint_t nsc_db_cis_key_gethash(nss_XbyY_key_t *, int);
static uint_t nsc_db_ces_key_gethash(nss_XbyY_key_t *, int);
static uint_t nsc_db_int_key_gethash(nss_XbyY_key_t *, int);

static umem_cache_t     *nsc_entry_cache;

static nsc_ctx_t *init_cache_ctx(int);
static void reaper(nsc_ctx_t *);
static void revalidate(nsc_ctx_t *);

static nss_status_t
dup_packed_buffer(void *src, void *dst) {
        nsc_lookup_args_t       *s = (nsc_lookup_args_t *)src;
        nsc_entry_t             *d = (nsc_entry_t *)dst;
        nss_pheader_t           *sphdr = (nss_pheader_t *)s->buffer;
        nss_pheader_t           *dphdr = (nss_pheader_t *)d->buffer;
        int                     slen, new_pbufsiz = 0;

        if (NSCD_GET_STATUS(sphdr) != NSS_SUCCESS) {

                /* no result, copy header only (status, errno, etc) */
                slen = sphdr->data_off;
        } else {
                /*
```

```
 127                  * lookup result returned, data to copy is the packed
 128                  * header plus result (add 1 for the terminating NULL
 129                  * just in case)
 130                  */
 131                 slen = sphdr->data_off + sphdr->data_len + 1;
 132         }

 134         /* allocate cache packed buffer */
 135         if (dphdr != NULL && d->bufsize <= slen && d->bufsize != 0) {
 136                 /* old buffer too small, free it */
 137                 free(dphdr);
 138                 d->buffer = NULL;
 139                 d->bufsize = 0;
 140                 dphdr = NULL;
 141         }
 142         if (dphdr == NULL) {
 143                 /* get new buffer */
 144                 dphdr = calloc(1, slen + 1);
 145                 if (dphdr == NULL)
 146                         return (NSS_ERROR);
 147                 d->buffer = dphdr;
 148                 d->bufsize = slen + 1;
 149                 new_pbufsiz = slen + 1;
 150         }

 152         (void) memcpy(dphdr, sphdr, slen);
 153         if (new_pbufsiz != 0)
 154                 dphdr->pbufsiz = new_pbufsiz;

 156         return (NSS_SUCCESS);
 157 }
_____unchanged_portion_omitted_


1818 static void
1819 revalidate(nsc_ctx_t *ctx)
1820 {
1821         (void) thr_setname(thr_self(), "revalidate");

1823         for (;;) {
1824                 int i, slp, interval, count;

1826                 (void) rw_rdlock(&ctx->cfg_rwlp);
1827                 slp = ctx->cfg.pos_ttl;
1828                 count = ctx->cfg.keephot;
1829                 (void) rw_unlock(&ctx->cfg_rwlp);

1831                 if (slp < 60)
1832                         slp = 60;
1833                 if (count != 0) {
1834                         interval = (slp/2)/count;
1835                         if (interval == 0)
1836                                 interval = 1;
1837                         (void) sleep(slp*2/3);
1838                         for (i = 0; i < ctx->db_count; i++) {
1839                                 getxy_keepalive(ctx, ctx->nsc_db[i],
1840                                     count, interval);
1841                         }
1842                 } else {
1843                         (void) sleep(slp);
1844                 }
1845         }
1846 }
_____unchanged_portion_omitted_
```



```
1963 static void
1964 do_update(nsc_lookup_args_t *in) {
1965         nss_pheader_t   *phdr = (nss_pheader_t *)in->buffer;

1967         (void) thr_setname(thr_self(), "do_update");

1969         /* update the length of the data buffer */
1970         phdr->data_len = phdr->pbufsiz - phdr->data_off;

1972         (void) lookup_int(in, UPDATEBIT);
1973         if (in->buffer)
1974                 free(in->buffer);
1975         free(in);
1976 }
_____unchanged_portion_omitted_

2190 static void
2191 reaper(nsc_ctx_t *ctx)
2192 {
2193         uint_t          ttl, extra_sleep, total_sleep, intervals;
2194         uint_t          nodes_per_interval, seconds_per_interval;
2195         ulong_t         nsc_entries;
2196         char            *me = "reaper";

2198         (void) thr_setname(thr_self(), me);

2200         for (;;) {
2201                 (void) mutex_lock(&ctx->stats_mutex);
2202                 nsc_entries = ctx->stats.entries;
2203                 (void) mutex_unlock(&ctx->stats_mutex);

2205                 (void) rw_rdlock(&ctx->cfg_rwlp);
2206                 ttl = ctx->cfg.pos_ttl;
2207                 (void) rw_unlock(&ctx->cfg_rwlp);

2209                 if (nsc_entries == 0) {
2210                         _NSCD_LOG(NSCD_LOG_CACHE, NSCD_LOG_LEVEL_DEBUG)
2211                                 (me, "%s: nothing to reap\n", ctx->dbname);

2213                         /* sleep for atleast 60 seconds */
2214                         if (ttl < 60)
2215                                 ttl = 60;
2216                         _NSCD_LOG(NSCD_LOG_CACHE, NSCD_LOG_LEVEL_DEBUG)
2217                         (me, "%s: sleep %d\n", ctx->dbname, ttl);
2218                         (void) sleep(ttl);
2219                         continue;
2220                 }

2222                 if (ttl < 32) ttl = 32;
2223                 if (ttl > (1<<28)) ttl = 1<<28;

2225                 /*
2226                  * minimum nodes_per_interval = 256 or 1<<8
2227                  * maximum nodes_per_interval = nsc_entries
2228                  * minimum seconds_per_interval = 32 or 1<<5
2229                  * maximum_seconds_per_interval = ttl
2230                  */
2231                 if (nsc_entries <= ttl) {
2232                         intervals = (nsc_entries >> 8) + 1;
2233                         seconds_per_interval = ttl / intervals;
2234                         nodes_per_interval = 256;
2235                 } else {
2236                         intervals = (ttl >> 5) + 1;
2237                         seconds_per_interval = 32;
2238                         nodes_per_interval = nsc_entries / intervals;
2239                         if (nodes_per_interval < 256)
```

```
2240                                    nodes_per_interval = 256;
2241                }

2243                _NSCD_LOG(NSCD_LOG_CACHE, NSCD_LOG_LEVEL_DEBUG)
2244                        (me, "%s: total entries = %d, "
2245                            "seconds per interval = %d, "
2246                            "nodes per interval = %d\n",
2247                            ctx->dbname, nsc_entries, seconds_per_interval,
2248                            nodes_per_interval);
2249                total_sleep = reap_cache(ctx, nodes_per_interval,
2250                    seconds_per_interval);
2251                extra_sleep = 1 + ttl - total_sleep;
2252                if (extra_sleep > 0)
2253                        (void) sleep(extra_sleep);
2254        }
2255 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   38025 Mon Oct 15 13:24:33 2018
new/usr/src/cmd/nscd/nscd_frontend.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 * Copyright 2012 Milan Jurik. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#include <stdlib.h>
#include <alloca.h>
#include <signal.h>
#include <sys/stat.h>
#include <unistd.h>
#include <pthread.h>
#include <time.h>
#include <errno.h>
#include <door.h>
#include <zone.h>
#include <resolv.h>
#include <sys/socket.h>
#include <net/route.h>
#include <string.h>
#include <net/if.h>
#include <sys/stat.h>
#include <fcntl.h>
#include "nscd_common.h"
#include "nscd_door.h"
#include "nscd_config.h"
#include "nscd_switch.h"
#include "nscd_log.h"
#include "nscd_selfcred.h"
#include "nscd_frontend.h"
#include "nscd_admin.h"

static void rts_mon(void);
static void keep_open_dns_socket(void);

extern nsc_ctx_t *cache_ctx_p[];

/*
 * Current active Configuration data for the frontend component
```



```
 */
static nscd_cfg_global_frontend_t       frontend_cfg_g;
static nscd_cfg_frontend_t              *frontend_cfg;

static int      max_servers = 0;
static int      max_servers_set = 0;
static int      per_user_is_on = 1;

static char     *main_execname;
static char     **main_argv;
extern int      _whoami;
extern long     activity;
extern mutex_t  activity_lock;

static sema_t   common_sema;

static thread_key_t     lookup_state_key;
static mutex_t          create_lock = DEFAULTMUTEX;
static int              num_servers = 0;
static thread_key_t     server_key;

/*
 * Bind a TSD value to a server thread. This enables the destructor to
 * be called if/when this thread exits.  This would be a programming
 * error, but better safe than sorry.
 */
/*ARGSUSED*/
static void *
server_tsd_bind(void *arg)
{
        static void *value = 0;

        (void) thr_setname(thr_self(), "server_tsd_bind");

        /* disable cancellation to avoid hangs if server threads disappear */
        (void) pthread_setcancelstate(PTHREAD_CANCEL_DISABLE, NULL);
        (void) thr_setspecific(server_key, value);
        (void) door_return(NULL, 0, NULL, 0);

        /* make lint happy */
        return (NULL);
}
```

*_____unchanged_portion_omitted_*

```
/*
 * get clearance
 */
int
_nscd_get_clearance(sema_t *sema)
{
_nscd_get_clearance(sema_t *sema) {
        if (sema_trywait(&common_sema) == 0) {
                (void) thr_setspecific(lookup_state_key, NULL);
                return (0);
        }

        if (sema_trywait(sema) == 0) {
                (void) thr_setspecific(lookup_state_key, (void*)1);
                return (0);
        }

        return (1);
}


/*
```

```
 155  * release clearance
 156  */
 157 int
 158 _nscd_release_clearance(sema_t *sema)
 159 {
 154 _nscd_release_clearance(sema_t *sema) {
 160         int     which;

 162         (void) thr_getspecific(lookup_state_key, (void**)&which);
 163         if (which == 0) /* from common pool */ {
 164                 (void) sema_post(&common_sema);
 165                 return (0);
 166         }

 168         (void) sema_post(sema);
 169         return (1);
 170 }
______unchanged_portion_omitted_

1463 /*
1464  * Monitor the routing socket.  Address lists stored in the ipnodes
1465  * cache are sorted based on destination address selection rules,
1466  * so when things change that could affect that sorting (interfaces
1467  * go up or down, flags change, etc.), we clear that cache so the
1468  * list will be re-ordered the next time the hostname is resolved.
1469  */
1470 static void
1471 rts_mon(void)
1472 {
1473         int     rt_sock, rdlen, idx;
1474         union {
1475                 struct {
1476                         struct rt_msghdr rtm;
1477                         struct sockaddr_storage addrs[RTA_NUMBITS];
1478                 } r;
1479                 struct if_msghdr ifm;
1480                 struct ifa_msghdr ifam;
1481         } mbuf;
1482         struct ifa_msghdr *ifam = &mbuf.ifam;
1483         char    *me = "rts_mon";

1485         (void) thr_setname(thr_self(), me);

1487         rt_sock = socket(PF_ROUTE, SOCK_RAW, 0);
1488         if (rt_sock < 0) {
1489                 _NSCD_LOG(NSCD_LOG_FRONT_END, NSCD_LOG_LEVEL_ERROR)
1490                 (me, "Failed to open routing socket: %s\n", strerror(errno));
1491                 thr_exit(0);
1492         }

1494         for (;;) {
1495                 rdlen = read(rt_sock, &mbuf, sizeof (mbuf));
1496                 if (rdlen <= 0) {
1497                         if (rdlen == 0 || (errno != EINTR && errno != EAGAIN)) {
1498                                 _NSCD_LOG(NSCD_LOG_FRONT_END,
1499                                     NSCD_LOG_LEVEL_ERROR)
1500                                 (me, "routing socket read: %s\n",
1501                                     strerror(errno));
1502                                 thr_exit(0);
1503                         }
1504                         continue;
1505                 }
1506                 if (ifam->ifam_version != RTM_VERSION) {
1507                                 _NSCD_LOG(NSCD_LOG_FRONT_END,
1508                                     NSCD_LOG_LEVEL_ERROR)
1509                                 (me, "rx unknown version (%d) on "
```



```
1510                                     "routing socket.\n",
1511                                     ifam->ifam_version);
1512                         continue;
1513                 }
1514                 switch (ifam->ifam_type) {
1515                 case RTM_NEWADDR:
1516                 case RTM_DELADDR:
1517                         /* if no ipnodes cache, then nothing to do */
1518                         idx = get_cache_idx("ipnodes");
1519                         if (cache_ctx_p[idx] == NULL ||
1520                             cache_ctx_p[idx]->reaper_on != nscd_true)
1521                                 break;
1522                         nsc_invalidate(cache_ctx_p[idx], NULL, NULL);
1523                         break;
1524                 case RTM_ADD:
1525                 case RTM_DELETE:
1526                 case RTM_CHANGE:
1527                 case RTM_GET:
1528                 case RTM_LOSING:
1529                 case RTM_REDIRECT:
1530                 case RTM_MISS:
1531                 case RTM_LOCK:
1532                 case RTM_OLDADD:
1533                 case RTM_OLDDEL:
1534                 case RTM_RESOLVE:
1535                 case RTM_IFINFO:
1536                 case RTM_CHGADDR:
1537                 case RTM_FREEADDR:
1538                         break;
1539                 default:
1540                         _NSCD_LOG(NSCD_LOG_FRONT_END, NSCD_LOG_LEVEL_ERROR)
1541                         (me, "rx unknown msg type (%d) on routing socket.\n",
1542                             ifam->ifam_type);
1543                         break;
1544                 }
1545         }
1546 }
______unchanged_portion_omitted_
```

```
**********************************************************
   18655 Mon Oct 15 13:24:35 2018
new/usr/src/cmd/nscd/nscd_getentctx.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 */

#include <sys/ccompile.h>

#include <stdlib.h>
#include <assert.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>

#include "nscd_db.h"
#include "nscd_log.h"
#include "nscd_switch.h"
#include "nscd_door.h"

extern int              _whoami;
static mutex_t          getent_monitor_mutex = DEFAULTMUTEX;
static int              getent_monitor_started = 0;

static rwlock_t         getent_ctxDB_rwlock = DEFAULTRWLOCK;
static nscd_db_t        *getent_ctxDB = NULL;

/*
 * internal structure representing a nscd getent context
 */
typedef struct nscd_getent_ctx {
        int                     to_delete; /* this ctx no longer valid */
        nscd_getent_context_t   *ptr;
        nscd_cookie_num_t       cookie_num;
} nscd_getent_ctx_t;
```

*unchanged_portion_omitted_*

```
/*
 * FUNCTION: reclaim_getent_ctx
 */
```



```
/*ARGSUSED*/
static void * __NORETURN
reclaim_getent_ctx(void *arg)
{
        void                    *cookie = NULL;
        nscd_db_entry_t         *ep;
        nscd_getent_ctx_t       *ctx;
        nscd_getent_context_t   *gctx, *c;
        nscd_getent_context_t   *first = NULL, *last = NULL;
        nss_getent_t            nssctx = { 0 };
        char                    *me = "reclaim_getent_ctx";

        (void) thr_setname(thr_self(), me);

        /*CONSTCOND*/
        while (1) {

                (void) sleep(60);

                (void) rw_rdlock(&getent_ctxDB_rwlock);

                for (ep = _nscd_walk_db(getent_ctxDB, &cookie); ep != NULL;
                    ep = _nscd_walk_db(getent_ctxDB, &cookie)) {

                        ctx = (nscd_getent_ctx_t *)*(ep->data_array);

                        gctx = ctx->ptr;

                        /*
                         * if the client process, which did the setent,
                         * exited, add the context to the orphan list
                         */
                        if (gctx->pid != -1 && process_exited(gctx->pid)) {

                                _NSCD_LOG(NSCD_LOG_GETENT_CTX,
                                    NSCD_LOG_LEVEL_DEBUG)
                                (me, "process  %d exited, "
                                    "getent context = %p, "
                                    "db index = %d, cookie # = %lld, "
                                    "sequence # = %lld\n",
                                    gctx->pid, gctx, gctx->dbi,
                                    gctx->cookie_num, gctx->seq_num);

                                if (first != NULL) {
                                        /* add to list if not in already */
                                        for (c = first; c != NULL;
                                            c = c->next_to_reclaim) {
                                                if (gctx == c)
                                                        break;
                                        }
                                        if (c == NULL) {
                                                last->next_to_reclaim = gctx;
                                                last = gctx;
                                        }
                                } else {
                                        first = gctx;
                                        last = gctx;
                                }
                        }
                }

                (void) rw_unlock(&getent_ctxDB_rwlock);


                /*
                 * return all the orphan getent contexts to the pool if not
```

```
717                  * in use
718                  */
719                 for (gctx = first; gctx; ) {
720                         int in_use, num_reclaim_check;

722                         c = gctx->next_to_reclaim;
723                         gctx->next_to_reclaim = NULL;
724                         gctx->aborted = 1;

726                         (void) mutex_lock(&gctx->getent_mutex);
727                         num_reclaim_check = gctx->num_reclaim_check++;
728                         if (num_reclaim_check > 1)
729                                 gctx->in_use = 0;
730                         in_use = gctx->in_use;
731                         (void) mutex_unlock(&gctx->getent_mutex);

733                         if (in_use == 0) {
734                                 _NSCD_LOG(NSCD_LOG_GETENT_CTX,
735                                     NSCD_LOG_LEVEL_DEBUG)
736                                 (me, "process  %d exited, "
737                                     "freeing getent context = %p\n",
738                                     gctx->pid, gctx);
739                                 nssctx.ctx = (struct nss_getent_context *)gctx;
740                                 nss_endent(NULL, NULL, &nssctx);
741                         }
742                         gctx = c;
743                 }
744                 first = last = NULL;
745         }
746         /*NOTREACHED*/
747         /*LINTED E_FUNC_HAS_NO_RETURN_STMT*/
748 }

750 static nscd_rc_t
751 _nscd_init_getent_ctx_monitor()
752 {
747 _nscd_init_getent_ctx_monitor() {

754         int     errnum;
755         char    *me = "_nscd_init_getent_ctx_monitor";

757         _NSCD_LOG(NSCD_LOG_GETENT_CTX, NSCD_LOG_LEVEL_DEBUG)
758         (me, "initializing the getent context monitor\n");

760         /*
761          * the forker nscd does not process getent requests
762          * so no need to monitor orphan getent contexts
763          */
764         if (_whoami == NSCD_FORKER)
765                 return (NSCD_SUCCESS);

767         /*
768          * start a thread to reclaim unused getent contexts
769          */
770         if (thr_create(NULL, NULL, reclaim_getent_ctx,
771             NULL, THR_DETACHED, NULL) != 0) {
772                 errnum = errno;
773                 _NSCD_LOG(NSCD_LOG_GETENT_CTX, NSCD_LOG_LEVEL_ERROR)
774                 (me, "thr_create: %s\n", strerror(errnum));
775                 return (NSCD_THREAD_CREATE_ERROR);
776         }

778         return (NSCD_SUCCESS);
779 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   35384 Mon Oct 15 13:24:37 2018
new/usr/src/cmd/nscd/nscd_selfcred.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2012 Milan Jurik. All rights reserved.
 * Copyright 2018 Joyent Inc.
 */

#include <stdio.h>
#include <stdlib.h>
#include <synch.h>
#include <thread.h>
#include <string.h>
#include <errno.h>
#include <dlfcn.h>
#include <door.h>
#include <libscf.h>
#include <ucred.h>
#include <sys/varargs.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <sys/proc.h>
#include <procfs.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <libscf.h>
#include "nscd_door.h"
#include "nscd_config.h"
#include "nscd_log.h"
#include "nscd_frontend.h"
#include "nscd_selfcred.h"
#include "nscd_admin.h"
#include "nscd_common.h"
#include "ns_sldap.h"

extern int      _logfd;
static char     *execpath;
static char     **execargv;
static char     *selfcred_dbs = NULL;
```



```
static void *get_smf_prop(const char *var, char type, void *def_val);

/* current self-cred configuration data being used */
static nscd_cfg_global_selfcred_t       nscd_selfcred_cfg_g;

#define _NSCD_PUN_BLOCK 1024
static uint8_t  pu_nscd_enabled;
static int      max_pu_nscd = _NSCD_PUN_BLOCK;
static int      pu_nscd_ttl;

static nscd_rc_t setup_ldap_backend();
static nscd_rc_t init_user_proc_monitor();

/*
 * clild state
 */
typedef enum {
        CHILD_STATE_NONE        = 0,
        CHILD_STATE_UIDKNOWN,
        CHILD_STATE_FORKSENT,
        CHILD_STATE_PIDKNOWN
} child_state_t;
```

*unchanged_portion_omitted_*

```
/*ARGSUSED*/
static void *
forker_monitor(
        void            *arg)
{
        pid_t           fpid;
        char            *fmri;
        char            *me = "forker_monitor";

        (void) thr_setname(thr_self(), me);

        /* wait until forker exits */
        fpid = forker_pid;
        (void) selfcred_pulse(forking_door);

        _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
        (me, "forker (pid = %d) exited or crashed, "
            "killing all child processes\n", fpid);

        (void) mutex_lock(&forking_lock);
        forking_door = -1;
        forker_pid = -1;
        (void) mutex_unlock(&forking_lock);

        /* forker exited/crashed, kill all the child processes */
        _nscd_kill_all_children();

        /* restart forker */
        _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
        (me, "restarting the forker ...\n");

        switch (fpid = fork1()) {
        case (pid_t)-1:
                _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
                (me, "unable to fork and start the forker ...\n");

                /* enter the maintenance mode */
                if ((fmri = getenv("SMF_FMRI")) != NULL) {
                        _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
                        (me, "entering maintenance mode ...\n");
                        (void) smf_maintain_instance(fmri, SMF_TEMPORARY);
```

```
 461                 }
 462                 return ((void *)1);
 463         case 0:
 464                 _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
 465                 (me, "execv path = %s\n", execpath);

 467                 (void) execv(execpath, execargv);
 468                 exit(0);
 469         default:
 470                 _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_DEBUG)
 471                 (me, "new forker's pid is %d\n", fpid);
 472                 forker_pid = fpid;
 473                 break;
 474         }

 476         return (NULL);
 477 }
```

unchanged_portion_omitted_

```
1459 /*
1460  * FUNCTION: check_user_process
1461  */
1462 /*ARGSUSED*/
1463 static void *
1464 check_user_process(void *arg)
1465 {

1467         DIR             *dp;
1468         struct dirent   *ep;
1469         int             found;
1470         char            *me = "check_user_process";

1472         (void) thr_setname(thr_self(), me);

1474         for (;;) {
1475                 (void) sleep(60);

1477                 found = 0;

1479                 /*
1480                  * search the /proc directory and look at each process
1481                  */
1482                 if ((dp = opendir("/proc")) == NULL) {
1483                         _NSCD_LOG(NSCD_LOG_SELF_CRED, NSCD_LOG_LEVEL_ERROR)
1484                         (me, "unable to open the /proc directory\n");
1485                         continue;
1486                 }

1488                 /* for each active process */
1489                 while (ep = readdir(dp)) {
1490                         if (ep->d_name[0] == '.')    /* skip . and .. */
1491                                 continue;
1492                         if (check_uid(ep->d_name) == 0) {
1493                                 found = 1;
1494                                 break;
1495                         }
1496                 }

1498                 /*
1499                  * if no process running as the PUN uid found, exit
1500                  * to kill this PUN
1501                  */
1502                 if (found == 0) {
1503                         (void) closedir(dp);
1504                         exit(1);
```



```
1505                 }
1506                 (void) closedir(dp);
1507         }
1508         /*LINTED E_FUNC_HAS_NO_RETURN_STMT*/
1509 }
```

unchanged_portion_omitted_

```
**********************************************************
    5600 Mon Oct 15 13:24:38 2018
new/usr/src/cmd/nscd/nscd_smfmonitor.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 */

#include <stdlib.h>
#include <libscf.h>
#include <string.h>
#include "nscd_switch.h"
#include "nscd_log.h"
#include "nscd_door.h"

extern int      _whoami;

/*
 * Service states monitored by nscd. Protected by
 * readers/writer lock nscd_smf_service_state_lock
 */
nscd_smf_state_t *nscd_smf_service_state;
static rwlock_t nscd_smf_service_state_lock = DEFAULTRWLOCK;
/*
 * init service state table
 */
nscd_rc_t
_nscd_alloc_service_state_table()
{
        int i;

        nscd_smf_service_state = calloc(NSCD_NUM_SMF_FMRI,
            sizeof (nscd_smf_state_t));

        if (nscd_smf_service_state == NULL)
                return (NSCD_NO_MEMORY);

        for (i = 1; i < NSCD_NUM_SMF_FMRI; i++)
                NSCD_SMF_SVC_STATE(i) = NSCD_SVC_STATE_UNINITED;

        return (NSCD_SUCCESS);
```



```
}
_____unchanged_portion_omitted_

/* ARGSUSED */
static void *
set_smf_state(void *arg)
{

        int     i;
        int     st;

        (void) thr_setname(thr_self(), "set_smf_state");

        /*
         * the forker nscd needs not monitor the state
         * of the client services
         */
        if (_whoami == NSCD_FORKER)
                thr_exit(0);

        /*CONSTCOND*/
        while (1) {

                /* skip the first service which is nscd */
                for (i = 1; i < NSCD_NUM_SMF_FMRI; i++) {
                        st = query_smf_state(i);
                        if (st == NSCD_SVC_STATE_UNINITED)
                                break;
                }

                (void) sleep(NSCD_SW_CFG_G.check_smf_state_interval_g);
        }
        /* NOTREACHED */
        /*LINTED E_FUNC_HAS_NO_RETURN_STMT*/
}
_____unchanged_portion_omitted_
```

```
**********************************************************
    1503 Mon Oct 15 13:24:40 2018
new/usr/src/cmd/prstat/Makefile.com
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2018, Joyent, Inc.
#
# cmd/prstat/Makefile.com
#

PROG = prstat
OBJS = prstat.o prfile.o prtable.o prsort.o prutil.o
SRCS = $(OBJS:%.o=../%.c)

include ../../Makefile.cmd

CSTD = $(CSTD_GNU99)
CFLAGS += $(CCVERBOSE)
CERRWARN += -_gcc=-Wno-parentheses
LDLIBS += -lcurses -lproject
LINTFLAGS += -u
LINTFLAGS64 += -u

FILEMODE = 0555

.KEEP_STATE:

.PARALLEL : $(OBJS)

all: $(PROG)

clean:
        $(RM) $(OBJS)

$(PROG): $(OBJS)
        $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
        $(POST_PROCESS)

%.o:    ../%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)
```



```
lint:
        $(LINT.c) $(SRCS) $(LDLIBS)

include ../../Makefile.targ
```

```
**********************************************************
   45910 Mon Oct 15 13:24:41 2018
new/usr/src/cmd/prstat/prstat.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Portions Copyright 2009 Chad Mynhier
 * Copyright 2018 Joyent, Inc.  All rights reserved.
 */

#include <sys/types.h>
#include <sys/resource.h>
#include <sys/loadavg.h>
#include <sys/time.h>
#include <sys/pset.h>
#include <sys/vm_usage.h>
#include <zone.h>
#include <libzonecfg.h>

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <dirent.h>
#include <string.h>
#include <errno.h>
#include <poll.h>
#include <ctype.h>
#include <fcntl.h>
#include <limits.h>
#include <signal.h>
#include <time.h>
#include <project.h>

#include <langinfo.h>
#include <libintl.h>
#include <locale.h>

#include "prstat.h"
#include "prutil.h"
```



```
#include "prtable.h"
#include "prsort.h"
#include "prfile.h"

/*
 * x86 <sys/regs.h> ERR conflicts with <curses.h> ERR.  For the purposes
 * of this file, we care about the curses.h ERR so include that last.
 */

#if     defined(ERR)
#undef  ERR
#endif

#ifndef TEXT_DOMAIN                     /* should be defined by cc -D */
#define TEXT_DOMAIN     "SYS_TEST"      /* use this only if it wasn't */
#endif

#include <curses.h>
#include <term.h>

#define LOGIN_WIDTH     8
#define ZONE_WIDTH      28
#define PROJECT_WIDTH   28

#define PSINFO_HEADER_PROC \
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP       "
#define PSINFO_HEADER_PROC_LGRP \
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU LGRP PROCESS/NLWP  "
#define PSINFO_HEADER_LWP \
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/LWP        "
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/LWPID      "
#define PSINFO_HEADER_LWP_LGRP \
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU LGRP PROCESS/LWP   "
"   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU LGRP PROCESS/LWPID "
#define USAGE_HEADER_PROC \
"   PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/NLWP  "
#define USAGE_HEADER_LWP \
"   PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWP   "
"   PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWPID "
#define USER_HEADER_PROC \
" NPROC USERNAME  SWAP   RSS MEMORY      TIME  CPU                            "
#define USER_HEADER_LWP \
"  NLWP USERNAME  SWAP   RSS MEMORY      TIME  CPU                            "
#define TASK_HEADER_PROC \
"TASKID    NPROC  SWAP   RSS MEMORY      TIME  CPU PROJECT                    "
#define TASK_HEADER_LWP \
"TASKID     NLWP  SWAP   RSS MEMORY      TIME  CPU PROJECT                    "
#define PROJECT_HEADER_PROC \
"PROJID    NPROC  SWAP   RSS MEMORY      TIME  CPU PROJECT                    "
#define PROJECT_HEADER_LWP \
"PROJID     NLWP  SWAP   RSS MEMORY      TIME  CPU PROJECT                    "
#define ZONE_HEADER_PROC \
"ZONEID    NPROC  SWAP   RSS MEMORY      TIME  CPU ZONE                       "
#define ZONE_HEADER_LWP \
"ZONEID     NLWP  SWAP   RSS MEMORY      TIME  CPU ZONE                       "
#define PSINFO_LINE \
"%6d %-8s %5s %5s %-6s %3s  %3s %9s %3.3s%% %s"
"%6d %-8s %5s %5s %-6s %3s  %3s %9s %3.3s%% %-.16s/%d"
#define PSINFO_LINE_LGRP \
"%6d %-8s %5s %5s %-6s %3s  %3s %9s %3.3s%% %4d %s"
"%6d %-8s %5s %5s %-6s %3s  %3s %9s %3.3s%% %4d %-.16s/%d"
#define USAGE_LINE \
"%6d %-8s %3.3s %3.3s %3.3s %3.3s %3.3s %3.3s %3.3s %3.3s %3.3s %3.3s "\
"%3.3s %3.3s %s"
"%3.3s %3.3s %-.12s/%d"
#define USER_LINE \
```

```
 121 "%6d %-8s %5.5s %5.5s   %3.3s%% %9s %3.3s%%"
 122 #define TASK_LINE \
 123 "%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"
 124 #define PROJECT_LINE \
 125 "%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"
 126 #define ZONE_LINE \
 127 "%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"

 129 #define TOTAL_LINE \
 130 "Total: %d processes, %d lwps, load averages: %3.2f, %3.2f, %3.2f"

 132 /* global variables */

 134 static char     *t_ulon;                        /* termcap: start underline */
 135 static char     *t_uloff;                       /* termcap: end underline */
 136 static char     *t_up;                          /* termcap: cursor 1 line up */
 137 static char     *t_eol;                         /* termcap: clear end of line */
 138 static char     *t_smcup;                       /* termcap: cursor mvcap on */
 139 static char     *t_rmcup;                       /* termcap: cursor mvcap off */
 140 static char     *t_home;                        /* termcap: move cursor home */
 141 static char     *movecur = NULL;                /* termcap: move up string */
 142 static char     *empty_string = "\0";           /* termcap: empty string */
 143 static uint_t   print_movecur = FALSE;          /* print movecur or not */
 144 static int      is_curses_on = FALSE;           /* current curses state */

 146 static table_t  pid_tbl = {0, 0, NULL};         /* selected processes */
 147 static table_t  cpu_tbl = {0, 0, NULL};         /* selected processors */
 148 static table_t  set_tbl = {0, 0, NULL};         /* selected processor sets */
 149 static table_t  prj_tbl = {0, 0, NULL};         /* selected projects */
 150 static table_t  tsk_tbl = {0, 0, NULL};         /* selected tasks */
 151 static table_t  lgr_tbl = {0, 0, NULL};         /* selected lgroups */
 152 static zonetbl_t zone_tbl = {0, 0, NULL};       /* selected zones */
 153 static uidtbl_t euid_tbl = {0, 0, NULL};        /* selected effective users */
 154 static uidtbl_t ruid_tbl = {0, 0, NULL};        /* selected real users */

 156 static uint_t   total_procs;                    /* total number of procs */
 157 static uint_t   total_lwps;                     /* total number of lwps */
 158 static float    total_cpu;                      /* total cpu usage */
 159 static float    total_mem;                      /* total memory usage */

 161 static list_t   lwps;                           /* list of lwps/processes */
 162 static list_t   users;                          /* list of users */
 163 static list_t   tasks;                          /* list of tasks */
 164 static list_t   projects;                       /* list of projects */
 165 static list_t   zones;                          /* list of zones */
 166 static list_t   lgroups;                        /* list of lgroups */

 168 static volatile uint_t sigwinch = 0;
 169 static volatile uint_t sigtstp = 0;
 170 static volatile uint_t sigterm = 0;

 172 static long pagesize;

 174 /* default settings */

 176 optdesc_t opts = {
 175 static optdesc_t opts = {
 177         5,                      /* interval between updates, seconds */
 178         15,                     /* number of lines in top part */
 179         5,                      /* number of lines in bottom part */
 180         -1,                     /* number of iterations; infinitely */
 181         OPT_PSINFO | OPT_FULLSCREEN | OPT_USEHOME | OPT_TERMCAP,
 182         -1                      /* sort in decreasing order */
 183 };
```

_____***unchanged_portion_omitted_***


```
 351 /*
 352  * A routine to display the contents of the list on the screen
 353  */
 354 static void
 355 list_print(list_t *list)
 356 {
 357         lwp_info_t *lwp;
 358         id_info_t *id;
 359         char usr[4], sys[4], trp[4], tfl[4];
 360         char dfl[4], lck[4], slp[4], lat[4];
 361         char vcx[4], icx[4], scl[4], sig[4];
 362         char psize[6], prssize[6], pmem[6], pcpu[6], ptime[12];
 363         char pstate[7], pnice[4], ppri[4];
 364         char pname[LOGNAME_MAX+1];
 365         char name[PRFNSZ + THREAD_NAME_MAX + 2];
 366         char projname[PROJNAME_MAX+1];
 367         char zonename[ZONENAME_MAX+1];
 368         float cpu, mem;
 369         double loadavg[3] = {0, 0, 0};
 370         int i, n;
 368         int i, lwpid;

 372         if (list->l_size == 0)
 373                 return;

 375         if (foreach_element(&set_tbl, &loadavg, psetloadavg) == 0) {
 376                 /*
 377                  * If processor sets aren't specified, we display system-wide
 378                  * load averages.
 379                  */
 380                 (void) getloadavg(loadavg, 3);
 381         }

 383         if (((opts.o_outpmode & OPT_UDATE) || (opts.o_outpmode & OPT_DDATE)) &&
 384             ((list->l_type == LT_LWPS) || !(opts.o_outpmode & OPT_SPLIT)))
 385                 print_timestamp();
 386         if (opts.o_outpmode & OPT_TTY)
 387                 (void) putchar('\r');
 388         (void) putp(t_ulon);

 390         n = opts.o_cols;
 391         switch (list->l_type) {
 392         case LT_PROJECTS:
 393                 if (opts.o_outpmode & OPT_LWPS)
 394                         n = printf(PROJECT_HEADER_LWP);
 391                         (void) printf(PROJECT_HEADER_LWP);
 395                 else
 396                         n = printf(PROJECT_HEADER_PROC);
 393                         (void) printf(PROJECT_HEADER_PROC);
 397                 break;
 398         case LT_TASKS:
 399                 if (opts.o_outpmode & OPT_LWPS)
 400                         n = printf(TASK_HEADER_LWP);
 397                         (void) printf(TASK_HEADER_LWP);
 401                 else
 402                         n = printf(TASK_HEADER_PROC);
 399                         (void) printf(TASK_HEADER_PROC);
 403                 break;
 404         case LT_ZONES:
 405                 if (opts.o_outpmode & OPT_LWPS)
 406                         n = printf(ZONE_HEADER_LWP);
 403                         (void) printf(ZONE_HEADER_LWP);
 407                 else
 408                         n = printf(ZONE_HEADER_PROC);
 405                         (void) printf(ZONE_HEADER_PROC);
 409                 break;
```

```
410         case LT_USERS:
411                 if (opts.o_outpmode & OPT_LWPS)
412                         n = printf(USER_HEADER_LWP);
409                         (void) printf(USER_HEADER_LWP);
413                 else
414                         n = printf(USER_HEADER_PROC);
411                         (void) printf(USER_HEADER_PROC);
415                 break;
416         case LT_LWPS:
417                 if (opts.o_outpmode & OPT_LWPS) {
418                         if (opts.o_outpmode & OPT_PSINFO) {
419                                 if (opts.o_outpmode & OPT_LGRP)
420                                         n = printf(PSINFO_HEADER_LWP_LGRP);
417                                         (void) printf(PSINFO_HEADER_LWP_LGRP);
421                                 else
422                                         n = printf(PSINFO_HEADER_LWP);
419                                         (void) printf(PSINFO_HEADER_LWP);
423                         }
424                         if (opts.o_outpmode & OPT_MSACCT)
425                                 n = printf(USAGE_HEADER_LWP);
422                                 (void) printf(USAGE_HEADER_LWP);
426                 } else {
427                         if (opts.o_outpmode & OPT_PSINFO) {
428                                 if (opts.o_outpmode & OPT_LGRP)
429                                         n = printf(PSINFO_HEADER_PROC_LGRP);
426                                         (void) printf(PSINFO_HEADER_PROC_LGRP);
430                                 else
431                                         n = printf(PSINFO_HEADER_PROC);
428                                         (void) printf(PSINFO_HEADER_PROC);
432                         }
433                         if (opts.o_outpmode & OPT_MSACCT)
434                                 n = printf(USAGE_HEADER_PROC);
431                                 (void) printf(USAGE_HEADER_PROC);
435                 }
436                 break;
437         }

439         /* Pad out the header line so the underline spans the whole width */
440         if ((opts.o_outpmode & OPT_TERMCAP) && n < opts.o_cols)
441                 (void) printf("%*s", (int)(opts.o_cols - n), "");

443         (void) putp(t_uloff);
444         (void) putp(t_eol);
445         (void) putchar('\n');

447         for (i = 0; i < list->l_used; i++) {
448                 switch (list->l_type) {
449                 case LT_PROJECTS:
450                 case LT_TASKS:
451                 case LT_USERS:
452                 case LT_ZONES:
453                         id = list->l_ptrs[i];
454                         /*
455                          * CPU usage and memory usage normalization
456                          */
457                         if (total_cpu >= 100)
458                                 cpu = (100 * id->id_pctcpu) / total_cpu;
459                         else
460                                 cpu = id->id_pctcpu;
461                         if (id->id_sizematch == B_FALSE && total_mem >= 100)
462                                 mem = (100 * id->id_pctmem) / total_mem;
463                         else
464                                 mem = id->id_pctmem;
465                         if (list->l_type == LT_USERS) {
466                                 pwd_getname(id->id_uid, pname, sizeof (pname),
467                                     opts.o_outpmode & OPT_NORESOLVE,
```



```
468                                     opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
469                                     LOGIN_WIDTH);
470                         } else if (list->l_type == LT_ZONES) {
471                                 getzonename(id->id_zoneid, zonename,
472                                     sizeof (zonename),
473                                     opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
474                                     ZONE_WIDTH);
475                         } else {
476                                 getprojname(id->id_projid, projname,
477                                     sizeof (projname),
478                                     opts.o_outpmode & OPT_NORESOLVE,
479                                     opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
480                                     PROJECT_WIDTH);
481                         }
482                         Format_size(psize, id->id_size, 6);
483                         Format_size(prssize, id->id_rssize, 6);
484                         Format_pct(pmem, mem, 4);
485                         Format_pct(pcpu, cpu, 4);
486                         Format_time(ptime, id->id_time, 10);
487                         if (opts.o_outpmode & OPT_TTY)
488                                 (void) putchar('\r');
489                         if (list->l_type == LT_PROJECTS)
490                                 (void) printf(PROJECT_LINE, (int)id->id_projid,
491                                     id->id_nproc, psize, prssize, pmem, ptime,
492                                     pcpu, projname);
493                         else if (list->l_type == LT_TASKS)
494                                 (void) printf(TASK_LINE, (int)id->id_taskid,
495                                     id->id_nproc, psize, prssize, pmem, ptime,
496                                     pcpu, projname);
497                         else if (list->l_type == LT_ZONES)
498                                 (void) printf(ZONE_LINE, (int)id->id_zoneid,
499                                     id->id_nproc, psize, prssize, pmem, ptime,
500                                     pcpu, zonename);
501                         else
502                                 (void) printf(USER_LINE, id->id_nproc, pname,
503                                     psize, prssize, pmem, ptime, pcpu);
504                         (void) putp(t_eol);
505                         (void) putchar('\n');
506                         break;
507                 case LT_LWPS:
508                         lwp = list->l_ptrs[i];

510                         format_name(lwp, name, sizeof (name));

502                         if (opts.o_outpmode & OPT_LWPS)
503                                 lwpid = lwp->li_info.pr_lwp.pr_lwpid;
504                         else
505                                 lwpid = lwp->li_info.pr_nlwp +
506                                     lwp->li_info.pr_nzomb;
512                         pwd_getname(lwp->li_info.pr_uid, pname, sizeof (pname),
513                             opts.o_outpmode & OPT_NORESOLVE,
514                             opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
515                             LOGIN_WIDTH);

517                         if (opts.o_outpmode & OPT_PSINFO) {
518                                 Format_size(psize, lwp->li_info.pr_size, 6);
519                                 Format_size(prssize, lwp->li_info.pr_rssize, 6);
520                                 Format_state(pstate,
521                                     lwp->li_info.pr_lwp.pr_sname,
522                                     lwp->li_info.pr_lwp.pr_onpro, 7);
523                                 if (strcmp(lwp->li_info.pr_lwp.pr_clname,
524                                     "RT") == 0 ||
525                                     strcmp(lwp->li_info.pr_lwp.pr_clname,
526                                     "SYS") == 0 ||
527                                     lwp->li_info.pr_lwp.pr_sname == 'Z')
528                                         (void) strcpy(pnice, "  -");
```

```
529                                 else
530                                         Format_num(pnice,
531                                             lwp->li_info.pr_lwp.pr_nice - NZERO,
532                                             4);
533                                 Format_num(ppri, lwp->li_info.pr_lwp.pr_pri, 4);
534                                 Format_pct(pcpu,
535                                     FRC2PCT(lwp->li_info.pr_lwp.pr_pctcpu), 4);
536                                 if (opts.o_outpmode & OPT_LWPS)
537                                         Format_time(ptime,
538                                             lwp->li_info.pr_lwp.pr_time.tv_sec,
539                                             10);
540                                 else
541                                         Format_time(ptime,
542                                             lwp->li_info.pr_time.tv_sec, 10);
543                                 if (opts.o_outpmode & OPT_TTY)
544                                         (void) putchar('\r');
539                                 stripfname(lwp->li_info.pr_fname);
545                                 if (opts.o_outpmode & OPT_LGRP) {
546                                         (void) printf(PSINFO_LINE_LGRP,
547                                             (int)lwp->li_info.pr_pid, pname,
548                                             psize, prssize, pstate,
549                                             ppri, pnice, ptime, pcpu,
550                                             lwp->li_info.pr_lwp.pr_lgrp, name);
545                                             (int)lwp->li_info.pr_lwp.pr_lgrp,
546                                             lwp->li_info.pr_fname, lwpid);
551                                 } else {
552                                         (void) printf(PSINFO_LINE,
553                                             (int)lwp->li_info.pr_pid, pname,
554                                             psize, prssize, pstate, ppri, pnice,
555                                             ptime, pcpu, name);
550                                             psize, prssize,
551                                             pstate, ppri, pnice,
552                                             ptime, pcpu,
553                                             lwp->li_info.pr_fname, lwpid);
556                                 }
557                                 (void) putp(t_eol);
558                                 (void) putchar('\n');
559                         }
560                         if (opts.o_outpmode & OPT_MSACCT) {
561                                 Format_pct(usr, lwp->li_usr, 4);
562                                 Format_pct(sys, lwp->li_sys, 4);
563                                 Format_pct(slp, lwp->li_slp, 4);
564                                 Format_num(vcx, lwp->li_vcx, 4);
565                                 Format_num(icx, lwp->li_icx, 4);
566                                 Format_num(scl, lwp->li_scl, 4);
567                                 Format_num(sig, lwp->li_sig, 4);
568                                 Format_pct(trp, lwp->li_trp, 4);
569                                 Format_pct(tfl, lwp->li_tfl, 4);
570                                 Format_pct(dfl, lwp->li_dfl, 4);
571                                 Format_pct(lck, lwp->li_lck, 4);
572                                 Format_pct(lat, lwp->li_lat, 4);
573                                 if (opts.o_outpmode & OPT_TTY)
574                                         (void) putchar('\r');
573                                 stripfname(lwp->li_info.pr_fname);
575                                 (void) printf(USAGE_LINE,
576                                     (int)lwp->li_info.pr_pid, pname,
577                                     usr, sys, trp, tfl, dfl, lck,
578                                     slp, lat, vcx, icx, scl, sig,
579                                     name);
578                                     lwp->li_info.pr_fname, lwpid);
580                                 (void) putp(t_eol);
581                                 (void) putchar('\n');
582                         }
583                         break;
584                 }
585         }
```



```
587         if (opts.o_outpmode & OPT_TTY)
588                 (void) putchar('\r');
589         if (opts.o_outpmode & OPT_TERMCAP) {
590                 switch (list->l_type) {
591                 case LT_PROJECTS:
592                 case LT_USERS:
593                 case LT_TASKS:
594                 case LT_ZONES:
595                         while (i++ < opts.o_nbottom) {
596                                 (void) putp(t_eol);
597                                 (void) putchar('\n');
598                         }
599                         break;
600                 case LT_LWPS:
601                         while (i++ < opts.o_ntop) {
602                                 (void) putp(t_eol);
603                                 (void) putchar('\n');
604                         }
605                 }
606         }

608         if (opts.o_outpmode & OPT_TTY)
609                 (void) putchar('\r');

611         if ((opts.o_outpmode & OPT_SPLIT) && list->l_type == LT_LWPS)
612                 return;

614         (void) printf(TOTAL_LINE, total_procs, total_lwps,
615             loadavg[LOADAVG_1MIN], loadavg[LOADAVG_5MIN],
616             loadavg[LOADAVG_15MIN]);
617         (void) putp(t_eol);
618         (void) putchar('\n');
619         if (opts.o_outpmode & OPT_TTY)
620                 (void) putchar('\r');
621         (void) putp(t_eol);
622         (void) fflush(stdout);
623 }
```

unchanged_portion_omitted_

```
880 static void
881 get_lwpname(pid_t pid, id_t lwpid, char *buf, size_t bufsize)
882 {
883         char *path = NULL;
884         int fd;

886         buf[0] = '\0';

888         if (asprintf(&path, "/proc/%d/lwp/%d/lwpname",
889             (int)pid, (int)lwpid) == -1)
890                 return;

892         if ((fd = open(path, O_RDONLY)) != -1) {
893                 (void) read(fd, buf, bufsize);
894                 buf[bufsize - 1] = '\0';
895                 (void) close(fd);
896         }

898         free(path);
899 }

901 static void
902 add_lwp(psinfo_t *psinfo, lwpsinfo_t *lwpsinfo, int flags)
903 {
904         lwp_info_t *lwp;
905         pid_t pid = psinfo->pr_pid;
```

```
 906         id_t lwpid = lwpsinfo->pr_lwpid;

 908         if ((lwp = lwpid_get(pid, lwpid)) == NULL)
 909                 lwp = list_add_lwp(&lwps, pid, lwpid);
 910         lwp->li_flags &= ~LWP_REPRESENT;
 911         lwp->li_flags |= LWP_ALIVE;
 912         lwp->li_flags |= flags;
 913         (void) memcpy(&lwp->li_info, psinfo,
 914             sizeof (psinfo_t) - sizeof (lwpsinfo_t));
 915         (void) memcpy(&lwp->li_info.pr_lwp, lwpsinfo, sizeof (lwpsinfo_t));
 916         get_lwpname(pid, lwpid, lwp->li_lwpname, sizeof (lwp->li_lwpname));
 917 }
______unchanged_portion_omitted_

1138 static void
1139 curses_on(void)
1116 curses_on()
1140 {
1141         if ((opts.o_outpmode & OPT_TERMCAP) && (is_curses_on == FALSE)) {
1142                 (void) initscr();
1143                 (void) nonl();
1144                 (void) putp(t_smcup);
1145                 is_curses_on = TRUE;
1146         }
1147 }

1149 static void
1150 curses_off(void)
1127 curses_off()
1151 {
1152         if ((is_curses_on == TRUE) && (opts.o_outpmode & OPT_TERMCAP)) {
1153                 (void) putp(t_rmcup);
1154                 (void) endwin();
1155                 is_curses_on = FALSE;
1156         }
1157         (void) fflush(stdout);
1158 }

1160 static int
1161 nlines(int *linesp, int *colsp)
1138 nlines()
1162 {
1163         struct winsize ws;
1164         char *envp;
1165         int n;

1167         *linesp = -1;
1168         *colsp = -1;
1169         if (ioctl(STDOUT_FILENO, TIOCGWINSZ, &ws) != -1) {
1170                 if (ws.ws_row > 0)
1171                         *linesp = ws.ws_row;
1172                 if (ws.ws_col > 0)
1173                         *colsp = ws.ws_col;
1174                 if (ws.ws_row > 0 && ws.ws_col > 0)
1175                         return (0);
1145                         return (ws.ws_row);
1176         }

1178         if ((envp = getenv("LINES")) != NULL) {
1147         if (envp = getenv("LINES")) {
1179                 if ((n = Atoi(envp)) > 0) {
1180                         opts.o_outpmode &= ~OPT_USEHOME;
1181                         *linesp = n;
1150                         return (n);
1182                 }
1183         }
```



```
1184         if ((envp = getenv("COLUMNS")) != NULL) {
1185                 if ((n = Atoi(envp)) > 0) {
1186                         *colsp = n;
1187                 }
1188         }

1190         return ((*linesp > 0 && *colsp > 0) ? 0 : -1);
1153         return (-1);
1191 }

1193 static void
1194 setmovecur(void)
1157 setmovecur()
1195 {
1196         int i, n;
1197         if ((opts.o_outpmode & OPT_FULLSCREEN) &&
1198             (opts.o_outpmode & OPT_USEHOME)) {
1199                 movecur = t_home;
1200                 return;
1201         }
1202         if (opts.o_outpmode & OPT_SPLIT) {
1203                 if (opts.o_ntop == 0)
1204                         n = opts.o_nbottom + 1;
1205                 else
1206                         n = opts.o_ntop + opts.o_nbottom + 2;
1207         } else {
1208                 if (opts.o_outpmode & OPT_USERS)
1209                         n = opts.o_nbottom + 1;
1210                 else
1211                         n = opts.o_ntop + 1;
1212         }
1213         if (((opts.o_outpmode & OPT_UDATE) || (opts.o_outpmode & OPT_DDATE)))
1214                 n++;

1216         if (movecur != NULL && movecur != empty_string && movecur != t_home)
1217                 free(movecur);
1218         movecur = Zalloc(strlen(t_up) * (n + 5));
1219         for (i = 0; i <= n; i++)
1220                 (void) strcat(movecur, t_up);
1221 }

1223 static int
1224 setsize(void)
1187 setsize()
1225 {
1226         static int oldn = 0;
1227         int cols, n, ret;
1190         int n;

1229         if (opts.o_outpmode & OPT_FULLSCREEN) {
1230                 ret = nlines(&n, &cols);
1231                 if (ret != -1)
1232                         opts.o_cols = cols;
1193                 n = nlines();
1233                 if (n == oldn)
1234                         return (0);
1235                 oldn = n;
1236                 if (ret == -1) {
1197                 if (n == -1) {
1237                         opts.o_outpmode &= ~OPT_USEHOME;
1238                         setmovecur();           /* set default window size */
1239                         return (1);
1240                 }
1241                 n = n - 3;      /* minus header, total and cursor lines */
1242                 if ((opts.o_outpmode & OPT_UDATE) ||
1243                     (opts.o_outpmode & OPT_DDATE))
```

```
1244                        n--;    /* minus timestamp */
1245                if (n < 1)
1246                        Die(gettext("window is too small (try -n)\n"));
1247                if (opts.o_outpmode & OPT_SPLIT) {
1248                        if (n < 8) {
1249                                Die(gettext("window is too small (try -n)\n"));
1250                        } else {
1251                                opts.o_ntop = (n / 4) * 3;
1252                                opts.o_nbottom = n - 1 - opts.o_ntop;
1253                        }
1254                } else {
1255                        if (opts.o_outpmode & OPT_USERS)
1256                                opts.o_nbottom = n;
1257                        else
1258                                opts.o_ntop = n;
1259                }
1260        }
1261        setmovecur();
1262        return (1);
1263 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6029 Mon Oct 15 13:24:44 2018
new/usr/src/cmd/prstat/prstat.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Portions Copyright 2009 Chad Mynhier
 * Copyright 2018 Joyent, Inc.  All rights reserved.
 */

#ifndef _PRSTAT_H
#define _PRSTAT_H

#include <sys/sysmacros.h>
#include <sys/time.h>
#include <sys/types.h>
#include <procfs.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * FRC2PCT macro is used to convert 16-bit binary fractions in the range
 * 0.0 to 1.0 with binary point to the right of the high order bit
 * (i.e. 1.0 == 0x8000) to percentage value.
 */

#define FRC2PCT(pp)     (((float)(pp))/0x8000*100)

#define TIME2NSEC(__t)\
(hrtime_t)(((hrtime_t)__t.tv_sec * (hrtime_t)NANOSEC) + (hrtime_t)__t.tv_nsec)
#define TIME2SEC(__t)\
(hrtime_t)(__t.tv_sec)

/*
 * List of available output modes
 */
#define OPT_PSINFO      0x0001          /* read process's data from "psinfo" */
```



```
#define OPT_LWPS        0x0002          /* report about all lwps */
#define OPT_USERS       0x0004          /* report about most active users */
#define OPT_UNUSED      0x0008          /* reserved for future use */
#define OPT_REALTIME    0x0010          /* real-time scheduling class flag */
#define OPT_MSACCT      0x0020          /* microstate accounting flag */
#define OPT_TERMCAP     0x0040          /* use termcap data to move cursor */
#define OPT_SPLIT       0x0080          /* split-screen mode flag */
#define OPT_TTY         0x0100          /* report results to tty or file */
#define OPT_FULLSCREEN  0x0200          /* full-screen mode flag */
#define OPT_USEHOME     0x0400          /* use 'home' to move cursor up */
#define OPT_TASKS       0x0800          /* report about system tasks */
#define OPT_PROJECTS    0x1000          /* report about system projects */
#define OPT_ZONES       0x2000          /* report about zones */
#define OPT_PSETS       0x4000          /* report for specified psets */
#define OPT_LGRP        0x8000          /* report home lgroups */
#define OPT_UDATE       0x20000         /* print unix timestamp */
#define OPT_DDATE       0x40000         /* print timestamp in date(1) format */
#define OPT_NORESOLVE   0x80000         /* no nsswitch lookups */
#define OPT_TRUNC       0x100000        /* truncate long names */

/*
 * Flags to keep track of process or lwp status
 */
#define LWP_ALIVE       0x0008          /* this pid/lwp still exists */
#define LWP_REPRESENT   0x0010          /* this LWP represents the process */

/*
 * Possible list types
 */
#define LT_LWPS         0x0001
#define LT_USERS        0x0002
#define LT_TASKS        0x0004
#define LT_PROJECTS     0x0008
#define LT_ZONES        0x0010
#define LT_LGRPS        0x0020

/*
 * Linked list of per-process or per-lwp statistics
 */
typedef struct lwp_info {
        psinfo_t        li_info;        /* data read from psinfo file */
        prusage_t       li_usage;       /* data read from usage file */
        ulong_t         li_key;         /* value of the key for this lwp */
        int             li_flags;       /* process/lwp flags */
        float           li_usr;         /* user level CPU time */
        float           li_sys;         /* system call CPU time */
        float           li_trp;         /* other system trap CPU time */
        float           li_tfl;         /* text page fault sleep time */
        float           li_dfl;         /* data page fault sleep time */
        float           li_lck;         /* user lock wait sleep time */
        float           li_slp;         /* all other sleep time */
        float           li_lat;         /* wait-cpu (latency) time */
        ulong_t         li_vcx;         /* voluntary context switches */
        ulong_t         li_icx;         /* involuntary context switches */
        ulong_t         li_scl;         /* system calls */
        ulong_t         li_sig;         /* received signals */
        char            li_lwpname[THREAD_NAME_MAX];
        struct lwp_info *li_next;       /* pointer to next lwp */
        struct lwp_info *li_prev;       /* pointer to previous lwp */
} lwp_info_t;
```

*unchanged_portion_omitted_*

```
/*
 * Command line options
 */
typedef struct optdesc {
```

```
 166         int             o_interval;     /* interval between updates */
 167         int             o_ntop;         /* number of lines in top half */
 168         int             o_nbottom;      /* number of lines in bottom half */
 169         int             o_count;        /* number of iterations */
 170         int             o_outpmode;     /* selected output mode */
 171         int             o_sortorder;    /* +1 ascending, -1 descending */
 172         int             o_cols;         /* number of columns */
 173 } optdesc_t;

 175 extern optdesc_t opts;

 177 #ifdef  __cplusplus
 178 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    9536 Mon Oct 15 13:24:45 2018
new/usr/src/cmd/prstat/prutil.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Portions Copyright 2009 Chad Mynhier
 * Copyright 2018 Joyent, Inc.  All rights reserved.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/resource.h>
#include <sys/priocntl.h>
#include <sys/rtpriocntl.h>
#include <sys/tspriocntl.h>
#include <zone.h>

#include <libintl.h>
#include <limits.h>
#include <wchar.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdio_ext.h>
#include <errno.h>
#include <ctype.h>
#include <poll.h>
#include <project.h>

#include "prfile.h"
#include "prstat.h"
#include "prutil.h"

static char PRG_FMT[] = "%s: ";
static char ERR_FMT[] = ": %s\n";
static char *progname;
static char projbuf[PROJECT_BUFSZ];
```



```
#define RLIMIT_NOFILE_MAX       32767

/*PRINTFLIKE1*/
void
Warn(char *format, ...)
{
        int err = errno;
        va_list alist;

        if (progname != NULL)
                (void) fprintf(stderr, PRG_FMT, progname);
        va_start(alist, format);
        (void) vfprintf(stderr, format, alist);
        va_end(alist);
        if (strchr(format, '\n') == NULL)
                (void) fprintf(stderr, gettext(ERR_FMT), strerror(err));
}
```

*unchanged_portion_omitted_*

```
/*
 * Remove all unprintable characters from process name
 */
static void
stripfname(char *buf, size_t bufsize, const char *pname)
void
stripfname(char *buf)
{
        int bytesleft = PRFNSZ;
        wchar_t wchar;
        int length;
        char *cp;

        (void) strlcpy(buf, pname, bufsize);

        buf[bytesleft - 1] = '\0';

        for (cp = buf; *cp != '\0'; cp += length) {
                length = mbtowc(&wchar, cp, MB_LEN_MAX);
                if (length <= 0) {
                        *cp = '\0';
                        break;
                }
                if (!iswprint(wchar)) {
                        if (bytesleft <= length) {
                                *cp = '\0';
                                break;
                        }
                        (void) memmove(cp, cp + length, bytesleft - length);
                        length = 0;
                }
                bytesleft -= length;
        }
}


/*
 * prstat has always implicitly wanted a terminal width of at least 80 columns
 * (when a TTY is present).  If run in a terminal narrower than 80 columns,
 * prstat output may wrap.  For wider terminals, we allow the last column to use
 * the additional space.
 *
 * We never truncate if using -c, or not outputting to a TTY.
 */
static int
format_namewidth(void)
```

```
369 {
370         int prefixlen = 0;

372         if (opts.o_cols == 0 || !(opts.o_outpmode & (OPT_TERMCAP | OPT_TRUNC)))
373                 return (0);

375         if (opts.o_outpmode & OPT_PSINFO) {
376                 if (opts.o_outpmode & OPT_LGRP)
377                         prefixlen = 64;
378                 else
379                         prefixlen = 59;
380         } else if (opts.o_outpmode & OPT_MSACCT) {
381                 prefixlen = 64;
382         }

384         return (opts.o_cols - prefixlen);
385 }

387 void
388 format_name(lwp_info_t *lwp, char *buf, size_t buflen)
389 {
390         int pname_width = PRFNSZ;
391         char nr_suffix[20];
392         char pname[PRFNSZ];
393         int width;
394         int n;

396         stripfname(pname, sizeof (pname), lwp->li_info.pr_fname);

398         if (opts.o_outpmode & OPT_LWPS) {
399                 n = snprintf(nr_suffix, sizeof (nr_suffix), "%d",
400                     lwp->li_info.pr_lwp.pr_lwpid);
401         } else {
402                 n = snprintf(nr_suffix, sizeof (nr_suffix), "%d",
403                     lwp->li_info.pr_nlwp + lwp->li_info.pr_nzomb);
404         }

406         width = format_namewidth();

408         /* If we're over budget, truncate the process name not the LWP part. */
409         if (strlen(pname) > (width - n - 1)) {
410                 pname_width = width - n - 1;
411                 pname[pname_width - 1] = '*';
412         }

414         if ((opts.o_outpmode & OPT_LWPS) && lwp->li_lwpname[0] != '\0') {
415                 n = snprintf(buf, buflen, "%.*s/%s [%s]", pname_width,
416                     pname, nr_suffix, lwp->li_lwpname);
417         } else {
418                 n = snprintf(buf, buflen, "%.*s/%s", pname_width,
419                     pname, nr_suffix);
420         }

422         if (width > 0 && strlen(buf) > width)
423                 buf[width] = '\0';
424 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    1933 Mon Oct 15 13:24:46 2018
new/usr/src/cmd/prstat/prutil.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Portions Copyright 2009 Chad Mynhier
 * Copyright 2018, Joyent, Inc.
 */

#ifndef _PRUTIL_H
#define _PRUTIL_H

#include <sys/processor.h>
#include <sys/types.h>

#ifdef  __cplusplus
extern "C" {
#endif

extern void Die(char *, ...);
extern void Warn(char *, ...);
extern void Progname(char *);
extern void Usage();
extern int Atoi(char *);
extern void Format_size(char *, size_t, int);
extern void Format_pct(char *, float, int);
extern void Format_num(char *, int, int);
extern void Format_time(char *, ulong_t, int);
extern void Format_state(char *, char, processorid_t, int);
extern void *Realloc(void *, size_t);
extern void *Malloc(size_t);
extern void *Zalloc(size_t);
extern int Setrlimit();
extern void Priocntl(char *);
extern void getprojname(projid_t, char *, size_t, int, int, size_t);
extern void getzonename(projid_t, char *, size_t, int, size_t);
extern void format_name(lwp_info_t *, char *, size_t);
extern void stripfname(char *);
```



```
#ifdef  __cplusplus
}
_____unchanged_portion_omitted_
```

```
**********************************************************
   61962 Mon Oct 15 13:24:48 2018
new/usr/src/cmd/ps/ps.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright (c) 2018, Joyent, Inc.
 * Copyright (c) 2012, Joyent, Inc. All rights reserved.
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*
 * ps -- print things about processes.
 */
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <pwd.h>
#include <grp.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mkdev.h>
#include <unistd.h>
#include <stdlib.h>
#include <limits.h>
#include <dirent.h>
#include <sys/signal.h>
#include <sys/fault.h>
#include <sys/syscall.h>
#include <sys/time.h>
#include <procfs.h>
#include <locale.h>
#include <wctype.h>
```



```
#include <wchar.h>
#include <libw.h>
#include <stdarg.h>
#include <sys/proc.h>
#include <sys/pset.h>
#include <project.h>
#include <zone.h>

#define min(a, b)       ((a) > (b) ? (b) : (a))
#define max(a, b)       ((a) < (b) ? (b) : (a))

#define NTTYS   20      /* initial size of table for -t option  */
#define SIZ     30      /* initial size of tables for -p, -s, -g, -h and -z */

/*
 * Size of buffer holding args for t, p, s, g, u, U, G, z options.
 * Set to ZONENAME_MAX, the minimum value needed to allow any
 * zone to be specified.
 */
#define ARGSIZ ZONENAME_MAX

/* Max chars in a user/group name or printed u/g id */
#define MAXUGNAME (LOGNAME_MAX+2)

/* Structure for storing user or group info */
struct ugdata {
        id_t    id;                     /* numeric user-id or group-id */
        char    name[MAXUGNAME+1];      /* user/group name, null terminated */
};
```

*unchanged_portion_omitted_*

```
enum fname {    /* enumeration of field names */
        F_USER,         /* effective user of the process */
        F_RUSER,        /* real user of the process */
        F_GROUP,        /* effective group of the process */
        F_RGROUP,       /* real group of the process */
        F_UID,          /* numeric effective uid of the process */
        F_RUID,         /* numeric real uid of the process */
        F_GID,          /* numeric effective gid of the process */
        F_RGID,         /* numeric real gid of the process */
        F_PID,          /* process id */
        F_PPID,         /* parent process id */
        F_PGID,         /* process group id */
        F_SID,          /* session id */
        F_PSR,          /* bound processor */
        F_LWP,          /* lwp-id */
        F_LWPNAME,      /* lwp name */
        F_NLWP,         /* number of lwps */
        F_OPRI,         /* old priority (obsolete) */
        F_PRI,          /* new priority */
        F_F,            /* process flags */
        F_S,            /* letter indicating the state */
        F_C,            /* processor utilization (obsolete) */
        F_PCPU,         /* percent of recently used cpu time */
        F_PMEM,         /* percent of physical memory used (rss) */
        F_OSZ,          /* virtual size of the process in pages */
        F_VSZ,          /* virtual size of the process in kilobytes */
        F_RSS,          /* resident set size of the process in kilobytes */
        F_NICE,         /* "nice" value of the process */
        F_CLASS,        /* scheduler class */
        F_STIME,        /* start time of the process, hh:mm:ss or Month Day */
        F_ETIME,        /* elapsed time of the process, [[dd-]hh:]mm:ss */
        F_TIME,         /* cpu time of the process, [[dd-]hh:]mm:ss */
        F_TTY,          /* name of the controlling terminal */
        F_ADDR,         /* address of the process (obsolete) */
        F_WCHAN,        /* wait channel (sleep condition variable) */
```

```
131         F_FNAME,        /* file name of command */
132         F_COMM,         /* name of command (argv[0] value) */
133         F_ARGS,         /* name of command plus all its arguments */
134         F_TASKID,       /* task id */
135         F_PROJID,       /* project id */
136         F_PROJECT,      /* project name of the process */
137         F_PSET,         /* bound processor set */
138         F_ZONE,         /* zone name */
139         F_ZONEID,       /* zone id */
140         F_CTID,         /* process contract id */
141         F_LGRP,         /* process home lgroup */
142         F_DMODEL        /* process data model */
143 };
______unchanged_portion_omitted_

165 static struct def_field fname[] = {
166         /* fname        header          width   minwidth */
167         { "user",       "USER",         8,      8       },
168         { "ruser",      "RUSER",        8,      8       },
169         { "group",      "GROUP",        8,      8       },
170         { "rgroup",     "RGROUP",       8,      8       },
171         { "uid",        "UID",          5,      5       },
172         { "ruid",       "RUID",         5,      5       },
173         { "gid",        "GID",          5,      5       },
174         { "rgid",       "RGID",         5,      5       },
175         { "pid",        "PID",          5,      5       },
176         { "ppid",       "PPID",         5,      5       },
177         { "pgid",       "PGID",         5,      5       },
178         { "sid",        "SID",          5,      5       },
179         { "psr",        "PSR",          3,      2       },
180         { "lwp",        "LWP",          6,      2       },
181         { "lwpname",    "LWPNAME",      32,     8       },
182         { "nlwp",       "NLWP",         4,      2       },
183         { "opri",       "PRI",          3,      2       },
184         { "pri",        "PRI",          3,      2       },
185         { "f",          "F",            2,      2       },
186         { "s",          "S",            1,      1       },
187         { "c",          "C",            2,      2       },
188         { "pcpu",       "%CPU",         4,      4       },
189         { "pmem",       "%MEM",         4,      4       },
190         { "osz",        "SZ",           4,      4       },
191         { "vsz",        "VSZ",          4,      4       },
192         { "rss",        "RSS",          4,      4       },
193         { "nice",       "NI",           2,      2       },
194         { "class",      "CLS",          4,      2       },
195         { "stime",      "STIME",        8,      8       },
196         { "etime",      "ELAPSED",      11,     7       },
197         { "time",       "TIME",         11,     5       },
198         { "tty",        "TT",           7,      7       },
199 #ifdef _LP64
200         { "addr",       "ADDR",         16,     8       },
201         { "wchan",      "WCHAN",        16,     8       },
202 #else
203         { "addr",       "ADDR",         8,      8       },
204         { "wchan",      "WCHAN",        8,      8       },
205 #endif
206         { "fname",      "COMMAND",      8,      8       },
207         { "comm",       "COMMAND",      80,     8       },
208         { "args",       "COMMAND",      80,     80      },
209         { "taskid",     "TASKID",       5,      5       },
210         { "projid",     "PROJID",       5,      5       },
211         { "project",    "PROJECT",      8,      8       },
212         { "pset",       "PSET",         3,      3       },
213         { "zone",       "ZONE",         8,      8       },
214         { "zoneid",     "ZONEID",       5,      5       },
215         { "ctid",       "CTID",         5,      5       },
```



```
216         { "lgrp",       "LGRP",         4,      2       },
217         { "dmodel",     "DMODEL",       6,      6       },
218 };
______unchanged_portion_omitted_

376 static int
377 stdmain(int argc, char **argv)
378 {
379         char    *p;
380         char    *p1;
381         char    *parg;
382         int     c;
383         int     i;
384         int     pgerrflg = 0;   /* err flg: non-numeric arg w/p & g options */
385         size_t  size, len;
386         DIR     *dirp;
387         struct dirent *dentp;
388         pid_t   maxpid;
389         pid_t   id;
390         int     ret;
391         char    loc_stime_str[32];

393         (void) setlocale(LC_ALL, "");
394 #if !defined(TEXT_DOMAIN)       /* Should be defined by cc -D */
395 #define TEXT_DOMAIN     "SYS_TEST"      /* Use this only if it weren't */
396 #endif
397         (void) textdomain(TEXT_DOMAIN);

399         (void) memset(&euid_tbl, 0, sizeof (euid_tbl));
400         (void) memset(&ruid_tbl, 0, sizeof (ruid_tbl));
401         (void) memset(&egid_tbl, 0, sizeof (egid_tbl));
402         (void) memset(&rgid_tbl, 0, sizeof (rgid_tbl));

404         kbytes_per_page = sysconf(_SC_PAGESIZE) / 1024;

406         (void) gettimeofday(&now, NULL);

408         /*
409          * calculate width of pid fields based on configured MAXPID
410          * (must be at least 5 to retain output format compatibility)
411          */
412         id = maxpid = (pid_t)sysconf(_SC_MAXPID);
413         pidwidth = 1;
414         while ((id /= 10) > 0)
415                 ++pidwidth;
416         pidwidth = pidwidth < 5 ? 5 : pidwidth;

418         fname[F_PID].width = fname[F_PPID].width = pidwidth;
419         fname[F_PGID].width = fname[F_SID].width = pidwidth;

421         /*
422          * TRANSLATION_NOTE
423          * Specify the printf format with width and precision for
424          * the STIME field.
425          */
426         len = snprintf(loc_stime_str, sizeof (loc_stime_str),
427             dcgettext(NULL, "%8.8s", LC_TIME), "STIME");
428         if (len >= sizeof (loc_stime_str))
429                 len = sizeof (loc_stime_str) - 1;

431         fname[F_STIME].width = fname[F_STIME].minwidth = len;

433         while ((c = getopt(argc, argv, "jlfceAadLPWyZHh:t:p:g:u:U:G:n:s:o:z:"))
434             != EOF)
435                 switch (c) {
436                 case 'H':               /* Show home lgroups */
```

```
437                     Hflg++;
438                     break;
439             case 'h':
440                     /*
441                      * Show processes/threads with given home lgroups
442                      */
443                     hflg++;
444                     p1 = optarg;
445                     do {
446                             int id;

448                             /*
449                              * Get all IDs in the list, verify for
450                              * correctness and place in lgrps array.
451                              */
452                             parg = getarg(&p1);
453                             /* Convert string to integer */
454                             ret = str2id(parg, (pid_t *)&id, 0,
455                                 MAX_LGRP_ID);
456                             /* Complain if ID didn't parse correctly */
457                             if (ret != 0) {
458                                     pgerrflg++;
459                                     (void) fprintf(stderr,
460                                         gettext("ps: %s "), parg);
461                                     if (ret == EINVAL)
462                                             (void) fprintf(stderr,
463                                                 gettext("is an invalid "
464                                                 "non-numeric argument"));
465                                     else
466                                             (void) fprintf(stderr,
467                                                 gettext("exceeds valid "
468                                                 "range"));
469                                     (void) fprintf(stderr,
470                                         gettext(" for -h option\n"));
471                                     continue;
472                             }

474                             /* Extend lgrps array if needed */
475                             if (nlgrps == lgrps_size) {
476                                     /* Double the size of the lgrps array */
477                                     if (lgrps_size == 0)
478                                             lgrps_size = SIZ;
479                                     lgrps_size *= 2;
480                                     lgrps = Realloc(lgrps,
481                                         lgrps_size * sizeof (int));
482                             }
483                             /* place the id in the lgrps table */
484                             lgrps[nlgrps++] = id;
485                     } while (*p1);
486                     break;
487             case 'l':               /* long listing */
488                     lflg++;
489                     break;
490             case 'f':               /* full listing */
491                     fflg++;
492                     break;
493             case 'j':
494                     jflg++;
495                     break;
496             case 'c':
497                     /*
498                      * Format output to reflect scheduler changes:
499                      * high numbers for high priorities and don't
500                      * print nice or p_cpu values.  'c' option only
501                      * effective when used with 'l' or 'f' options.
502                      */
```



```
503                     cflg++;
504                     break;
505             case 'A':               /* list every process */
506             case 'e':               /* (obsolete) list every process */
507                     Aflg++;
508                     tflg = Gflg = Uflg = uflg = pflg = gflg = sflg = 0;
509                     zflg = hflg = 0;
510                     break;
511             case 'a':
512                     /*
513                      * Same as 'e' except no session group leaders
514                      * and no non-terminal processes.
515                      */
516                     aflg++;
517                     break;
518             case 'd':       /* same as e except no session leaders */
519                     dflg++;
520                     break;
521             case 'L':       /* show lwps */
522                     Lflg++;
523                     break;
524             case 'P':       /* show bound processor */
525                     Pflg++;
526                     break;
527             case 'W':       /* truncate long names */
528                     Wflg++;
529                     break;
530             case 'y':       /* omit F & ADDR, report RSS & SZ in Kby */
531                     yflg++;
532                     break;
533             case 'n':       /* no longer needed; retain as no-op */
534                     (void) fprintf(stderr,
535                         gettext("ps: warning: -n option ignored\n"));
536                     break;
537             case 't':               /* terminals */
538 #define TSZ     30
539                     tflg++;
540                     p1 = optarg;
541                     do {
542                             char nambuf[TSZ+6];     /* for "/dev/" + '\0' */
543                             struct stat64 s;
544                             parg = getarg(&p1);
545                             p = Realloc(NULL, TSZ+1);       /* for '\0' */
546                             /* zero the buffer before using it */
547                             p[0] = '\0';
548                             size = TSZ;
549                             if (isdigit(*parg)) {
550                                     (void) strcpy(p, "tty");
551                                     size -= 3;
552                             }
553                             (void) strncat(p, parg, size);
554                             if (ntty == ttysz) {
555                                     if ((ttysz *= 2) == 0)
556                                             ttysz = NTTYS;
557                                     tty = Realloc(tty,
558                                         (ttysz + 1) * sizeof (struct tty));
559                             }
560                             tty[ntty].tdev = PRNODEV;
561                             (void) strcpy(nambuf, "/dev/");
562                             (void) strcat(nambuf, p);
563                             if (stat64(nambuf, &s) == 0)
564                                     tty[ntty].tdev = s.st_rdev;
565                             tty[ntty++].tname = p;
566                     } while (*p1);
567                     break;
568             case 'p':               /* proc ids */
```

```
569                         pflg++;
570                         p1 = optarg;
571                         do {
572                                 pid_t id;

574                                 parg = getarg(&p1);
575                                 if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
576                                         pgerrflg++;
577                                         (void) fprintf(stderr,
578                                             gettext("ps: %s "), parg);
579                                         if (ret == EINVAL)
580                                                 (void) fprintf(stderr,
581                                                     gettext("is an invalid "
582                                                     "non-numeric argument"));
583                                         else
584                                                 (void) fprintf(stderr,
585                                                     gettext("exceeds valid "
586                                                     "range"));
587                                         (void) fprintf(stderr,
588                                             gettext(" for -p option\n"));
589                                         continue;
590                                 }

592                                 if (npid == pidsz) {
593                                         if ((pidsz *= 2) == 0)
594                                                 pidsz = SIZ;
595                                         pid = Realloc(pid,
596                                             pidsz * sizeof (pid_t));
597                                 }
598                                 pid[npid++] = id;
599                         } while (*p1);
600                         break;
601                 case 's':               /* session */
602                         sflg++;
603                         p1 = optarg;
604                         do {
605                                 pid_t id;

607                                 parg = getarg(&p1);
608                                 if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
609                                         pgerrflg++;
610                                         (void) fprintf(stderr,
611                                             gettext("ps: %s "), parg);
612                                         if (ret == EINVAL)
613                                                 (void) fprintf(stderr,
614                                                     gettext("is an invalid "
615                                                     "non-numeric argument"));
616                                         else
617                                                 (void) fprintf(stderr,
618                                                     gettext("exceeds valid "
619                                                     "range"));
620                                         (void) fprintf(stderr,
621                                             gettext(" for -s option\n"));
622                                         continue;
623                                 }

625                                 if (nsessid == sessidsz) {
626                                         if ((sessidsz *= 2) == 0)
627                                                 sessidsz = SIZ;
628                                         sessid = Realloc(sessid,
629                                             sessidsz * sizeof (pid_t));
630                                 }
631                                 sessid[nsessid++] = id;
632                         } while (*p1);
633                         break;
634                 case 'g':               /* proc group */
```



```
635                         gflg++;
636                         p1 = optarg;
637                         do {
638                                 pid_t id;

640                                 parg = getarg(&p1);
641                                 if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
642                                         pgerrflg++;
643                                         (void) fprintf(stderr,
644                                             gettext("ps: %s "), parg);
645                                         if (ret == EINVAL)
646                                                 (void) fprintf(stderr,
647                                                     gettext("is an invalid "
648                                                     "non-numeric argument"));
649                                         else
650                                                 (void) fprintf(stderr,
651                                                     gettext("exceeds valid "
652                                                     "range"));
653                                         (void) fprintf(stderr,
654                                             gettext(" for -g option\n"));
655                                         continue;
656                                 }

658                                 if (ngrpid == grpidsz) {
659                                         if ((grpidsz *= 2) == 0)
660                                                 grpidsz = SIZ;
661                                         grpid = Realloc(grpid,
662                                             grpidsz * sizeof (pid_t));
663                                 }
664                                 grpid[ngrpid++] = id;
665                         } while (*p1);
666                         break;
667                 case 'u':               /* effective user name or number */
668                         uflg++;
669                         p1 = optarg;
670                         do {
671                                 parg = getarg(&p1);
672                                 add_ugentry(&euid_tbl, parg);
673                         } while (*p1);
674                         break;
675                 case 'U':               /* real user name or number */
676                         Uflg++;
677                         p1 = optarg;
678                         do {
679                                 parg = getarg(&p1);
680                                 add_ugentry(&ruid_tbl, parg);
681                         } while (*p1);
682                         break;
683                 case 'G':               /* real group name or number */
684                         Gflg++;
685                         p1 = optarg;
686                         do {
687                                 parg = getarg(&p1);
688                                 add_ugentry(&rgid_tbl, parg);
689                         } while (*p1);
690                         break;
691                 case 'o':               /* output format */
692                         p = optarg;
693                         while ((p = parse_format(p)) != NULL)
694                                 ;
695                         break;
696                 case 'z':               /* zone name or number */
697                         zflg++;
698                         p1 = optarg;
699                         do {
700                                 zoneid_t id;
```

```
702                                 parg = getarg(&p1);
703                                 if (zone_get_id(parg, &id) != 0) {
704                                         pgerrflg++;
705                                         (void) fprintf(stderr,
706                                             gettext("ps: unknown zone %s\n"),
707                                             parg);
708                                         continue;
709                                 }

711                                 if (nzoneid == zoneidsz) {
712                                         if ((zoneidsz *= 2) == 0)
713                                                 zoneidsz = SIZ;
714                                         zoneid = Realloc(zoneid,
715                                             zoneidsz * sizeof (zoneid_t));
716                                 }
717                                 zoneid[nzoneid++] = id;
718                         } while (*p1);
719                         break;
720                 case 'Z':               /* show zone name */
721                         Zflg++;
722                         break;
723                 default:                        /* error on ? */
724                         errflg++;
725                         break;
726                 }

728         if (errflg || optind < argc || pgerrflg)
729                 usage();

731         if (tflg)
732                 tty[ntty].tname = NULL;
733         /*
734          * If an appropriate option has not been specified, use the
735          * current terminal and effective uid as the default.
736          */
737         if (!(aflg|Aflg|dflg|Gflg|hflg|Uflg|uflg|tflg|pflg|gflg|sflg|zflg)) {
738                 psinfo_t info;
739                 int procfd;
740                 char *name;
741                 char pname[100];

743                 /* get our own controlling tty name using /proc */
744                 (void) snprintf(pname, sizeof (pname),
745                     "%s/self/psinfo", procdir);
746                 if ((procfd = open(pname, O_RDONLY)) < 0 ||
747                     read(procfd, (char *)&info, sizeof (info)) < 0 ||
748                     info.pr_ttydev == PRNODEV) {
749                         (void) fprintf(stderr,
750                             gettext("ps: no controlling terminal\n"));
751                         exit(1);
752                 }
753                 (void) close(procfd);

755                 i = 0;
756                 name = gettty(&info);
757                 if (*name == '?') {
758                         (void) fprintf(stderr,
759                             gettext("ps: can't find controlling terminal\n"));
760                         exit(1);
761                 }
762                 if (ntty == ttysz) {
763                         if ((ttysz *= 2) == 0)
764                                 ttysz = NTTYS;
765                         tty = Realloc(tty, (ttysz + 1) * sizeof (struct tty));
766                 }
```



```
767                 tty[ntty].tdev = info.pr_ttydev;
768                 tty[ntty++].tname = name;
769                 tty[ntty].tname = NULL;
770                 tflg++;
771                 tuid = getuid();
772         }
773         if (Aflg) {
774                 Gflg = Uflg = uflg = pflg = sflg = gflg = aflg = dflg = 0;
775                 zflg = hflg = 0;
776         }
777         if (Aflg | aflg | dflg)
778                 tflg = 0;

780         i = 0;          /* prepare to exit on name lookup errors */
781         i += uconv(&euid_tbl);
782         i += uconv(&ruid_tbl);
783         i += gconv(&egid_tbl);
784         i += gconv(&rgid_tbl);
785         if (i)
786                 exit(1);

788         /* allocate a buffer for lwpsinfo structures */
789         lpbufsize = 4096;
790         if (Lflg && (lpsinfobuf = malloc(lpbufsize)) == NULL) {
791                 (void) fprintf(stderr,
792                     gettext("ps: no memory\n"));
793                 exit(1);
794         }

796         if (fields) {   /* print user-specified header */
797                 if (do_header) {
798                         struct field *f;

800                         for (f = fields; f != NULL; f = f->next) {
801                                 if (f != fields)
802                                         (void) printf(" ");
803                                 switch (f->fname) {
804                                 case F_TTY:
805                                         (void) printf("%-*s",
806                                             f->width, f->header);
807                                         break;
808                                 case F_LWPNAME:
809                                 case F_FNAME:
810                                 case F_COMM:
811                                 case F_ARGS:
812                                         /*
813                                          * Print these headers full width
814                                          * unless they appear at the end.
815                                          */
816                                         if (f->next != NULL) {
817                                                 (void) printf("%-*s",
818                                                     f->width, f->header);
819                                         } else {
820                                                 (void) printf("%s",
821                                                     f->header);
822                                         }
823                                         break;
824                                 default:
825                                         (void) printf("%*s",
826                                             f->width, f->header);
827                                         break;
828                                 }
829                         }
830                         (void) printf("\n");
831                 }
832         } else {        /* print standard header */
```

```
833                 /*
834                  * All fields before 'PID' are printed with a trailing space
835                  * as a separator and that is how we print the headers too.
836                  */
837                 if (lflg) {
838                         if (yflg)
839                                 (void) printf("S ");
840                         else
841                                 (void) printf(" F S ");
842                 }
843                 if (Zflg)
844                         (void) printf("    ZONE ");
845                 if (fflg) {
846                         (void) printf("     UID ");
847                 } else if (lflg)
848                         (void) printf("   UID ");

850                 (void) printf("%*s", pidwidth,  "PID");
851                 if (lflg || fflg)
852                         (void) printf(" %*s", pidwidth, "PPID");
853                 if (jflg)
854                         (void) printf(" %*s %*s", pidwidth, "PGID",
855                             pidwidth, "SID");
856                 if (Lflg)
857                         (void) printf("   LWP");
858                 if (Pflg)
859                         (void) printf(" PSR");
860                 if (Lflg && fflg)
861                         (void) printf("  NLWP");
862                 if (cflg)
863                         (void) printf("  CLS PRI");
864                 else if (lflg || fflg) {
865                         (void) printf("   C");
866                         if (lflg)
867                                 (void) printf(" PRI NI");
868                 }
869                 if (lflg) {
870                         if (yflg)
871                                 (void) printf("   RSS     SZ    WCHAN");
872                         else
873                                 (void) printf("     ADDR     SZ    WCHAN");
874                 }
875                 if (fflg)
876                         (void) printf(" %s", loc_stime_str);
877                 if (Hflg)
878                         (void) printf(" LGRP");
879                 if (Lflg)
880                         (void) printf(" TTY        LTIME CMD\n");
881                 else
882                         (void) printf(" TTY         TIME CMD\n");
883         }


886         if (pflg && !(aflg|Aflg|dflg|Gflg|Uflg|uflg|hflg|tflg|gflg|sflg|zflg) &&
887             npid <= PTHRESHOLD) {
888                 /*
889                  * If we are looking at specific processes go straight
890                  * to their /proc entries and don't scan /proc.
891                  */
892                 int i;

894                 (void) qsort(pid, npid, sizeof (pid_t), pidcmp);
895                 for (i = 0; i < npid; i++) {
896                         char pname[12];

898                         if (i >= 1 && pid[i] == pid[i - 1])
```



```
899                                 continue;
900                         (void) sprintf(pname, "%d", (int)pid[i]);
901                         if (print_proc(pname) == 0)
902                                 retcode = 0;
903                 }
904         } else {
905                 /*
906                  * Determine which processes to print info about by searching
907                  * the /proc directory and looking at each process.
908                  */
909                 if ((dirp = opendir(procdir)) == NULL) {
910                         (void) fprintf(stderr,
911                             gettext("ps: cannot open PROC directory %s\n"),
912                             procdir);
913                         exit(1);
914                 }

916                 /* for each active process --- */
917                 while ((dentp = readdir(dirp)) != NULL) {
918                         if (dentp->d_name[0] == '.')    /* skip . and .. */
919                                 continue;
920                         if (print_proc(dentp->d_name) == 0)
921                                 retcode = 0;
922                 }

924                 (void) closedir(dirp);
925         }
926         return (retcode);
927 }
```

_____***unchanged_portion_omitted_***

```
1144 /*
1145  * parse_format() takes the argument to the -o option,
1146  * sets up the next output field structure, and returns
1147  * a pointer to any further output field specifier(s).
1148  * As a side-effect, it increments errflg if encounters a format error.
1149  */
1150 static char *
1151 parse_format(char *arg)
1152 {
1153         int c;
1154         char *name;
1155         char *header = NULL;
1156         int width = 0;
1157         struct def_field *df;
1158         struct field *f;

1160         while ((c = *arg) != '\0' && (c == ',' || isspace(c)))
1161                 arg++;
1162         if (c == '\0')
1163                 return (NULL);
1164         name = arg;
1165         arg = strpbrk(arg, " \t\r\v\f\n,=");
1166         if (arg != NULL) {
1167                 c = *arg;
1168                 *arg++ = '\0';
1169                 if (c == '=') {
1170                         char *s;

1172                         header = arg;
1173                         arg = NULL;
1174                         width = strlen(header);
1175                         s = header + width;
1176                         while (s > header && isspace(*--s))
1177                                 *s = '\0';
1178                         while (isspace(*header))
```

```
1179                                    header++;
1180                    }
1181            }
1182            for (df = &fname[0]; df < &fname[NFIELDS]; df++)
1183                    if (strcmp(name, df->fname) == 0) {
1184                            if (strcmp(name, "lwp") == 0 ||
1185                                strcmp(name, "lwpname") == 0)
1181                            if (strcmp(name, "lwp") == 0)
1186                                    Lflg++;
1187                            break;
1188                    }
1189            if (df >= &fname[NFIELDS]) {
1190                    (void) fprintf(stderr,
1191                        gettext("ps: unknown output format: -o %s\n"),
1192                        name);
1193                    errflg++;
1194                    return (arg);
1195            }
1196            if ((f = malloc(sizeof (*f))) == NULL) {
1197                    (void) fprintf(stderr,
1198                        gettext("ps: malloc() for output format failed, %s\n"),
1199                        err_string(errno));
1200                    exit(1);
1201            }
1202            f->next = NULL;
1203            f->fname = df - &fname[0];
1204            f->header = header? header : df->header;
1205            if (width == 0)
1206                    width = df->width;
1207            if (*f->header != '\0')
1208                    do_header = 1;
1209            f->width = max(width, df->minwidth);

1211            if (fields == NULL)
1212                    fields = last_field = f;
1213            else {
1214                    last_field->next = f;
1215                    last_field = f;
1216            }

1218            return (arg);
1219 }
```

*_____unchanged_portion_omitted_*

```
1653 static void
1654 print_field(psinfo_t *psinfo, struct field *f, const char *ttyp)
1655 {
1656            int width = f->width;
1657            struct passwd *pwd;
1658            struct group *grp;
1659            time_t cputime;
1660            int bytesleft;
1661            int wcnt;
1662            wchar_t wchar;
1663            char *cp;
1664            int length;
1665            ulong_t mask;
1666            char c = '\0', *csave = NULL;
1667            int zombie_lwp;

1669            zombie_lwp = (Lflg && psinfo->pr_lwp.pr_sname == 'Z');

1671            switch (f->fname) {
1672            case F_RUSER:
1673                    if ((pwd = getpwuid(psinfo->pr_uid)) != NULL) {
1674                            size_t nw;
```



```
1676                            nw = mbstowcs(NULL, pwd->pw_name, 0);
1677                            if (nw == (size_t)-1)
1678                                    (void) printf("%*s ", width, "ERROR");
1679                            else if (Wflg && nw > width)
1680                                    (void) wprintf(L"%.*s%c", width - 1,
1681                                        pwd->pw_name, '*');
1682                            else
1683                                    (void) wprintf(L"%*s", width, pwd->pw_name);
1684                    } else {
1685                            if (Wflg && snprintf(NULL, 0, "%u",
1686                                (psinfo->pr_uid)) > width)

1688                                    (void) printf("%*u%c", width - 1,
1689                                        psinfo->pr_uid, '*');
1690                            else
1691                                    (void) printf("%*u", width, psinfo->pr_uid);
1692                    }
1693                    break;
1694            case F_USER:
1695                    if ((pwd = getpwuid(psinfo->pr_euid)) != NULL) {
1696                            size_t nw;

1698                            nw = mbstowcs(NULL, pwd->pw_name, 0);
1699                            if (nw == (size_t)-1)
1700                                    (void) printf("%*s ", width, "ERROR");
1701                            else if (Wflg && nw > width)
1702                                    (void) wprintf(L"%.*s%c", width - 1,
1703                                        pwd->pw_name, '*');
1704                            else
1705                                    (void) wprintf(L"%*s", width, pwd->pw_name);
1706                    } else {
1707                            if (Wflg && snprintf(NULL, 0, "%u",
1708                                (psinfo->pr_euid)) > width)

1710                                    (void) printf("%*u%c", width - 1,
1711                                        psinfo->pr_euid, '*');
1712                            else
1713                                    (void) printf("%*u", width, psinfo->pr_euid);
1714                    }
1715                    break;
1716            case F_RGROUP:
1717                    if ((grp = getgrgid(psinfo->pr_gid)) != NULL)
1718                            (void) printf("%*s", width, grp->gr_name);
1719                    else
1720                            (void) printf("%*u", width, psinfo->pr_gid);
1721                    break;
1722            case F_GROUP:
1723                    if ((grp = getgrgid(psinfo->pr_egid)) != NULL)
1724                            (void) printf("%*s", width, grp->gr_name);
1725                    else
1726                            (void) printf("%*u", width, psinfo->pr_egid);
1727                    break;
1728            case F_RUID:
1729                    (void) printf("%*u", width, psinfo->pr_uid);
1730                    break;
1731            case F_UID:
1732                    (void) printf("%*u", width, psinfo->pr_euid);
1733                    break;
1734            case F_RGID:
1735                    (void) printf("%*u", width, psinfo->pr_gid);
1736                    break;
1737            case F_GID:
1738                    (void) printf("%*u", width, psinfo->pr_egid);
1739                    break;
1740            case F_PID:
```

```
1741                    (void) printf("%*d", width, (int)psinfo->pr_pid);
1742                    break;
1743            case F_PPID:
1744                    (void) printf("%*d", width, (int)psinfo->pr_ppid);
1745                    break;
1746            case F_PGID:
1747                    (void) printf("%*d", width, (int)psinfo->pr_pgid);
1748                    break;
1749            case F_SID:
1750                    (void) printf("%*d", width, (int)psinfo->pr_sid);
1751                    break;
1752            case F_PSR:
1753                    if (zombie_lwp || psinfo->pr_lwp.pr_bindpro == PBIND_NONE)
1754                            (void) printf("%*s", width, "-");
1755                    else
1756                            (void) printf("%*d", width, psinfo->pr_lwp.pr_bindpro);
1757                    break;
1758            case F_LWP:
1759                    (void) printf("%*d", width, (int)psinfo->pr_lwp.pr_lwpid);
1760                    break;
1761            case F_LWPNAME: {
1762                    char lwpname[THREAD_NAME_MAX] = "";
1763                    char *path = NULL;
1764                    int fd;

1766                    if (asprintf(&path, "%s/%d/lwp/%d/lwpname", procdir,
1767                        (int)psinfo->pr_pid, (int)psinfo->pr_lwp.pr_lwpid) != -1 &&
1768                        (fd = open(path, O_RDONLY)) != -1) {
1769                            (void) read(fd, lwpname, sizeof (lwpname));
1770                            lwpname[THREAD_NAME_MAX - 1] = '\0';
1771                            (void) close(fd);
1772                    }

1774                    free(path);

1776                    if (f->next != NULL)
1777                            (void) printf("%-*s", width, lwpname);
1778                    else
1779                            (void) printf("%s", lwpname);
1780                    break;
1781            }
1782            case F_NLWP:
1783                    (void) printf("%*d", width, psinfo->pr_nlwp + psinfo->pr_nzomb);
1784                    break;
1785            case F_OPRI:
1786                    if (zombie_lwp)
1787                            (void) printf("%*s", width, "-");
1788                    else
1789                            (void) printf("%*d", width, psinfo->pr_lwp.pr_oldpri);
1790                    break;
1791            case F_PRI:
1792                    if (zombie_lwp)
1793                            (void) printf("%*s", width, "-");
1794                    else
1795                            (void) printf("%*d", width, psinfo->pr_lwp.pr_pri);
1796                    break;
1797            case F_F:
1798                    mask = 0xffffffffUL;
1799                    if (width < 8)
1800                            mask >>= (8 - width) * 4;
1801                    (void) printf("%*lx", width, psinfo->pr_flag & mask);
1802                    break;
1803            case F_S:
1804                    (void) printf("%*c", width, psinfo->pr_lwp.pr_sname);
1805                    break;
1806            case F_C:
```



```
1807                    if (zombie_lwp)
1808                            (void) printf("%*s", width, "-");
1809                    else
1810                            (void) printf("%*d", width, psinfo->pr_lwp.pr_cpu);
1811                    break;
1812            case F_PCPU:
1813                    if (zombie_lwp)
1814                            (void) printf("%*s", width, "-");
1815                    else if (Lflg)
1816                            prtpct(psinfo->pr_lwp.pr_pctcpu, width);
1817                    else
1818                            prtpct(psinfo->pr_pctcpu, width);
1819                    break;
1820            case F_PMEM:
1821                    prtpct(psinfo->pr_pctmem, width);
1822                    break;
1823            case F_OSZ:
1824                    (void) printf("%*lu", width,
1825                        (ulong_t)psinfo->pr_size / kbytes_per_page);
1826                    break;
1827            case F_VSZ:
1828                    (void) printf("%*lu", width, (ulong_t)psinfo->pr_size);
1829                    break;
1830            case F_RSS:
1831                    (void) printf("%*lu", width, (ulong_t)psinfo->pr_rssize);
1832                    break;
1833            case F_NICE:
1834                    /* if pr_oldpri is zero, then this class has no nice */
1835                    if (zombie_lwp)
1836                            (void) printf("%*s", width, "-");
1837                    else if (psinfo->pr_lwp.pr_oldpri != 0)
1838                            (void) printf("%*d", width, psinfo->pr_lwp.pr_nice);
1839                    else
1840                            (void) printf("%*.*s", width, width,
1841                                psinfo->pr_lwp.pr_clname);
1842                    break;
1843            case F_CLASS:
1844                    if (zombie_lwp)
1845                            (void) printf("%*s", width, "-");
1846                    else
1847                            (void) printf("%*.*s", width, width,
1848                                psinfo->pr_lwp.pr_clname);
1849                    break;
1850            case F_STIME:
1851                    if (Lflg)
1852                            prtime(psinfo->pr_lwp.pr_start, width, 0);
1853                    else
1854                            prtime(psinfo->pr_start, width, 0);
1855                    break;
1856            case F_ETIME:
1857                    if (Lflg)
1858                            print_time(delta_secs(&psinfo->pr_lwp.pr_start),
1859                                width);
1860                    else
1861                            print_time(delta_secs(&psinfo->pr_start), width);
1862                    break;
1863            case F_TIME:
1864                    if (Lflg) {
1865                            cputime = psinfo->pr_lwp.pr_time.tv_sec;
1866                            if (psinfo->pr_lwp.pr_time.tv_nsec > 500000000)
1867                                    cputime++;
1868                    } else {
1869                            cputime = psinfo->pr_time.tv_sec;
1870                            if (psinfo->pr_time.tv_nsec > 500000000)
1871                                    cputime++;
1872                    }
```

```
1873                print_time(cputime, width);
1874                break;
1875        case F_TTY:
1876                (void) printf("%-*s", width, ttyp);
1877                break;
1878        case F_ADDR:
1879                if (zombie_lwp)
1880                        (void) printf("%*s", width, "-");
1881                else if (Lflg)
1882                        (void) printf("%*lx", width,
1883                            (long)psinfo->pr_lwp.pr_addr);
1884                else
1885                        (void) printf("%*lx", width, (long)psinfo->pr_addr);
1886                break;
1887        case F_WCHAN:
1888                if (!zombie_lwp && psinfo->pr_lwp.pr_wchan)
1889                        (void) printf("%*lx", width,
1890                            (long)psinfo->pr_lwp.pr_wchan);
1891                else
1892                        (void) printf("%*.*s", width, width, "-");
1893                break;
1894        case F_FNAME:
1895                /*
1896                 * Print full width unless this is the last output format.
1897                 */
1898                if (zombie_lwp) {
1899                        if (f->next != NULL)
1900                                (void) printf("%-*s", width, "<defunct>");
1901                        else
1902                                (void) printf("%s", "<defunct>");
1903                        break;
1904                }
1905                wcnt = namencnt(psinfo->pr_fname, 16, width);
1906                if (f->next != NULL)
1907                        (void) printf("%-*.*s", width, wcnt, psinfo->pr_fname);
1908                else
1909                        (void) printf("%-.*s", wcnt, psinfo->pr_fname);
1910                break;
1911        case F_COMM:
1912                if (zombie_lwp) {
1913                        if (f->next != NULL)
1914                                (void) printf("%-*s", width, "<defunct>");
1915                        else
1916                                (void) printf("%s", "<defunct>");
1917                        break;
1918                }
1919                csave = strpbrk(psinfo->pr_psargs, " \t\r\v\f\n");
1920                if (csave) {
1921                        c = *csave;
1922                        *csave = '\0';
1923                }
1924                /* FALLTHROUGH */
1925        case F_ARGS:
1926                /*
1927                 * PRARGSZ == length of cmd arg string.
1928                 */
1929                if (zombie_lwp) {
1930                        (void) printf("%-*s", width, "<defunct>");
1931                        break;
1932                }
1933                psinfo->pr_psargs[PRARGSZ-1] = '\0';
1934                bytesleft = PRARGSZ;
1935                for (cp = psinfo->pr_psargs; *cp != '\0'; cp += length) {
1936                        length = mbtowc(&wchar, cp, MB_LEN_MAX);
1937                        if (length == 0)
1938                                break;
```



```
1939                        if (length < 0 || !iswprint(wchar)) {
1940                                if (length < 0)
1941                                        length = 1;
1942                                if (bytesleft <= length) {
1943                                        *cp = '\0';
1944                                        break;
1945                                }
1946                                /* omit the unprintable character */
1947                                (void) memmove(cp, cp+length, bytesleft-length);
1948                                length = 0;
1949                        }
1950                        bytesleft -= length;
1951                }
1952                wcnt = namencnt(psinfo->pr_psargs, PRARGSZ, width);
1953                /*
1954                 * Print full width unless this is the last format.
1955                 */
1956                if (f->next != NULL)
1957                        (void) printf("%-*.*s", width, wcnt,
1958                            psinfo->pr_psargs);
1959                else
1960                        (void) printf("%-.*s", wcnt,
1961                            psinfo->pr_psargs);
1962                if (f->fname == F_COMM && csave)
1963                        *csave = c;
1964                break;
1965        case F_TASKID:
1966                (void) printf("%*d", width, (int)psinfo->pr_taskid);
1967                break;
1968        case F_PROJID:
1969                (void) printf("%*d", width, (int)psinfo->pr_projid);
1970                break;
1971        case F_PROJECT:
1972                {
1973                        struct project cproj;
1974                        char proj_buf[PROJECT_BUFSZ];

1976                        if ((getprojbyid(psinfo->pr_projid, &cproj,
1977                            (void *)&proj_buf, PROJECT_BUFSZ)) == NULL) {
1978                                if (Wflg && snprintf(NULL, 0, "%d",
1979                                    ((int)psinfo->pr_projid)) > width)
1980                                        (void) printf("%.*d%c", width - 1,
1981                                            ((int)psinfo->pr_projid), '*');
1982                                else
1983                                        (void) printf("%*d", width,
1984                                            (int)psinfo->pr_projid);
1985                        } else {
1986                                size_t nw;

1988                                if (cproj.pj_name != NULL)
1989                                        nw = mbstowcs(NULL, cproj.pj_name, 0);
1990                                if (cproj.pj_name == NULL)
1991                                        (void) printf("%*s ", width, "---");
1992                                else if (nw == (size_t)-1)
1993                                        (void) printf("%*s ", width, "ERROR");
1994                                else if (Wflg && nw > width)
1995                                        (void) wprintf(L"%.*s%c", width - 1,
1996                                            cproj.pj_name, '*');
1997                                else
1998                                        (void) wprintf(L"%*s", width,
1999                                            cproj.pj_name);
2000                        }
2001                }
2002                break;
2003        case F_PSET:
2004                if (zombie_lwp || psinfo->pr_lwp.pr_bindpset == PS_NONE)
```

```
2005                        (void) printf("%*s", width, "-");
2006                else
2007                        (void) printf("%*d", width, psinfo->pr_lwp.pr_bindpset);
2008                break;
2009        case F_ZONEID:
2010                (void) printf("%*d", width, (int)psinfo->pr_zoneid);
2011                break;
2012        case F_ZONE:
2013                {
2014                        char zonename[ZONENAME_MAX];

2016                        if (getzonenamebyid(psinfo->pr_zoneid, zonename,
2017                            sizeof (zonename)) < 0) {
2018                                if (Wflg && snprintf(NULL, 0, "%d",
2019                                    ((int)psinfo->pr_zoneid)) > width)
2020                                        (void) printf("%.*d%c", width - 1,
2021                                            ((int)psinfo->pr_zoneid), '*');
2022                                else
2023                                        (void) printf("%*d", width,
2024                                            (int)psinfo->pr_zoneid);
2025                        } else {
2026                                size_t nw;

2028                                nw = mbstowcs(NULL, zonename, 0);
2029                                if (nw == (size_t)-1)
2030                                        (void) printf("%*s ", width, "ERROR");
2031                                else if (Wflg && nw > width)
2032                                        (void) wprintf(L"%.*s%c", width - 1,
2033                                            zonename, '*');
2034                                else
2035                                        (void) wprintf(L"%*s", width, zonename);
2036                        }
2037                }
2038                break;
2039        case F_CTID:
2040                if (psinfo->pr_contract == -1)
2041                        (void) printf("%*s", width, "-");
2042                else
2043                        (void) printf("%*ld", width, (long)psinfo->pr_contract);
2044                break;
2045        case F_LGRP:
2046                /* Display home lgroup */
2047                (void) printf("%*d", width, (int)psinfo->pr_lwp.pr_lgrp);
2048                break;

2050        case F_DMODEL:
2051                (void) printf("%*s", width,
2052                    psinfo->pr_dmodel == PR_MODEL_LP64 ? "_LP64" : "_ILP32");
2053                break;
2054        }
2055 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   21478 Mon Oct 15 13:24:52 2018
new/usr/src/cmd/ptools/pstack/pstack.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 */

#include <sys/isa_defs.h>

#include <stdio.h>
#include <stdio_ext.h>
#include <fcntl.h>
#include <ctype.h>
#include <string.h>
#include <signal.h>
#include <dirent.h>
#include <errno.h>
#include <stdlib.h>
#include <stdarg.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/stack.h>
#include <link.h>
#include <limits.h>
#include <libelf.h>
#include <thread_db.h>
#include <libproc.h>
#include <setjmp.h>

static  char    *command;
static  int     Fflag;
static  int     is64;
static  GElf_Sym sigh;

/*
 * To keep the list of user-level threads for a multithreaded process.
 */
struct threadinfo {
        struct threadinfo *next;
```



```
        id_t    threadid;
        id_t    lwpid;
        td_thr_state_e state;
        uintptr_t startfunc;
        uintptr_t exitval;
        prgregset_t regs;
};
_____unchanged_portion_omitted_

static  int     thr_stack(const td_thrhandle_t *, void *);
static  void    free_threadinfo(void);
static  struct threadinfo *find_thread(id_t);
static  int     all_call_stacks(pstack_handle_t *, int);
static  void    tlhead(id_t, id_t, const char *);
static  void    tlhead(id_t, id_t);
static  int     print_frame(void *, prgregset_t, uint_t, const long *);
static  void    print_zombie(struct ps_prochandle *, struct threadinfo *);
static  void    print_syscall(const lwpstatus_t *, prgregset_t);
static  void    call_stack(pstack_handle_t *, const lwpstatus_t *);

/*
 * The number of active and zombie threads.
 */
static  int     nthreads;

int
main(int argc, char **argv)
{
        int retc = 0;
        int opt;
        int errflg = FALSE;
        core_content_t content = CC_CONTENT_DATA | CC_CONTENT_ANON |
            CC_CONTENT_STACK;
        struct rlimit rlim;

        if ((command = strrchr(argv[0], '/')) != NULL)
                command++;
        else
                command = argv[0];

        /* options */
        while ((opt = getopt(argc, argv, "F")) != EOF) {
                switch (opt) {
                case 'F':
                        /*
                         * If the user specifies the force option, we'll
                         * consent to printing out other threads' stacks
                         * even if the main stack is absent.
                         */
                        content &= ~CC_CONTENT_STACK;
                        Fflag = PGRAB_FORCE;
                        break;
                default:
                        errflg = TRUE;
                        break;
                }
        }

        argc -= optind;
        argv += optind;

        if (errflg || argc <= 0) {
                (void) fprintf(stderr,
                    "usage:\t%s [-F] { pid | core }[/lwps] ...\n", command);
                (void) fprintf(stderr, "  (show process call stack)\n");
                (void) fprintf(stderr,
```

```
194                     "  -F: force grabbing of the target process\n");
195                 exit(2);
196         }

198         /*
199          * Make sure we'll have enough file descriptors to handle a target
200          * that has many many mappings.
201          */
202         if (getrlimit(RLIMIT_NOFILE, &rlim) == 0) {
203                 rlim.rlim_cur = rlim.rlim_max;
204                 (void) setrlimit(RLIMIT_NOFILE, &rlim);
205                 (void) enable_extended_FILE_stdio(-1, -1);
206         }

208         (void) proc_initstdio();

210         while (--argc >= 0) {
211                 int gcode;
212                 psinfo_t psinfo;
213                 const psinfo_t *tpsinfo;
214                 struct ps_prochandle *Pr = NULL;
215                 td_thragent_t *Tap;
216                 int threaded;
217                 pstack_handle_t handle;
218                 const char *lwps, *arg;

220                 (void) proc_flushstdio();

222                 arg = *argv++;

224                 if ((Pr = proc_arg_xgrab(arg, NULL, PR_ARG_ANY,
225                     Fflag, &gcode, &lwps)) == NULL) {
226                         (void) fprintf(stderr, "%s: cannot examine %s: %s\n",
227                             command, arg, Pgrab_error(gcode));
228                         retc++;
229                         continue;
230                 }

232                 if ((tpsinfo = Ppsinfo(Pr)) == NULL) {
233                         (void) fprintf(stderr, "%s: cannot examine %s: "
234                             "lost control of process\n", command, arg);
235                         Prelease(Pr, 0);
236                         retc++;
237                         continue;
238                 }
239                 (void) memcpy(&psinfo, tpsinfo, sizeof (psinfo_t));
240                 proc_unctrl_psinfo(&psinfo);

242                 if (Pstate(Pr) == PS_DEAD) {
243                         if ((Pcontent(Pr) & content) != content) {
244                                 (void) fprintf(stderr, "%s: core '%s' has "
245                                     "insufficient content\n", command, arg);
246                                 retc++;
247                                 continue;
248                         }
249                         (void) printf("core '%s' of %d:\t%.70s\n",
250                             arg, (int)psinfo.pr_pid, psinfo.pr_psargs);
251                 } else {
252                         (void) printf("%d:\t%.70s\n",
253                             (int)psinfo.pr_pid, psinfo.pr_psargs);
254                 }

256                 is64 = (psinfo.pr_dmodel == PR_MODEL_LP64);

258                 if (Pgetauxval(Pr, AT_BASE) != -1L && Prd_agent(Pr) == NULL) {
259                         (void) fprintf(stderr, "%s: warning: librtld_db failed "
```



```
260                             "to initialize; symbols from shared libraries will "
261                             "not be available\n", command);
262                 }

264                 /*
265                  * First we need to get a thread agent handle.
266                  */
267                 if (td_init() != TD_OK ||
268                     td_ta_new(Pr, &Tap) != TD_OK)       /* no libc */
269                         threaded = FALSE;
270                 else {
271                         /*
272                          * Iterate over all threads, calling:
273                          *   thr_stack(td_thrhandle_t *Thp, NULL);
274                          * for each one to generate the list of threads.
275                          */
276                         nthreads = 0;
277                         (void) td_ta_thr_iter(Tap, thr_stack, NULL,
278                             TD_THR_ANY_STATE, TD_THR_LOWEST_PRIORITY,
279                             TD_SIGNO_MASK, TD_THR_ANY_USER_FLAGS);

281                         (void) td_ta_delete(Tap);
282                         threaded = TRUE;
283                 }

285                 handle.proc = Pr;
286                 handle.jvm = load_libjvm(Pr);
287                 handle.pydb = load_libpython(Pr);
288                 handle.lwps = lwps;
289                 handle.count = 0;

291                 if (all_call_stacks(&handle, threaded) != 0)
292                         retc++;
293                 if (threaded)
294                         free_threadinfo();

296                 reset_libjvm(handle.jvm);
297                 reset_libpython(handle.pydb);
298                 Prelease(Pr, 0);

300                 if (handle.count == 0)
301                         (void) fprintf(stderr, "%s: no matching LWPs found\n",
302                             command);
303         }

305         (void) proc_finistdio();

307         return (retc);
308 }
_____unchanged_portion_omitted_

381 static int
382 thread_call_stack(void *data, const lwpstatus_t *psp,
383     const lwpsinfo_t *pip)
384 {
385         char lwpname[THREAD_NAME_MAX] = "";
386         pstack_handle_t *h = data;
387         lwpstatus_t lwpstatus;
388         struct threadinfo *tip;

390         if (!proc_lwp_in_set(h->lwps, pip->pr_lwpid))
391                 return (0);
392         h->count++;

394         if ((tip = find_thread(pip->pr_lwpid)) == NULL)
395                 return (0);
```

```
397         (void) Plwp_getname(h->proc, pip->pr_lwpid,
398             lwpname, sizeof (lwpname));

400         tlhead(tip->threadid, pip->pr_lwpid, lwpname);
394         tlhead(tip->threadid, pip->pr_lwpid);
401         tip->threadid = 0;      /* finish eliminating tid */
402         if (psp)
403                 call_stack(h, psp);
404         else {
405                 if (tip->state == TD_THR_ZOMBIE)
406                         print_zombie(h->proc, tip);
407                 else {
408                         (void) memset(&lwpstatus, 0, sizeof (lwpstatus));
409                         (void) memcpy(lwpstatus.pr_reg, tip->regs,
410                             sizeof (prgregset_t));
411                         call_stack(h, &lwpstatus);
412                 }
413         }
414         return (0);
415 }

417 static int
418 lwp_call_stack(void *data,
419     const lwpstatus_t *psp, const lwpsinfo_t *pip)
420 {
421         char lwpname[THREAD_NAME_MAX] = "";
422         pstack_handle_t *h = data;

424         if (!proc_lwp_in_set(h->lwps, pip->pr_lwpid))
425                 return (0);
426         h->count++;

428         (void) Plwp_getname(h->proc, pip->pr_lwpid,
429             lwpname, sizeof (lwpname));

431         tlhead(0, pip->pr_lwpid, lwpname);
421         tlhead(0, pip->pr_lwpid);
432         if (psp)
433                 call_stack(h, psp);
434         else
435                 (void) printf("\t** zombie "
436                     "(exited, not detached, not yet joined) **\n");
437         return (0);
438 }

440 static int
441 all_call_stacks(pstack_handle_t *h, int dothreads)
442 {
443         struct ps_prochandle *Pr = h->proc;
444         pstatus_t status = *Pstatus(Pr);

446         (void) memset(&sigh, 0, sizeof (GElf_Sym));
447         (void) Plookup_by_name(Pr, "libc.so", "sigacthandler", &sigh);

449         if ((status.pr_nlwp + status.pr_nzomb) <= 1 &&
450             !(dothreads && nthreads > 1)) {
451                 if (proc_lwp_in_set(h->lwps, status.pr_lwp.pr_lwpid)) {
452                         call_stack(h, &status.pr_lwp);
453                         h->count++;
454                 }
455         } else {
456                 lwpstatus_t lwpstatus;
457                 struct threadinfo *tip;
458                 id_t tid;
```



```
460                 if (dothreads)
461                         (void) Plwp_iter_all(Pr, thread_call_stack, h);
462                 else
463                         (void) Plwp_iter_all(Pr, lwp_call_stack, h);

465                 /* for each remaining thread w/o an lwp */
466                 (void) memset(&lwpstatus, 0, sizeof (lwpstatus));
467                 for (tip = thr_head; tip; tip = tip->next) {

469                         if (!proc_lwp_in_set(h->lwps, tip->lwpid))
470                                 tip->threadid = 0;

472                         if ((tid = tip->threadid) != 0) {
473                                 (void) memcpy(lwpstatus.pr_reg, tip->regs,
474                                     sizeof (prgregset_t));
475                                 tlhead(tid, tip->lwpid, NULL);
465                                 tlhead(tid, tip->lwpid);
476                                 if (tip->state == TD_THR_ZOMBIE)
477                                         print_zombie(Pr, tip);
478                                 else
479                                         call_stack(h, &lwpstatus);
480                         }
481                         tip->threadid = 0;
482                         tip->lwpid = 0;
483                 }
484         }
485         return (0);
486 }

488 /* The width of the header */
489 #define HEAD_WIDTH      (62)
490 static void
491 tlhead(id_t threadid, id_t lwpid, const char *name)
479 tlhead(id_t threadid, id_t lwpid)
492 {
493         char buf[128] = { 0 };
494         char num[16];
495         ssize_t amt = 0;
496         int i;

498         if (threadid == 0 && lwpid == 0)
499                 return;

501         if (lwpid > 0) {
502                 (void) snprintf(num, sizeof (num), "%d", (int)lwpid);
503                 (void) strlcat(buf, "thread# ", sizeof (buf));
504                 (void) strlcat(buf, num, sizeof (buf));
505         }
484         (void) printf("-----------------");

507         if (threadid > 0) {
508                 (void) snprintf(num, sizeof (num), "%d", (int)threadid);
509                 if (lwpid > 0)
510                         (void) strlcat(buf, " / ", sizeof (buf));
511                 (void) strlcat(buf, "lwp# ", sizeof (buf));
512                 (void) strlcat(buf, num, sizeof (buf));
513         }
486         if (threadid && lwpid)
487                 (void) printf("  lwp# %d / thread# %d  ",
488                     (int)lwpid, (int)threadid);
489         else if (threadid)
490                 (void) printf("---------  thread# %d  ", (int)threadid);
491         else if (lwpid)
492                 (void) printf("  lwp# %d  ------------", (int)lwpid);

515         if (name != NULL && strlen(name) > 0) {
```

```
516                 (void) strlcat(buf, " [", sizeof (buf));
517                 (void) strlcat(buf, name, sizeof (buf));
518                 (void) strlcat(buf, "]", sizeof (buf));
519         }

521         amt = (HEAD_WIDTH - strlen(buf) - 2);
522         if (amt < 4)
523                 amt = 4;

525         for (i = 0; i < amt / 2; i++)
526                 (void) putc('-', stdout);
527         (void) printf(" %s ", buf);
528         for (i = 0; i < (amt / 2) + (amt % 2); i++)
529                 (void) putc('-', stdout);
530         (void) putc('\n', stdout);
494         (void) printf("--------------------\n");
531 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   57488 Mon Oct 15 13:24:55 2018
new/usr/src/cmd/sgs/elfdump/common/corenote.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
______unchanged_portion_omitted_


1259 static void
1260 dump_lwpname(note_state_t *state, const char *title)
1261 {
1262         const sl_prlwpname_layout_t *layout = state->ns_arch->prlwpname;

1264         indent_enter(state, title, &layout->pr_lwpid);

1266         PRINT_DEC(MSG_ORIG(MSG_CNOTE_T_PR_LWPID), pr_lwpid);
1267         PRINT_STRBUF(MSG_ORIG(MSG_CNOTE_T_PR_LWPNAME), pr_lwpname);

1269         indent_exit(state);
1270 }


1273 /*
1274  * Print percent from 16-bit binary fraction [0 .. 1]
1275  * Round up .01 to .1 to indicate some small percentage (the 0x7000 below).
1276  *
1277  * Note: This routine was copied from ps(1) and then modified.
1278  */
1279 static const char *
1280 prtpct_value(note_state_t *state, const sl_field_t *fdesc,
1281     sl_fmtbuf_t buf)
1282 {
1283         uint_t value;           /* need 32 bits to compute with */

1285         value = extract_as_word(state, fdesc);
1286         value = ((value * 1000) + 0x7000) >> 15;        /* [0 .. 1000] */
1287         if (value >= 1000)
1288                 value = 999;

1290         (void) snprintf(buf, sizeof (sl_fmtbuf_t),
1291             MSG_ORIG(MSG_CNOTE_FMT_PRTPCT), value / 10, value % 10);

1293         return (buf);
1294 }
______unchanged_portion_omitted_

1762 corenote_ret_t
1763 corenote(Half mach, int do_swap, Word type,
1764     const char *desc, Word descsz)
1765 {
1766         note_state_t            state;

1768         /*
1769          * Get the per-architecture layout definition
1770          */
1771         state.ns_mach = mach;
1772         state.ns_arch = sl_mach(state.ns_mach);
1773         if (sl_mach(state.ns_mach) == NULL)
1774                 return (CORENOTE_R_BADARCH);

1776         state.ns_swap = do_swap;
1777         state.ns_indent = 4;
1778         state.ns_t2col = state.ns_v2col = 0;
1779         state.ns_data = desc;
1780         state.ns_len = descsz;
```



```
1782         switch (type) {
1783         case NT_PRSTATUS:               /* prstatus_t <sys/old_procfs.h> */
1784                 state.ns_vcol = 26;
1785                 state.ns_t2col = 46;
1786                 state.ns_v2col = 60;
1787                 dump_prstatus(&state, MSG_ORIG(MSG_CNOTE_DESC_PRSTATUS_T));
1788                 return (CORENOTE_R_OK);

1790         case NT_PRFPREG:                /* prfpregset_t <sys/procfs_isa.h> */
1791                 return (CORENOTE_R_OK_DUMP);

1793         case NT_PRPSINFO:               /* prpsinfo_t   <sys/old_procfs.h> */
1794                 state.ns_vcol = 20;
1795                 state.ns_t2col = 41;
1796                 state.ns_v2col = 54;
1797                 dump_prpsinfo(&state, MSG_ORIG(MSG_CNOTE_DESC_PRPSINFO_T));
1798                 return (CORENOTE_R_OK);

1800         case NT_PRXREG:                 /* prxregset_t <sys/procfs_isa.h> */
1801                 return (CORENOTE_R_OK_DUMP);

1803         case NT_PLATFORM:               /* string from sysinfo(SI_PLATFORM) */
1804                 dbg_print(0, MSG_ORIG(MSG_NOTE_DESC));
1805                 dbg_print(0, MSG_ORIG(MSG_FMT_INDENT), safe_str(desc, descsz));
1806                 return (CORENOTE_R_OK);

1808         case NT_AUXV:                   /* auxv_t array <sys/auxv.h> */
1809                 state.ns_vcol = 18;
1810                 dump_auxv(&state, MSG_ORIG(MSG_CNOTE_DESC_AUXV_T));
1811                 return (CORENOTE_R_OK);

1813         case NT_GWINDOWS:               /* gwindows_t SPARC only */
1814                 return (CORENOTE_R_OK_DUMP);

1816         case NT_ASRS:                   /* asrset_t <sys/regset> sparcv9 only */
1817                 state.ns_vcol = 18;
1818                 state.ns_t2col = 38;
1819                 state.ns_v2col = 46;
1820                 dump_asrset(&state, MSG_ORIG(MSG_CNOTE_DESC_ASRSET_T));
1821                 return (CORENOTE_R_OK);

1823         case NT_LDT:                    /* ssd array <sys/sysi86.h> IA32 only */
1824                 return (CORENOTE_R_OK_DUMP);

1826         case NT_PSTATUS:                /* pstatus_t <sys/procfs.h> */
1827                 state.ns_vcol = 22;
1828                 state.ns_t2col = 42;
1829                 state.ns_v2col = 54;
1830                 dump_pstatus(&state, MSG_ORIG(MSG_CNOTE_DESC_PSTATUS_T));
1831                 return (CORENOTE_R_OK);

1833         case NT_PSINFO:                 /* psinfo_t <sys/procfs.h> */
1834                 state.ns_vcol = 25;
1835                 state.ns_t2col = 45;
1836                 state.ns_v2col = 58;
1837                 dump_psinfo(&state, MSG_ORIG(MSG_CNOTE_DESC_PSINFO_T));
1838                 return (CORENOTE_R_OK);

1840         case NT_PRCRED:                 /* prcred_t <sys/procfs.h> */
1841                 state.ns_vcol = 20;
1842                 state.ns_t2col = 34;
1843                 state.ns_v2col = 44;
1844                 dump_prcred(&state, MSG_ORIG(MSG_CNOTE_DESC_PRCRED_T));
1845                 return (CORENOTE_R_OK);
```

```
1847         case NT_UTSNAME:               /* struct utsname <sys/utsname.h> */
1848                 state.ns_vcol = 18;
1849                 dump_utsname(&state, MSG_ORIG(MSG_CNOTE_DESC_STRUCT_UTSNAME));
1850                 return (CORENOTE_R_OK);

1852         case NT_LWPSTATUS:             /* lwpstatus_t <sys/procfs.h> */
1853                 state.ns_vcol = 24;
1854                 state.ns_t2col = 44;
1855                 state.ns_v2col = 54;
1856                 dump_lwpstatus(&state, MSG_ORIG(MSG_CNOTE_DESC_LWPSTATUS_T));
1857                 return (CORENOTE_R_OK);

1859         case NT_LWPSINFO:              /* lwpsinfo_t <sys/procfs.h> */
1860                 state.ns_vcol = 22;
1861                 state.ns_t2col = 42;
1862                 state.ns_v2col = 54;
1863                 dump_lwpsinfo(&state, MSG_ORIG(MSG_CNOTE_DESC_LWPSINFO_T));
1864                 return (CORENOTE_R_OK);

1866         case NT_PRPRIV:                /* prpriv_t <sys/procfs.h> */
1867                 state.ns_vcol = 21;
1868                 state.ns_t2col = 34;
1869                 state.ns_v2col = 38;
1870                 dump_prpriv(&state, MSG_ORIG(MSG_CNOTE_DESC_PRPRIV_T));
1871                 return (CORENOTE_R_OK);

1873         case NT_PRPRIVINFO:            /* priv_impl_info_t <sys/priv.h> */
1874                 state.ns_vcol = 29;
1875                 state.ns_t2col = 41;
1876                 state.ns_v2col = 56;
1877                 dump_priv_impl_info(&state,
1878                     MSG_ORIG(MSG_CNOTE_DESC_PRIV_IMPL_INFO_T));
1879                 return (CORENOTE_R_OK);

1881         case NT_CONTENT:               /* core_content_t <sys/corectl.h> */
1882                 if (sizeof (core_content_t) > descsz)
1883                         return (CORENOTE_R_BADDATA);
1884                 {
1885                         static sl_field_t fdesc = { 0, 8, 0, 0 };
1886                         Conv_cnote_cc_content_buf_t conv_buf;
1887                         core_content_t content;

1889                         state.ns_vcol = 8;
1890                         indent_enter(&state,
1891                             MSG_ORIG(MSG_CNOTE_DESC_CORE_CONTENT_T),
1892                             &fdesc);
1893                         content = extract_as_lword(&state, &fdesc);
1894                         print_str(&state, MSG_ORIG(MSG_STR_EMPTY),
1895                             conv_cnote_cc_content(content, 0, &conv_buf));
1896                         indent_exit(&state);
1897                 }
1898                 return (CORENOTE_R_OK);

1900         case NT_ZONENAME:              /* string from getzonenamebyid(3C) */
1901                 dbg_print(0, MSG_ORIG(MSG_NOTE_DESC));
1902                 dbg_print(0, MSG_ORIG(MSG_FMT_INDENT), safe_str(desc, descsz));
1903                 return (CORENOTE_R_OK);


1906         case NT_FDINFO:
1907                 state.ns_vcol = 22;
1908                 state.ns_t2col = 41;
1909                 state.ns_v2col = 54;
1910                 dump_prfdinfo(&state, MSG_ORIG(MSG_CNOTE_DESC_PRFDINFO_T));
1911                 return (CORENOTE_R_OK);
```



```
1913         case NT_SPYMASTER:
1914                 state.ns_vcol = 25;
1915                 state.ns_t2col = 45;
1916                 state.ns_v2col = 58;
1917                 dump_psinfo(&state, MSG_ORIG(MSG_CNOTE_DESC_PSINFO_T));
1918                 return (CORENOTE_R_OK);

1920         case NT_SECFLAGS:
1921                 state.ns_vcol = 23;
1922                 state.ns_t2col = 41;
1923                 state.ns_v2col = 54;
1924                 dump_secflags(&state, MSG_ORIG(MSG_CNOTE_DESC_PRSECFLAGS_T));
1925                 return (CORENOTE_R_OK);

1927         case NT_LWPNAME:
1928                 state.ns_vcol = 20;
1929                 dump_lwpname(&state, MSG_ORIG(MSG_CNOTE_DESC_PRLWPNAME_T));
1930                 return (CORENOTE_R_OK);
1931         }

1933         return (CORENOTE_R_BADTYPE);
1934 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   26697 Mon Oct 15 13:24:59 2018
new/usr/src/cmd/sgs/elfdump/common/elfdump.msg
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1997, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
# Copyright 2018 Joyent, Inc.
#

@ _START_

# Message file for cmd/sgs/elfdump.

@ MSG_ID_ELFDUMP

# Usage Messages

@ MSG_USAGE_BRIEF       "usage: %s [-cCdegGhHiklmnrSsuvy] [-I index] \
                         [-N name] [-O osabi] [-T type] [-p | -w outfile] \
                         file...\n"
@ MSG_USAGE_DETAIL1     "\t[-c]\t\tdump section header information\n"
@ MSG_USAGE_DETAIL2     "\t[-C]\t\tdemangle C++ symbol names\n"
@ MSG_USAGE_DETAIL3     "\t[-d]\t\tdump the contents of the .dynamic section\n"
@ MSG_USAGE_DETAIL4     "\t[-e]\t\tdump the elf header\n"
@ MSG_USAGE_DETAIL5     "\t[-g]\t\tdump the contents of the .group sections\n"
@ MSG_USAGE_DETAIL6     "\t[-G]\t\tdump the contents of the .got section\n"
@ MSG_USAGE_DETAIL7     "\t[-h]\t\tdump the contents of the .hash section\n"
@ MSG_USAGE_DETAIL8     "\t[-H]\t\tdump the contents of the .SUNW_cap section\n"
@ MSG_USAGE_DETAIL9     "\t[-i]\t\tdump the contents of the .interp section\n"
@ MSG_USAGE_DETAIL10    "\t[-I index]\tqualify an option with an index\n"
@ MSG_USAGE_DETAIL11    "\t[-l]\t\tdump with no truncated section names\n"
@ MSG_USAGE_DETAIL12    "\t[-k]\t\tcalculate elf checksum\n"
@ MSG_USAGE_DETAIL13    "\t[-m]\t\tdump the contents of the .SUNW_move \
                         section\n"
@ MSG_USAGE_DETAIL14    "\t[-n]\t\tdump the contents of the .note section\n"
@ MSG_USAGE_DETAIL15    "\t[-N name]\tqualify an option with a 'name'\n"
@ MSG_USAGE_DETAIL16    "\t[-O osabi]\tuse given osabi to interpret object\n"
@ MSG_USAGE_DETAIL17    "\t[-P]\t\tuse program headers to generate \
                         section headers\n"
@ MSG_USAGE_DETAIL18    "\t[-p]\t\tdump the program headers\n"
@ MSG_USAGE_DETAIL19    "\t[-r]\t\tdump the contents of the relocation \
                         sections\n"
```



```
@ MSG_USAGE_DETAIL20    "\t[-S]\t\tdump the contents of the sort index \
                         sections\n"
@ MSG_USAGE_DETAIL21    "\t[-s]\t\tdump the contents of the symbol table \
                         sections\n"
@ MSG_USAGE_DETAIL22    "\t[-T type]\tqualify an option with a section or \
                         program header type\n"
@ MSG_USAGE_DETAIL23    "\t[-u]\t\tdump the contents of a frame unwind \
                         section\n"
@ MSG_USAGE_DETAIL24    "\t[-v]\t\tdump the contents of the version sections\n"
@ MSG_USAGE_DETAIL25    "\t[-w file]\twrite the contents of specified section \
                         to 'file'\n"
@ MSG_USAGE_DETAIL26    "\t[-y]\t\tdump the contents of the .SUNW_syminfo \
                         section\n"

# Errors

@ MSG_ERR_BADFILE       "%s: invalid file type\n"
@ MSG_ERR_BADREL1       "%s: bad relocation entry: %s: relocation requires \
                         symbol\n"
@ MSG_ERR_NOSHDR        "%s: section headers are not present: synthesizing \
                         from program headers (-P option)\n"
@ MSG_ERR_PNEEDSPH      "%s: file without program headers is incompatible \
                         with -P option\n"
@ MSG_ERR_NOSHSTRSEC    "%s: no header string table section (shstrtab). \
                         header names are not available\n"
@ MSG_ERR_SHSTRNULL     "%s: zero size section header string table \
                         (shstrtab). header names are not available\n"
@ MSG_ERR_FAILURE       "%s: %s failed: %s\n"
@ MSG_ERR_BADSHDRALIGN  "%s: bad section header array alignment\n"
@ MSG_ERR_BADPHDRALIGN  "%s: bad program header array alignment\n"
@ MSG_ERR_BADSHNAME     "%s: section[%d]: invalid sh_name: %lld\n"
@ MSG_ERR_BADSHLINK     "%s: %s: invalid sh_link: %d\n"
@ MSG_ERR_BADSHINFO     "%s: %s: invalid sh_info: %d\n"
@ MSG_ERR_BADSHTYPE     "%s: %s: invalid sh_type: %d\n"
@ MSG_ERR_BADALIGN      "%s: %s: bad sh_offset alignment\n"
@ MSG_ERR_BADSYM2       "%s: %s: index[%d]: bad symbol entry: %s: must be \
                         SHN_COMMON or defined in SHT_NOBITS section\n"
@ MSG_ERR_BADSYM3       "%s: %s: index[%d]: bad symbol entry: %s: must be \
                         defined in a SHF_TLS section\n"
@ MSG_ERR_BADSYM4       "%s: %s: index[%d]: bad symbol entry: %s: must be \
                         defined in a non-SHF_TLS section\n"
@ MSG_ERR_BADSYM5       "%s: %s: index[%d]: bad symbol entry: %s: \
                         invalid shndx: %d\n"
@ MSG_ERR_BADSYM6       "%s: %s: index[%d]: bad symbol entry: %s: section[%d] \
                         size: %#llx: symbol (address %#llx, size %#llx) \
                         lies outside of containing section\n"
@ MSG_ERR_BADSYM7       "%s: %s: index[%d]: suspicious global symbol entry: \
                         %s: lies within local symbol range (index < %lld)\n"
@ MSG_ERR_BADSYM8       "%s: %s: index[%d]: suspicious local symbol entry: \
                         %s: lies within global symbol range (index >= %lld)\n"

@ MSG_ERR_RELBADSYMNDX  "%s: bad symbol reference %d: from relocation \
                         entry: %d\n"
@ MSG_ERR_HSBADSYMNDX   "%s: bad symbol reference %d: from hash entry: %d\n"
@ MSG_ERR_BADSYMXINDEX1 "%s: %s: index[%d]: invalid SHN_XINDEX reference: \
                         SHT_SYMTAB_SHNDX section truncated: no entry for this \
                         symbol\n"
@ MSG_ERR_BADSYMXINDEX2 "%s: %s: index[%d]: invalid SHN_XINDEX reference: \
                         bad SHT_SYMTAB_SHNDX entry: invalid shndx: 0x%x\n"
@ MSG_ERR_BADSYMXINDEX3 "%s: %s: index[%d]: invalid SHN_XINDEX reference: \
                         SHT_SYMTAB_SHNDX section not found\n"

@ MSG_ERR_BADSTOFF      "%s: %s: index[%d]: bad %s offset: 0x%x: max 0x%x\n"
@ MSG_ERR_BADSZ         "%s: %s: zero size or zero entry size information\n"
@ MSG_ERR_BADMINFO      "%s: %s: invalid m_info: 0x%llx\n"
@ MSG_ERR_BADHASH       "%s: %s: bad hash entry: symbol %s: exists in bucket \
```

```
 127                          %d, should be bucket %ld\n"
 128 @ MSG_ERR_NODYNSYM      "%s: %s: associated SHT_DYNSYM section not found\n"
 129 @ MSG_ERR_BADNDXSEC     "%s: %s: unexpected section type associated with \
 130                          index section: %s\n"
 131 @ MSG_ERR_BADSYMNDX     "%s: %s: bad symbol index: %d\n"
 132 @ MSG_ERR_BADVER        "%s: %s: index[%d]: version %d is out of range: \
 133                          version definitions available: 0-%d\n"
 134 @ MSG_ERR_NOTSTRTAB     "%s: section[%d] is not a string table as expected \
 135                          by section[%d]\n";
 136 @ MSG_ERR_BADCHAINIDX   "%s: %s: invalid chain index %d for bucket %d (max %d)\n

 138 @ MSG_ERR_LDYNNOTADJ    "%s: bad dynamic symbol table layout: %s and %s \
 139                          sections are not adjacent\n"
 140 @ MSG_ERR_SECMEMOVER    "%s: memory overlap between section[%d]: %s: %llx:%llx \
 141                          and section[%d]: %s: %llx:%llx\n"
 142 @ MSG_ERR_SHDRMEMOVER   "%s: memory overlap between section header table: \
 143                          %llx:%llx and section[%d]: %s: %llx:%llx\n"
 144 @ MSG_ERR_MULTDYN       "%s: %d dynamic sections seen (1 expected)\n"
 145 @ MSG_ERR_DYNNOBCKSEC   "%s: object lacks %s section required by %s dynamic \
 146                          entry\n"
 147 @ MSG_ERR_DYNBADADDR    "%s: %s (%#llx) does not match \
 148                          shdr[%d: %s].sh_addr (%#llx)\n"
 149 @ MSG_ERR_DYNBADSIZE    "%s: %s (%#llx) does not match \
 150                          shdr[%d: %s].sh_size (%#llx)\n"
 151 @ MSG_ERR_DYNBADENTSIZE "%s: %s (%#llx) does not match \
 152                          shdr[%d: %s].sh_entsize (%#llx)\n"
 153 @ MSG_ERR_DYNSYMVAL     "%s: %s: symbol value does not match \
 154                          %s entry: %s: value: %#llx\n"
 155 @ MSG_ERR_MALSTR        "%s: %s: malformed string table, initial or final \
 156                          byte\n"
 157 @ MSG_ERR_MULTEHFRMHDR  "%s: [%d: %s] multiple .eh_frame_hdr sections seen \
 158                          (1 expected)\n"
 159 @ MSG_ERR_BADEHFRMPTR   "%s: section[%d: %s] FramePtr (%#llx) does not match \
 160                          shdr[%d: %s].sh_addr (%#llx)\n"
 161 @ MSG_ERR_BADSORT       "%s: %s: index[%d]: invalid sort order\n"
 162 @ MSG_ERR_BADSIDYNNDX   "%s: [%d: %s][%d]: dynamic section index out of \
 163                          range (0 - %d): %d\n";
 164 @ MSG_ERR_BADSIDYNTAG   "%s: [%d: %s][%d]: dynamic element \
 165                          [%d: %s][%d] should have type %s: %s\n";
 166 @ MSG_ERR_BADCIEFDELEN  "%s: %s: invalid CIE/FDE length: %#llx at %#llx\n"


 169 @ MSG_WARN_INVINTERP1   "%s: PT_INTERP header has no associated section\n"
 170 @ MSG_WARN_INVINTERP2   "%s: interp section: %s: and PT_INTERP program \
 171                          header have conflicting size or offsets\n"
 172 @ MSG_WARN_INVCAP1      "%s: PT_SUNWCAP header has no associated section\n"
 173 @ MSG_WARN_INVCAP2      "%s: capabilities section[%d]: %s: requires PT_CAP \
 174                          program header\n"
 175 @ MSG_WARN_INVCAP3      "%s: capabilities section[%d]: %s: and PT_CAP program \
 176                          header have conflicting size or offsets\n"
 177 @ MSG_WARN_INVCAP4      "%s: capabilities section[%d]: %s: requires string \
 178                          table: invalid sh_info: %d\n";
 179 @ MSG_WARN_INADDR32SF1  "%s: capabilities section %s: software capability \
 180                          ADDR32: is ineffective within a 32-bit object\n"
 181 @ MSG_WARN_MULTEHFRM    "%s: section[%d: %s]: %s object has multiple \
 182                         .eh_frame sections\n"

 184 @ MSG_INFO_LINUXOSABI   "%s: %s object has Linux .note.ABI-tag section. \
 185                         Assuming %s\n"

 187 @ MSG_ERR_DWOVRFLW      "%s: %s: encoded DWARF data exceeds section size\n"
 188 @ MSG_ERR_DWBADENC      "%s: %s: bad DWARF encoding: %#x\n"
 189 @ MSG_ERR_DWNOCIE       "%s: %s: no CIE prior to FDE\n"

 191 # exception_range_entry table entries.
 192 # TRANSLATION_NOTE - the following entries provide for a series of one or more
```



```
 193 # standard 32-bit and 64-bit .exception_ranges table entries that align with
 194 # the initial title.

 196 @ MSG_EXR_TITLE_32      "     index    offset     ret_addr   \
 197                          length     handler    type_blk"
 198 @ MSG_EXR_ENTRY_32      "%10.10s  0x%8.8llx 0x%8.8llx 0x%8.8llx 0x%8.8llx \
 199                          0x%8.8llx"
 200 @ MSG_EXR_TITLE_64      "     index    offset             ret_addr           \
 201                          length             handler            type_blk"
 202 @ MSG_EXR_ENTRY_64      "%10.10s  0x%16.16llx 0x%16.16llx 0x%16.16llx \
 203                          0x%16.16llx 0x%16.16llx"

 205 # Elf Output Messages

 207 @ MSG_ELF_SHDR          "Section Header[%d]:  sh_name: %s"
 208 @ MSG_ELF_PHDR          "Program Header[%d]:"

 210 @ MSG_ELF_SCN_CAP       "Capabilities Section:  %s"
 211 @ MSG_ELF_SCN_CAPCHAIN  "Capabilities Chain Section:  %s"
 212 @ MSG_ELF_SCN_INTERP    "Interpreter Section:  %s"
 213 @ MSG_ELF_SCN_VERDEF    "Version Definition Section:  %s"
 214 @ MSG_ELF_SCN_VERNEED   "Version Needed Section:  %s"
 215 @ MSG_ELF_SCN_SYMTAB    "Symbol Table Section:  %s"
 216 @ MSG_ELF_SCN_RELOC     "Relocation Section:  %s"
 217 @ MSG_ELF_SCN_UNWIND    "Unwind Section: %s"
 218 @ MSG_ELF_SCN_DYNAMIC   "Dynamic Section:  %s"
 219 @ MSG_ELF_SCN_NOTE      "Note Section:  %s"
 220 @ MSG_ELF_SCN_HASH      "Hash Section:  %s"
 221 @ MSG_ELF_SCN_SYMINFO   "Syminfo Section:  %s"
 222 @ MSG_ELF_SCN_GOT       "Global Offset Table Section:  %s"
 223 @ MSG_ELF_SCN_GRP       "Group Section:  %s"
 224 @ MSG_ELF_SCN_MOVE      "Move Section:  %s"
 225 @ MSG_ELF_SCN_SYMSORT1  "Symbol Sort Section:  %s (%s)"
 226 @ MSG_ELF_SCN_SYMSORT2  "Symbol Sort Section:  %s (%s / %s)"

 228 @ MSG_OBJ_CAP_TITLE     " Object Capabilities:"
 229 @ MSG_SYM_CAP_TITLE     " Symbol Capabilities:"
 230 @ MSG_CAPINFO_ENTRIES   "  Symbols:"
 231 @ MSG_CAPCHAIN_TITLE    " Capabilities family: %s"
 232 @ MSG_CAPCHAIN_ENTRY    "  chainndx  symndx      name"
 233 @ MSG_ERR_INVCAP        "%s: capabilities section: %s: contains symbol \
 234                          capabilities groups, but no capabilities information \
 235                          section is defined: invalid sh_link: %d\n"
 236 @ MSG_ERR_INVCAPINFO1   "%s: capabilities information section: %s: no symbol \
 237                          table is defined: invalid sh_link: %d\n"
 238 @ MSG_ERR_INVCAPINFO2   "%s: capabilities information section: %s: no \
 239                          capabilities chain is defined: invalid sh_info: %d\n"
 240 @ MSG_ERR_INVCAPINFO3   "%s: capabilities information section: %s: index %d: \
 241                          bad capabilities chain index defined: %d\n"
 242 @ MSG_ERR_CHBADSYMNDX   "%s: bad symbol reference %d: from capability chain: \
 243                          %s entry: %d\n"

 245 @ MSG_ELF_HASH_BKTS1    "%10.10s  buckets contain %8d symbols"
 246 @ MSG_ELF_HASH_BKTS2    "%10.10s  buckets         %8d symbols (globals)"
 247 @ MSG_ELF_HASH_INFO     "    bucket  symndx      name"
 248 @ MSG_HASH_OVERFLW      "%s: warning: section %s: too many symbols to count, \
 249                          bucket=%d count=%d"
 250 @ MSG_ELF_ERR_SHDR      "\tunable to obtain section header: shstrtab[%lld]\n"
 251 @ MSG_ELF_ERR_DATA      "\tunable to obtain section data: shstrtab[%lld]\n"
 252 @ MSG_ELF_ERR_SCN       "\tunable to obtain section header: section[%d]\n"
 253 @ MSG_ELF_ERR_SCNDATA   "\tunable to obtain section data: section[%d]\n"
 254 @ MSG_ARCHIVE_SYMTAB_32 "\nSymbol Table: (archive, 32-bit offsets)"
 255 @ MSG_ARCHIVE_SYMTAB_64 "\nSymbol Table: (archive, 64-bit offsets)"
 256 @ MSG_ARCHIVE_FIELDS_32 "     index    offset    member name and symbol"
 257 @ MSG_ARCHIVE_FIELDS_64 "     index    offset            member name and symbol"
```

```
259 @ MSG_GOT_MULTIPLE      "%s: multiple relocations against \
260                          the same GOT entry ndx: %d addr: 0x%llx\n"
261 @ MSG_GOT_UNEXPECTED    "%s: warning: section %s: section unexpected within \
262                          relocatable object\n"

264 # Miscellaneous clutter

266 @ MSG_STR_NULL          "(null)"
267 @ MSG_STR_DEPRECATED    "(deprecated value)"
268 @ MSG_STR_UNKNOWN       "<unknown>"
269 @ MSG_STR_SECTION       "%s (section)"
270 @ MSG_STR_CHECKSUM      "elf checksum: 0x%lx"

272 @ MSG_FMT_SCNNDX        "section[%d]"
273 @ MSG_FMT_NOTEENTNDX    "  entry [%d]";


276 @ MSG_ERR_MALLOC        "%s: malloc: %s\n"
277 @ MSG_ERR_OPEN          "%s: open: %s\n"
278 @ MSG_ERR_READ          "%s: read: %s\n"
279 @ MSG_ERR_WRITE         "%s: write: %s\n"
280 @ MSG_ERR_BAD_T_SHT     "%s: unrecognized section header type: %s\n"
281 @ MSG_ERR_BAD_T_PT      "%s: unrecognized program header type: %s\n"
282 @ MSG_ERR_BAD_T_OSABI   "%s: unrecognized operating system ABI: %s\n"
283 @ MSG_ERR_AMBIG_MATCH   "%s: ambiguous use of -I, -N, or -T. Remove \
284                          -p option or section selection option(s)\n"

286 #
287 # SHT_MOVE messages
288 #
289 @ MSG_MOVE_TITLE        "    symndx     offset   size repeat stride           \
290                          value  with respect to"
291 @ MSG_MOVE_ENTRY        "%10.10s %#10llx %6d %6d %6d %#16llx  %s"

293 #
294 # SHT_GROUP messages
295 #
296 @ MSG_GRP_TITLE         "     index    flags / section         signature symbol"
297 @ MSG_GRP_SIGNATURE     "       [0]   %-24s %s"
298 @ MSG_GRP_INVALSCN      "<invalid section>"

300 #
301 # SHT_NOTE messages
302 #
303 @ MSG_NOTE_BADDATASZ    "%s: %s: note header exceeds section size. \
304                          offset: 0x%x\n"
305 @ MSG_NOTE_BADNMSZ      "%s: %s: note name value exceeds section size. \
306                          offset: 0x%x namesize: 0x%x\n"
307 @ MSG_NOTE_BADDESZ      "%s: %s: note data size exceeds section size. \
308                          offset: 0x%x datasize: 0x%x\n"
309 @ MSG_NOTE_BADCOREARCH  "%s: elfdump core file note support not available for \
310                          architecture: %s\n"
311 @ MSG_NOTE_BADCOREDATA  "%s: elfdump core file note data truncated or \
312                          otherwise malformed\n"
313 @ MSG_NOTE_BADCORETYPE  "%s: unknown note type %#x\n"

315 @ MSG_NOTE_BAD_SECFLAGS_VER     "unknown prsecflags_t version: "

317 @ _END_

319 # The following strings represent reserved words, files, pathnames and symbols.
320 # Reference to this strings is via the MSG_ORIG() macro, and thus no message
321 # translation is required.

323 @ MSG_STR_OSQBRKT       "["
324 @ MSG_STR_CSQBRKT       "]"
```



```
326 @ MSG_GRP_COMDAT        " COMDAT "
327 @ MSG_GRP_ENTRY         "%10.10s   %s [%lld]\n"
328 @ MSG_GRP_UNKNOWN       " 0x%x "

330 @ MSG_ELF_GOT           ".got"
331 @ MSG_ELF_INIT          ".init"
332 @ MSG_ELF_FINI          ".fini"
333 @ MSG_ELF_INTERP        ".interp"

335 @ MSG_ELF_GETEHDR       "elf_getehdr"
336 @ MSG_ELF_GETPHDR       "elf_getphdr"
337 @ MSG_ELF_GETSHDR       "elf_getshdr"
338 @ MSG_ELF_GETSCN        "elf_getscn"
339 @ MSG_ELF_GETDATA       "elf_getdata"
340 @ MSG_ELF_GETARHDR      "elf_getarhdr"
341 @ MSG_ELF_GETARSYM      "elf_getarsym"
342 @ MSG_ELF_RAND          "elf_rand"
343 @ MSG_ELF_BEGIN         "elf_begin"
344 @ MSG_ELF_GETPHDRNUM    "elf_getphdrnum"
345 @ MSG_ELF_GETSHDRNUM    "elf_getshdrnum"
346 @ MSG_ELF_GETSHDRSTRNDX "elf_getshdrstrndx"
347 @ MSG_ELF_XLATETOM      "elf_xlatetom"
348 @ MSG_ELF_ARSYM         "ARSYM"

350 @ MSG_SYM_INIT          "_init"
351 @ MSG_SYM_FINI          "_fini"
352 @ MSG_SYM_GOT           "_GLOBAL_OFFSET_TABLE_"

354 @ MSG_STR_OPTIONS       "CcdeGgHhiI:klmN:nO:PprSsT:uvw:y"

356 @ MSG_STR_8SP           "        "
357 @ MSG_STR_EMPTY         ""
358 @ MSG_STR_CORE          "CORE"
359 @ MSG_STR_NOTEABITAG    ".note.ABI-tag"
360 @ MSG_STR_GNU           "GNU"
361 @ MSG_STR_LOC           "loc"
362 @ MSG_STR_INITLOC       "initloc"

364 @ MSG_FMT_INDENT        "       %s"
365 @ MSG_FMT_INDEX         " [%lld]"
366 @ MSG_FMT_INDEX2        "[%d]"
367 @ MSG_FMT_ASRINDEX      "[ asr%d ]"
368 @ MSG_FMT_INDEXRNG      "[%d-%d]"
369 @ MSG_FMT_INTEGER       " %d"
370 @ MSG_FMT_HASH_INFO     "%10.10s  %-10s  %s"
371 @ MSG_FMT_CHAIN_INFO    "%10.10s  %-10s  %s"
372 @ MSG_FMT_ARSYM1_32     "%10.10s  0x%8.8llx  (%s):%s"
373 @ MSG_FMT_ARSYM2_32     "%10.10s  0x%8.8llx"
374 @ MSG_FMT_ARSYM1_64     "%10.10s  0x%16.16llx  (%s):%s"
375 @ MSG_FMT_ARSYM2_64     "%10.10s  0x%16.16llx"
376 @ MSG_FMT_ARNAME        "%s(%s)"
377 @ MSG_FMT_NLSTR         "\n%s:"
378 @ MSG_FMT_NLSTRNL       "\n%s:\n"
379 @ MSG_FMT_SECSYM        "%.*s%s"

381 @ MSG_HEXDUMP_ROW       "%*s%-*s%s"
382 @ MSG_HEXDUMP_TOK       "%2.2x"

384 @ MSG_SUNW_OST_SGS      "SUNW_OST_SGS"

386 # Unwind info

388 @ MSG_SCN_FRM           ".eh_frame"
389 @ MSG_SCN_FRMHDR        ".eh_frame_hdr"
390 @ MSG_SCN_EXRANGE       ".exception_ranges"
```

```
392 @ MSG_UNW_FRMHDR        "Frame Header:"
393 @ MSG_UNW_FRMVERS       "  Version: %d"
394 @ MSG_UNW_FRPTRENC      "  FramePtrEnc: %-20s  FramePtr: %#llx"
395 @ MSG_UNW_FDCNENC       "  FdeCntEnc:   %-20s  FdeCnt: %lld"
396 @ MSG_UNW_TABENC        "  TableEnc:    %-20s"
397 @ MSG_UNW_BINSRTAB1     "  Binary Search Table:"
398 @ MSG_UNW_BINSRTAB2_32  "      InitialLoc    FdeLoc"
399 @ MSG_UNW_BINSRTAB2_64  "      InitialLoc          FdeLoc"
400 @ MSG_UNW_BINSRTABENT_32 "    0x%08llx    0x%08llx"
401 @ MSG_UNW_BINSRTABENT_64 "    0x%016llx  0x%016llx"
402 @ MSG_UNW_ZEROTERM      "ZERO terminator: [0x00000000]"
403 @ MSG_UNW_CIE           "CIE: [%#llx]"
404 @ MSG_UNW_CIELNGTH      " length: 0x%02x cieid: %d"
405 @ MSG_UNW_CIEVERS       " version: %d  augmentation: '%s'"
406 @ MSG_UNW_CIECALGN      " codealign: %#llx  dataalign: %lld  \
407                         retaddr: %d"
408 @ MSG_UNW_CIEAXVAL      " Augmentation Data:"
409 @ MSG_UNW_CIEAXSIZ      "   size: %lld"
410 @ MSG_UNW_CIEAXPERS     "   personality:"
411 @ MSG_UNW_CIEAXPERSENC  "       encoding: 0x%02x %s"
412 @ MSG_UNW_CIEAXPERSRTN  "       routine:  %#08llx"
413 @ MSG_UNW_CIEAXCENC     "   code pointer encoding: 0x%02x %s"
414 @ MSG_UNW_CIEAXLSDA     "   lsda encoding: 0x%02x %s"
415 @ MSG_UNW_CIEAXUNEC     "   Unexpected aug val: %c"
416 @ MSG_UNW_CIECFI        " CallFrameInstructions:"

418 @ MSG_UNW_FDE           " FDE: [%#llx]"
419 @ MSG_UNW_FDELNGTH      "    length:  %#x  cieptr: %#x"
420 @ MSG_UNW_FDEINITLOC    "    initloc: %#llx  addrrange: %#llx  endloc: %#llx"
421 @ MSG_UNW_FDEAXVAL      "   Augmentation Data:"
422 @ MSG_UNW_FDEAXSIZE     "      size: %#llx"
423 @ MSG_UNW_FDEAXLSDA     "      lsda: %#llx"
424 @ MSG_UNW_FDECFI        "   CallFrameInstructions:"

426 # Unwind section Call Frame Instructions. These all start with a leading
427 # "%*s%s", used to insert leading white space and the opcode name.

429 @ MSG_CFA_ADV_LOC       "%*s%s: %s + %llu => %#llx"
430 @ MSG_CFA_CFAOFF        "%*s%s: %s, cfa%+lld"
431 @ MSG_CFA_CFASET        "%*s%s: cfa=%#llx"
432 @ MSG_CFA_LLD           "%*s%s: %lld"
433 @ MSG_CFA_LLU           "%*s%s: %llu"
434 @ MSG_CFA_REG           "%*s%s: %s"
435 @ MSG_CFA_REG_OFFLLD    "%*s%s: %s, offset=%lld"
436 @ MSG_CFA_REG_OFFLLU    "%*s%s: %s, offset=%llu"
437 @ MSG_CFA_REG_REG       "%*s%s: %s, %s"
438 @ MSG_CFA_SIMPLE        "%*s%s"
439 @ MSG_CFA_SIMPLEREP     "%*s%s [%d]"
440 @ MSG_CFA_EBLK          "%*s%s: expr(%llu bytes)"
441 @ MSG_CFA_REG_EBLK      "%*s%s: %s, expr(%llu bytes)"

443 # Architecture specific register name formats

445 @ MSG_REG_FMT_BASIC     "r%d"
446 @ MSG_REG_FMT_NAME      "r%d (%s)"


449 # Note messages

451 @ MSG_NOTE_TYPE         "    type:   %#x"
452 @ MSG_NOTE_TYPE_STR     "    type:   %s"
453 @ MSG_NOTE_NAMESZ       "    namesz: %#x"
454 @ MSG_NOTE_NAME         "    name:"
455 @ MSG_NOTE_DESCSZ       "    descsz: %#x"
```



```
457 @ MSG_NOTE_DESC                        "    desc:"
458 @ MSG_CNOTE_DESC_ASRSET_T              "desc: (asrset_t)"
459 @ MSG_CNOTE_DESC_AUXV_T                "desc: (auxv_t)"
460 @ MSG_CNOTE_DESC_CORE_CONTENT_T        "desc: (core_content_t)"
461 @ MSG_CNOTE_DESC_LWPSINFO_T            "desc: (lwpsinfo_t)"
462 @ MSG_CNOTE_DESC_LWPSTATUS_T           "desc: (lwpstatus_t)"
463 @ MSG_CNOTE_DESC_PRCRED_T              "desc: (prcred_t)"
464 @ MSG_CNOTE_DESC_PRIV_IMPL_INFO_T      "desc: (priv_impl_info_t)"
465 @ MSG_CNOTE_DESC_PRPRIV_T              "desc: (prpriv_t)"
466 @ MSG_CNOTE_DESC_PRPSINFO_T            "desc: (prpsinfo_t)"
467 @ MSG_CNOTE_DESC_PRSTATUS_T            "desc: (prstatus_t)"
468 @ MSG_CNOTE_DESC_PSINFO_T              "desc: (psinfo_t)"
469 @ MSG_CNOTE_DESC_PSTATUS_T             "desc: (pstatus_t)"
470 @ MSG_CNOTE_DESC_STRUCT_UTSNAME        "desc: (struct utsname)"
471 @ MSG_CNOTE_DESC_PRFDINFO_T            "desc: (prfdinfo_t)"
472 @ MSG_CNOTE_DESC_PRSECFLAGS_T          "desc: (prsecflags_t)"
473 @ MSG_CNOTE_DESC_PRLWPNAME_T           "desc: (prlwpname_t)"

475 @ MSG_CNOTE_FMT_LINE            "%*s%-*s%s"
476 @ MSG_CNOTE_FMT_LINE_2UP        "%*s%-*s%-*s%-*s%s"
477 @ MSG_CNOTE_FMT_D               "%d"
478 @ MSG_CNOTE_FMT_LLD             "%lld"
479 @ MSG_CNOTE_FMT_U               "%u"
480 @ MSG_CNOTE_FMT_LLU             "%llu"
481 @ MSG_CNOTE_FMT_X               "%#x"
482 @ MSG_CNOTE_FMT_LLX             "%#llx"
483 @ MSG_CNOTE_FMT_Z2X             "0x%2.2x"
484 @ MSG_CNOTE_FMT_Z4X             "0x%4.4x"
485 @ MSG_CNOTE_FMT_Z8X             "0x%8.8x"
486 @ MSG_CNOTE_FMT_Z16LLX          "0x%16.16llx"
487 @ MSG_CNOTE_FMT_TITLE           "%*s%s"
488 @ MSG_CNOTE_FMT_AUXVLINE        "%*s%10.10s  %-*s  %s"
489 @ MSG_CNOTE_FMT_PRTPCT          "%u.%u%%"

491 @ MSG_CNOTE_T_PRIV_FLAGS                "priv_flags:"
492 @ MSG_CNOTE_T_PRIV_GLOBALINFOSIZE       "priv_globalinfosize:"
493 @ MSG_CNOTE_T_PRIV_HEADERSIZE           "priv_headersize:"
494 @ MSG_CNOTE_T_PRIV_INFOSIZE             "priv_infosize:"
495 @ MSG_CNOTE_T_PRIV_MAX                  "priv_max:"
496 @ MSG_CNOTE_T_PRIV_NSETS                "priv_nsets:"
497 @ MSG_CNOTE_T_PRIV_SETSIZE              "priv_setsize:"
498 @ MSG_CNOTE_T_PR_ACTION                 "pr_action:"
499 @ MSG_CNOTE_T_PR_ADDR                   "pr_addr:"
500 @ MSG_CNOTE_T_PR_AGENTID                "pr_agentid:"
501 @ MSG_CNOTE_T_PR_ALTSTACK               "pr_altstack:"
502 @ MSG_CNOTE_T_PR_ARGC                   "pr_argc:"
503 @ MSG_CNOTE_T_PR_ARGV                   "pr_argv:"
504 @ MSG_CNOTE_T_PR_ASLWPID                "pr_aslwpid:"
505 @ MSG_CNOTE_T_PR_BIND                   "pr_bind:"
506 @ MSG_CNOTE_T_PR_BINDPRO                "pr_bindpro:"
507 @ MSG_CNOTE_T_PR_BINDPSET               "pr_bindpset:"
508 @ MSG_CNOTE_T_PR_BRKBASE                "pr_brkbase:"
509 @ MSG_CNOTE_T_PR_BRKSIZE                "pr_brksize:"
510 @ MSG_CNOTE_T_PR_BYRSSIZE               "pr_byrssize:"
511 @ MSG_CNOTE_T_PR_BYSIZE                 "pr_bysize:"
512 @ MSG_CNOTE_T_PR_CLNAME                 "pr_clname:"
513 @ MSG_CNOTE_T_PR_CONTRACT               "pr_contract:"
514 @ MSG_CNOTE_T_PR_CPU                    "pr_cpu:"
515 @ MSG_CNOTE_T_PR_CSTIME                 "pr_cstime:"
516 @ MSG_CNOTE_T_PR_CTIME                  "pr_ctime:"
517 @ MSG_CNOTE_T_PR_CURSIG                 "pr_cursig:"
518 @ MSG_CNOTE_T_PR_CUTIME                 "pr_cutime:"
519 @ MSG_CNOTE_T_PR_DMODEL                 "pr_dmodel:"
520 @ MSG_CNOTE_T_PR_EGID                   "pr_egid:"
521 @ MSG_CNOTE_T_PR_ENVP                   "pr_envp:"
522 @ MSG_CNOTE_T_PR_ERRNO                  "pr_errno:"
```

```
523 @ MSG_CNOTE_T_PR_ERRPRIV              "pr_errpriv:"
524 @ MSG_CNOTE_T_PR_EUID                 "pr_euid:"
525 @ MSG_CNOTE_T_PR_FLAG                 "pr_flag:"
526 @ MSG_CNOTE_T_PR_FLAGS                "pr_flags:"
527 @ MSG_CNOTE_T_PR_FLTTRACE             "pr_flttrace:"
528 @ MSG_CNOTE_T_PR_FNAME                "pr_fname:"
529 @ MSG_CNOTE_T_PR_FPREG                "pr_fpreg:"
530 @ MSG_CNOTE_T_PR_GID                  "pr_gid:"
531 @ MSG_CNOTE_T_PR_GROUPS               "pr_groups:"
532 @ MSG_CNOTE_T_PR_INFO                 "pr_info:"
533 @ MSG_CNOTE_T_PR_INFOSIZE             "pr_infosize:"
534 @ MSG_CNOTE_T_PR_INSTR                "pr_instr:"
535 @ MSG_CNOTE_T_PR_LGRP                 "pr_lgrp:"
536 @ MSG_CNOTE_T_PR_LTTYDEV              "pr_lttydev:"
537 @ MSG_CNOTE_T_PR_LWP                  "pr_lwp:"
538 @ MSG_CNOTE_T_PR_LWPHOLD              "pr_lwphold:"
539 @ MSG_CNOTE_T_PR_LWPID                "pr_lwpid:"
540 @ MSG_CNOTE_T_PR_LWPNAME              "pr_lwpname:"
541 @ MSG_CNOTE_T_PR_LWPPEND              "pr_lwppend:"
542 @ MSG_CNOTE_T_PR_NAME                 "pr_name:"
543 @ MSG_CNOTE_T_PR_NGROUPS              "pr_ngroups:"
544 @ MSG_CNOTE_T_PR_NICE                 "pr_nice:"
545 @ MSG_CNOTE_T_PR_NLWP                 "pr_nlwp:"
546 @ MSG_CNOTE_T_PR_NSETS                "pr_nsets:"
547 @ MSG_CNOTE_T_PR_NSYSARG              "pr_nsysarg:"
548 @ MSG_CNOTE_T_PR_NZOMB                "pr_nzomb:"
549 @ MSG_CNOTE_T_PR_OLDCONTEXT           "pr_oldcontext:"
550 @ MSG_CNOTE_T_PR_OLDPRI               "pr_oldpri:"
551 @ MSG_CNOTE_T_PR_ONPRO                "pr_onpro:"
552 @ MSG_CNOTE_T_PR_OTTYDEV              "pr_ottydev:"
553 @ MSG_CNOTE_T_PR_PCTCPU               "pr_pctcpu:"
554 @ MSG_CNOTE_T_PR_PCTMEM               "pr_pctmem:"
555 @ MSG_CNOTE_T_PR_PGID                 "pr_pgid:"
556 @ MSG_CNOTE_T_PR_PGRP                 "pr_pgrp:"
557 @ MSG_CNOTE_T_PR_PID                  "pr_pid:"
558 @ MSG_CNOTE_T_PR_POOLID               "pr_poolid:"
559 @ MSG_CNOTE_T_PR_PPID                 "pr_ppid:"
560 @ MSG_CNOTE_T_PR_PRI                  "pr_pri:"
561 @ MSG_CNOTE_T_PR_PROCESSOR            "pr_processor:"
562 @ MSG_CNOTE_T_PR_PROJID               "pr_projid:"
563 @ MSG_CNOTE_T_PR_PSARGS               "pr_psargs:"
564 @ MSG_CNOTE_T_PR_REG                  "pr_reg:"
565 @ MSG_CNOTE_T_PR_RGID                 "pr_rgid:"
566 @ MSG_CNOTE_T_PR_RSSIZE               "pr_rssize:"
567 @ MSG_CNOTE_T_PR_RUID                 "pr_ruid:"
568 @ MSG_CNOTE_T_PR_RVAL1                "pr_rval1:"
569 @ MSG_CNOTE_T_PR_RVAL2                "pr_rval2:"
570 @ MSG_CNOTE_T_PR_SETS                 "pr_sets:"
571 @ MSG_CNOTE_T_PR_SETSIZE              "pr_setsize:"
572 @ MSG_CNOTE_T_PR_SGID                 "pr_sgid:"
573 @ MSG_CNOTE_T_PR_SID                  "pr_sid:"
574 @ MSG_CNOTE_T_PR_SIGHOLD              "pr_sighold:"
575 @ MSG_CNOTE_T_PR_SIGPEND              "pr_sigpend:"
576 @ MSG_CNOTE_T_PR_SIGTRACE             "pr_sigtrace:"
577 @ MSG_CNOTE_T_PR_SIZE                 "pr_size:"
578 @ MSG_CNOTE_T_PR_SNAME                "pr_sname:"
579 @ MSG_CNOTE_T_PR_START                "pr_start:"
580 @ MSG_CNOTE_T_PR_STATE                "pr_state:"
581 @ MSG_CNOTE_T_PR_STIME                "pr_stime:"
582 @ MSG_CNOTE_T_PR_STKBASE              "pr_stkbase:"
583 @ MSG_CNOTE_T_PR_STKSIZE              "pr_stksize:"
584 @ MSG_CNOTE_T_PR_STYPE                "pr_stype:"
585 @ MSG_CNOTE_T_PR_SUID                 "pr_suid:"
586 @ MSG_CNOTE_T_PR_SYSARG               "pr_sysarg:"
587 @ MSG_CNOTE_T_PR_SYSCALL              "pr_syscall:"
588 @ MSG_CNOTE_T_PR_SYSENTRY             "pr_sysentry:"
```



```
589 @ MSG_CNOTE_T_PR_SYSEXIT              "pr_sysexit:"
590 @ MSG_CNOTE_T_PR_TASKID              "pr_taskid:"
591 @ MSG_CNOTE_T_PR_TIME                 "pr_time:"
592 @ MSG_CNOTE_T_PR_TSTAMP               "pr_tstamp:"
593 @ MSG_CNOTE_T_PR_TTYDEV               "pr_ttydev:"
594 @ MSG_CNOTE_T_PR_UID                  "pr_uid:"
595 @ MSG_CNOTE_T_PR_USTACK               "pr_ustack:"
596 @ MSG_CNOTE_T_PR_UTIME                "pr_utime:"
597 @ MSG_CNOTE_T_PR_WCHAN                "pr_wchan:"
598 @ MSG_CNOTE_T_PR_WHAT                 "pr_what:"
599 @ MSG_CNOTE_T_PR_WHO                  "pr_who:"
600 @ MSG_CNOTE_T_PR_WHY                  "pr_why:"
601 @ MSG_CNOTE_T_PR_WSTAT                "pr_wstat:"
602 @ MSG_CNOTE_T_PR_ZOMB                 "pr_zomb:"
603 @ MSG_CNOTE_T_PR_ZONEID               "pr_zoneid:"
604 @ MSG_CNOTE_T_PR_EFFECTIVE            "pr_effective:"
605 @ MSG_CNOTE_T_PR_INHERIT              "pr_inherit:"
606 @ MSG_CNOTE_T_PR_LOWER                "pr_lower:"
607 @ MSG_CNOTE_T_PR_UPPER                "pr_upper:"
608 @ MSG_CNOTE_T_PR_VERSION              "pr_version:"
609 @ MSG_CNOTE_T_SA_FLAGS                "sa_flags:"
610 @ MSG_CNOTE_T_SA_HANDLER              "sa_handler:"
611 @ MSG_CNOTE_T_SA_MASK                 "sa_mask:"
612 @ MSG_CNOTE_T_SA_SIGACTION            "sa_sigaction:"
613 @ MSG_CNOTE_T_SIVAL_INT               "sival_int:"
614 @ MSG_CNOTE_T_SIVAL_PTR               "sival_ptr:"
615 @ MSG_CNOTE_T_SI_ADDR                 "si_addr:"
616 @ MSG_CNOTE_T_SI_BAND                 "si_band:"
617 @ MSG_CNOTE_T_SI_CODE                 "si_code:"
618 @ MSG_CNOTE_T_SI_CTID                 "si_ctid:"
619 @ MSG_CNOTE_T_SI_ENTITY               "si_entity"
620 @ MSG_CNOTE_T_SI_ERRNO                "si_errno:"
621 @ MSG_CNOTE_T_SI_PID                  "si_pid:"
622 @ MSG_CNOTE_T_SI_SIGNO                "si_signo:"
623 @ MSG_CNOTE_T_SI_STATUS               "si_status:"
624 @ MSG_CNOTE_T_SI_UID                  "si_uid:"
625 @ MSG_CNOTE_T_SI_VALUE                "si_value:"
626 @ MSG_CNOTE_T_SI_ZONEID               "si_zoneid:"
627 @ MSG_CNOTE_T_SS_FLAGS                "ss_flags:"
628 @ MSG_CNOTE_T_SS_SIZE                 "ss_size:"
629 @ MSG_CNOTE_T_SS_SP                   "ss_sp:"
630 @ MSG_CNOTE_T_TV_NSEC                 "tv_nsec:"
631 @ MSG_CNOTE_T_TV_SEC                  "tv_sec:"
632 @ MSG_CNOTE_T_UTS_MACHINE             "machine:"
633 @ MSG_CNOTE_T_UTS_NODENAME            "nodename:"
634 @ MSG_CNOTE_T_UTS_RELEASE             "release:"
635 @ MSG_CNOTE_T_UTS_SYSNAME             "sysname:"
636 @ MSG_CNOTE_T_UTS_VERSION             "version:"
637 @ MSG_CNOTE_T_PR_FD                   "pr_fd:"
638 @ MSG_CNOTE_T_PR_MODE                 "pr_mode:"
639 @ MSG_CNOTE_T_PR_PATH                 "pr_path:"
640 @ MSG_CNOTE_T_PR_MAJOR                "pr_major:"
641 @ MSG_CNOTE_T_PR_MINOR                "pr_minor:"
642 @ MSG_CNOTE_T_PR_RMAJOR               "pr_rmajor:"
643 @ MSG_CNOTE_T_PR_RMINOR               "pr_rminor:"
644 @ MSG_CNOTE_T_PR_OFFSET               "pr_offset:"
645 @ MSG_CNOTE_T_PR_INO                  "pr_ino:"
646 @ MSG_CNOTE_T_PR_FILEFLAGS            "pr_fileflags:"
647 @ MSG_CNOTE_T_PR_FDFLAGS              "pr_fdflags:"


650 # Names of fake sections generated from program header data
651 @ MSG_PHDRNAM_CAP               ".SUNW_cap(phdr)"
652 @ MSG_PHDRNAM_CAPINFO           ".SUNW_capinfo(phdr)"
653 @ MSG_PHDRNAM_CAPCHAIN          ".SUNW_capchain(phdr)"
654 @ MSG_PHDRNAM_DYN               ".dynamic(phdr)"
```

```
 655 @ MSG_PHDRNAM_DYNSTR            ".dynstr(phdr)"
 656 @ MSG_PHDRNAM_DYNSYM            ".dynsym(phdr)"
 657 @ MSG_PHDRNAM_FINIARR           ".fini_array(phdr)"
 658 @ MSG_PHDRNAM_HASH              ".hash(phdr)"
 659 @ MSG_PHDRNAM_INITARR           ".init_array(phdr)"
 660 @ MSG_PHDRNAM_INTERP            ".interp(phdr)"
 661 @ MSG_PHDRNAM_LDYNSYM           ".SUNW_ldynsym(phdr)"
 662 @ MSG_PHDRNAM_MOVE              ".move(phdr)"
 663 @ MSG_PHDRNAM_NOTE              ".note(phdr)"
 664 @ MSG_PHDRNAM_PREINITARR        ".preinit_array(phdr)"
 665 @ MSG_PHDRNAM_REL               ".rel(phdr)"
 666 @ MSG_PHDRNAM_RELA              ".rela(phdr)"
 667 @ MSG_PHDRNAM_SYMINFO           ".syminfo(phdr)"
 668 @ MSG_PHDRNAM_SYMSORT           ".SUNW_symsort(phdr)"
 669 @ MSG_PHDRNAM_TLSSORT           ".SUNW_tlssort(phdr)"
 670 @ MSG_PHDRNAM_UNWIND            ".eh_frame_hdr(phdr)"
 671 @ MSG_PHDRNAM_VER               ".SUNW_version(phdr)"
```

```
**********************************************************
    1454 Mon Oct 15 13:25:01 2018
new/usr/src/cmd/sgs/elfdump/common/gen_layout_obj.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 /*
   2  * This file and its contents are supplied under the terms of the
   3  * Common Development and Distribution License ("CDDL"), version 1.0.
   4  * You may only use this file in accordance with the terms of version
   5  * 1.0 of the CDDL.
   6  *
   7  * A full copy of the text of the CDDL should have accompanied this
   8  * source.  A copy of the CDDL is also available via the Internet at
   9  * http://www.illumos.org/license/CDDL.
  10  */

  12 /*
  13  * Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
  14  * Copyright 2018 Joyent, Inc.
  15  */

  17 /*
  18  * A little program who's only purpose is to get all the
  19  * CTF type information we want into an object.
  20  */

  22 #include <sys/types.h>
  23 #include <sys/stat.h>
  24 #include <sys/sysmacros.h>
  25 #include <sys/corectl.h>
  26 #define _STRUCTURED_PROC        1
  27 #include <sys/procfs.h>
  28 #include <sys/auxv.h>
  29 #include <sys/old_procfs.h>
  30 #include <sys/utsname.h>
  31 #include <sys/secflags.h>

  33 /* prgregset_t is a define on intel */
  34 #ifdef prgregset_t
  35 typedef prgregset_t
  36 #undef prgregset_t
  37     prgregset_t;
  38 #endif

  40 /* instantiate the types for CTF */
  41 auxv_t auxv;
  42 prgregset_t prgregset;
  43 lwpstatus_t lwpstatus;
  44 pstatus_t pstatus;
  45 prstatus_t prstatus;
  46 psinfo_t psinfo;
  47 prpsinfo_t prpsinfo;
  48 lwpsinfo_t lwpsinfo;
  49 prcred_t prcred;
  50 prpriv_t prpriv;
  51 priv_impl_info_t priv_impl;
  52 fltset_t fltset;
  53 siginfo_t siginfo;
  54 sigset_t sigset;
  55 struct sigaction sigact;
  56 stack_t stack;
  57 sysset_t sysset;
  58 timestruc_t ts;
  59 struct utsname uts;
  60 prfdinfo_t ptfd;
```



```
  61 prsecflags_t psf;
  62 prlwpname_t psn;
```

```
**********************************************************
   25316 Mon Oct 15 13:25:02 2018
new/usr/src/cmd/sgs/elfdump/common/gen_struct_layout.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

/*
 * This program is used to generate the contents of the
 * struct_layout_XXX.c files that contain per-architecture
 * struct_layout_XXX.c files that contain per-archtecture
 * structure layout information.
 *
 * Although not part of elfdump, it is built by the makefile
 * along with it.
 * To use it:
 *
 *      1) Run it, capturing the output in a file.
 *      2) If this is a replacement for an existing file,
 *              diff the new and old copies to ensure only
 *              the changes you expected are present.
 *      3) Put the new file in the common directory under the name
 *              struct_layout_XXX.c, where XXX is the name of
 *              the architecture (i386, amd64, sparc, sparcv9, etc).
 *      2) Add any necessary header and copyright comments.
 *      3) If this is a new architecture:
 *              - Add an extern statement for struct_layout_XXX()
 *                      to struct_layout.h
 *              - Add a case for it to the function sl_struct_layout()
 *                      in struct_layout.c.
 */

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <err.h>
#include <sys/types.h>
```



```
#include <libctf.h>

/*
 * This extracts CTF information from a temporary object file.
 *
 * START and END bracket a struct layout definition. They issue
 * the typedef boilerplate, and the standard first element (sizeof)
 * which captures the overall size of the structure.
 *
 * SCALAR_FIELD is for scalar struct fields
 *
 * ARRAY_FIELD is for  array struct fields
 *
 * ARRAY_TYPE is for plain (non-struct) array types
 */
#define START(_name, _type) \
        do_start(#_name, #_type)
#define END (void) \
        do_end()
#define SCALAR_FIELD(_type, _field, _sign) \
        do_scalar_field(#_type, #_field, _sign, NULL)
#define SCALAR_FIELD4(_type, _field, _sign, _rtype) \
        do_scalar_field(#_type, #_field, _sign, _rtype)
#define ARRAY_FIELD(_type, _field, _sign) \
        do_array_field(#_type, #_field, _sign, NULL)
#define ARRAY_TYPE(_type, _sign) \
        do_array_type(#_type, "elt0", _sign)

static void do_start(char *_name, char *_type);
static void do_end(void);
static void do_start_name(char *name);
static void do_start_sizeof(char *_type, char *realtype);
static void do_scalar_field(char *_type, char *_field,
        int _sign, char *dotfield);
static void do_array_field(char *_type, char *_field,
        int _sign, char *dotfield);
static void do_array_type(char *_type, char *_field, int _sign);

static void get_ctf_file(char *fname);
static int get_field_info(char *tname, char *fname, char *dotname,
        int *offp, int *sizep);

static ctf_file_t *ctf;
static char *objfile;
static char *machname;

/* auxv_t, <sys/auxv.h> */
static void
gen_auxv(void)
{
        START(auxv, auxv_t);

        SCALAR_FIELD(auxv_t,    a_type, 1);
        SCALAR_FIELD(auxv_t,    a_un.a_val,     1);
        SCALAR_FIELD(auxv_t,    a_un.a_ptr,     0);
        SCALAR_FIELD(auxv_t,    a_un.a_fcn,     0);

        END;
}
______unchanged_portion_omitted_

static void
gen_prlwpname(void)
{
        START(prlwpname, prlwpname_t);
        SCALAR_FIELD(prlwpname_t, pr_lwpid, 0);
```

```
608         ARRAY_FIELD(prlwpname_t, pr_lwpname, 0);
609         END;
610 }

612 /*ARGSUSED*/
613 int
614 main(int argc, char *argv[])
615 {
616         const char *fmt = "\t&%s_layout,\n";

618         /* get obj file for input */
619         if (argc < 3) {
620                 (void) fprintf(stderr,
621                     "usage: %s {object_file} {MACH}\n", argv[0]);
622                 exit(1);
623         }

625         objfile = argv[1];
626         machname = argv[2];

628         get_ctf_file(objfile);

630         (void) printf("#include <struct_layout.h>\n");

632         gen_auxv();
633         gen_prgregset();
634         gen_lwpstatus();
635         gen_pstatus();
636         gen_prstatus();
637         gen_psinfo();
638         gen_prpsinfo();
639         gen_lwpsinfo();
640         gen_prcred();
641         gen_prpriv();
642         gen_priv_impl_info();
643         gen_fltset();
644         gen_siginfo();
645         gen_sigset();
646         gen_sigaction();
647         gen_stack();
648         gen_sysset();
649         gen_timestruc();
650         gen_utsname();
651         gen_prfdinfo();
652         gen_prsecflags();
653         gen_prlwpname();

655         /*
656          * Generate the full arch_layout description
657          */
658         (void) printf(
659             "\n\n\n\nstatic const sl_arch_layout_t layout_%s = {\n",
660             machname);
661         (void) printf(fmt, "auxv");
662         (void) printf(fmt, "fltset");
663         (void) printf(fmt, "lwpsinfo");
664         (void) printf(fmt, "lwpstatus");
665         (void) printf(fmt, "prcred");
666         (void) printf(fmt, "priv_impl_info");
667         (void) printf(fmt, "prpriv");
668         (void) printf(fmt, "psinfo");
669         (void) printf(fmt, "pstatus");
670         (void) printf(fmt, "prgregset");
671         (void) printf(fmt, "prpsinfo");
672         (void) printf(fmt, "prstatus");
673         (void) printf(fmt, "sigaction");
```



```
674         (void) printf(fmt, "siginfo");
675         (void) printf(fmt, "sigset");
676         (void) printf(fmt, "stack");
677         (void) printf(fmt, "sysset");
678         (void) printf(fmt, "timestruc");
679         (void) printf(fmt, "utsname");
680         (void) printf(fmt, "prfdinfo");
681         (void) printf(fmt, "prsecflags");
682         (void) printf(fmt, "prlwpname");
683         (void) printf("};\n");

685         /*
686          * A public function, to make the information available
687          */
688         (void) printf("\n\nconst sl_arch_layout_t *\n");
689         (void) printf("struct_layout_%s(void)\n", machname);
690         (void) printf("{\n\treturn (&layout_%s);\n}\n", machname);

692         return (0);
693 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   15860 Mon Oct 15 13:25:03 2018
new/usr/src/cmd/sgs/elfdump/common/struct_layout.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _STRUCT_LAYOUT_H
#define _STRUCT_LAYOUT_H

#include        <conv.h>
#include        <_machelf.h>

/*
 * Local include file for elfdump, used to define structure layout
 * definitions for various system structs.
 */

#ifdef  __cplusplus
extern "C" {
#endif


/*
 * Solaris defines system structs that elfdump needs to display
 * data from. We have a variety of hurdles to overcome in doing this:
 *
 *      - The size of system types can differ between ELFCLASS32 and
 *              ELFCLASS64.
 *      - Stucture layout can differ between architectures, so a given
 *              field can have a different struct offset than is native
 *              for the system running elfdump. Depending on the struct
 *              in question, the layout for one platform may be impossible
 *              to achieve on another.
 *      - The byte order of the core object can differ from that
 *              of the system running elfdump.
```



```
 *
 * The result is that in the fully general case, each architecture
 * can have a slightly different definition of these structures.
 * The usual approach of assigning a pointer of the desired structure
 * type and then accessing fields through that pointer cannot be used
 * here. That approach can only be used to access structures with the
 * native layout of the elfdump host. We want any instance of elfdump
 * to be able to examine a Solaris object for any supported architecture,
 * so we need a more flexible approach.
 *
 * The solution to this problem lies in the fact that the binary
 * layout of these public types cannot be changed, except in backward
 * compatible ways. They are written to core files or published in
 * other ways such that we can't make changes that would make it
 * impossible to analyze old files. This means that we can build
 * table of offsets and sizes for each field of each struct, on
 * a per-archecture basis. These tables can be used to access the
 * struct fields directly from the note desc data, and elfdump
 * on any host can read the data from any other host.
 *
 * When reading these tables, it can be very helpful to examine
 * the struct definition at the same time.
 */

/*
 * sl_field_t is used to describe a struct field
 */
typedef struct {
        ushort_t        slf_offset;     /* Offset from start of struct */
        ushort_t        slf_eltlen;     /* Size of datum, in bytes */
        ushort_t        slf_nelts;      /* 0 for scalar, # of els for array */
        uchar_t         slf_sign;       /* True (1) if signed quantity */
} sl_field_t;
```

*unchanged_portion_omitted_*

```
typedef struct {
        sl_field_t              sizeof_struct;
        sl_field_t              pr_lwpid;
        sl_field_t              pr_lwpname;
} sl_prlwpname_layout_t;

/*
 * This type collects all of the layout definitions for
 * a given architecture.
 */
typedef struct {
        const sl_auxv_layout_t          *auxv;          /* auxv_t */
        const sl_fltset_layout_t        *fltset;        /* fltset_t */
        const sl_lwpsinfo_layout_t      *lwpsinfo;      /* lwpsinfo_t */
        const sl_lwpstatus_layout_t     *lwpstatus;     /* lwpstatus_t */
        const sl_prcred_layout_t        *prcred;        /* prcred_t */
        const sl_priv_impl_info_layout_t *priv_impl_info; /* priv_impl_info_t */
        const sl_prpriv_layout_t        *prpriv;        /* prpriv_t */
        const sl_psinfo_layout_t        *psinfo;        /* psinfo_t */
        const sl_pstatus_layout_t       *pstatus;       /* pstatus_t */
        const sl_prgregset_layout_t     *prgregset;     /* prgregset_t */
        const sl_prpsinfo_layout_t      *prpsinfo;      /* prpsinfo_t */
        const sl_prstatus_layout_t      *prstatus;      /* prstatus_t */
        const sl_sigaction_layout_t     *sigaction;     /* struct sigaction */
        const sl_siginfo_layout_t       *siginfo;       /* siginfo_t */
        const sl_sigset_layout_t        *sigset;        /* sigset_t */
        const sl_stack_layout_t         *stack;         /* stack_t */
        const sl_sysset_layout_t        *sysset;        /* sysset_t */
        const sl_timestruc_layout_t     *timestruc;     /* timestruc_t */
        const sl_utsname_layout_t       *utsname;       /* struct utsname */
        const sl_prfdinfo_layout_t      *prfdinfo;      /* prdinfo_t */
```

```
 570         const sl_prsecflags_layout_t    *prsecflags;    /* prsecflags_t */
 571         const sl_prlwpname_layout_t     *prlwpname;     /* prlwpname_t */
 572 } sl_arch_layout_t;
```

*unchanged_portion_omitted_*

```
**********************************************************
   12776 Mon Oct 15 13:25:04 2018
new/usr/src/cmd/sgs/elfdump/common/struct_layout_amd64.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#include <struct_layout.h>


static const sl_auxv_layout_t auxv_layout = {
        { 0,    16,     0,      0 },            /* sizeof (auxv_t) */
        { 0,    4,      0,      1 },            /* a_type */
        { 8,    8,      0,      1 },            /* a_un.a_val */
        { 8,    8,      0,      0 },            /* a_un.a_ptr */
        { 8,    8,      0,      0 },            /* a_un.a_fcn */
};
```

_____*unchanged_portion_omitted_*

```
static const sl_prlwpname_layout_t prlwpname_layout = {
        { 0,    40,     0,      0 },            /* sizeof (prlwpname_t) */
        { 0,    8,      0,      0 },            /* pr_lwpid */
        { 8,    1,      32,     0 },            /* pr_lwpname[] */
};
```

```
static const sl_arch_layout_t layout_amd64 = {
        &auxv_layout,
        &fltset_layout,
        &lwpsinfo_layout,
        &lwpstatus_layout,
        &prcred_layout,
        &priv_impl_info_layout,
        &prpriv_layout,
```



```
        &psinfo_layout,
        &pstatus_layout,
        &prgregset_layout,
        &prpsinfo_layout,
        &prstatus_layout,
        &sigaction_layout,
        &siginfo_layout,
        &sigset_layout,
        &stack_layout,
        &sysset_layout,
        &timestruc_layout,
        &utsname_layout,
        &prfdinfo_layout,
        &prsecflags_layout,
        &prlwpname_layout,
};
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   12730 Mon Oct 15 13:25:06 2018
new/usr/src/cmd/sgs/elfdump/common/struct_layout_i386.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#include <struct_layout.h>


static const sl_auxv_layout_t auxv_layout = {
        { 0,    8,      0,      0 },            /* sizeof (auxv_t) */
        { 0,    4,      0,      1 },            /* a_type */
        { 4,    4,      0,      1 },            /* a_un.a_val */
        { 4,    4,      0,      0 },            /* a_un.a_ptr */
        { 4,    4,      0,      0 },            /* a_un.a_fcn */
};
______unchanged_portion_omitted_


static const sl_prlwpname_layout_t prlwpname_layout = {
        { 0,    40,     0,      0 },            /* sizeof (prlwpname_t) */
        { 0,    8,      0,      0 },            /* pr_lwpid */
        { 8,    1,      32,     0 },            /* pr_lwpname[] */
};



static const sl_arch_layout_t layout_i386 = {
        &auxv_layout,
        &fltset_layout,
        &lwpsinfo_layout,
        &lwpstatus_layout,
        &prcred_layout,
        &priv_impl_info_layout,
        &prpriv_layout,
```



```
        &psinfo_layout,
        &pstatus_layout,
        &prgregset_layout,
        &prpsinfo_layout,
        &prstatus_layout,
        &sigaction_layout,
        &siginfo_layout,
        &sigset_layout,
        &stack_layout,
        &sysset_layout,
        &timestruc_layout,
        &utsname_layout,
        &prfdinfo_layout,
        &prsecflags_layout,
        &prlwpname_layout,
};
______unchanged_portion_omitted_
```

```
**********************************************************
   12736 Mon Oct 15 13:25:09 2018
new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparc.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#include <struct_layout.h>


static const sl_auxv_layout_t auxv_layout = {
        { 0,    8,      0,      0 },            /* sizeof (auxv_t) */
        { 0,    4,      0,      1 },            /* a_type */
        { 4,    4,      0,      1 },            /* a_un.a_val */
        { 4,    4,      0,      0 },            /* a_un.a_ptr */
        { 4,    4,      0,      0 },            /* a_un.a_fcn */
};
```

*unchanged_portion_omitted_*

```
static const sl_prlwpname_layout_t prlwpname_layout = {
        { 0,    40,     0,      0 },            /* sizeof (prlwpname_t) */
        { 0,    8,      0,      0 },            /* pr_lwpid */
        { 8,    1,      32,     0 },            /* pr_lwpname[] */
};



static const sl_arch_layout_t layout_sparc = {
        &auxv_layout,
        &fltset_layout,
        &lwpsinfo_layout,
        &lwpstatus_layout,
        &prcred_layout,
        &priv_impl_info_layout,
        &prpriv_layout,
```



```
        &psinfo_layout,
        &pstatus_layout,
        &prgregset_layout,
        &prpsinfo_layout,
        &prstatus_layout,
        &sigaction_layout,
        &siginfo_layout,
        &sigset_layout,
        &stack_layout,
        &sysset_layout,
        &timestruc_layout,
        &utsname_layout,
        &prfdinfo_layout,
        &prsecflags_layout,
        &prlwpname_layout,
};
```

*unchanged_portion_omitted_*

```
**********************************************************
   12782 Mon Oct 15 13:25:11 2018
new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparcv9.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#include <struct_layout.h>


static const sl_auxv_layout_t auxv_layout = {
        { 0,    16,     0,      0 },            /* sizeof (auxv_t) */
        { 0,    4,      0,      1 },            /* a_type */
        { 8,    8,      0,      1 },            /* a_un.a_val */
        { 8,    8,      0,      0 },            /* a_un.a_ptr */
        { 8,    8,      0,      0 },            /* a_un.a_fcn */
};
```

_____*unchanged_portion_omitted_*

```
static const sl_prlwpname_layout_t prlwpname_layout = {
        { 0,    40,     0,      0 },            /* sizeof (prlwpname_t) */
        { 0,    8,      0,      0 },            /* pr_lwpid */
        { 8,    1,      32,     0 },            /* pr_lwpname[] */
};
```

```
static const sl_arch_layout_t layout_sparcv9 = {
        &auxv_layout,
        &fltset_layout,
        &lwpsinfo_layout,
        &lwpstatus_layout,
        &prcred_layout,
        &priv_impl_info_layout,
        &prpriv_layout,
```



```
        &psinfo_layout,
        &pstatus_layout,
        &prgregset_layout,
        &prpsinfo_layout,
        &prstatus_layout,
        &sigaction_layout,
        &siginfo_layout,
        &sigset_layout,
        &stack_layout,
        &sysset_layout,
        &timestruc_layout,
        &utsname_layout,
        &prfdinfo_layout,
        &prsecflags_layout,
        &prlwpname_layout,
};
```

_____*unchanged_portion_omitted_*

```
**********************************************************
    3939 Mon Oct 15 13:25:11 2018
new/usr/src/cmd/sgs/libconv/Makefile.com
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1994, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2018 Joyent, Inc.
# Copyright 2018, Joyent, Inc.
#

LIBRARY =       libconv.a

COMOBJS32 =     cap_machelf32.o         dynamic_machelf32.o \
                globals_machelf32.o     sections_machelf32.o \
                symbols_machelf32.o     symbols_sparc_machelf32.o

COMOBJS64 =     cap_machelf64.o         dynamic_machelf64.o \
                globals_machelf64.o     sections_machelf64.o \
                symbols_machelf64.o     symbols_sparc_machelf64.o

COMOBJS=        arch.o                  audit.o \
                c_literal.o \
                cap.o                   config.o \
                corenote.o              data.o \
                deftag.o                demangle.o \
                dl.o                    dwarf.o \
                dwarf_ehe.o             dynamic.o \
                elf.o                   entry.o \
                globals.o               group.o \
                lddstub.o               map.o \
                phdr.o                  relocate.o \
                relocate_i386.o         relocate_amd64.o \
                relocate_sparc.o        sections.o \
                segments.o              strproc.o \
                symbols.o               syminfo.o \
                tokens.o                time.o \
                version.o

ELFCAP_OBJS=    elfcap.o

ASOBJS=         vernote.o

BLTOBJS=        arch_msg.o              audit_msg.o \
```



```
                c_literal_msg.o \
                cap_msg.o               config_msg.o \
                corenote_msg.o          data_msg.o \
                deftag_msg.o            demangle_msg.o \
                dl_msg.o                dwarf_msg.o \
                dwarf_ehe_msg.o         dynamic_msg.o \
                elf_msg.o               entry_msg.o \
                globals_msg.o           group_msg.o \
                map_msg.o               lddstub_msg.o \
                phdr_msg.o              relocate_amd64_msg.o \
                relocate_i386_msg.o     relocate_sparc_msg.o \
                sections_msg.o          segments_msg.o \
                symbols_msg.o           symbols_sparc_msg.o \
                syminfo_msg.o           time_msg.o \
                version_msg.o


OBJECTS =       $(COMOBJS) $(COMOBJS32) $(COMOBJS64) $(ELFCAP_OBJS) \
                $(ASOBJS) $(BLTOBJS)

#
# This library is unusual since it's a static archive of PIC objects.
# Since static archives should never contain CTF data (regardless of
# whether the object code is position-independent), we disable CTF.
#
NOCTFOBJS =     $(OBJECTS)
CTFMERGE_LIB =  :

include $(SRC)/lib/Makefile.lib
include $(SRC)/cmd/sgs/Makefile.com

CERRWARN        += -_gcc=-Wno-type-limits
CERRWARN        += -_gcc=-Wno-switch

CTFCONVERT_O=

README_REVISION=../../packages/common/readme_revision
ONLDREADME=     ../../packages/common/SUNWonld-README

PICS=           $(OBJECTS:%=pics/%)

CPPFLAGS +=     -I$(SRCBASE)/lib/libc/inc -I$(ELFCAP) \
                -I$(SRC)/common/sgsrtcid

ARFLAGS=        cr

AS_CPPFLAGS=    -P -D_ASM $(CPPFLAGS)

BLTDATA=        $(BLTOBJS:%.o=%.c) $(BLTOBJS:%.o=%.h) report_bufsize.h

SRCS=           ../common/llib-lconv
LINTSRCS=       $(COMOBJS:%.o=../common/%.c) \
                    $(COMOBJS_NOMSG:%.o=../common/%.c) \
                    $(ELFCOM_OBJS:%.o=$(ELFCAP)/%.c) ../common/lintsup.c
LINTSRCS32 =    $(COMOBJS32:%32.o=../common/%.c)
LINTSRCS64 =    $(COMOBJS64:%64.o=../common/%.c)

# Since libconv uses dlopen(3C) to load libdemangle-sys.so (much like it did
# for the old Sun Studio libdemangle.so) in order to avoid messy bootstrapping
# problems, but it also needs the definitions from demangle-sys.h for
# SYSDEM_LANG_AUTO, lint will complain about sysdemangle() being defined but not
# used unless it is explicitly included during the lint pass
$(LINTOUT32)    := LDLIBS += -ldemangle-sys
$(LINTOUT64)    := LDLIBS += -ldemangle-sys

SGSMSGTARG=     $(BLTOBJS:%_msg.o=../common/%.msg)
```

```
127 LINTFLAGS +=    -u -erroff=E_NAME_DECL_NOT_USED_DEF2
128 LINTFLAGS64 +=  -u -erroff=E_NAME_DECL_NOT_USED_DEF2
127 LINTFLAGS +=    -u
128 LINTFLAGS64 +=  -u

130 CLEANFILES +=   $(BLTDATA) $(LINTOUTS) bld_vernote vernote.s
131 CLOBBERFILES += $(LINTLIBS)
```

```
**********************************************************
   88087 Mon Oct 15 13:25:11 2018
new/usr/src/cmd/sgs/libconv/common/corenote.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright (c) 2018 Joyent, Inc.
 */

/*
 * String conversion routines the system structs found in
 * Solaris core file note sections. These items are not
 * ELF constructs. However, elfdump contains code for decoding
 * them, and therefore requires formatting support.
 */
#include        <stdio.h>
#include        <procfs.h>
#include        <sys/corectl.h>
#include        <sys/secflags.h>
#include        <string.h>
#include        <_conv.h>
#include        <corenote_msg.h>

const char *
conv_cnote_type(Word type, Conv_fmt_flags_t fmt_flags,
    Conv_inv_buf_t *inv_buf)
{
        static const Msg        types[] = {
                MSG_NT_PRSTATUS,        MSG_NT_PRFPREG,
                MSG_NT_PRPSINFO,        MSG_NT_PRXREG,
                MSG_NT_PLATFORM,        MSG_NT_AUXV,
                MSG_NT_GWINDOWS,        MSG_NT_ASRS,
                MSG_NT_LDT,             MSG_NT_PSTATUS,
                0,                      0,
                MSG_NT_PSINFO,          MSG_NT_PRCRED,
                MSG_NT_UTSNAME,         MSG_NT_LWPSTATUS,
                MSG_NT_LWPSINFO,        MSG_NT_PRPRIV,
                MSG_NT_PRPRIVINFO,      MSG_NT_CONTENT,
                MSG_NT_ZONENAME,        MSG_NT_FDINFO,
```



```
                MSG_NT_SPYMASTER,       MSG_NT_SECFLAGS,
                MSG_NT_LWPNAME,
                MSG_NT_SPYMASTER,       MSG_NT_SECFLAGS
        };
#if NT_NUM != NT_LWPNAME
#if NT_NUM != NT_SECFLAGS
#error "NT_NUM has grown. Update core note types[]"
#endif
        static const conv_ds_msg_t ds_types = {
            CONV_DS_MSG_INIT(NT_PRSTATUS, types) };
        static const conv_ds_t  *ds[] = { CONV_DS_ADDR(ds_types), NULL };


        return (conv_map_ds(ELFOSABI_NONE, EM_NONE, type, ds, fmt_flags,
            inv_buf));
}
_____unchanged_portion_omitted_
```

```
**********************************************************
   39857 Mon Oct 15 13:25:11 2018
new/usr/src/cmd/sgs/libconv/common/corenote.msg
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
# Copyright (c) 2018 Joyent, Inc.
#

@ MSG_NT_PRSTATUS               "[ NT_PRSTATUS ]"
@ MSG_NT_PRFPREG                "[ NT_PRFPREG ]"
@ MSG_NT_PRPSINFO               "[ NT_PRPSINFO ]"
@ MSG_NT_PRXREG                 "[ NT_PRXREG ]"
@ MSG_NT_PLATFORM               "[ NT_PLATFORM ]"
@ MSG_NT_AUXV                   "[ NT_AUXV ]"
@ MSG_NT_GWINDOWS               "[ NT_GWINDOWS ]"
@ MSG_NT_ASRS                   "[ NT_ASRS ]"
@ MSG_NT_LDT                    "[ NT_LDT ]"
@ MSG_NT_PSTATUS                "[ NT_PSTATUS ]"
@ MSG_NT_PSINFO                 "[ NT_PSINFO ]"
@ MSG_NT_PRCRED                 "[ NT_PRCRED ]"
@ MSG_NT_UTSNAME                "[ NT_UTSNAME ]"
@ MSG_NT_LWPSTATUS              "[ NT_LWPSTATUS ]"
@ MSG_NT_LWPSINFO               "[ NT_LWPSINFO ]"
@ MSG_NT_PRPRIV                 "[ NT_PRPRIV ]"
@ MSG_NT_PRPRIVINFO             "[ NT_PRPRIVINFO ]"
@ MSG_NT_CONTENT                "[ NT_CONTENT ]"
@ MSG_NT_ZONENAME               "[ NT_ZONENAME ]"
@ MSG_NT_FDINFO                 "[ NT_FDINFO ]"
@ MSG_NT_SPYMASTER              "[ NT_SPYMASTER ]"
@ MSG_NT_SECFLAGS               "[ NT_SECFLAGS ]"
@ MSG_NT_LWPNAME                "[ NT_LWPNAME ]"


@ MSG_AUXV_AF_SUN_SETUGID               "AF_SUN_SETUGID"
@ MSG_AUXV_AF_SUN_HWCAPVERIFY           "AF_SUN_HWCAPVERIFY"
@ MSG_AUXV_AF_SUN_NOPLM                 "AF_SUN_NOPLM"


@ MSG_AUXV_AT_NULL                      "NULL"
```



```
@ MSG_AUXV_AT_IGNORE                    "IGNORE"
@ MSG_AUXV_AT_EXECFD                    "EXECFD"
@ MSG_AUXV_AT_PHDR                      "PHDR"
@ MSG_AUXV_AT_PHENT                     "PHENT"
@ MSG_AUXV_AT_PHNUM                     "PHNUM"
@ MSG_AUXV_AT_PAGESZ                    "PAGESZ"
@ MSG_AUXV_AT_BASE                      "BASE"
@ MSG_AUXV_AT_FLAGS                     "FLAGS"
@ MSG_AUXV_AT_ENTRY                     "ENTRY"
@ MSG_AUXV_AT_NOTELF                    "NOTELF"
@ MSG_AUXV_AT_UID                       "UID"
@ MSG_AUXV_AT_EUID                      "EUID"
@ MSG_AUXV_AT_GID                       "GID"
@ MSG_AUXV_AT_EGID                      "EGID"
@ MSG_AUXV_AT_PLATFORM                  "PLATFORM"
@ MSG_AUXV_AT_HWCAP                     "HWCAP"
@ MSG_AUXV_AT_CLKTCK                    "CLKTCK"
@ MSG_AUXV_AT_FPUCW                     "FPUCW"
@ MSG_AUXV_AT_DCACHEBSIZE               "DCACHEBSIZE"
@ MSG_AUXV_AT_ICACHEBSIZE               "ICACHEBSIZE"
@ MSG_AUXV_AT_UCACHEBSIZE               "UCACHEBSIZE"
@ MSG_AUXV_AT_IGNOREPPC                 "IGNOREPPC"
@ MSG_AUXV_AT_SUN_UID                   "SUN_UID"
@ MSG_AUXV_AT_SUN_RUID                  "SUN_RUID"
@ MSG_AUXV_AT_SUN_GID                   "SUN_GID"
@ MSG_AUXV_AT_SUN_RGID                  "SUN_RGID"
@ MSG_AUXV_AT_SUN_LDELF                 "SUN_LDELF"
@ MSG_AUXV_AT_SUN_LDSHDR                "SUN_LDSHDR"
@ MSG_AUXV_AT_SUN_LDNAME                "SUN_LDNAME"
@ MSG_AUXV_AT_SUN_LPAGESZ               "SUN_LPAGESZ"
@ MSG_AUXV_AT_SUN_PLATFORM              "SUN_PLATFORM"
@ MSG_AUXV_AT_SUN_HWCAP                 "SUN_HWCAP"
@ MSG_AUXV_AT_SUN_IFLUSH                "SUN_IFLUSH"
@ MSG_AUXV_AT_SUN_CPU                   "SUN_CPU"
@ MSG_AUXV_AT_SUN_EXECNAME              "SUN_EXECNAME"
@ MSG_AUXV_AT_SUN_MMU                   "SUN_MMU"
@ MSG_AUXV_AT_SUN_LDDATA                "SUN_LDDATA"
@ MSG_AUXV_AT_SUN_AUXFLAGS              "SUN_AUXFLAGS"
@ MSG_AUXV_AT_SUN_EMULATOR              "SUN_EMULATOR"
@ MSG_AUXV_AT_SUN_BRANDNAME             "SUN_BRANDNAME"
@ MSG_AUXV_AT_SUN_BRAND_AUX1            "SUN_BRAND_AUX1"
@ MSG_AUXV_AT_SUN_BRAND_AUX2            "SUN_BRAND_AUX2"
@ MSG_AUXV_AT_SUN_BRAND_AUX3            "SUN_BRAND_AUX3"
@ MSG_AUXV_AT_SUN_HWCAP2                "SUN_HWCAP2"
@ MSG_AUXV_AT_SUN_COMMPAGE              "SUN_COMMPAGE"
@ MSG_AUXV_AT_SUN_FPTYPE                "SUN_FPTYPE"
@ MSG_AUXV_AT_SUN_FPSIZE                "SUN_FPSIZE"


@ MSG_CC_CONTENT_STACK          "STACK"
@ MSG_CC_CONTENT_HEAP           "HEAP"
@ MSG_CC_CONTENT_SHFILE         "SHFILE"
@ MSG_CC_CONTENT_SHANON         "SHANON"
@ MSG_CC_CONTENT_TEXT           "TEXT"
@ MSG_CC_CONTENT_DATA           "DATA"
@ MSG_CC_CONTENT_RODATA         "RODATA"
@ MSG_CC_CONTENT_ANON           "ANON"
@ MSG_CC_CONTENT_SHM            "SHM"
@ MSG_CC_CONTENT_ISM            "ISM"
@ MSG_CC_CONTENT_DISM           "DISM"
@ MSG_CC_CONTENT_CTF            "CTF"
@ MSG_CC_CONTENT_SYMTAB         "SYMTAB"


@ MSG_ERRNO_EPERM               "[ EPERM ]"                     # 1
@ MSG_ERRNO_ENOENT              "[ ENOENT ]"            # 2
```

```
127 @ MSG_ERRNO_ESRCH               "[ ESRCH ]"                 # 3
128 @ MSG_ERRNO_EINTR               "[ EINTR ]"                 # 4
129 @ MSG_ERRNO_EIO                 "[ EIO ]"                   # 5
130 @ MSG_ERRNO_ENXIO               "[ ENXIO ]"                 # 6
131 @ MSG_ERRNO_E2BIG               "[ E2BIG ]"                 # 7
132 @ MSG_ERRNO_ENOEXEC             "[ ENOEXEC ]"           # 8
133 @ MSG_ERRNO_EBADF               "[ EBADF ]"                 # 9
134 @ MSG_ERRNO_ECHILD              "[ ECHILD ]"            # 10
135 @ MSG_ERRNO_EAGAIN              "[ EAGAIN ]"            # 11
136 @ MSG_ERRNO_ENOMEM              "[ ENOMEM ]"            # 12
137 @ MSG_ERRNO_EACCES              "[ EACCES ]"            # 13
138 @ MSG_ERRNO_EFAULT              "[ EFAULT ]"            # 14
139 @ MSG_ERRNO_ENOTBLK             "[ ENOTBLK ]"           # 15
140 @ MSG_ERRNO_EBUSY               "[ EBUSY ]"                 # 16
141 @ MSG_ERRNO_EEXIST              "[ EEXIST ]"            # 17
142 @ MSG_ERRNO_EXDEV               "[ EXDEV ]"                 # 18
143 @ MSG_ERRNO_ENODEV              "[ ENODEV ]"            # 19
144 @ MSG_ERRNO_ENOTDIR             "[ ENOTDIR ]"           # 20
145 @ MSG_ERRNO_EISDIR              "[ EISDIR ]"            # 21
146 @ MSG_ERRNO_EINVAL              "[ EINVAL ]"            # 22
147 @ MSG_ERRNO_ENFILE              "[ ENFILE ]"            # 23
148 @ MSG_ERRNO_EMFILE              "[ EMFILE ]"            # 24
149 @ MSG_ERRNO_ENOTTY              "[ ENOTTY ]"            # 25
150 @ MSG_ERRNO_ETXTBSY             "[ ETXTBSY ]"           # 26
151 @ MSG_ERRNO_EFBIG               "[ EFBIG ]"                 # 27
152 @ MSG_ERRNO_ENOSPC              "[ ENOSPC ]"            # 28
153 @ MSG_ERRNO_ESPIPE              "[ ESPIPE ]"            # 29
154 @ MSG_ERRNO_EROFS               "[ EROFS ]"                 # 30
155 @ MSG_ERRNO_EMLINK              "[ EMLINK ]"            # 31
156 @ MSG_ERRNO_EPIPE               "[ EPIPE ]"                 # 32
157 @ MSG_ERRNO_EDOM                "[ EDOM ]"                  # 33
158 @ MSG_ERRNO_ERANGE              "[ ERANGE ]"            # 34
159 @ MSG_ERRNO_ENOMSG             "[ ENOMSG ]"            # 35
160 @ MSG_ERRNO_EIDRM               "[ EIDRM ]"                 # 36
161 @ MSG_ERRNO_ECHRNG              "[ ECHRNG ]"            # 37
162 @ MSG_ERRNO_EL2NSYNC            "[ EL2NSYNC ]"          # 38
163 @ MSG_ERRNO_EL3HLT              "[ EL3HLT ]"            # 39
164 @ MSG_ERRNO_EL3RST              "[ EL3RST ]"            # 40
165 @ MSG_ERRNO_ELNRNG              "[ ELNRNG ]"            # 41
166 @ MSG_ERRNO_EUNATCH             "[ EUNATCH ]"           # 42
167 @ MSG_ERRNO_ENOCSI              "[ ENOCSI ]"            # 43
168 @ MSG_ERRNO_EL2HLT              "[ EL2HLT ]"            # 44
169 @ MSG_ERRNO_EDEADLK             "[ EDEADLK ]"           # 45
170 @ MSG_ERRNO_ENOLCK              "[ ENOLCK ]"            # 46
171 @ MSG_ERRNO_ECANCELED           "[ ECANCELED ]"         # 47
172 @ MSG_ERRNO_ENOTSUP             "[ ENOTSUP ]"           # 48
173 @ MSG_ERRNO_EDQUOT              "[ EDQUOT ]"            # 49
174 @ MSG_ERRNO_EBADE               "[ EBADE ]"                 # 50
175 @ MSG_ERRNO_EBADR               "[ EBADR ]"                 # 51
176 @ MSG_ERRNO_EXFULL              "[ EXFULL ]"            # 52
177 @ MSG_ERRNO_ENOANO              "[ ENOANO ]"            # 53
178 @ MSG_ERRNO_EBADRQC             "[ EBADRQC ]"           # 54
179 @ MSG_ERRNO_EBADSLT             "[ EBADSLT ]"           # 55
180 @ MSG_ERRNO_EDEADLOCK           "[ EDEADLOCK ]"         # 56
181 @ MSG_ERRNO_EBFONT              "[ EBFONT ]"            # 57
182 @ MSG_ERRNO_EOWNERDEAD          "[ EOWNERDEAD ]"            # 58
183 @ MSG_ERRNO_ENOTRECOVERABLE     "[ ENOTRECOVERABLE ]"   # 59
184 @ MSG_ERRNO_ENOSTR             "[ ENOSTR ]"            # 60
185 @ MSG_ERRNO_ENODATA            "[ ENODATA ]"           # 61
186 @ MSG_ERRNO_ETIME               "[ ETIME ]"                 # 62
187 @ MSG_ERRNO_ENOSR               "[ ENOSR ]"                 # 63
188 @ MSG_ERRNO_ENONET             "[ ENONET ]"            # 64
189 @ MSG_ERRNO_ENOPKG             "[ ENOPKG ]"            # 65
190 @ MSG_ERRNO_EREMOTE            "[ EREMOTE ]"           # 66
191 @ MSG_ERRNO_ENOLINK            "[ ENOLINK ]"           # 67
192 @ MSG_ERRNO_EADV               "[ EADV ]"                  # 68
```



```
193 @ MSG_ERRNO_ESRMNT             "[ ESRMNT ]"            # 69
194 @ MSG_ERRNO_ECOMM              "[ ECOMM ]"                 # 70
195 @ MSG_ERRNO_EPROTO             "[ EPROTO ]"            # 71
196 @ MSG_ERRNO_ELOCKUNMAPPED      "[ ELOCKUNMAPPED ]"         # 72
197 @ MSG_ERRNO_ENOTACTIVE         "[ ENOTACTIVE ]"            # 73
198 @ MSG_ERRNO_EMULTIHOP          "[ EMULTIHOP ]"         # 74
199 # errno 75 and 76 are not defined
200 @ MSG_ERRNO_EBADMSG            "[ EBADMSG ]"           # 77
201 @ MSG_ERRNO_ENAMETOOLONG       "[ ENAMETOOLONG ]"          # 78
202 @ MSG_ERRNO_EOVERFLOW          "[ EOVERFLOW ]"         # 79
203 @ MSG_ERRNO_ENOTUNIQ          "[ ENOTUNIQ ]"          # 80
204 @ MSG_ERRNO_EBADFD            "[ EBADFD ]"            # 81
205 @ MSG_ERRNO_EREMCHG           "[ EREMCHG ]"           # 82
206 @ MSG_ERRNO_ELIBACC           "[ ELIBACC ]"           # 83
207 @ MSG_ERRNO_ELIBBAD           "[ ELIBBAD ]"           # 84
208 @ MSG_ERRNO_ELIBSCN           "[ ELIBSCN ]"           # 85
209 @ MSG_ERRNO_ELIBMAX           "[ ELIBMAX ]"           # 86
210 @ MSG_ERRNO_ELIBEXEC          "[ ELIBEXEC ]"          # 87
211 @ MSG_ERRNO_EILSEQ            "[ EILSEQ ]"            # 88
212 @ MSG_ERRNO_ENOSYS            "[ ENOSYS ]"            # 89
213 @ MSG_ERRNO_ELOOP             "[ ELOOP ]"                 # 90
214 @ MSG_ERRNO_ERESTART          "[ ERESTART ]"          # 91
215 @ MSG_ERRNO_ESTRPIPE          "[ ESTRPIPE ]"          # 92
216 @ MSG_ERRNO_ENOTEMPTY         "[ ENOTEMPTY ]"         # 93
217 @ MSG_ERRNO_EUSERS            "[ EUSERS ]"            # 94
218 @ MSG_ERRNO_ENOTSOCK          "[ ENOTSOCK ]"          # 95
219 @ MSG_ERRNO_EDESTADDRREQ      "[ EDESTADDRREQ ]"          # 96
220 @ MSG_ERRNO_EMSGSIZE          "[ EMSGSIZE ]"          # 97
221 @ MSG_ERRNO_EPROTOTYPE        "[ EPROTOTYPE ]"            # 98
222 @ MSG_ERRNO_ENOPROTOOPT       "[ ENOPROTOOPT ]"           # 99
223 # Gap in ERRNO values 100-119
224 @ MSG_ERRNO_EPROTONOSUPPORT   "[ EPROTONOSUPPORT ]"   # 120
225 @ MSG_ERRNO_ESOCKTNOSUPPORT   "[ ESOCKTNOSUPPORT ]"   # 121
226 @ MSG_ERRNO_EOPNOTSUPP        "[ EOPNOTSUPP ]"            # 122
227 @ MSG_ERRNO_EPFNOSUPPORT      "[ EPFNOSUPPORT ]"          # 123
228 @ MSG_ERRNO_EAFNOSUPPORT      "[ EAFNOSUPPORT ]"          # 124
229 @ MSG_ERRNO_EADDRINUSE        "[ EADDRINUSE ]"            # 125
230 @ MSG_ERRNO_EADDRNOTAVAIL     "[ EADDRNOTAVAIL ]"         # 126
231 @ MSG_ERRNO_ENETDOWN          "[ ENETDOWN ]"          # 127
232 @ MSG_ERRNO_ENETUNREACH       "[ ENETUNREACH ]"           # 128
233 @ MSG_ERRNO_ENETRESET         "[ ENETRESET ]"         # 129
234 @ MSG_ERRNO_ECONNABORTED      "[ ECONNABORTED ]"          # 130
235 @ MSG_ERRNO_ECONNRESET        "[ ECONNRESET ]"            # 131
236 @ MSG_ERRNO_ENOBUFS          "[ ENOBUFS ]"           # 132
237 @ MSG_ERRNO_EISCONN           "[ EISCONN ]"           # 133
238 @ MSG_ERRNO_ENOTCONN          "[ ENOTCONN ]"          # 134
239 #XENIX has 135 - 142
240 @ MSG_ERRNO_ESHUTDOWN         "[ ESHUTDOWN ]"         # 143
241 @ MSG_ERRNO_ETOOMANYREFS      "[ ETOOMANYREFS ]"          # 144
242 @ MSG_ERRNO_ETIMEDOUT         "[ ETIMEDOUT ]"         # 145
243 @ MSG_ERRNO_ECONNREFUSED      "[ ECONNREFUSED ]"          # 146
244 @ MSG_ERRNO_EHOSTDOWN        "[ EHOSTDOWN ]"         # 147
245 @ MSG_ERRNO_EHOSTUNREACH     "[ EHOSTUNREACH ]"          # 148
246 @ MSG_ERRNO_EALREADY         "[ EALREADY ]"          # 149
247 @ MSG_ERRNO_EINPROGRESS      "[ EINPROGRESS ]"           # 150
248 @ MSG_ERRNO_ESTALE           "[ ESTALE ]"            # 151


251 @ MSG_PR_MODEL_UNKNOWN       "[ PR_MODEL_UNKNOWN ]"
252 @ MSG_PR_MODEL_ILP32         "[ PR_MODEL_ILP32 ]"
253 @ MSG_PR_MODEL_LP64          "[ PR_MODEL_LP64 ]"

255 @ MSG_PR_FLAGS_STOPPED       "PR_STOPPED"
256 @ MSG_PR_FLAGS_ISTOP         "PR_ISTOP"
257 @ MSG_PR_FLAGS_DSTOP         "PR_DSTOP"
258 @ MSG_PR_FLAGS_STEP          "PR_STEP"
```

```
 259 @ MSG_PR_FLAGS_ASLEEP           "PR_ASLEEP"
 260 @ MSG_PR_FLAGS_PCINVAL          "PR_PCINVAL"
 261 @ MSG_PR_FLAGS_ASLWP            "PR_ASLWP"
 262 @ MSG_PR_FLAGS_AGENT            "PR_AGENT"
 263 @ MSG_PR_FLAGS_DETACH           "PR_DETACH"
 264 @ MSG_PR_FLAGS_DAEMON           "PR_DAEMON"
 265 @ MSG_PR_FLAGS_IDLE             "PR_IDLE"
 266 @ MSG_PR_FLAGS_ISSYS            "PR_ISSYS"
 267 @ MSG_PR_FLAGS_VFORKP           "PR_VFORKP"
 268 @ MSG_PR_FLAGS_ORPHAN           "PR_ORPHAN"
 269 @ MSG_PR_FLAGS_NOSIGCHLD        "PR_NOSIGCHLD"
 270 @ MSG_PR_FLAGS_WAITPID          "PR_WAITPID"
 271 @ MSG_PR_FLAGS_FORK             "PR_FORK"
 272 @ MSG_PR_FLAGS_RLC              "PR_RLC"
 273 @ MSG_PR_FLAGS_KLC              "PR_KLC"
 274 @ MSG_PR_FLAGS_ASYNC            "PR_ASYNC"
 275 @ MSG_PR_FLAGS_MSACCT           "PR_MSACCT"
 276 @ MSG_PR_FLAGS_BPTADJ           "PR_BPTADJ"
 277 @ MSG_PR_FLAGS_PTRACE           "PR_PTRACE"
 278 @ MSG_PR_FLAGS_MSFORK           "PR_MSFORK"
 279 @ MSG_PR_FLAGS_PCOMPAT          "PR_PCOMPAT"


 282 @ MSG_PROC_FLAG_SSYS           "SSYS"
 283 @ MSG_PROC_FLAG_SMSACCT        "SMSACCT"

 285 @ MSG_ASLR              "ASLR"
 286 @ MSG_FORBIDNULLMAP     "FORBIDNULLMAP"
 287 @ MSG_NOEXECSTACK       "NOEXECSTACK"

 289 @ MSG_PS_NONE                   "[ PS_NONE ]"
 290 @ MSG_PS_QUERY                  "[ PS_QUERY ]"
 291 @ MSG_PS_MYID                   "[ PS_MYID ]"
 292 @ MSG_PS_SOFT                   "[ PS_SOFT ]"
 293 @ MSG_PS_HARD                   "[ PS_HARD ]"
 294 @ MSG_PS_QUERY_TYPE             "[ PS_QUERY_TYPE ]"


 297 @ MSG_REG_SPARC_G0              "[ r0/g0 ]"
 298 @ MSG_REG_SPARC_G1              "[ r1/g1 ]"
 299 @ MSG_REG_SPARC_G2              "[ r2/g2 ]"
 300 @ MSG_REG_SPARC_G3              "[ r3/g3 ]"
 301 @ MSG_REG_SPARC_G4              "[ r4/g4 ]"
 302 @ MSG_REG_SPARC_G5              "[ r5/g5 ]"
 303 @ MSG_REG_SPARC_G6              "[ r6/g6 ]"
 304 @ MSG_REG_SPARC_G7              "[ r7/g7 ]"
 305 @ MSG_REG_SPARC_O0              "[ r8/o0 ]"
 306 @ MSG_REG_SPARC_O1              "[ r9/o1 ]"
 307 @ MSG_REG_SPARC_O2              "[ r10/o2 ]"
 308 @ MSG_REG_SPARC_O3              "[ r11/o3 ]"
 309 @ MSG_REG_SPARC_O4              "[ r12/o4 ]"
 310 @ MSG_REG_SPARC_O5              "[ r13/o5 ]"
 311 @ MSG_REG_SPARC_O6              "[ r14/o6/sp ]"
 312 @ MSG_REG_SPARC_O7              "[ r15/o7 ]"
 313 @ MSG_REG_SPARC_L0              "[ r16/l0 ]"
 314 @ MSG_REG_SPARC_L1              "[ r17/l1 ]"
 315 @ MSG_REG_SPARC_L2              "[ r18/l2 ]"
 316 @ MSG_REG_SPARC_L3              "[ r19/l3 ]"
 317 @ MSG_REG_SPARC_L4              "[ r20/l4 ]"
 318 @ MSG_REG_SPARC_L5              "[ r21/l5 ]"
 319 @ MSG_REG_SPARC_L6              "[ r22/l6 ]"
 320 @ MSG_REG_SPARC_L7              "[ r23/l7 ]"
 321 @ MSG_REG_SPARC_I0              "[ r24/i0 ]"
 322 @ MSG_REG_SPARC_I1              "[ r25/i1 ]"
 323 @ MSG_REG_SPARC_I2              "[ r26/i2 ]"
 324 @ MSG_REG_SPARC_I3              "[ r27/i3 ]"
```



```
 325 @ MSG_REG_SPARC_I4              "[ r28/i4 ]"
 326 @ MSG_REG_SPARC_I5              "[ r29/i5 ]"
 327 @ MSG_REG_SPARC_I6              "[ r30/i6/fp ]"
 328 @ MSG_REG_SPARC_I7              "[ r31/i7 ]"
 329 @ MSG_REG_SPARC_CCR             "[ ccr ]"
 330 @ MSG_REG_SPARC_PSR             "[ psr ]"
 331 @ MSG_REG_SPARC_PC              "[ pc ]"
 332 @ MSG_REG_SPARC_nPC             "[ npc ]"
 333 @ MSG_REG_SPARC_Y               "[ y ]"
 334 @ MSG_REG_SPARC_ASI             "[ asi ]"
 335 @ MSG_REG_SPARC_FPRS            "[ fprs ]"
 336 @ MSG_REG_SPARC_WIM             "[ wim ]"
 337 @ MSG_REG_SPARC_TBR             "[ tbr ]"

 339 @ MSG_REG_AMD64_R15             "[ r15 ]"
 340 @ MSG_REG_AMD64_R14             "[ r14 ]"
 341 @ MSG_REG_AMD64_R13             "[ r13 ]"
 342 @ MSG_REG_AMD64_R12             "[ r12 ]"
 343 @ MSG_REG_AMD64_R11             "[ r11 ]"
 344 @ MSG_REG_AMD64_R10             "[ r10 ]"
 345 @ MSG_REG_AMD64_R9              "[ r9 ]"
 346 @ MSG_REG_AMD64_R8              "[ r8 ]"
 347 @ MSG_REG_AMD64_RDI             "[ rdi ]"
 348 @ MSG_REG_AMD64_RSI             "[ rsi ]"
 349 @ MSG_REG_AMD64_RBP             "[ rbp ]"
 350 @ MSG_REG_AMD64_RBX             "[ rbx ]"
 351 @ MSG_REG_AMD64_RDX             "[ rdx ]"
 352 @ MSG_REG_AMD64_RCX             "[ rcx ]"
 353 @ MSG_REG_AMD64_RAX             "[ rax ]"
 354 @ MSG_REG_AMD64_TRAPNO          "[ trapno ]"
 355 @ MSG_REG_AMD64_ERR             "[ err ]"
 356 @ MSG_REG_AMD64_RIP             "[ rip ]"
 357 @ MSG_REG_AMD64_CS              "[ cs ]"
 358 @ MSG_REG_AMD64_RFL             "[ rfl ]"
 359 @ MSG_REG_AMD64_RSP             "[ rsp ]"
 360 @ MSG_REG_AMD64_SS              "[ ss ]"
 361 @ MSG_REG_AMD64_FS              "[ fs ]"
 362 @ MSG_REG_AMD64_GS              "[ gs ]"
 363 @ MSG_REG_AMD64_ES              "[ es ]"
 364 @ MSG_REG_AMD64_DS              "[ ds ]"
 365 @ MSG_REG_AMD64_FSBASE          "[ fsbase ]"
 366 @ MSG_REG_AMD64_GSBASE          "[ gsbase ]"

 368 @ MSG_REG_I86_GS                "[ gs ]"
 369 @ MSG_REG_I86_FS                "[ fs ]"
 370 @ MSG_REG_I86_ES                "[ es ]"
 371 @ MSG_REG_I86_DS                "[ ds ]"
 372 @ MSG_REG_I86_EDI               "[ edi ]"
 373 @ MSG_REG_I86_ESI               "[ esi ]"
 374 @ MSG_REG_I86_EBP               "[ ebp ]"
 375 @ MSG_REG_I86_ESP               "[ esp ]"
 376 @ MSG_REG_I86_EBX               "[ ebx ]"
 377 @ MSG_REG_I86_EDX               "[ edx ]"
 378 @ MSG_REG_I86_ECX               "[ ecx ]"
 379 @ MSG_REG_I86_EAX               "[ eax ]"
 380 @ MSG_REG_I86_TRAPNO            "[ trapno ]"
 381 @ MSG_REG_I86_ERR               "[ err ]"
 382 @ MSG_REG_I86_EIP               "[ eip ]"
 383 @ MSG_REG_I86_CS                "[ cs ]"
 384 @ MSG_REG_I86_EFL               "[ efl ]"
 385 @ MSG_REG_I86_UESP              "[ uesp ]"
 386 @ MSG_REG_I86_SS                "[ ss ]"

 388 @ MSG_PR_WHY_REQUESTED          "[ PR_REQUESTED ]"
 389 @ MSG_PR_WHY_SIGNALLED          "[ PR_SIGNALLED ]"
 390 @ MSG_PR_WHY_SYSENTRY           "[ PR_SYSENTRY ]"
```

```
391 @ MSG_PR_WHY_SYSEXIT            "[ PR_SYSEXIT ]"
392 @ MSG_PR_WHY_JOBCONTROL         "[ PR_JOBCONTROL ]"
393 @ MSG_PR_WHY_FAULTED            "[ PR_FAULTED ]"
394 @ MSG_PR_WHY_SUSPENDED          "[ PR_SUSPENDED ]"
395 @ MSG_PR_WHY_CHECKPOINT         "[ PR_CHECKPOINT ]"

397 @ MSG_PRIV_ALL                  "[ PRIV_ALL ]"
398 @ MSG_PRIV_MULTIPLE             "[ PRIV_MULTIPLE ]"
399 @ MSG_PRIV_NONE                 "[ PRIV_NONE ]"
400 @ MSG_PRIV_ALLZONE              "[ PRIV_ALLZONE ]"
401 @ MSG_PRIV_GLOBAL               "[ PRIV_ALLGLOBAL ]"


404 @ MSG_SA_ONSTACK                "SA_ONSTACK"
405 @ MSG_SA_RESETHAND              "SA_RESETHAND"
406 @ MSG_SA_RESTART                "SA_RESTART"
407 @ MSG_SA_SIGINFO               "SA_SIGINFO"
408 @ MSG_SA_NODEFER               "SA_NODEFER"
409 @ MSG_SA_NOCLDWAIT             "SA_NOCLDWAIT"
410 @ MSG_SA_NOCLDSTOP             "SA_NOCLDSTOP"


413 @ MSG_SOBJ_NONE                "[ SOBJ_NONE ]"
414 @ MSG_SOBJ_MUTEX               "[ SOBJ_MUTEX ]"
415 @ MSG_SOBJ_RWLOCK              "[ SOBJ_RWLOCK ]"
416 @ MSG_SOBJ_CV                  "[ SOBJ_CV ]"
417 @ MSG_SOBJ_SEMA                "[ SOBJ_SEMA ]"
418 @ MSG_SOBJ_USER                "[ SOBJ_USER ]"
419 @ MSG_SOBJ_USER_PI             "[ SOBJ_USER_PI ]"
420 @ MSG_SOBJ_SHUTTLE             "[ SOBJ_SHUTTLE ]"


423 @ MSG_SS_ONSTACK                "SS_ONSTACK"
424 @ MSG_SS_DISABLE               "SS_DISABLE"


427 @ MSG_SI_NOINFO                "[ SI_NOINFO ]"
428 @ MSG_SI_DTRACE                "[ SI_DTRACE ]"
429 @ MSG_SI_RCTL                  "[ SI_RCTL ]"
430 @ MSG_SI_USER                  "[ SI_USER ]"
431 @ MSG_SI_LWP                   "[ SI_LWP ]"
432 @ MSG_SI_QUEUE                 "[ SI_QUEUE ]"
433 @ MSG_SI_TIMER                 "[ SI_TIMER ]"
434 @ MSG_SI_ASYNCIO               "[ SI_ASYNCIO ]"
435 @ MSG_SI_MESGQ                 "[ SI_MESGQ ]"

437 @ MSG_SI_ILL_ILLOPC            "[ ILL_ILLOPC ]"
438 @ MSG_SI_ILL_ILLOPN            "[ ILL_ILLOPN ]"
439 @ MSG_SI_ILL_ILLADR            "[ ILL_ILLADR ]"
440 @ MSG_SI_ILL_ILLTRP            "[ ILL_ILLTRP ]"
441 @ MSG_SI_ILL_PRVOPC            "[ ILL_PRVOPC ]"
442 @ MSG_SI_ILL_PRVREG            "[ ILL_PRVREG ]"
443 @ MSG_SI_ILL_COPROC            "[ ILL_COPROC ]"
444 @ MSG_SI_ILL_BADSTK            "[ ILL_BADSTK ]"

446 @ MSG_SI_EMT_TAGOVF            "[ EMT_TAGOVF ]"
447 @ MSG_SI_EMT_CPCOVF            "[ EMT_CPCOVF ]"

449 @ MSG_SI_FPE_INTDIV            "[ FPE_INTDIV ]"
450 @ MSG_SI_FPE_INTOVF            "[ FPE_INTOVF ]"
451 @ MSG_SI_FPE_FLTDIV            "[ FPE_FLTDIV ]"
452 @ MSG_SI_FPE_FLTOVF            "[ FPE_FLTOVF ]"
453 @ MSG_SI_FPE_FLTUND            "[ FPE_FLTUND ]"
454 @ MSG_SI_FPE_FLTRES            "[ FPE_FLTRES ]"
455 @ MSG_SI_FPE_FLTINV            "[ FPE_FLTINV ]"
456 @ MSG_SI_FPE_FLTSUB            "[ FPE_FLTSUB ]"
```



```
457 @ MSG_SI_FPE_FLTDEN            "[ FPE_FLTDEN ]"

459 @ MSG_SI_SEGV_MAPERR           "[ SEGV_MAPERR ]"
460 @ MSG_SI_SEGV_ACCERR           "[ SEGV_ACCERR ]"

462 @ MSG_SI_BUS_ADRALN            "[ BUS_ADRALN ]"
463 @ MSG_SI_BUS_ADRERR            "[ BUS_ADRERR ]"
464 @ MSG_SI_BUS_OBJERR            "[ BUS_OBJERR ]"

466 @ MSG_SI_TRAP_BRKPT            "[ TRAP_BRKPT ]"
467 @ MSG_SI_TRAP_TRACE            "[ TRAP_TRACE ]"
468 @ MSG_SI_TRAP_RWATCH           "[ TRAP_RWATCH ]"
469 @ MSG_SI_TRAP_WWATCH           "[ TRAP_WWATCH ]"
470 @ MSG_SI_TRAP_XWATCH           "[ TRAP_XWATCH ]"
471 @ MSG_SI_TRAP_DTRACE           "[ TRAP_DTRACE ]"

473 @ MSG_SI_CLD_EXITED            "[ CLD_EXITED ]"
474 @ MSG_SI_CLD_KILLED            "[ CLD_KILLED ]"
475 @ MSG_SI_CLD_DUMPED            "[ CLD_DUMPED ]"
476 @ MSG_SI_CLD_TRAPPED           "[ CLD_TRAPPED ]"
477 @ MSG_SI_CLD_STOPPED           "[ CLD_STOPPED ]"
478 @ MSG_SI_CLD_CONTINUED         "[ CLD_CONTINUED ]"

480 @ MSG_SI_POLL_IN               "[ POLL_IN ]"
481 @ MSG_SI_POLL_OUT              "[ POLL_OUT ]"
482 @ MSG_SI_POLL_MSG              "[ POLL_MSG ]"
483 @ MSG_SI_POLL_ERR              "[ POLL_ERR ]"
484 @ MSG_SI_POLL_PRI              "[ POLL_PRI ]"
485 @ MSG_SI_POLL_HUP              "[ POLL_HUP ]"

487 @ MSG_SIGHUP                   "[ SIGHUP ]"
488 @ MSG_SIGHUP_ALT               "SIGHUP"
489 @ MSG_SIGINT                   "[ SIGINT ]"
490 @ MSG_SIGINT_ALT               "SIGINT"
491 @ MSG_SIGQUIT                  "[ SIGQUIT ]"
492 @ MSG_SIGQUIT_ALT              "SIGQUIT"
493 @ MSG_SIGILL                   "[ SIGILL ]"
494 @ MSG_SIGILL_ALT               "SIGILL"
495 @ MSG_SIGTRAP                  "[ SIGTRAP ]"
496 @ MSG_SIGTRAP_ALT              "SIGTRAP"
497 @ MSG_SIGABRT                  "[ SIGABRT ]"
498 @ MSG_SIGABRT_ALT              "SIGABRT"
499 @ MSG_SIGEMT                   "[ SIGEMT ]"
500 @ MSG_SIGEMT_ALT               "SIGEMT"
501 @ MSG_SIGFPE                   "[ SIGFPE ]"
502 @ MSG_SIGFPE_ALT               "SIGFPE"
503 @ MSG_SIGKILL                  "[ SIGKILL ]"
504 @ MSG_SIGKILL_ALT              "SIGKILL"
505 @ MSG_SIGBUS                   "[ SIGBUS ]"
506 @ MSG_SIGBUS_ALT               "SIGBUS"
507 @ MSG_SIGSEGV                  "[ SIGSEGV ]"
508 @ MSG_SIGSEGV_ALT              "SIGSEGV"
509 @ MSG_SIGSYS                   "[ SIGSYS ]"
510 @ MSG_SIGSYS_ALT               "SIGSYS"
511 @ MSG_SIGPIPE                  "[ SIGPIPE ]"
512 @ MSG_SIGPIPE_ALT              "SIGPIPE"
513 @ MSG_SIGALRM                  "[ SIGALRM ]"
514 @ MSG_SIGALRM_ALT              "SIGALRM"
515 @ MSG_SIGTERM                  "[ SIGTERM ]"
516 @ MSG_SIGTERM_ALT              "SIGTERM"
517 @ MSG_SIGUSR1                  "[ SIGUSR1 ]"
518 @ MSG_SIGUSR1_ALT              "SIGUSR1"
519 @ MSG_SIGUSR2                  "[ SIGUSR2 ]"
520 @ MSG_SIGUSR2_ALT              "SIGUSR2"
521 @ MSG_SIGCHLD                  "[ SIGCHLD ]"
522 @ MSG_SIGCHLD_ALT              "SIGCHLD"
```

```
523 @ MSG_SIGPWR                    "[ SIGPWR ]"
524 @ MSG_SIGPWR_ALT                "SIGPWR"
525 @ MSG_SIGWINCH                  "[ SIGWINCH ]"
526 @ MSG_SIGWINCH_ALT              "SIGWINCH"
527 @ MSG_SIGURG                    "[ SIGURG ]"
528 @ MSG_SIGURG_ALT                "SIGURG"
529 @ MSG_SIGPOLL                   "[ SIGPOLL ]"
530 @ MSG_SIGPOLL_ALT               "SIGPOLL"
531 @ MSG_SIGSTOP                   "[ SIGSTOP ]"
532 @ MSG_SIGSTOP_ALT               "SIGSTOP"
533 @ MSG_SIGTSTP                   "[ SIGTSTP ]"
534 @ MSG_SIGTSTP_ALT               "SIGTSTP"
535 @ MSG_SIGCONT                   "[ SIGCONT ]"
536 @ MSG_SIGCONT_ALT               "SIGCONT"
537 @ MSG_SIGTTIN                   "[ SIGTTIN ]"
538 @ MSG_SIGTTIN_ALT               "SIGTTIN"
539 @ MSG_SIGTTOU                   "[ SIGTTOU ]"
540 @ MSG_SIGTTOU_ALT               "SIGTTOU"
541 @ MSG_SIGVTALRM                 "[ SIGVTALRM ]"
542 @ MSG_SIGVTALRM_ALT             "SIGVTALRM"
543 @ MSG_SIGPROF                   "[ SIGPROF ]"
544 @ MSG_SIGPROF_ALT               "SIGPROF"
545 @ MSG_SIGXCPU                   "[ SIGXCPU ]"
546 @ MSG_SIGXCPU_ALT               "SIGXCPU"
547 @ MSG_SIGXFSZ                   "[ SIGXFSZ ]"
548 @ MSG_SIGXFSZ_ALT               "SIGXFSZ"
549 @ MSG_SIGWAITING                "[ SIGWAITING ]"
550 @ MSG_SIGWAITING_ALT            "SIGWAITING"
551 @ MSG_SIGLWP                    "[ SIGLWP ]"
552 @ MSG_SIGLWP_ALT                "SIGLWP"
553 @ MSG_SIGFREEZE                 "[ SIGFREEZE ]"
554 @ MSG_SIGFREEZE_ALT             "SIGFREEZE"
555 @ MSG_SIGTHAW                   "[ SIGTHAW ]"
556 @ MSG_SIGTHAW_ALT               "SIGTHAW"
557 @ MSG_SIGCANCEL                 "[ SIGCANCEL ]"
558 @ MSG_SIGCANCEL_ALT             "SIGCANCEL"
559 @ MSG_SIGLOST                   "[ SIGLOST ]"
560 @ MSG_SIGLOST_ALT               "SIGLOST"
561 @ MSG_SIGXRES                   "[ SIGXRES ]"
562 @ MSG_SIGXRES_ALT               "SIGXRES"
563 @ MSG_SIGJVM1                   "[ SIGJVM1 ]"
564 @ MSG_SIGJVM1_ALT               "SIGJVM1"
565 @ MSG_SIGJVM2                   "[ SIGJVM2 ]"
566 @ MSG_SIGJVM2_ALT               "SIGJVM2"

568 @ MSG_FLTILL                    "[ FLTILL ]"
569 @ MSG_FLTILL_ALT                "FLTILL"
570 @ MSG_FLTPRIV                   "[ FLTPRIV ]"
571 @ MSG_FLTPRIV_ALT               "FLTPRIV"
572 @ MSG_FLTBPT                    "[ FLTBPT ]"
573 @ MSG_FLTBPT_ALT                "FLTBPT"
574 @ MSG_FLTTRACE                  "[ FLTTRACE ]"
575 @ MSG_FLTTRACE_ALT              "FLTTRACE"
576 @ MSG_FLTACCESS                 "[ FLTACCESS ]"
577 @ MSG_FLTACCESS_ALT             "FLTACCESS"
578 @ MSG_FLTBOUNDS                 "[ FLTBOUNDS ]"
579 @ MSG_FLTBOUNDS_ALT             "FLTBOUNDS"
580 @ MSG_FLTIOVF                   "[ FLTIOVF ]"
581 @ MSG_FLTIOVF_ALT               "FLTIOVF"
582 @ MSG_FLTIZDIV                  "[ FLTIZDIV ]"
583 @ MSG_FLTIZDIV_ALT              "FLTIZDIV"
584 @ MSG_FLTFPE                    "[ FLTFPE ]"
585 @ MSG_FLTFPE_ALT                "FLTFPE"
586 @ MSG_FLTSTACK                  "[ FLTSTACK ]"
587 @ MSG_FLTSTACK_ALT              "FLTSTACK"
588 @ MSG_FLTPAGE                   "[ FLTPAGE ]"
```



```
589 @ MSG_FLTPAGE_ALT               "FLTPAGE"
590 @ MSG_FLTWATCH                  "[ FLTWATCH ]"
591 @ MSG_FLTWATCH_ALT              "FLTWATCH"
592 @ MSG_FLTCPCOVF                 "[ FLTCPCOVF ]"
593 @ MSG_FLTCPCOVF_ALT             "FLTCPCOVF"

595 @ MSG_SYS_EXIT                          "[ exit ]"                  # 1
596 @ MSG_SYS_EXIT_ALT                      "exit"
597 @ MSG_SYS_2                             "2"                         # 2 (u)
598 @ MSG_SYS_READ                          "[ read ]"                  # 3
599 @ MSG_SYS_READ_ALT                      "read"
600 @ MSG_SYS_WRITE                         "[ write ]"                 # 4
601 @ MSG_SYS_WRITE_ALT                     "write"
602 @ MSG_SYS_OPEN                          "[ open ]"                  # 5
603 @ MSG_SYS_OPEN_ALT                      "open"
604 @ MSG_SYS_CLOSE                         "[ close ]"                 # 6
605 @ MSG_SYS_CLOSE_ALT                     "close"
606 @ MSG_SYS_7                             "7"                         # 7 (u)
607 @ MSG_SYS_8                             "8"                         # 8 (u)
608 @ MSG_SYS_LINK                          "[ link ]"                  # 9
609 @ MSG_SYS_LINK_ALT                      "link"
610 @ MSG_SYS_UNLINK                        "[ unlink ]"                # 10
611 @ MSG_SYS_UNLINK_ALT                    "unlink"
612 @ MSG_SYS_11                            "11"                        # 11 (u)
613 @ MSG_SYS_CHDIR                         "[ chdir ]"                 # 12
614 @ MSG_SYS_CHDIR_ALT                     "chdir"
615 @ MSG_SYS_TIME                          "[ time ]"                  # 13
616 @ MSG_SYS_TIME_ALT                      "time"
617 @ MSG_SYS_MKNOD                         "[ mknod ]"                 # 14
618 @ MSG_SYS_MKNOD_ALT                     "mknod"
619 @ MSG_SYS_CHMOD                         "[ chmod ]"                 # 15
620 @ MSG_SYS_CHMOD_ALT                     "chmod"
621 @ MSG_SYS_CHOWN                         "[ chown ]"                 # 16
622 @ MSG_SYS_CHOWN_ALT                     "chown"
623 @ MSG_SYS_BRK                           "[ brk ]"                   # 17
624 @ MSG_SYS_BRK_ALT                       "brk"
625 @ MSG_SYS_STAT                          "[ stat ]"                  # 18
626 @ MSG_SYS_STAT_ALT                      "stat"
627 @ MSG_SYS_LSEEK                         "[ lseek ]"                 # 19
628 @ MSG_SYS_LSEEK_ALT                     "lseek"
629 @ MSG_SYS_GETPID                        "[ getpid ]"                # 20
630 @ MSG_SYS_GETPID_ALT                    "getpid"
631 @ MSG_SYS_MOUNT                         "[ mount ]"                 # 21
632 @ MSG_SYS_MOUNT_ALT                     "mount"
633 @ MSG_SYS_22                            "22"                        # 22 (u)
634 @ MSG_SYS_SETUID                        "[ setuid ]"                # 23
635 @ MSG_SYS_SETUID_ALT                    "setuid"
636 @ MSG_SYS_GETUID                        "[ getuid ]"                # 24
637 @ MSG_SYS_GETUID_ALT                    "getuid"
638 @ MSG_SYS_STIME                         "[ stime ]"                 # 25
639 @ MSG_SYS_STIME_ALT                     "stime"
640 @ MSG_SYS_PCSAMPLE                      "[ pcsample ]"              # 26
641 @ MSG_SYS_PCSAMPLE_ALT                  "pcsample"
642 @ MSG_SYS_ALARM                         "[ alarm ]"                 # 27
643 @ MSG_SYS_ALARM_ALT                     "alarm"
644 @ MSG_SYS_FSTAT                         "[ fstat ]"                 # 28
645 @ MSG_SYS_FSTAT_ALT                     "fstat"
646 @ MSG_SYS_PAUSE                         "[ pause ]"                 # 29
647 @ MSG_SYS_PAUSE_ALT                     "pause"
648 @ MSG_SYS_30                            "30"                        # 30 (u)
649 @ MSG_SYS_STTY                          "[ stty ]"                  # 31
650 @ MSG_SYS_STTY_ALT                      "stty"
651 @ MSG_SYS_GTTY                          "[ gtty ]"                  # 32
652 @ MSG_SYS_GTTY_ALT                      "gtty"
653 @ MSG_SYS_ACCESS                        "[ access ]"                # 33
654 @ MSG_SYS_ACCESS_ALT                    "access"
```

```
655 @ MSG_SYS_NICE                      "[ nice ]"              # 34
656 @ MSG_SYS_NICE_ALT                  "nice"
657 @ MSG_SYS_STATFS                    "[ statfs ]"            # 35
658 @ MSG_SYS_STATFS_ALT                "statfs"
659 @ MSG_SYS_SYNC                      "[ sync ]"              # 36
660 @ MSG_SYS_SYNC_ALT                  "sync"
661 @ MSG_SYS_KILL                      "[ kill ]"              # 37
662 @ MSG_SYS_KILL_ALT                  "kill"
663 @ MSG_SYS_FSTATFS                   "[ fstatfs ]"           # 38
664 @ MSG_SYS_FSTATFS_ALT               "fstatfs"
665 @ MSG_SYS_PGRPSYS                   "[ pgrpsys ]"           # 39
666 @ MSG_SYS_PGRPSYS_ALT               "pgrpsys"
667 @ MSG_SYS_UUCOPYSTR                 "[ uucopystr ]"         # 40
668 @ MSG_SYS_UUCOPYSTR_ALT             "uucopystr"
669 @ MSG_SYS_41                        "41"                    # 41 (u)
670 @ MSG_SYS_PIPE                      "[ pipe ]"              # 42
671 @ MSG_SYS_PIPE_ALT                  "pipe"
672 @ MSG_SYS_TIMES                     "[ times ]"             # 43
673 @ MSG_SYS_TIMES_ALT                 "times"
674 @ MSG_SYS_PROFIL                    "[ profil ]"            # 44
675 @ MSG_SYS_PROFIL_ALT                "profil"
676 @ MSG_SYS_FACCESSAT                 "[ faccessat ]"         # 45
677 @ MSG_SYS_FACCESSAT_ALT             "faccessat"
678 @ MSG_SYS_SETGID                    "[ setgid ]"            # 46
679 @ MSG_SYS_SETGID_ALT                "setgid"
680 @ MSG_SYS_GETGID                    "[ getgid ]"            # 47
681 @ MSG_SYS_GETGID_ALT                "getgid"
682 @ MSG_SYS_48                        "48"                    # 48 (u)
683 @ MSG_SYS_MSGSYS                    "[ msgsys ]"            # 49
684 @ MSG_SYS_MSGSYS_ALT                "msgsys"
685 @ MSG_SYS_SYSI86                    "[ sysi86 ]"            # 50
686 @ MSG_SYS_SYSI86_ALT                "sysi86"
687 @ MSG_SYS_ACCT                      "[ acct ]"              # 51
688 @ MSG_SYS_ACCT_ALT                  "acct"
689 @ MSG_SYS_SHMSYS                    "[ shmsys ]"            # 52
690 @ MSG_SYS_SHMSYS_ALT                "shmsys"
691 @ MSG_SYS_SEMSYS                    "[ semsys ]"            # 53
692 @ MSG_SYS_SEMSYS_ALT                "semsys"
693 @ MSG_SYS_IOCTL                     "[ ioctl ]"             # 54
694 @ MSG_SYS_IOCTL_ALT                 "ioctl"
695 @ MSG_SYS_UADMIN                    "[ uadmin ]"            # 55
696 @ MSG_SYS_UADMIN_ALT                "uadmin"
697 @ MSG_SYS_FCHOWNAT                  "[ fchownat ]"          # 56
698 @ MSG_SYS_FCHOWNAT_ALT              "fchownat"
699 @ MSG_SYS_UTSSYS                    "[ utssys ]"            # 57
700 @ MSG_SYS_UTSSYS_ALT                "utssys"
701 @ MSG_SYS_FDSYNC                    "[ fdsync ]"            # 58
702 @ MSG_SYS_FDSYNC_ALT                "fdsync"
703 @ MSG_SYS_EXECVE                    "[ execve ]"            # 59
704 @ MSG_SYS_EXECVE_ALT                "execve"
705 @ MSG_SYS_UMASK                     "[ umask ]"             # 60
706 @ MSG_SYS_UMASK_ALT                 "umask"
707 @ MSG_SYS_CHROOT                    "[ chroot ]"            # 61
708 @ MSG_SYS_CHROOT_ALT                "chroot"
709 @ MSG_SYS_FCNTL                     "[ fcntl ]"             # 62
710 @ MSG_SYS_FCNTL_ALT                 "fcntl"
711 @ MSG_SYS_ULIMIT                    "[ ulimit ]"            # 63
712 @ MSG_SYS_ULIMIT_ALT                "ulimit"
713 @ MSG_SYS_RENAMEAT                  "[ renameat ]"          # 64
714 @ MSG_SYS_RENAMEAT_ALT              "renameat"
715 @ MSG_SYS_UNLINKAT                  "[ unlinkat ]"          # 65
716 @ MSG_SYS_UNLINKAT_ALT              "unlinkat"
717 @ MSG_SYS_FSTATAT                   "[ fstatat ]"           # 66
718 @ MSG_SYS_FSTATAT_ALT               "fstatat"
719 @ MSG_SYS_FSTATAT64                 "[ fstatat64 ]"         # 67
720 @ MSG_SYS_FSTATAT64_ALT             "fstatat64"
```



```
721 @ MSG_SYS_OPENAT                    "[ openat ]"            # 68
722 @ MSG_SYS_OPENAT_ALT                "openat"
723 @ MSG_SYS_OPENAT64                  "[ openat64 ]"          # 69
724 @ MSG_SYS_OPENAT64_ALT              "openat64"
725 @ MSG_SYS_TASKSYS                   "[ tasksys ]"           # 70
726 @ MSG_SYS_TASKSYS_ALT               "tasksys        "
727 @ MSG_SYS_ACCTCTL                   "[ acctctl ]"           # 71
728 @ MSG_SYS_ACCTCTL_ALT               "acctctl        "
729 @ MSG_SYS_EXACCTSYS                 "[ exacctsys ]"         # 72
730 @ MSG_SYS_EXACCTSYS_ALT             "exacctsys"
731 @ MSG_SYS_GETPAGESIZES              "[ getpagesizes ]"      # 73
732 @ MSG_SYS_GETPAGESIZES_ALT          "getpagesizes"
733 @ MSG_SYS_RCTLSYS                   "[ rctlsys ]"           # 74
734 @ MSG_SYS_RCTLSYS_ALT               "rctlsys        "
735 @ MSG_SYS_SIDSYS                    "[ sidsys ]"            # 75
736 @ MSG_SYS_SIDSYS_ALT                "sidsys"
737 @ MSG_SYS_76                        "76"                    # 76 (u)
738 @ MSG_SYS_LWP_PARK                  "[ lwp_park ]"          # 77
739 @ MSG_SYS_LWP_PARK_ALT              "lwp_park"
740 @ MSG_SYS_SENDFILEV                 "[ sendfilev ]"         # 78
741 @ MSG_SYS_SENDFILEV_ALT             "sendfilev"
742 @ MSG_SYS_RMDIR                     "[ rmdir ]"             # 79
743 @ MSG_SYS_RMDIR_ALT                 "rmdir"
744 @ MSG_SYS_MKDIR                     "[ mkdir ]"             # 80
745 @ MSG_SYS_MKDIR_ALT                 "mkdir"
746 @ MSG_SYS_GETDENTS                  "[ getdents ]"          # 81
747 @ MSG_SYS_GETDENTS_ALT              "getdents"
748 @ MSG_SYS_PRIVSYS                   "[ privsys ]"           # 82
749 @ MSG_SYS_PRIVSYS_ALT               "privsys"
750 @ MSG_SYS_UCREDSYS                  "[ ucredsys ]"          # 83
751 @ MSG_SYS_UCREDSYS_ALT              "ucredsys"
752 @ MSG_SYS_SYSFS                     "[ sysfs ]"             # 84
753 @ MSG_SYS_SYSFS_ALT                 "sysfs"
754 @ MSG_SYS_GETMSG                    "[ getmsg ]"            # 85
755 @ MSG_SYS_GETMSG_ALT                "getmsg"
756 @ MSG_SYS_PUTMSG                    "[ putmsg ]"            # 86
757 @ MSG_SYS_PUTMSG_ALT                "putmsg"
758 @ MSG_SYS_87                        "87"                    # 87 (u)
759 @ MSG_SYS_LSTAT                     "[ lstat ]"             # 88
760 @ MSG_SYS_LSTAT_ALT                 "lstat"
761 @ MSG_SYS_SYMLINK                   "[ symlink ]"           # 89
762 @ MSG_SYS_SYMLINK_ALT               "symlink"
763 @ MSG_SYS_READLINK                  "[ readlink ]"          # 90
764 @ MSG_SYS_READLINK_ALT              "readlink"
765 @ MSG_SYS_SETGROUPS                 "[ setgroups ]"         # 91
766 @ MSG_SYS_SETGROUPS_ALT             "setgroups"
767 @ MSG_SYS_GETGROUPS                 "[ getgroups ]"         # 92
768 @ MSG_SYS_GETGROUPS_ALT             "getgroups"
769 @ MSG_SYS_FCHMOD                    "[ fchmod ]"            # 93
770 @ MSG_SYS_FCHMOD_ALT                "fchmod"
771 @ MSG_SYS_FCHOWN                    "[ fchown ]"            # 94
772 @ MSG_SYS_FCHOWN_ALT                "fchown"
773 @ MSG_SYS_SIGPROCMASK               "[ sigprocmask ]"       # 95
774 @ MSG_SYS_SIGPROCMASK_ALT           "sigprocmask"
775 @ MSG_SYS_SIGSUSPEND                "[ sigsuspend ]"        # 96
776 @ MSG_SYS_SIGSUSPEND_ALT            "sigsuspend"
777 @ MSG_SYS_SIGALTSTACK               "[ sigaltstack ]"       # 97
778 @ MSG_SYS_SIGALTSTACK_ALT           "sigaltstack"
779 @ MSG_SYS_SIGACTION                 "[ sigaction ]"         # 98
780 @ MSG_SYS_SIGACTION_ALT             "sigaction"
781 @ MSG_SYS_SIGPENDING                "[ sigpending ]"        # 99
782 @ MSG_SYS_SIGPENDING_ALT            "sigpending"
783 @ MSG_SYS_CONTEXT                   "[ context ]"           # 100
784 @ MSG_SYS_CONTEXT_ALT               "context"
785 @ MSG_SYS_101                       "101"                   # 101(u)
786 @ MSG_SYS_102                       "102"                   # 102(u)
```

```
787 @ MSG_SYS_STATVFS                   "[ statvfs ]"               # 103
788 @ MSG_SYS_STATVFS_ALT               "statvfs"
789 @ MSG_SYS_FSTATVFS                  "[ fstatvfs ]"              # 104
790 @ MSG_SYS_FSTATVFS_ALT              "fstatvfs"
791 @ MSG_SYS_GETLOADAVG                "[ getloadavg ]"            # 105
792 @ MSG_SYS_GETLOADAVG_ALT            "getloadavg"
793 @ MSG_SYS_NFSSYS                    "[ nfssys ]"                # 106
794 @ MSG_SYS_NFSSYS_ALT                "nfssys"
795 @ MSG_SYS_WAITID                    "[ waitid ]"                # 107
796 @ MSG_SYS_WAITID_ALT                "waitid"
797 @ MSG_SYS_SIGSENDSYS                "[ sigsendsys ]"            # 108
798 @ MSG_SYS_SIGSENDSYS_ALT            "sigsendsys"
799 @ MSG_SYS_HRTSYS                    "[ hrtsys ]"                # 109
800 @ MSG_SYS_HRTSYS_ALT                "hrtsys"
801 @ MSG_SYS_UTIMESYS                  "[ utimesys ]"              # 110
802 @ MSG_SYS_UTIMESYS_ALT              "utimesys"
803 @ MSG_SYS_SIGRESEND                 "[ sigresend ]"             # 111
804 @ MSG_SYS_SIGRESEND_ALT             "sigresend"
805 @ MSG_SYS_PRIOCNTLSYS               "[ priocntlsys ]"           # 112
806 @ MSG_SYS_PRIOCNTLSYS_ALT           "priocntlsys"
807 @ MSG_SYS_PATHCONF                  "[ pathconf ]"              # 113
808 @ MSG_SYS_PATHCONF_ALT              "pathconf"
809 @ MSG_SYS_MINCORE                   "[ mincore ]"               # 114
810 @ MSG_SYS_MINCORE_ALT               "mincore"
811 @ MSG_SYS_MMAP                      "[ mmap ]"                  # 115
812 @ MSG_SYS_MMAP_ALT                  "mmap"
813 @ MSG_SYS_MPROTECT                  "[ mprotect ]"              # 116
814 @ MSG_SYS_MPROTECT_ALT              "mprotect"
815 @ MSG_SYS_MUNMAP                    "[ munmap ]"                # 117
816 @ MSG_SYS_MUNMAP_ALT                "munmap"
817 @ MSG_SYS_FPATHCONF                 "[ fpathconf ]"             # 118
818 @ MSG_SYS_FPATHCONF_ALT             "fpathconf"
819 @ MSG_SYS_VFORK                     "[ vfork ]"                 # 119
820 @ MSG_SYS_VFORK_ALT                 "vfork"
821 @ MSG_SYS_FCHDIR                    "[ fchdir ]"                # 120
822 @ MSG_SYS_FCHDIR_ALT                "fchdir"
823 @ MSG_SYS_READV                     "[ readv ]"                 # 121
824 @ MSG_SYS_READV_ALT                 "readv"
825 @ MSG_SYS_WRITEV                    "[ writev ]"                # 122
826 @ MSG_SYS_WRITEV_ALT                "writev"
827 @ MSG_SYS_123                       "123"                       # 123(u)
828 @ MSG_SYS_124                       "124"                       # 124(u)
829 @ MSG_SYS_125                       "125"                       # 125(u)
830 @ MSG_SYS_126                       "126"                       # 126(u)
831 @ MSG_SYS_MMAPOBJ                   "[ mmapobj ]"               # 127
832 @ MSG_SYS_MMAPOBJ_ALT               "mmapobj"
833 @ MSG_SYS_SETRLIMIT                 "[ setrlimit ]"             # 128
834 @ MSG_SYS_SETRLIMIT_ALT             "setrlimit"
835 @ MSG_SYS_GETRLIMIT                 "[ getrlimit ]"             # 129
836 @ MSG_SYS_GETRLIMIT_ALT             "getrlimit"
837 @ MSG_SYS_LCHOWN                    "[ lchown ]"                # 130
838 @ MSG_SYS_LCHOWN_ALT                "lchown"
839 @ MSG_SYS_MEMCNTL                   "[ memcntl ]"               # 131
840 @ MSG_SYS_MEMCNTL_ALT               "memcntl"
841 @ MSG_SYS_GETPMSG                   "[ getpmsg ]"               # 132
842 @ MSG_SYS_GETPMSG_ALT               "getpmsg"
843 @ MSG_SYS_PUTPMSG                   "[ putpmsg ]"               # 133
844 @ MSG_SYS_PUTPMSG_ALT               "putpmsg"
845 @ MSG_SYS_RENAME                    "[ rename ]"                # 134
846 @ MSG_SYS_RENAME_ALT                "rename"
847 @ MSG_SYS_UNAME                     "[ uname ]"                 # 135
848 @ MSG_SYS_UNAME_ALT                 "uname"
849 @ MSG_SYS_SETEGID                   "[ setegid ]"               # 136
850 @ MSG_SYS_SETEGID_ALT               "setegid"
851 @ MSG_SYS_SYSCONFIG                 "[ sysconfig ]"             # 137
852 @ MSG_SYS_SYSCONFIG_ALT             "sysconfig"
```



```
853 @ MSG_SYS_ADJTIME                   "[ adjtime ]"               # 138
854 @ MSG_SYS_ADJTIME_ALT               "adjtime"
855 @ MSG_SYS_SYSTEMINFO                "[ systeminfo ]"            # 139
856 @ MSG_SYS_SYSTEMINFO_ALT            "systeminfo"
857 @ MSG_SYS_SHAREFS                   "[ sharefs ]"               # 140
858 @ MSG_SYS_SHAREFS_ALT               "sharefs"
859 @ MSG_SYS_SETEUID                   "[ seteuid ]"               # 141
860 @ MSG_SYS_SETEUID_ALT               "seteuid"
861 @ MSG_SYS_FORKSYS                   "[ forksys ]"               # 142
862 @ MSG_SYS_FORKSYS_ALT               "forksys"
863 @ MSG_SYS_143                       "143"                       # 143(u)
864 @ MSG_SYS_SIGTIMEDWAIT              "[ sigtimedwait ]"          # 144
865 @ MSG_SYS_SIGTIMEDWAIT_ALT          "sigtimedwait"
866 @ MSG_SYS_LWP_INFO                  "[ lwp_info ]"              # 145
867 @ MSG_SYS_LWP_INFO_ALT              "lwp_info"
868 @ MSG_SYS_YIELD                     "[ yield ]"                 # 146
869 @ MSG_SYS_YIELD_ALT                 "yield"
870 @ MSG_SYS_147                       "147"                       # 147(u)
871 @ MSG_SYS_LWP_SEMA_POST             "[ lwp_sema_post ]"         # 148
872 @ MSG_SYS_LWP_SEMA_POST_ALT         "lwp_sema_post"
873 @ MSG_SYS_LWP_SEMA_TRYWAIT          "[ lwp_sema_trywait ]"      # 149
874 @ MSG_SYS_LWP_SEMA_TRYWAIT_ALT      "lwp_sema_trywait"
875 @ MSG_SYS_LWP_DETACH                "[ lwp_detach ]"            # 150
876 @ MSG_SYS_LWP_DETACH_ALT            "lwp_detach"
877 @ MSG_SYS_CORECTL                   "[ corectl ]"               # 151
878 @ MSG_SYS_CORECTL_ALT               "corectl"
879 @ MSG_SYS_MODCTL                    "[ modctl ]"                # 152
880 @ MSG_SYS_MODCTL_ALT                "modctl"
881 @ MSG_SYS_FCHROOT                   "[ fchroot ]"               # 153
882 @ MSG_SYS_FCHROOT_ALT               "fchroot"
883 @ MSG_SYS_154                       "154"                       # 154(u)
884 @ MSG_SYS_VHANGUP                   "[ vhangup ]"               # 155
885 @ MSG_SYS_VHANGUP_ALT               "vhangup"
886 @ MSG_SYS_GETTIMEOFDAY              "[ gettimeofday ]"          # 156
887 @ MSG_SYS_GETTIMEOFDAY_ALT          "gettimeofday"
888 @ MSG_SYS_GETITIMER                 "[ getitimer ]"             # 157
889 @ MSG_SYS_GETITIMER_ALT             "getitimer"
890 @ MSG_SYS_SETITIMER                 "[ setitimer ]"             # 158
891 @ MSG_SYS_SETITIMER_ALT             "setitimer"
892 @ MSG_SYS_LWP_CREATE                "[ lwp_create ]"            # 159
893 @ MSG_SYS_LWP_CREATE_ALT            "lwp_create"
894 @ MSG_SYS_LWP_EXIT                  "[ lwp_exit ]"              # 160
895 @ MSG_SYS_LWP_EXIT_ALT              "lwp_exit"
896 @ MSG_SYS_LWP_SUSPEND               "[ lwp_suspend ]"           # 161
897 @ MSG_SYS_LWP_SUSPEND_ALT           "lwp_suspend"
898 @ MSG_SYS_LWP_CONTINUE              "[ lwp_continue ]"          # 162
899 @ MSG_SYS_LWP_CONTINUE_ALT          "lwp_continue"
900 @ MSG_SYS_LWP_KILL                  "[ lwp_kill ]"              # 163
901 @ MSG_SYS_LWP_KILL_ALT              "lwp_kill"
902 @ MSG_SYS_LWP_SELF                  "[ lwp_self ]"              # 164
903 @ MSG_SYS_LWP_SELF_ALT              "lwp_self"
904 @ MSG_SYS_LWP_SIGMASK               "[ lwp_sigmask ]"           # 165
905 @ MSG_SYS_LWP_SIGMASK_ALT           "lwp_sigmask"
906 @ MSG_SYS_LWP_PRIVATE               "[ lwp_private ]"           # 166
907 @ MSG_SYS_LWP_PRIVATE_ALT           "lwp_private"
908 @ MSG_SYS_LWP_WAIT                  "[ lwp_wait ]"              # 167
909 @ MSG_SYS_LWP_WAIT_ALT              "lwp_wait"
910 @ MSG_SYS_LWP_MUTEX_WAKEUP          "[ lwp_mutex_wakeup ]"      # 168
911 @ MSG_SYS_LWP_MUTEX_WAKEUP_ALT      "lwp_mutex_wakeup"
912 @ MSG_SYS_169                       "169"                       # 169(u)
913 @ MSG_SYS_LWP_COND_WAIT             "[ lwp_cond_wait ]"         # 170
914 @ MSG_SYS_LWP_COND_WAIT_ALT         "lwp_cond_wait"
915 @ MSG_SYS_LWP_COND_SIGNAL           "[ lwp_cond_signal ]"       # 171
916 @ MSG_SYS_LWP_COND_SIGNAL_ALT       "lwp_cond_signal"
917 @ MSG_SYS_LWP_COND_BROADCAST        "[ lwp_cond_broadcast ]"    # 172
918 @ MSG_SYS_LWP_COND_BROADCAST_ALT    "lwp_cond_broadcast"
```

```
 919 @ MSG_SYS_PREAD                     "[ pread ]"                 # 173
 920 @ MSG_SYS_PREAD_ALT                 "pread"
 921 @ MSG_SYS_PWRITE                    "[ pwrite ]"                # 174
 922 @ MSG_SYS_PWRITE_ALT                "pwrite"
 923 @ MSG_SYS_LLSEEK                    "[ llseek ]"                # 175
 924 @ MSG_SYS_LLSEEK_ALT                "llseek"
 925 @ MSG_SYS_INST_SYNC                 "[ inst_sync ]"             # 176
 926 @ MSG_SYS_INST_SYNC_ALT             "inst_sync"
 927 @ MSG_SYS_BRAND                     "[ brand ]"                 # 177
 928 @ MSG_SYS_BRAND_ALT                 "brand"
 929 @ MSG_SYS_KAIO                      "[ kaio ]"                  # 178
 930 @ MSG_SYS_KAIO_ALT                  "kaio"
 931 @ MSG_SYS_CPC                       "[ cpc ]"                   # 179
 932 @ MSG_SYS_CPC_ALT                   "cpc"
 933 @ MSG_SYS_LGRPSYS                   "[ lgrpsys ]"               # 180
 934 @ MSG_SYS_LGRPSYS_ALT               "lgrpsys"
 935 @ MSG_SYS_RUSAGESYS                 "[ rusagesys ]"             # 181
 936 @ MSG_SYS_RUSAGESYS_ALT             "rusagesys"
 937 @ MSG_SYS_PORT                      "[ port ]"                  # 182
 938 @ MSG_SYS_PORT_ALT                  "port"
 939 @ MSG_SYS_POLLSYS                   "[ pollsys ]"               # 183
 940 @ MSG_SYS_POLLSYS_ALT               "pollsys"
 941 @ MSG_SYS_LABELSYS                  "[ labelsys ]"              # 184
 942 @ MSG_SYS_LABELSYS_ALT              "labelsys"
 943 @ MSG_SYS_ACL                       "[ acl ]"                   # 185
 944 @ MSG_SYS_ACL_ALT                   "acl"
 945 @ MSG_SYS_AUDITSYS                  "[ auditsys ]"              # 186
 946 @ MSG_SYS_AUDITSYS_ALT              "auditsys"
 947 @ MSG_SYS_PROCESSOR_BIND           "[ processor_bind ]"        # 187
 948 @ MSG_SYS_PROCESSOR_BIND_ALT       "processor_bind"
 949 @ MSG_SYS_PROCESSOR_INFO           "[ processor_info ]"        # 188
 950 @ MSG_SYS_PROCESSOR_INFO_ALT       "processor_info"
 951 @ MSG_SYS_P_ONLINE                 "[ p_online ]"              # 189
 952 @ MSG_SYS_P_ONLINE_ALT             "p_online"
 953 @ MSG_SYS_SIGQUEUE                 "[ sigqueue ]"              # 190
 954 @ MSG_SYS_SIGQUEUE_ALT             "sigqueue"
 955 @ MSG_SYS_CLOCK_GETTIME            "[ clock_gettime ]"         # 191
 956 @ MSG_SYS_CLOCK_GETTIME_ALT        "clock_gettime"
 957 @ MSG_SYS_CLOCK_SETTIME            "[ clock_settime ]"         # 192
 958 @ MSG_SYS_CLOCK_SETTIME_ALT        "clock_settime"
 959 @ MSG_SYS_CLOCK_GETRES             "[ clock_getres ]"          # 193
 960 @ MSG_SYS_CLOCK_GETRES_ALT         "clock_getres"
 961 @ MSG_SYS_TIMER_CREATE            "[ timer_create ]"          # 194
 962 @ MSG_SYS_TIMER_CREATE_ALT        "timer_create"
 963 @ MSG_SYS_TIMER_DELETE            "[ timer_delete ]"          # 195
 964 @ MSG_SYS_TIMER_DELETE_ALT        "timer_delete"
 965 @ MSG_SYS_TIMER_SETTIME           "[ timer_settime ]"         # 196
 966 @ MSG_SYS_TIMER_SETTIME_ALT       "timer_settime"
 967 @ MSG_SYS_TIMER_GETTIME           "[ timer_gettime ]"         # 197
 968 @ MSG_SYS_TIMER_GETTIME_ALT       "timer_gettime"
 969 @ MSG_SYS_TIMER_GETOVERRUN        "[ timer_getoverrun ]"      # 198
 970 @ MSG_SYS_TIMER_GETOVERRUN_ALT    "timer_getoverrun"
 971 @ MSG_SYS_NANOSLEEP               "[ nanosleep ]"             # 199
 972 @ MSG_SYS_NANOSLEEP_ALT           "nanosleep"
 973 @ MSG_SYS_FACL                    "[ facl ]"                  # 200
 974 @ MSG_SYS_FACL_ALT                "facl"
 975 @ MSG_SYS_DOOR                    "[ door ]"                  # 201
 976 @ MSG_SYS_DOOR_ALT                "door"
 977 @ MSG_SYS_SETREUID                "[ setreuid ]"              # 202
 978 @ MSG_SYS_SETREUID_ALT            "setreuid"
 979 @ MSG_SYS_SETREGID                "[ setregid ]"              # 203
 980 @ MSG_SYS_SETREGID_ALT            "setregid"
 981 @ MSG_SYS_INSTALL_UTRAP           "[ install_utrap ]"         # 204
 982 @ MSG_SYS_INSTALL_UTRAP_ALT       "install_utrap"
 983 @ MSG_SYS_SIGNOTIFY               "[ signotify ]"             # 205
 984 @ MSG_SYS_SIGNOTIFY_ALT           "signotify"
```



```
 985 @ MSG_SYS_SCHEDCTL                "[ schedctl ]"              # 206
 986 @ MSG_SYS_SCHEDCTL_ALT            "schedctl"
 987 @ MSG_SYS_PSET                    "[ pset ]"                  # 207
 988 @ MSG_SYS_PSET_ALT                "pset"
 989 @ MSG_SYS_SPARC_UTRAP_INSTALL     "[ sparc_utrap_install ]"   # 208
 990 @ MSG_SYS_SPARC_UTRAP_INSTALL_ALT "sparc_utrap_install"
 991 @ MSG_SYS_RESOLVEPATH             "[ resolvepath ]"           # 209
 992 @ MSG_SYS_RESOLVEPATH_ALT         "resolvepath"
 993 @ MSG_SYS_LWP_MUTEX_TIMEDLOCK    "[ lwp_mutex_timedlock ]"   # 210
 994 @ MSG_SYS_LWP_MUTEX_TIMEDLOCK_ALT "lwp_mutex_timedlock"
 995 @ MSG_SYS_LWP_SEMA_TIMEDWAIT     "[ lwp_sema_timedwait ]"    # 211
 996 @ MSG_SYS_LWP_SEMA_TIMEDWAIT_ALT "lwp_sema_timedwait"
 997 @ MSG_SYS_LWP_RWLOCK_SYS        "[ lwp_rwlock_sys ]"        # 212
 998 @ MSG_SYS_LWP_RWLOCK_SYS_ALT    "lwp_rwlock_sys"
 999 @ MSG_SYS_GETDENTS64             "[ getdents64 ]"            # 213
1000 @ MSG_SYS_GETDENTS64_ALT         "getdents64"
1001 @ MSG_SYS_MMAP64                 "[ mmap64 ]"                # 214
1002 @ MSG_SYS_MMAP64_ALT             "mmap64"
1003 @ MSG_SYS_STAT64                 "[ stat64 ]"                # 215
1004 @ MSG_SYS_STAT64_ALT             "stat64"
1005 @ MSG_SYS_LSTAT64                "[ lstat64 ]"               # 216
1006 @ MSG_SYS_LSTAT64_ALT            "lstat64"
1007 @ MSG_SYS_FSTAT64                "[ fstat64 ]"               # 217
1008 @ MSG_SYS_FSTAT64_ALT            "fstat64"
1009 @ MSG_SYS_STATVFS64              "[ statvfs64 ]"             # 218
1010 @ MSG_SYS_STATVFS64_ALT          "statvfs64"
1011 @ MSG_SYS_FSTATVFS64             "[ fstatvfs64 ]"            # 219
1012 @ MSG_SYS_FSTATVFS64_ALT         "fstatvfs64"
1013 @ MSG_SYS_SETRLIMIT64            "[ setrlimit64 ]"           # 220
1014 @ MSG_SYS_SETRLIMIT64_ALT        "setrlimit64"
1015 @ MSG_SYS_GETRLIMIT64            "[ getrlimit64 ]"           # 221
1016 @ MSG_SYS_GETRLIMIT64_ALT        "getrlimit64"
1017 @ MSG_SYS_PREAD64                "[ pread64 ]"               # 222
1018 @ MSG_SYS_PREAD64_ALT            "pread64"
1019 @ MSG_SYS_PWRITE64               "[ pwrite64 ]"              # 223
1020 @ MSG_SYS_PWRITE64_ALT           "pwrite64"
1021 @ MSG_SYS_224                    "224"                       # 224(u)
1022 @ MSG_SYS_OPEN64                 "[ open64 ]"                # 225
1023 @ MSG_SYS_OPEN64_ALT             "open64"
1024 @ MSG_SYS_RPCSYS                 "[ rpcsys ]"                # 226
1025 @ MSG_SYS_RPCSYS_ALT             "rpcsys"
1026 @ MSG_SYS_ZONE                   "[ zone ]"                  # 227
1027 @ MSG_SYS_ZONE_ALT               "zone"
1028 @ MSG_SYS_AUTOFSSYS              "[ autofssys ]"             # 228
1029 @ MSG_SYS_AUTOFSSYS_ALT          "autofssys"
1030 @ MSG_SYS_GETCWD                 "[ getcwd ]"                # 229
1031 @ MSG_SYS_GETCWD_ALT             "getcwd"
1032 @ MSG_SYS_SO_SOCKET              "[ so_socket ]"             # 230
1033 @ MSG_SYS_SO_SOCKET_ALT          "so_socket"
1034 @ MSG_SYS_SO_SOCKETPAIR          "[ so_socketpair ]"         # 231
1035 @ MSG_SYS_SO_SOCKETPAIR_ALT      "so_socketpair"
1036 @ MSG_SYS_BIND                   "[ bind ]"                  # 232
1037 @ MSG_SYS_BIND_ALT               "bind"
1038 @ MSG_SYS_LISTEN                 "[ listen ]"                # 233
1039 @ MSG_SYS_LISTEN_ALT             "listen"
1040 @ MSG_SYS_ACCEPT                 "[ accept ]"                # 234
1041 @ MSG_SYS_ACCEPT_ALT             "accept"
1042 @ MSG_SYS_CONNECT                "[ connect ]"               # 235
1043 @ MSG_SYS_CONNECT_ALT            "connect"
1044 @ MSG_SYS_SHUTDOWN               "[ shutdown ]"              # 236
1045 @ MSG_SYS_SHUTDOWN_ALT           "shutdown"
1046 @ MSG_SYS_RECV                   "[ recv ]"                  # 237
1047 @ MSG_SYS_RECV_ALT               "recv"
1048 @ MSG_SYS_RECVFROM               "[ recvfrom ]"              # 238
1049 @ MSG_SYS_RECVFROM_ALT           "recvfrom"
1050 @ MSG_SYS_RECVMSG                "[ recvmsg ]"               # 239
```

```
1051 @ MSG_SYS_RECVMSG_ALT                "recvmsg"
1052 @ MSG_SYS_SEND                       "[ send ]"                    # 240
1053 @ MSG_SYS_SEND_ALT                   "send"
1054 @ MSG_SYS_SENDMSG                    "[ sendmsg ]"                 # 241
1055 @ MSG_SYS_SENDMSG_ALT                "sendmsg"
1056 @ MSG_SYS_SENDTO                     "[ sendto ]"                  # 242
1057 @ MSG_SYS_SENDTO_ALT                 "sendto"
1058 @ MSG_SYS_GETPEERNAME               "[ getpeername ]"             # 243
1059 @ MSG_SYS_GETPEERNAME_ALT           "getpeername"
1060 @ MSG_SYS_GETSOCKNAME               "[ getsockname ]"             # 244
1061 @ MSG_SYS_GETSOCKNAME_ALT           "getsockname"
1062 @ MSG_SYS_GETSOCKOPT                "[ getsockopt ]"              # 245
1063 @ MSG_SYS_GETSOCKOPT_ALT            "getsockopt"
1064 @ MSG_SYS_SETSOCKOPT                "[ setsockopt ]"              # 246
1065 @ MSG_SYS_SETSOCKOPT_ALT            "setsockopt"
1066 @ MSG_SYS_SOCKCONFIG                "[ sockconfig ]"              # 247
1067 @ MSG_SYS_SOCKCONFIG_ALT            "sockconfig"
1068 @ MSG_SYS_NTP_GETTIME               "[ ntp_gettime ]"             # 248
1069 @ MSG_SYS_NTP_GETTIME_ALT           "ntp_gettime"
1070 @ MSG_SYS_NTP_ADJTIME               "[ ntp_adjtime ]"             # 249
1071 @ MSG_SYS_NTP_ADJTIME_ALT           "ntp_adjtime"
1072 @ MSG_SYS_LWP_MUTEX_UNLOCK          "[ lwp_mutex_unlock ]"        # 250
1073 @ MSG_SYS_LWP_MUTEX_UNLOCK_ALT      "lwp_mutex_unlock"
1074 @ MSG_SYS_LWP_MUTEX_TRYLOCK         "[ lwp_mutex_trylock ]"       # 251
1075 @ MSG_SYS_LWP_MUTEX_TRYLOCK_ALT     "lwp_mutex_trylock"
1076 @ MSG_SYS_LWP_MUTEX_REGISTER        "[ lwp_mutex_register ]"      # 252
1077 @ MSG_SYS_LWP_MUTEX_REGISTER_ALT    "lwp_mutex_register"
1078 @ MSG_SYS_CLADM                     "[ cladm ]"                   # 253
1079 @ MSG_SYS_CLADM_ALT                 "cladm"
1080 @ MSG_SYS_UUCOPY                    "[ uucopy ]"                  # 254
1081 @ MSG_SYS_UUCOPY_ALT                "uucopy"
1082 @ MSG_SYS_UMOUNT2                   "[ umount2 ]"                 # 255
1083 @ MSG_SYS_UMOUNT2_ALT               "umount2"

1085 @ MSG_PR_O_RDONLY                   "O_RDONLY"
1086 @ MSG_PR_O_WRONLY                   "O_WRONLY"
1087 @ MSG_PR_O_RDWR                     "O_RDWR"
1088 @ MSG_PR_O_SEARCH                   "O_SEARCH"
1089 @ MSG_PR_O_EXEC                     "O_EXEC"
1090 @ MSG_PR_O_NDELAY                   "O_NDELAY"
1091 @ MSG_PR_O_NONBLOCK                 "O_NONBLOCK"
1092 @ MSG_PR_O_APPEND                   "O_APPEND"
1093 @ MSG_PR_O_SYNC                     "O_SYNC"
1094 @ MSG_PR_O_DSYNC                    "O_DSYNC"
1095 @ MSG_PR_O_RSYNC                    "O_RSYNC"
1096 @ MSG_PR_O_CREAT                    "O_CREAT"
1097 @ MSG_PR_O_TRUNC                    "O_TRUNC"
1098 @ MSG_PR_O_EXCL                     "O_EXCL"
1099 @ MSG_PR_O_NOCTTY                   "O_NOCTTY"
1100 @ MSG_PR_O_LARGEFILE                "O_LARGEFILE"
1101 @ MSG_PR_O_XATTR                    "O_XATTR"
1102 @ MSG_PR_O_NOFOLLOW                 "O_NOFOLLOW"
1103 @ MSG_PR_O_NOLINKS                  "O_NOLINKS"

1105 @ MSG_S_IFIFO                       "S_IFIFO"
1106 @ MSG_S_IFCHR                       "S_IFCHR"
1107 @ MSG_S_IFDIR                       "S_IFDIR"
1108 @ MSG_S_IFNAM                       "S_IFNAM"
1109 @ MSG_S_IFBLK                       "S_IFBLK"
1110 @ MSG_S_IFREG                       "S_IFREG"
1111 @ MSG_S_IFLNK                       "S_IFLNK"
1112 @ MSG_S_IFSOCK                      "S_IFSOCK"
1113 @ MSG_S_IFDOOR                      "S_IFDOOR"
1114 @ MSG_S_IFPORT                      "S_IFPORT"
1115 @ MSG_S_ISUID                       "S_ISUID"
1116 @ MSG_S_ISGID                       "S_ISGID"
```



```
1117 @ MSG_S_ISVTX                       "S_ISVTX"
1118 @ MSG_S_IRUSR                       "S_IRUSR"
1119 @ MSG_S_IWUSR                       "S_IWUSR"
1120 @ MSG_S_IXUSR                       "S_IXUSR"
1121 @ MSG_S_IRGRP                       "S_IRGRP"
1122 @ MSG_S_IWGRP                       "S_IWGRP"
1123 @ MSG_S_IXGRP                       "S_IXGRP"
1124 @ MSG_S_IROTH                       "S_IROTH"
1125 @ MSG_S_IWOTH                       "S_IWOTH"
1126 @ MSG_S_IXOTH                       "S_IXOTH"

1128 @ MSG_GBL_ZERO          "0"

1130 @ MSG_FMT_INT           "%d"
1131 @ MSG_FMT_WORD          "%u"
1132 @ MSG_FMT_HEXINT        "%#x"
```

```
**********************************************************
  170477 Mon Oct 15 13:25:12 2018
new/usr/src/cmd/svc/startd/graph.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright (c) 2015, Syneto S.R.L. All rights reserved.
 * Copyright 2016 Toomas Soome <tsoome@me.com>
 * Copyright 2016 RackTop Systems.
 */

/*
 * graph.c - master restarter graph engine
 *
 *   The graph engine keeps a dependency graph of all service instances on the
 *   system, as recorded in the repository.  It decides when services should
 *   be brought up or down based on service states and dependencies and sends
 *   commands to restarters to effect any changes.  It also executes
 *   administrator commands sent by svcadm via the repository.
 *
 *   The graph is stored in uu_list_t *dgraph and its vertices are
 *   graph_vertex_t's, each of which has a name and an integer id unique to
 *   its name (see dict.c).  A vertex's type attribute designates the type
 *   of object it represents: GVT_INST for service instances, GVT_SVC for
 *   service objects (since service instances may depend on another service,
 *   rather than service instance), GVT_FILE for files (which services may
 *   depend on), and GVT_GROUP for dependencies on multiple objects.  GVT_GROUP
 *   vertices are necessary because dependency lists may have particular
 *   grouping types (require any, require all, optional, or exclude) and
 *   event-propagation characteristics.
 *
 *   The initial graph is built by libscf_populate_graph() invoking
 *   dgraph_add_instance() for each instance in the repository.  The function
 *   adds a GVT_SVC vertex for the service if one does not already exist, adds
 *   a GVT_INST vertex named by the FMRI of the instance, and sets up the edges.
 *   The resulting web of vertices & edges associated with an instance's vertex
 *   includes
 *
 *     - an edge from the GVT_SVC vertex for the instance's service
 *
 *     - an edge to the GVT_INST vertex of the instance's resarter, if its
 *       restarter is not svc.startd
 *
 *     - edges from other GVT_INST vertices if the instance is a restarter
 *
 *     - for each dependency property group in the instance's "running"
 *       snapshot, an edge to a GVT_GROUP vertex named by the FMRI of the
 *       instance and the name of the property group
 *
 *     - for each value of the "entities" property in each dependency property
 *       group, an edge from the corresponding GVT_GROUP vertex to a
 *       GVT_INST, GVT_SVC, or GVT_FILE vertex
 *
 *     - edges from GVT_GROUP vertices for each dependent instance
 *
 *   After the edges are set up the vertex's GV_CONFIGURED flag is set.  If
 *   there are problems, or if a service is mentioned in a dependency but does
 *   not exist in the repository, the GV_CONFIGURED flag will be clear.
 *
 *   The graph and all of its vertices are protected by the dgraph_lock mutex.
 *   See restarter.c for more information.
 *
 *   The properties of an instance fall into two classes: immediate and
 *   snapshotted.  Immediate properties should have an immediate effect when
 *   changed.  Snapshotted properties should be read from a snapshot, so they
 *   only change when the snapshot changes.  The immediate properties used by
 *   the graph engine are general/enabled, general/restarter, and the properties
 *   in the restarter_actions property group.  Since they are immediate, they
 *   are not read out of a snapshot.  The snapshotted properties used by the
 *   graph engine are those in the property groups with type "dependency" and
 *   are read out of the "running" snapshot.  The "running" snapshot is created
 *   by the the graph engine as soon as possible, and it is updated, along with
 *   in-core copies of the data (dependency information for the graph engine) on
 *   receipt of the refresh command from svcadm.  In addition, the graph engine
 *   updates the "start" snapshot from the "running" snapshot whenever a service
 *   comes online.
 *
 *   When a DISABLE event is requested by the administrator, svc.startd shutdown
 *   the dependents first before shutting down the requested service.
 *   In graph_enable_by_vertex, we create a subtree that contains the dependent
 *   vertices by marking those vertices with the GV_TOOFFLINE flag. And we mark
 *   the vertex to disable with the GV_TODISABLE flag. Once the tree is created,
 *   we send the _ADMIN_DISABLE event to the leaves. The leaves will then
 *   transition from STATE_ONLINE/STATE_DEGRADED to STATE_OFFLINE/STATE_MAINT.
 *   In gt_enter_offline and gt_enter_maint if the vertex was in a subtree then
 *   we clear the GV_TOOFFLINE flag and walk the dependencies to offline the new
 *   exposed leaves. We do the same until we reach the last leaf (the one with
 *   the GV_TODISABLE flag). If the vertex to disable is also part of a larger
 *   subtree (eg. multiple DISABLE events on vertices in the same subtree) then
 *   once the first vertex is disabled (GV_TODISABLE flag is removed), we
 *   continue to propagate the offline event to the vertex's dependencies.
 *
 *
 * SMF state transition notifications
 *
 *   When an instance of a service managed by SMF changes state, svc.startd may
 *   publish a GPEC sysevent. All transitions to or from maintenance, a
 *   transition cause by a hardware error will generate an event.
 *   Other transitions will generate an event if there exist notification
 *   parameter for that transition. Notification parameters are stored in the
 *   SMF repository for the service/instance they refer to. System-wide
 *   notification parameters are stored in the global instance.
 *   svc.startd can be told to send events for all SMF state transitions despite
 *   of notification parameters by setting options/info_events_all to true in
 *   restarter:default
 *
 *   The set of transitions that generate events is cached in the
 *   dgraph_vertex_t gv_stn_tset for service/instance and in the global
```

```
 127  *   stn_global for the system-wide set. They are re-read when instances are
 128  *   refreshed.
 129  *
 130  *   The GPEC events published by svc.startd are consumed by fmd(1M). After
 131  *   processing these events, fmd(1M) publishes the processed events to
 132  *   notification agents. The notification agents read the notification
 133  *   parameters from the SMF repository through libscf(3LIB) interfaces and send
 134  *   the notification, or not, based on those parameters.
 135  *
 136  *   Subscription and publishing to the GPEC channels is done with the
 137  *   libfmevent(3LIB) wrappers fmev_[r]publish_*() and
 138  *   fmev_shdl_(un)subscribe().
 139  *
 140  */

 142 #include <sys/uadmin.h>
 143 #include <sys/wait.h>

 145 #include <assert.h>
 146 #include <errno.h>
 147 #include <fcntl.h>
 148 #include <fm/libfmevent.h>
 149 #include <libscf.h>
 150 #include <libscf_priv.h>
 151 #include <librestart.h>
 152 #include <libuutil.h>
 153 #include <locale.h>
 154 #include <poll.h>
 155 #include <pthread.h>
 156 #include <signal.h>
 157 #include <stddef.h>
 158 #include <stdio.h>
 159 #include <stdlib.h>
 160 #include <string.h>
 161 #include <strings.h>
 162 #include <sys/statvfs.h>
 163 #include <sys/uadmin.h>
 164 #include <zone.h>
 165 #if defined(__x86)
 166 #include <libbe.h>
 167 #endif  /* __x86 */

 169 #include "startd.h"
 170 #include "protocol.h"


 173 #define MILESTONE_NONE  ((graph_vertex_t *)1)

 175 #define CONSOLE_LOGIN_FMRI      "svc:/system/console-login:default"
 176 #define FS_MINIMAL_FMRI         "svc:/system/filesystem/minimal:default"

 178 #define VERTEX_REMOVED  0       /* vertex has been freed  */
 179 #define VERTEX_INUSE    1       /* vertex is still in use */

 181 #define IS_ENABLED(v) ((v)->gv_flags & (GV_ENABLED | GV_ENBLD_NOOVR))

 183 /*
 184  * stn_global holds the tset for the system wide notification parameters.
 185  * It is updated on refresh of svc:/system/svc/global:default
 186  *
 187  * There are two assumptions that relax the need for a mutex:
 188  *     1. 32-bit value assignments are atomic
 189  *     2. Its value is consumed only in one point at
 190  *     dgraph_state_transition_notify(). There are no test and set races.
 191  *
 192  *     If either assumption is broken, we'll need a mutex to synchronize
```



```
 193  *     access to stn_global
 194  */
 195 int32_t stn_global;
 196 /*
 197  * info_events_all holds a flag to override notification parameters and send
 198  * Information events for all state transitions.
 199  * same about the need of a mutex here.
 200  */
 201 int info_events_all;

 203 /*
 204  * Services in these states are not considered 'down' by the
 205  * milestone/shutdown code.
 206  */
 207 #define up_state(state) ((state) == RESTARTER_STATE_ONLINE || \
 208         (state) == RESTARTER_STATE_DEGRADED || \
 209         (state) == RESTARTER_STATE_OFFLINE)

 211 #define is_depgrp_bypassed(v) ((v->gv_type == GVT_GROUP) && \
 212         ((v->gv_depgroup == DEPGRP_EXCLUDE_ALL) || \
 213         (v->gv_restart < RERR_RESTART)))

 215 #define is_inst_bypassed(v) ((v->gv_type == GVT_INST) && \
 216         ((v->gv_flags & GV_TODISABLE) || \
 217         (v->gv_flags & GV_TOOFFLINE)))

 219 static uu_list_pool_t *graph_edge_pool, *graph_vertex_pool;
 220 static uu_list_t *dgraph;
 221 static pthread_mutex_t dgraph_lock;

 223 /*
 224  * milestone indicates the current subgraph.  When NULL, it is the entire
 225  * graph.  When MILESTONE_NONE, it is the empty graph.  Otherwise, it is all
 226  * services on which the target vertex depends.
 227  */
 228 static graph_vertex_t *milestone = NULL;
 229 static boolean_t initial_milestone_set = B_FALSE;
 230 static pthread_cond_t initial_milestone_cv = PTHREAD_COND_INITIALIZER;

 232 /* protected by dgraph_lock */
 233 static boolean_t sulogin_thread_running = B_FALSE;
 234 static boolean_t sulogin_running = B_FALSE;
 235 static boolean_t console_login_ready = B_FALSE;

 237 /* Number of services to come down to complete milestone transition. */
 238 static uint_t non_subgraph_svcs;

 240 /*
 241  * These variables indicate what should be done when we reach the milestone
 242  * target milestone, i.e., when non_subgraph_svcs == 0.  They are acted upon in
 243  * dgraph_set_instance_state().
 244  */
 245 static int halting = -1;
 246 static boolean_t go_single_user_mode = B_FALSE;
 247 static boolean_t go_to_level1 = B_FALSE;

 249 /*
 250  * Tracks when we started halting.
 251  */
 252 static time_t halting_time = 0;

 254 /*
 255  * This tracks the legacy runlevel to ensure we signal init and manage
 256  * utmpx entries correctly.
 257  */
 258 static char current_runlevel = '\0';
```

```
 260 /* Number of single user threads currently running */
 261 static pthread_mutex_t single_user_thread_lock;
 262 static int single_user_thread_count = 0;

 264 /* Statistics for dependency cycle-checking */
 265 static u_longlong_t dep_inserts = 0;
 266 static u_longlong_t dep_cycle_ns = 0;
 267 static u_longlong_t dep_insert_ns = 0;


 270 static const char * const emsg_invalid_restarter =
 271         "Transitioning %s to maintenance, restarter FMRI %s is invalid "
 272         "(see 'svcs -xv' for details).\n";
 273 static const char * const console_login_fmri = CONSOLE_LOGIN_FMRI;
 274 static const char * const single_user_fmri = SCF_MILESTONE_SINGLE_USER;
 275 static const char * const multi_user_fmri = SCF_MILESTONE_MULTI_USER;
 276 static const char * const multi_user_svr_fmri = SCF_MILESTONE_MULTI_USER_SERVER;


 279 /*
 280  * These services define the system being "up".  If none of them can come
 281  * online, then we will run sulogin on the console.  Note that the install ones
 282  * are for the miniroot and when installing CDs after the first.  can_come_up()
 283  * does the decision making, and an sulogin_thread() runs sulogin, which can be
 284  * started by dgraph_set_instance_state() or single_user_thread().
 285  *
 286  * NOTE: can_come_up() relies on SCF_MILESTONE_SINGLE_USER being the first
 287  * entry, which is only used when booting_to_single_user (boot -s) is set.
 288  * This is because when doing a "boot -s", sulogin is started from specials.c
 289  * after milestone/single-user comes online, for backwards compatibility.
 290  * In this case, SCF_MILESTONE_SINGLE_USER needs to be part of up_svcs
 291  * to ensure sulogin will be spawned if milestone/single-user cannot be reached.
 292  */
 293 static const char * const up_svcs[] = {
 294         SCF_MILESTONE_SINGLE_USER,
 295         CONSOLE_LOGIN_FMRI,
 296         "svc:/system/install-setup:default",
 297         "svc:/system/install:default",
 298         NULL
 299 };
_____unchanged_portion_omitted_

3845 /*
3846  * The sulogin thread runs sulogin while can_come_up() is false.  run_sulogin()
3847  * keeps sulogin from stepping on console-login's toes.
3848  */
3849 /* ARGSUSED */
3850 static void *
3851 sulogin_thread(void *unused)
3852 {
3853         (void) pthread_setname_np(pthread_self(), "sulogin");

3855         MUTEX_LOCK(&dgraph_lock);

3857         assert(sulogin_thread_running);

3859         do {
3860                 (void) run_sulogin("Console login service(s) cannot run\n");
3861         } while (!can_come_up());

3863         sulogin_thread_running = B_FALSE;
3864         MUTEX_UNLOCK(&dgraph_lock);

3866         return (NULL);
3867 }
```



```
3869 /* ARGSUSED */
3870 void *
3871 single_user_thread(void *unused)
3872 {
3873         uint_t left;
3874         scf_handle_t *h;
3875         scf_instance_t *inst;
3876         scf_property_t *prop;
3877         scf_value_t *val;
3878         const char *msg;
3879         char *buf;
3880         int r;

3882         (void) pthread_setname_np(pthread_self(), "single_user");

3884         MUTEX_LOCK(&single_user_thread_lock);
3885         single_user_thread_count++;

3887         if (!booting_to_single_user)
3888                 kill_user_procs();

3890         if (go_single_user_mode || booting_to_single_user) {
3891                 msg = "SINGLE USER MODE\n";
3892         } else {
3893                 assert(go_to_level1);

3895                 fork_rc_script('1', "start", B_TRUE);

3897                 uu_warn("The system is ready for administration.\n");

3899                 msg = "";
3900         }

3902         MUTEX_UNLOCK(&single_user_thread_lock);

3904         for (;;) {
3905                 MUTEX_LOCK(&dgraph_lock);
3906                 r = run_sulogin(msg);
3907                 MUTEX_UNLOCK(&dgraph_lock);
3908                 if (r == 0)
3909                         break;

3911                 assert(r == EALREADY || r == EBUSY);

3913                 left = 3;
3914                 while (left > 0)
3915                         left = sleep(left);
3916         }

3918         MUTEX_LOCK(&single_user_thread_lock);

3920         /*
3921          * If another single user thread has started, let it finish changing
3922          * the run level.
3923          */
3924         if (single_user_thread_count > 1) {
3925                 single_user_thread_count--;
3926                 MUTEX_UNLOCK(&single_user_thread_lock);
3927                 return (NULL);
3928         }

3930         h = libscf_handle_create_bound_loop();
3931         inst = scf_instance_create(h);
3932         prop = safe_scf_property_create(h);
3933         val = safe_scf_value_create(h);
```

```
3934         buf = startd_alloc(max_scf_fmri_size);

3936 lookup:
3937         if (scf_handle_decode_fmri(h, SCF_SERVICE_STARTD, NULL, NULL, inst,
3938             NULL, NULL, SCF_DECODE_FMRI_EXACT) != 0) {
3939                 switch (scf_error()) {
3940                 case SCF_ERROR_NOT_FOUND:
3941                         r = libscf_create_self(h);
3942                         if (r == 0)
3943                                 goto lookup;
3944                         assert(r == ECONNABORTED);
3945                         /* FALLTHROUGH */

3947                 case SCF_ERROR_CONNECTION_BROKEN:
3948                         libscf_handle_rebind(h);
3949                         goto lookup;

3951                 case SCF_ERROR_INVALID_ARGUMENT:
3952                 case SCF_ERROR_CONSTRAINT_VIOLATED:
3953                 case SCF_ERROR_NOT_BOUND:
3954                 case SCF_ERROR_HANDLE_MISMATCH:
3955                 default:
3956                         bad_error("scf_handle_decode_fmri", scf_error());
3957                 }
3958         }

3960         MUTEX_LOCK(&dgraph_lock);

3962         r = scf_instance_delete_prop(inst, SCF_PG_OPTIONS_OVR,
3963             SCF_PROPERTY_MILESTONE);
3964         switch (r) {
3965         case 0:
3966         case ECANCELED:
3967                 break;

3969         case ECONNABORTED:
3970                 MUTEX_UNLOCK(&dgraph_lock);
3971                 libscf_handle_rebind(h);
3972                 goto lookup;

3974         case EPERM:
3975         case EACCES:
3976         case EROFS:
3977                 log_error(LOG_WARNING, "Could not clear temporary milestone: "
3978                     "%s.\n", strerror(r));
3979                 break;

3981         default:
3982                 bad_error("scf_instance_delete_prop", r);
3983         }

3985         MUTEX_UNLOCK(&dgraph_lock);

3987         r = libscf_get_milestone(inst, prop, val, buf, max_scf_fmri_size);
3988         switch (r) {
3989         case ECANCELED:
3990         case ENOENT:
3991         case EINVAL:
3992                 (void) strcpy(buf, "all");
3993                 /* FALLTHROUGH */

3995         case 0:
3996                 uu_warn("Returning to milestone %s.\n", buf);
3997                 break;

3999         case ECONNABORTED:
```



```
4000                 libscf_handle_rebind(h);
4001                 goto lookup;

4003         default:
4004                 bad_error("libscf_get_milestone", r);
4005         }

4007         r = dgraph_set_milestone(buf, h, B_FALSE);
4008         switch (r) {
4009         case 0:
4010         case ECONNRESET:
4011         case EALREADY:
4012         case EINVAL:
4013         case ENOENT:
4014                 break;

4016         default:
4017                 bad_error("dgraph_set_milestone", r);
4018         }

4020         /*
4021          * See graph_runlevel_changed().
4022          */
4023         MUTEX_LOCK(&dgraph_lock);
4024         utmpx_set_runlevel(target_milestone_as_runlevel(), 'S', B_TRUE);
4025         MUTEX_UNLOCK(&dgraph_lock);

4027         startd_free(buf, max_scf_fmri_size);
4028         scf_value_destroy(val);
4029         scf_property_destroy(prop);
4030         scf_instance_destroy(inst);
4031         scf_handle_destroy(h);

4033         /*
4034          * We'll give ourselves 3 seconds to respond to all of the enablings
4035          * that setting the milestone should have created before checking
4036          * whether to run sulogin.
4037          */
4038         left = 3;
4039         while (left > 0)
4040                 left = sleep(left);

4042         MUTEX_LOCK(&dgraph_lock);
4043         /*
4044          * Clearing these variables will allow the sulogin thread to run.  We
4045          * check here in case there aren't any more state updates anytime soon.
4046          */
4047         go_to_level1 = go_single_user_mode = booting_to_single_user = B_FALSE;
4048         if (!sulogin_thread_running && !can_come_up()) {
4049                 (void) startd_thread_create(sulogin_thread, NULL);
4050                 sulogin_thread_running = B_TRUE;
4051         }
4052         MUTEX_UNLOCK(&dgraph_lock);
4053         single_user_thread_count--;
4054         MUTEX_UNLOCK(&single_user_thread_lock);
4055         return (NULL);
4056 }
______unchanged_portion_omitted_

5776 /*
5777  * graph_event_thread()
5778  *    Wait for state changes from the restarters.
5779  */
5780 /*ARGSUSED*/
5781 void *
5782 graph_event_thread(void *unused)
```

```
5783 {
5784         scf_handle_t *h;
5785         int err;

5787         (void) pthread_setname_np(pthread_self(), "graph_event");

5789         h = libscf_handle_create_bound_loop();

5791         /*CONSTCOND*/
5792         while (1) {
5793                 graph_protocol_event_t *e;

5795                 MUTEX_LOCK(&gu->gu_lock);

5797                 while (gu->gu_wakeup == 0)
5798                         (void) pthread_cond_wait(&gu->gu_cv, &gu->gu_lock);

5800                 gu->gu_wakeup = 0;

5802                 while ((e = graph_event_dequeue()) != NULL) {
5803                         MUTEX_LOCK(&e->gpe_lock);
5804                         MUTEX_UNLOCK(&gu->gu_lock);

5806                         while ((err = handle_graph_update_event(h, e)) ==
5807                             ECONNABORTED)
5808                                 libscf_handle_rebind(h);

5810                         if (err == 0)
5811                                 graph_event_release(e);
5812                         else
5813                                 graph_event_requeue(e);

5815                         MUTEX_LOCK(&gu->gu_lock);
5816                 }

5818                 MUTEX_UNLOCK(&gu->gu_lock);
5819         }

5821         /*
5822          * Unreachable for now -- there's currently no graceful cleanup
5823          * called on exit().
5824          */
5825         MUTEX_UNLOCK(&gu->gu_lock);
5826         scf_handle_destroy(h);
5827         return (NULL);
5828 }
```

*unchanged_portion_omitted_*

```
6136 /*
6137  * void *graph_thread(void *)
6138  *
6139  * Graph management thread.
6140  */
6141 /*ARGSUSED*/
6142 void *
6143 graph_thread(void *arg)
6144 {
6145         scf_handle_t *h;
6146         int err;

6148         (void) pthread_setname_np(pthread_self(), "graph");

6150         h = libscf_handle_create_bound_loop();

6152         if (st->st_initial)
6153                 set_initial_milestone(h);
```



```
6155         MUTEX_LOCK(&dgraph_lock);
6156         initial_milestone_set = B_TRUE;
6157         err = pthread_cond_broadcast(&initial_milestone_cv);
6158         assert(err == 0);
6159         MUTEX_UNLOCK(&dgraph_lock);

6161         libscf_populate_graph(h);

6163         if (!st->st_initial)
6164                 set_restart_milestone(h);

6166         MUTEX_LOCK(&st->st_load_lock);
6167         st->st_load_complete = 1;
6168         (void) pthread_cond_broadcast(&st->st_load_cv);
6169         MUTEX_UNLOCK(&st->st_load_lock);

6171         MUTEX_LOCK(&dgraph_lock);
6172         /*
6173          * Now that we've set st_load_complete we need to check can_come_up()
6174          * since if we booted to a milestone, then there won't be any more
6175          * state updates.
6176          */
6177         if (!go_single_user_mode && !go_to_level1 &&
6178             halting == -1) {
6179                 if (!sulogin_thread_running && !can_come_up()) {
6180                         (void) startd_thread_create(sulogin_thread, NULL);
6181                         sulogin_thread_running = B_TRUE;
6182                 }
6183         }
6184         MUTEX_UNLOCK(&dgraph_lock);

6186         (void) pthread_mutex_lock(&gu->gu_freeze_lock);

6188         /*CONSTCOND*/
6189         while (1) {
6190                 (void) pthread_cond_wait(&gu->gu_freeze_cv,
6191                     &gu->gu_freeze_lock);
6192         }

6194         /*
6195          * Unreachable for now -- there's currently no graceful cleanup
6196          * called on exit().
6197          */
6198         (void) pthread_mutex_unlock(&gu->gu_freeze_lock);
6199         scf_handle_destroy(h);

6201         return (NULL);
6202 }
```

*unchanged_portion_omitted_*

```
6795 /*ARGSUSED*/
6796 void *
6797 repository_event_thread(void *unused)
6798 {
6799         scf_handle_t *h;
6800         scf_propertygroup_t *pg;
6801         scf_instance_t *inst;
6802         char *fmri = startd_alloc(max_scf_fmri_size);
6803         char *pg_name = startd_alloc(max_scf_value_size);
6804         int r;

6806         (void) pthread_setname_np(pthread_self(), "repository_event");

6808         h = libscf_handle_create_bound_loop();
```

```
6810         pg = safe_scf_pg_create(h);
6811         inst = safe_scf_instance_create(h);

6813 retry:
6814         if (_scf_notify_add_pgtype(h, SCF_GROUP_FRAMEWORK) != SCF_SUCCESS) {
6815                 if (scf_error() == SCF_ERROR_CONNECTION_BROKEN) {
6816                         libscf_handle_rebind(h);
6817                 } else {
6818                         log_error(LOG_WARNING,
6819                             "Couldn't set up repository notification "
6820                             "for property group type %s: %s\n",
6821                             SCF_GROUP_FRAMEWORK, scf_strerror(scf_error()));

6823                         (void) sleep(1);
6824                 }

6826                 goto retry;
6827         }

6829         /*CONSTCOND*/
6830         while (1) {
6831                 ssize_t res;

6833                 /* Note: fmri is only set on delete events. */
6834                 res = _scf_notify_wait(pg, fmri, max_scf_fmri_size);
6835                 if (res < 0) {
6836                         libscf_handle_rebind(h);
6837                         goto retry;
6838                 } else if (res == 0) {
6839                         /*
6840                          * property group modified.  inst and pg_name are
6841                          * pre-allocated scratch space.
6842                          */
6843                         if (scf_pg_update(pg) < 0) {
6844                                 switch (scf_error()) {
6845                                 case SCF_ERROR_DELETED:
6846                                         continue;

6848                                 case SCF_ERROR_CONNECTION_BROKEN:
6849                                         log_error(LOG_WARNING,
6850                                             "Lost repository event due to "
6851                                             "disconnection.\n");
6852                                         libscf_handle_rebind(h);
6853                                         goto retry;

6855                                 case SCF_ERROR_NOT_BOUND:
6856                                 case SCF_ERROR_NOT_SET:
6857                                 default:
6858                                         bad_error("scf_pg_update", scf_error());
6859                                 }
6860                         }

6862                         r = process_pg_event(h, pg, inst, pg_name);
6863                         switch (r) {
6864                         case 0:
6865                                 break;

6867                         case ECONNABORTED:
6868                                 log_error(LOG_WARNING, "Lost repository event "
6869                                     "due to disconnection.\n");
6870                                 libscf_handle_rebind(h);
6871                                 /* FALLTHROUGH */

6873                         case ECONNRESET:
6874                                 goto retry;
```



```
6876                         default:
6877                                 bad_error("process_pg_event", r);
6878                         }
6879                 } else {
6880                         /*
6881                          * Service, instance, or pg deleted.
6882                          * Don't trust fmri on return.
6883                          */
6884                         process_delete(fmri, h);
6885                 }
6886         }

6888         /*NOTREACHED*/
6889         return (NULL);
6890 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   33715 Mon Oct 15 13:25:12 2018
new/usr/src/cmd/svc/startd/method.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright 2011 Joyent Inc.
 */

/*
 * method.c - method execution functions
 *
 * This file contains the routines needed to run a method:  a fork(2)-exec(2)
 * invocation monitored using either the contract filesystem or waitpid(2).
 * (Plain fork1(2) support is provided in fork.c.)
 *
 * Contract Transfer
 *   When we restart a service, we want to transfer any contracts that the old
 *   service's contract inherited.  This means that (a) we must not abandon the
 *   old contract when the service dies and (b) we must write the id of the old
 *   contract into the terms of the new contract.  There should be limits to
 *   (a), though, since we don't want to keep the contract around forever.  To
 *   this end we'll say that services in the offline state may have a contract
 *   to be transfered and services in the disabled or maintenance states cannot.
 *   This means that when a service transitions from online (or degraded) to
 *   offline, the contract should be preserved, and when the service transitions
 *   from offline to online (i.e., the start method), we'll transfer inherited
 *   contracts.
 */

#include <sys/contract/process.h>
#include <sys/ctfs.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <sys/wait.h>
#include <alloca.h>
#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <libcontract.h>
```



```
#include <libcontract_priv.h>
#include <libgen.h>
#include <librestart.h>
#include <libscf.h>
#include <limits.h>
#include <port.h>
#include <sac.h>
#include <signal.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <unistd.h>
#include <atomic.h>
#include <poll.h>
#include <libscf_priv.h>

#include "startd.h"

#define SBIN_SH         "/sbin/sh"

/*
 * Used to tell if contracts are in the process of being
 * stored into the svc.startd internal hash table.
 */
volatile uint16_t       storing_contract = 0;

/*
 * Mapping from restart_on method-type to contract events.  Must correspond to
 * enum method_restart_t.
 */
static uint_t method_events[] = {
        /* METHOD_RESTART_ALL */
        CT_PR_EV_HWERR | CT_PR_EV_SIGNAL | CT_PR_EV_CORE | CT_PR_EV_EMPTY,
        /* METHOD_RESTART_EXTERNAL_FAULT */
        CT_PR_EV_HWERR | CT_PR_EV_SIGNAL,
        /* METHOD_RESTART_ANY_FAULT */
        CT_PR_EV_HWERR | CT_PR_EV_SIGNAL | CT_PR_EV_CORE
};
```

*unchanged_portion_omitted_*

```
/*
 * The method thread executes a service method to effect a state transition.
 * The next_state of info->sf_id should be non-_NONE on entrance, and it will
 * be _NONE on exit (state will either be what next_state was (on success), or
 * it will be _MAINT (on error)).
 *
 * There are six classes of methods to consider: start & other (stop, refresh)
 * for each of "normal" services, wait services, and transient services.  For
 * each, the method must be fetched from the repository & executed.  fork()ed
 * methods must be waited on, except for the start method of wait services
 * (which must be registered with the wait subsystem via wait_register()).  If
 * the method succeeded (returned 0), then for start methods its contract
 * should be recorded as the primary contract for the service.  For other
 * methods, it should be abandoned.  If the method fails, then depending on
 * the failure, either the method should be reexecuted or the service should
 * be put into maintenance.  Either way the contract should be abandoned.
 */
void *
method_thread(void *arg)
{
        fork_info_t *info = arg;
        restarter_inst_t *inst;
        scf_handle_t    *local_handle;
        scf_instance_t  *s_inst = NULL;
        int r, exit_code;
        boolean_t retryable;
```

```
1116         restarter_str_t reason;
1118         (void) pthread_setname_np(pthread_self(), "method");
1120         assert(0 <= info->sf_method_type && info->sf_method_type <= 2);
1122         /* Get (and lock) the restarter_inst_t. */
1123         inst = inst_lookup_by_id(info->sf_id);
1125         assert(inst->ri_method_thread != 0);
1126         assert(instance_in_transition(inst) == 1);
1128         /*
1129          * We cannot leave this function with inst in transition, because
1130          * protocol.c withholds messages for inst otherwise.
1131          */
1133         log_framework(LOG_DEBUG, "method_thread() running %s method for %s.\n",
1134             method_names[info->sf_method_type], inst->ri_i.i_fmri);
1136         local_handle = libscf_handle_create_bound_loop();
1138 rebind_retry:
1139         /* get scf_instance_t */
1140         switch (r = libscf_fmri_get_instance(local_handle, inst->ri_i.i_fmri,
1141             &s_inst)) {
1142         case 0:
1143                 break;
1145         case ECONNABORTED:
1146                 libscf_handle_rebind(local_handle);
1147                 goto rebind_retry;
1149         case ENOENT:
1150                 /*
1151                  * It's not there, but we need to call this so protocol.c
1152                  * doesn't think it's in transition anymore.
1153                  */
1154                 (void) restarter_instance_update_states(local_handle, inst,
1155                     inst->ri_i.i_state, RESTARTER_STATE_NONE, RERR_NONE,
1156                     restarter_str_none);
1157                 goto out;
1159         case EINVAL:
1160         case ENOTSUP:
1161         default:
1162                 bad_error("libscf_fmri_get_instance", r);
1163         }
1165         inst->ri_m_inst = s_inst;
1166         inst->ri_mi_deleted = B_FALSE;
1168 retry:
1169         if (info->sf_method_type == METHOD_START)
1170                 log_transition(inst, START_REQUESTED);
1172         r = method_run(&inst, info->sf_method_type, &exit_code);
1174         if (r == 0 && exit_code == 0) {
1175                 /* Success! */
1176                 assert(inst->ri_i.i_next_state != RESTARTER_STATE_NONE);
1178                 /*
1179                  * When a stop method succeeds, remove the primary contract of
1180                  * the service, unless we're going to offline, in which case
1181                  * retain the contract so we can transfer inherited contracts to
```



```
1182                  * the replacement service.
1183                  */
1185                 if (info->sf_method_type == METHOD_STOP &&
1186                     inst->ri_i.i_primary_ctid != 0) {
1187                         if (inst->ri_i.i_next_state == RESTARTER_STATE_OFFLINE)
1188                                 inst->ri_i.i_primary_ctid_stopped = 1;
1189                         else
1190                                 method_remove_contract(inst, B_TRUE, B_TRUE);
1191                 }
1192                 /*
1193                  * We don't care whether the handle was rebound because this is
1194                  * the last thing we do with it.
1195                  */
1196                 (void) restarter_instance_update_states(local_handle, inst,
1197                     inst->ri_i.i_next_state, RESTARTER_STATE_NONE,
1198                     info->sf_event_type, info->sf_reason);
1200                 (void) update_fault_count(inst, FAULT_COUNT_RESET);
1202                 goto out;
1203         }
1205         /* Failure.  Retry or go to maintenance. */
1207         if (r != 0 && r != EAGAIN) {
1208                 retryable = B_FALSE;
1209         } else {
1210                 switch (exit_code) {
1211                 case SMF_EXIT_ERR_CONFIG:
1212                 case SMF_EXIT_ERR_NOSMF:
1213                 case SMF_EXIT_ERR_PERM:
1214                 case SMF_EXIT_ERR_FATAL:
1215                         retryable = B_FALSE;
1216                         break;
1218                 default:
1219                         retryable = B_TRUE;
1220                 }
1221         }
1223         if (retryable && update_fault_count(inst, FAULT_COUNT_INCR) != 1)
1224                 goto retry;
1226         /* maintenance */
1227         if (r == ELOOP)
1228                 log_transition(inst, START_FAILED_REPEATEDLY);
1229         else if (r == ERANGE)
1230                 log_transition(inst, START_FAILED_TIMEOUT_FATAL);
1231         else if (exit_code == SMF_EXIT_ERR_CONFIG)
1232                 log_transition(inst, START_FAILED_CONFIGURATION);
1233         else if (exit_code == SMF_EXIT_ERR_FATAL)
1234                 log_transition(inst, START_FAILED_FATAL);
1235         else
1236                 log_transition(inst, START_FAILED_OTHER);
1238         if (r == ELOOP) {
1239                 reason = restarter_str_restarting_too_quickly;
1240         } else if (retryable) {
1241                 reason = restarter_str_fault_threshold_reached;
1242         } else {
1243                 reason = restarter_str_method_failed;
1244         }
1246         (void) restarter_instance_update_states(local_handle, inst,
1247             RESTARTER_STATE_MAINT, RESTARTER_STATE_NONE, RERR_FAULT,
```

```
1248             reason);

1250         if (!method_is_transient(inst, info->sf_method_type) &&
1251             inst->ri_i.i_primary_ctid != 0)
1252                 method_remove_contract(inst, B_TRUE, B_TRUE);

1254 out:
1255         inst->ri_method_thread = 0;

1257         /*
1258          * Unlock the mutex after broadcasting to avoid a race condition
1259          * with restarter_delete_inst() when the 'inst' structure is freed.
1260          */
1261         (void) pthread_cond_broadcast(&inst->ri_method_cv);
1262         MUTEX_UNLOCK(&inst->ri_lock);

1264         scf_instance_destroy(s_inst);
1265         scf_handle_destroy(local_handle);
1266         startd_free(info, sizeof (fork_info_t));
1267         return (NULL);
1268 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   65598 Mon Oct 15 13:25:12 2018
new/usr/src/cmd/svc/startd/restarter.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 */

/*
 * restarter.c - service manipulation
 *
 * This component manages services whose restarter is svc.startd, the standard
 * restarter.  It translates restarter protocol events from the graph engine
 * into actions on processes, as a delegated restarter would do.
 *
 * The master restarter manages a number of always-running threads:
 *   - restarter event thread: events from the graph engine
 *   - timeout thread: thread to fire queued timeouts
 *   - contract thread: thread to handle contract events
 *   - wait thread: thread to handle wait-based services
 *
 * The other threads are created as-needed:
 *   - per-instance method threads
 *   - per-instance event processing threads
 *
 * The interaction of all threads must result in the following conditions
 * being satisfied (on a per-instance basis):
 *   - restarter events must be processed in order
 *   - method execution must be serialized
 *   - instance delete must be held until outstanding methods are complete
 *   - contract events shouldn't be processed while a method is running
 *   - timeouts should fire even when a method is running
 *
 * Service instances are represented by restarter_inst_t's and are kept in the
 * instance_list list.
 *
 * Service States
 *   The current state of a service instance is kept in
 *   restarter_inst_t->ri_i.i_state.  If transition to a new state could take
 *   some time, then before we effect the transition we set
 *   restarter_inst_t->ri_i.i_next_state to the target state, and afterwards we
```



```
 *   rotate i_next_state to i_state and set i_next_state to
 *   RESTARTER_STATE_NONE.  So usually i_next_state is _NONE when ri_lock is not
 *   held.  The exception is when we launch methods, which are done with
 *   a separate thread.  To keep any other threads from grabbing ri_lock before
 *   method_thread() does, we set ri_method_thread to the thread id of the
 *   method thread, and when it is nonzero any thread with a different thread id
 *   waits on ri_method_cv.
 *
 * Method execution is serialized by blocking on ri_method_cv in
 * inst_lookup_by_id() and waiting for a 0 value of ri_method_thread.  This
 * also prevents the instance structure from being deleted until all
 * outstanding operations such as method_thread() have finished.
 *
 * Lock ordering:
 *
 * dgraph_lock [can be held when taking:]
 *   utmpx_lock
 *   dictionary->dict_lock
 *   st->st_load_lock
 *   wait_info_lock
 *   ru->restarter_update_lock
 *     restarter_queue->rpeq_lock
 *   instance_list.ril_lock
 *     inst->ri_lock
 *   st->st_configd_live_lock
 *
 * instance_list.ril_lock
 *   graph_queue->gpeq_lock
 *   gu->gu_lock
 *   st->st_configd_live_lock
 *   dictionary->dict_lock
 *   inst->ri_lock
 *     graph_queue->gpeq_lock
 *     gu->gu_lock
 *     tu->tu_lock
 *     tq->tq_lock
 *     inst->ri_queue_lock
 *       wait_info_lock
 *       bp->cb_lock
 *     utmpx_lock
 *
 * single_user_thread_lock
 *   wait_info_lock
 *   utmpx_lock
 *
 * gu_freeze_lock
 *
 * logbuf_mutex nests inside pretty much everything.
 */

#include <sys/contract/process.h>
#include <sys/ctfs.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <sys/wait.h>
#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <libcontract.h>
#include <libcontract_priv.h>
#include <libintl.h>
#include <librestart.h>
#include <librestart_priv.h>
#include <libuutil.h>
```

```
 126 #include <limits.h>
 127 #include <poll.h>
 128 #include <port.h>
 129 #include <pthread.h>
 130 #include <stdarg.h>
 131 #include <stdio.h>
 132 #include <strings.h>
 133 #include <unistd.h>

 135 #include "startd.h"
 136 #include "protocol.h"

 138 static uu_list_pool_t *restarter_instance_pool;
 139 static restarter_instance_list_t instance_list;

 141 static uu_list_pool_t *restarter_queue_pool;

 143 #define WT_SVC_ERR_THROTTLE     1       /* 1 sec delay for erroring wait svc */

 145 /*
 146  * Function used to reset the restart times for an instance, when
 147  * an administrative task comes along and essentially makes the times
 148  * in this array ineffective.
 149  */
 150 static void
 151 reset_start_times(restarter_inst_t *inst)
 152 {
 153         inst->ri_start_index = 0;
 154         bzero(inst->ri_start_time, sizeof (inst->ri_start_time));
 155 }
```

*unchanged_portion_omitted_*

```
 990 /* ARGSUSED */
 991 void *
 992 restarter_post_fsminimal_thread(void *unused)
 993 {
 994         scf_handle_t *h;
 995         int r;

 997         (void) pthread_setname_np(pthread_self(), "restarter_post_fsmin");

 999         h = libscf_handle_create_bound_loop();

1001         for (;;) {
1002                 r = libscf_create_self(h);
1003                 if (r == 0)
1004                         break;

1006                 assert(r == ECONNABORTED);
1007                 libscf_handle_rebind(h);
1008         }

1010         restarter_take_pending_snapshots(h);

1012         (void) scf_handle_unbind(h);
1013         scf_handle_destroy(h);

1015         return (NULL);
1016 }
```

*unchanged_portion_omitted_*

```
1759 /*
1760  * void *restarter_process_events()
1761  *
1762  *   Called in a separate thread to process the events on an instance's
1763  *   queue.  Empties the queue completely, and tries to keep the thread
```



```
1764  *   around for a little while after the queue is empty to save on
1765  *   startup costs.
1766  */
1767 static void *
1768 restarter_process_events(void *arg)
1769 {
1770         scf_handle_t *h;
1771         restarter_instance_qentry_t *event;
1772         restarter_inst_t *rip;
1773         char *fmri = (char *)arg;
1774         struct timespec to;

1776         (void) pthread_setname_np(pthread_self(), "restarter_process_events");

1778         assert(fmri != NULL);

1780         h = libscf_handle_create_bound_loop();

1782         /* grab the queue lock */
1783         rip = inst_lookup_queue(fmri);
1784         if (rip == NULL)
1785                 goto out;

1787 again:

1789         while ((event = uu_list_first(rip->ri_queue)) != NULL) {
1790                 restarter_inst_t *inst;

1792                 /* drop the queue lock */
1793                 MUTEX_UNLOCK(&rip->ri_queue_lock);

1795                 /*
1796                  * Grab the inst lock -- this waits until any outstanding
1797                  * method finishes running.
1798                  */
1799                 inst = inst_lookup_by_name(fmri);
1800                 if (inst == NULL) {
1801                         /* Getting deleted in the middle isn't an error. */
1802                         goto cont;
1803                 }

1805                 assert(instance_in_transition(inst) == 0);

1807                 /* process the event */
1808                 switch (event->riq_type) {
1809                 case RESTARTER_EVENT_TYPE_ENABLE:
1810                 case RESTARTER_EVENT_TYPE_DISABLE:
1811                         (void) enable_inst(h, inst, event);
1812                         break;

1814                 case RESTARTER_EVENT_TYPE_ADMIN_DISABLE:
1815                         if (enable_inst(h, inst, event) == 0)
1816                                 reset_start_times(inst);
1817                         break;

1819                 case RESTARTER_EVENT_TYPE_REMOVE_INSTANCE:
1820                         restarter_delete_inst(inst);
1821                         inst = NULL;
1822                         goto cont;

1824                 case RESTARTER_EVENT_TYPE_STOP_RESET:
1825                         reset_start_times(inst);
1826                         /* FALLTHROUGH */
1827                 case RESTARTER_EVENT_TYPE_STOP:
1828                         (void) stop_instance(h, inst, RSTOP_DEPENDENCY);
1829                         break;
```

```
1831                case RESTARTER_EVENT_TYPE_START:
1832                        start_instance(h, inst, event->riq_reason);
1833                        break;

1835                case RESTARTER_EVENT_TYPE_DEPENDENCY_CYCLE:
1836                        maintain_instance(h, inst, 0,
1837                            restarter_str_dependency_cycle);
1838                        break;

1840                case RESTARTER_EVENT_TYPE_INVALID_DEPENDENCY:
1841                        maintain_instance(h, inst, 0,
1842                            restarter_str_invalid_dependency);
1843                        break;

1845                case RESTARTER_EVENT_TYPE_ADMIN_MAINT_ON:
1846                        if (event_from_tty(h, inst) == 0)
1847                                maintain_instance(h, inst, 0,
1848                                    restarter_str_service_request);
1849                        else
1850                                maintain_instance(h, inst, 0,
1851                                    restarter_str_administrative_request);
1852                        break;

1854                case RESTARTER_EVENT_TYPE_ADMIN_MAINT_ON_IMMEDIATE:
1855                        if (event_from_tty(h, inst) == 0)
1856                                maintain_instance(h, inst, 1,
1857                                    restarter_str_service_request);
1858                        else
1859                                maintain_instance(h, inst, 1,
1860                                    restarter_str_administrative_request);
1861                        break;

1863                case RESTARTER_EVENT_TYPE_ADMIN_MAINT_OFF:
1864                        unmaintain_instance(h, inst, RUNMAINT_CLEAR);
1865                        reset_start_times(inst);
1866                        break;

1868                case RESTARTER_EVENT_TYPE_ADMIN_REFRESH:
1869                        refresh_instance(h, inst);
1870                        break;

1872                case RESTARTER_EVENT_TYPE_ADMIN_DEGRADED:
1873                        log_framework(LOG_WARNING, "Restarter: "
1874                            "%s command (for %s) unimplemented.\n",
1875                            event_names[event->riq_type], inst->ri_i.i_fmri);
1876                        break;

1878                case RESTARTER_EVENT_TYPE_ADMIN_RESTART:
1879                        if (!instance_started(inst)) {
1880                                log_framework(LOG_DEBUG, "Restarter: "
1881                                    "Not restarting %s; not running.\n",
1882                                    inst->ri_i.i_fmri);
1883                        } else {
1884                                /*
1885                                 * Stop the instance.  If it can be restarted,
1886                                 * the graph engine will send a new event.
1887                                 */
1888                                if (restart_dump(h, inst)) {
1889                                        (void) contract_kill(
1890                                            inst->ri_i.i_primary_ctid, SIGABRT,
1891                                            inst->ri_i.i_fmri);
1892                                } else if (stop_instance(h, inst,
1893                                    RSTOP_RESTART) == 0) {
1894                                        reset_start_times(inst);
1895                                }
```



```
1896                        }
1897                        break;

1899                case RESTARTER_EVENT_TYPE_ADD_INSTANCE:
1900                default:
1901 #ifndef NDEBUG
1902                        uu_warn("%s:%d: Bad restarter event %d.  "
1903                            "Aborting.\n", __FILE__, __LINE__, event->riq_type);
1904 #endif
1905                        abort();
1906                }

1908                assert(inst != NULL);
1909                MUTEX_UNLOCK(&inst->ri_lock);

1911 cont:
1912                /* grab the queue lock */
1913                rip = inst_lookup_queue(fmri);
1914                if (rip == NULL)
1915                        goto out;

1917                /* delete the event */
1918                uu_list_remove(rip->ri_queue, event);
1919                startd_free(event, sizeof (restarter_instance_qentry_t));
1920        }

1922        assert(rip != NULL);

1924        /*
1925         * Try to preserve the thread for a little while for future use.
1926         */
1927        to.tv_sec = 3;
1928        to.tv_nsec = 0;
1929        (void) pthread_cond_reltimedwait_np(&rip->ri_queue_cv,
1930            &rip->ri_queue_lock, &to);

1932        if (uu_list_first(rip->ri_queue) != NULL)
1933                goto again;

1935        rip->ri_queue_thread = 0;
1936        MUTEX_UNLOCK(&rip->ri_queue_lock);

1938 out:
1939        (void) scf_handle_unbind(h);
1940        scf_handle_destroy(h);
1941        free(fmri);
1942        return (NULL);
1943 }

1945 static int
1946 is_admin_event(restarter_event_type_t t)
1947 {
1942 is_admin_event(restarter_event_type_t t) {

1948        switch (t) {
1949        case RESTARTER_EVENT_TYPE_ADMIN_MAINT_ON:
1950        case RESTARTER_EVENT_TYPE_ADMIN_MAINT_ON_IMMEDIATE:
1951        case RESTARTER_EVENT_TYPE_ADMIN_MAINT_OFF:
1952        case RESTARTER_EVENT_TYPE_ADMIN_REFRESH:
1953        case RESTARTER_EVENT_TYPE_ADMIN_DEGRADED:
1954        case RESTARTER_EVENT_TYPE_ADMIN_RESTART:
1955                return (1);
1956        default:
1957                return (0);
1958        }
1959 }
```

*unchanged_portion_omitted_*

```
1979 /*
1980  * void *restarter_event_thread()
1981  *
1982  *  Handle incoming graph events by placing them on a per-instance
1983  *  queue.  We can't lock the main part of the instance structure, so
1984  *  just modify the seprarately locked event queue portion.
1985  */
1986 /*ARGSUSED*/
1987 static void *
1988 restarter_event_thread(void *unused)
1989 {
1990         scf_handle_t *h;

1992         (void) pthread_setname_np(pthread_self(), "restarter_event");

1994         /*
1995          * This is a new thread, and thus, gets its own handle
1996          * to the repository.
1997          */
1998         h = libscf_handle_create_bound_loop();

2000         MUTEX_LOCK(&ru->restarter_update_lock);

2002         /*CONSTCOND*/
2003         while (1) {
2004                 restarter_protocol_event_t *e;

2006                 while (ru->restarter_update_wakeup == 0)
2007                         (void) pthread_cond_wait(&ru->restarter_update_cv,
2008                             &ru->restarter_update_lock);

2010                 ru->restarter_update_wakeup = 0;

2012                 while ((e = restarter_event_dequeue()) != NULL) {
2013                         restarter_inst_t *rip;
2014                         char *fmri;

2016                         MUTEX_UNLOCK(&ru->restarter_update_lock);

2018                         /*
2019                          * ADD_INSTANCE is special: there's likely no
2020                          * instance structure yet, so we need to handle the
2021                          * addition synchronously.
2022                          */
2023                         switch (e->rpe_type) {
2024                         case RESTARTER_EVENT_TYPE_ADD_INSTANCE:
2025                                 if (restarter_insert_inst(h, e->rpe_inst) != 0)
2026                                         log_error(LOG_INFO, "Restarter: "
2027                                             "Could not add %s.\n", e->rpe_inst);

2029                                 MUTEX_LOCK(&st->st_load_lock);
2030                                 if (--st->st_load_instances == 0)
2031                                         (void) pthread_cond_broadcast(
2032                                             &st->st_load_cv);
2033                                 MUTEX_UNLOCK(&st->st_load_lock);

2035                                 goto nolookup;
2036                         }

2038                         /*
2039                          * Lookup the instance, locking only the event queue.
2040                          * Can't grab ri_lock here because it might be held
2041                          * by a long-running method.
2042                          */
2043                         rip = inst_lookup_queue(e->rpe_inst);
```



```
2044                         if (rip == NULL) {
2045                                 log_error(LOG_INFO, "Restarter: "
2046                                     "Ignoring %s command for unknown service "
2047                                     "%s.\n", event_names[e->rpe_type],
2048                                     e->rpe_inst);
2049                                 goto nolookup;
2050                         }

2052                         /* Keep ADMIN events from filling up the queue. */
2053                         if (is_admin_event(e->rpe_type) &&
2054                             uu_list_numnodes(rip->ri_queue) >
2055                             RINST_QUEUE_THRESHOLD) {
2056                                 MUTEX_UNLOCK(&rip->ri_queue_lock);
2057                                 log_instance(rip, B_TRUE, "Instance event "
2058                                     "queue overflow.  Dropping administrative "
2059                                     "request.");
2060                                 log_framework(LOG_DEBUG, "%s: Instance event "
2061                                     "queue overflow.  Dropping administrative "
2062                                     "request.\n", rip->ri_i.i_fmri);
2063                                 goto nolookup;
2064                         }

2066                         /* Now add the event to the instance queue. */
2067                         restarter_queue_event(rip, e);

2069                         if (rip->ri_queue_thread == 0) {
2070                                 /*
2071                                  * Start a thread if one isn't already
2072                                  * running.
2073                                  */
2074                                 fmri = safe_strdup(e->rpe_inst);
2075                                 rip->ri_queue_thread =  startd_thread_create(
2076                                     restarter_process_events, (void *)fmri);
2077                         } else {
2078                                 /*
2079                                  * Signal the existing thread that there's
2080                                  * a new event.
2081                                  */
2082                                 (void) pthread_cond_broadcast(
2083                                     &rip->ri_queue_cv);
2084                         }

2086                         MUTEX_UNLOCK(&rip->ri_queue_lock);
2087 nolookup:
2088                         restarter_event_release(e);

2090                         MUTEX_LOCK(&ru->restarter_update_lock);
2091                 }
2092         }

2094         /*
2095          * Unreachable for now -- there's currently no graceful cleanup
2096          * called on exit().
2097          */
2098         (void) scf_handle_unbind(h);
2099         scf_handle_destroy(h);
2100         return (NULL);
2101 }
```

_____***unchanged_portion_omitted_***

```
2193 /*
2194  * void *restarter_contract_event_thread(void *)
2195  *   Listens to the process contract bundle for critical events, taking action
2196  *   on events from contracts we know we are responsible for.
2197  */
2198 /*ARGSUSED*/
```

```
2199 static void *
2200 restarter_contracts_event_thread(void *unused)
2201 {
2202         int fd, err;
2203         scf_handle_t *local_handle;

2205         (void) pthread_setname_np(pthread_self(), "restarter_contracts_event");

2207         /*
2208          * Await graph load completion.  That is, stop here, until we've scanned
2209          * the repository for contract - instance associations.
2210          */
2211         MUTEX_LOCK(&st->st_load_lock);
2212         while (!(st->st_load_complete && st->st_load_instances == 0))
2213                 (void) pthread_cond_wait(&st->st_load_cv, &st->st_load_lock);
2214         MUTEX_UNLOCK(&st->st_load_lock);

2216         /*
2217          * This is a new thread, and thus, gets its own handle
2218          * to the repository.
2219          */
2220         if ((local_handle = libscf_handle_create_bound(SCF_VERSION)) == NULL)
2221                 uu_die("Unable to bind a new repository handle: %s\n",
2222                     scf_strerror(scf_error()));

2224         fd = open64(CTFS_ROOT "/process/pbundle", O_RDONLY);
2225         if (fd == -1)
2226                 uu_die("process bundle open failed");

2228         /*
2229          * Make sure we get all events (including those generated by configd
2230          * before this thread was started).
2231          */
2232         err = ct_event_reset(fd);
2233         assert(err == 0);

2235         for (;;) {
2236                 int efd, sfd;
2237                 ct_evthdl_t ev;
2238                 uint32_t type;
2239                 ctevid_t evid;
2240                 ct_stathdl_t status;
2241                 ctid_t ctid;
2242                 restarter_inst_t *inst;
2243                 uint64_t cookie;

2245                 if (err = ct_event_read_critical(fd, &ev)) {
2246                         log_error(LOG_WARNING,
2247                             "Error reading next contract event: %s",
2248                             strerror(err));
2249                         continue;
2250                 }

2252                 evid = ct_event_get_evid(ev);
2253                 ctid = ct_event_get_ctid(ev);
2254                 type = ct_event_get_type(ev);

2256                 /* Fetch cookie. */
2257                 if ((sfd = contract_open(ctid, "process", "status", O_RDONLY))
2258                     < 0) {
2259                         ct_event_free(ev);
2260                         continue;
2261                 }

2263                 if (err = ct_status_read(sfd, CTD_COMMON, &status)) {
2264                         log_framework(LOG_WARNING, "Could not get status for "
```



```
2265                             "contract %ld: %s\n", ctid, strerror(err));

2267                         startd_close(sfd);
2268                         ct_event_free(ev);
2269                         continue;
2270                 }

2272                 cookie = ct_status_get_cookie(status);

2274                 log_framework(LOG_DEBUG, "Received event %d for ctid %ld "
2275                     "cookie %lld\n", type, ctid, cookie);

2277                 ct_status_free(status);

2279                 startd_close(sfd);

2281                 /*
2282                  * svc.configd(1M) restart handling performed by the
2283                  * fork_configd_thread.  We don't acknowledge, as that thread
2284                  * will do so.
2285                  */
2286                 if (cookie == CONFIGD_COOKIE) {
2287                         ct_event_free(ev);
2288                         continue;
2289                 }

2291                 inst = NULL;
2292                 if (storing_contract != 0 &&
2293                     (inst = contract_to_inst(ctid)) == NULL) {
2294                         /*
2295                          * This can happen for two reasons:
2296                          * - method_run() has not yet stored the
2297                          *    the contract into the internal hash table.
2298                          * - we receive an EMPTY event for an abandoned
2299                          *    contract.
2300                          * If there is any contract in the process of
2301                          * being stored into the hash table then re-read
2302                          * the event later.
2303                          */
2304                         log_framework(LOG_DEBUG,
2305                             "Reset event %d for unknown "
2306                             "contract id %ld\n", type, ctid);

2308                         /* don't go too fast */
2309                         (void) poll(NULL, 0, 100);

2311                         (void) ct_event_reset(fd);
2312                         ct_event_free(ev);
2313                         continue;
2314                 }

2316                 /*
2317                  * Do not call contract_to_inst() again if first
2318                  * call succeeded.
2319                  */
2320                 if (inst == NULL)
2321                         inst = contract_to_inst(ctid);
2322                 if (inst == NULL) {
2323                         /*
2324                          * This can happen if we receive an EMPTY
2325                          * event for an abandoned contract.
2326                          */
2327                         log_framework(LOG_DEBUG,
2328                             "Received event %d for unknown contract id "
2329                             "%ld\n", type, ctid);
2330                 } else {
```

```
2331                        log_framework(LOG_DEBUG,
2332                            "Received event %d for contract id "
2333                            "%ld (%s)\n", type, ctid,
2334                            inst->ri_i.i_fmri);

2336                        contract_action(local_handle, inst, ctid, type);

2338                        MUTEX_UNLOCK(&inst->ri_lock);
2339                }

2341                efd = contract_open(ct_event_get_ctid(ev), "process", "ctl",
2342                    O_WRONLY);
2343                if (efd != -1) {
2344                        (void) ct_ctl_ack(efd, evid);
2345                        startd_close(efd);
2346                }

2348                ct_event_free(ev);

2350        }

2352        /*NOTREACHED*/
2353        return (NULL);
2354 }
```

*unchanged_portion_omitted_*

```
2540 /*
2541  * void *restarter_timeouts_event_thread(void *)
2542  *   Responsible for monitoring the method timeouts.  This thread must
2543  *   be started before any methods are called.
2544  */
2545 /*ARGSUSED*/
2546 static void *
2547 restarter_timeouts_event_thread(void *unused)
2548 {
2549         /*
2550          * Timeouts are entered on a priority queue, which is processed by
2551          * this thread.  As timeouts are specified in seconds, we'll do
2552          * the necessary processing every second, as long as the queue
2553          * is not empty.
2554          */

2556         (void) pthread_setname_np(pthread_self(), "restarter_timeouts_event");

2558         /*CONSTCOND*/
2559         while (1) {
2560                 /*
2561                  * As long as the timeout list isn't empty, process it
2562                  * every second.
2563                  */
2564                 if (timeout_now() == 0) {
2565                         (void) sleep(1);
2566                         continue;
2567                 }

2569                 /* The list is empty, wait until we have more timeouts. */
2570                 MUTEX_LOCK(&tu->tu_lock);

2572                 while (tu->tu_wakeup == 0)
2573                         (void) pthread_cond_wait(&tu->tu_cv, &tu->tu_lock);

2575                 tu->tu_wakeup = 0;
2576                 MUTEX_UNLOCK(&tu->tu_lock);
2577         }

2579         return (NULL);
```



```
2580 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    9967 Mon Oct 15 13:25:13 2018
new/usr/src/cmd/svc/startd/wait.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 * Copyright 2012, Joyent, Inc.  All rights reserved.
 */

/*
 * wait.c - asynchronous monitoring of "wait registered" start methods
 *
 * Use event ports to poll on the set of fds representing the /proc/[pid]/psinfo
 * files.  If one of these fds returns an event, then we inform the restarter
 * that it has stopped.
 *
 * The wait_info_list holds the series of processes currently being monitored
 * for exit.  The wi_fd member, which contains the file descriptor of the psinfo
 * file being polled upon ("event ported upon"), will be set to -1 if the file
 * descriptor is inactive (already closed or not yet opened).
 */

#ifdef _FILE_OFFSET_BITS
#undef _FILE_OFFSET_BITS
#endif /* _FILE_OFFSET_BITS */

#include <sys/resource.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <sys/wait.h>

#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <libuutil.h>
#include <poll.h>
#include <port.h>
#include <pthread.h>
#include <procfs.h>
#include <string.h>
```



```
#include <stropts.h>
#include <unistd.h>

#include "startd.h"

#define WAIT_FILES      262144          /* reasonably high maximum */

static int port_fd;
static scf_handle_t *wait_hndl;
static struct rlimit init_fd_rlimit;

static uu_list_pool_t *wait_info_pool;
static uu_list_t *wait_info_list;

static pthread_mutex_t wait_info_lock;

/*
 * void wait_remove(wait_info_t *, int)
 *   Remove the given wait_info structure from our list, performing various
 *   cleanup operations along the way.  If the direct flag is false (meaning
 *   that we are being called with from restarter instance list context) and
 *   the instance should not be ignored, then notify the restarter that the
 *   associated instance has exited. If the wi_ignore flag is true then it
 *   means that the stop was initiated from within svc.startd, rather than
 *   from outside it.
 *
 *   Since we may no longer be the startd that started this process, we only are
 *   concerned with a waitpid(3C) failure if the wi_parent field is non-zero.
 */
static void
wait_remove(wait_info_t *wi, int direct)
{
        int status;
        stop_cause_t cause = RSTOP_EXIT;

        if (waitpid(wi->wi_pid, &status, 0) == -1) {
                if (wi->wi_parent)
                        log_framework(LOG_INFO,
                            "instance %s waitpid failure: %s\n", wi->wi_fmri,
                            strerror(errno));
        } else {
                if (WEXITSTATUS(status) != 0) {
                        log_framework(LOG_NOTICE,
                            "instance %s exited with status %d\n", wi->wi_fmri,
                            WEXITSTATUS(status));
                        if (WEXITSTATUS(status) == SMF_EXIT_ERR_CONFIG)
                                cause = RSTOP_ERR_CFG;
                        else
                                cause = RSTOP_ERR_EXIT;
                }
        }

        MUTEX_LOCK(&wait_info_lock);
        if (wi->wi_fd != -1) {
                startd_close(wi->wi_fd);
                wi->wi_fd = -1;
        }
        uu_list_remove(wait_info_list, wi);
        MUTEX_UNLOCK(&wait_info_lock);

        /*
         * Make an attempt to clear out any utmpx record associated with this
         * PID.
         */
        utmpx_mark_dead(wi->wi_pid, status, B_FALSE);
```

```
126         if (!direct && !wi->wi_ignore) {
127                 /*
128                  * Bind wait_hndl lazily.
129                  */
130                 if (wait_hndl == NULL) {
131                         for (wait_hndl =
132                             libscf_handle_create_bound(SCF_VERSION);
133                             wait_hndl == NULL;
134                             wait_hndl =
135                             libscf_handle_create_bound(SCF_VERSION)) {
136                                 log_error(LOG_INFO, "[wait_remove] Unable to "
137                                     "bind a new repository handle: %s\n",
138                                     scf_strerror(scf_error()));
139                                 (void) sleep(2);
140                         }
141                 }

143                 log_framework(LOG_DEBUG,
144                     "wait_remove requesting stop of %s\n", wi->wi_fmri);
145                 (void) stop_instance_fmri(wait_hndl, wi->wi_fmri, cause);
146         }

148         uu_list_node_fini(wi, &wi->wi_link, wait_info_pool);
149         startd_free(wi, sizeof (wait_info_t));
150 }
```

*unchanged_portion_omitted_*

```
252 /*ARGSUSED*/
253 void *
254 wait_thread(void *args)
255 {
256         (void) pthread_setname_np(pthread_self(), "wait");

258         for (;;) {
259                 port_event_t pe;
260                 int fd;
261                 wait_info_t *wi;

263                 if (port_get(port_fd, &pe, NULL) != 0) {
264                         if (errno == EINTR)
265                                 continue;
266                         else {
267                                 log_error(LOG_WARNING,
268                                     "port_get() failed with %s\n",
269                                     strerror(errno));
270                                 bad_error("port_get", errno);
271                         }
272                 }

274                 fd = pe.portev_object;
275                 wi = pe.portev_user;
276                 assert(wi != NULL);
277                 assert(fd == wi->wi_fd);

279                 if ((pe.portev_events & POLLHUP) == POLLHUP) {
280                         psinfo_t psi;

282                         if (lseek(fd, 0, SEEK_SET) != 0 ||
283                             read(fd, &psi, sizeof (psinfo_t)) !=
284                             sizeof (psinfo_t)) {
285                                 log_framework(LOG_WARNING,
286                                     "couldn't get psinfo data for %s (%s); "
287                                     "assuming failed\n", wi->wi_fmri,
288                                     strerror(errno));
289                                 goto err_remove;
290                         }
```



```
292                         if (psi.pr_nlwp != 0 ||
293                             psi.pr_nzomb != 0 ||
294                             psi.pr_lwp.pr_lwpid != 0) {
295                                 /*
296                                  * We have determined, in accordance with the
297                                  * definition in proc(4), this process is not a
298                                  * zombie.  Reassociate.
299                                  */
300                                 if (port_associate(port_fd, PORT_SOURCE_FD, fd,
301                                     0, wi))
302                                         log_error(LOG_WARNING,
303                                             "port_association of %d / %s "
304                                             "failed\n", fd, wi->wi_fmri);
305                                 continue;
306                         }
307                 } else if (
308                     (pe.portev_events & POLLERR) == 0) {
309                         if (port_associate(port_fd, PORT_SOURCE_FD, fd, 0, wi))
310                                 log_error(LOG_WARNING,
311                                     "port_association of %d / %s "
312                                     "failed\n", fd, wi->wi_fmri);
313                         continue;
314                 }

316 err_remove:
317                 wait_remove(wi, 0);
318         }

320         /*LINTED E_FUNC_HAS_NO_RETURN_STMT*/
321 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   14246 Mon Oct 15 13:25:13 2018
new/usr/src/head/pthread.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 * Copyright 2018 Joyent, Inc.
 * Copyright 2016 Joyent, Inc.
 *
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifndef _PTHREAD_H
#define _PTHREAD_H

#include <sys/feature_tests.h>

#ifndef _ASM
#include <sys/types.h>
#include <time.h>
#include <sched.h>
#endif  /* _ASM */

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Thread related attribute values defined as in thread.h.
 * These are defined as bit pattern in thread.h.
 * Any change here should be reflected in thread.h.
 */
/* detach */
#define PTHREAD_CREATE_DETACHED         0x40    /* = THR_DETACHED */
#define PTHREAD_CREATE_JOINABLE         0
/* scope */
#define PTHREAD_SCOPE_SYSTEM            0x01    /* = THR_BOUND */
#define PTHREAD_SCOPE_PROCESS           0

/*
 * Other attributes which are not defined in thread.h
 */
```



```
/* inherit */
#define PTHREAD_INHERIT_SCHED           1
#define PTHREAD_EXPLICIT_SCHED          0

/*
 * Value of process-shared attribute
 * These are defined as values defined in sys/synch.h
 * Any change here should be reflected in sys/synch.h.
 */
#define PTHREAD_PROCESS_SHARED          1       /* = USYNC_PROCESS */
#define PTHREAD_PROCESS_PRIVATE         0       /* = USYNC_THREAD */

/*
 * mutex types
 * keep these in synch which sys/synch.h lock flags
 */
#define PTHREAD_MUTEX_NORMAL            0x0
#define PTHREAD_MUTEX_ERRORCHECK        0x2
#define PTHREAD_MUTEX_RECURSIVE         0x4
#define PTHREAD_MUTEX_DEFAULT           PTHREAD_MUTEX_NORMAL

/*
 * Mutex protocol values. Keep these in synch with sys/synch.h lock types.
 */
#define PTHREAD_PRIO_NONE               0x0
#define PTHREAD_PRIO_INHERIT            0x10
#define PTHREAD_PRIO_PROTECT            0x20

/*
 * Mutex robust attribute values.
 * Keep these in synch with sys/synch.h lock types.
 */
#define PTHREAD_MUTEX_STALLED           0x0
#define PTHREAD_MUTEX_ROBUST            0x40
/*
 * Historical solaris-specific names,
 * from before pthread_mutexattr_getrobust() became standardized
 */
#define PTHREAD_MUTEX_STALL_NP          PTHREAD_MUTEX_STALLED
#define PTHREAD_MUTEX_ROBUST_NP         PTHREAD_MUTEX_ROBUST

/*
 * macros - default initializers defined as in synch.h
 * Any change here should be reflected in synch.h.
 *
 * NOTE:
 * Make sure that any change in the macros is consistent with the definition
 * of the corresponding types in sys/types.h (e.g. PTHREAD_MUTEX_INITIALIZER
 * should be consistent with the definition for pthread_mutex_t).
 */
#define PTHREAD_MUTEX_INITIALIZER               /* = DEFAULTMUTEX */    \
        {{0, 0, 0, PTHREAD_PROCESS_PRIVATE, _MUTEX_MAGIC}, {{{0}}}, 0}

#define PTHREAD_COND_INITIALIZER                /* = DEFAULTCV */       \
        {{{0, 0, 0, 0}, PTHREAD_PROCESS_PRIVATE, _COND_MAGIC}, 0}

#define PTHREAD_RWLOCK_INITIALIZER              /* = DEFAULTRWLOCK */   \
        {0, PTHREAD_PROCESS_PRIVATE, _RWL_MAGIC, PTHREAD_MUTEX_INITIALIZER, \
        PTHREAD_COND_INITIALIZER, PTHREAD_COND_INITIALIZER}

/* cancellation type and state */
#define PTHREAD_CANCEL_ENABLE           0x00
#define PTHREAD_CANCEL_DISABLE          0x01
#define PTHREAD_CANCEL_DEFERRED         0x00
#define PTHREAD_CANCEL_ASYNCHRONOUS     0x02
#define PTHREAD_CANCELED                (void *)-19
```

```
127 /* pthread_once related values */
128 #define PTHREAD_ONCE_NOTDONE    0
129 #define PTHREAD_ONCE_DONE       1
130 #define PTHREAD_ONCE_INIT       { {0, 0, 0, PTHREAD_ONCE_NOTDONE} }

132 /*
133  * The key to be created by pthread_key_create_once_np()
134  * must be statically initialized with PTHREAD_ONCE_KEY_NP.
135  * This must be the same as THR_ONCE_KEY in <thread.h>
136  */
137 #define PTHREAD_ONCE_KEY_NP     (pthread_key_t)(-1)

139 /* barriers */
140 #define PTHREAD_BARRIER_SERIAL_THREAD   -2

142 /* For pthread_{get,set}name_np(). */
143 #define PTHREAD_MAX_NAMELEN_NP (32)

145 #ifndef _ASM

147 /*
148  * cancellation cleanup structure
149  */
150 typedef struct _cleanup {
151         uintptr_t       pthread_cleanup_pad[4];
152 } _cleanup_t;
      unchanged_portion_omitted_

177 /*
178  * function prototypes - thread related calls
179  */

181 /*
182  * pthread_atfork() is also declared in <unistd.h> as per SUSv2. The
183  * declarations are identical. A change to either one may also require
184  * appropriate namespace updates in order to avoid redeclaration
185  * warnings in the case where both prototypes are exposed via inclusion
186  * of both <pthread.h> and <unistd.h>.
187  */
188 extern int pthread_atfork(void (*) (void), void (*) (void), void (*) (void));
189 extern int pthread_attr_init(pthread_attr_t *);
190 extern int pthread_attr_destroy(pthread_attr_t *);
191 extern int pthread_attr_setstack(pthread_attr_t *, void *, size_t);
192 extern int pthread_attr_getstack(const pthread_attr_t *_RESTRICT_KYWD,
193                 void **_RESTRICT_KYWD, size_t *_RESTRICT_KYWD);
194 extern int pthread_attr_setstacksize(pthread_attr_t *, size_t);
195 extern int pthread_attr_getstacksize(const pthread_attr_t *_RESTRICT_KYWD,
196                 size_t *_RESTRICT_KYWD);
197 extern int pthread_attr_setstackaddr(pthread_attr_t *, void *);
198 extern int pthread_attr_getstackaddr(const pthread_attr_t *_RESTRICT_KYWD,
199                 void **_RESTRICT_KYWD);
200 extern int pthread_attr_setdetachstate(pthread_attr_t *, int);
201 extern int pthread_attr_getdetachstate(const pthread_attr_t *, int *);
202 extern int pthread_attr_setscope(pthread_attr_t *, int);
203 extern int pthread_attr_getscope(const pthread_attr_t *_RESTRICT_KYWD,
204         int *_RESTRICT_KYWD);
205 extern int pthread_attr_setinheritsched(pthread_attr_t *, int);
206 extern int pthread_attr_getinheritsched(const pthread_attr_t *_RESTRICT_KYWD,
207         int *_RESTRICT_KYWD);
208 extern int pthread_attr_setschedpolicy(pthread_attr_t *, int);
209 extern int pthread_attr_getschedpolicy(const pthread_attr_t *_RESTRICT_KYWD,
210         int *_RESTRICT_KYWD);
211 extern int pthread_attr_setschedparam(pthread_attr_t *_RESTRICT_KYWD,
212                 const struct sched_param *_RESTRICT_KYWD);
213 extern int pthread_attr_getschedparam(const pthread_attr_t *_RESTRICT_KYWD,
```



```
214                 struct sched_param *_RESTRICT_KYWD);
215 extern int pthread_attr_setname_np(pthread_attr_t *_RESTRICT_KYWD,
216     const char *_RESTRICT_KYWD);
217 extern int pthread_attr_getname_np(pthread_attr_t *_RESTRICT_KYWD,
218     char *_RESTRICT_KYWD, size_t);
219 extern int pthread_create(pthread_t *_RESTRICT_KYWD,
220                 const pthread_attr_t *_RESTRICT_KYWD, void * (*)(void *),
221                 void *_RESTRICT_KYWD);
222 extern int pthread_once(pthread_once_t *, void (*)(void));
223 extern int pthread_join(pthread_t, void **);
224 extern int pthread_detach(pthread_t);
225 extern void pthread_exit(void *) __NORETURN;
226 extern int pthread_cancel(pthread_t);
227 extern int pthread_setschedparam(pthread_t, int, const struct sched_param *);
228 extern int pthread_getschedparam(pthread_t, int *_RESTRICT_KYWD,
229                 struct sched_param *_RESTRICT_KYWD);
230 extern int pthread_setschedprio(pthread_t, int);
231 extern int pthread_setcancelstate(int, int *);
232 extern int pthread_setcanceltype(int, int *);
233 extern void pthread_testcancel(void);
234 extern int pthread_equal(pthread_t, pthread_t);
235 extern int pthread_key_create(pthread_key_t *, void (*)(void *));
236 extern int pthread_key_create_once_np(pthread_key_t *, void (*)(void *));
237 extern int pthread_key_delete(pthread_key_t);
238 extern int pthread_setspecific(pthread_key_t, const void *);
239 extern void *pthread_getspecific(pthread_key_t);
240 extern pthread_t pthread_self(void);
241 extern int pthread_setname_np(pthread_t, const char *);
242 extern int pthread_getname_np(pthread_t, char *, size_t);

244 /*
245  * function prototypes - synchronization related calls
246  */
247 extern int pthread_mutexattr_init(pthread_mutexattr_t *);
248 extern int pthread_mutexattr_destroy(pthread_mutexattr_t *);
249 extern int pthread_mutexattr_setpshared(pthread_mutexattr_t *, int);
250 extern int pthread_mutexattr_getpshared(
251         const pthread_mutexattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
252 extern int pthread_mutexattr_setprotocol(pthread_mutexattr_t *, int);
253 extern int pthread_mutexattr_getprotocol(
254         const pthread_mutexattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
255 extern int pthread_mutexattr_setprioceiling(pthread_mutexattr_t *, int);
256 extern int pthread_mutexattr_getprioceiling(
257         const pthread_mutexattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
258 extern int pthread_mutexattr_setrobust(pthread_mutexattr_t *, int);
259 extern int pthread_mutexattr_getrobust(
260         const pthread_mutexattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
261 extern int pthread_mutex_init(pthread_mutex_t *_RESTRICT_KYWD,
262         const pthread_mutexattr_t *_RESTRICT_KYWD);
263 extern int pthread_mutex_consistent(pthread_mutex_t *);
264 extern int pthread_mutex_destroy(pthread_mutex_t *);
265 extern int pthread_mutex_lock(pthread_mutex_t *);
266 extern int pthread_mutex_timedlock(pthread_mutex_t *_RESTRICT_KYWD,
267         const struct timespec *_RESTRICT_KYWD);
268 extern int pthread_mutex_reltimedlock_np(pthread_mutex_t *_RESTRICT_KYWD,
269         const struct timespec *_RESTRICT_KYWD);
270 extern int pthread_mutex_unlock(pthread_mutex_t *);
271 extern int pthread_mutex_trylock(pthread_mutex_t *);
272 extern int pthread_mutex_setprioceiling(pthread_mutex_t *_RESTRICT_KYWD,
273         int, int *_RESTRICT_KYWD);
274 extern int pthread_mutex_getprioceiling(const pthread_mutex_t *_RESTRICT_KYWD,
275         int *_RESTRICT_KYWD);
276 extern int pthread_condattr_init(pthread_condattr_t *);
277 extern int pthread_condattr_destroy(pthread_condattr_t *);
278 extern int pthread_condattr_setclock(pthread_condattr_t *, clockid_t);
279 extern int pthread_condattr_getclock(const pthread_condattr_t *_RESTRICT_KYWD,
```

```
280         clockid_t *_RESTRICT_KYWD);
281 extern int pthread_condattr_setpshared(pthread_condattr_t *, int);
282 extern int pthread_condattr_getpshared(const pthread_condattr_t *_RESTRICT_KYWD,
283         int *_RESTRICT_KYWD);
284 extern int pthread_cond_init(pthread_cond_t *_RESTRICT_KYWD,
285         const pthread_condattr_t *_RESTRICT_KYWD);
286 extern int pthread_cond_destroy(pthread_cond_t *);
287 extern int pthread_cond_broadcast(pthread_cond_t *);
288 extern int pthread_cond_signal(pthread_cond_t *);
289 extern int pthread_cond_wait(pthread_cond_t *_RESTRICT_KYWD,
290         pthread_mutex_t *_RESTRICT_KYWD);
291 extern int pthread_cond_timedwait(pthread_cond_t *_RESTRICT_KYWD,
292         pthread_mutex_t *_RESTRICT_KYWD, const struct timespec *_RESTRICT_KYWD);
293 extern int pthread_cond_reltimedwait_np(pthread_cond_t *_RESTRICT_KYWD,
294         pthread_mutex_t *_RESTRICT_KYWD, const struct timespec *_RESTRICT_KYWD);
295 extern int pthread_attr_getguardsize(const pthread_attr_t *_RESTRICT_KYWD,
296         size_t *_RESTRICT_KYWD);
297 extern int pthread_attr_setguardsize(pthread_attr_t *, size_t);
298 extern int pthread_getconcurrency(void);
299 extern int pthread_setconcurrency(int);
300 extern int pthread_mutexattr_settype(pthread_mutexattr_t *, int);
301 extern int pthread_mutexattr_gettype(const pthread_mutexattr_t *_RESTRICT_KYWD,
302         int *_RESTRICT_KYWD);
303 extern int pthread_rwlock_init(pthread_rwlock_t *_RESTRICT_KYWD,
304         const pthread_rwlockattr_t *_RESTRICT_KYWD);
305 extern int pthread_rwlock_destroy(pthread_rwlock_t *);
306 extern int pthread_rwlock_rdlock(pthread_rwlock_t *);
307 extern int pthread_rwlock_timedrdlock(pthread_rwlock_t *_RESTRICT_KYWD,
308         const struct timespec *_RESTRICT_KYWD);
309 extern int pthread_rwlock_reltimedrdlock_np(pthread_rwlock_t *_RESTRICT_KYWD,
310         const struct timespec *_RESTRICT_KYWD);
311 extern int pthread_rwlock_tryrdlock(pthread_rwlock_t *);
312 extern int pthread_rwlock_wrlock(pthread_rwlock_t *);
313 extern int pthread_rwlock_timedwrlock(pthread_rwlock_t *_RESTRICT_KYWD,
314         const struct timespec *_RESTRICT_KYWD);
315 extern int pthread_rwlock_reltimedwrlock_np(pthread_rwlock_t *_RESTRICT_KYWD,
316         const struct timespec *_RESTRICT_KYWD);
317 extern int pthread_rwlock_trywrlock(pthread_rwlock_t *);
318 extern int pthread_rwlock_unlock(pthread_rwlock_t *);
319 extern int pthread_rwlockattr_init(pthread_rwlockattr_t *);
320 extern int pthread_rwlockattr_destroy(pthread_rwlockattr_t *);
321 extern int pthread_rwlockattr_getpshared(
322         const pthread_rwlockattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
323 extern int pthread_rwlockattr_setpshared(pthread_rwlockattr_t *, int);
324 extern int pthread_spin_init(pthread_spinlock_t *, int);
325 extern int pthread_spin_destroy(pthread_spinlock_t *);
326 extern int pthread_spin_lock(pthread_spinlock_t *);
327 extern int pthread_spin_trylock(pthread_spinlock_t *);
328 extern int pthread_spin_unlock(pthread_spinlock_t *);
329 extern int pthread_barrierattr_init(pthread_barrierattr_t *);
330 extern int pthread_barrierattr_destroy(pthread_barrierattr_t *);
331 extern int pthread_barrierattr_setpshared(pthread_barrierattr_t *, int);
332 extern int pthread_barrierattr_getpshared(
333         const pthread_barrierattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
334 extern int pthread_barrier_init(pthread_barrier_t *_RESTRICT_KYWD,
335         const pthread_barrierattr_t *_RESTRICT_KYWD, uint_t);
336 extern int pthread_barrier_destroy(pthread_barrier_t *);
337 extern int pthread_barrier_wait(pthread_barrier_t *);

339 /* Historical names -- present only for binary compatibility */
340 extern int pthread_mutex_consistent_np(pthread_mutex_t *);
341 extern int pthread_mutexattr_setrobust_np(pthread_mutexattr_t *, int);
342 extern int pthread_mutexattr_getrobust_np(
343         const pthread_mutexattr_t *_RESTRICT_KYWD, int *_RESTRICT_KYWD);

345 /*
```



```
346  * These are non-standardized extensions that we provide. Their origins are
347  * documented in their manual pages.
348  */
349 #if !defined(_STRICT_SYMBOLS) || defined(__EXTENSIONS__)
350 extern int pthread_attr_get_np(pthread_t, pthread_attr_t *);
351 #endif  /* !_STRICT_SYMBOLS || __EXTENSIONS__ */

353 #endif  /* _ASM */

355 #ifdef  __cplusplus
356 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3962 Mon Oct 15 13:25:13 2018
new/usr/src/head/thread.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 *
 * Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _THREAD_H
#define _THREAD_H

/*
 * thread.h:
 * definitions needed to use the thread interface except synchronization.
 * use <synch.h> for thread synchronization.
 */

#ifndef _ASM
#include <sys/signal.h>
#include <sys/time.h>
#include <synch.h>
#endif  /* _ASM */

#ifdef __cplusplus
extern "C" {
#endif

#ifndef _ASM
typedef unsigned int thread_t;
typedef unsigned int thread_key_t;

extern int thr_create(void *, size_t, void *(*)(void *), void *, long,
                      thread_t *);
extern int thr_join(thread_t, thread_t *, void **);
extern int thr_setconcurrency(int);
extern int thr_getconcurrency(void);
extern void thr_exit(void *) __NORETURN;
extern thread_t thr_self(void);
```



```
/*
 * the definition of thr_sigsetmask() is not strict ansi-c since sigset_t is
 * not in the strict ansi-c name space. Hence, include the prototype for
 * thr_sigsetmask() only if strict ansi-c conformance is not turned on.
 */
#if !defined(_STRICT_STDC) || defined(__EXTENSIONS__)
extern int thr_sigsetmask(int, const sigset_t *, sigset_t *);
#endif

/*
 * the definition of thr_stksegment() is not strict ansi-c since stack_t is
 * not in the strict ansi-c name space. Hence, include the prototype for
 * thr_stksegment() only if strict ansi-c conformance is not turned on.
 */
#if !defined(_STRICT_STDC) || defined(__EXTENSIONS__)
extern int thr_stksegment(stack_t *);
#endif

extern int thr_main(void);
extern int thr_kill(thread_t, int);
extern int thr_suspend(thread_t);
extern int thr_continue(thread_t);
extern void thr_yield(void);
extern int thr_setprio(thread_t, int);
extern int thr_getprio(thread_t, int *);
extern int thr_keycreate(thread_key_t *, void(*)(void *));
extern int thr_keycreate_once(thread_key_t *, void(*)(void *));
extern int thr_setspecific(thread_key_t, void *);
extern int thr_getspecific(thread_key_t, void **);
extern size_t thr_min_stack(void);
extern int thr_getname(thread_t, char *, size_t);
extern int thr_setname(thread_t, const char *);

#endif /* _ASM */

#define THR_MIN_STACK   thr_min_stack()
/*
 * thread flags (one word bit mask)
 */
/*
 * POSIX.1c Note:
 * THR_BOUND is defined same as PTHREAD_SCOPE_SYSTEM in <pthread.h>
 * THR_DETACHED is defined same as PTHREAD_CREATE_DETACHED in <pthread.h>
 * Any changes in these definitions should be reflected in <pthread.h>
 */
#define THR_BOUND               0x00000001      /* = PTHREAD_SCOPE_SYSTEM */
#define THR_NEW_LWP             0x00000002
#define THR_DETACHED            0x00000040      /* = PTHREAD_CREATE_DETACHED */
#define THR_SUSPENDED           0x00000080
#define THR_DAEMON              0x00000100

/*
 * The key to be created by thr_keycreate_once()
 * must be statically initialized with THR_ONCE_KEY.
 * This must be the same as PTHREAD_ONCE_KEY_NP in <pthread.h>
 */
#define THR_ONCE_KEY    (thread_key_t)(-1)

/*
 * The available register states returned by thr_getstate().
 */
#define TRS_VALID       0
#define TRS_NONVOLATILE 1
#define TRS_LWPID       2
#define TRS_INVALID     3
```

```
 128 #ifdef __cplusplus
 129 }
______unchanged_portion_omitted_
```

```
**********************************************************
   52785 Mon Oct 15 13:25:13 2018
new/usr/src/lib/libc/inc/thr_uberdata.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
  24  */

  26 /*
  27  * Copyright 2016 Joyent, Inc.
  28  * Copyright 2018 Nexenta Systems, Inc.
  29  */

  31 #ifndef _THR_UBERDATA_H
  32 #define _THR_UBERDATA_H

  34 #include <stdlib.h>
  35 #include <unistd.h>
  36 #include <sys/types.h>
  37 #include <fcntl.h>
  38 #include <string.h>
  39 #include <signal.h>
  40 #include <ucontext.h>
  41 #include <thread.h>
  42 #include <pthread.h>
  43 #include <atomic.h>
  44 #include <link.h>
  45 #include <sys/resource.h>
  46 #include <sys/lwp.h>
  47 #include <errno.h>
  48 #include <sys/asm_linkage.h>
  49 #include <sys/regset.h>
  50 #include <sys/fcntl.h>
  51 #include <sys/mman.h>
  52 #include <synch.h>
  53 #include <door.h>
  54 #include <limits.h>
  55 #include <sys/synch32.h>
  56 #include <schedctl.h>
  57 #include <sys/priocntl.h>
  58 #include <thread_db.h>
  59 #include <setjmp.h>
  60 #include <sys/thread.h>
```



```
  61 #include "libc_int.h"
  62 #include "tdb_agent.h"
  63 #include "thr_debug.h"

  65 /*
  66  * This is an implementation-specific include file for threading support.
  67  * It is not to be seen by the clients of the library.
  68  *
  69  * This file also describes uberdata in libc.
  70  *
  71  * The term "uberdata" refers to data that is unique and visible across
  72  * all link maps.  The name is meant to imply that such data is truly
  73  * global, not just locally global to a particular link map.
  74  *
  75  * See the Linker and Libraries Guide for a full description of alternate
  76  * link maps and how they are set up and used.
  77  *
  78  * Alternate link maps implement multiple global namespaces within a single
  79  * process.  There may be multiple instances of identical dynamic libraries
  80  * loaded in a process's address space at the same time, each on a different
  81  * link map (as determined by the dynamic linker), each with its own set of
  82  * global variables.  Which particular instance of a global variable is seen
  83  * by a thread running in the process is determined by the link map on which
  84  * the thread happens to be executing at the time.
  85  *
  86  * However, there are aspects of a process that are unique across all
  87  * link maps, in particular the structures used to implement threads
  88  * of control (in Sparc terminology, there is only one %g7 regardless
  89  * of the link map on which the thread is executing).
  90  *
  91  * All uberdata is referenced from a base pointer in the thread's ulwp_t
  92  * structure (which is also uberdata).  All allocations and deallocations
  93  * of uberdata are made via the uberdata-aware lmalloc() and lfree()
  94  * interfaces (malloc() and free() are simply locally-global).
  95  */

  97 /*
  98  * Special libc-private access to errno.
  99  * We do this so that references to errno do not invoke the dynamic linker.
 100  */
 101 #undef errno
 102 #define errno (*curthread->ul_errnop)

 104 /*
 105  * See <sys/synch32.h> for the reasons for these values
 106  * and why they are different for sparc and intel.
 107  */
 108 #if defined(__sparc)

 110 /* lock.lock64.pad[x]      4 5 6 7 */
 111 #define LOCKMASK        0xff000000
 112 #define WAITERMASK      0x000000ff
 113 #define SPINNERMASK     0x00ff0000
 114 #define SPINNERSHIFT    16
 115 #define WAITER          0x00000001
 116 #define LOCKSET         0xff
 117 #define LOCKCLEAR       0

 119 #define PIDSHIFT        32
 120 #define LOCKMASK64      0xffffffffff000000ULL
 121 #define LOCKBYTE64      0x00000000ff000000ULL
 122 #define WAITERMASK64    0x00000000000000ffULL
 123 #define SPINNERMASK64   0x0000000000ff0000ULL

 125 #elif defined(__x86)
```

```
 127 /* lock.lock64.pad[x]      7 6 5 4 */
 128 #define LOCKMASK        0xff000000
 129 #define WAITERMASK      0x00ff0000
 130 #define SPINNERMASK     0x0000ff00
 131 #define SPINNERSHIFT    8
 132 #define WAITER          0x00010000
 133 #define LOCKSET         0x01
 134 #define LOCKCLEAR       0

 136 #define PIDSHIFT        0
 137 #define LOCKMASK64      0xff000000ffffffffULL
 138 #define LOCKBYTE64      0x0100000000000000ULL
 139 #define WAITERMASK64    0x00ff000000000000ULL
 140 #define SPINNERMASK64   0x0000ff0000000000ULL

 142 #else
 143 #error "neither __sparc nor __x86 is defined"
 144 #endif

 146 /*
 147  * Fetch the owner of a USYNC_THREAD mutex.
 148  * Don't use this with process-shared mutexes;
 149  * the owing thread may be in a different process.
 150  */
 151 #define MUTEX_OWNER(mp) ((ulwp_t *)(uintptr_t)(mp)->mutex_owner)

 153 /*
 154  * Test if a thread owns a process-private (USYNC_THREAD) mutex.
 155  * This is inappropriate for a process-shared (USYNC_PROCESS) mutex.
 156  * The 'mp' argument must not have side-effects since it is evaluated twice.
 157  */
 158 #define MUTEX_OWNED(mp, thrp)   \
 159         ((mp)->mutex_lockw != 0 && MUTEX_OWNER(mp) == thrp)


 162 /*
 163  * uberflags.uf_tdb_register_sync is an interface with libc_db to enable the
 164  * collection of lock statistics by a debugger or other collecting tool.
 165  *
 166  * uberflags.uf_thread_error_detection is set by an environment variable:
 167  *      _THREAD_ERROR_DETECTION
 168  *              0 == no detection of locking primitive errors.
 169  *              1 == detect errors and issue a warning message.
 170  *              2 == detect errors, issue a warning message, and dump core.
 171  *
 172  * We bundle these together in uberflags.uf_trs_ted to make a test of either
 173  * being non-zero a single memory reference (for speed of mutex_lock(), etc).
 174  *
 175  * uberflags.uf_mt is set non-zero when the first thread (in addition
 176  * to the main thread) is created.
 177  *
 178  * We bundle all these flags together in uberflags.uf_all to make a test
 179  * of any being non-zero a single memory reference (again, for speed).
 180  */
 181 typedef union {
 182         int     uf_all;                 /* combined all flags */
 183         struct {
 184                 short   h_pad;
 185                 short   h_trs_ted;      /* combined reg sync & error detect */
 186         } uf_h;
 187         struct {
 188                 char    x_mt;
 189                 char    x_pad;
 190                 char    x_tdb_register_sync;
 191                 char    x_thread_error_detection;
 192         } uf_x;
```



```
 193 } uberflags_t;
______unchanged_portion_omitted_

1227 typedef struct  _thrattr {
1228         size_t  stksize;
1229         void    *stkaddr;
1230         int     detachstate;
1231         int     daemonstate;
1232         int     scope;
1233         int     prio;
1234         int     policy;
1235         int     inherit;
1236         size_t  guardsize;
1237         char    name[THREAD_NAME_MAX];
1238 } thrattr_t;
______unchanged_portion_omitted_

1244 /* _curthread() is inline for speed */
1245 extern  ulwp_t          *_curthread(void);
1246 #define curthread       (_curthread())

1248 /* this version (also inline) can be tested for NULL */
1249 extern  ulwp_t          *__curthread(void);

1251 /* get the current stack pointer (also inline) */
1252 extern  greg_t          stkptr(void);

1254 /*
1255  * Suppress __attribute__((...)) if we are not compiling with gcc
1256  */
1257 #if !defined(__GNUC__)
1258 #define __attribute__(string)
1259 #endif

1261 /* Fetch the dispatch (kernel) priority of a thread */
1262 #define real_priority(ulwp)     \
1263         ((ulwp)->ul_schedctl? (ulwp)->ul_schedctl->sc_priority : 0)

1265 /*
1266  * Implementation functions.  Not visible outside of the library itself.
1267  */
1268 extern  int     __nanosleep(const timespec_t *, timespec_t *);
1269 extern  void    getgregs(ulwp_t *, gregset_t);
1270 extern  void    setgregs(ulwp_t *, gregset_t);
1271 extern  void    thr_panic(const char *);
1272 #pragma rarely_called(thr_panic)
1273 extern  void    mutex_panic(mutex_t *, const char *);
1274 #pragma rarely_called(mutex_panic)
1275 extern  ulwp_t  *find_lwp(thread_t);
1276 extern  void    finish_init(void);
1277 extern  void    update_sched(ulwp_t *);
1278 extern  void    queue_alloc(void);
1279 extern  void    tmem_exit(void);
1280 extern  void    tsd_exit(void);
1281 extern  void    tsd_free(ulwp_t *);
1282 extern  void    tls_setup(void);
1283 extern  void    tls_exit(void);
1284 extern  void    tls_free(ulwp_t *);
1285 extern  void    rwl_free(ulwp_t *);
1286 extern  void    heldlock_exit(void);
1287 extern  void    heldlock_free(ulwp_t *);
1288 extern  void    sigacthandler(int, siginfo_t *, void *);
1289 extern  void    signal_init(void);
1290 extern  int     sigequalset(const sigset_t *, const sigset_t *);
1291 extern  void    mutex_setup(void);
1292 extern  void    take_deferred_signal(int);
```

```
1293 extern  void    *setup_top_frame(void *, size_t, ulwp_t *);
1294 extern  int     setup_context(ucontext_t *, void *(*func)(ulwp_t *),
1295                         ulwp_t *ulwp, caddr_t stk, size_t stksize);
1296 extern  volatile sc_shared_t *setup_schedctl(void);
1297 extern  void    *lmalloc(size_t);
1298 extern  void    lfree(void *, size_t);
1299 extern  void    *libc_malloc(size_t);
1300 extern  void    *libc_realloc(void *, size_t);
1301 extern  void    libc_free(void *);
1302 extern  char    *libc_strdup(const char *);
1303 extern  void    ultos(uint64_t, int, char *);
1304 extern  void    lock_error(const mutex_t *, const char *, void *, const char *);
1305 extern  void    rwlock_error(const rwlock_t *, const char *, const char *);
1306 extern  void    thread_error(const char *);
1307 extern  void    grab_assert_lock(void);
1308 extern  void    dump_queue_statistics(void);
1309 extern  void    collect_queue_statistics(void);
1310 extern  void    record_spin_locks(ulwp_t *);
1311 extern  void    remember_lock(mutex_t *);
1312 extern  void    forget_lock(mutex_t *);
1313 extern  void    register_lock(mutex_t *);
1314 extern  void    unregister_locks(void);
1315 #if defined(__sparc)
1316 extern  void    _flush_windows(void);
1317 #else
1318 #define _flush_windows()
1319 #endif
1320 extern  void    set_curthread(void *);

1322 /*
1323  * Utility function used when waking up many threads (more than MAXLWPS)
1324  * all at once.  See mutex_wakeup_all(), cond_broadcast(), and rw_unlock().
1325  */
1326 #define MAXLWPS 128     /* max remembered lwpids before overflow */
1327 #define NEWLWPS 2048    /* max remembered lwpids at first overflow */
1328 extern  lwpid_t *alloc_lwpids(lwpid_t *, int *, int *);

1330 /* enter a critical section */
1331 #define enter_critical(self)    (self->ul_critical++)

1333 /* exit a critical section, take deferred actions if necessary */
1334 extern  void    do_exit_critical(void);
1335 #define exit_critical(self)                                     \
1336         (void) (self->ul_critical--,                           \
1337             ((self->ul_curplease && self->ul_critical == 0)?    \
1338             (do_exit_critical(), 0) : 0))

1340 /*
1341  * Like enter_critical()/exit_critical() but just for deferring signals.
1342  * Unlike enter_critical()/exit_critical(), ul_sigdefer may be set while
1343  * calling application functions like constructors and destructors.
1344  * Care must be taken if the application function attempts to set
1345  * the signal mask while a deferred signal is present; the setting
1346  * of the signal mask must also be deferred.
1347  */
1348 #define sigoff(self)    (self->ul_sigdefer++)
1349 #define sigon(self)                                             \
1350         (void) ((--self->ul_sigdefer == 0 &&                   \
1351             self->ul_curplease && self->ul_critical == 0)?     \
1352             (do_exit_critical(), 0) : 0)

1354 /* these are exported functions */
1355 extern  void    _sigoff(void);
1356 extern  void    _sigon(void);

1358 #define sigorset(s1, s2)                                \
```



```
1359         (((s1)->__sigbits[0] |= (s2)->__sigbits[0]),    \
1360         ((s1)->__sigbits[1] |= (s2)->__sigbits[1]),     \
1361         ((s1)->__sigbits[2] |= (s2)->__sigbits[2]),     \
1362         ((s1)->__sigbits[3] |= (s2)->__sigbits[3]))

1364 #define sigandset(s1, s2)                               \
1365         (((s1)->__sigbits[0] &= (s2)->__sigbits[0]),    \
1366         ((s1)->__sigbits[1] &= (s2)->__sigbits[1]),     \
1367         ((s1)->__sigbits[2] &= (s2)->__sigbits[2]),     \
1368         ((s1)->__sigbits[3] &= (s2)->__sigbits[3]))

1370 #define sigdiffset(s1, s2)                              \
1371         (((s1)->__sigbits[0] &= ~(s2)->__sigbits[0]),   \
1372         ((s1)->__sigbits[1] &= ~(s2)->__sigbits[1]),    \
1373         ((s1)->__sigbits[2] &= ~(s2)->__sigbits[2]),    \
1374         ((s1)->__sigbits[3] &= ~(s2)->__sigbits[3]))

1376 #define delete_reserved_signals(s)                      \
1377         (((s)->__sigbits[0] &= MASKSET0),               \
1378         ((s)->__sigbits[1] &= (MASKSET1 & ~SIGMASK(SIGCANCEL))),\
1379         ((s)->__sigbits[2] &= MASKSET2),                \
1380         ((s)->__sigbits[3] &= MASKSET3))

1382 extern  void    block_all_signals(ulwp_t *self);

1384 /*
1385  * When restoring the signal mask after having previously called
1386  * block_all_signals(), if we have a deferred signal present then
1387  * do nothing other than ASSERT() that we are in a critical region.
1388  * The signal mask will be set when we emerge from the critical region
1389  * and call take_deferred_signal().  There is no race condition here
1390  * because the kernel currently has all signals blocked for this thread.
1391  */
1392 #define restore_signals(self)                                           \
1393         ((void) ((self)->ul_cursig?                                    \
1394         (ASSERT((self)->ul_critical + (self)->ul_sigdefer != 0), 0) :   \
1395         __lwp_sigmask(SIG_SETMASK, &(self)->ul_sigmask)))

1397 extern  void    set_cancel_pending_flag(ulwp_t *, int);
1398 extern  void    set_cancel_eintr_flag(ulwp_t *);
1399 extern  void    set_parking_flag(ulwp_t *, int);
1400 extern  int     cancel_active(void);

1402 extern  void    *_thrp_setup(ulwp_t *);
1403 extern  void    _fpinherit(ulwp_t *);
1404 extern  void    _lwp_start(void);
1405 extern  void    _lwp_terminate(void);
1406 extern  void    lmutex_lock(mutex_t *);
1407 extern  void    lmutex_unlock(mutex_t *);
1408 extern  void    lrw_rdlock(rwlock_t *);
1409 extern  void    lrw_wrlock(rwlock_t *);
1410 extern  void    lrw_unlock(rwlock_t *);
1411 extern  void    sig_mutex_lock(mutex_t *);
1412 extern  void    sig_mutex_unlock(mutex_t *);
1413 extern  int     sig_mutex_trylock(mutex_t *);
1414 extern  int     sig_cond_wait(cond_t *, mutex_t *);
1415 extern  int     sig_cond_reltimedwait(cond_t *, mutex_t *, const timespec_t *);
1416 extern  void    cancel_safe_mutex_lock(mutex_t *);
1417 extern  void    cancel_safe_mutex_unlock(mutex_t *);
1418 extern  int     cancel_safe_mutex_trylock(mutex_t *);
1419 extern  void    _prefork_handler(void);
1420 extern  void    _postfork_parent_handler(void);
1421 extern  void    _postfork_child_handler(void);
1422 extern  void    postfork1_child(void);
1423 extern  void    postfork1_child_aio(void);
1424 extern  void    postfork1_child_sigev_aio(void);
```

```
1425 extern  void    postfork1_child_sigev_mq(void);
1426 extern  void    postfork1_child_sigev_timer(void);
1427 extern  void    postfork1_child_tpool(void);
1428 extern  void    fork_lock_enter(void);
1429 extern  void    fork_lock_exit(void);
1430 extern  void    suspend_fork(void);
1431 extern  void    continue_fork(int);
1432 extern  void    do_sigcancel(void);
1433 extern  void    setup_cancelsig(int);
1434 extern  void    init_sigev_thread(void);
1435 extern  void    init_aio(void);
1436 extern  void    init_progname(void);
1437 extern  void    _cancelon(void);
1438 extern  void    _canceloff(void);
1439 extern  void    _canceloff_nocancel(void);
1440 extern  void    _cancel_prologue(void);
1441 extern  void    _cancel_epilogue(void);
1442 extern  void    no_preempt(ulwp_t *);
1443 extern  void    preempt(ulwp_t *);
1444 extern  void    _thrp_unwind(void *);

1446 extern  pid_t   __forkx(int);
1447 extern  pid_t   __forkallx(int);
1448 extern  int     __open(const char *, int, mode_t);
1449 extern  int     __open64(const char *, int, mode_t);
1450 extern  int     __openat(int, const char *, int, mode_t);
1451 extern  int     __openat64(int, const char *, int, mode_t);
1452 extern  int     __close(int);
1453 extern  ssize_t __read(int, void *, size_t);
1454 extern  ssize_t __write(int, const void *, size_t);
1455 extern  int     __fcntl(int, int, ...);
1456 extern  int     __lwp_continue(lwpid_t);
1457 extern  int     __lwp_create(ucontext_t *, uint_t, lwpid_t *);
1458 extern  int     ___lwp_suspend(lwpid_t);
1459 extern  int     lwp_wait(lwpid_t, lwpid_t *);
1460 extern  int     __lwp_wait(lwpid_t, lwpid_t *);
1461 extern  int     __lwp_detach(lwpid_t);
1462 extern  sc_shared_t *__schedctl(void);

1464 /* actual system call traps */
1465 extern  int     __setcontext(const ucontext_t *);
1466 extern  int     __getcontext(ucontext_t *);
1467 extern  int     __clock_gettime(clockid_t, timespec_t *);
1468 extern  void    abstime_to_reltime(clockid_t, const timespec_t *, timespec_t *);
1469 extern  void    hrt2ts(hrtime_t, timespec_t *);

1471 extern  int     __sigaction(int, const struct sigaction *, struct sigaction *);
1472 extern  int     __sigprocmask(int, const sigset_t *, sigset_t *);
1473 extern  int     __lwp_sigmask(int, const sigset_t *);
1474 extern  void    __sighndlr(int, siginfo_t *, ucontext_t *, void (*)());
1475 extern  caddr_t __sighndlrend;
1476 #pragma unknown_control_flow(__sighndlr)

1478 /* belongs in <pthread.h> */
1479 #define PTHREAD_CREATE_DAEMON_NP        0x100   /* = THR_DAEMON */
1480 #define PTHREAD_CREATE_NONDAEMON_NP     0
1481 extern  int     pthread_attr_setdaemonstate_np(pthread_attr_t *, int);
1482 extern  int     pthread_attr_getdaemonstate_np(const pthread_attr_t *, int *);

1484 extern  int     mutex_held(mutex_t *);
1485 extern  int     mutex_lock_internal(mutex_t *, timespec_t *, int);
1486 extern  int     mutex_unlock_internal(mutex_t *, int);

1488 /* not cancellation points: */
1489 extern  int     __cond_wait(cond_t *, mutex_t *);
1490 extern  int     __cond_timedwait(cond_t *, mutex_t *, const timespec_t *);
```



```
1491 extern  int     __cond_reltimedwait(cond_t *, mutex_t *, const timespec_t *);

1493 extern  int     rw_read_held(rwlock_t *);
1494 extern  int     rw_write_held(rwlock_t *);

1496 extern  int     _thrp_create(void *, size_t, void *(*)(void *), void *, long,
1497                         thread_t *, size_t, const char *);
1495                         thread_t *, size_t);
1498 extern  int     _thrp_suspend(thread_t, uchar_t);
1499 extern  int     _thrp_continue(thread_t, uchar_t);

1501 extern  void    _thrp_terminate(void *);
1502 extern  void    _thrp_exit(void);

1504 extern  const pcclass_t *get_info_by_class(id_t);
1505 extern  const pcclass_t *get_info_by_policy(int);
1506 extern  const thrattr_t *def_thrattr(void);
1507 extern  id_t    setparam(idtype_t, id_t, int, int);
1508 extern  id_t    setprio(idtype_t, id_t, int, int *);
1509 extern  id_t    getparam(idtype_t, id_t, int *, struct sched_param *);

1511 /*
1512  * System call wrappers (direct interfaces to the kernel)
1513  */
1514 extern  int     ___lwp_mutex_register(mutex_t *, mutex_t **);
1515 extern  int     ___lwp_mutex_trylock(mutex_t *, ulwp_t *);
1516 extern  int     ___lwp_mutex_timedlock(mutex_t *, timespec_t *, ulwp_t *);
1517 extern  int     ___lwp_mutex_unlock(mutex_t *);
1518 extern  int     ___lwp_mutex_wakeup(mutex_t *, int);
1519 extern  int     ___lwp_cond_wait(cond_t *, mutex_t *, timespec_t *, int);
1520 extern  int     ___lwp_sema_timedwait(lwp_sema_t *, timespec_t *, int);
1521 extern  int     __lwp_rwlock_rdlock(rwlock_t *, timespec_t *);
1522 extern  int     __lwp_rwlock_wrlock(rwlock_t *, timespec_t *);
1523 extern  int     __lwp_rwlock_tryrdlock(rwlock_t *);
1524 extern  int     __lwp_rwlock_trywrlock(rwlock_t *);
1525 extern  int     __lwp_rwlock_unlock(rwlock_t *);
1526 extern  int     __lwp_park(timespec_t *, lwpid_t);
1527 extern  int     __lwp_unpark(lwpid_t);
1528 extern  int     __lwp_unpark_all(lwpid_t *, int);
1529 #if defined(__x86)
1530 extern  int     ___lwp_private(int, int, void *);
1531 #endif  /* __x86 */

1533 /*
1534  * inlines
1535  */
1536 extern  int             set_lock_byte(volatile uint8_t *);
1537 extern  uint32_t        atomic_swap_32(volatile uint32_t *, uint32_t);
1538 extern  uint32_t        atomic_cas_32(volatile uint32_t *, uint32_t, uint32_t);
1539 extern  void            atomic_inc_32(volatile uint32_t *);
1540 extern  void            atomic_dec_32(volatile uint32_t *);
1541 extern  void            atomic_and_32(volatile uint32_t *, uint32_t);
1542 extern  void            atomic_or_32(volatile uint32_t *, uint32_t);
1543 #if defined(__sparc)
1544 extern  ulong_t         caller(void);
1545 extern  ulong_t         getfp(void);
1546 #endif  /* __sparc */

1548 #include "thr_inlines.h"

1550 #endif  /* _THR_UBERDATA_H */
```

```
**********************************************************
   59457 Mon Oct 15 13:25:13 2018
new/usr/src/lib/libc/port/mapfile-vers
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2018 Nexenta Systems, Inc.
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright 2018 Joyent, Inc.
# Copyright 2016 Joyent, Inc.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright (c) 2013 Gary Mills
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
#

#
# MAPFILE HEADER START
#
# WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
# Object versioning must comply with the rules detailed in
#
#       usr/src/lib/README.mapfiles
#
# You should not be making modifications here until you've read the most current
# copy of that file. If you need help, contact a gatekeeper for guidance.
#
# MAPFILE HEADER END
#

$mapfile_version 2

#
# All function names added to this or any other libc mapfile
# must be placed under the 'protected:' designation.
# The 'global:' designation is used *only* for data
# items and for the members of the malloc() family.
#

# Mnemonic conditional input identifiers:
#
# - amd64, i386, sparc32, sparcv9: Correspond to ISA subdirectories used to
#       hold per-platform code. Note however that we use 'sparc32' instead of
#       'sparc'. Since '_sparc' is predefined to apply to, all sparc platforms,
```



```
#       naming the 32-bit version 'sparc' would be too likely to cause errors.
#
# -     lf64: Defined on platforms that offer the 32-bit largefile APIs
#
$if _ELF32
$add lf64
$endif
$if _sparc && _ELF32
$add sparc32
$endif
$if _sparc && _ELF64
$add sparcv9
$endif
$if _x86 && _ELF32
$add i386
$endif
$if _x86 && _ELF64
$add amd64
$endif

SYMBOL_VERSION ILLUMOS_0.28 {
    protected:
        pthread_attr_getname_np;
        pthread_attr_setname_np;
        pthread_getname_np;
        pthread_setname_np;
        thr_getname;
        thr_setname;
} ILLUMOS_0.27;

SYMBOL_VERSION ILLUMOS_0.27 {   # memset_s(3C) and set_constraint_handler_s(3C)
    protected:
        abort_handler_s;
        ignore_handler_s;
        memset_s;
        set_constraint_handler_s;
} ILLUMOS_0.26;
```

*unchanged_portion_omitted_*

```
**********************************************************
   14697 Mon Oct 15 13:25:14 2018
new/usr/src/lib/libc/port/threads/pthr_attr.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2018, Joyent, Inc.
 * Copyright 2015, Joyent, Inc.
 */

#include "lint.h"
#include "thr_uberdata.h"
#include <sys/ctype.h>
#include <strings.h>
#include <sched.h>

/*
 * Default attribute object for pthread_create() with NULL attr pointer.
 * Note that the 'guardsize' field is initialized on the first call.
 */
const thrattr_t *
def_thrattr(void)
{
        static thrattr_t thrattr = {
                0,                              /* stksize */
                NULL,                           /* stkaddr */
                PTHREAD_CREATE_JOINABLE,        /* detachstate */
                PTHREAD_CREATE_NONDAEMON_NP,    /* daemonstate */
                PTHREAD_SCOPE_PROCESS,          /* scope */
                0,                              /* prio */
                SCHED_OTHER,                    /* policy */
                PTHREAD_INHERIT_SCHED,          /* inherit */
                0,                              /* guardsize */
                { 0 }                           /* name */
                0                               /* guardsize */
        };
        if (thrattr.guardsize == 0)
                thrattr.guardsize = _sysconf(_SC_PAGESIZE);
        return (&thrattr);
}
```



```
}
_____unchanged_portion_omitted_

/*
 * pthread_attr_clone: make a copy of a pthread_attr_t.
 */
int
pthread_attr_clone(pthread_attr_t *attr, const pthread_attr_t *old_attr)
{
        thrattr_t *ap;
        const thrattr_t *old_ap =
            old_attr ? old_attr->__pthread_attrp : def_thrattr();
            old_attr? old_attr->__pthread_attrp : def_thrattr();

        if (old_ap == NULL)
                return (EINVAL);
        if ((ap = lmalloc(sizeof (thrattr_t))) == NULL)
                return (ENOMEM);
        *ap = *old_ap;
        attr->__pthread_attrp = ap;
        return (0);
}

/*
 * pthread_attr_equal: compare two pthread_attr_t's, return 1 if equal.
 * A NULL pthread_attr_t pointer implies default attributes.
 * This is a consolidation-private interface, for librt.
 */
int
pthread_attr_equal(const pthread_attr_t *attr1, const pthread_attr_t *attr2)
{
        const thrattr_t *ap1 = attr1 ? attr1->__pthread_attrp : def_thrattr();
        const thrattr_t *ap2 = attr2 ? attr2->__pthread_attrp : def_thrattr();
        const thrattr_t *ap1 = attr1? attr1->__pthread_attrp : def_thrattr();
        const thrattr_t *ap2 = attr2? attr2->__pthread_attrp : def_thrattr();

        if (ap1 == NULL || ap2 == NULL)
                return (0);
        return (ap1 == ap2 || memcmp(ap1, ap2, sizeof (thrattr_t)) == 0);
}
_____unchanged_portion_omitted_

int
pthread_attr_setname_np(pthread_attr_t *attr, const char *name)
{
        thrattr_t *ap;

        if (attr == NULL || (ap = attr->__pthread_attrp) == NULL)
                return (EINVAL);

        if (name == NULL) {
                bzero(ap->name, sizeof (ap->name));
                return (0);
        }

        if (strlen(name) >= sizeof (ap->name))
                return (ERANGE);

        /*
         * We really want the ASCII version of isprint() here...
         */
        for (size_t i = 0; name[i] != '\0'; i++) {
                if (!ISPRINT(name[i]))
                        return (EINVAL);
        }
```

```
506         /*
507          * not having garbage after the end of the string simplifies attr
508          * comparison
509          */
510         bzero(ap->name, sizeof (ap->name));
511         (void) strlcpy(ap->name, name, sizeof (ap->name));
512         return (0);
513 }

515 int
516 pthread_attr_getname_np(pthread_attr_t *attr, char *buf, size_t len)
517 {
518         thrattr_t *ap;

520         if (buf == NULL || attr == NULL ||
521             (ap = attr->__pthread_attrp) == NULL)
522                 return (EINVAL);

524         if (strlcpy(buf, ap->name, len) > len)
525                 return (ERANGE);
526         return (0);
527 }

529 /*
530  * This function is a common BSD extension to pthread which is used to obtain
531  * the attributes of a thread that might have changed after its creation, for
532  * example, it's stack address.
533  *
534  * Note, there is no setattr analogue, nor do we desire to add one at this time.
535  * Similarly there is no native threads API analogue (nor should we add one for
536  * C11).
537  *
538  * The astute reader may note that there is a GNU version of this called
539  * pthread_getattr_np(). The two functions are similar, but subtley different in
540  * a rather important way. While the pthread_attr_get_np() expects to be given
541  * a pthread_attr_t that has had pthread_attr_init() called on in,
542  * pthread_getattr_np() does not. However, on GNU systems, where the function
543  * originates, the pthread_attr_t is not opaque and thus it is entirely safe to
544  * both call pthread_attr_init() and then call pthread_getattr_np() on the same
545  * attributes object. On illumos, since the pthread_attr_t is opaque, that would
546  * be a memory leak. As such, we don't provide it.
547  */
548 int
549 pthread_attr_get_np(pthread_t tid, pthread_attr_t *attr)
550 {
551         int ret;
552         ulwp_t *self = curthread;
553         uberdata_t *udp = self->ul_uberdata;
554         ulwp_t *target = NULL;
555         thrattr_t *ap;

557         /*
558          * To ensure that information about the target thread does not change or
559          * disappear while we're trying to interrogate it, we grab the uwlp
560          * lock.
561          */
562         if (self->ul_lwpid == tid) {
563                 ulwp_lock(self, udp);
564                 target = self;
565         } else {
566                 target = find_lwp(tid);
567                 if (target == NULL)
568                         return (ESRCH);
569         }

571         if (attr == NULL) {
```



```
572                 ret = EINVAL;
573                 goto out;
574         }

576         if ((ap = attr->__pthread_attrp) == NULL) {
577                 ret = EINVAL;
578                 goto out;
579         }

581         ap->stksize = target->ul_stksiz;
582         ap->stkaddr = target->ul_stk;
583         if (target->ul_usropts & THR_DETACHED) {
584                 ap->detachstate = PTHREAD_CREATE_DETACHED;
585         } else {
586                 ap->detachstate = PTHREAD_CREATE_JOINABLE;
587         }

589         if (target->ul_usropts & THR_DAEMON) {
590                 ap->daemonstate = PTHREAD_CREATE_DAEMON_NP;
591         } else {
592                 ap->daemonstate = PTHREAD_CREATE_NONDAEMON_NP;
593         }

595         if (target->ul_usropts & THR_BOUND) {
596                 ap->scope = PTHREAD_SCOPE_SYSTEM;
597         } else {
598                 ap->scope = PTHREAD_SCOPE_PROCESS;
599         }
600         ap->prio = target->ul_pri;
601         ap->policy = target->ul_policy;
602         ap->inherit = target->ul_ptinherit;
603         ap->guardsize = target->ul_guardsize;
604         (void) pthread_getname_np(tid, ap->name, sizeof (ap->name));

606         ret = 0;
607 out:
608         ulwp_unlock(target, udp);
609         return (ret);
610 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    7429 Mon Oct 15 13:25:14 2018
new/usr/src/lib/libc/port/threads/pthread.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2018 Joyent, Inc.
 * Copyright 2016 Joyent, Inc.
 */

#include "lint.h"
#include "thr_uberdata.h"

/*
 * pthread_once related data
 * This structure is exported as pthread_once_t in pthread.h.
 * We export only the size of this structure. so check
 * pthread_once_t in pthread.h before making a change here.
 */
typedef struct  __once {
        mutex_t mlock;
        union {
                uint32_t        pad32_flag[2];
                uint64_t        pad64_flag;
        } oflag;
} __once_t;
_____unchanged_portion_omitted_

/*
 * pthread_create: creates a thread in the current process.
 * calls common _thrp_create() after copying the attributes.
 */
#pragma weak _pthread_create = pthread_create
int
pthread_create(pthread_t *thread, const pthread_attr_t *attr,
    void * (*start_routine)(void *), void *arg)
{
        ulwp_t          *self = curthread;
        const thrattr_t *ap = attr? attr->__pthread_attrp : def_thrattr();
        const pcclass_t *pccp;
```



```
        long            flag;
        pthread_t       tid;
        int             error;

        update_sched(self);

        if (ap == NULL)
                return (EINVAL);

        /* validate explicit scheduling attributes */
        if (ap->inherit == PTHREAD_EXPLICIT_SCHED &&
            (ap->policy == SCHED_SYS ||
            (pccp = get_info_by_policy(ap->policy)) == NULL ||
            ap->prio < pccp->pcc_primin || ap->prio > pccp->pcc_primax))
                return (EINVAL);

        flag = ap->scope | ap->detachstate | ap->daemonstate | THR_SUSPENDED;
        error = _thrp_create(ap->stkaddr, ap->stksize, start_routine, arg,
            flag, &tid, ap->guardsize, ap->name);
            flag, &tid, ap->guardsize);
        if (error == 0) {
                /*
                 * Record the original inheritence value for
                 * pthread_getattr_np(). We should always be able to find the
                 * thread.
                 */
                (void) _thr_setinherit(tid, ap->inherit);

                if (ap->inherit == PTHREAD_EXPLICIT_SCHED &&
                    (ap->policy != self->ul_policy ||
                    ap->prio != (self->ul_epri ? self->ul_epri :
                    self->ul_pri))) {
                        /*
                         * The SUSv3 specification requires pthread_create()
                         * to fail with EPERM if it cannot set the scheduling
                         * policy and parameters on the new thread.
                         */
                        error = _thr_setparam(tid, ap->policy, ap->prio);
                }

                if (error) {
                        /*
                         * We couldn't determine this error before
                         * actually creating the thread.  To recover,
                         * mark the thread detached and cancel it.
                         * It is as though it was never created.
                         */
                        ulwp_t *ulwp = find_lwp(tid);
                        if (ulwp->ul_detached == 0) {
                                ulwp->ul_detached = 1;
                                ulwp->ul_usropts |= THR_DETACHED;
                                (void) __lwp_detach(tid);
                        }
                        ulwp->ul_cancel_pending = 2; /* cancelled on creation */
                        ulwp->ul_cancel_disabled = 0;
                        ulwp_unlock(ulwp, self->ul_uberdata);
                } else if (thread) {
                        *thread = tid;
                }
                (void) thr_continue(tid);
        }

        /* posix version expects EAGAIN for lack of memory */
        if (error == ENOMEM)
                error = EAGAIN;
        return (error);
```

```
 178 }
______unchanged_portion_omitted_
```

```
**********************************************************
   77121 Mon Oct 15 13:25:14 2018
new/usr/src/lib/libc/port/threads/thr.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2016 by Delphix. All rights reserved.
 * Copyright (c) 2017 by The MathWorks, Inc. All rights reserved.
 */
/*
 * Copyright 2018 Joyent, Inc.
 * Copyright 2016 Joyent, Inc.
 */

#include "lint.h"
#include "thr_uberdata.h"
#include <pthread.h>
#include <procfs.h>
#include <sys/uio.h>
#include <ctype.h>
#include "libc.h"

/*
 * These symbols should not be exported from libc, but
 * /lib/libm.so.2 references _thr_main.  libm needs to be fixed.
 * Also, some older versions of the Studio compiler/debugger
 * components reference them.  These need to be fixed, too.
 */
#pragma weak _thr_main = thr_main
#pragma weak _thr_create = thr_create
#pragma weak _thr_join = thr_join
#pragma weak _thr_self = thr_self

#undef errno
extern int errno;

/*
 * Between Solaris 2.5 and Solaris 9, __threaded was used to indicate
 * "we are linked with libthread".  The Sun Workshop 6 update 1 compilation
 * system used it illegally (it is a consolidation private symbol).
 * To accommodate this and possibly other abusers of the symbol,
 * we make it always equal to 1 now that libthread has been folded
 * into libc.  The new __libc_threaded symbol is used to indicate
```



```
 * the new meaning, "more than one thread exists".
 */
int __threaded = 1;             /* always equal to 1 */
int __libc_threaded = 0;        /* zero until first thr_create() */

/*
 * thr_concurrency and pthread_concurrency are not used by the library.
 * They exist solely to hold and return the values set by calls to
 * thr_setconcurrency() and pthread_setconcurrency().
 * Because thr_concurrency is affected by the THR_NEW_LWP flag
 * to thr_create(), thr_concurrency is protected by link_lock.
 */
static  int     thr_concurrency = 1;
static  int     pthread_concurrency;

#define HASHTBLSZ       1024    /* must be a power of two */
#define TIDHASH(tid, udp)       (tid & (udp)->hash_mask)

/* initial allocation, just enough for one lwp */
#pragma align 64(init_hash_table)
thr_hash_table_t init_hash_table[1] = {
        { DEFAULTMUTEX, DEFAULTCV, NULL },
};
```

_____***unchanged_portion_omitted_***

```
int
_thrp_create(void *stk, size_t stksize, void *(*func)(void *), void *arg,
    long flags, thread_t *new_thread, size_t guardsize, const char *name)
    long flags, thread_t *new_thread, size_t guardsize)
{
        ulwp_t *self = curthread;
        uberdata_t *udp = self->ul_uberdata;
        ucontext_t uc;
        uint_t lwp_flags;
        thread_t tid;
        int error;
        ulwp_t *ulwp;

        /*
         * Enforce the restriction of not creating any threads
         * until the primary link map has been initialized.
         * Also, disallow thread creation to a child of vfork().
         */
        if (!self->ul_primarymap || self->ul_vfork)
                return (ENOTSUP);

        if (udp->hash_size == 1)
                finish_init();

        if ((stk || stksize) && stksize < MINSTACK)
                return (EINVAL);

        if (stk == NULL) {
                if ((ulwp = find_stack(stksize, guardsize)) == NULL)
                        return (ENOMEM);
                stksize = ulwp->ul_mapsiz - ulwp->ul_guardsize;
        } else {
                /* initialize the private stack */
                if ((ulwp = ulwp_alloc()) == NULL)
                        return (ENOMEM);
                ulwp->ul_stk = stk;
                ulwp->ul_stktop = (uintptr_t)stk + stksize;
                ulwp->ul_stksiz = stksize;
        }
        /* ulwp is not in the hash table; make sure hash_out() doesn't fail */
        ulwp->ul_ix = -1;
```

```
601         ulwp->ul_errnop = &ulwp->ul_errno;

603         lwp_flags = LWP_SUSPENDED;
604         if (flags & (THR_DETACHED|THR_DAEMON)) {
605                 flags |= THR_DETACHED;
606                 lwp_flags |= LWP_DETACHED;
607         }
608         if (flags & THR_DAEMON)
609                 lwp_flags |= LWP_DAEMON;

611         /* creating a thread: enforce mt-correctness in mutex_lock() */
612         self->ul_async_safe = 1;

614         /* per-thread copies of global variables, for speed */
615         ulwp->ul_queue_fifo = self->ul_queue_fifo;
616         ulwp->ul_cond_wait_defer = self->ul_cond_wait_defer;
617         ulwp->ul_error_detection = self->ul_error_detection;
618         ulwp->ul_async_safe = self->ul_async_safe;
619         ulwp->ul_max_spinners = self->ul_max_spinners;
620         ulwp->ul_adaptive_spin = self->ul_adaptive_spin;
621         ulwp->ul_queue_spin = self->ul_queue_spin;
622         ulwp->ul_door_noreserve = self->ul_door_noreserve;
623         ulwp->ul_misaligned = self->ul_misaligned;

625         /* new thread inherits creating thread's scheduling parameters */
626         ulwp->ul_policy = self->ul_policy;
627         ulwp->ul_pri = (self->ul_epri? self->ul_epri : self->ul_pri);
628         ulwp->ul_cid = self->ul_cid;
629         ulwp->ul_rtclassid = self->ul_rtclassid;

631         ulwp->ul_primarymap = self->ul_primarymap;
632         ulwp->ul_self = ulwp;
633         ulwp->ul_uberdata = udp;

635         /* debugger support */
636         ulwp->ul_usropts = flags;

638 #ifdef __sparc
639         /*
640          * We cache several instructions in the thread structure for use
641          * by the fasttrap DTrace provider. When changing this, read the
642          * comment in fasttrap.h for the all the other places that must
643          * be changed.
644          */
645         ulwp->ul_dsave = 0x9de04000;    /* save %g1, %g0, %sp */
646         ulwp->ul_drestore = 0x81e80000; /* restore %g0, %g0, %g0 */
647         ulwp->ul_dftret = 0x91d0203a;   /* ta 0x3a */
648         ulwp->ul_dreturn = 0x81ca0000;  /* return %o0 */
649 #endif

651         ulwp->ul_startpc = func;
652         ulwp->ul_startarg = arg;
653         _fpinherit(ulwp);
654         /*
655          * Defer signals on the new thread until its TLS constructors
656          * have been called.  _thrp_setup() will call sigon() after
657          * it has called tls_setup().
658          */
659         ulwp->ul_sigdefer = 1;

661         error = setup_context(&uc, _thrp_setup, ulwp,
662             (caddr_t)ulwp->ul_stk + ulwp->ul_guardsize, stksize);
663         if (error != 0 && stk != NULL)  /* inaccessible stack */
664                 error = EFAULT;

666         /*
```



```
667          * Call enter_critical() to avoid being suspended until we
668          * have linked the new thread into the proper lists.
669          * This is necessary because forkall() and fork1() must
670          * suspend all threads and they must see a complete list.
671          */
672         enter_critical(self);
673         uc.uc_sigmask = ulwp->ul_sigmask = self->ul_sigmask;
674         if (error != 0 ||
675             (error = __lwp_create(&uc, lwp_flags, &tid)) != 0) {
676                 exit_critical(self);
677                 ulwp->ul_lwpid = (lwpid_t)(-1);
678                 ulwp->ul_dead = 1;
679                 ulwp->ul_detached = 1;
680                 lmutex_lock(&udp->link_lock);
681                 ulwp_free(ulwp);
682                 lmutex_unlock(&udp->link_lock);
683                 return (error);
684         }
685         self->ul_nocancel = 0;  /* cancellation is now possible */
686         udp->uberflags.uf_mt = 1;
687         if (new_thread)
688                 *new_thread = tid;
689         if (flags & THR_DETACHED)
690                 ulwp->ul_detached = 1;
691         ulwp->ul_lwpid = tid;
692         ulwp->ul_stop = TSTP_REGULAR;
693         if (flags & THR_SUSPENDED)
694                 ulwp->ul_created = 1;

696         lmutex_lock(&udp->link_lock);
697         ulwp->ul_forw = udp->all_lwps;
698         ulwp->ul_back = udp->all_lwps->ul_back;
699         ulwp->ul_back->ul_forw = ulwp;
700         ulwp->ul_forw->ul_back = ulwp;
701         hash_in(ulwp, udp);
702         udp->nthreads++;
703         if (flags & THR_DAEMON)
704                 udp->ndaemons++;
705         if (flags & THR_NEW_LWP)
706                 thr_concurrency++;
707         __libc_threaded = 1;            /* inform stdio */
708         lmutex_unlock(&udp->link_lock);

710         if (__td_event_report(self, TD_CREATE, udp)) {
711                 self->ul_td_evbuf.eventnum = TD_CREATE;
712                 self->ul_td_evbuf.eventdata = (void *)(uintptr_t)tid;
713                 tdb_event(TD_CREATE, udp);
714         }

716         exit_critical(self);

718         if (name != NULL)
719                 (void) pthread_setname_np(tid, name);

721         if (!(flags & THR_SUSPENDED))
722                 (void) _thrp_continue(tid, TSTP_REGULAR);

724         return (0);
725 }

727 int
728 thr_create(void *stk, size_t stksize, void *(*func)(void *), void *arg,
729     long flags, thread_t *new_thread)
730 {
731         return (_thrp_create(stk, stksize, func, arg, flags, new_thread, 0,
732             NULL));
```

```
 728         return (_thrp_create(stk, stksize, func, arg, flags, new_thread, 0));
 733 }
_____unchanged_portion_omitted_

2405 /* "/proc/self/lwp/%u/lwpname" w/o stdio */
2406 static void
2407 lwpname_path(pthread_t tid, char *buf, size_t bufsize)
2408 {
2409         (void) strlcpy(buf, "/proc/self/lwp/", bufsize);
2410         ultos((uint64_t)tid, 10, buf + strlen(buf));
2411         (void) strlcat(buf, "/lwpname", bufsize);
2412 }

2414 #pragma weak pthread_setname_np = thr_setname
2415 int
2416 thr_setname(pthread_t tid, const char *name)
2417 {
2418         extern ssize_t __write(int, const void *, size_t);
2419         char path[PATH_MAX];
2420         int saved_errno;
2421         size_t len;
2422         ssize_t n;
2423         int fd;

2425         if (name == NULL)
2426                 name = "";

2428         len = strlen(name) + 1;
2429         if (len > THREAD_NAME_MAX)
2430                 return (ERANGE);

2432         lwpname_path(tid, path, sizeof (path));

2434         if ((fd = __open(path, O_WRONLY, 0)) < 0) {
2435                 if (errno == ENOENT)
2436                         errno = ESRCH;
2437                 return (errno);
2438         }

2440         n = __write(fd, name, len);
2441         saved_errno = errno;
2442         (void) __close(fd);

2444         if (n < 0)
2445                 return (saved_errno);
2446         if (n != len)
2447                 return (EFAULT);
2448         return (0);
2449 }

2451 #pragma weak pthread_getname_np = thr_getname
2452 int
2453 thr_getname(pthread_t tid, char *buf, size_t bufsize)
2454 {
2455         extern ssize_t __read(int, void *, size_t);
2456         char name[THREAD_NAME_MAX];
2457         char path[PATH_MAX];
2458         int saved_errno;
2459         ssize_t n;
2460         int fd;

2462         if (buf == NULL)
2463                 return (EINVAL);

2465         lwpname_path(tid, path, sizeof (path));
```



```
2467         if ((fd = __open(path, O_RDONLY, 0)) < 0) {
2468                 if (errno == ENOENT)
2469                         errno = ESRCH;
2470                 return (errno);
2471         }

2473         n = __read(fd, name, sizeof (name));
2474         saved_errno = errno;
2475         (void) __close(fd);

2477         if (n < 0)
2478                 return (saved_errno);
2479         if (n != sizeof (name))
2480                 return (EFAULT);
2481         if (strlcpy(buf, name, bufsize) >= bufsize)
2482                 return (ERANGE);
2483         return (0);
2484 }

2486 /*
2487  * XXX
2488  * The remainder of this file implements the private interfaces to java for
2489  * garbage collection.  It is no longer used, at least by java 1.2.
2490  * It can all go away once all old JVMs have disappeared.
2491  */

2493 int     suspendingallmutators;  /* when non-zero, suspending all mutators. */
2494 int     suspendedallmutators;   /* when non-zero, all mutators suspended. */
2495 int     mutatorsbarrier;        /* when non-zero, mutators barrier imposed. */
2496 mutex_t mutatorslock = DEFAULTMUTEX;    /* used to enforce mutators barrier. */
2497 cond_t  mutatorscv = DEFAULTCV;         /* where non-mutators sleep. */

2499 /*
2500  * Get the available register state for the target thread.
2501  * Return non-volatile registers: TRS_NONVOLATILE
2502  */
2503 #pragma weak _thr_getstate = thr_getstate
2504 int
2505 thr_getstate(thread_t tid, int *flag, lwpid_t *lwp, stack_t *ss, gregset_t rs)
2506 {
2507         ulwp_t *self = curthread;
2508         uberdata_t *udp = self->ul_uberdata;
2509         ulwp_t **ulwpp;
2510         ulwp_t *ulwp;
2511         int error = 0;
2512         int trs_flag = TRS_LWPID;

2514         if (tid == 0 || self->ul_lwpid == tid) {
2515                 ulwp = self;
2516                 ulwp_lock(ulwp, udp);
2517         } else if ((ulwpp = find_lwpp(tid)) != NULL) {
2518                 ulwp = *ulwpp;
2519         } else {
2520                 if (flag)
2521                         *flag = TRS_INVALID;
2522                 return (ESRCH);
2523         }

2525         if (ulwp->ul_dead) {
2526                 trs_flag = TRS_INVALID;
2527         } else if (!ulwp->ul_stop && !suspendedallmutators) {
2528                 error = EINVAL;
2529                 trs_flag = TRS_INVALID;
2530         } else if (ulwp->ul_stop) {
2531                 trs_flag = TRS_NONVOLATILE;
2532                 getgregs(ulwp, rs);
```

```
2533         }

2535         if (flag)
2536                 *flag = trs_flag;
2537         if (lwp)
2538                 *lwp = tid;
2539         if (ss != NULL)
2540                 (void) _thrp_stksegment(ulwp, ss);

2542         ulwp_unlock(ulwp, udp);
2543         return (error);
2544 }
```

unchanged_portion_omitted_

```
**********************************************************
   54655 Mon Oct 15 13:25:14 2018
new/usr/src/lib/libdtrace/common/dt_open.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2018, Joyent, Inc.
 * Copyright (c) 2017, Joyent, Inc.
 * Copyright (c) 2012, 2016 by Delphix. All rights reserved.
 */

#include <sys/types.h>
#include <sys/modctl.h>
#include <sys/systeminfo.h>
#include <sys/resource.h>

#include <libelf.h>
#include <strings.h>
#include <alloca.h>
#include <limits.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>
#include <assert.h>

#define _POSIX_PTHREAD_SEMANTICS
#include <dirent.h>
#undef  _POSIX_PTHREAD_SEMANTICS

#include <dt_impl.h>
#include <dt_program.h>
#include <dt_module.h>
#include <dt_printf.h>
#include <dt_string.h>
#include <dt_provider.h>

/*
 * Stability and versioning definitions.  These #defines are used in the tables
 * of identifiers below to fill in the attribute and version fields associated
 * with each identifier.  The DT_ATTR_* macros are a convenience to permit more
 * concise declarations of common attributes such as Stable/Stable/Common.  The
```



```
 * DT_VERS_* macros declare the encoded integer values of all versions used so
 * far.  DT_VERS_LATEST must correspond to the latest version value among all
 * versions exported by the D compiler.  DT_VERS_STRING must be an ASCII string
 * that contains DT_VERS_LATEST within it along with any suffixes (e.g. Beta).
 * You must update DT_VERS_LATEST and DT_VERS_STRING when adding a new version,
 * and then add the new version to the _dtrace_versions[] array declared below.
 * Refer to the Solaris Dynamic Tracing Guide Stability and Versioning chapters
 * respectively for an explanation of these DTrace features and their values.
 *
 * NOTE: Although the DTrace versioning scheme supports the labeling and
 *       introduction of incompatible changes (e.g. dropping an interface in a
 *       major release), the libdtrace code does not currently support this.
 *       All versions are assumed to strictly inherit from one another.  If
 *       we ever need to provide divergent interfaces, this will need work.
 */
#define DT_ATTR_STABCMN { DTRACE_STABILITY_STABLE, \
        DTRACE_STABILITY_STABLE, DTRACE_CLASS_COMMON }

#define DT_ATTR_EVOLCMN { DTRACE_STABILITY_EVOLVING, \
        DTRACE_STABILITY_EVOLVING, DTRACE_CLASS_COMMON \
}

/*
 * The version number should be increased for every customer visible release
 * of DTrace. The major number should be incremented when a fundamental
 * change has been made that would affect all consumers, and would reflect
 * sweeping changes to DTrace or the D language. The minor number should be
 * incremented when a change is introduced that could break scripts that had
 * previously worked; for example, adding a new built-in variable could break
 * a script which was already using that identifier. The micro number should
 * be changed when introducing functionality changes or major bug fixes that
 * do not affect backward compatibility -- this is merely to make capabilities
 * easily determined from the version number. Minor bugs do not require any
 * modification to the version number.
 */
#define DT_VERS_1_0     DT_VERSION_NUMBER(1, 0, 0)
#define DT_VERS_1_1     DT_VERSION_NUMBER(1, 1, 0)
#define DT_VERS_1_2     DT_VERSION_NUMBER(1, 2, 0)
#define DT_VERS_1_2_1   DT_VERSION_NUMBER(1, 2, 1)
#define DT_VERS_1_2_2   DT_VERSION_NUMBER(1, 2, 2)
#define DT_VERS_1_3     DT_VERSION_NUMBER(1, 3, 0)
#define DT_VERS_1_4     DT_VERSION_NUMBER(1, 4, 0)
#define DT_VERS_1_4_1   DT_VERSION_NUMBER(1, 4, 1)
#define DT_VERS_1_5     DT_VERSION_NUMBER(1, 5, 0)
#define DT_VERS_1_6     DT_VERSION_NUMBER(1, 6, 0)
#define DT_VERS_1_6_1   DT_VERSION_NUMBER(1, 6, 1)
#define DT_VERS_1_6_2   DT_VERSION_NUMBER(1, 6, 2)
#define DT_VERS_1_6_3   DT_VERSION_NUMBER(1, 6, 3)
#define DT_VERS_1_7     DT_VERSION_NUMBER(1, 7, 0)
#define DT_VERS_1_7_1   DT_VERSION_NUMBER(1, 7, 1)
#define DT_VERS_1_8     DT_VERSION_NUMBER(1, 8, 0)
#define DT_VERS_1_8_1   DT_VERSION_NUMBER(1, 8, 1)
#define DT_VERS_1_9     DT_VERSION_NUMBER(1, 9, 0)
#define DT_VERS_1_9_1   DT_VERSION_NUMBER(1, 9, 1)
#define DT_VERS_1_10    DT_VERSION_NUMBER(1, 10, 0)
#define DT_VERS_1_11    DT_VERSION_NUMBER(1, 11, 0)
#define DT_VERS_1_12    DT_VERSION_NUMBER(1, 12, 0)
#define DT_VERS_1_12_1  DT_VERSION_NUMBER(1, 12, 1)
#define DT_VERS_1_13    DT_VERSION_NUMBER(1, 13, 0)
#define DT_VERS_1_14    DT_VERSION_NUMBER(1, 14, 0)
#define DT_VERS_LATEST  DT_VERS_1_14
#define DT_VERS_STRING  "Sun D 1.14"
#define DT_VERS_LATEST  DT_VERS_1_13
#define DT_VERS_STRING  "Sun D 1.13"

const dt_version_t _dtrace_versions[] = {
```

```
124         DT_VERS_1_0,    /* D API 1.0.0 (PSARC 2001/466) Solaris 10 FCS */
125         DT_VERS_1_1,    /* D API 1.1.0 Solaris Express 6/05 */
126         DT_VERS_1_2,    /* D API 1.2.0 Solaris 10 Update 1 */
127         DT_VERS_1_2_1,  /* D API 1.2.1 Solaris Express 4/06 */
128         DT_VERS_1_2_2,  /* D API 1.2.2 Solaris Express 6/06 */
129         DT_VERS_1_3,    /* D API 1.3 Solaris Express 10/06 */
130         DT_VERS_1_4,    /* D API 1.4 Solaris Express 2/07 */
131         DT_VERS_1_4_1,  /* D API 1.4.1 Solaris Express 4/07 */
132         DT_VERS_1_5,    /* D API 1.5 Solaris Express 7/07 */
133         DT_VERS_1_6,    /* D API 1.6 */
134         DT_VERS_1_6_1,  /* D API 1.6.1 */
135         DT_VERS_1_6_2,  /* D API 1.6.2 */
136         DT_VERS_1_6_3,  /* D API 1.6.3 */
137         DT_VERS_1_7,    /* D API 1.7 */
138         DT_VERS_1_7_1,  /* D API 1.7.1 */
139         DT_VERS_1_8,    /* D API 1.8 */
140         DT_VERS_1_8_1,  /* D API 1.8.1 */
141         DT_VERS_1_9,    /* D API 1.9 */
142         DT_VERS_1_9_1,  /* D API 1.9.1 */
143         DT_VERS_1_10,   /* D API 1.10 */
144         DT_VERS_1_11,   /* D API 1.11 */
145         DT_VERS_1_12,   /* D API 1.12 */
146         DT_VERS_1_12_1, /* D API 1.12.1 */
147         DT_VERS_1_13,   /* D API 1.13 */
148         DT_VERS_1_14,   /* D API 1.14 */
149         0
150 };

152 /*
153  * Table of global identifiers.  This is used to populate the global identifier
154  * hash when a new dtrace client open occurs.  For more info see dt_ident.h.
155  * The global identifiers that represent functions use the dt_idops_func ops
156  * and specify the private data pointer as a prototype string which is parsed
157  * when the identifier is first encountered.  These prototypes look like ANSI
158  * C function prototypes except that the special symbol "@" can be used as a
159  * wildcard to represent a single parameter of any type (i.e. any dt_node_t).
160  * The standard "..." notation can also be used to represent varargs.  An empty
161  * parameter list is taken to mean void (that is, no arguments are permitted).
162  * A parameter enclosed in square brackets (e.g. "[int]") denotes an optional
163  * argument.
164  */
165 static const dt_ident_t _dtrace_globals[] = {
166 { "alloca", DT_IDENT_FUNC, 0, DIF_SUBR_ALLOCA, DT_ATTR_STABCMN, DT_VERS_1_0,
167         &dt_idops_func, "void *(size_t)" },
168 { "arg0", DT_IDENT_SCALAR, 0, DIF_VAR_ARG0, DT_ATTR_STABCMN, DT_VERS_1_0,
169         &dt_idops_type, "int64_t" },
170 { "arg1", DT_IDENT_SCALAR, 0, DIF_VAR_ARG1, DT_ATTR_STABCMN, DT_VERS_1_0,
171         &dt_idops_type, "int64_t" },
172 { "arg2", DT_IDENT_SCALAR, 0, DIF_VAR_ARG2, DT_ATTR_STABCMN, DT_VERS_1_0,
173         &dt_idops_type, "int64_t" },
174 { "arg3", DT_IDENT_SCALAR, 0, DIF_VAR_ARG3, DT_ATTR_STABCMN, DT_VERS_1_0,
175         &dt_idops_type, "int64_t" },
176 { "arg4", DT_IDENT_SCALAR, 0, DIF_VAR_ARG4, DT_ATTR_STABCMN, DT_VERS_1_0,
177         &dt_idops_type, "int64_t" },
178 { "arg5", DT_IDENT_SCALAR, 0, DIF_VAR_ARG5, DT_ATTR_STABCMN, DT_VERS_1_0,
179         &dt_idops_type, "int64_t" },
180 { "arg6", DT_IDENT_SCALAR, 0, DIF_VAR_ARG6, DT_ATTR_STABCMN, DT_VERS_1_0,
181         &dt_idops_type, "int64_t" },
182 { "arg7", DT_IDENT_SCALAR, 0, DIF_VAR_ARG7, DT_ATTR_STABCMN, DT_VERS_1_0,
183         &dt_idops_type, "int64_t" },
184 { "arg8", DT_IDENT_SCALAR, 0, DIF_VAR_ARG8, DT_ATTR_STABCMN, DT_VERS_1_0,
185         &dt_idops_type, "int64_t" },
186 { "arg9", DT_IDENT_SCALAR, 0, DIF_VAR_ARG9, DT_ATTR_STABCMN, DT_VERS_1_0,
187         &dt_idops_type, "int64_t" },
188 { "args", DT_IDENT_ARRAY, 0, DIF_VAR_ARGS, DT_ATTR_STABCMN, DT_VERS_1_0,
189         &dt_idops_args, NULL },
```



```
190 { "avg", DT_IDENT_AGGFUNC, 0, DTRACEAGG_AVG, DT_ATTR_STABCMN, DT_VERS_1_0,
191         &dt_idops_func, "void(@)" },
192 { "basename", DT_IDENT_FUNC, 0, DIF_SUBR_BASENAME, DT_ATTR_STABCMN, DT_VERS_1_0,
193         &dt_idops_func, "string(const char *)" },
194 { "bcopy", DT_IDENT_FUNC, 0, DIF_SUBR_BCOPY, DT_ATTR_STABCMN, DT_VERS_1_0,
195         &dt_idops_func, "void(void *, void *, size_t)" },
196 { "breakpoint", DT_IDENT_ACTFUNC, 0, DT_ACT_BREAKPOINT,
197         DT_ATTR_STABCMN, DT_VERS_1_0,
198         &dt_idops_func, "void()" },
199 { "caller", DT_IDENT_SCALAR, 0, DIF_VAR_CALLER, DT_ATTR_STABCMN, DT_VERS_1_0,
200         &dt_idops_type, "uintptr_t" },
201 { "chill", DT_IDENT_ACTFUNC, 0, DT_ACT_CHILL, DT_ATTR_STABCMN, DT_VERS_1_0,
202         &dt_idops_func, "void(int)" },
203 { "cleanpath", DT_IDENT_FUNC, 0, DIF_SUBR_CLEANPATH, DT_ATTR_STABCMN,
204         DT_VERS_1_0, &dt_idops_func, "string(const char *)" },
205 { "clear", DT_IDENT_ACTFUNC, 0, DT_ACT_CLEAR, DT_ATTR_STABCMN, DT_VERS_1_0,
206         &dt_idops_func, "void(...)" },
207 { "commit", DT_IDENT_ACTFUNC, 0, DT_ACT_COMMIT, DT_ATTR_STABCMN, DT_VERS_1_0,
208         &dt_idops_func, "void(int)" },
209 { "copyin", DT_IDENT_FUNC, 0, DIF_SUBR_COPYIN, DT_ATTR_STABCMN, DT_VERS_1_0,
210         &dt_idops_func, "void *(uintptr_t, size_t)" },
211 { "copyinstr", DT_IDENT_FUNC, 0, DIF_SUBR_COPYINSTR,
212         DT_ATTR_STABCMN, DT_VERS_1_0,
213         &dt_idops_func, "string(uintptr_t, [size_t])" },
214 { "copyinto", DT_IDENT_FUNC, 0, DIF_SUBR_COPYINTO, DT_ATTR_STABCMN,
215         DT_VERS_1_0, &dt_idops_func, "void(uintptr_t, size_t, void *)" },
216 { "copyout", DT_IDENT_FUNC, 0, DIF_SUBR_COPYOUT, DT_ATTR_STABCMN, DT_VERS_1_0,
217         &dt_idops_func, "void(void *, uintptr_t, size_t)" },
218 { "copyoutstr", DT_IDENT_FUNC, 0, DIF_SUBR_COPYOUTSTR,
219         DT_ATTR_STABCMN, DT_VERS_1_0,
220         &dt_idops_func, "void(char *, uintptr_t, size_t)" },
221 { "count", DT_IDENT_AGGFUNC, 0, DTRACEAGG_COUNT, DT_ATTR_STABCMN, DT_VERS_1_0,
222         &dt_idops_func, "void()" },
223 { "curthread", DT_IDENT_SCALAR, 0, DIF_VAR_CURTHREAD,
224         { DTRACE_STABILITY_STABLE, DTRACE_STABILITY_PRIVATE,
225         DTRACE_CLASS_COMMON }, DT_VERS_1_0,
226         &dt_idops_type, "genunix`kthread_t *" },
227 { "ddi_pathname", DT_IDENT_FUNC, 0, DIF_SUBR_DDI_PATHNAME,
228         DT_ATTR_EVOLCMN, DT_VERS_1_0,
229         &dt_idops_func, "string(void *, int64_t)" },
230 { "denormalize", DT_IDENT_ACTFUNC, 0, DT_ACT_DENORMALIZE, DT_ATTR_STABCMN,
231         DT_VERS_1_0, &dt_idops_func, "void(...)" },
232 { "dirname", DT_IDENT_FUNC, 0, DIF_SUBR_DIRNAME, DT_ATTR_STABCMN, DT_VERS_1_0,
233         &dt_idops_func, "string(const char *)" },
234 { "discard", DT_IDENT_ACTFUNC, 0, DT_ACT_DISCARD, DT_ATTR_STABCMN, DT_VERS_1_0,
235         &dt_idops_func, "void(int)" },
236 { "epid", DT_IDENT_SCALAR, 0, DIF_VAR_EPID, DT_ATTR_STABCMN, DT_VERS_1_0,
237         &dt_idops_type, "uint_t" },
238 { "errno", DT_IDENT_SCALAR, 0, DIF_VAR_ERRNO, DT_ATTR_STABCMN, DT_VERS_1_0,
239         &dt_idops_type, "int" },
240 { "execname", DT_IDENT_SCALAR, 0, DIF_VAR_EXECNAME,
241         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
242 { "exit", DT_IDENT_ACTFUNC, 0, DT_ACT_EXIT, DT_ATTR_STABCMN, DT_VERS_1_0,
243         &dt_idops_func, "void(int)" },
244 { "freopen", DT_IDENT_ACTFUNC, 0, DT_ACT_FREOPEN, DT_ATTR_STABCMN,
245         DT_VERS_1_1, &dt_idops_func, "void(@, ...)" },
246 { "ftruncate", DT_IDENT_ACTFUNC, 0, DT_ACT_FTRUNCATE, DT_ATTR_STABCMN,
247         DT_VERS_1_0, &dt_idops_func, "void()" },
248 { "func", DT_IDENT_ACTFUNC, 0, DT_ACT_SYM, DT_ATTR_STABCMN,
249         DT_VERS_1_2, &dt_idops_func, "_symaddr(uintptr_t)" },
250 { "getmajor", DT_IDENT_FUNC, 0, DIF_SUBR_GETMAJOR,
251         DT_ATTR_EVOLCMN, DT_VERS_1_0,
252         &dt_idops_func, "genunix`major_t(genunix`dev_t)" },
253 { "getminor", DT_IDENT_FUNC, 0, DIF_SUBR_GETMINOR,
254         DT_ATTR_EVOLCMN, DT_VERS_1_0,
255         &dt_idops_func, "genunix`minor_t(genunix`dev_t)" },
```

```
256 { "htonl", DT_IDENT_FUNC, 0, DIF_SUBR_HTONL, DT_ATTR_EVOLCMN, DT_VERS_1_3,
257         &dt_idops_func, "uint32_t(uint32_t)" },
258 { "htonll", DT_IDENT_FUNC, 0, DIF_SUBR_HTONLL, DT_ATTR_EVOLCMN, DT_VERS_1_3,
259         &dt_idops_func, "uint64_t(uint64_t)" },
260 { "htons", DT_IDENT_FUNC, 0, DIF_SUBR_HTONS, DT_ATTR_EVOLCMN, DT_VERS_1_3,
261         &dt_idops_func, "uint16_t(uint16_t)" },
262 { "getf", DT_IDENT_FUNC, 0, DIF_SUBR_GETF, DT_ATTR_STABCMN, DT_VERS_1_10,
263         &dt_idops_func, "file_t *(int)" },
264 { "gid", DT_IDENT_SCALAR, 0, DIF_VAR_GID, DT_ATTR_STABCMN, DT_VERS_1_0,
265         &dt_idops_type, "gid_t" },
266 { "id", DT_IDENT_SCALAR, 0, DIF_VAR_ID, DT_ATTR_STABCMN, DT_VERS_1_0,
267         &dt_idops_type, "uint_t" },
268 { "index", DT_IDENT_FUNC, 0, DIF_SUBR_INDEX, DT_ATTR_STABCMN, DT_VERS_1_1,
269         &dt_idops_func, "int(const char *, const char *, [int])" },
270 { "inet_ntoa", DT_IDENT_FUNC, 0, DIF_SUBR_INET_NTOA, DT_ATTR_STABCMN,
271         DT_VERS_1_5, &dt_idops_func, "string(ipaddr_t *)" },
272 { "inet_ntoa6", DT_IDENT_FUNC, 0, DIF_SUBR_INET_NTOA6, DT_ATTR_STABCMN,
273         DT_VERS_1_5, &dt_idops_func, "string(in6_addr_t *)" },
274 { "inet_ntop", DT_IDENT_FUNC, 0, DIF_SUBR_INET_NTOP, DT_ATTR_STABCMN,
275         DT_VERS_1_5, &dt_idops_func, "string(int, void *)" },
276 { "ipl", DT_IDENT_SCALAR, 0, DIF_VAR_IPL, DT_ATTR_STABCMN, DT_VERS_1_0,
277         &dt_idops_type, "uint_t" },
278 { "json", DT_IDENT_FUNC, 0, DIF_SUBR_JSON, DT_ATTR_STABCMN, DT_VERS_1_11,
279         &dt_idops_func, "string(const char *, const char *)" },
280 { "jstack", DT_IDENT_ACTFUNC, 0, DT_ACT_JSTACK, DT_ATTR_STABCMN, DT_VERS_1_0,
281         &dt_idops_func, "stack(...)" },
282 { "lltostr", DT_IDENT_FUNC, 0, DIF_SUBR_LLTOSTR, DT_ATTR_STABCMN, DT_VERS_1_0,
283         &dt_idops_func, "string(int64_t, [int])" },
284 { "llquantize", DT_IDENT_AGGFUNC, 0, DTRACEAGG_LLQUANTIZE, DT_ATTR_STABCMN,
285         DT_VERS_1_7, &dt_idops_func,
286         "void(@, int32_t, int32_t, int32_t, int32_t, ...)" },
287 { "lquantize", DT_IDENT_AGGFUNC, 0, DTRACEAGG_LQUANTIZE,
288         DT_ATTR_STABCMN, DT_VERS_1_0,
289         &dt_idops_func, "void(@, int32_t, int32_t, ...)" },
290 { "max", DT_IDENT_AGGFUNC, 0, DTRACEAGG_MAX, DT_ATTR_STABCMN, DT_VERS_1_0,
291         &dt_idops_func, "void(@)" },
292 { "min", DT_IDENT_AGGFUNC, 0, DTRACEAGG_MIN, DT_ATTR_STABCMN, DT_VERS_1_0,
293         &dt_idops_func, "void(@)" },
294 { "mod", DT_IDENT_ACTFUNC, 0, DT_ACT_MOD, DT_ATTR_STABCMN,
295         DT_VERS_1_2, &dt_idops_func, "_symaddr(uintptr_t)" },
296 { "msgdsize", DT_IDENT_FUNC, 0, DIF_SUBR_MSGDSIZE,
297         DT_ATTR_STABCMN, DT_VERS_1_0,
298         &dt_idops_func, "size_t(mblk_t *)" },
299 { "msgsize", DT_IDENT_FUNC, 0, DIF_SUBR_MSGSIZE,
300         DT_ATTR_STABCMN, DT_VERS_1_0,
301         &dt_idops_func, "size_t(mblk_t *)" },
302 { "mutex_owned", DT_IDENT_FUNC, 0, DIF_SUBR_MUTEX_OWNED,
303         DT_ATTR_EVOLCMN, DT_VERS_1_0,
304         &dt_idops_func, "int(genunix`kmutex_t *)" },
305 { "mutex_owner", DT_IDENT_FUNC, 0, DIF_SUBR_MUTEX_OWNER,
306         DT_ATTR_EVOLCMN, DT_VERS_1_0,
307         &dt_idops_func, "genunix`kthread_t *(genunix`kmutex_t *)" },
308 { "mutex_type_adaptive", DT_IDENT_FUNC, 0, DIF_SUBR_MUTEX_TYPE_ADAPTIVE,
309         DT_ATTR_EVOLCMN, DT_VERS_1_0,
310         &dt_idops_func, "int(genunix`kmutex_t *)" },
311 { "mutex_type_spin", DT_IDENT_FUNC, 0, DIF_SUBR_MUTEX_TYPE_SPIN,
312         DT_ATTR_EVOLCMN, DT_VERS_1_0,
313         &dt_idops_func, "int(genunix`kmutex_t *)" },
314 { "ntohl", DT_IDENT_FUNC, 0, DIF_SUBR_NTOHL, DT_ATTR_EVOLCMN, DT_VERS_1_3,
315         &dt_idops_func, "uint32_t(uint32_t)" },
316 { "ntohll", DT_IDENT_FUNC, 0, DIF_SUBR_NTOHLL, DT_ATTR_EVOLCMN, DT_VERS_1_3,
317         &dt_idops_func, "uint64_t(uint64_t)" },
318 { "ntohs", DT_IDENT_FUNC, 0, DIF_SUBR_NTOHS, DT_ATTR_EVOLCMN, DT_VERS_1_3,
319         &dt_idops_func, "uint16_t(uint16_t)" },
320 { "normalize", DT_IDENT_ACTFUNC, 0, DT_ACT_NORMALIZE, DT_ATTR_STABCMN,
321         DT_VERS_1_0, &dt_idops_func, "void(...)" },
```



```
322 { "panic", DT_IDENT_ACTFUNC, 0, DT_ACT_PANIC, DT_ATTR_STABCMN, DT_VERS_1_0,
323         &dt_idops_func, "void()" },
324 { "pid", DT_IDENT_SCALAR, 0, DIF_VAR_PID, DT_ATTR_STABCMN, DT_VERS_1_0,
325         &dt_idops_type, "pid_t" },
326 { "ppid", DT_IDENT_SCALAR, 0, DIF_VAR_PPID, DT_ATTR_STABCMN, DT_VERS_1_0,
327         &dt_idops_type, "pid_t" },
328 { "print", DT_IDENT_ACTFUNC, 0, DT_ACT_PRINT, DT_ATTR_STABCMN, DT_VERS_1_9,
329         &dt_idops_func, "void(@)" },
330 { "printa", DT_IDENT_ACTFUNC, 0, DT_ACT_PRINTA, DT_ATTR_STABCMN, DT_VERS_1_0,
331         &dt_idops_func, "void(@, ...)" },
332 { "printf", DT_IDENT_ACTFUNC, 0, DT_ACT_PRINTF, DT_ATTR_STABCMN, DT_VERS_1_0,
333         &dt_idops_func, "void(@, ...)" },
334 { "probefunc", DT_IDENT_SCALAR, 0, DIF_VAR_PROBEFUNC,
335         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
336 { "probemod", DT_IDENT_SCALAR, 0, DIF_VAR_PROBEMOD,
337         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
338 { "probename", DT_IDENT_SCALAR, 0, DIF_VAR_PROBENAME,
339         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
340 { "probeprov", DT_IDENT_SCALAR, 0, DIF_VAR_PROBEPROV,
341         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
342 { "progenyof", DT_IDENT_FUNC, 0, DIF_SUBR_PROGENYOF,
343         DT_ATTR_STABCMN, DT_VERS_1_0,
344         &dt_idops_func, "int(pid_t)" },
345 { "quantize", DT_IDENT_AGGFUNC, 0, DTRACEAGG_QUANTIZE,
346         DT_ATTR_STABCMN, DT_VERS_1_0,
347         &dt_idops_func, "void(@, ...)" },
348 { "raise", DT_IDENT_ACTFUNC, 0, DT_ACT_RAISE, DT_ATTR_STABCMN, DT_VERS_1_0,
349         &dt_idops_func, "void(int)" },
350 { "rand", DT_IDENT_FUNC, 0, DIF_SUBR_RAND, DT_ATTR_STABCMN, DT_VERS_1_0,
351         &dt_idops_func, "int()" },
352 { "rindex", DT_IDENT_FUNC, 0, DIF_SUBR_RINDEX, DT_ATTR_STABCMN, DT_VERS_1_1,
353         &dt_idops_func, "int(const char *, const char *, [int])" },
354 { "rw_iswriter", DT_IDENT_FUNC, 0, DIF_SUBR_RW_ISWRITER,
355         DT_ATTR_EVOLCMN, DT_VERS_1_0,
356         &dt_idops_func, "int(genunix`krwlock_t *)" },
357 { "rw_read_held", DT_IDENT_FUNC, 0, DIF_SUBR_RW_READ_HELD,
358         DT_ATTR_EVOLCMN, DT_VERS_1_0,
359         &dt_idops_func, "int(genunix`krwlock_t *)" },
360 { "rw_write_held", DT_IDENT_FUNC, 0, DIF_SUBR_RW_WRITE_HELD,
361         DT_ATTR_EVOLCMN, DT_VERS_1_0,
362         &dt_idops_func, "int(genunix`krwlock_t *)" },
363 { "self", DT_IDENT_PTR, 0, 0, DT_ATTR_STABCMN, DT_VERS_1_0,
364         &dt_idops_type, "void" },
365 { "setopt", DT_IDENT_ACTFUNC, 0, DT_ACT_SETOPT, DT_ATTR_STABCMN,
366         DT_VERS_1_2, &dt_idops_func, "void(const char *, [const char *])" },
367 { "speculate", DT_IDENT_ACTFUNC, 0, DT_ACT_SPECULATE,
368         DT_ATTR_STABCMN, DT_VERS_1_0,
369         &dt_idops_func, "void(int)" },
370 { "speculation", DT_IDENT_FUNC, 0, DIF_SUBR_SPECULATION,
371         DT_ATTR_STABCMN, DT_VERS_1_0,
372         &dt_idops_func, "int()" },
373 { "stack", DT_IDENT_ACTFUNC, 0, DT_ACT_STACK, DT_ATTR_STABCMN, DT_VERS_1_0,
374         &dt_idops_func, "stack(...)" },
375 { "stackdepth", DT_IDENT_SCALAR, 0, DIF_VAR_STACKDEPTH,
376         DT_ATTR_STABCMN, DT_VERS_1_0,
377         &dt_idops_type, "uint32_t" },
378 { "stddev", DT_IDENT_AGGFUNC, 0, DTRACEAGG_STDDEV, DT_ATTR_STABCMN,
379         DT_VERS_1_6, &dt_idops_func, "void(@)" },
380 { "stop", DT_IDENT_ACTFUNC, 0, DT_ACT_STOP, DT_ATTR_STABCMN, DT_VERS_1_0,
381         &dt_idops_func, "void()" },
382 { "strchr", DT_IDENT_FUNC, 0, DIF_SUBR_STRCHR, DT_ATTR_STABCMN, DT_VERS_1_1,
383         &dt_idops_func, "string(const char *, char)" },
384 { "strlen", DT_IDENT_FUNC, 0, DIF_SUBR_STRLEN, DT_ATTR_STABCMN, DT_VERS_1_0,
385         &dt_idops_func, "size_t(const char *)" },
386 { "strjoin", DT_IDENT_FUNC, 0, DIF_SUBR_STRJOIN, DT_ATTR_STABCMN, DT_VERS_1_0,
387         &dt_idops_func, "string(const char *, const char *)" },
```

```
388 { "strrchr", DT_IDENT_FUNC, 0, DIF_SUBR_STRRCHR, DT_ATTR_STABCMN, DT_VERS_1_1,
389         &dt_idops_func, "string(const char *, char)" },
390 { "strstr", DT_IDENT_FUNC, 0, DIF_SUBR_STRSTR, DT_ATTR_STABCMN, DT_VERS_1_1,
391         &dt_idops_func, "string(const char *, const char *)" },
392 { "strtok", DT_IDENT_FUNC, 0, DIF_SUBR_STRTOK, DT_ATTR_STABCMN, DT_VERS_1_1,
393         &dt_idops_func, "string(const char *, const char *)" },
394 { "strtoll", DT_IDENT_FUNC, 0, DIF_SUBR_STRTOLL, DT_ATTR_STABCMN, DT_VERS_1_11,
395         &dt_idops_func, "int64_t(const char *, [int])" },
396 { "substr", DT_IDENT_FUNC, 0, DIF_SUBR_SUBSTR, DT_ATTR_STABCMN, DT_VERS_1_1,
397         &dt_idops_func, "string(const char *, int, [int])" },
398 { "sum", DT_IDENT_AGGFUNC, 0, DTRACEAGG_SUM, DT_ATTR_STABCMN, DT_VERS_1_0,
399         &dt_idops_func, "void(@)" },
400 { "sym", DT_IDENT_ACTFUNC, 0, DT_ACT_SYM, DT_ATTR_STABCMN,
401         DT_VERS_1_2, &dt_idops_func, "_symaddr(uintptr_t)" },
402 { "system", DT_IDENT_ACTFUNC, 0, DT_ACT_SYSTEM, DT_ATTR_STABCMN, DT_VERS_1_0,
403         &dt_idops_func, "void(@, ...)" },
404 { "this", DT_IDENT_PTR, 0, 0, DT_ATTR_STABCMN, DT_VERS_1_0,
405         &dt_idops_type, "void" },
406 { "threadname", DT_IDENT_SCALAR, 0, DIF_VAR_THREADNAME,
407         DT_ATTR_STABCMN, DT_VERS_1_14, &dt_idops_type, "string" },
408 { "tid", DT_IDENT_SCALAR, 0, DIF_VAR_TID, DT_ATTR_STABCMN, DT_VERS_1_0,
409         &dt_idops_type, "id_t" },
410 { "timestamp", DT_IDENT_SCALAR, 0, DIF_VAR_TIMESTAMP,
411         DT_ATTR_STABCMN, DT_VERS_1_0,
412         &dt_idops_type, "uint64_t" },
413 { "tolower", DT_IDENT_FUNC, 0, DIF_SUBR_TOLOWER, DT_ATTR_STABCMN, DT_VERS_1_8,
414         &dt_idops_func, "string(const char *)" },
415 { "toupper", DT_IDENT_FUNC, 0, DIF_SUBR_TOUPPER, DT_ATTR_STABCMN, DT_VERS_1_8,
416         &dt_idops_func, "string(const char *)" },
417 { "trace", DT_IDENT_ACTFUNC, 0, DT_ACT_TRACE, DT_ATTR_STABCMN, DT_VERS_1_0,
418         &dt_idops_func, "void(@)" },
419 { "tracemem", DT_IDENT_ACTFUNC, 0, DT_ACT_TRACEMEM,
420         DT_ATTR_STABCMN, DT_VERS_1_0,
421         &dt_idops_func, "void(@, size_t, ...)" },
422 { "trunc", DT_IDENT_ACTFUNC, 0, DT_ACT_TRUNC, DT_ATTR_STABCMN,
423         DT_VERS_1_0, &dt_idops_func, "void(...)" },
424 { "uaddr", DT_IDENT_ACTFUNC, 0, DT_ACT_UADDR, DT_ATTR_STABCMN,
425         DT_VERS_1_2, &dt_idops_func, "_usymaddr(uintptr_t)" },
426 { "ucaller", DT_IDENT_SCALAR, 0, DIF_VAR_UCALLER, DT_ATTR_STABCMN,
427         DT_VERS_1_2, &dt_idops_type, "uint64_t" },
428 { "ufunc", DT_IDENT_ACTFUNC, 0, DT_ACT_USYM, DT_ATTR_STABCMN,
429         DT_VERS_1_2, &dt_idops_func, "_usymaddr(uintptr_t)" },
430 { "uid", DT_IDENT_SCALAR, 0, DIF_VAR_UID, DT_ATTR_STABCMN, DT_VERS_1_0,
431         &dt_idops_type, "uid_t" },
432 { "umod", DT_IDENT_ACTFUNC, 0, DT_ACT_UMOD, DT_ATTR_STABCMN,
433         DT_VERS_1_2, &dt_idops_func, "_usymaddr(uintptr_t)" },
434 { "uregs", DT_IDENT_ARRAY, DT_IDFLG_WRITE, DIF_VAR_UREGS, DT_ATTR_STABCMN,
435         DT_VERS_1_0, &dt_idops_regs, NULL },
436 { "ustack", DT_IDENT_ACTFUNC, 0, DT_ACT_USTACK, DT_ATTR_STABCMN, DT_VERS_1_0,
437         &dt_idops_func, "stack(...)" },
438 { "ustackdepth", DT_IDENT_SCALAR, 0, DIF_VAR_USTACKDEPTH,
439         DT_ATTR_STABCMN, DT_VERS_1_2,
440         &dt_idops_type, "uint32_t" },
441 { "usym", DT_IDENT_ACTFUNC, 0, DT_ACT_USYM, DT_ATTR_STABCMN,
442         DT_VERS_1_2, &dt_idops_func, "_usymaddr(uintptr_t)" },
443 { "vmregs", DT_IDENT_ARRAY, 0, DIF_VAR_VMREGS, DT_ATTR_STABCMN, DT_VERS_1_7,
444         &dt_idops_regs, NULL },
445 { "vtimestamp", DT_IDENT_SCALAR, 0, DIF_VAR_VTIMESTAMP,
446         DT_ATTR_STABCMN, DT_VERS_1_0,
447         &dt_idops_type, "uint64_t" },
448 { "walltimestamp", DT_IDENT_SCALAR, 0, DIF_VAR_WALLTIMESTAMP,
449         DT_ATTR_STABCMN, DT_VERS_1_0,
450         &dt_idops_type, "int64_t" },
451 { "zonename", DT_IDENT_SCALAR, 0, DIF_VAR_ZONENAME,
452         DT_ATTR_STABCMN, DT_VERS_1_0, &dt_idops_type, "string" },
453 { NULL, 0, 0, 0, { 0, 0, 0 }, 0, NULL, NULL }
```



```
454 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3112 Mon Oct 15 13:25:15 2018
new/usr/src/lib/libfakekernel/common/sys/thread.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _SYS_THREAD_H
#define _SYS_THREAD_H

#include <sys/types.h>
#include <sys/t_lock.h>
#include <sys/klwp.h>
#include <sys/signal.h>  /* expected by including code */

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * The thread object, its states, and the methods by which it
 * is accessed.
 */

/*
 * Values that t_state may assume. Note that t_state cannot have more
 * than one of these flags set at a time.
 */
#define TS_FREE         0x00    /* Thread at loose ends */
#define TS_SLEEP        0x01    /* Awaiting an event */
#define TS_RUN          0x02    /* Runnable, but not yet on a processor */
#define TS_ONPROC       0x04    /* Thread is being run on a processor */
#define TS_ZOMB         0x08    /* Thread has died but hasn't been reaped */
#define TS_STOPPED      0x10    /* Stopped, initial state */
#define TS_WAIT         0x20    /* Waiting to become runnable */

/* ctxop_t */
```



```
/* afd_t needed by sys/file.h via sys/t_lock.h */
typedef struct _afd_not_used afd_t;

struct turnstile;
struct panic_trap_info;
struct upimutex;
struct kproject;
struct on_trap_data;
struct waitq;
struct _kcpc_ctx;
struct _kcpc_set;

/* Definition for kernel thread identifier type */
typedef uint64_t kt_did_t;

struct _kthread;
typedef struct _kthread *kthread_id_t;

typedef struct _kthread kthread_t;

extern  kthread_t       *_curthread(void);      /* returns thread pointer */
#define curthread       (_curthread())          /* current thread pointer */

#define _KTHREAD_INVALID        ((void *)(uintptr_t)-1)

#define THREAD_NAME_MAX (32)

struct proc;
extern struct proc      *_curproc(void);
#define curproc         (_curproc())            /* current proc pointer */

struct zone;
extern struct zone      *_curzone(void);
#define curzone         (_curzone())            /* current zone pointer */

extern  kthread_t       *thread_create(
        caddr_t         stk,
        size_t          stksize,
        void            (*proc)(),
        void            *arg,
        size_t          len,
        struct proc     *pp,
        int             state,
        pri_t           pri);
extern  void    thread_exit(void) __NORETURN;
extern  void    thread_join(kt_did_t);

extern kthread_t *zthread_create(caddr_t, size_t, void (*)(), void *, size_t,
    pri_t);
extern void zthread_exit(void) __NORETURN;

#ifdef  __cplusplus
}
```

_____*unchanged_portion_omitted_*

```
**********************************************************
   12910 Mon Oct 15 13:25:15 2018
new/usr/src/lib/libproc/common/Pcontrol.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright (c) 2014, Joyent, Inc. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _PCONTROL_H
#define _PCONTROL_H

/*
 * Implemention-specific include file for libproc process management.
 * This is not to be seen by the clients of libproc.
 */

#include <stdio.h>
#include <gelf.h>
#include <synch.h>
#include <procfs.h>
#include <rtld_db.h>
#include <libproc.h>
#include <libctf.h>
#include <limits.h>
#include <libproc.h>
#include <thread.h>
#include <sys/secflags.h>

#ifdef  __cplusplus
extern "C" {
#endif

#include "Putil.h"

/*
 * Definitions of the process control structures, internal to libproc.
 * These may change without affecting clients of libproc.
```



```
 */

/*
 * sym_tbl_t contains a primary and an (optional) auxiliary symbol table, which
 * we wish to treat as a single logical symbol table. In this logical table,
 * the data from the auxiliary table preceeds that from the primary. Symbol
 * indices start at [0], which is the first item in the auxiliary table
 * if there is one. The sole purpose for this is so that we can treat the
 * combination of .SUNW_ldynsym and .dynsym sections as a logically single
 * entity without having to violate the public interface to libelf.
 *
 * Both tables must share the same string table section.
 *
 * The symtab_getsym() function serves as a gelf_getsym() replacement
 * that is aware of the two tables and makes them look like a single table
 * to the caller.
 *
 */
typedef struct sym_tbl {        /* symbol table */
        Elf_Data *sym_data_pri; /* primary table */
        Elf_Data *sym_data_aux; /* auxiliary table */
        size_t  sym_symn_aux;   /* number of entries in auxiliary table */
        size_t  sym_symn;       /* total number of entries in both tables */
        char    *sym_strs;      /* ptr to strings */
        size_t  sym_strsz;      /* size of string table */
        GElf_Shdr sym_hdr_pri;  /* primary symbol table section header */
        GElf_Shdr sym_hdr_aux;  /* auxiliary symbol table section header */
        GElf_Shdr sym_strhdr;   /* string table section header */
        Elf     *sym_elf;       /* faked-up ELF handle from core file */
        void    *sym_elfmem;    /* data for faked-up ELF handle */
        uint_t  *sym_byname;    /* symbols sorted by name */
        uint_t  *sym_byaddr;    /* symbols sorted by addr */
        size_t  sym_count;      /* number of symbols in each sorted list */
} sym_tbl_t;
_____unchanged_portion_omitted_

typedef struct lwp_info {       /* per-lwp information from core file */
        plist_t lwp_list;       /* linked list */
        lwpid_t lwp_id;         /* lwp identifier */
        lwpsinfo_t lwp_psinfo;  /* /proc/<pid>/lwp/<lwpid>/lwpsinfo data */
        lwpstatus_t lwp_status; /* /proc/<pid>/lwp/<lwpid>/lwpstatus data */
        char lwp_name[THREAD_NAME_MAX];
#if defined(sparc) || defined(__sparc)
        gwindows_t *lwp_gwins;  /* /proc/<pid>/lwp/<lwpid>/gwindows data */
        prxregset_t *lwp_xregs; /* /proc/<pid>/lwp/<lwpid>/xregs data */
        int64_t *lwp_asrs;      /* /proc/<pid>/lwp/<lwpid>/asrs data */
#endif
} lwp_info_t;
_____unchanged_portion_omitted_
```

```
**********************************************************
   75696 Mon Oct 15 13:25:15 2018
new/usr/src/lib/libproc/common/Pcore.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright (c) 2018, Joyent, Inc. All rights reserved.
 * Copyright (c) 2014, Joyent, Inc. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 * Copyright 2015 Gary Mills
 */

#include <sys/types.h>
#include <sys/utsname.h>
#include <sys/sysmacros.h>
#include <sys/proc.h>

#include <alloca.h>
#include <rtld_db.h>
#include <libgen.h>
#include <limits.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <gelf.h>
#include <stddef.h>
#include <signal.h>

#include "libproc.h"
#include "Pcontrol.h"
#include "P32ton.h"
#include "Putil.h"
#ifdef __x86
#include "Pcore_linux.h"
#endif

/*
 * Pcore.c - Code to initialize a ps_prochandle from a core dump.  We
 * allocate an additional structure to hold information from the core
```



```
 * file, and attach this to the standard ps_prochandle in place of the
 * ability to examine /proc/<pid>/ files.
 */

/*
 * Basic i/o function for reading and writing from the process address space
 * stored in the core file and associated shared libraries.  We compute the
 * appropriate fd and offsets, and let the provided prw function do the rest.
 */
static ssize_t
core_rw(struct ps_prochandle *P, void *buf, size_t n, uintptr_t addr,
    ssize_t (*prw)(int, void *, size_t, off64_t))
{
        ssize_t resid = n;

        while (resid != 0) {
                map_info_t *mp = Paddr2mptr(P, addr);

                uintptr_t mapoff;
                ssize_t len;
                off64_t off;
                int fd;

                if (mp == NULL)
                        break;  /* No mapping for this address */

                if (mp->map_pmap.pr_mflags & MA_RESERVED1) {
                        if (mp->map_file == NULL || mp->map_file->file_fd < 0)
                                break;  /* No file or file not open */

                        fd = mp->map_file->file_fd;
                } else
                        fd = P->asfd;

                mapoff = addr - mp->map_pmap.pr_vaddr;
                len = MIN(resid, mp->map_pmap.pr_size - mapoff);
                off = mp->map_offset + mapoff;

                if ((len = prw(fd, buf, len, off)) <= 0)
                        break;

                resid -= len;
                addr += len;
                buf = (char *)buf + len;
        }

        /*
         * Important: Be consistent with the behavior of i/o on the as file:
         * writing to an invalid address yields EIO; reading from an invalid
         * address falls through to returning success and zero bytes.
         */
        if (resid == n && n != 0 && prw != pread64) {
                errno = EIO;
                return (-1);
        }

        return (n - resid);
}
______unchanged_portion_omitted_

static int
note_lwpname(struct ps_prochandle *P, size_t nbytes)
{
        prlwpname_t name;
        lwp_info_t *lwp;
```

```
 734         if (nbytes != sizeof (name) ||
 735             read(P->asfd, &name, sizeof (name)) != sizeof (name))
 736                 goto err;

 738         if ((lwp = lwpid2info(P, name.pr_lwpid)) == NULL)
 739                 goto err;

 741         if (strlcpy(lwp->lwp_name, name.pr_lwpname,
 742             sizeof (lwp->lwp_name)) >= sizeof (lwp->lwp_name)) {
 743                 errno = ENAMETOOLONG;
 744                 goto err;
 745         }

 747         return (0);

 749 err:
 750         dprintf("Pgrab_core: failed to read NT_LWPNAME\n");
 751         return (-1);
 752 }

 754 static int
 755 note_fdinfo(struct ps_prochandle *P, size_t nbytes)
 756 {
 757         prfdinfo_t prfd;
 758         fd_info_t *fip;

 760         if ((nbytes < sizeof (prfd)) ||
 761             (read(P->asfd, &prfd, sizeof (prfd)) != sizeof (prfd))) {
 762                 dprintf("Pgrab_core: failed to read NT_FDINFO\n");
 763                 return (-1);
 764         }

 766         if ((fip = Pfd2info(P, prfd.pr_fd)) == NULL) {
 767                 dprintf("Pgrab_core: failed to add NT_FDINFO\n");
 768                 return (-1);
 769         }
 770         (void) memcpy(&fip->fd_info, &prfd, sizeof (prfd));
 771         return (0);
 772 }
```

*_____unchanged_portion_omitted_*

```
1205 /*
1206  * Populate a table of function pointers indexed by Note type with our
1207  * functions to process each type of core file note:
1208  */
1209 static int (*nhdlrs[])(struct ps_prochandle *, size_t) = {
1210         note_notsup,            /*  0   unassigned              */
1211 #ifdef __x86
1212         note_linux_prstatus,            /*  1   NT_PRSTATUS (old)       */
1213 #else
1214         note_notsup,            /*  1   NT_PRSTATUS (old)       */
1215 #endif
1216         note_notsup,            /*  2   NT_PRFPREG (old)        */
1217 #ifdef __x86
1218         note_linux_psinfo,              /*  3   NT_PRPSINFO (old)       */
1219 #else
1220         note_notsup,            /*  3   NT_PRPSINFO (old)       */
1221 #endif
1222 #ifdef __sparc
1223         note_xreg,              /*  4   NT_PRXREG               */
1224 #else
1225         note_notsup,            /*  4   NT_PRXREG               */
1226 #endif
1227         note_platform,          /*  5   NT_PLATFORM             */
1228         note_auxv,              /*  6   NT_AUXV                 */
1229 #ifdef __sparc
```



```
1230         note_gwindows,          /*  7   NT_GWINDOWS             */
1231 #ifdef __sparcv9
1232         note_asrs,              /*  8   NT_ASRS                 */
1233 #else
1234         note_notsup,            /*  8   NT_ASRS                 */
1235 #endif
1236 #else
1237         note_notsup,            /*  7   NT_GWINDOWS             */
1238         note_notsup,            /*  8   NT_ASRS                 */
1239 #endif
1240 #ifdef __x86
1241         note_ldt,               /*  9   NT_LDT                  */
1242 #else
1243         note_notsup,            /*  9   NT_LDT                  */
1244 #endif
1245         note_pstatus,           /* 10   NT_PSTATUS              */
1246         note_notsup,            /* 11   unassigned              */
1247         note_notsup,            /* 12   unassigned              */
1248         note_psinfo,            /* 13   NT_PSINFO               */
1249         note_cred,              /* 14   NT_PRCRED               */
1250         note_utsname,           /* 15   NT_UTSNAME              */
1251         note_lwpstatus,         /* 16   NT_LWPSTATUS            */
1252         note_lwpsinfo,          /* 17   NT_LWPSINFO             */
1253         note_priv,              /* 18   NT_PRPRIV               */
1254         note_priv_info,         /* 19   NT_PRPRIVINFO           */
1255         note_content,           /* 20   NT_CONTENT              */
1256         note_zonename,          /* 21   NT_ZONENAME             */
1257         note_fdinfo,            /* 22   NT_FDINFO               */
1258         note_spymaster,         /* 23   NT_SPYMASTER            */
1259         note_secflags,          /* 24   NT_SECFLAGS             */
1260         note_lwpname,           /* 25   NT_LWPNAME              */
1261 };
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   40654 Mon Oct 15 13:25:15 2018
new/usr/src/lib/libproc/common/Pgcore.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright 2015 Joyent, Inc.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 */

#define _STRUCTURED_PROC        1

#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <strings.h>
#include <errno.h>
#include <procfs.h>
#include <priv.h>
#include <sys/elf.h>
#include <sys/machelf.h>
#include <sys/sysmacros.h>
#include <sys/systeminfo.h>
#include <sys/proc.h>
#include <sys/utsname.h>

#include <sys/old_procfs.h>

#include "Pcontrol.h"
#include "P32ton.h"

typedef enum {
        STR_NONE,
        STR_CTF,
        STR_SYMTAB,
        STR_DYNSYM,
        STR_STRTAB,
        STR_DYNSTR,
```



```
        STR_SHSTRTAB,
        STR_NUM
} shstrtype_t;
```

*unchanged_portion_omitted_*

```
static int
new_per_lwp(void *data, const lwpstatus_t *lsp, const lwpsinfo_t *lip)
{
        pgcore_t *pgc = data;
        struct ps_prochandle *P = pgc->P;
        prlwpname_t name = { 0, "" };
        psinfo_t ps;

        /*
         * If lsp is NULL this indicates that this is a zombie LWP in
         * which case we dump only the lwpsinfo_t structure and none of
         * the other ancillary LWP state data.
         */
        if (P->status.pr_dmodel == PR_MODEL_NATIVE) {
                if (write_note(pgc->pgc_fd, NT_LWPSINFO, lip,
                    sizeof (lwpsinfo_t), pgc->pgc_doff) != 0)
                        return (1);
                if (lsp == NULL)
                        return (0);
                if (write_note(pgc->pgc_fd, NT_LWPSTATUS, lsp,
                    sizeof (lwpstatus_t), pgc->pgc_doff) != 0)
                        return (1);
#ifdef _LP64
        } else {
                lwpsinfo32_t li32;
                lwpstatus32_t ls32;
                lwpsinfo_n_to_32(lip, &li32);
                if (write_note(pgc->pgc_fd, NT_LWPSINFO, &li32,
                    sizeof (lwpsinfo32_t), pgc->pgc_doff) != 0)
                        return (1);
                if (lsp == NULL)
                        return (0);
                lwpstatus_n_to_32(lsp, &ls32);
                if (write_note(pgc->pgc_fd, NT_LWPSTATUS, &ls32,
                    sizeof (lwpstatus32_t), pgc->pgc_doff) != 0)
                        return (1);
#endif  /* _LP64 */
        }

#ifdef sparc
        {
                prxregset_t xregs;
                gwindows_t gwins;
                size_t size;

                if (Plwp_getxregs(P, lsp->pr_lwpid, &xregs) == 0) {
                        if (write_note(pgc->pgc_fd, NT_PRXREG, &xregs,
                            sizeof (prxregset_t), pgc->pgc_doff) != 0)
                                return (1);
                }

                if (Plwp_getgwindows(P, lsp->pr_lwpid, &gwins) == 0 &&
                    gwins.wbcnt > 0) {
                        size = sizeof (gwins) - sizeof (gwins.wbuf) +
                            gwins.wbcnt * sizeof (gwins.wbuf[0]);

                        if (write_note(pgc->pgc_fd, NT_GWINDOWS, &gwins, size,
                            pgc->pgc_doff) != 0)
                                return (1);
                }
```

```
568         }
569 #ifdef __sparcv9
570         if (P->status.pr_dmodel == PR_MODEL_LP64) {
571                 asrset_t asrs;
572                 if (Plwp_getasrs(P, lsp->pr_lwpid, asrs) == 0) {
573                         if (write_note(pgc->pgc_fd, NT_ASRS, &asrs,
574                             sizeof (asrset_t), pgc->pgc_doff) != 0)
575                                 return (1);
576                 }
577         }
578 #endif  /* __sparcv9 */
579 #endif  /* sparc */

581         if (Plwp_getname(P, lsp->pr_lwpid, name.pr_lwpname,
582             sizeof (name.pr_lwpname)) == 0) {
583                 name.pr_lwpid = lsp->pr_lwpid;
584                 if (write_note(pgc->pgc_fd, NT_LWPNAME, &name,
585                     sizeof (name), pgc->pgc_doff) != 0)
586                         return (1);
587         }

589         if (!(lsp->pr_flags & PR_AGENT))
590                 return (0);

592         if (Plwp_getspymaster(P, lsp->pr_lwpid, &ps) != 0)
593                 return (0);

595         if (P->status.pr_dmodel == PR_MODEL_NATIVE) {
596                 if (write_note(pgc->pgc_fd, NT_SPYMASTER, &ps,
597                     sizeof (psinfo_t), pgc->pgc_doff) != 0)
598                         return (1);
599 #ifdef _LP64
600         } else {
601                 psinfo32_t ps32;
602                 psinfo_n_to_32(&ps, &ps32);
603                 if (write_note(pgc->pgc_fd, NT_SPYMASTER, &ps32,
604                     sizeof (psinfo32_t), pgc->pgc_doff) != 0)
605                         return (1);
606 #endif  /* _LP64 */
607         }


610         return (0);
611 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   12164 Mon Oct 15 13:25:16 2018
new/usr/src/lib/libproc/common/Plwpregs.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2018 Joyent, Inc.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 */

#include <sys/types.h>
#include <sys/uio.h>
#include <string.h>
#include <errno.h>
#include <limits.h>

#include "Pcontrol.h"
#include "P32ton.h"

/*
 * This file implements the routines to read and write per-lwp register
 * information from either a live process or core file opened with libproc.
 * We build up a few common routines for reading and writing register
 * information, and then the public functions are all trivial calls to these.
 */

/*
 * Utility function to return a pointer to the structure of cached information
 * about an lwp in the core file, given its lwpid.
 */
static lwp_info_t *
getlwpcore(struct ps_prochandle *P, lwpid_t lwpid)
{
        core_info_t *core = P->data;
        lwp_info_t *lwp = list_next(&core->core_lwp_head);
        uint_t i;

        for (i = 0; i < core->core_nlwp; i++, lwp = list_next(lwp)) {
                if (lwp->lwp_id == lwpid)
```



```
                        return (lwp);
        }

        errno = EINVAL;
        return (NULL);
}
```

*unchanged_portion_omitted_*

```
int
Plwp_getname(struct ps_prochandle *P, lwpid_t lwpid,
    char *buf, size_t bufsize)
{
        char lwpname[THREAD_NAME_MAX];
        char *from = NULL;
        lwp_info_t *lwp;

        if (P->state == PS_IDLE) {
                errno = ENODATA;
                return (-1);
        }

        if (P->state != PS_DEAD) {
                if (getlwpfile(P, lwpid, "lwpname",
                    lwpname, sizeof (lwpname)) != 0)
                        return (-1);
                from = lwpname;
        } else {
                if ((lwp = getlwpcore(P, lwpid)) == NULL)
                        return (-1);
                from = lwp->lwp_name;
        }

        if (strlcpy(buf, from, bufsize) >= bufsize) {
                errno = ENAMETOOLONG;
                return (-1);
        }

        return (0);
}

int
Plwp_getspymaster(struct ps_prochandle *P, lwpid_t lwpid, psinfo_t *ps)
{
        lwpstatus_t lps;

        if (P->state == PS_IDLE) {
                errno = ENODATA;
                return (-1);
        }

        if (getlwpstatus(P, lwpid, &lps) != 0)
                return (-1);

        if (!(lps.pr_flags & PR_AGENT)) {
                errno = EINVAL;
                return (-1);
        }

        if (P->state != PS_DEAD) {
                return (getlwpfile(P, lwpid, "spymaster",
                    ps, sizeof (psinfo_t)));
        }

        if (P->spymaster.pr_nlwp != 0) {
                (void) memcpy(ps, &P->spymaster, sizeof (psinfo_t));
                return (0);
```

```
 413         }

 415         errno = ENODATA;

 417         return (-1);
 418 }
```

_____unchanged_portion_omitted_

```
**********************************************************
   31352 Mon Oct 15 13:25:16 2018
new/usr/src/lib/libproc/common/libproc.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Portions Copyright 2007 Chad Mynhier
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018, Joyent, Inc.
 * Copyright 2015, Joyent, Inc.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 */

/*
 * Interfaces available from the process control library, libproc.
 */

#ifndef _LIBPROC_H
#define _LIBPROC_H

#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <nlist.h>
#include <door.h>
#include <gelf.h>
#include <proc_service.h>
#include <rtld_db.h>
#include <procfs.h>
#include <ucred.h>
#include <rctl.h>
#include <libctf.h>
#include <sys/stat.h>
#include <sys/statvfs.h>
#include <sys/auxv.h>
#include <sys/resource.h>
#include <sys/socket.h>
#include <sys/utsname.h>
#include <sys/corectl.h>
#include <sys/secflags.h>
#if defined(__i386) || defined(__amd64)
```



```
#include <sys/sysi86.h>
#endif

#ifdef __cplusplus
extern "C" {
#endif

/*
 * Opaque structure tag reference to a process control structure.
 * Clients of libproc cannot look inside the process control structure.
 * The implementation of struct ps_prochandle can change w/o affecting clients.
 */
struct ps_prochandle;

/*
 * Opaque structure tag reference to an lwp control structure.
 */
struct ps_lwphandle;

extern  int     _libproc_debug; /* set non-zero to enable debugging fprintfs */
extern  int     _libproc_no_qsort;      /* set non-zero to inhibit sorting */
                                        /* of symbol tables */
extern  int     _libproc_incore_elf;    /* only use in-core elf data */

#if defined(__sparc)
#define R_RVAL1 R_O0            /* register holding a function return value */
#define R_RVAL2 R_O1            /* 32 more bits for a 64-bit return value */
#endif  /* __sparc */

#if defined(__amd64)
#define R_PC    REG_RIP
#define R_SP    REG_RSP
#define R_RVAL1 REG_RAX         /* register holding a function return value */
#define R_RVAL2 REG_RDX         /* 32 more bits for a 64-bit return value */
#elif defined(__i386)
#define R_PC    EIP
#define R_SP    UESP
#define R_RVAL1 EAX             /* register holding a function return value */
#define R_RVAL2 EDX             /* 32 more bits for a 64-bit return value */
#endif  /* __amd64 || __i386 */

#define R_RVAL  R_RVAL1         /* simple function return value register */

/* maximum sizes of things */
#define PRMAXSIG        (32 * sizeof (sigset_t) / sizeof (uint32_t))
#define PRMAXFAULT      (32 * sizeof (fltset_t) / sizeof (uint32_t))
#define PRMAXSYS        (32 * sizeof (sysset_t) / sizeof (uint32_t))

/* State values returned by Pstate() */
#define PS_RUN          1       /* process is running */
#define PS_STOP         2       /* process is stopped */
#define PS_LOST         3       /* process is lost to control (EAGAIN) */
#define PS_UNDEAD       4       /* process is terminated (zombie) */
#define PS_DEAD         5       /* process is terminated (core file) */
#define PS_IDLE         6       /* process has not been run */

/* Flags accepted by Pgrab() */
#define PGRAB_RETAIN    0x01    /* Retain tracing flags, else clear flags */
#define PGRAB_FORCE     0x02    /* Open the process w/o O_EXCL */
#define PGRAB_RDONLY    0x04    /* Open the process or core w/ O_RDONLY */
#define PGRAB_NOSTOP    0x08    /* Open the process but do not stop it */
#define PGRAB_INCORE    0x10    /* Use in-core data to build symbol tables */

/* Error codes from Pcreate() */
#define C_STRANGE       -1      /* Unanticipated error, errno is meaningful */
#define C_FORK          1       /* Unable to fork */
```

```
126 #define C_PERM          2       /* No permission (file set-id or unreadable) */
127 #define C_NOEXEC        3       /* Cannot execute file */
128 #define C_INTR          4       /* Interrupt received while creating */
129 #define C_LP64          5       /* Program is _LP64, self is _ILP32 */
130 #define C_NOENT         6       /* Cannot find executable file */

132 /* Error codes from Pgrab(), Pfgrab_core(), and Pgrab_core() */
133 #define G_STRANGE       -1      /* Unanticipated error, errno is meaningful */
134 #define G_NOPROC        1       /* No such process */
135 #define G_NOCORE        2       /* No such core file */
136 #define G_NOPROCORCORE  3       /* No such proc or core (for proc_arg_grab) */
137 #define G_NOEXEC        4       /* Cannot locate executable file */
138 #define G_ZOMB          5       /* Zombie process */
139 #define G_PERM          6       /* No permission */
140 #define G_BUSY          7       /* Another process has control */
141 #define G_SYS           8       /* System process */
142 #define G_SELF          9       /* Process is self */
143 #define G_INTR          10      /* Interrupt received while grabbing */
144 #define G_LP64          11      /* Process is _LP64, self is ILP32 */
145 #define G_FORMAT        12      /* File is not an ELF format core file */
146 #define G_ELF           13      /* Libelf error, elf_errno() is meaningful */
147 #define G_NOTE          14      /* Required PT_NOTE Phdr not present in core */
148 #define G_ISAINVAL      15      /* Wrong ELF machine type */
149 #define G_BADLWPS       16      /* Bad '/lwps' specification */
150 #define G_NOFD          17      /* No more file descriptors */


153 /* Flags accepted by Prelease */
154 #define PRELEASE_CLEAR  0x10    /* Clear all tracing flags */
155 #define PRELEASE_RETAIN 0x20    /* Retain final tracing flags */
156 #define PRELEASE_HANG   0x40    /* Leave the process stopped */
157 #define PRELEASE_KILL   0x80    /* Terminate the process */

159 typedef struct {        /* argument descriptor for system call (Psyscall) */
160         long    arg_value;      /* value of argument given to system call */
161         void    *arg_object;    /* pointer to object in controlling process */
162         char    arg_type;       /* AT_BYVAL, AT_BYREF */
163         char    arg_inout;      /* AI_INPUT, AI_OUTPUT, AI_INOUT */
164         ushort_t arg_size;      /* if AT_BYREF, size of object in bytes */
165 } argdes_t;
______unchanged_portion_omitted_

233 /*
234  * Function prototypes for routines in the process control package.
235  */
236 extern struct ps_prochandle *Pcreate(const char *, char *const *,
237     int *, char *, size_t);
238 extern struct ps_prochandle *Pxcreate(const char *, char *const *,
239     char *const *, int *, char *, size_t);

241 extern const char *Pcreate_error(int);

243 extern struct ps_prochandle *Pgrab(pid_t, int, int *);
244 extern struct ps_prochandle *Pgrab_core(const char *, const char *, int, int *);
245 extern struct ps_prochandle *Pfgrab_core(int, const char *, int *);
246 extern struct ps_prochandle *Pgrab_file(const char *, int *);
247 extern struct ps_prochandle *Pgrab_ops(pid_t, void *, const ps_ops_t *, int);
248 extern const char *Pgrab_error(int);

250 extern  int     Preopen(struct ps_prochandle *);
251 extern  void    Prelease(struct ps_prochandle *, int);
252 extern  void    Pfree(struct ps_prochandle *);

254 extern  int     Pasfd(struct ps_prochandle *);
255 extern  char   *Pbrandname(struct ps_prochandle *, char *, size_t);
256 extern  int     Pctlfd(struct ps_prochandle *);
```



```
257 extern  int     Pcreate_agent(struct ps_prochandle *);
258 extern  void    Pdestroy_agent(struct ps_prochandle *);
259 extern  int     Pstopstatus(struct ps_prochandle *, long, uint_t);
260 extern  int     Pwait(struct ps_prochandle *, uint_t);
261 extern  int     Pstop(struct ps_prochandle *, uint_t);
262 extern  int     Pdstop(struct ps_prochandle *);
263 extern  int     Pstate(struct ps_prochandle *);
264 extern  const psinfo_t *Ppsinfo(struct ps_prochandle *);
265 extern  const pstatus_t *Pstatus(struct ps_prochandle *);
266 extern  int     Pcred(struct ps_prochandle *, prcred_t *, int);
267 extern  int     Psetcred(struct ps_prochandle *, const prcred_t *);
268 extern  int     Ppriv(struct ps_prochandle *, prpriv_t **);
269 extern  void    Ppriv_free(struct ps_prochandle *, prpriv_t *);
270 extern  int     Psetpriv(struct ps_prochandle *, prpriv_t *);
271 extern  void   *Pprivinfo(struct ps_prochandle *);
272 extern  int     Psetzoneid(struct ps_prochandle *, zoneid_t);
273 extern  int     Pgetareg(struct ps_prochandle *, int, prgreg_t *);
274 extern  int     Pputareg(struct ps_prochandle *, int, prgreg_t);
275 extern  int     Psetrun(struct ps_prochandle *, int, int);
276 extern  int     Psecflags(struct ps_prochandle *, prsecflags_t **);
277 extern  void    Psecflags_free(prsecflags_t *);
278 extern  ssize_t Pread(struct ps_prochandle *, void *, size_t, uintptr_t);
279 extern  ssize_t Pread_string(struct ps_prochandle *, char *, size_t, uintptr_t);
280 extern  ssize_t Pwrite(struct ps_prochandle *, const void *, size_t, uintptr_t);
281 extern  int     Pclearsig(struct ps_prochandle *);
282 extern  int     Pclearfault(struct ps_prochandle *);
283 extern  int     Psetbkpt(struct ps_prochandle *, uintptr_t, ulong_t *);
284 extern  int     Pdelbkpt(struct ps_prochandle *, uintptr_t, ulong_t);
285 extern  int     Pxecbkpt(struct ps_prochandle *, ulong_t);
286 extern  int     Psetwapt(struct ps_prochandle *, const prwatch_t *);
287 extern  int     Pdelwapt(struct ps_prochandle *, const prwatch_t *);
288 extern  int     Pxecwapt(struct ps_prochandle *, const prwatch_t *);
289 extern  int     Psetflags(struct ps_prochandle *, long);
290 extern  int     Punsetflags(struct ps_prochandle *, long);
291 extern  int     Psignal(struct ps_prochandle *, int, int);
292 extern  int     Pfault(struct ps_prochandle *, int, int);
293 extern  int     Psysentry(struct ps_prochandle *, int, int);
294 extern  int     Psysexit(struct ps_prochandle *, int, int);
295 extern  void    Psetsignal(struct ps_prochandle *, const sigset_t *);
296 extern  void    Psetfault(struct ps_prochandle *, const fltset_t *);
297 extern  void    Psetsysentry(struct ps_prochandle *, const sysset_t *);
298 extern  void    Psetsysexit(struct ps_prochandle *, const sysset_t *);

300 extern  void    Psync(struct ps_prochandle *);
301 extern  int     Psyscall(struct ps_prochandle *, sysret_t *,
302                         int, uint_t, argdes_t *);
303 extern  int     Pisprocdir(struct ps_prochandle *, const char *);

305 /*
306  * Function prototypes for lwp-specific operations.
307  */
308 extern  struct ps_lwphandle *Lgrab(struct ps_prochandle *, lwpid_t, int *);
309 extern  const char *Lgrab_error(int);

311 extern  struct ps_prochandle *Lprochandle(struct ps_lwphandle *);
312 extern  void    Lfree(struct ps_lwphandle *);

314 extern  int     Lctlfd(struct ps_lwphandle *);
315 extern  int     Lwait(struct ps_lwphandle *, uint_t);
316 extern  int     Lstop(struct ps_lwphandle *, uint_t);
317 extern  int     Ldstop(struct ps_lwphandle *);
318 extern  int     Lstate(struct ps_lwphandle *);
319 extern  const lwpsinfo_t *Lpsinfo(struct ps_lwphandle *);
320 extern  const lwpstatus_t *Lstatus(struct ps_lwphandle *);
321 extern  int     Lgetareg(struct ps_lwphandle *, int, prgreg_t *);
322 extern  int     Lputareg(struct ps_lwphandle *, int, prgreg_t);
```

```
323 extern  int     Lsetrun(struct ps_lwphandle *, int, int);
324 extern  int     Lclearsig(struct ps_lwphandle *);
325 extern  int     Lclearfault(struct ps_lwphandle *);
326 extern  int     Lxecbkpt(struct ps_lwphandle *, ulong_t);
327 extern  int     Lxecwapt(struct ps_lwphandle *, const prwatch_t *);
328 extern  void    Lsync(struct ps_lwphandle *);

330 extern  int     Lstack(struct ps_lwphandle *, stack_t *);
331 extern  int     Lmain_stack(struct ps_lwphandle *, stack_t *);
332 extern  int     Lalt_stack(struct ps_lwphandle *, stack_t *);

334 /*
335  * Function prototypes for system calls forced on the victim process.
336  */
337 extern  int     pr_open(struct ps_prochandle *, const char *, int, mode_t);
338 extern  int     pr_creat(struct ps_prochandle *, const char *, mode_t);
339 extern  int     pr_close(struct ps_prochandle *, int);
340 extern  int     pr_access(struct ps_prochandle *, const char *, int);
341 extern  int     pr_door_info(struct ps_prochandle *, int, struct door_info *);
342 extern  void    *pr_mmap(struct ps_prochandle *,
343                         void *, size_t, int, int, int, off_t);
344 extern  void    *pr_zmap(struct ps_prochandle *,
345                         void *, size_t, int, int);
346 extern  int     pr_munmap(struct ps_prochandle *, void *, size_t);
347 extern  int     pr_memcntl(struct ps_prochandle *,
348                         caddr_t, size_t, int, caddr_t, int, int);
349 extern  int     pr_meminfo(struct ps_prochandle *, const uint64_t *addrs,
350                         int addr_count, const uint_t *info, int info_count,
351                         uint64_t *outdata, uint_t *validity);
352 extern  int     pr_sigaction(struct ps_prochandle *,
353                         int, const struct sigaction *, struct sigaction *);
354 extern  int     pr_getitimer(struct ps_prochandle *,
355                         int, struct itimerval *);
356 extern  int     pr_setitimer(struct ps_prochandle *,
357                         int, const struct itimerval *, struct itimerval *);
358 extern  int     pr_ioctl(struct ps_prochandle *, int, int, void *, size_t);
359 extern  int     pr_fcntl(struct ps_prochandle *, int, int, void *);
360 extern  int     pr_stat(struct ps_prochandle *, const char *, struct stat *);
361 extern  int     pr_lstat(struct ps_prochandle *, const char *, struct stat *);
362 extern  int     pr_fstat(struct ps_prochandle *, int, struct stat *);
363 extern  int     pr_stat64(struct ps_prochandle *, const char *,
364                         struct stat64 *);
365 extern  int     pr_lstat64(struct ps_prochandle *, const char *,
366                         struct stat64 *);
367 extern  int     pr_fstat64(struct ps_prochandle *, int, struct stat64 *);
368 extern  int     pr_statvfs(struct ps_prochandle *, const char *, statvfs_t *);
369 extern  int     pr_fstatvfs(struct ps_prochandle *, int, statvfs_t *);
370 extern  projid_t pr_getprojid(struct ps_prochandle *Pr);
371 extern  taskid_t pr_gettaskid(struct ps_prochandle *Pr);
372 extern  taskid_t pr_settaskid(struct ps_prochandle *Pr, projid_t project,
373                         int flags);
374 extern  zoneid_t pr_getzoneid(struct ps_prochandle *Pr);
375 extern  int     pr_getrctl(struct ps_prochandle *,
376                         const char *, rctlblk_t *, rctlblk_t *, int);
377 extern  int     pr_setrctl(struct ps_prochandle *,
378                         const char *, rctlblk_t *, rctlblk_t *, int);
379 extern  int     pr_getrlimit(struct ps_prochandle *,
380                         int, struct rlimit *);
381 extern  int     pr_setrlimit(struct ps_prochandle *,
382                         int, const struct rlimit *);
383 extern  int     pr_setprojrctl(struct ps_prochandle *, const char *,
384                         rctlblk_t *, size_t, int);
385 #if defined(_LARGEFILE64_SOURCE)
386 extern  int     pr_getrlimit64(struct ps_prochandle *,
387                         int, struct rlimit64 *);
388 extern  int     pr_setrlimit64(struct ps_prochandle *,
```



```
389                         int, const struct rlimit64 *);
390 #endif  /* _LARGEFILE64_SOURCE */
391 extern  int     pr_lwp_exit(struct ps_prochandle *);
392 extern  int     pr_exit(struct ps_prochandle *, int);
393 extern  int     pr_waitid(struct ps_prochandle *,
394                         idtype_t, id_t, siginfo_t *, int);
395 extern  off_t   pr_lseek(struct ps_prochandle *, int, off_t, int);
396 extern  offset_t pr_llseek(struct ps_prochandle *, int, offset_t, int);
397 extern  int     pr_rename(struct ps_prochandle *, const char *, const char *);
398 extern  int     pr_link(struct ps_prochandle *, const char *, const char *);
399 extern  int     pr_unlink(struct ps_prochandle *, const char *);
400 extern  int     pr_getpeerucred(struct ps_prochandle *, int, ucred_t **);
401 extern  int     pr_getpeername(struct ps_prochandle *,
402                         int, struct sockaddr *, socklen_t *);
403 extern  int     pr_getsockname(struct ps_prochandle *,
404                         int, struct sockaddr *, socklen_t *);
405 extern  int     pr_getsockopt(struct ps_prochandle *,
406                         int, int, int, void *, int *);
407 extern  int     pr_processor_bind(struct ps_prochandle *,
408                         idtype_t, id_t, int, int *);

410 /*
411  * Function prototypes for accessing per-LWP register information.
412  */
413 extern int Plwp_getregs(struct ps_prochandle *, lwpid_t, prgregset_t);
414 extern int Plwp_setregs(struct ps_prochandle *, lwpid_t, const prgregset_t);

416 extern int Plwp_getfpregs(struct ps_prochandle *, lwpid_t, prfpregset_t *);
417 extern int Plwp_setfpregs(struct ps_prochandle *, lwpid_t,
418     const prfpregset_t *);

420 #if defined(__sparc)

422 extern int Plwp_getxregs(struct ps_prochandle *, lwpid_t, prxregset_t *);
423 extern int Plwp_setxregs(struct ps_prochandle *, lwpid_t, const prxregset_t *);

425 extern int Plwp_getgwindows(struct ps_prochandle *, lwpid_t, gwindows_t *);

427 #if defined(__sparcv9)
428 extern int Plwp_getasrs(struct ps_prochandle *, lwpid_t, asrset_t);
429 extern int Plwp_setasrs(struct ps_prochandle *, lwpid_t, const asrset_t);
430 #endif  /* __sparcv9 */

432 #endif  /* __sparc */

434 #if defined(__i386) || defined(__amd64)
435 extern  int     Pldt(struct ps_prochandle *, struct ssd *, int);
436 extern  int     proc_get_ldt(pid_t, struct ssd *, int);
437 #endif  /* __i386 || __amd64 */

439 extern int Plwp_getname(struct ps_prochandle *, lwpid_t, char *, size_t);
440 extern int Plwp_getpsinfo(struct ps_prochandle *, lwpid_t, lwpsinfo_t *);
441 extern int Plwp_getspymaster(struct ps_prochandle *, lwpid_t, psinfo_t *);

443 extern int Plwp_stack(struct ps_prochandle *, lwpid_t, stack_t *);
444 extern int Plwp_main_stack(struct ps_prochandle *, lwpid_t, stack_t *);
445 extern int Plwp_alt_stack(struct ps_prochandle *, lwpid_t, stack_t *);

447 /*
448  * LWP iteration interface; iterate over all active LWPs.
449  */
450 typedef int proc_lwp_f(void *, const lwpstatus_t *);
451 extern int Plwp_iter(struct ps_prochandle *, proc_lwp_f *, void *);

453 /*
454  * LWP iteration interface; iterate over all LWPs, active and zombie.
```

```
 455  */
 456 typedef int proc_lwp_all_f(void *, const lwpstatus_t *, const lwpsinfo_t *);
 457 extern int Plwp_iter_all(struct ps_prochandle *, proc_lwp_all_f *, void *);

 459 /*
 460  * Process iteration interface; iterate over all non-system processes.
 461  */
 462 typedef int proc_walk_f(psinfo_t *, lwpsinfo_t *, void *);
 463 extern int proc_walk(proc_walk_f *, void *, int);

 465 #define PR_WALK_PROC    0               /* walk processes only */
 466 #define PR_WALK_LWP     1               /* walk all lwps */

 468 /*
 469  * Determine if an lwp is in a set as returned from proc_arg_xgrab().
 470  */
 471 extern int proc_lwp_in_set(const char *, lwpid_t);
 472 extern int proc_lwp_range_valid(const char *);

 474 /*
 475  * Symbol table interfaces.
 476  */

 478 /*
 479  * Pseudo-names passed to Plookup_by_name() for well-known load objects.
 480  * NOTE: It is required that PR_OBJ_EXEC and PR_OBJ_LDSO exactly match
 481  * the definitions of PS_OBJ_EXEC and PS_OBJ_LDSO from <proc_service.h>.
 482  */
 483 #define PR_OBJ_EXEC     ((const char *)0)       /* search the executable file */
 484 #define PR_OBJ_LDSO     ((const char *)1)       /* search ld.so.1 */
 485 #define PR_OBJ_EVERY    ((const char *)-1)      /* search every load object */

 487 /*
 488  * Special Lmid_t passed to Plookup_by_lmid() to search all link maps.  The
 489  * special values LM_ID_BASE and LM_ID_LDSO from <link.h> may also be used.
 490  * If PR_OBJ_EXEC is used as the object name, the lmid must be PR_LMID_EVERY
 491  * or LM_ID_BASE in order to return a match.  If PR_OBJ_LDSO is used as the
 492  * object name, the lmid must be PR_LMID_EVERY or LM_ID_LDSO to return a match.
 493  */
 494 #define PR_LMID_EVERY   ((Lmid_t)-1UL)          /* search every link map */

 496 /*
 497  * 'object_name' is the name of a load object obtained from an
 498  * iteration over the process's address space mappings (Pmapping_iter),
 499  * or an iteration over the process's mapped objects (Pobject_iter),
 500  * or else it is one of the special PR_OBJ_* values above.
 501  */
 502 extern int Plookup_by_name(struct ps_prochandle *,
 503     const char *, const char *, GElf_Sym *);

 505 extern int Plookup_by_addr(struct ps_prochandle *,
 506     uintptr_t, char *, size_t, GElf_Sym *);

 508 typedef struct prsyminfo {
 509         const char      *prs_object;            /* object name */
 510         const char      *prs_name;              /* symbol name */
 511         Lmid_t          prs_lmid;               /* link map id */
 512         uint_t          prs_id;                 /* symbol id */
 513         uint_t          prs_table;              /* symbol table id */
 514 } prsyminfo_t;
```

*unchanged_portion_omitted_*

```
**********************************************************
    5698 Mon Oct 15 13:25:17 2018
new/usr/src/lib/libproc/common/mapfile-vers
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
# Copyright 2018 Joyent, Inc.
# Copyright (c) 2013, Joyent, Inc. All rights reserved.
# Copyright (c) 2013 by Delphix. All rights reserved.
#

#
# MAPFILE HEADER START
#
# WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
# Object versioning must comply with the rules detailed in
#
#       usr/src/lib/README.mapfiles
#
# You should not be making modifications here until you've read the most current
# copy of that file. If you need help, contact a gatekeeper for guidance.
#
# MAPFILE HEADER END
#

$mapfile_version 2

# Due to mistakes made early in the history of this library, there are no
# SUNW_1.1 through SUNW_1.4 symbols, but they are now kept as placeholders.
# Don't add any symbols to these versions.

SYMBOL_VERSION SUNW_1.4 {
    global:
        SUNW_1.4;
} SUNW_1.3;
```

*unchanged_portion_omitted_*

```
SYMBOL_VERSION SUNWprivate_1.1 {
    global:
        Lalt_stack;
        Lclearfault;
        Lclearsig;
        Lctlfd;
        Ldstop;
        Lfree;
        Lgetareg;
        Lgrab;
        Lgrab_error;
        _libproc_debug;
        Lmain_stack;
        Lprochandle;
        Lpsinfo;
        Lputareg;
        Lsetrun;
        Lstack;
        Lstate;
        Lstatus;
        Lstop;
        Lsync;
        Lwait;
        Lxecbkpt;
        Lxecwapt;
        Paddr_to_ctf;
        Paddr_to_loadobj;
        Paddr_to_map;
        Paddr_to_text_map;
        Pasfd;
        Pclearfault;
        Pclearsig;
        Pcontent;
        Pcreate;
        Pcreate_agent;
        Pcreate_callback;
        Pcreate_error;
        Pcred;
        Pctlfd;
        Pdelbkpt;
        Pdelwapt;
        Pdestroy_agent;
        Pdstop;
        Penv_iter;
        Perror_printf;
        Pexecname;
        Pfault;
        Pfgcore;
        Pfgrab_core;
        Pfree;
        Pgcore;
        Pgetareg;
        Pgetauxval;
        Pgetauxvec;
        Pgetenv;
        Pgrab;
        Pgrab_core;
        Pgrab_error;
        Pgrab_file;
        Pgrab_ops;
        Pisprocdir;
        Pissyscall;
        Pissyscall_prev;
        Plmid;
        Plmid_to_ctf;
        Plmid_to_loadobj;
        Plmid_to_map;
        Plookup_by_addr;
        Plookup_by_name;
        Plwp_alt_stack;
        Plwp_getfpregs;
        Plwp_getname;
```

```
140         Plwp_getpsinfo;
141         Plwp_getregs;
142         Plwp_getspymaster;
143         Plwp_iter;
144         Plwp_iter_all;
145         Plwp_main_stack;
146         Plwp_setfpregs;
147         Plwp_setregs;
148         Plwp_stack;
149         Pmapping_iter;
150         Pmapping_iter_resolved;
151         Pname_to_ctf;
152         Pname_to_loadobj;
153         Pname_to_map;
154         Pobject_iter;
155         Pobject_iter_resolved;
156         Pobjname;
157         Pobjname_resolved;
158         Pplatform;
159         Ppltdest;
160         Ppriv;
161         Pprivinfo;
162         Ppriv_free;
163         Ppsinfo;
164         Pputareg;
165         pr_access;
166         pr_close;
167         pr_creat;
168         Prd_agent;
169         pr_door_info;
170         Pread;
171         Pread_string;
172         Prelease;
173         Preopen;
174         Preset_maps;
175         pr_exit;
176         pr_fcntl;
177         pr_fstat;
178         pr_fstat64;
179         pr_fstatvfs;
180         pr_getitimer;
181         pr_getpeername;
182         pr_getpeerucred;
183         pr_getprojid;
184         pr_getrctl;
185         pr_getrlimit;
186         pr_getrlimit64;
187         pr_getsockname;
188         pr_getsockopt;
189         pr_gettaskid;
190         pr_getzoneid;
191         pr_ioctl;
192         pr_link;
193         pr_llseek;
194         pr_lseek;
195         pr_lstat;
196         pr_lstat64;
197         pr_lwp_exit;
198         pr_memcntl;
199         pr_meminfo;
200         pr_mmap;
201         pr_munmap;
202         proc_arg_grab;
203         proc_arg_psinfo;
204         proc_arg_xgrab;
205         proc_arg_xpsinfo;
```



```
206         proc_content2str;
207         proc_finistdio;
208         proc_fltname;
209         proc_fltset2str;
210         proc_flushstdio;
211         proc_free_priv;
212         proc_get_auxv;
213         proc_get_cred;
214         proc_get_priv;
215         proc_get_psinfo;
216         proc_get_secflags;
217         proc_get_status;
218         proc_initstdio;
219         proc_lwp_in_set;
220         proc_lwp_range_valid;
221         proc_signame;
222         proc_sigset2str;
223         proc_str2content;
224         proc_str2flt;
225         proc_str2fltset;
226         proc_str2sig;
227         proc_str2sigset;
228         proc_str2sys;
229         proc_str2sysset;
230         proc_sysname;
231         proc_sysset2str;
232         proc_unctrl_psinfo;
233         proc_walk;
234         pr_open;
235         pr_processor_bind;
236         pr_rename;
237         pr_setitimer;
238         pr_setprojrctl;
239         pr_setrctl;
240         pr_setrlimit;
241         pr_setrlimit64;
242         pr_settaskid;
243         pr_sigaction;
244         pr_stat;
245         pr_stat64;
246         pr_statvfs;
247         pr_unlink;
248         pr_waitid;
249         pr_zmap;
250         Pset_procfs_path;
251         Psetbkpt;
252         Psetcred;
253         Psetfault;
254         Psetflags;
255         Psetpriv;
256         Psetrun;
257         Psetsignal;
258         Psetsysentry;
259         Psetsysexit;
260         Psetwapt;
261         Psetzoneid;
262         Psignal;
263         ps_lcontinue;
264         ps_lgetfpregs;
265         ps_lgetregs;
266         ps_lsetfpregs;
267         ps_lsetregs;
268         ps_lstop;
269         ps_pauxv;
270         ps_pbrandname;
271         ps_pcontinue;
```

```
272         ps_pdmodel;
273         ps_pdread               { FLAGS = NODYNSORT };  # Alias of ps_pread
274         ps_pdwrite              { FLAGS = NODYNSORT };  # Alias of ps_pwrite
275         ps_pglobal_lookup;
276         ps_pglobal_sym;
277         ps_plog;
278         ps_pread;
279         ps_pstop;
280         ps_ptread               { FLAGS = NODYNSORT };  # Alias of ps_pread
281         ps_ptwrite              { FLAGS = NODYNSORT };  # Alias of ps_pwrite
282         ps_pwrite;
283         Psecflags;
284         Psecflags_free;
285         Pstack_iter;
286         Pstate;
287         Pstatus;
288         Pstop;
289         Pstopstatus;
290         Psymbol_iter;
291         Psymbol_iter_by_addr;
292         Psymbol_iter_by_lmid;
293         Psymbol_iter_by_name;
294         Psync;
295         Psyscall;
296         Psysentry;
297         Psysexit;
298         Puname;
299         Punsetflags;
300         Pupdate_maps;
301         Pupdate_syms;
302         Pwait;
303         Pwrite;
304         Pxcreate;
305         Pxecbkpt;
306         Pxecwapt;
307         Pxlookup_by_addr;
308         Pxlookup_by_addr_resolved;
309         Pxlookup_by_name;
310         Pxsymbol_iter;
311         Pzonename;
312         Pzonepath;
313         Pzoneroot;
314         Pfdinfo_iter;

316 $if _x86 && _ELF32
317         Pldt;
318         proc_get_ldt;
319         ps_lgetLDT;
320 $endif

322 $if _sparc
323         Plwp_getgwindows;
324         Plwp_getxregs;
325         Plwp_setxregs;
326         ps_lgetxregs;
327         ps_lgetxregsize;
328         ps_lsetxregs;

330 $if _ELF64
331         Plwp_getasrs;
332         Plwp_setasrs;
333 $endif
334 $endif

336     local:
337         *;
```



```
338 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   26689 Mon Oct 15 13:25:21 2018
new/usr/src/man/man1/ps.1
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 .\"
   2 .\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
   3 .\" permission to reproduce portions of its copyrighted documentation.
   4 .\" Original documentation from The Open Group can be obtained online at
   5 .\" http://www.opengroup.org/bookstore/.
   6 .\"
   7 .\" The Institute of Electrical and Electronics Engineers and The Open
   8 .\" Group, have given us permission to reprint portions of their
   9 .\" documentation.
  10 .\"
  11 .\" In the following statement, the phrase ‘‘this text’’ refers to portions
  12 .\" of the system documentation.
  13 .\"
  14 .\" Portions of this text are reprinted and reproduced in electronic form
  15 .\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
  16 .\" Standard for Information Technology -- Portable Operating System
  17 .\" Interface (POSIX), The Open Group Base Specifications Issue 6,
  18 .\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
  19 .\" Engineers, Inc and The Open Group.  In the event of any discrepancy
  20 .\" between these versions and the original IEEE and The Open Group
  21 .\" Standard, the original IEEE and The Open Group Standard is the referee
  22 .\" document.  The original Standard can be obtained online at
  23 .\" http://www.opengroup.org/unix/online.html.
  24 .\"
  25 .\" This notice shall appear on any product containing this material.
  26 .\"
  27 .\" The contents of this file are subject to the terms of the
  28 .\" Common Development and Distribution License (the "License").
  29 .\" You may not use this file except in compliance with the License.
  30 .\"
  31 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  32 .\" or http://www.opensolaris.org/os/licensing.
  33 .\" See the License for the specific language governing permissions
  34 .\" and limitations under the License.
  35 .\"
  36 .\" When distributing Covered Code, include this CDDL HEADER in each
  37 .\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  38 .\" If applicable, add the following below this CDDL HEADER, with the
  39 .\" fields enclosed by brackets "[]" replaced with your own identifying
  40 .\" information: Portions Copyright [yyyy] [name of copyright owner]
  41 .\"
  42 .\"
  43 .\" Copyright 1989 AT&T
  44 .\" Portions Copyright (c) 1992, X/Open Company Limited All Rights Reserved
  45 .\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved
  46 .\" Copyright (c) 2013 Gary Mills
  47 .\" Copyright (c) 2018, Joyent, Inc.
  47 .\" Copyright (c) 2017, Joyent, Inc.
  48 .\"
  49 .TH PS 1 "August 22, 2018"
  49 .TH PS 1 "Jun 13, 2017"
  50 .SH NAME
  51 ps \- report process status
  52 .SH SYNOPSIS
  53 .LP
  54 .nf
  55 \fBps\fR [\fB-aAcdefjHlLPWyZ\fR] [\fB-g\fR \fIgrplist\fR] [\fB-h\fR \fIlgrplist\
  56      [\fB-n\fR \fInamelist\fR] [\fB-o\fR \fIformat\fR]... [\fB-p\fR \fIproclist\
  57      [\fB-s\fR \fIsidlist\fR] [\fB-t\fR \fIterm\fR] [\fB-u\fR \fIuidlist\fR] [\f
  58      [\fB-G\fR \fIgidlist\fR] [\fB-z\fR \fIzonelist\fR]
```



```
  59 .fi

  61 .SH DESCRIPTION
  62 .LP
  63 The \fBps\fR command prints information about active processes. Without
  64 options, \fBps\fR prints information about processes that have the same
  65 effective user \fBID\fR and the same controlling terminal as the invoker. The
  66 output contains only the process \fBID\fR, terminal identifier, cumulative
  67 execution time, and the command name. Otherwise, the information that is
  68 displayed is controlled by the options.
  69 .sp
  70 .LP
  71 Some options accept lists as arguments. Items in a list can be either separated
  72 by commas or else enclosed in quotes and separated by commas or spaces. Values
  73 for \fIproclist\fR and \fIgrplist\fR must be numeric.
  74 .SH OPTIONS
  75 .LP
  76 The following options are supported:
  77 .sp
  78 .ne 2
  79 .na
  80 \fB\fB-a\fR\fR
  81 .ad
  82 .RS 15n
  83 Lists information about \fBa\fRll processes most frequently requested: all
  84 those except session leaders and processes not associated with a terminal.
  85 .sp
  86 This option is ignored when the \fB-e\fR option is also specified.
  87 .RE

  89 .sp
  90 .ne 2
  91 .na
  92 \fB\fB-A\fR\fR
  93 .ad
  94 .RS 15n
  95 Lists information for all processes. Identical to \fB-e\fR, below.
  96 .RE

  98 .sp
  99 .ne 2
 100 .na
 101 \fB\fB-c\fR\fR
 102 .ad
 103 .RS 15n
 104 Prints information in a format that reflects scheduler properties as described
 105 in \fBpriocntl\fR(1). The \fB-c\fR option affects the output of the \fB-f\fR
 106 and \fB-l\fR options, as described below.
 107 .RE

 109 .sp
 110 .ne 2
 111 .na
 112 \fB\fB-d\fR\fR
 113 .ad
 114 .RS 15n
 115 Lists information about all processes except session leaders.
 116 .RE

 118 .sp
 119 .ne 2
 120 .na
 121 \fB\fB-e\fR\fR
 122 .ad
 123 .RS 15n
 124 Lists information about \fBe\fRvery process now running.
```

```
125 .sp
126 When the \fB-e\fR option is specified, options \fB-z\fR, \fB-t\fR, \fB-u\fR,
127 \fB-U\fR, \fB-g\fR, \fB-G\fR, \fB-p\fR, \fB-h\fR, \fB-s\fR and \fB-a\fR
128 have no effect.
129 .RE

131 .sp
132 .ne 2
133 .na
134 \fB\fB-f\fR\fR
135 .ad
136 .RS 15n
137 Generates a \fBf\fRull listing. (See below for significance of columns in a
138 full listing.)
139 .RE

141 .sp
142 .ne 2
143 .na
144 \fB\fB-g\fR \fIgrplist\fR\fR
145 .ad
146 .RS 15n
147 Lists only process data whose group leader's \fBID\fR number(s) appears in
148 \fIgrplist\fR. (A group leader is a process whose process \fBID\fR number is
149 identical to its process group \fBID\fR number.)
150 .sp
151 This option is ignored when the \fB-e\fR option is also specified.
152 .RE

154 .sp
155 .ne 2
156 .na
157 \fB\fB-G\fR \fIgidlist\fR\fR
158 .ad
159 .RS 15n
160 Lists information for processes whose real group ID numbers are given in
161 \fIgidlist\fR. The \fIgidlist\fR must be a single argument in the form of a
162 blank- or comma-separated list.
163 .sp
164 This option is ignored when the \fB-e\fR option is also specified.
165 .RE

167 .sp
168 .ne 2
169 .na
170 \fB\fB-h\fR \fIlgrplist\fR\fR
171 .ad
172 .RS 15n
173 Lists only the processes homed to the specified \fIlgrplist\fR. Nothing is
174 listed for any invalid group specified in \fIlgrplist\fR.
175 .sp
176 This option is ignored when the \fB-e\fR option is also specified.
177 .RE

179 .sp
180 .ne 2
181 .na
182 \fB\fB-H\fR\fR
183 .ad
184 .RS 15n
185 Prints the home lgroup of the process under an additional column header, LGRP.
186 .RE

188 .sp
189 .ne 2
190 .na
```



```
191 \fB\fB-j\fR\fR
192 .ad
193 .RS 15n
194 Prints session \fBID\fR and process group \fBID\fR.
195 .RE

197 .sp
198 .ne 2
199 .na
200 \fB\fB-l\fR\fR
201 .ad
202 .RS 15n
203 Generates a \fBl\fRong listing. (See below.)
204 .RE

206 .sp
207 .ne 2
208 .na
209 \fB\fB-L\fR\fR
210 .ad
211 .RS 15n
212 Prints information about each light weight process (\fIlwp\fR) in each selected
213 process. (See below.)
214 .RE

216 .sp
217 .ne 2
218 .na
219 \fB\fB-n\fR \fInamelist\fR\fR
220 .ad
221 .RS 15n
222 Specifies the name of an alternative system \fInamelist\fR file in place of the
223 default. This option is accepted for compatibility, but is ignored.
224 .RE

226 .sp
227 .ne 2
228 .na
229 \fB\fB-o\fR \fIformat\fR\fR
230 .ad
231 .RS 15n
232 Prints information according to the format specification given in \fIformat\fR.
233 This is fully described in \fBDISPLAY FORMATS\fR. Multiple \fB-o\fR options can
234 be specified; the format specification is interpreted as the
235 space-character-separated concatenation of all the \fIformat\fR
236 option-arguments.
237 .RE

239 .sp
240 .ne 2
241 .na
242 \fB\fB-p\fR \fIproclist\fR\fR
243 .ad
244 .RS 15n
245 Lists only process data whose process \fBID\fR numbers are given in
246 \fIproclist\fR.
247 .sp
248 This option is ignored when the \fB-e\fR option is also specified.
249 .RE

251 .sp
252 .ne 2
253 .na
254 \fB\fB-P\fR\fR
255 .ad
256 .RS 15n
```

```
257 Prints the number of the processor to which the process or lwp is bound, if
258 any, under an additional column header, \fBPSR\fR.
259 .RE

261 .sp
262 .ne 2
263 .na
264 \fB\fB-s\fR \fIsidlist\fR\fR
265 .ad
266 .RS 15n
267 Lists information on all session leaders whose \fBID\fRs appear in
268 \fIsidlist\fR.
269 .sp
270 This option is ignored when the \fB-e\fR option is also specified.
271 .RE

273 .sp
274 .ne 2
275 .na
276 \fB\fB-t\fR \fIterm\fR\fR
277 .ad
278 .RS 15n
279 Lists only process data associated with \fIterm\fR. Terminal identifiers are
280 specified as a device file name, and an identifier. For example, \fBterm/a\fR,
281 or \fBpts/0\fR.
282 .sp
283 This option is ignored when the \fB-e\fR option is also specified.
284 .RE

286 .sp
287 .ne 2
288 .na
289 \fB\fB-u\fR \fIuidlist\fR\fR
290 .ad
291 .RS 15n
292 Lists only process data whose effective user \fBID\fR number or login name is
293 given in \fIuidlist\fR. In the listing, the numerical user \fBID\fR is printed
294 unless you give the \fB-f\fR option, which prints the login name.
295 .sp
296 This option is ignored when the \fB-e\fR option is also specified.
297 .RE

299 .sp
300 .ne 2
301 .na
302 \fB\fB-U\fR \fIuidlist\fR\fR
303 .ad
304 .RS 15n
305 Lists information for processes whose real user \fBID\fR numbers or login names
306 are given in \fIuidlist\fR. The \fIuidlist\fR must be a single argument in the
307 form of a blank- or comma-separated list.
308 .sp
309 This option is ignored when the \fB-e\fR option is also specified.
310 .RE

312 .sp
313 .ne 2
314 .na
315 \fB\fB-W\fR\fR
316 .ad
317 .RS 15n
318 Truncate long names even when \fBps\fR would normally print them
319 in full.
320 A trailing asterisk marks a long name that has been truncated
321 to fit the column.
322 .RE
```



```
324 .sp
325 .ne 2
326 .na
327 \fB\fB-y\fR\fR
328 .ad
329 .RS 15n
330 Under a long listing (\fB-l\fR), omits the obsolete \fBF\fR and \fBADDR\fR
331 columns and includes an \fBRSS\fR column to report the resident set size of the
332 process. Under the \fB-y\fR option, both \fBRSS\fR and \fBSZ\fR (see below) is
333 reported in units of kilobytes instead of pages.
334 .RE

336 .sp
337 .ne 2
338 .na
339 \fB\fB-z\fR \fIzonelist\fR\fR
340 .ad
341 .RS 15n
342 Lists only processes in the specified zones. Zones can be specified either by
343 name or ID. This option is only useful when executed in the global zone.
344 .sp
345 This option is ignored when the \fB-e\fR option is also specified.
346 .RE

348 .sp
349 .ne 2
350 .na
351 \fB\fB-Z\fR\fR
352 .ad
353 .RS 15n
354 Prints the name of the zone with which the process is associated under an
355 additional column header, \fBZONE\fR. The \fBZONE\fR column width is limited to
356 8 characters. Use \fBps\fR \fB-eZ\fR for a quick way to see information about
357 every process now running along with the associated zone name. Use
358 .sp
359 .in +2
360 .nf
361 ps -eo zone,uid,pid,ppid,time,comm,...
362 .fi
363 .in -2
364 .sp

366 to see zone names wider than 8 characters.
367 .RE

369 .sp
370 .LP
371 Many of the options shown are used to select processes to list. If any are
372 specified, the default list is ignored and \fBps\fR selects the processes
373 represented by the inclusive OR of all the selection-criteria options.
374 .SH DISPLAY FORMATS
375 .LP
376 Under the \fB-f\fR option, \fBps\fR tries to determine the command name and
377 arguments given when the process was created by examining the user block.
378 Failing this, the command name is printed, as it would have appeared without
379 the \fB-f\fR option, in square brackets.
380 .sp
381 .LP
382 The column headings and the meaning of the columns in a \fBps\fR listing are
383 given below; the letters \fBf\fR and \fBl\fR indicate the option (f\fBull\fR or
384 \fBl\fRong, respectively) that causes the corresponding heading to appear;
385 \fBall\fR means that the heading always appears. \fBNote:\fR These two options
386 determine only what information is provided for a process; they do not
387 determine which processes are listed.
388 .sp
```

```
.ne 2
.na
\fB\fBF\fR(l)\fR
.ad
.RS 14n
Flags (hexadecimal and additive) associated with the process. These flags are
available for historical purposes; no meaning should be currently ascribed to
them.
.RE

.sp
.ne 2
.na
\fB\fBS\fR (l)\fR
.ad
.RS 14n
The state of the process:
.sp
.ne 2
.na
\fBO\fR
.ad
.RS 5n
Process is running on a processor.
.RE

.sp
.ne 2
.na
\fBS\fR
.ad
.RS 5n
Sleeping: process is waiting for an event to complete.
.RE

.sp
.ne 2
.na
\fBR\fR
.ad
.RS 5n
Runnable: process is on run queue.
.RE

.sp
.ne 2
.na
\fBT\fR
.ad
.RS 5n
Process is stopped, either by a job control signal or because it is being
traced.
.RE

.sp
.ne 2
.na
\fBW\fR
.ad
.RS 5n
Waiting: process is waiting for CPU usage to drop to the CPU-caps enforced
limits.
.RE

.sp
.ne 2
```



```
.na
\fBZ\fR
.ad
.RS 5n
Zombie state: process terminated and parent not waiting.
.RE

.RE

.sp
.ne 2
.na
\fB\fBUID\fR (f,l)\fR
.ad
.RS 14n
The effective user \fBID\fR number of the process (the login name is printed
under the \fB-f\fR option).
A trailing asterisk marks a long name that has been truncated
to fit the column.
.RE

.sp
.ne 2
.na
\fB\fBPID\fR(all)\fR
.ad
.RS 14n
The process \fBID\fR of the process (this datum is necessary in order to kill a
process).
.RE

.sp
.ne 2
.na
\fB\fBPPID\fR(f,l)\fR
.ad
.RS 14n
The process \fBID\fR of the parent process.
.RE

.sp
.ne 2
.na
\fB\fBC\fR(f,l)\fR
.ad
.RS 14n
Processor utilization for scheduling (obsolete). Not printed when the \fB-c\fR
option is used.
.RE

.sp
.ne 2
.na
\fB\fBCLS\fR(f,l)\fR
.ad
.RS 14n
Scheduling class. Printed only when the \fB-c\fR option is used.
.RE

.sp
.ne 2
.na
\fB\fBPRI\fR(l)\fR
.ad
.RS 14n
The priority of the process. Without the \fB-c\fR option, higher numbers mean
```

```
521 lower priority. With the \fB-c\fR option, higher numbers mean higher priority.
522 .RE

524 .sp
525 .ne 2
526 .na
527 \fB\fBNI\fR(l)\fR
528 .ad
529 .RS 14n
530 Nice value, used in priority computation. Not printed when the \fB-c\fR option
531 is used. Only processes in the certain scheduling classes have a nice value.
532 .RE

534 .sp
535 .ne 2
536 .na
537 \fB\fBADDR\fR(l)\fR
538 .ad
539 .RS 14n
540 The memory address of the process.
541 .RE

543 .sp
544 .ne 2
545 .na
546 \fB\fBSZ\fR(l)\fR
547 .ad
548 .RS 14n
549 The total size of the process in virtual memory, including all mapped files and
550 devices, in pages. See \fBpagesize\fR(1).
551 .RE

553 .sp
554 .ne 2
555 .na
556 \fB\fBWCHAN\fR(l)\fR
557 .ad
558 .RS 14n
559 The address of an event for which the process is sleeping (if blank, the
560 process is running).
561 .RE

563 .sp
564 .ne 2
565 .na
566 \fB\fBSTIME\fR(f)\fR
567 .ad
568 .RS 14n
569 The starting time of the process, given in hours, minutes, and seconds. (A
570 process begun more than twenty-four hours before the \fBps\fR inquiry is
571 executed is given in months and days.)
572 .RE

574 .sp
575 .ne 2
576 .na
577 \fB\fBTTY\fR(all)\fR
578 .ad
579 .RS 14n
580 The controlling terminal for the process (the message, \fB?\fR, is printed when
581 there is no controlling terminal).
582 .RE

584 .sp
585 .ne 2
586 .na
```



```
587 \fB\fBTIME\fR(all)\fR
588 .ad
589 .RS 14n
590 The cumulative execution time for the process.
591 .RE

593 .sp
594 .ne 2
595 .na
596 \fB\fBLTIME\fR(all)\fR
597 .ad
598 .RS 14n
599 The execution time for the lwp being reported.
600 .RE

602 .sp
603 .ne 2
604 .na
605 \fB\fBCMD\fR(all)\fR
606 .ad
607 .RS 14n
608 The command name (the full command name and its arguments, up to a limit of 80
609 characters, are printed under the \fB-f\fR option).
610 .RE

612 .sp
613 .LP
614 The following two additional columns are printed when the \fB-j\fR option is
615 specified:
616 .sp
617 .ne 2
618 .na
619 \fB\fBPGID\fR\fR
620 .ad
621 .RS 8n
622 The process ID of the process group leader.
623 .RE

625 .sp
626 .ne 2
627 .na
628 \fB\fBSID\fR\fR
629 .ad
630 .RS 8n
631 The process ID of the session leader.
632 .RE

634 .sp
635 .LP
636 The following two additional columns are printed when the \fB-L\fR option is
637 specified:
638 .sp
639 .ne 2
640 .na
641 \fB\fBLWP\fR\fR
642 .ad
643 .RS 8n
644 The lwp ID of the lwp being reported.
645 .RE

647 .sp
648 .ne 2
649 .na
650 \fB\fBNLWP\fR\fR
651 .ad
652 .RS 8n
```

```
653 The number of lwps in the process (if \fB-f\fR is also specified).
654 .RE

656 .sp
657 .LP
658 Under the \fB-L\fR option, one line is printed for each lwp in the process and
659 the time-reporting fields \fBSTIME\fR and \fBLTIME\fR show the values for the
660 lwp, not the process. A traditional single-threaded process contains only one
661 lwp.
662 .sp
663 .LP
664 A process that has exited and has a parent, but has not yet been waited for by
665 the parent, is marked \fB<defunct>\fR\&.
666 .SS "\fB-o\fR format"
667 .LP
668 The \fB-o\fR option allows the output format to be specified under user
669 control.
670 .sp
671 .LP
672 The format specification must be a list of names presented as a single
673 argument, blank- or comma-separated. Each variable has a default header. The
674 default header can be overridden by appending an equals sign and the new text
675 of the header. The rest of the characters in the argument is used as the header
676 text. The fields specified are written in the order specified on the command
677 line, and should be arranged in columns in the output. The field widths are
678 selected by the system to be at least as wide as the header text (default or
679 overridden value). If the header text is null, such as \fB-o\fR \fIuser=,\fR
680 the field width is at least as wide as the default header text.
681 Long names are not truncated in this mode.
682 If all header text fields are null, no header line is written.
683 .sp
684 .LP
685 The following names are recognized in the POSIX locale:
686 .sp
687 .ne 2
688 .na
689 \fB\fBuser\fR\fR
690 .ad
691 .RS 10n
692 The effective user \fBID\fR of the process. This is the textual user \fBID\fR,
693 if it can be obtained and the field width permits, or a decimal representation
694 otherwise.
695 .RE

697 .sp
698 .ne 2
699 .na
700 \fB\fBruser\fR\fR
701 .ad
702 .RS 10n
703 The real user \fBID\fR of the process. This is the textual user \fBID\fR, if it
704 can be obtained and the field width permits, or a decimal representation
705 otherwise.
706 .RE

708 .sp
709 .ne 2
710 .na
711 \fB\fBgroup\fR\fR
712 .ad
713 .RS 10n
714 The effective group \fBID\fR of the process. This is the textual group
715 \fBID,\fR if it can be obtained and the field width permits, or a decimal
716 representation otherwise.
717 .RE
```



```
719 .sp
720 .ne 2
721 .na
722 \fB\fBrgroup\fR\fR
723 .ad
724 .RS 10n
725 The real group \fBID\fR of the process. This is the textual group \fBID,\fR if
726 it can be obtained and the field width permits, or a decimal representation
727 otherwise.
728 .RE

730 .sp
731 .ne 2
732 .na
733 \fB\fBpid\fR\fR
734 .ad
735 .RS 10n
736 The decimal value of the process \fBID\fR.
737 .RE

739 .sp
740 .ne 2
741 .na
742 \fB\fBppid\fR\fR
743 .ad
744 .RS 10n
745 The decimal value of the parent process \fBID\fR.
746 .RE

748 .sp
749 .ne 2
750 .na
751 \fB\fBpgid\fR\fR
752 .ad
753 .RS 10n
754 The decimal value of the process group \fBID.\fR
755 .RE

757 .sp
758 .ne 2
759 .na
760 \fB\fBpcpu\fR\fR
761 .ad
762 .RS 10n
763 The ratio of CPU time used recently to CPU time available in the same period,
764 expressed as a percentage. The meaning of ''recently'' in this context is
765 unspecified. The CPU time available is determined in an unspecified manner.
766 .RE

768 .sp
769 .ne 2
770 .na
771 \fB\fBvsz\fR\fR
772 .ad
773 .RS 10n
774 The total size of the process in virtual memory, in kilobytes.
775 .RE

777 .sp
778 .ne 2
779 .na
780 \fB\fBnice\fR\fR
781 .ad
782 .RS 10n
783 The decimal value of the system scheduling priority of the process. See
784 \fBnice\fR(1).
```

```
.RE

.sp
.ne 2
.na
\fB\fBetime\fR\fR
.ad
.RS 10n
In the POSIX locale, the elapsed time since the process was started, in the
form:
.sp
\fB[[\fR\fIdd\fR-\fB]\fR\fIhh\fR:\fB]\fR\fImm\fR:\fIss\fR
.sp
where
.sp
.ne 2
.na
\fB\fIdd\fR\fR
.ad
.RS 6n
is the number of days
.RE

.sp
.ne 2
.na
\fB\fIhh\fR\fR
.ad
.RS 6n
is the number of hours
.RE

.sp
.ne 2
.na
\fB\fImm\fR\fR
.ad
.RS 6n
is the number of minutes
.RE

.sp
.ne 2
.na
\fB\fIss\fR\fR
.ad
.RS 6n
is the number of seconds
.RE

The \fIdd\fR field is a decimal integer. The \fIhh\fR, \fImm\fR and \fIss\fR
fields is two-digit decimal integers padded on the left with zeros.
.RE

.sp
.ne 2
.na
\fB\fBtime\fR\fR
.ad
.RS 10n
In the POSIX locale, the cumulative CPU time of the process in the form:
.sp
\fB[\fR\fIdd\fR-\fB]\fR\fIhh\fR:\fImm\fR:\fIss\fR
.sp
The \fIdd\fR, \fIhh\fR, \fImm\fR, and \fIss\fR fields is as described in the
\fBetime\fR specifier.
```



```
.RE

.sp
.ne 2
.na
\fB\fBtty\fR\fR
.ad
.RS 10n
The name of the controlling terminal of the process (if any) in the same format
used by the \fBwho\fR(1) command.
.RE

.sp
.ne 2
.na
\fB\fBcomm\fR\fR
.ad
.RS 10n
The name of the command being executed (\fBargv[0]\fR value) as a string.
.RE

.sp
.ne 2
.na
\fB\fBargs\fR\fR
.ad
.RS 10n
The command with all its arguments as a string. The implementation might
truncate this value to the field width; it is implementation-dependent whether
any further truncation occurs. It is unspecified whether the string represented
is a version of the argument list as it was passed to the command when it
started, or is a version of the arguments as they might have been modified by
the application. Applications cannot depend on being able to modify their
argument list and having that modification be reflected in the output of
\fBps\fR. The Solaris implementation limits the string to 80 bytes; the string
is the version of the argument list as it was passed to the command when it
started.
.RE

.sp
.LP
The following names are recognized in the Solaris implementation:
.sp
.ne 2
.na
\fB\fBf\fR\fR
.ad
.RS 11n
Flags (hexadecimal and additive) associated with the process.
.RE

.sp
.ne 2
.na
\fB\fBs\fR\fR
.ad
.RS 11n
The state of the process.
.RE

.sp
.ne 2
.na
\fB\fBc\fR\fR
.ad
.RS 11n
```

```
 917 Processor utilization for scheduling (obsolete).
 918 .RE

 920 .sp
 921 .ne 2
 922 .na
 923 \fB\fBuid\fR\fR
 924 .ad
 925 .RS 11n
 926 The effective user \fBID\fR number of the process as a decimal integer.
 927 .RE

 929 .sp
 930 .ne 2
 931 .na
 932 \fB\fBruid\fR\fR
 933 .ad
 934 .RS 11n
 935 The real user \fBID\fR number of the process as a decimal integer.
 936 .RE

 938 .sp
 939 .ne 2
 940 .na
 941 \fB\fBgid\fR\fR
 942 .ad
 943 .RS 11n
 944 The effective group \fBID\fR number of the process as a decimal integer.
 945 .RE

 947 .sp
 948 .ne 2
 949 .na
 950 \fB\fBrgid\fR\fR
 951 .ad
 952 .RS 11n
 953 The real group \fBID\fR number of the process as a decimal integer.
 954 .RE

 956 .sp
 957 .ne 2
 958 .na
 959 \fB\fBprojid\fR\fR
 960 .ad
 961 .RS 11n
 962 The project \fBID\fR number of the process as a decimal integer.
 963 .RE

 965 .sp
 966 .ne 2
 967 .na
 968 \fB\fBproject\fR\fR
 969 .ad
 970 .RS 11n
 971 The project \fBID\fR of the process as a textual value if that value can be
 972 obtained; otherwise, as a decimal integer.
 973 .RE

 975 .sp
 976 .ne 2
 977 .na
 978 \fB\fBzoneid\fR\fR
 979 .ad
 980 .RS 11n
 981 The zone \fBID\fR number of the process as a decimal integer.
 982 .RE
```



```
 984 .sp
 985 .ne 2
 986 .na
 987 \fB\fBzone\fR\fR
 988 .ad
 989 .RS 11n
 990 The zone \fBID\fR of the process as a textual value if that value can be
 991 obtained; otherwise, as a decimal integer.
 992 .RE

 994 .sp
 995 .ne 2
 996 .na
 997 \fB\fBsid\fR\fR
 998 .ad
 999 .RS 11n
1000 The process ID of the session leader.
1001 .RE

1003 .sp
1004 .ne 2
1005 .na
1006 \fB\fBtaskid\fR\fR
1007 .ad
1008 .RS 11n
1009 The task \fBID\fR of the process.
1010 .RE

1012 .sp
1013 .ne 2
1014 .na
1015 \fB\fBclass\fR\fR
1016 .ad
1017 .RS 11n
1018 The scheduling class of the process.
1019 .RE

1021 .sp
1022 .ne 2
1023 .na
1024 \fB\fBpri\fR\fR
1025 .ad
1026 .RS 11n
1027 The priority of the process. Higher numbers mean higher priority.
1028 .RE

1030 .sp
1031 .ne 2
1032 .na
1033 \fB\fBopri\fR\fR
1034 .ad
1035 .RS 11n
1036 The obsolete priority of the process. Lower numbers mean higher priority.
1037 .RE

1039 .sp
1040 .ne 2
1041 .na
1042 \fB\fBlwp\fR\fR
1043 .ad
1044 .RS 11n
1045 The decimal value of the lwp \fBID\fR. Requesting this formatting option causes
1046 one line to be printed for each lwp in the process.
1047 .RE
```

```
1049 .sp
1050 .ne 2
1051 .na
1052 \fB\fBlwpname\fR\fR
1053 .ad
1054 .RS 11n
1055 The name of the lwp, if set. Requesting this formatting option causes
1056 one line to be printed for each lwp in the process.
1057 .RE

1059 .sp
1060 .ne 2
1061 .na
1062 \fB\fBnlwp\fR\fR
1063 .ad
1064 .RS 11n
1065 The number of lwps in the process.
1066 .RE

1068 .sp
1069 .ne 2
1070 .na
1071 \fB\fBpsr\fR\fR
1072 .ad
1073 .RS 11n
1074 The number of the processor to which the process or lwp is bound.
1075 .RE

1077 .sp
1078 .ne 2
1079 .na
1080 \fB\fBpset\fR\fR
1081 .ad
1082 .RS 11n
1083 The \fBID\fR of the processor set to which the process or lwp is bound.
1084 .RE

1086 .sp
1087 .ne 2
1088 .na
1089 \fB\fBaddr\fR\fR
1090 .ad
1091 .RS 11n
1092 The memory address of the process.
1093 .RE

1095 .sp
1096 .ne 2
1097 .na
1098 \fB\fBosz\fR\fR
1099 .ad
1100 .RS 11n
1101 The total size of the process in virtual memory, in pages.
1102 .RE

1104 .sp
1105 .ne 2
1106 .na
1107 \fB\fBwchan\fR\fR
1108 .ad
1109 .RS 11n
1110 The address of an event for which the process is sleeping (if \(mi, the process
1111 is running).
1112 .RE

1114 .sp
```



```
1115 .ne 2
1116 .na
1117 \fB\fBstime\fR\fR
1118 .ad
1119 .RS 11n
1120 The starting time or date of the process, printed with no blanks.
1121 .RE

1123 .sp
1124 .ne 2
1125 .na
1126 \fB\fBrss\fR\fR
1127 .ad
1128 .RS 11n
1129 The resident set size of the process, in kilobytes. The \fBrss\fR value
1130 reported by \fBps\fR is an estimate provided by \fBproc\fR(4) that might
1131 underestimate the actual resident set size. Users who wish to get more accurate
1132 usage information for capacity planning should use \fBpmap\fR(1) \fB-x\fR
1133 instead.
1134 .RE

1136 .sp
1137 .ne 2
1138 .na
1139 \fB\fBpmem\fR\fR
1140 .ad
1141 .RS 11n
1142 The ratio of the process's resident set size to the physical memory on the
1143 machine, expressed as a percentage.
1144 .RE

1146 .sp
1147 .ne 2
1148 .na
1149 \fB\fBfname\fR\fR
1150 .ad
1151 .RS 11n
1152 The first 8 bytes of the base name of the process's executable file.
1153 .RE

1155 .sp
1156 .ne 2
1157 .na
1158 \fB\fBctid\fR\fR
1159 .ad
1160 .RS 11n
1161 The contract ID of the process contract the process is a member of as a decimal
1162 integer.
1163 .RE

1165 .sp
1166 .ne 2
1167 .na
1168 \fB\fBlgrp\fR\fR
1169 .ad
1170 .RS 11n
1171 The home lgroup of the process.
1172 .RE

1174 .sp
1175 .ne 2
1176 .na
1177 \fB\fBdmodel\fR\fR
1178 .ad
1179 .RS 11n
1180 The data model of the process, printed in the same manner as via
```

\fBpflags\fR(1). The currently supported data models are _ILP32 and _LP64.
.RE

.sp
.LP
Only \fBcomm\fR, \fBlwpname\fR, and \fBargs\fR are allowed to contain blank characters; all others, including the Solaris implementation variables, are not.
Only \fBcomm\fR and \fBargs\fR are allowed to contain blank characters; all others, including the Solaris implementation variables, are not.
.sp
.LP
The following table specifies the default header to be used in the POSIX locale corresponding to each format specifier.
.sp

.sp
.TS
box;
c c c c
c c c c .
Format Default Format Default
Specifier   Header Specifier   Header
_
args   COMMAND ppid   PPID
comm   COMMAND rgroup RGROUP
etime   ELAPSED ruser   RUSER
group   GROUP   time   TIME
nice   NI   tty   TT
pcpu   %CPU   user   USER
pgid   PGID   vsz   VSZ
pid   PID
.TE

.sp
.LP
The following table lists the Solaris implementation format specifiers and the default header used with each.
.sp

.sp
.TS
box;
c c c c
c c c c .
Format Default Format Default
Specifier   Header Specifier   Header
_
addr   ADDR   projid PROJID
c   C   project PROJECT
class   CLS   psr   PSR
f   F   rgid   RGID
fname   COMMAND rss   RSS
gid   GID   ruid   RUID
lgrp   LGRP   s   S
lwp   LWP   sid   SID
lwpname LWPNAME stime   STIME
nlwp   NLWP   taskid TASKID
opri   PRI   uid   UID
osz   SZ   wchan   WCHAN
pmem   %MEM   zone   ZONE
pri   PRI   zoneid ZONEID
ctid   CTID
nlwp   NLWP   stime   STIME
opri   PRI   taskid TASKID
osz   SZ   uid   UID
pmem   %MEM   wchan   WCHAN

pri   PRI   zone   ZONE
ctid   CTID   zoneid ZONEID
.TE

.SH EXAMPLES
.LP
\fBExample 1 \fRUsing \fBps\fR Command
.sp
.LP
The command:

.sp
.in +2
.nf
example% \fBps -o user,pid,ppid=MOM -o args\fR
.fi
.in -2
.sp

.sp
.LP
writes the following in the POSIX locale:

.sp
.in +2
.nf
 USER PID   MOM   COMMAND
helene 34   12   ps -o uid,pid,ppid=MOM -o args
.fi
.in -2
.sp

.sp
.LP
The contents of the \fBCOMMAND\fR field need not be the same due to possible truncation.

.SH ENVIRONMENT VARIABLES
.LP
See \fBenviron\fR(5) for descriptions of the following environment variables that affect the execution of \fBps\fR: \fBLANG\fR, \fBLC_ALL\fR, \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, \fBLC_TIME\fR, and \fBNLSPATH\fR.
.sp
.ne 2
.na
\fB\fBCOLUMNS\fR\fR
.ad
.RS 11n
Override the system-selected horizontal screen size, used to determine the number of text columns to display.
.RE

.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 6n
Successful completion.
.RE

.sp
.ne 2

```
.na
\fB\fB>0\fR\fR
.ad
.RS 6n
An error occurred.
.RE

.SH FILES
.ne 2
.na
\fB\fB/dev/pts/*\fR\fR
.ad
.RS 15n

.RE

.sp
.ne 2
.na
\fB\fB/dev/term/*\fR\fR
.ad
.RS 15n
terminal (‘‘tty’’) names searcher files
.RE

.sp
.ne 2
.na
\fB\fB/etc/passwd\fR\fR
.ad
.RS 15n
\fBUID\fR information supplier
.RE

.sp
.ne 2
.na
\fB\fB/proc/*\fR\fR
.ad
.RS 15n
process control files
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled (see USAGE)
_
Interface Stability     Committed
_
Standard        See \fBstandards\fR(5).
.TE

.SH SEE ALSO
.LP
\fBkill\fR(1), \fBlgrpinfo\fR(1), \fBnice\fR(1), \fBpagesize\fR(1),
\fBpmap\fR(1), \fBpriocntl\fR(1), \fBwho\fR(1), \fBgetty\fR(1M), \fBproc\fR(4),
```



```
\fBttysrch\fR(4), \fBattributes\fR(5), \fBenviron\fR(5),
\fBresource_controls\fR(5), \fBstandards\fR(5), \fBzones\fR(5)
.SH NOTES
.LP
Things can change while \fBps\fR is running. The snapshot it gives is true only
for a split-second, and it might not be accurate by the time you see it. Some
data printed for defunct processes is irrelevant.
.sp
.LP
If no options to select processes are specified, \fBps\fR reports all processes
associated with the controlling terminal. If there is no controlling terminal,
there is no report other than the header.
.sp
.LP
\fBps\fR \fB-ef\fR or \fBps\fR \fB-o\fR \fBstime\fR might not report the actual
start of a tty login session, but rather an earlier time, when a getty was last
respawned on the tty line.
.sp
.LP
\fBps\fR is \fBCSI\fR-enabled except for login names (usernames).
```

```
**********************************************************
   17007 Mon Oct 15 13:25:27 2018
new/usr/src/man/man1m/prstat.1m
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2013 Gary Mills
   3 .\" Copyright (c) 2006, 2009 Sun Microsystems, Inc. All Rights Reserved.
   4 .\" Copyright (c) 2018, Joyent, Inc. All Rights Reserved.
   5 .\" The contents of this file are subject to the terms of the Common Development
   6 .\"  See the License for the specific language governing permissions and limitat
   7 .\" the fields enclosed by brackets "[]" replaced with your own identifying info
   8 .TH PRSTAT 1M "September 1, 2018"
   7 .TH PRSTAT 1M "Nov 14, 2014"
   9 .SH NAME
  10 prstat \- report active process statistics
  11 .SH SYNOPSIS
  12 .LP
  13 .nf
  14 \fBprstat\fR [\fB-acHJLmRrtTvWZ\fR] [\fB-d\fR u | d] [\fB-C\fR \fIpsrsetlist\fR]
  15      [\fB-j\fR \fIprojlist\fR] [\fB-k\fR \fItasklist\fR] [\fB-n\fR \fIntop\fR[,\
  16      [\fB-p\fR \fIpidlist\fR] [\fB-P\fR \fIcpulist\fR] [\fB-s\fR \fIkey\fR | \fB
  17      [\fB-u\fR \fIeuidlist\fR] [\fB-U\fR \fIuidlist\fR] [\fB-z\fR \fIzoneidlist\
  18      [\fIinterval\fR [\fIcount\fR]]
  19 .fi

  21 .SH DESCRIPTION
  22 .LP
  23 The \fBprstat\fR utility iteratively examines all active processes on the
  24 system and reports statistics based on the selected output mode and sort order.
  25 \fBprstat\fR provides options to examine only processes matching specified
  26 \fBPID\fRs, \fBUID\fRs, zone \fBID\fRs, \fBCPU\fR \fBID\fRs, and processor set
  27 \fBID\fRs.
  28 .sp
  29 .LP
  30 The \fB-j\fR, \fB-k\fR, \fB-C\fR, \fB-p\fR, \fB-P\fR, \fB-u\fR, \fB-U\fR, and
  31 \fB-z\fR options accept lists as arguments. Items in a list can be either
  32 separated by commas or enclosed in quotes and separated by commas or spaces.
  33 .sp
  34 .LP
  35 If you do not specify an option, \fBprstat\fR examines all processes and
  36 reports statistics sorted by \fBCPU\fR usage.
  37 .SH OPTIONS
  38 .LP
  39 The following options are supported:
  40 .sp
  41 .ne 2
  42 .na
  43 \fB\fB-a\fR\fR
  44 .ad
  45 .sp .6
  46 .RS 4n
  47 Report information about processes and users. In this mode \fBprstat\fR
  48 displays separate reports about processes and users at the same time.
  49 .RE

  51 .sp
  52 .ne 2
  53 .na
  54 \fB\fB-c\fR\fR
  55 .ad
  56 .sp .6
  57 .RS 4n
  58 Print new reports below previous reports instead of overprinting them.
  59 Long names are not truncated in this mode.
```



```
  60 .RE

  62 .sp
  63 .ne 2
  64 .na
  65 \fB\fB-C\fR \fIpsrsetlist\fR\fR
  66 .ad
  67 .sp .6
  68 .RS 4n
  69 Report only processes or lwps that are bound to processor sets in the given
  70 list. Each processor set is identified by an integer as reported by
  71 \fBpsrset\fR(1M). The load averages displayed are the sum of the load averages
  72 of the specified processor sets (see \fBpset_getloadavg\fR(3C)). Processes with
  73 one or more LWPs bound to processor sets in the given list are reported even
  74 when the \fB-L\fR option is not used.
  75 .RE

  77 .sp
  78 .ne 2
  79 .na
  80 \fB\fB-d\fR \fBu | d\fR\fR
  81 .ad
  82 .sp .6
  83 .RS 4n
  84 Specify \fBu\fR for a printed representation of the internal representation of
  85 time. See \fBtime\fR(2). Specify \fBd\fR for standard date format. See
  86 \fBdate\fR(1).
  87 .RE

  89 .sp
  90 .ne 2
  91 .na
  92 \fB\fB-h\fR \fIlgrplist\fR\fR
  93 .ad
  94 .sp .6
  95 .RS 4n
  96 Report only processes or lwps whose home \fIlgroup\fR is in the given list of
  97 \fIlgroups\fR. No processes or lwps will be listed for invalid \fIlgroups\fR.
  98 .RE

 100 .sp
 101 .ne 2
 102 .na
 103 \fB\fB-H\fR\fR
 104 .ad
 105 .sp .6
 106 .RS 4n
 107 Report information about home \fIlgroup\fR. In this mode, \fBprstat\fR adds an
 108 extra column showing process or lwps home \fIlgroup\fR with the header LGRP.
 109 .RE

 111 .sp
 112 .ne 2
 113 .na
 114 \fB\fB-j\fR \fIprojlist\fR\fR
 115 .ad
 116 .sp .6
 117 .RS 4n
 118 Report only processes or lwps whose project \fBID\fR is in the given list. Each
 119 project \fBID\fR can be specified as either a project name or a numerical
 120 project \fBID\fR. See \fBproject\fR(4).
 121 .RE

 123 .sp
 124 .ne 2
 125 .na
```

```
\fB\fB-J\fR\fR
.ad
.sp .6
.RS 4n
Report information about processes and projects. In this mode \fBprstat\fR
displays separate reports about processes and projects at the same time.
A trailing asterisk marks a long name that has been truncated
to fit the column.
.RE

.sp
.ne 2
.na
\fB\fB-k\fR \fItasklist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or lwps whose task \fBID\fR is in \fItasklist\fR.
.RE

.sp
.ne 2
.na
\fB\fB-L\fR\fR
.ad
.sp .6
.RS 4n
Report statistics for each light-weight process (\fBLWP\fR). By default,
\fBprstat\fR reports only the number of \fBLWP\fRs for each process.
.RE

.sp
.ne 2
.na
\fB\fB-m\fR\fR
.ad
.sp .6
.RS 4n
Report microstate process accounting information. In addition to all fields
listed in \fB-v\fR mode, this mode also includes the percentage of time the
process has spent processing system traps, text page faults, data page faults,
waiting for user locks and waiting for \fBCPU\fR (latency time).
.RE

.sp
.ne 2
.na
\fB\fB-n\fR \fIntop\fR[\fI,nbottom\fR]\fR
.ad
.sp .6
.RS 4n
Restrict number of output lines. The \fIntop\fR argument determines how many
lines of process or \fBlwp\fR statistics are reported, and the \fInbottom\fR
argument determines how many lines of user, task, project or zone statistics
are reported if the \fB-a\fR, \fB-t\fR, \fB-T\fR, \fB-J\fR or \fB-Z\fR options
are specified. By default, \fBprstat\fR displays as many lines of output that
fit in a window or terminal. When you specify the \fB-c\fR option or direct the
output to a file, the default values for \fBntop\fR and \fBnbottom\fR are
\fB15\fR and \fB5\fR.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR \fIpidlist\fR\fR
.ad
```



```
.sp .6
.RS 4n
Report only processes whose process \fBID\fR is in the given list.
.RE

.sp
.ne 2
.na
\fB\fB-P\fR \fIcpulist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or \fBlwp\fRs which have most recently executed on a
\fBCPU\fR in the given list. Each \fBCPU\fR is identified by an integer as
reported by \fBpsrinfo\fR(1M).
.RE

.sp
.ne 2
.na
\fB\fB-R\fR\fR
.ad
.sp .6
.RS 4n
Put \fBprstat\fR in the real time scheduling class. When this option is used,
\fBprstat\fR is given priority over time-sharing and interactive processes.
This option is available only for superuser.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Disable lookups for user names and project names. (Note that this does not
apply to lookups for the \fB-j\fR, \fB-u\fR, or \fB-U\fR options.)
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIkey\fR\fR
.ad
.sp .6
.RS 4n
Sort output lines (that is, processes, \fBlwp\fRs, or users) by \fIkey\fR in
descending order. Only one \fIkey\fR can be used as an argument.
.sp
There are five possible key values:
.sp
.ne 2
.na
\fBcpu\fR
.ad
.sp .6
.RS 4n
Sort by process \fBCPU\fR usage. This is the default.
.RE

.sp
.ne 2
.na
\fBpri\fR
.ad
```

```
.sp .6
.RS 4n
Sort by process priority.
.RE

.sp
.ne 2
.na
\fBrss\fR
.ad
.sp .6
.RS 4n
Sort by resident set size.
.RE

.sp
.ne 2
.na
\fBsize\fR
.ad
.sp .6
.RS 4n
Sort by size of process image.
.RE

.sp
.ne 2
.na
\fBtime\fR
.ad
.sp .6
.RS 4n
Sort by process execution time.
.RE

.RE

.sp
.ne 2
.na
\fB\fB-S\fR \fIkey\fR\fR
.ad
.sp .6
.RS 4n
Sort output lines by \fIkey\fR in ascending order. Possible \fIkey\fR values
are the same as for the \fB-s\fR option. See \fB-s\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR\fR
.ad
.sp .6
.RS 4n
Report total usage summary for each user. The summary includes the total number
of processes or \fBLWP\fRs owned by the user, total size of process images,
total resident set size, total cpu time, and percentages of recent cpu time and
system memory.
.RE

.sp
.ne 2
.na
\fB\fB-T\fR\fR
.ad
```



```
.sp .6
.RS 4n
Report information about processes and tasks. In this mode \fBprstat\fR
displays separate reports about processes and tasks at the same time.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIeuidlist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes whose effective user \fBID\fR is in the given list. Each
user \fBID\fR may be specified as either a login name or a numerical user
\fBID\fR.
.RE

.sp
.ne 2
.na
\fB\fB-U\fR \fIuidlis\fRt\fR
.ad
.sp .6
.RS 4n
Report only processes whose real user \fBID\fR is in the given list. Each user
\fBID\fR may be specified as either a login name or a numerical user \fBID\fR.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Report verbose process usage. This output format includes the percentage of
time the process has spent in user mode, in system mode, and sleeping. It also
includes the number of voluntary and involuntary context switches, system calls
and the number of signals received. Statistics that are not reported are marked
with the \fB-\fR sign.
.RE

.sp
.ne 2
.na
\fB\fB-W\fR\fR
.ad
.sp .6
.RS 4n
Truncate long names even when \fBprstat\fR would normally print them
in full.
A trailing asterisk marks a long name that has been truncated
to fit the column.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fIzoneidlist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or LWPs whose zone ID is in the given list. Each zone ID
can be specified as either a zone name or a numerical zone ID. See
\fBzones\fR(5).
```

```
.RE

.sp
.ne 2
.na
\fB\fB-Z\fR\fR
.ad
.sp .6
.RS 4n
Report information about processes and zones. In this mode, \fBprstat\fR
displays separate reports about processes and zones at the same time.
A trailing asterisk marks a long name that has been truncated
to fit the column.
.RE

.SH OUTPUT
.LP
The following list defines the column headings and the meanings of a
\fBprstat\fR report:
.sp
.ne 2
.na
\fBPID\fR
.ad
.sp .6
.RS 4n
The process \fBID\fR of the process.
.RE

.sp
.ne 2
.na
\fBUSERNAME\fR
.ad
.sp .6
.RS 4n
The real user (login) name or real user \fBID\fR.
A trailing asterisk marks a long name that has been truncated
to fit the column.
.RE

.sp
.ne 2
.na
\fBSWAP\fR
.ad
.sp .6
.RS 4n
The total virtual memory size of the process, including all mapped files and
devices, in kilobytes (\fBK\fR), megabytes (\fBM\fR), or gigabytes (\fBG\fR).
.RE

.sp
.ne 2
.na
\fBRSS\fR
.ad
.sp .6
.RS 4n
The resident set size of the process (\fBRSS\fR), in kilobytes (\fBK\fR),
megabytes (\fBM\fR), or gigabytes (\fBG\fR). The RSS value is an estimate
provided by \fBproc\fR(4) that might underestimate the actual resident set
size. Users who want to get more accurate usage information for capacity
planning should use the \fB-x\fR option to \fBpmap\fR(1) instead.
.RE
```



```
.sp
.ne 2
.na
\fBSTATE\fR
.ad
.sp .6
.RS 4n
The state of the process:
.sp
.ne 2
.na
\fBcpu\fIN\fR\fR
.ad
.sp .6
.RS 4n
Process is running on \fBCPU\fR \fIN\fR.
.RE

.sp
.ne 2
.na
\fBsleep\fR
.ad
.sp .6
.RS 4n
Sleeping: process is waiting for an event to complete.
.RE

.sp
.ne 2
.na
\fBwait\fR
.ad
.sp .6
.RS 4n
Waiting: process is waiting for CPU usage to drop to the CPU-caps enforced
limits. See the description of \fBCPU-caps\fR in \fBresource_controls\fR(5).
.RE

.sp
.ne 2
.na
\fBrun\fR
.ad
.sp .6
.RS 4n
Runnable: process in on run queue.
.RE

.sp
.ne 2
.na
\fBzombie\fR
.ad
.sp .6
.RS 4n
Zombie state: process terminated and parent not waiting.
.RE

.sp
.ne 2
.na
\fBstop\fR
.ad
.sp .6
.RS 4n
```

```
Process is stopped.
.RE

.RE

.sp
.ne 2
.na
\fBPRI\fR
.ad
.sp .6
.RS 4n
The priority of the process. Larger numbers mean higher priority.
.RE

.sp
.ne 2
.na
\fBNICE\fR
.ad
.sp .6
.RS 4n
Nice value used in priority computation. Only processes in certain scheduling
classes have a nice value.
.RE

.sp
.ne 2
.na
\fBTIME\fR
.ad
.sp .6
.RS 4n
The cumulative execution time for the process.
.RE

.sp
.ne 2
.na
\fBCPU\fR
.ad
.sp .6
.RS 4n
The percentage of recent \fBCPU\fR time used by the process. If executing in a
non-global \fBzone\fR and the pools facility is active, the percentage will be
that of the processors in the processor set in use by the pool to which the
\fBzone\fR is bound.
.RE

.sp
.ne 2
.na
\fBPROCESS\fR
.ad
.sp .6
.RS 4n
The name of the process (name of executed file).
.RE

.sp
.ne 2
.na
\fBLWP\fR
\fILWPID\fR
.ad
.sp .6
```



```
.RS 4n
The \fBlwp\fR \fBID\fR of the \fBlwp\fR being reported, as well as the LWP
name if any is set.
The \fBlwp\fR \fBID\fR of the \fBlwp\fR being reported.
.RE

.sp
.ne 2
.na
\fBNLWP\fR
.ad
.sp .6
.RS 4n
The number of \fBlwp\fRs in the process.
.RE

.sp
.LP
With the some options, in addition to a number of the column headings shown
above, there are:
.sp
.ne 2
.na
\fBNPROC\fR
.ad
.sp .6
.RS 4n
Number of processes in a specified collection.
.RE

.sp
.ne 2
.na
\fBMEMORY\fR
.ad
.sp .6
.RS 4n
Percentage of memory used by a specified collection of processes.
.RE

.sp
.LP
The following columns are displayed when the \fB-v\fR or \fB-m\fR option is
specified
.sp
.ne 2
.na
\fBUSR\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent in user mode.
.RE

.sp
.ne 2
.na
\fBSYS\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent in system mode.
.RE

.sp
.ne 2
```

```
.na
\fBTRP\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent in processing system traps.
.RE

.sp
.ne 2
.na
\fBTFL\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent processing text page faults.
.RE

.sp
.ne 2
.na
\fBDFL\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent processing data page faults.
.RE

.sp
.ne 2
.na
\fBLCK\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent waiting for user locks.
.RE

.sp
.ne 2
.na
\fBSLP\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent sleeping.
.RE

.sp
.ne 2
.na
\fBLAT\fR
.ad
.sp .6
.RS 4n
The percentage of time the process has spent waiting for CPU.
.RE

.sp
.ne 2
.na
\fBVCX\fR
.ad
.sp .6
.RS 4n
The number of voluntary context switches.
```



```
.RE

.sp
.ne 2
.na
\fBICX\fR
.ad
.sp .6
.RS 4n
The number of involuntary context switches.
.RE

.sp
.ne 2
.na
\fBSCL\fR
.ad
.sp .6
.RS 4n
The number of system calls.
.RE

.sp
.ne 2
.na
\fBSIG\fR
.ad
.sp .6
.RS 4n
The number of signals received.
.RE

.sp
.LP
Under the \fB-L\fR option, one line is printed for each \fBlwp\fR in the
process and some reporting fields show the values for the \fBlwp\fR, not the
process.
.sp
.LP
The following column is displayed when the \fB-H\fR option is specified:
.sp
.ne 2
.na
\fBLGRP\fR
.ad
.sp .6
.RS 4n
The home \fIlgroup\fR of the process or lwp.
.RE

.SH OPERANDS
.LP
The following operands are supported:
.sp
.ne 2
.na
\fB\fIcount\fR\fR
.ad
.sp .6
.RS 4n
Specifies the number of times that the statistics are repeated. By default,
\fBprstat\fR reports statistics until a termination signal is received.
.RE

.sp
.ne 2
```

```
.na
\fB\fIinterval\fR\fR
.ad
.sp .6
.RS 4n
Specifies the sampling interval in seconds; the default interval is \fB5\fR
seconds.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRReporting the Five Most Active Super-User Processes
.sp
.LP
The following command reports the five most active super-user processes running
on \fBCPU1\fR and \fBCPU2\fR:

.sp
.in +2
.nf
example% prstat -u root -n 5 -P 1,2 1 1

  PID   USERNAME  SWAP   RSS STATE  PRI  NICE      TIME  CPU PROCESS/LWP
  306    root     3024K 1448K sleep   58    0   0:00.00 0.3% sendmail/1
  102    root     1600K  592K sleep   59    0   0:00.00 0.1% in.rdisc/1
  250    root     1000K  552K sleep   58    0   0:00.00 0.0% utmpd/1
  288    root     1720K 1032K sleep   58    0   0:00.00 0.0% sac/1
    1    root      744K  168K sleep   58    0   0:00.00 0.0% init/1
TOTAL:       25, load averages:  0.05, 0.08, 0.12
.fi
.in -2
.sp

.LP
\fBExample 2 \fRDisplaying Verbose Process Usage Information
.sp
.LP
The following command displays verbose process usage information about
processes with lowest resident set sizes owned by users \fBroot\fR and
\fBjohn\fR.

.sp
.in +2
.nf
example% prstat -S rss -n 5 -vc -u root,john

  PID USERNAME USR SYS TRP TFL DFL LCK SLP LAT VCX ICX SCL SIG PROCESS/LWP
    1 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 init/1
  102 root     0.0 0.0  -   -   -   -  100  -    0   0   3   0 in.rdisc/1
  250 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 utmpd/1
 1185 john    0.0 0.0  -   -   -   -  100  -    0   0   0   0 csh/1
  240 root     0.0 0.0  -   -   -   -  100  -    0   0   0   0 powerd/4
  TOTAL:       71, load averages:  0.02, 0.04, 0.08

.fi
.in -2
.sp

.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
```



```
.sp .6
.RS 4n
Successful completion.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.sp .6
.RS 4n
An error occurred.
.RE

.SH SEE ALSO
.LP
\fBdate\fR(1), \fBlgrpinfo\fR(1), \fBplgrp\fR(1), \fBproc\fR(1), \fBps\fR(1),
\fBtime\fR(2), \fBpsrinfo\fR(1M), \fBpsrset\fR(1M), \fBsar\fR(1M),
\fBpset_getloadavg\fR(3C), \fBproc\fR(4), \fBproject\fR(4),
\fBattributes\fR(5), \fBresource_controls\fR(5), \fBzones\fR(5)
.SH NOTES
.LP
The snapshot of system usage displayed by \fBprstat\fR is true only for a
split-second, and it may not be accurate by the time it is displayed. When the
\fB-m\fR option is specified, \fBprstat\fR tries to turn on microstate
accounting for each process; the original state is restored when \fBprstat\fR
exits. See \fBproc\fR(4) for additional information about the microstate
accounting facility.
.sp
.LP
The total memory size reported in the SWAP and RSS columns for groups of
processes can sometimes overestimate the actual amount of memory used by
processes with shared memory segments.
```

```
**********************************************************
   70155 Mon Oct 15 13:25:33 2018
new/usr/src/man/man3c/Makefile
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2018 Nexenta Systems, Inc.
  15 # Copyright 2013, OmniTI Computer Consulting, Inc. All rights reserved.
  16 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  17 # Copyright 2018 Joyent, Inc.
  18 # Copyright 2018 Jason King
  19 #

  21 include         $(SRC)/Makefile.master

  23 MANSECT=        3c

  25 MANFILES=       __fbufsize.3c                                   \
  26                 _longjmp.3c                                     \
  27                 _stack_grow.3c                                  \
  28                 a64l.3c                                         \
  29                 abort.3c                                        \
  30                 abs.3c                                          \
  31                 addsev.3c                                       \
  32                 addseverity.3c                                  \
  33                 aio_cancel.3c                                   \
  34                 aio_error.3c                                    \
  35                 aio_fsync.3c                                    \
  36                 aio_read.3c                                     \
  37                 aio_return.3c                                   \
  38                 aio_suspend.3c                                  \
  39                 aio_waitn.3c                                    \
  40                 aio_write.3c                                    \
  41                 aiocancel.3c                                    \
  42                 aioread.3c                                      \
  43                 aiowait.3c                                      \
  44                 aligned_alloc.3c                                \
  45                 arc4random.3c                                   \
  46                 assert.3c                                       \
  47                 atexit.3c                                       \
  48                 atomic_add.3c                                   \
  49                 atomic_and.3c                                   \
  50                 atomic_bits.3c                                  \
  51                 atomic_cas.3c                                   \
  52                 atomic_dec.3c                                   \
  53                 atomic_inc.3c                                   \
  54                 atomic_ops.3c                                   \
  55                 atomic_or.3c                                    \
  56                 atomic_swap.3c                                  \
  57                 attropen.3c                                     \
  58                 basename.3c                                     \
  59                 bsd_signal.3c                                   \
  60                 bsearch.3c                                      \
```



```
  61                 bstring.3c                                      \
  62                 btowc.3c                                        \
  63                 byteorder.3c                                    \
  64                 call_once.3c                                    \
  65                 catgets.3c                                      \
  66                 catopen.3c                                      \
  67                 cfgetispeed.3c                                  \
  68                 cfsetispeed.3c                                  \
  69                 clearenv.3c                                     \
  70                 clock.3c                                        \
  71                 clock_nanosleep.3c                              \
  72                 clock_settime.3c                                \
  73                 closedir.3c                                     \
  74                 closefrom.3c                                    \
  75                 cnd.3c                                          \
  76                 cond_init.3c                                    \
  77                 confstr.3c                                      \
  78                 crypt.3c                                        \
  79                 crypt_genhash_impl.3c                           \
  80                 crypt_gensalt.3c                                \
  81                 crypt_gensalt_impl.3c                           \
  82                 cset.3c                                         \
  83                 ctermid.3c                                      \
  84                 ctime.3c                                        \
  85                 ctype.3c                                        \
  86                 cuserid.3c                                      \
  87                 daemon.3c                                       \
  88                 decimal_to_floating.3c                          \
  89                 difftime.3c                                     \
  90                 directio.3c                                     \
  91                 dirfd.3c                                        \
  92                 dirname.3c                                      \
  93                 div.3c                                          \
  94                 dladdr.3c                                       \
  95                 dlclose.3c                                      \
  96                 dldump.3c                                       \
  97                 dlerror.3c                                      \
  98                 dlinfo.3c                                       \
  99                 dlopen.3c                                       \
 100                 dlsym.3c                                        \
 101                 door_bind.3c                                    \
 102                 door_call.3c                                    \
 103                 door_create.3c                                  \
 104                 door_cred.3c                                    \
 105                 door_getparam.3c                                \
 106                 door_info.3c                                    \
 107                 door_return.3c                                  \
 108                 door_revoke.3c                                  \
 109                 door_server_create.3c                           \
 110                 door_ucred.3c                                   \
 111                 drand48.3c                                      \
 112                 dup2.3c                                         \
 113                 econvert.3c                                     \
 114                 ecvt.3c                                         \
 115                 enable_extended_FILE_stdio.3c                   \
 116                 encrypt.3c                                      \
 117                 end.3c                                          \
 118                 endian.3c                                       \
 119                 epoll_create.3c                                 \
 120                 epoll_ctl.3c                                    \
 121                 epoll_wait.3c                                   \
 122                 err.3c                                          \
 123                 euclen.3c                                       \
 124                 eventfd.3c                                      \
 125                 exit.3c                                         \
 126                 fattach.3c                                      \
```

```
127                     fclose.3c                                       \
128                     fcloseall.3c                                    \
129                     fdatasync.3c                                    \
130                     fdetach.3c                                      \
131                     fdopen.3c                                       \
132                     ferror.3c                                       \
133                     fflush.3c                                       \
134                     ffs.3c                                          \
135                     fgetattr.3c                                     \
136                     fgetc.3c                                        \
137                     fgetpos.3c                                      \
138                     fgetwc.3c                                       \
139                     floating_to_decimal.3c                          \
140                     flock.3c                                        \
141                     flockfile.3c                                    \
142                     fmtmsg.3c                                       \
143                     fnmatch.3c                                      \
144                     fopen.3c                                        \
145                     fpgetround.3c                                   \
146                     fputc.3c                                        \
147                     fputwc.3c                                       \
148                     fputws.3c                                       \
149                     fread.3c                                        \
150                     freopen.3c                                      \
151                     fseek.3c                                        \
152                     fsetpos.3c                                      \
153                     fsync.3c                                        \
154                     ftell.3c                                        \
155                     ftime.3c                                        \
156                     ftok.3c                                         \
157                     fts.3c                                          \
158                     ftw.3c                                          \
159                     fwide.3c                                        \
160                     fwprintf.3c                                     \
161                     fwrite.3c                                       \
162                     fwscanf.3c                                      \
163                     get_nprocs.3c                                   \
164                     getcpuid.3c                                     \
165                     getcwd.3c                                       \
166                     getdate.3c                                      \
167                     getdtablesize.3c                                \
168                     getentropy.3c                                   \
169                     getenv.3c                                       \
170                     getexecname.3c                                  \
171                     getgrnam.3c                                     \
172                     gethostid.3c                                    \
173                     gethostname.3c                                  \
174                     gethrtime.3c                                    \
175                     getline.3c                                      \
176                     getloadavg.3c                                   \
177                     getlogin.3c                                     \
178                     getmntent.3c                                    \
179                     getnetgrent.3c                                  \
180                     getopt.3c                                       \
181                     getopt_long.3c                                  \
182                     getpagesize.3c                                  \
183                     getpagesizes.3c                                 \
184                     getpass.3c                                      \
185                     getpeerucred.3c                                 \
186                     getpriority.3c                                  \
187                     getprogname.3c                                  \
188                     getpw.3c                                        \
189                     getpwnam.3c                                     \
190                     getrusage.3c                                    \
191                     gets.3c                                         \
192                     getspnam.3c                                     \
```



```
193                     getsubopt.3c                                    \
194                     gettext.3c                                      \
195                     gettimeofday.3c                                 \
196                     gettxt.3c                                       \
197                     getusershell.3c                                 \
198                     getutent.3c                                     \
199                     getutxent.3c                                    \
200                     getvfsent.3c                                    \
201                     getwc.3c                                        \
202                     getwchar.3c                                     \
203                     getwd.3c                                        \
204                     getwidth.3c                                     \
205                     getws.3c                                        \
206                     getzoneid.3c                                    \
207                     glob.3c                                         \
208                     grantpt.3c                                      \
209                     hsearch.3c                                      \
210                     iconv.3c                                        \
211                     iconv_close.3c                                  \
212                     iconv_open.3c                                   \
213                     imaxabs.3c                                      \
214                     imaxdiv.3c                                      \
215                     index.3c                                        \
216                     inet.3c                                         \
217                     initgroups.3c                                   \
218                     insque.3c                                       \
219                     is_system_labeled.3c                            \
220                     isaexec.3c                                      \
221                     isastream.3c                                    \
222                     isatty.3c                                       \
223                     isnand.3c                                       \
224                     iswalpha.3c                                     \
225                     iswctype.3c                                     \
226                     killpg.3c                                       \
227                     lckpwdf.3c                                      \
228                     lfmt.3c                                         \
229                     lio_listio.3c                                   \
230                     localeconv.3c                                   \
231                     lockf.3c                                        \
232                     lsearch.3c                                      \
233                     madvise.3c                                      \
234                     makecontext.3c                                  \
235                     makedev.3c                                      \
236                     malloc.3c                                       \
237                     mblen.3c                                        \
238                     mbrlen.3c                                       \
239                     mbrtowc.3c                                      \
240                     mbsinit.3c                                      \
241                     mbsrtowcs.3c                                    \
242                     mbstowcs.3c                                     \
243                     mbtowc.3c                                       \
244                     membar_ops.3c                                   \
245                     memory.3c                                       \
246                     memset_s.3c                                     \
247                     mkfifo.3c                                       \
248                     mkstemp.3c                                      \
249                     mktemp.3c                                       \
250                     mktime.3c                                       \
251                     mlock.3c                                        \
252                     mlockall.3c                                     \
253                     monitor.3c                                      \
254                     mq_close.3c                                     \
255                     mq_getattr.3c                                   \
256                     mq_notify.3c                                    \
257                     mq_open.3c                                      \
258                     mq_receive.3c                                   \
```

```
259                mq_send.3c                                   \
260                mq_setattr.3c                                \
261                mq_unlink.3c                                 \
262                msync.3c                                     \
263                mtx.3c                                       \
264                mutex_init.3c                                \
265                nanosleep.3c                                 \
266                ndbm.3c                                      \
267                newlocale.3c                                 \
268                nl_langinfo.3c                               \
269                offsetof.3c                                  \
270                opendir.3c                                   \
271                perror.3c                                    \
272                pfmt.3c                                      \
273                plock.3c                                     \
274                popen.3c                                     \
275                port_alert.3c                                \
276                port_associate.3c                            \
277                port_create.3c                               \
278                port_get.3c                                  \
279                port_send.3c                                 \
280                posix_fadvise.3c                             \
281                posix_fallocate.3c                           \
282                posix_madvise.3c                             \
283                posix_memalign.3c                            \
284                posix_openpt.3c                              \
285                posix_spawn.3c                               \
286                posix_spawn_file_actions_addclose.3c         \
287                posix_spawn_file_actions_addclosefrom_np.3c  \
288                posix_spawn_file_actions_adddup2.3c          \
289                posix_spawn_file_actions_destroy.3c          \
290                posix_spawn_pipe_np.3c                       \
291                posix_spawnattr_destroy.3c                   \
292                posix_spawnattr_getflags.3c                  \
293                posix_spawnattr_getpgroup.3c                 \
294                posix_spawnattr_getschedparam.3c             \
295                posix_spawnattr_getschedpolicy.3c            \
296                posix_spawnattr_getsigdefault.3c             \
297                posix_spawnattr_getsigignore_np.3c           \
298                posix_spawnattr_getsigmask.3c                \
299                printf.3c                                    \
300                priv_addset.3c                               \
301                priv_set.3c                                  \
302                priv_str_to_set.3c                           \
303                pset_getloadavg.3c                           \
304                psignal.3c                                   \
305                pthread_atfork.3c                            \
306                pthread_attr_get_np.3c                       \
307                pthread_attr_getdetachstate.3c               \
308                pthread_attr_getguardsize.3c                 \
309                pthread_attr_getinheritsched.3c              \
310                pthread_attr_getname_np.3c                   \
311                pthread_attr_getschedparam.3c                \
312                pthread_attr_getschedpolicy.3c               \
313                pthread_attr_getscope.3c                     \
314                pthread_attr_getstack.3c                     \
315                pthread_attr_getstackaddr.3c                 \
316                pthread_attr_getstacksize.3c                 \
317                pthread_attr_init.3c                         \
318                pthread_barrier_destroy.3c                   \
319                pthread_barrier_wait.3c                      \
320                pthread_barrierattr_destroy.3c               \
321                pthread_barrierattr_getpshared.3c            \
322                pthread_cancel.3c                            \
323                pthread_cleanup_pop.3c                       \
324                pthread_cleanup_push.3c                      \
```



```
325                pthread_cond_init.3c                         \
326                pthread_cond_signal.3c                       \
327                pthread_cond_wait.3c                         \
328                pthread_condattr_getclock.3c                 \
329                pthread_condattr_getpshared.3c               \
330                pthread_condattr_init.3c                     \
331                pthread_create.3c                            \
332                pthread_detach.3c                            \
333                pthread_equal.3c                             \
334                pthread_exit.3c                              \
335                pthread_getconcurrency.3c                    \
336                pthread_getname_np.3c                        \
337                pthread_getschedparam.3c                     \
338                pthread_getspecific.3c                       \
339                pthread_join.3c                              \
340                pthread_key_create.3c                        \
341                pthread_key_delete.3c                        \
342                pthread_kill.3c                              \
343                pthread_mutex_consistent.3c                  \
344                pthread_mutex_getprioceiling.3c              \
345                pthread_mutex_init.3c                        \
346                pthread_mutex_lock.3c                        \
347                pthread_mutex_timedlock.3c                   \
348                pthread_mutexattr_getprioceiling.3c          \
349                pthread_mutexattr_getprotocol.3c             \
350                pthread_mutexattr_getpshared.3c              \
351                pthread_mutexattr_getrobust.3c               \
352                pthread_mutexattr_gettype.3c                 \
353                pthread_mutexattr_init.3c                    \
354                pthread_once.3c                              \
355                pthread_rwlock_init.3c                       \
356                pthread_rwlock_rdlock.3c                     \
357                pthread_rwlock_timedrdlock.3c                \
358                pthread_rwlock_timedwrlock.3c                \
359                pthread_rwlock_unlock.3c                     \
360                pthread_rwlock_wrlock.3c                     \
361                pthread_rwlockattr_getpshared.3c             \
362                pthread_rwlockattr_init.3c                   \
363                pthread_self.3c                              \
364                pthread_setcancelstate.3c                    \
365                pthread_setcanceltype.3c                     \
366                pthread_setschedprio.3c                      \
367                pthread_sigmask.3c                           \
368                pthread_spin_destroy.3c                      \
369                pthread_spin_lock.3c                         \
370                pthread_spin_unlock.3c                       \
371                pthread_testcancel.3c                        \
372                ptrace.3c                                    \
373                ptsname.3c                                   \
374                putenv.3c                                    \
375                putpwent.3c                                  \
376                puts.3c                                      \
377                putspent.3c                                  \
378                putws.3c                                     \
379                qsort.3c                                     \
380                quick_exit.3c                                \
381                raise.3c                                     \
382                rand.3c                                      \
383                random.3c                                    \
384                rctl_walk.3c                                 \
385                rctlblk_set_value.3c                         \
386                re_comp.3c                                   \
387                readdir.3c                                   \
388                realpath.3c                                  \
389                reboot.3c                                    \
390                regcmp.3c                                    \
```

```
391                 regcomp.3c                                  \
392                 remove.3c                                   \
393                 rewind.3c                                   \
394                 rewinddir.3c                                \
395                 rwlock.3c                                   \
396                 scandir.3c                                  \
397                 scanf.3c                                    \
398                 sched_get_priority_max.3c                   \
399                 sched_getparam.3c                           \
400                 sched_getscheduler.3c                       \
401                 sched_rr_get_interval.3c                    \
402                 sched_setparam.3c                           \
403                 sched_setscheduler.3c                       \
404                 sched_yield.3c                              \
405                 schedctl_init.3c                            \
406                 seekdir.3c                                  \
407                 select.3c                                   \
408                 sem_close.3c                                \
409                 sem_destroy.3c                              \
410                 sem_getvalue.3c                             \
411                 sem_init.3c                                 \
412                 sem_open.3c                                 \
413                 sem_post.3c                                 \
414                 sem_timedwait.3c                            \
415                 sem_unlink.3c                               \
416                 sem_wait.3c                                 \
417                 semaphore.3c                                \
418                 set_constraint_handler_s.3c                 \
419                 setbuf.3c                                   \
420                 setbuffer.3c                                \
421                 setcat.3c                                   \
422                 setenv.3c                                   \
423                 setjmp.3c                                   \
424                 setkey.3c                                   \
425                 setlabel.3c                                 \
426                 setlocale.3c                                \
427                 shm_open.3c                                 \
428                 shm_unlink.3c                               \
429                 sigfpe.3c                                   \
430                 siginterrupt.3c                             \
431                 signal.3c                                   \
432                 signalfd.3c                                 \
433                 sigqueue.3c                                 \
434                 sigsetops.3c                                \
435                 sigstack.3c                                 \
436                 sigwaitinfo.3c                              \
437                 sleep.3c                                    \
438                 smt_pause.3c                                \
439                 ssignal.3c                                  \
440                 stack_getbounds.3c                          \
441                 stack_inbounds.3c                           \
442                 stack_setbounds.3c                          \
443                 stack_violation.3c                          \
444                 stdio.3c                                    \
445                 str2sig.3c                                  \
446                 strcoll.3c                                  \
447                 strerror.3c                                 \
448                 strfmon.3c                                  \
449                 strftime.3c                                 \
450                 string.3c                                   \
451                 string_to_decimal.3c                        \
452                 strptime.3c                                 \
453                 strsignal.3c                                \
454                 strtod.3c                                   \
455                 strtoimax.3c                                \
456                 strtol.3c                                   \
```



```
457                 strtonum.3c                                 \
458                 strtoul.3c                                  \
459                 strtows.3c                                  \
460                 strxfrm.3c                                  \
461                 swab.3c                                     \
462                 sync_instruction_memory.3c                  \
463                 sysconf.3c                                  \
464                 syslog.3c                                   \
465                 system.3c                                   \
466                 tcdrain.3c                                  \
467                 tcflow.3c                                   \
468                 tcflush.3c                                  \
469                 tcgetattr.3c                                \
470                 tcgetpgrp.3c                                \
471                 tcgetsid.3c                                 \
472                 tcsendbreak.3c                              \
473                 tcsetattr.3c                                \
474                 tcsetpgrp.3c                                \
475                 tell.3c                                     \
476                 telldir.3c                                  \
477                 termios.3c                                  \
478                 thr_create.3c                               \
479                 thr_exit.3c                                 \
480                 thr_getname.3c                              \
481                 thr_getconcurrency.3c                       \
482                 thr_getprio.3c                              \
483                 thr_join.3c                                 \
484                 thr_keycreate.3c                            \
485                 thr_kill.3c                                 \
486                 thr_main.3c                                 \
487                 thr_min_stack.3c                            \
488                 thr_self.3c                                 \
489                 thr_sigsetmask.3c                           \
490                 thr_stksegment.3c                           \
491                 thr_suspend.3c                              \
492                 thr_yield.3c                                \
493                 thrd_create.3c                              \
494                 thrd_current.3c                             \
495                 thrd_detach.3c                              \
496                 thrd_equal.3c                               \
497                 thrd_exit.3c                                \
498                 thrd_join.3c                                \
499                 thrd_yield.3c                               \
500                 timer_create.3c                             \
501                 timer_delete.3c                             \
502                 timer_settime.3c                            \
503                 timeradd.3c                                 \
504                 timerfd_create.3c                           \
505                 timespec_get.3c                             \
506                 tmpfile.3c                                  \
507                 tmpnam.3c                                   \
508                 toascii.3c                                  \
509                 tolower.3c                                  \
510                 toupper.3c                                  \
511                 towlower.3c                                 \
512                 towupper.3c                                 \
513                 truncate.3c                                 \
514                 tsearch.3c                                  \
515                 tss.3c                                      \
516                 ttyname.3c                                  \
517                 ttyslot.3c                                  \
518                 u8_strcmp.3c                                \
519                 u8_textprep_str.3c                          \
520                 u8_validate.3c                              \
521                 ualarm.3c                                   \
522                 uconv_u16tou32.3c                           \
```

```
523                 ucred.3c                                    \
524                 ungetc.3c                                   \
525                 ungetwc.3c                                  \
526                 unlockpt.3c                                 \
527                 unsetenv.3c                                 \
528                 uselocale.3c                                \
529                 usleep.3c                                   \
530                 vfwprintf.3c                                \
531                 vlfmt.3c                                    \
532                 vpfmt.3c                                    \
533                 vprintf.3c                                  \
534                 vsyslog.3c                                  \
535                 wait.3c                                     \
536                 wait3.3c                                    \
537                 waitpid.3c                                  \
538                 walkcontext.3c                              \
539                 wcpcpy.3c                                   \
540                 wcrtomb.3c                                  \
541                 wcscasecmp.3c                               \
542                 wcscoll.3c                                  \
543                 wcsdup.3c                                   \
544                 wcsftime.3c                                 \
545                 wcslen.3c                                   \
546                 wcsrtombs.3c                                \
547                 wcsstr.3c                                   \
548                 wcstod.3c                                   \
549                 wcstoimax.3c                                \
550                 wcstol.3c                                   \
551                 wcstombs.3c                                 \
552                 wcstoul.3c                                  \
553                 wcstring.3c                                 \
554                 wcswidth.3c                                 \
555                 wcsxfrm.3c                                  \
556                 wctob.3c                                    \
557                 wctomb.3c                                   \
558                 wctrans.3c                                  \
559                 wctype.3c                                   \
560                 wcwidth.3c                                  \
561                 wmemchr.3c                                  \
562                 wmemcmp.3c                                  \
563                 wmemcpy.3c                                  \
564                 wmemmove.3c                                 \
565                 wmemset.3c                                  \
566                 wordexp.3c                                  \
567                 wsprintf.3c                                 \
568                 wsscanf.3c                                  \
569                 wstring.3c

571 MANLINKS=       FD_CLR.3c                           \
572                 FD_ISSET.3c                         \
573                 FD_SET.3c                           \
574                 FD_ZERO.3c                          \
575                 __flbf.3c                           \
576                 __fpending.3c                       \
577                 __fpurge.3c                         \
578                 __freadable.3c                      \
579                 __freading.3c                       \
580                 __fsetlocking.3c                    \
581                 __fwritable.3c                      \
582                 __fwriting.3c                       \
583                 _edata.3c                           \
584                 _end.3c                             \
585                 _etext.3c                           \
586                 _exithandle.3c                      \
587                 _flushlbf.3c                        \
588                 _setjmp.3c                          \
```



```
589                 abort_handler_s.3c                  \
590                 addrtosymstr.3c                     \
591                 aiowrite.3c                         \
592                 alloca.3c                           \
593                 alphasort.3c                        \
594                 arc4random_buf.3c                   \
595                 arc4random_uniform.3c               \
596                 ascftime.3c                         \
597                 asctime.3c                          \
598                 asctime_r.3c                        \
599                 asprintf.3c                         \
600                 at_quick_exit.3c                    \
601                 atof.3c                             \
602                 atoi.3c                             \
603                 atol.3c                             \
604                 atoll.3c                            \
605                 atomic_add_16.3c                    \
606                 atomic_add_16_nv.3c                 \
607                 atomic_add_32.3c                    \
608                 atomic_add_32_nv.3c                 \
609                 atomic_add_64.3c                    \
610                 atomic_add_64_nv.3c                 \
611                 atomic_add_8.3c                     \
612                 atomic_add_8_nv.3c                  \
613                 atomic_add_char.3c                  \
614                 atomic_add_char_nv.3c               \
615                 atomic_add_int.3c                   \
616                 atomic_add_int_nv.3c                \
617                 atomic_add_long.3c                  \
618                 atomic_add_long_nv.3c               \
619                 atomic_add_ptr.3c                   \
620                 atomic_add_ptr_nv.3c                \
621                 atomic_add_short.3c                 \
622                 atomic_add_short_nv.3c              \
623                 atomic_and_16.3c                    \
624                 atomic_and_16_nv.3c                 \
625                 atomic_and_32.3c                    \
626                 atomic_and_32_nv.3c                 \
627                 atomic_and_64.3c                    \
628                 atomic_and_64_nv.3c                 \
629                 atomic_and_8.3c                     \
630                 atomic_and_8_nv.3c                  \
631                 atomic_and_uchar.3c                 \
632                 atomic_and_uchar_nv.3c              \
633                 atomic_and_uint.3c                  \
634                 atomic_and_uint_nv.3c               \
635                 atomic_and_ulong.3c                 \
636                 atomic_and_ulong_nv.3c              \
637                 atomic_and_ushort.3c                \
638                 atomic_and_ushort_nv.3c             \
639                 atomic_cas_16.3c                    \
640                 atomic_cas_32.3c                    \
641                 atomic_cas_64.3c                    \
642                 atomic_cas_8.3c                     \
643                 atomic_cas_ptr.3c                   \
644                 atomic_cas_uchar.3c                 \
645                 atomic_cas_uint.3c                  \
646                 atomic_cas_ulong.3c                 \
647                 atomic_cas_ushort.3c                \
648                 atomic_clear_long_excl.3c           \
649                 atomic_dec_16.3c                    \
650                 atomic_dec_16_nv.3c                 \
651                 atomic_dec_32.3c                    \
652                 atomic_dec_32_nv.3c                 \
653                 atomic_dec_64.3c                    \
654                 atomic_dec_64_nv.3c                 \
```

```
655                     atomic_dec_8.3c                 \
656                     atomic_dec_8_nv.3c              \
657                     atomic_dec_ptr.3c               \
658                     atomic_dec_ptr_nv.3c            \
659                     atomic_dec_uchar.3c             \
660                     atomic_dec_uchar_nv.3c          \
661                     atomic_dec_uint.3c              \
662                     atomic_dec_uint_nv.3c           \
663                     atomic_dec_ulong.3c             \
664                     atomic_dec_ulong_nv.3c          \
665                     atomic_dec_ushort.3c            \
666                     atomic_dec_ushort_nv.3c         \
667                     atomic_inc_16.3c                \
668                     atomic_inc_16_nv.3c             \
669                     atomic_inc_32.3c                \
670                     atomic_inc_32_nv.3c             \
671                     atomic_inc_64.3c                \
672                     atomic_inc_64_nv.3c             \
673                     atomic_inc_8.3c                 \
674                     atomic_inc_8_nv.3c              \
675                     atomic_inc_ptr.3c               \
676                     atomic_inc_ptr_nv.3c            \
677                     atomic_inc_uchar.3c             \
678                     atomic_inc_uchar_nv.3c          \
679                     atomic_inc_uint.3c              \
680                     atomic_inc_uint_nv.3c           \
681                     atomic_inc_ulong.3c             \
682                     atomic_inc_ulong_nv.3c          \
683                     atomic_inc_ushort.3c            \
684                     atomic_inc_ushort_nv.3c         \
685                     atomic_or_16.3c                 \
686                     atomic_or_16_nv.3c              \
687                     atomic_or_32.3c                 \
688                     atomic_or_32_nv.3c              \
689                     atomic_or_64.3c                 \
690                     atomic_or_64_nv.3c              \
691                     atomic_or_8.3c                  \
692                     atomic_or_8_nv.3c               \
693                     atomic_or_uchar.3c              \
694                     atomic_or_uchar_nv.3c           \
695                     atomic_or_uint.3c               \
696                     atomic_or_uint_nv.3c            \
697                     atomic_or_ulong.3c              \
698                     atomic_or_ulong_nv.3c           \
699                     atomic_or_ushort.3c             \
700                     atomic_or_ushort_nv.3c          \
701                     atomic_set_long_excl.3c         \
702                     atomic_swap_16.3c               \
703                     atomic_swap_32.3c               \
704                     atomic_swap_64.3c               \
705                     atomic_swap_8.3c                \
706                     atomic_swap_ptr.3c              \
707                     atomic_swap_uchar.3c            \
708                     atomic_swap_uint.3c             \
709                     atomic_swap_ulong.3c            \
710                     atomic_swap_ushort.3c           \
711                     backtrace.3c                    \
712                     backtrace_symbols.3c            \
713                     backtrace_symbols_fd.3c         \
714                     bcmp.3c                         \
715                     bcopy.3c                        \
716                     be16toh.3c                      \
717                     be32toh.3c                      \
718                     be64toh.3c                      \
719                     betoh16.3c                      \
720                     betoh32.3c                      \
```



```
721                     betoh64.3c                      \
722                     bind_textdomain_codeset.3c      \
723                     bindtextdomain.3c               \
724                     btowc_l.3c                      \
725                     bzero.3c                        \
726                     calloc.3c                       \
727                     canonicalize_file_name.3c       \
728                     catclose.3c                     \
729                     cfgetospeed.3c                  \
730                     cfsetospeed.3c                  \
731                     cftime.3c                       \
732                     clearerr.3c                     \
733                     clock_getres.3c                 \
734                     clock_gettime.3c                \
735                     closelog.3c                     \
736                     cnd_broadcast.3c                \
737                     cnd_destroy.3c                  \
738                     cnd_init.3c                     \
739                     cnd_signal.3c                   \
740                     cnd_timedwait.3c                \
741                     cnd_wait.3c                     \
742                     cond_broadcast.3c               \
743                     cond_destroy.3c                 \
744                     cond_reltimedwait.3c            \
745                     cond_signal.3c                  \
746                     cond_timedwait.3c               \
747                     cond_wait.3c                    \
748                     csetcol.3c                      \
749                     csetlen.3c                      \
750                     csetno.3c                       \
751                     ctermid_r.3c                    \
752                     ctime_r.3c                      \
753                     dbm_clearerr.3c                 \
754                     dbm_close.3c                    \
755                     dbm_delete.3c                   \
756                     dbm_error.3c                    \
757                     dbm_fetch.3c                    \
758                     dbm_firstkey.3c                 \
759                     dbm_nextkey.3c                  \
760                     dbm_open.3c                     \
761                     dbm_store.3c                    \
762                     dcgettext.3c                    \
763                     dcngettext.3c                   \
764                     decimal_to_double.3c            \
765                     decimal_to_extended.3c          \
766                     decimal_to_quadruple.3c         \
767                     decimal_to_single.3c            \
768                     dgettext.3c                     \
769                     dladdr1.3c                      \
770                     dlmopen.3c                      \
771                     dngettext.3c                    \
772                     door_setparam.3c                \
773                     door_unbind.3c                  \
774                     double_to_decimal.3c            \
775                     dup3.3c                         \
776                     duplocale.3c                    \
777                     edata.3c                        \
778                     endgrent.3c                     \
779                     endnetgrent.3c                  \
780                     endpwent.3c                     \
781                     endspent.3c                     \
782                     endusershell.3c                 \
783                     endutent.3c                     \
784                     endutxent.3c                    \
785                     epoll_create1.3c                \
786                     epoll_pwait.3c                  \
```

```
787                 erand48.3c                          \
788                 errno.3c                            \
789                 errx.3c                             \
790                 etext.3c                            \
791                 euccol.3c                           \
792                 eucscol.3c                          \
793                 explicit_bzero.3c                   \
794                 extended_to_decimal.3c              \
795                 fconvert.3c                         \
796                 fcvt.3c                             \
797                 fdopendir.3c                        \
798                 fdwalk.3c                           \
799                 feof.3c                             \
800                 ffsl.3c                             \
801                 ffsll.3c                            \
802                 fgetgrent.3c                        \
803                 fgetgrent_r.3c                      \
804                 fgetpwent.3c                        \
805                 fgetpwent_r.3c                      \
806                 fgets.3c                            \
807                 fgetspent.3c                        \
808                 fgetspent_r.3c                      \
809                 fgetwc_l.3c                         \
810                 fgetws.3c                           \
811                 file_to_decimal.3c                  \
812                 fileno.3c                           \
813                 finite.3c                           \
814                 fls.3c                              \
815                 flsl.3c                             \
816                 flsll.3c                            \
817                 fpclass.3c                          \
818                 fpgetmask.3c                        \
819                 fpgetsticky.3c                      \
820                 fprintf.3c                          \
821                 fpsetmask.3c                        \
822                 fpsetround.3c                       \
823                 fpsetsticky.3c                      \
824                 fputs.3c                            \
825                 free.3c                             \
826                 freelocale.3c                       \
827                 freezero.3c                         \
828                 fscanf.3c                           \
829                 fseeko.3c                           \
830                 fsetattr.3c                         \
831                 ftello.3c                           \
832                 ftruncate.3c                        \
833                 ftrylockfile.3c                     \
834                 fts_children.3c                     \
835                 fts_close.3c                        \
836                 fts_open.3c                         \
837                 fts_read.3c                         \
838                 fts_set.3c                          \
839                 func_to_decimal.3c                  \
840                 funlockfile.3c                      \
841                 gconvert.3c                         \
842                 gcvt.3c                             \
843                 get_nprocs_conf.3c                  \
844                 getattrat.3c                        \
845                 getc.3c                             \
846                 getc_unlocked.3c                    \
847                 getchar.3c                          \
848                 getchar_unlocked.3c                 \
849                 getdelim.3c                         \
850                 getextmntent.3c                     \
851                 getgrent.3c                         \
852                 getgrent_r.3c                       \
```



```
853                 getgrgid.3c                         \
854                 getgrgid_r.3c                       \
855                 getgrnam_r.3c                       \
856                 gethomelgroup.3c                    \
857                 gethrvtime.3c                       \
858                 getlogin_r.3c                       \
859                 getmntany.3c                        \
860                 getnetgrent_r.3c                    \
861                 getopt_long_only.3c                 \
862                 getopt_long_clip.3c                 \
863                 getpassphrase.3c                    \
864                 getpwent.3c                         \
865                 getpwent_r.3c                       \
866                 getpwnam_r.3c                       \
867                 getpwuid.3c                         \
868                 getpwuid_r.3c                       \
869                 getspent.3c                         \
870                 getspent_r.3c                       \
871                 getspnam_r.3c                       \
872                 getutid.3c                          \
873                 getutline.3c                        \
874                 getutmp.3c                          \
875                 getutmpx.3c                         \
876                 getutxid.3c                         \
877                 getutxline.3c                       \
878                 getvfsany.3c                        \
879                 getvfsfile.3c                       \
880                 getvfsspec.3c                       \
881                 getw.3c                             \
882                 getwc_l.3c                          \
883                 getwchar_l.3c                       \
884                 getzoneidbyname.3c                  \
885                 getzonenamebyid.3c                  \
886                 globfree.3c                         \
887                 gmtime.3c                           \
888                 gmtime_r.3c                         \
889                 gsignal.3c                          \
890                 hasmntopt.3c                        \
891                 hcreate.3c                          \
892                 hdestroy.3c                         \
893                 htobe16.3c                          \
894                 htobe32.3c                          \
895                 htobe64.3c                          \
896                 htole16.3c                          \
897                 htole32.3c                          \
898                 htole64.3c                          \
899                 htonl.3c                            \
900                 htonll.3c                           \
901                 htons.3c                            \
902                 ignore_handler_s.3c                 \
903                 inet6.3c                            \
904                 inet_addr.3c                        \
905                 inet_aton.3c                        \
906                 inet_lnaof.3c                       \
907                 inet_makeaddr.3c                    \
908                 inet_netof.3c                       \
909                 inet_network.3c                     \
910                 inet_ntoa.3c                        \
911                 inet_ntop.3c                        \
912                 inet_pton.3c                        \
913                 initstate.3c                        \
914                 innetgr.3c                          \
915                 isalnum.3c                          \
916                 isalnum_l.3c                        \
917                 isalpha.3c                          \
918                 isalpha_l.3c                        \
```

```
                isascii.3c                      \
                isblank.3c                      \
                isblank_l.3c                    \
                iscntrl.3c                      \
                iscntrl_l.3c                    \
                isdigit.3c                      \
                isdigit_l.3c                    \
                isenglish.3c                    \
                isgraph.3c                      \
                isgraph_l.3c                    \
                isideogram.3c                   \
                islower.3c                      \
                islower_l.3c                    \
                isnanf.3c                       \
                isnumber.3c                     \
                isphonogram.3c                  \
                isprint.3c                      \
                isprint_l.3c                    \
                ispunct.3c                      \
                ispunct_l.3c                    \
                isspace.3c                      \
                isspace_l.3c                    \
                isspecial.3c                    \
                isupper.3c                      \
                isupper_l.3c                    \
                iswalnum.3c                     \
                iswalnum_l.3c                   \
                iswalpha_l.3c                   \
                iswascii.3c                     \
                iswblank.3c                     \
                iswblank_l.3c                   \
                iswcntrl.3c                     \
                iswcntrl_l.3c                   \
                iswctype_l.3c                   \
                iswdigit.3c                     \
                iswdigit_l.3c                   \
                iswgraph.3c                     \
                iswgraph_l.3c                   \
                iswhexnumber.3c                 \
                iswhexnumber_l.3c               \
                iswideogram.3c                  \
                iswideogram_l.3c                \
                iswlower.3c                     \
                iswlower_l.3c                   \
                iswnumber.3c                    \
                iswnumber_l.3c                  \
                iswphonogram.3c                 \
                iswphonogram_l.3c               \
                iswprint.3c                     \
                iswprint_l.3c                   \
                iswpunct.3c                     \
                iswpunct_l.3c                   \
                iswspace.3c                     \
                iswspace_l.3c                   \
                iswspecial.3c                   \
                iswspecial_l.3c                 \
                iswupper.3c                     \
                iswupper_l.3c                   \
                iswxdigit.3c                    \
                iswxdigit_l.3c                  \
                isxdigit.3c                     \
                isxdigit_l.3c                   \
                jrand48.3c                      \
                l64a.3c                         \
                labs.3c                         \
                lcong48.3c                      \
```



```
                ldiv.3c                         \
                le16toh.3c                      \
                le32toh.3c                      \
                le64toh.3c                      \
                letoh16.3c                      \
                letoh32.3c                      \
                letoh64.3c                      \
                lfind.3c                        \
                llabs.3c                        \
                lldiv.3c                        \
                lltostr.3c                      \
                localtime.3c                    \
                localtime_r.3c                  \
                longjmp.3c                      \
                lrand48.3c                      \
                major.3c                        \
                mblen_l.3c                      \
                mbrlen_l.3c                     \
                mbrtowc_l.3c                    \
                mbsinit_l.3c                    \
                mbsnrtowcs.3c                   \
                mbsnrtowcs_l.3c                 \
                mbsrtowcs_l.3c                  \
                mbstowcs_l.3c                   \
                mbtowc_l.3c                     \
                memalign.3c                     \
                membar_consumer.3c              \
                membar_enter.3c                 \
                membar_exit.3c                  \
                membar_producer.3c              \
                memccpy.3c                      \
                memchr.3c                       \
                memcmp.3c                       \
                memcpy.3c                       \
                memmem.3c                       \
                memmove.3c                      \
                memset.3c                       \
                minor.3c                        \
                mkdtemp.3c                      \
                mkfifoat.3c                     \
                mkostemp.3c                     \
                mkostemps.3c                    \
                mkstemps.3c                     \
                mq_reltimedreceive_np.3c        \
                mq_reltimedsend_np.3c           \
                mq_timedreceive.3c              \
                mq_timedsend.3c                 \
                mrand48.3c                      \
                mtx_destroy.3c                  \
                mtx_init.3c                     \
                mtx_lock.3c                     \
                mtx_timedlock.3c                \
                mtx_trylock.3c                  \
                mtx_unlock.3c                   \
                munlock.3c                      \
                munlockall.3c                   \
                mutex_consistent.3c             \
                mutex_destroy.3c                \
                mutex_lock.3c                   \
                mutex_trylock.3c                \
                mutex_unlock.3c                 \
                nftw.3c                         \
                ngettext.3c                     \
                nl_langinfo_l.3c                \
                nrand48.3c                      \
                ntohl.3c                        \
```

```
1051                ntohll.3c                           \
1052                ntohs.3c                            \
1053                openlog.3c                          \
1054                pclose.3c                           \
1055                port_dissociate.3c                  \
1056                port_getn.3c                        \
1057                port_sendn.3c                       \
1058                posix_spawn_file_actions_addopen.3c  \
1059                posix_spawn_file_actions_init.3c    \
1060                posix_spawnattr_init.3c             \
1061                posix_spawnattr_setflags.3c         \
1062                posix_spawnattr_setpgroup.3c        \
1063                posix_spawnattr_setschedparam.3c    \
1064                posix_spawnattr_setschedpolicy.3c   \
1065                posix_spawnattr_setsigdefault.3c    \
1066                posix_spawnattr_setsigignore_np.3c  \
1067                posix_spawnattr_setsigmask.3c       \
1068                posix_spawnp.3c                     \
1069                printstack.3c                       \
1070                priv_allocset.3c                    \
1071                priv_basicset.3c                    \
1072                priv_copyset.3c                     \
1073                priv_delset.3c                      \
1074                priv_emptyset.3c                    \
1075                priv_fillset.3c                     \
1076                priv_freeset.3c                     \
1077                priv_getbyname.3c                   \
1078                priv_getbynum.3c                    \
1079                priv_getsetbyname.3c                \
1080                priv_getsetbynum.3c                 \
1081                priv_gettext.3c                     \
1082                priv_ineffect.3c                    \
1083                priv_intersect.3c                   \
1084                priv_inverse.3c                     \
1085                priv_isemptyset.3c                  \
1086                priv_isequalset.3c                  \
1087                priv_isfullset.3c                   \
1088                priv_ismember.3c                    \
1089                priv_issubset.3c                    \
1090                priv_set_to_str.3c                  \
1091                priv_union.3c                       \
1092                pselect.3c                          \
1093                psiginfo.3c                         \
1094                pthread_attr_destroy.3c             \
1095                pthread_attr_setdetachstate.3c      \
1096                pthread_attr_setguardsize.3c        \
1097                pthread_attr_setinheritsched.3c     \
1098                pthread_attr_setname_np.3c          \
1099                pthread_attr_setschedparam.3c       \
1100                pthread_attr_setschedpolicy.3c      \
1101                pthread_attr_setscope.3c            \
1102                pthread_attr_setstack.3c            \
1103                pthread_attr_setstackaddr.3c        \
1104                pthread_attr_setstacksize.3c        \
1105                pthread_barrier_init.3c             \
1106                pthread_barrierattr_init.3c         \
1107                pthread_barrierattr_setpshared.3c   \
1108                pthread_cond_broadcast.3c           \
1109                pthread_cond_destroy.3c             \
1110                pthread_cond_reltimedwait_np.3c     \
1111                pthread_cond_timedwait.3c           \
1112                pthread_condattr_destroy.3c         \
1113                pthread_condattr_setclock.3c        \
1114                pthread_condattr_setpshared.3c      \
1115                pthread_key_create_once_np.3c       \
1116                pthread_mutex_destroy.3c            \
```



```
1117                pthread_mutex_reltimedlock_np.3c    \
1118                pthread_mutex_setprioceiling.3c     \
1119                pthread_mutex_trylock.3c            \
1120                pthread_mutex_unlock.3c             \
1121                pthread_mutexattr_destroy.3c        \
1122                pthread_mutexattr_setprioceiling.3c  \
1123                pthread_mutexattr_setprotocol.3c    \
1124                pthread_mutexattr_setpshared.3c     \
1125                pthread_mutexattr_setrobust.3c      \
1126                pthread_mutexattr_settype.3c        \
1127                pthread_rwlock_destroy.3c           \
1128                pthread_rwlock_reltimedrdlock_np.3c  \
1129                pthread_rwlock_reltimedwrlock_np.3c  \
1130                pthread_rwlock_tryrdlock.3c         \
1131                pthread_rwlock_trywrlock.3c         \
1132                pthread_rwlockattr_destroy.3c       \
1133                pthread_rwlockattr_setpshared.3c    \
1134                pthread_setconcurrency.3c           \
1135                pthread_setname_np.3c               \
1136                pthread_setschedparam.3c            \
1137                pthread_setspecific.3c              \
1138                pthread_spin_init.3c                \
1139                pthread_spin_trylock.3c             \
1140                putc.3c                             \
1141                putc_unlocked.3c                    \
1142                putchar.3c                          \
1143                putchar_unlocked.3c                 \
1144                putmntent.3c                        \
1145                pututline.3c                        \
1146                pututxline.3c                       \
1147                putw.3c                             \
1148                putwc.3c                            \
1149                putwchar.3c                         \
1150                qeconvert.3c                        \
1151                qfconvert.3c                        \
1152                qgconvert.3c                        \
1153                quadruple_to_decimal.3c             \
1154                rand_r.3c                           \
1155                rctlblk_get_enforced_value.3c       \
1156                rctlblk_get_firing_time.3c          \
1157                rctlblk_get_global_action.3c        \
1158                rctlblk_get_global_flags.3c         \
1159                rctlblk_get_local_action.3c         \
1160                rctlblk_get_local_flags.3c          \
1161                rctlblk_get_privilege.3c            \
1162                rctlblk_get_recipient_pid.3c        \
1163                rctlblk_get_value.3c                \
1164                rctlblk_set_local_action.3c         \
1165                rctlblk_set_local_flags.3c          \
1166                rctlblk_set_privilege.3c            \
1167                rctlblk_set_recipient_pid.3c        \
1168                rctlblk_size.3c                     \
1169                re_exec.3c                          \
1170                readdir_r.3c                        \
1171                realloc.3c                          \
1172                reallocarray.3c                     \
1173                recallocarray.3c                    \
1174                regerror.3c                         \
1175                regex.3c                            \
1176                regexec.3c                          \
1177                regfree.3c                          \
1178                remque.3c                           \
1179                resetmnttab.3c                      \
1180                rindex.3c                           \
1181                rw_rdlock.3c                        \
1182                rw_tryrdlock.3c                     \
```

```
1183                rw_trywrlock.3c                 \
1184                rw_unlock.3c                    \
1185                rw_wrlock.3c                    \
1186                rwlock_destroy.3c               \
1187                rwlock_init.3c                  \
1188                sched_get_priority_min.3c       \
1189                schedctl_exit.3c                \
1190                schedctl_lookup.3c              \
1191                schedctl_start.3c               \
1192                schedctl_stop.3c                \
1193                seconvert.3c                    \
1194                seed48.3c                       \
1195                sem_reltimedwait_np.3c          \
1196                sem_trywait.3c                  \
1197                sema_destroy.3c                 \
1198                sema_init.3c                    \
1199                sema_post.3c                    \
1200                sema_trywait.3c                 \
1201                sema_wait.3c                    \
1202                setattrat.3c                    \
1203                setgrent.3c                     \
1204                sethostname.3c                  \
1205                setlinebuf.3c                   \
1206                setlogmask.3c                   \
1207                setnetgrent.3c                  \
1208                setpriority.3c                  \
1209                setprogname.3c                  \
1210                setpwent.3c                     \
1211                setspent.3c                     \
1212                setstate.3c                     \
1213                settimeofday.3c                 \
1214                setusershell.3c                 \
1215                setutent.3c                     \
1216                setutxent.3c                    \
1217                setvbuf.3c                      \
1218                sfconvert.3c                    \
1219                sgconvert.3c                    \
1220                sig2str.3c                      \
1221                sigaddset.3c                    \
1222                sigdelset.3c                    \
1223                sigemptyset.3c                  \
1224                sigfillset.3c                   \
1225                sighold.3c                      \
1226                sigignore.3c                    \
1227                sigismember.3c                  \
1228                siglongjmp.3c                   \
1229                sigpause.3c                     \
1230                sigrelse.3c                     \
1231                sigset.3c                       \
1232                sigsetjmp.3c                    \
1233                sigtimedwait.3c                 \
1234                single_to_decimal.3c            \
1235                snprintf.3c                     \
1236                sprintf.3c                      \
1237                srand.3c                        \
1238                srand48.3c                      \
1239                srandom.3c                      \
1240                sscanf.3c                       \
1241                stderr.3c                       \
1242                stdin.3c                        \
1243                stdout.3c                       \
1244                stpcpy.3c                       \
1245                stpncpy.3c                      \
1246                strcasecmp.3c                   \
1247                strcasecmp_l.3c                 \
1248                strcasestr.3c                   \
```



```
1249                strcasestr_l.3c                 \
1250                strcat.3c                       \
1251                strchr.3c                       \
1252                strchrnul.3c                    \
1253                strcmp.3c                       \
1254                strcoll_l.3c                    \
1255                strcpy.3c                       \
1256                strcspn.3c                      \
1257                strdup.3c                       \
1258                strdupa.3c                      \
1259                strerror_l.3c                   \
1260                strerror_r.3c                   \
1261                strfmon_l.3c                    \
1262                strftime_l.3c                   \
1263                strlcat.3c                      \
1264                strlcpy.3c                      \
1265                strlen.3c                       \
1266                strncasecmp.3c                  \
1267                strncasecmp_l.3c                \
1268                strncat.3c                      \
1269                strncmp.3c                      \
1270                strncpy.3c                      \
1271                strndup.3c                      \
1272                strndupa.3c                     \
1273                strnlen.3c                      \
1274                strnstr.3c                      \
1275                strpbrk.3c                      \
1276                strptime_l.3c                   \
1277                strrchr.3c                      \
1278                strsep.3c                       \
1279                strspn.3c                       \
1280                strstr.3c                       \
1281                strtof.3c                       \
1282                strtok.3c                       \
1283                strtok_r.3c                     \
1284                strtold.3c                      \
1285                strtoll.3c                      \
1286                strtoull.3c                     \
1287                strtoumax.3c                    \
1288                strxfrm_l.3c                    \
1289                swapcontext.3c                  \
1290                swprintf.3c                     \
1291                swscanf.3c                      \
1292                tdelete.3c                      \
1293                tempnam.3c                      \
1294                textdomain.3c                   \
1295                tfind.3c                        \
1296                thr_continue.3c                 \
1297                thr_getspecific.3c              \
1298                thr_keycreate_once.3c           \
1299                thr_setname.3c                  \
1300                thr_setconcurrency.3c           \
1301                thr_setprio.3c                  \
1302                thr_setspecific.3c              \
1303                thrd_sleep.3c                   \
1304                timegm.3c                       \
1305                timer_getoverrun.3c             \
1306                timer_gettime.3c                \
1307                timerclear.3c                   \
1308                timercmp.3c                     \
1309                timerfd_gettime.3c              \
1310                timerfd_settime.3c              \
1311                timerisset.3c                   \
1312                timersub.3c                     \
1313                tmpnam_r.3c                     \
1314                tolower_l.3c                    \
```

```
1315                    toupper_l.3c                    \
1316                    towctrans.3c                    \
1317                    towctrans_l.3c                  \
1318                    towlower_l.3c                   \
1319                    towupper_l.3c                   \
1320                    tss_create.3c                   \
1321                    tss_delete.3c                   \
1322                    tss_get.3c                      \
1323                    tss_set.3c                      \
1324                    ttyname_r.3c                    \
1325                    twalk.3c                        \
1326                    tzset.3c                        \
1327                    uconv_u16tou8.3c                \
1328                    uconv_u32tou16.3c               \
1329                    uconv_u32tou8.3c                \
1330                    uconv_u8tou16.3c                \
1331                    uconv_u8tou32.3c                \
1332                    ucred_free.3c                   \
1333                    ucred_get.3c                    \
1334                    ucred_getegid.3c                \
1335                    ucred_geteuid.3c                \
1336                    ucred_getgroups.3c              \
1337                    ucred_getlabel.3c               \
1338                    ucred_getpflags.3c              \
1339                    ucred_getpid.3c                 \
1340                    ucred_getprivset.3c             \
1341                    ucred_getprojid.3c              \
1342                    ucred_getrgid.3c                \
1343                    ucred_getruid.3c                \
1344                    ucred_getsgid.3c                \
1345                    ucred_getsuid.3c                \
1346                    ucred_getzoneid.3c              \
1347                    ucred_size.3c                   \
1348                    ulckpwdf.3c                     \
1349                    ulltostr.3c                     \
1350                    unordered.3c                    \
1351                    updwtmp.3c                      \
1352                    updwtmpx.3c                     \
1353                    utmpname.3c                     \
1354                    utmpxname.3c                    \
1355                    valloc.3c                       \
1356                    vasprintf.3c                    \
1357                    verr.3c                         \
1358                    verrx.3c                        \
1359                    vfprintf.3c                     \
1360                    vfscanf.3c                      \
1361                    vfwscanf.3c                     \
1362                    vscanf.3c                       \
1363                    vsnprintf.3c                    \
1364                    vsprintf.3c                     \
1365                    vsscanf.3c                      \
1366                    vswprintf.3c                    \
1367                    vswscanf.3c                     \
1368                    vwarn.3c                        \
1369                    vwarnx.3c                       \
1370                    vwprintf.3c                     \
1371                    vwscanf.3c                      \
1372                    wait4.3c                        \
1373                    warn.3c                         \
1374                    warnx.3c                        \
1375                    watof.3c                        \
1376                    watoi.3c                        \
1377                    watol.3c                        \
1378                    watoll.3c                       \
1379                    wcpncpy.3c                      \
1380                    wcrtomb_l.3c                    \
```



```
1381                    wcscasecmp_l.3c                 \
1382                    wcscat.3c                       \
1383                    wcschr.3c                       \
1384                    wcscmp.3c                       \
1385                    wcscoll_l.3c                    \
1386                    wcscpy.3c                       \
1387                    wcscspn.3c                      \
1388                    wcsetno.3c                      \
1389                    wcsncasecmp.3c                  \
1390                    wcsncasecmp_l.3c                \
1391                    wcsncat.3c                      \
1392                    wcsncmp.3c                      \
1393                    wcsncpy.3c                      \
1394                    wcsnlen.3c                      \
1395                    wcsnrtombs.3c                   \
1396                    wcsnrtombs_l.3c                 \
1397                    wcspbrk.3c                      \
1398                    wcsrchr.3c                      \
1399                    wcsrtombs_l.3c                  \
1400                    wcsspn.3c                       \
1401                    wcstof.3c                       \
1402                    wcstok.3c                       \
1403                    wcstold.3c                      \
1404                    wcstoll.3c                      \
1405                    wcstombs_l.3c                   \
1406                    wcstoull.3c                     \
1407                    wcstoumax.3c                    \
1408                    wcswcs.3c                       \
1409                    wcswidth_l.3c                   \
1410                    wctob_l.3c                      \
1411                    wctomb_l.3c                     \
1412                    wctrans_l.3c                    \
1413                    wctype_l.3c                     \
1414                    wcwidth_l.3c                    \
1415                    windex.3c                       \
1416                    wordfree.3c                     \
1417                    wprintf.3c                      \
1418                    wrindex.3c                      \
1419                    wscanf.3c                       \
1420                    wscasecmp.3c                    \
1421                    wscat.3c                        \
1422                    wschr.3c                        \
1423                    wscmp.3c                        \
1424                    wscol.3c                        \
1425                    wscoll.3c                       \
1426                    wscpy.3c                        \
1427                    wscspn.3c                       \
1428                    wsdup.3c                        \
1429                    wslen.3c                        \
1430                    wsncasecmp.3c                   \
1431                    wsncat.3c                       \
1432                    wsncmp.3c                       \
1433                    wsncpy.3c                       \
1434                    wspbrk.3c                       \
1435                    wsrchr.3c                       \
1436                    wsspn.3c                        \
1437                    wstod.3c                        \
1438                    wstok.3c                        \
1439                    wstol.3c                        \
1440                    wstostr.3c                      \
1441                    wsxfrm.3c

1443 __flbf.3c                          := LINKSRC = __fbufsize.3c
1444 __fpending.3c                      := LINKSRC = __fbufsize.3c
1445 __fpurge.3c                        := LINKSRC = __fbufsize.3c
1446 __freadable.3c                     := LINKSRC = __fbufsize.3c
```

```
__freading.3c                   := LINKSRC = __fbufsize.3c
__fsetlocking.3c                := LINKSRC = __fbufsize.3c
__fwritable.3c                  := LINKSRC = __fbufsize.3c
__fwriting.3c                   := LINKSRC = __fbufsize.3c
_flushlbf.3c                    := LINKSRC = __fbufsize.3c

_setjmp.3c                      := LINKSRC = _longjmp.3c

l64a.3c                         := LINKSRC = a64l.3c

labs.3c                         := LINKSRC = abs.3c
llabs.3c                        := LINKSRC = abs.3c

aiowrite.3c                     := LINKSRC = aioread.3c

arc4random_buf.3c               := LINKSRC = arc4random.3c
arc4random_uniform.3c           := LINKSRC = arc4random.3c

at_quick_exit.3c                := LINKSRC = quick_exit.3c

atomic_add_16.3c                := LINKSRC = atomic_add.3c
atomic_add_16_nv.3c             := LINKSRC = atomic_add.3c
atomic_add_32.3c                := LINKSRC = atomic_add.3c
atomic_add_32_nv.3c             := LINKSRC = atomic_add.3c
atomic_add_64.3c                := LINKSRC = atomic_add.3c
atomic_add_64_nv.3c             := LINKSRC = atomic_add.3c
atomic_add_8.3c                 := LINKSRC = atomic_add.3c
atomic_add_8_nv.3c              := LINKSRC = atomic_add.3c
atomic_add_char.3c              := LINKSRC = atomic_add.3c
atomic_add_char_nv.3c           := LINKSRC = atomic_add.3c
atomic_add_int.3c               := LINKSRC = atomic_add.3c
atomic_add_int_nv.3c            := LINKSRC = atomic_add.3c
atomic_add_long.3c              := LINKSRC = atomic_add.3c
atomic_add_long_nv.3c           := LINKSRC = atomic_add.3c
atomic_add_ptr.3c               := LINKSRC = atomic_add.3c
atomic_add_ptr_nv.3c            := LINKSRC = atomic_add.3c
atomic_add_short.3c             := LINKSRC = atomic_add.3c
atomic_add_short_nv.3c          := LINKSRC = atomic_add.3c
atomic_and_16.3c                := LINKSRC = atomic_and.3c
atomic_and_16_nv.3c             := LINKSRC = atomic_and.3c
atomic_and_32.3c                := LINKSRC = atomic_and.3c
atomic_and_32_nv.3c             := LINKSRC = atomic_and.3c
atomic_and_64.3c                := LINKSRC = atomic_and.3c
atomic_and_64_nv.3c             := LINKSRC = atomic_and.3c
atomic_and_8.3c                 := LINKSRC = atomic_and.3c
atomic_and_8_nv.3c              := LINKSRC = atomic_and.3c
atomic_and_uchar.3c             := LINKSRC = atomic_and.3c
atomic_and_uchar_nv.3c          := LINKSRC = atomic_and.3c
atomic_and_uint.3c              := LINKSRC = atomic_and.3c
atomic_and_uint_nv.3c           := LINKSRC = atomic_and.3c
atomic_and_ulong.3c             := LINKSRC = atomic_and.3c
atomic_and_ulong_nv.3c          := LINKSRC = atomic_and.3c
atomic_and_ushort.3c            := LINKSRC = atomic_and.3c
atomic_and_ushort_nv.3c         := LINKSRC = atomic_and.3c

atomic_clear_long_excl.3c       := LINKSRC = atomic_bits.3c
atomic_set_long_excl.3c         := LINKSRC = atomic_bits.3c

atomic_cas_16.3c                := LINKSRC = atomic_cas.3c
atomic_cas_32.3c                := LINKSRC = atomic_cas.3c
atomic_cas_64.3c                := LINKSRC = atomic_cas.3c
atomic_cas_8.3c                 := LINKSRC = atomic_cas.3c
atomic_cas_ptr.3c               := LINKSRC = atomic_cas.3c
atomic_cas_uchar.3c             := LINKSRC = atomic_cas.3c
atomic_cas_uint.3c              := LINKSRC = atomic_cas.3c
atomic_cas_ulong.3c             := LINKSRC = atomic_cas.3c
```



```
atomic_cas_ushort.3c            := LINKSRC = atomic_cas.3c

atomic_dec_16.3c                := LINKSRC = atomic_dec.3c
atomic_dec_16_nv.3c             := LINKSRC = atomic_dec.3c
atomic_dec_32.3c                := LINKSRC = atomic_dec.3c
atomic_dec_32_nv.3c             := LINKSRC = atomic_dec.3c
atomic_dec_64.3c                := LINKSRC = atomic_dec.3c
atomic_dec_64_nv.3c             := LINKSRC = atomic_dec.3c
atomic_dec_8.3c                 := LINKSRC = atomic_dec.3c
atomic_dec_8_nv.3c              := LINKSRC = atomic_dec.3c
atomic_dec_ptr.3c               := LINKSRC = atomic_dec.3c
atomic_dec_ptr_nv.3c            := LINKSRC = atomic_dec.3c
atomic_dec_uchar.3c             := LINKSRC = atomic_dec.3c
atomic_dec_uchar_nv.3c          := LINKSRC = atomic_dec.3c
atomic_dec_uint.3c              := LINKSRC = atomic_dec.3c
atomic_dec_uint_nv.3c           := LINKSRC = atomic_dec.3c
atomic_dec_ulong.3c             := LINKSRC = atomic_dec.3c
atomic_dec_ulong_nv.3c          := LINKSRC = atomic_dec.3c
atomic_dec_ushort.3c            := LINKSRC = atomic_dec.3c
atomic_dec_ushort_nv.3c         := LINKSRC = atomic_dec.3c

atomic_inc_16.3c                := LINKSRC = atomic_inc.3c
atomic_inc_16_nv.3c             := LINKSRC = atomic_inc.3c
atomic_inc_32.3c                := LINKSRC = atomic_inc.3c
atomic_inc_32_nv.3c             := LINKSRC = atomic_inc.3c
atomic_inc_64.3c                := LINKSRC = atomic_inc.3c
atomic_inc_64_nv.3c             := LINKSRC = atomic_inc.3c
atomic_inc_8.3c                 := LINKSRC = atomic_inc.3c
atomic_inc_8_nv.3c              := LINKSRC = atomic_inc.3c
atomic_inc_ptr.3c               := LINKSRC = atomic_inc.3c
atomic_inc_ptr_nv.3c            := LINKSRC = atomic_inc.3c
atomic_inc_uchar.3c             := LINKSRC = atomic_inc.3c
atomic_inc_uchar_nv.3c          := LINKSRC = atomic_inc.3c
atomic_inc_uint.3c              := LINKSRC = atomic_inc.3c
atomic_inc_uint_nv.3c           := LINKSRC = atomic_inc.3c
atomic_inc_ulong.3c             := LINKSRC = atomic_inc.3c
atomic_inc_ulong_nv.3c          := LINKSRC = atomic_inc.3c
atomic_inc_ushort.3c            := LINKSRC = atomic_inc.3c
atomic_inc_ushort_nv.3c         := LINKSRC = atomic_inc.3c

atomic_or_16.3c                 := LINKSRC = atomic_or.3c
atomic_or_16_nv.3c              := LINKSRC = atomic_or.3c
atomic_or_32.3c                 := LINKSRC = atomic_or.3c
atomic_or_32_nv.3c              := LINKSRC = atomic_or.3c
atomic_or_64.3c                 := LINKSRC = atomic_or.3c
atomic_or_64_nv.3c              := LINKSRC = atomic_or.3c
atomic_or_8.3c                  := LINKSRC = atomic_or.3c
atomic_or_8_nv.3c               := LINKSRC = atomic_or.3c
atomic_or_uchar.3c              := LINKSRC = atomic_or.3c
atomic_or_uchar_nv.3c           := LINKSRC = atomic_or.3c
atomic_or_uint.3c               := LINKSRC = atomic_or.3c
atomic_or_uint_nv.3c            := LINKSRC = atomic_or.3c
atomic_or_ulong.3c              := LINKSRC = atomic_or.3c
atomic_or_ulong_nv.3c           := LINKSRC = atomic_or.3c
atomic_or_ushort.3c             := LINKSRC = atomic_or.3c
atomic_or_ushort_nv.3c          := LINKSRC = atomic_or.3c

atomic_swap_16.3c               := LINKSRC = atomic_swap.3c
atomic_swap_32.3c               := LINKSRC = atomic_swap.3c
atomic_swap_64.3c               := LINKSRC = atomic_swap.3c
atomic_swap_8.3c                := LINKSRC = atomic_swap.3c
atomic_swap_ptr.3c              := LINKSRC = atomic_swap.3c
atomic_swap_uchar.3c            := LINKSRC = atomic_swap.3c
atomic_swap_uint.3c             := LINKSRC = atomic_swap.3c
atomic_swap_ulong.3c            := LINKSRC = atomic_swap.3c
atomic_swap_ushort.3c           := LINKSRC = atomic_swap.3c
```

```
1580 bcmp.3c                                := LINKSRC = bstring.3c
1581 bcopy.3c                               := LINKSRC = bstring.3c
1582 bzero.3c                               := LINKSRC = bstring.3c
1583 explicit_bzero.3c                      := LINKSRC = bstring.3c

1585 btowc_l.3c                             := LINKSRC = btowc.3c

1587 htonl.3c                               := LINKSRC = byteorder.3c
1588 htonll.3c                              := LINKSRC = byteorder.3c
1589 htons.3c                               := LINKSRC = byteorder.3c
1590 ntohl.3c                               := LINKSRC = byteorder.3c
1591 ntohll.3c                              := LINKSRC = byteorder.3c
1592 ntohs.3c                               := LINKSRC = byteorder.3c

1594 canonicalize_file_name.3c             := LINKSRC = realpath.3c

1596 catclose.3c                           := LINKSRC = catopen.3c

1598 cfgetospeed.3c                        := LINKSRC = cfgetispeed.3c

1600 cfsetospeed.3c                        := LINKSRC = cfsetispeed.3c

1602 clock_getres.3c                       := LINKSRC = clock_settime.3c
1603 clock_gettime.3c                      := LINKSRC = clock_settime.3c

1605 fdwalk.3c                            := LINKSRC = closefrom.3c

1607 cnd_broadcast.3c                      := LINKSRC = cnd.3c
1608 cnd_destroy.3c                        := LINKSRC = cnd.3c
1609 cnd_init.3c                           := LINKSRC = cnd.3c
1610 cnd_signal.3c                         := LINKSRC = cnd.3c
1611 cnd_timedwait.3c                      := LINKSRC = cnd.3c
1612 cnd_wait.3c                           := LINKSRC = cnd.3c

1614 cond_broadcast.3c                     := LINKSRC = cond_init.3c
1615 cond_destroy.3c                       := LINKSRC = cond_init.3c
1616 cond_reltimedwait.3c                  := LINKSRC = cond_init.3c
1617 cond_signal.3c                        := LINKSRC = cond_init.3c
1618 cond_timedwait.3c                     := LINKSRC = cond_init.3c
1619 cond_wait.3c                          := LINKSRC = cond_init.3c

1621 csetcol.3c                           := LINKSRC = cset.3c
1622 csetlen.3c                           := LINKSRC = cset.3c
1623 csetno.3c                            := LINKSRC = cset.3c
1624 wcsetno.3c                           := LINKSRC = cset.3c

1626 ctermid_r.3c                         := LINKSRC = ctermid.3c

1628 asctime.3c                           := LINKSRC = ctime.3c
1629 asctime_r.3c                         := LINKSRC = ctime.3c
1630 ctime_r.3c                           := LINKSRC = ctime.3c
1631 gmtime.3c                            := LINKSRC = ctime.3c
1632 gmtime_r.3c                          := LINKSRC = ctime.3c
1633 localtime.3c                         := LINKSRC = ctime.3c
1634 localtime_r.3c                       := LINKSRC = ctime.3c
1635 tzset.3c                             := LINKSRC = ctime.3c

1637 isalnum.3c                           := LINKSRC = ctype.3c
1638 isalnum_l.3c                         := LINKSRC = ctype.3c
1639 isalpha.3c                           := LINKSRC = ctype.3c
1640 isalpha_l.3c                         := LINKSRC = ctype.3c
1641 isascii.3c                           := LINKSRC = ctype.3c
1642 isblank.3c                           := LINKSRC = ctype.3c
1643 isblank_l.3c                         := LINKSRC = ctype.3c
1644 iscntrl.3c                           := LINKSRC = ctype.3c
```



```
1645 iscntrl_l.3c                         := LINKSRC = ctype.3c
1646 isdigit.3c                           := LINKSRC = ctype.3c
1647 isdigit_l.3c                         := LINKSRC = ctype.3c
1648 isgraph.3c                           := LINKSRC = ctype.3c
1649 isgraph_l.3c                         := LINKSRC = ctype.3c
1650 islower.3c                           := LINKSRC = ctype.3c
1651 islower_l.3c                         := LINKSRC = ctype.3c
1652 isprint.3c                           := LINKSRC = ctype.3c
1653 isprint_l.3c                         := LINKSRC = ctype.3c
1654 ispunct.3c                           := LINKSRC = ctype.3c
1655 ispunct_l.3c                         := LINKSRC = ctype.3c
1656 isspace.3c                           := LINKSRC = ctype.3c
1657 isspace_l.3c                         := LINKSRC = ctype.3c
1658 isupper.3c                           := LINKSRC = ctype.3c
1659 isupper_l.3c                         := LINKSRC = ctype.3c
1660 isxdigit.3c                          := LINKSRC = ctype.3c
1661 isxdigit_l.3c                        := LINKSRC = ctype.3c

1663 decimal_to_double.3c                 := LINKSRC = decimal_to_floating.3c
1664 decimal_to_extended.3c               := LINKSRC = decimal_to_floating.3c
1665 decimal_to_quadruple.3c              := LINKSRC = decimal_to_floating.3c
1666 decimal_to_single.3c                 := LINKSRC = decimal_to_floating.3c

1668 ldiv.3c                              := LINKSRC = div.3c
1669 lldiv.3c                             := LINKSRC = div.3c

1671 dladdr1.3c                          := LINKSRC = dladdr.3c

1673 dlmopen.3c                          := LINKSRC = dlopen.3c

1675 door_unbind.3c                       := LINKSRC = door_bind.3c

1677 door_setparam.3c                     := LINKSRC = door_getparam.3c

1679 erand48.3c                           := LINKSRC = drand48.3c
1680 jrand48.3c                           := LINKSRC = drand48.3c
1681 lcong48.3c                           := LINKSRC = drand48.3c
1682 lrand48.3c                           := LINKSRC = drand48.3c
1683 mrand48.3c                           := LINKSRC = drand48.3c
1684 nrand48.3c                           := LINKSRC = drand48.3c
1685 seed48.3c                            := LINKSRC = drand48.3c
1686 srand48.3c                           := LINKSRC = drand48.3c

1688 dup3.3c                              := LINKSRC = dup2.3c

1690 fconvert.3c                          := LINKSRC = econvert.3c
1691 gconvert.3c                          := LINKSRC = econvert.3c
1692 qeconvert.3c                         := LINKSRC = econvert.3c
1693 qfconvert.3c                         := LINKSRC = econvert.3c
1694 qgconvert.3c                         := LINKSRC = econvert.3c
1695 seconvert.3c                         := LINKSRC = econvert.3c
1696 sfconvert.3c                         := LINKSRC = econvert.3c
1697 sgconvert.3c                         := LINKSRC = econvert.3c

1699 fcvt.3c                              := LINKSRC = ecvt.3c
1700 gcvt.3c                              := LINKSRC = ecvt.3c

1702 _edata.3c                            := LINKSRC = end.3c
1703 _end.3c                              := LINKSRC = end.3c
1704 _etext.3c                            := LINKSRC = end.3c
1705 edata.3c                             := LINKSRC = end.3c
1706 etext.3c                             := LINKSRC = end.3c

1708 be16toh.3c                           := LINKSRC = endian.3c
1709 be32toh.3c                           := LINKSRC = endian.3c
1710 be64toh.3c                           := LINKSRC = endian.3c
```

```
1711 betoh16.3c                          := LINKSRC = endian.3c
1712 betoh32.3c                          := LINKSRC = endian.3c
1713 betoh64.3c                          := LINKSRC = endian.3c
1714 htobe16.3c                          := LINKSRC = endian.3c
1715 htobe32.3c                          := LINKSRC = endian.3c
1716 htobe64.3c                          := LINKSRC = endian.3c
1717 htole16.3c                          := LINKSRC = endian.3c
1718 htole32.3c                          := LINKSRC = endian.3c
1719 htole64.3c                          := LINKSRC = endian.3c
1720 le16toh.3c                          := LINKSRC = endian.3c
1721 le32toh.3c                          := LINKSRC = endian.3c
1722 le64toh.3c                          := LINKSRC = endian.3c
1723 letoh16.3c                          := LINKSRC = endian.3c
1724 letoh32.3c                          := LINKSRC = endian.3c
1725 letoh64.3c                          := LINKSRC = endian.3c

1727 epoll_create1.3c                     := LINKSRC = epoll_create.3c
1728 epoll_pwait.3c                       := LINKSRC = epoll_wait.3c

1730 errx.3c                             := LINKSRC = err.3c
1731 verr.3c                             := LINKSRC = err.3c
1732 verrx.3c                            := LINKSRC = err.3c
1733 vwarn.3c                            := LINKSRC = err.3c
1734 vwarnx.3c                           := LINKSRC = err.3c
1735 warn.3c                             := LINKSRC = err.3c
1736 warnx.3c                            := LINKSRC = err.3c

1738 euccol.3c                           := LINKSRC = euclen.3c
1739 eucscol.3c                          := LINKSRC = euclen.3c

1741 _exithandle.3c                      := LINKSRC = exit.3c

1743 clearerr.3c                         := LINKSRC = ferror.3c
1744 feof.3c                             := LINKSRC = ferror.3c
1745 fileno.3c                           := LINKSRC = ferror.3c

1747 ffsl.3c                             := LINKSRC = ffs.3c
1748 ffsll.3c                            := LINKSRC = ffs.3c
1749 fls.3c                              := LINKSRC = ffs.3c
1750 flsl.3c                             := LINKSRC = ffs.3c
1751 flsll.3c                            := LINKSRC = ffs.3c

1753 fsetattr.3c                         := LINKSRC = fgetattr.3c
1754 getattrat.3c                        := LINKSRC = fgetattr.3c
1755 setattrat.3c                        := LINKSRC = fgetattr.3c

1757 getc.3c                             := LINKSRC = fgetc.3c
1758 getc_unlocked.3c                    := LINKSRC = fgetc.3c
1759 getchar.3c                          := LINKSRC = fgetc.3c
1760 getchar_unlocked.3c                 := LINKSRC = fgetc.3c
1761 getw.3c                             := LINKSRC = fgetc.3c

1763 fgetwc_l.3c                         := LINKSRC = fgetwc.3c

1765 double_to_decimal.3c                := LINKSRC = floating_to_decimal.3c
1766 extended_to_decimal.3c              := LINKSRC = floating_to_decimal.3c
1767 quadruple_to_decimal.3c             := LINKSRC = floating_to_decimal.3c
1768 single_to_decimal.3c                := LINKSRC = floating_to_decimal.3c

1770 ftrylockfile.3c                     := LINKSRC = flockfile.3c
1771 funlockfile.3c                      := LINKSRC = flockfile.3c

1773 fpgetmask.3c                        := LINKSRC = fpgetround.3c
1774 fpgetsticky.3c                      := LINKSRC = fpgetround.3c
1775 fpsetmask.3c                        := LINKSRC = fpgetround.3c
1776 fpsetround.3c                       := LINKSRC = fpgetround.3c
```



```
1777 fpsetsticky.3c                      := LINKSRC = fpgetround.3c

1779 putc.3c                            := LINKSRC = fputc.3c
1780 putc_unlocked.3c                   := LINKSRC = fputc.3c
1781 putchar.3c                         := LINKSRC = fputc.3c
1782 putchar_unlocked.3c                := LINKSRC = fputc.3c
1783 putw.3c                            := LINKSRC = fputc.3c

1785 putwc.3c                           := LINKSRC = fputwc.3c
1786 putwchar.3c                        := LINKSRC = fputwc.3c

1788 fseeko.3c                          := LINKSRC = fseek.3c

1790 ftello.3c                          := LINKSRC = ftell.3c

1792 fts_children.3c                    := LINKSRC = fts.3c
1793 fts_close.3c                       := LINKSRC = fts.3c
1794 fts_open.3c                        := LINKSRC = fts.3c
1795 fts_read.3c                        := LINKSRC = fts.3c
1796 fts_set.3c                         := LINKSRC = fts.3c

1798 nftw.3c                            := LINKSRC = ftw.3c

1800 swprintf.3c                        := LINKSRC = fwprintf.3c
1801 wprintf.3c                         := LINKSRC = fwprintf.3c

1803 swscanf.3c                         := LINKSRC = fwscanf.3c
1804 vfwscanf.3c                        := LINKSRC = fwscanf.3c
1805 vswscanf.3c                        := LINKSRC = fwscanf.3c
1806 vwscanf.3c                         := LINKSRC = fwscanf.3c
1807 wscanf.3c                          := LINKSRC = fwscanf.3c

1809 gethomelgroup.3c                   := LINKSRC = getcpuid.3c

1811 endgrent.3c                        := LINKSRC = getgrnam.3c
1812 fgetgrent.3c                       := LINKSRC = getgrnam.3c
1813 fgetgrent_r.3c                     := LINKSRC = getgrnam.3c
1814 getgrent.3c                        := LINKSRC = getgrnam.3c
1815 getgrent_r.3c                      := LINKSRC = getgrnam.3c
1816 getgrgid.3c                        := LINKSRC = getgrnam.3c
1817 getgrgid_r.3c                      := LINKSRC = getgrnam.3c
1818 getgrnam_r.3c                      := LINKSRC = getgrnam.3c
1819 setgrent.3c                        := LINKSRC = getgrnam.3c

1821 sethostname.3c                     := LINKSRC = gethostname.3c

1823 gethrvtime.3c                      := LINKSRC = gethrtime.3c

1825 getdelim.3c                        := LINKSRC = getline.3c

1827 getlogin_r.3c                      := LINKSRC = getlogin.3c

1829 getextmntent.3c                    := LINKSRC = getmntent.3c
1830 getmntany.3c                      := LINKSRC = getmntent.3c
1831 hasmntopt.3c                      := LINKSRC = getmntent.3c
1832 putmntent.3c                      := LINKSRC = getmntent.3c
1833 resetmnttab.3c                    := LINKSRC = getmntent.3c

1835 endnetgrent.3c                     := LINKSRC = getnetgrent.3c
1836 getnetgrent_r.3c                   := LINKSRC = getnetgrent.3c
1837 innetgr.3c                         := LINKSRC = getnetgrent.3c
1838 setnetgrent.3c                     := LINKSRC = getnetgrent.3c

1840 get_nprocs_conf.3c                 := LINKSRC = get_nprocs.3c

1842 getopt_long_clip.3c                := LINKSRC = getopt_long.3c
```

```
1843 getopt_long_only.3c                    := LINKSRC = getopt_long.3c

1845 getpassphrase.3c                       := LINKSRC = getpass.3c

1847 setpriority.3c                         := LINKSRC = getpriority.3c

1849 setprogname.3c                         := LINKSRC = getprogname.3c

1851 endpwent.3c                            := LINKSRC = getpwnam.3c
1852 fgetpwent.3c                           := LINKSRC = getpwnam.3c
1853 fgetpwent_r.3c                         := LINKSRC = getpwnam.3c
1854 getpwent.3c                           := LINKSRC = getpwnam.3c
1855 getpwent_r.3c                         := LINKSRC = getpwnam.3c
1856 getpwnam_r.3c                         := LINKSRC = getpwnam.3c
1857 getpwuid.3c                           := LINKSRC = getpwnam.3c
1858 getpwuid_r.3c                         := LINKSRC = getpwnam.3c
1859 setpwent.3c                           := LINKSRC = getpwnam.3c

1861 fgets.3c                              := LINKSRC = gets.3c

1863 endspent.3c                           := LINKSRC = getspnam.3c
1864 fgetspent.3c                          := LINKSRC = getspnam.3c
1865 fgetspent_r.3c                        := LINKSRC = getspnam.3c
1866 getspent.3c                          := LINKSRC = getspnam.3c
1867 getspent_r.3c                        := LINKSRC = getspnam.3c
1868 getspnam_r.3c                        := LINKSRC = getspnam.3c
1869 setspent.3c                          := LINKSRC = getspnam.3c

1871 bind_textdomain_codeset.3c            := LINKSRC = gettext.3c
1872 bindtextdomain.3c                     := LINKSRC = gettext.3c
1873 dcgettext.3c                          := LINKSRC = gettext.3c
1874 dcngettext.3c                         := LINKSRC = gettext.3c
1875 dgettext.3c                           := LINKSRC = gettext.3c
1876 dngettext.3c                          := LINKSRC = gettext.3c
1877 ngettext.3c                           := LINKSRC = gettext.3c
1878 textdomain.3c                         := LINKSRC = gettext.3c

1880 settimeofday.3c                       := LINKSRC = gettimeofday.3c

1882 endusershell.3c                       := LINKSRC = getusershell.3c
1883 setusershell.3c                       := LINKSRC = getusershell.3c

1885 endutent.3c                           := LINKSRC = getutent.3c
1886 getutid.3c                           := LINKSRC = getutent.3c
1887 getutline.3c                         := LINKSRC = getutent.3c
1888 pututline.3c                         := LINKSRC = getutent.3c
1889 setutent.3c                          := LINKSRC = getutent.3c
1890 utmpname.3c                          := LINKSRC = getutent.3c

1892 endutxent.3c                          := LINKSRC = getutxent.3c
1893 getutmp.3c                           := LINKSRC = getutxent.3c
1894 getutmpx.3c                          := LINKSRC = getutxent.3c
1895 getutxid.3c                          := LINKSRC = getutxent.3c
1896 getutxline.3c                        := LINKSRC = getutxent.3c
1897 pututxline.3c                        := LINKSRC = getutxent.3c
1898 setutxent.3c                         := LINKSRC = getutxent.3c
1899 updwtmp.3c                           := LINKSRC = getutxent.3c
1900 updwtmpx.3c                          := LINKSRC = getutxent.3c
1901 utmpxname.3c                         := LINKSRC = getutxent.3c

1903 getvfsany.3c                          := LINKSRC = getvfsent.3c
1904 getvfsfile.3c                         := LINKSRC = getvfsent.3c
1905 getvfsspec.3c                         := LINKSRC = getvfsent.3c

1907 getwc_l.3c                            := LINKSRC = getwc.3c
```



```
1909 getwchar_l.3c                         := LINKSRC = getwchar.3c

1911 fgetws.3c                             := LINKSRC = getws.3c

1913 getzoneidbyname.3c                    := LINKSRC = getzoneid.3c
1914 getzonenamebyid.3c                    := LINKSRC = getzoneid.3c

1916 globfree.3c                           := LINKSRC = glob.3c

1918 hcreate.3c                            := LINKSRC = hsearch.3c
1919 hdestroy.3c                           := LINKSRC = hsearch.3c

1921 rindex.3c                             := LINKSRC = index.3c

1923 inet6.3c                              := LINKSRC = inet.3c
1924 inet_addr.3c                          := LINKSRC = inet.3c
1925 inet_aton.3c                          := LINKSRC = inet.3c
1926 inet_lnaof.3c                         := LINKSRC = inet.3c
1927 inet_makeaddr.3c                      := LINKSRC = inet.3c
1928 inet_netof.3c                         := LINKSRC = inet.3c
1929 inet_network.3c                       := LINKSRC = inet.3c
1930 inet_ntoa.3c                          := LINKSRC = inet.3c
1931 inet_ntop.3c                          := LINKSRC = inet.3c
1932 inet_pton.3c                          := LINKSRC = inet.3c

1934 remque.3c                             := LINKSRC = insque.3c

1936 finite.3c                             := LINKSRC = isnand.3c
1937 fpclass.3c                            := LINKSRC = isnand.3c
1938 isnanf.3c                             := LINKSRC = isnand.3c
1939 unordered.3c                          := LINKSRC = isnand.3c

1941 isenglish.3c                          := LINKSRC = iswalpha.3c
1942 isideogram.3c                         := LINKSRC = iswalpha.3c
1943 isnumber.3c                           := LINKSRC = iswalpha.3c
1944 isphonogram.3c                        := LINKSRC = iswalpha.3c
1945 isspecial.3c                          := LINKSRC = iswalpha.3c
1946 iswalnum.3c                           := LINKSRC = iswalpha.3c
1947 iswalnum_l.3c                         := LINKSRC = iswalpha.3c
1948 iswalpha_l.3c                         := LINKSRC = iswalpha.3c
1949 iswascii.3c                           := LINKSRC = iswalpha.3c
1950 iswblank.3c                           := LINKSRC = iswalpha.3c
1951 iswblank_l.3c                         := LINKSRC = iswalpha.3c
1952 iswcntrl.3c                           := LINKSRC = iswalpha.3c
1953 iswcntrl_l.3c                         := LINKSRC = iswalpha.3c
1954 iswdigit.3c                           := LINKSRC = iswalpha.3c
1955 iswdigit_l.3c                         := LINKSRC = iswalpha.3c
1956 iswgraph.3c                           := LINKSRC = iswalpha.3c
1957 iswgraph_l.3c                         := LINKSRC = iswalpha.3c
1958 iswhexnumber.3c                       := LINKSRC = iswalpha.3c
1959 iswhexnumber_l.3c                     := LINKSRC = iswalpha.3c
1960 iswideogram.3c                       := LINKSRC = iswalpha.3c
1961 iswideogram_l.3c                     := LINKSRC = iswalpha.3c
1962 iswlower.3c                           := LINKSRC = iswalpha.3c
1963 iswlower_l.3c                         := LINKSRC = iswalpha.3c
1964 iswnumber.3c                          := LINKSRC = iswalpha.3c
1965 iswnumber_l.3c                        := LINKSRC = iswalpha.3c
1966 iswphonogram.3c                      := LINKSRC = iswalpha.3c
1967 iswphonogram_l.3c                    := LINKSRC = iswalpha.3c
1968 iswprint.3c                          := LINKSRC = iswalpha.3c
1969 iswprint_l.3c                        := LINKSRC = iswalpha.3c
1970 iswpunct.3c                          := LINKSRC = iswalpha.3c
1971 iswpunct_l.3c                        := LINKSRC = iswalpha.3c
1972 iswspace.3c                          := LINKSRC = iswalpha.3c
1973 iswspace_l.3c                        := LINKSRC = iswalpha.3c
1974 iswspecial.3c                        := LINKSRC = iswalpha.3c
```

```
1975 iswspecial_l.3c                    := LINKSRC = iswalpha.3c
1976 iswupper.3c                       := LINKSRC = iswalpha.3c
1977 iswupper_l.3c                     := LINKSRC = iswalpha.3c
1978 iswxdigit.3c                      := LINKSRC = iswalpha.3c
1979 iswxdigit_l.3c                    := LINKSRC = iswalpha.3c

1981 iswctype_l.3c                     := LINKSRC = iswctype.3c

1983 ulckpwdf.3c                       := LINKSRC = lckpwdf.3c

1985 lfind.3c                          := LINKSRC = lsearch.3c

1987 swapcontext.3c                    := LINKSRC = makecontext.3c

1989 major.3c                          := LINKSRC = makedev.3c
1990 minor.3c                          := LINKSRC = makedev.3c

1992 alloca.3c                         := LINKSRC = malloc.3c
1993 calloc.3c                         := LINKSRC = malloc.3c
1994 free.3c                           := LINKSRC = malloc.3c
1995 freezero.3c                       := LINKSRC = malloc.3c
1996 memalign.3c                       := LINKSRC = malloc.3c
1997 realloc.3c                        := LINKSRC = malloc.3c
1998 reallocarray.3c                   := LINKSRC = malloc.3c
1999 recallocarray.3c                  := LINKSRC = malloc.3c
2000 valloc.3c                         := LINKSRC = malloc.3c

2002 mblen_l.3c                        := LINKSRC = mblen.3c

2004 mbrlen_l.3c                       := LINKSRC = mbrlen.3c

2006 mbrtowc_l.3c                      := LINKSRC = mbrtowc.3c

2008 mbsinit_l.3c                      := LINKSRC = mbsinit.3c

2010 mbsnrtowcs.3c                     := LINKSRC = mbsrtowcs.3c
2011 mbsnrtowcs_l.3c                   := LINKSRC = mbsrtowcs.3c
2012 mbsrtowcs_l.3c                    := LINKSRC = mbsrtowcs.3c

2014 mbstowcs_l.3c                     := LINKSRC = mbstowcs.3c

2016 mbtowc_l.3c                       := LINKSRC = mbtowc.3c

2018 membar_consumer.3c                := LINKSRC = membar_ops.3c
2019 membar_enter.3c                   := LINKSRC = membar_ops.3c
2020 membar_exit.3c                    := LINKSRC = membar_ops.3c
2021 membar_producer.3c                := LINKSRC = membar_ops.3c

2023 memccpy.3c                        := LINKSRC = memory.3c
2024 memchr.3c                         := LINKSRC = memory.3c
2025 memcmp.3c                         := LINKSRC = memory.3c
2026 memcpy.3c                         := LINKSRC = memory.3c
2027 memmem.3c                         := LINKSRC = memory.3c
2028 memmove.3c                        := LINKSRC = memory.3c
2029 memset.3c                         := LINKSRC = memory.3c

2031 mkfifoat.3c                       := LINKSRC = mkfifo.3c

2033 mkdtemp.3c                        := LINKSRC = mkstemp.3c
2034 mkostemp.3c                       := LINKSRC = mkstemp.3c
2035 mkostemps.3c                      := LINKSRC = mkstemp.3c
2036 mkstemps.3c                       := LINKSRC = mkstemp.3c

2038 munlock.3c                        := LINKSRC = mlock.3c

2040 munlockall.3c                     := LINKSRC = mlockall.3c
```



```
2042 mq_reltimedreceive_np.3c          := LINKSRC = mq_receive.3c
2043 mq_timedreceive.3c                := LINKSRC = mq_receive.3c

2045 mq_reltimedsend_np.3c             := LINKSRC = mq_send.3c
2046 mq_timedsend.3c                   := LINKSRC = mq_send.3c

2048 mtx_destroy.3c                    := LINKSRC = mtx.3c
2049 mtx_init.3c                       := LINKSRC = mtx.3c
2050 mtx_lock.3c                       := LINKSRC = mtx.3c
2051 mtx_timedlock.3c                  := LINKSRC = mtx.3c
2052 mtx_trylock.3c                    := LINKSRC = mtx.3c
2053 mtx_unlock.3c                     := LINKSRC = mtx.3c

2055 mutex_consistent.3c               := LINKSRC = mutex_init.3c
2056 mutex_destroy.3c                  := LINKSRC = mutex_init.3c
2057 mutex_lock.3c                     := LINKSRC = mutex_init.3c
2058 mutex_trylock.3c                  := LINKSRC = mutex_init.3c
2059 mutex_unlock.3c                   := LINKSRC = mutex_init.3c

2061 thrd_sleep.3c                     := LINKSRC = nanosleep.3c

2063 dbm_clearerr.3c                   := LINKSRC = ndbm.3c
2064 dbm_close.3c                      := LINKSRC = ndbm.3c
2065 dbm_delete.3c                     := LINKSRC = ndbm.3c
2066 dbm_error.3c                      := LINKSRC = ndbm.3c
2067 dbm_fetch.3c                      := LINKSRC = ndbm.3c
2068 dbm_firstkey.3c                   := LINKSRC = ndbm.3c
2069 dbm_nextkey.3c                    := LINKSRC = ndbm.3c
2070 dbm_open.3c                       := LINKSRC = ndbm.3c
2071 dbm_store.3c                      := LINKSRC = ndbm.3c

2073 duplocale.3c                      := LINKSRC = newlocale.3c
2074 freelocale.3c                     := LINKSRC = newlocale.3c

2076 nl_langinfo_l.3c                  := LINKSRC = nl_langinfo.3c

2078 fdopendir.3c                      := LINKSRC = opendir.3c

2080 errno.3c                          := LINKSRC = perror.3c

2082 pclose.3c                         := LINKSRC = popen.3c

2084 port_dissociate.3c                := LINKSRC = port_associate.3c

2086 port_getn.3c                      := LINKSRC = port_get.3c

2088 port_sendn.3c                     := LINKSRC = port_send.3c

2090 posix_spawnp.3c                   := LINKSRC = posix_spawn.3c

2092 posix_spawn_file_actions_addopen.3c  := LINKSRC = posix_spawn_file_actions_ad
2093 posix_spawn_file_actions_init.3c    := LINKSRC = posix_spawn_file_actions_de

2095 posix_spawnattr_init.3c             := LINKSRC = posix_spawnattr_destroy.3c

2097 posix_spawnattr_setflags.3c         := LINKSRC = posix_spawnattr_getflags.3c

2099 posix_spawnattr_setpgroup.3c        := LINKSRC = posix_spawnattr_getpgroup.3

2101 posix_spawnattr_setschedparam.3c    := LINKSRC = posix_spawnattr_getschedpar

2103 posix_spawnattr_setschedpolicy.3c   := LINKSRC = posix_spawnattr_getschedpol

2105 posix_spawnattr_setsigdefault.3c    := LINKSRC = posix_spawnattr_getsigdefau
```

```
2107 posix_spawnattr_setsigignore_np.3c      := LINKSRC = posix_spawnattr_getsigignor

2109 posix_spawnattr_setsigmask.3c           := LINKSRC = posix_spawnattr_getsigmask.

2111 asprintf.3c                            := LINKSRC = printf.3c
2112 fprintf.3c                             := LINKSRC = printf.3c
2113 snprintf.3c                            := LINKSRC = printf.3c
2114 sprintf.3c                             := LINKSRC = printf.3c

2116 priv_allocset.3c                       := LINKSRC = priv_addset.3c
2117 priv_basicset.3c                       := LINKSRC = priv_addset.3c
2118 priv_copyset.3c                        := LINKSRC = priv_addset.3c
2119 priv_delset.3c                         := LINKSRC = priv_addset.3c
2120 priv_emptyset.3c                       := LINKSRC = priv_addset.3c
2121 priv_fillset.3c                        := LINKSRC = priv_addset.3c
2122 priv_freeset.3c                        := LINKSRC = priv_addset.3c
2123 priv_intersect.3c                      := LINKSRC = priv_addset.3c
2124 priv_inverse.3c                        := LINKSRC = priv_addset.3c
2125 priv_isemptyset.3c                     := LINKSRC = priv_addset.3c
2126 priv_isequalset.3c                     := LINKSRC = priv_addset.3c
2127 priv_isfullset.3c                      := LINKSRC = priv_addset.3c
2128 priv_ismember.3c                       := LINKSRC = priv_addset.3c
2129 priv_issubset.3c                       := LINKSRC = priv_addset.3c
2130 priv_union.3c                          := LINKSRC = priv_addset.3c

2132 priv_ineffect.3c                       := LINKSRC = priv_set.3c

2134 priv_getbyname.3c                      := LINKSRC = priv_str_to_set.3c
2135 priv_getbynum.3c                       := LINKSRC = priv_str_to_set.3c
2136 priv_getsetbyname.3c                   := LINKSRC = priv_str_to_set.3c
2137 priv_getsetbynum.3c                    := LINKSRC = priv_str_to_set.3c
2138 priv_gettext.3c                        := LINKSRC = priv_str_to_set.3c
2139 priv_set_to_str.3c                     := LINKSRC = priv_str_to_set.3c

2141 psiginfo.3c                            := LINKSRC = psignal.3c

2143 pthread_attr_setdetachstate.3c         := LINKSRC = pthread_attr_getdetachstate

2145 pthread_attr_setguardsize.3c          := LINKSRC = pthread_attr_getguardsize.3

2147 pthread_attr_setinheritsched.3c        := LINKSRC = pthread_attr_getinheritsche

2149 pthread_attr_setname_np.3c             := LINKSRC = pthread_attr_getname_np.3c

2151 pthread_attr_setschedparam.3c          := LINKSRC = pthread_attr_getschedparam.

2153 pthread_attr_setschedpolicy.3c         := LINKSRC = pthread_attr_getschedpolicy

2155 pthread_attr_setscope.3c               := LINKSRC = pthread_attr_getscope.3c
2156 pthread_attr_setstack.3c               := LINKSRC = pthread_attr_getstack.3c

2158 pthread_attr_setstackaddr.3c           := LINKSRC = pthread_attr_getstackaddr.3

2160 pthread_attr_setstacksize.3c          := LINKSRC = pthread_attr_getstacksize.3

2162 pthread_attr_destroy.3c               := LINKSRC = pthread_attr_init.3c

2164 pthread_barrier_init.3c               := LINKSRC = pthread_barrier_destroy.3c

2166 pthread_barrierattr_init.3c           := LINKSRC = pthread_barrierattr_destroy

2168 pthread_barrierattr_setpshared.3c     := LINKSRC = pthread_barrierattr_getpsha

2170 pthread_cond_destroy.3c               := LINKSRC = pthread_cond_init.3c

2172 pthread_cond_broadcast.3c             := LINKSRC = pthread_cond_signal.3c
```



```
2174 pthread_cond_reltimedwait_np.3c       := LINKSRC = pthread_cond_wait.3c
2175 pthread_cond_timedwait.3c             := LINKSRC = pthread_cond_wait.3c

2177 pthread_condattr_setclock.3c          := LINKSRC = pthread_condattr_getclock.3

2179 pthread_condattr_setpshared.3c        := LINKSRC = pthread_condattr_getpshared

2181 pthread_condattr_destroy.3c           := LINKSRC = pthread_condattr_init.3c

2183 pthread_setconcurrency.3c             := LINKSRC = pthread_getconcurrency.3c

2185 pthread_setname_np.3c                 := LINKSRC = pthread_getname_np.3c

2187 pthread_setschedparam.3c              := LINKSRC = pthread_getschedparam.3c

2189 pthread_setspecific.3c                := LINKSRC = pthread_getspecific.3c

2191 pthread_key_create_once_np.3c         := LINKSRC = pthread_key_create.3c

2193 pthread_mutex_setprioceiling.3c       := LINKSRC = pthread_mutex_getprioceilin

2195 pthread_mutex_destroy.3c              := LINKSRC = pthread_mutex_init.3c

2197 pthread_mutex_trylock.3c              := LINKSRC = pthread_mutex_lock.3c
2198 pthread_mutex_unlock.3c               := LINKSRC = pthread_mutex_lock.3c

2200 pthread_mutex_reltimedlock_np.3c      := LINKSRC = pthread_mutex_timedlock.3c

2202 pthread_mutexattr_setprioceiling.3c   := LINKSRC = pthread_mutexattr_getprioce

2204 pthread_mutexattr_setprotocol.3c      := LINKSRC = pthread_mutexattr_getprotoc

2206 pthread_mutexattr_setpshared.3c       := LINKSRC = pthread_mutexattr_getpshare

2208 pthread_mutexattr_setrobust.3c        := LINKSRC = pthread_mutexattr_getrobust

2210 pthread_mutexattr_settype.3c          := LINKSRC = pthread_mutexattr_gettype.3

2212 pthread_mutexattr_destroy.3c          := LINKSRC = pthread_mutexattr_init.3c

2214 pthread_rwlock_destroy.3c             := LINKSRC = pthread_rwlock_init.3c

2216 pthread_rwlock_tryrdlock.3c           := LINKSRC = pthread_rwlock_rdlock.3c

2218 pthread_rwlock_reltimedrdlock_np.3c   := LINKSRC = pthread_rwlock_timedrdlock.

2220 pthread_rwlock_reltimedwrlock_np.3c   := LINKSRC = pthread_rwlock_timedwrlock.

2222 pthread_rwlock_trywrlock.3c           := LINKSRC = pthread_rwlock_wrlock.3c

2224 pthread_rwlockattr_setpshared.3c      := LINKSRC = pthread_rwlockattr_getpshar

2226 pthread_rwlockattr_destroy.3c         := LINKSRC = pthread_rwlockattr_init.3c

2228 pthread_spin_init.3c                  := LINKSRC = pthread_spin_destroy.3c

2230 pthread_spin_trylock.3c               := LINKSRC = pthread_spin_lock.3c

2232 fputs.3c                             := LINKSRC = puts.3c

2234 rand_r.3c                            := LINKSRC = rand.3c
2235 srand.3c                             := LINKSRC = rand.3c

2237 initstate.3c                         := LINKSRC = random.3c
2238 setstate.3c                          := LINKSRC = random.3c
```

```
2239 srandom.3c                          := LINKSRC = random.3c

2241 rctlblk_get_enforced_value.3c       := LINKSRC = rctlblk_set_value.3c
2242 rctlblk_get_firing_time.3c          := LINKSRC = rctlblk_set_value.3c
2243 rctlblk_get_global_action.3c        := LINKSRC = rctlblk_set_value.3c
2244 rctlblk_get_global_flags.3c         := LINKSRC = rctlblk_set_value.3c
2245 rctlblk_get_local_action.3c         := LINKSRC = rctlblk_set_value.3c
2246 rctlblk_get_local_flags.3c          := LINKSRC = rctlblk_set_value.3c
2247 rctlblk_get_privilege.3c            := LINKSRC = rctlblk_set_value.3c
2248 rctlblk_get_recipient_pid.3c        := LINKSRC = rctlblk_set_value.3c
2249 rctlblk_get_value.3c                := LINKSRC = rctlblk_set_value.3c
2250 rctlblk_set_local_action.3c         := LINKSRC = rctlblk_set_value.3c
2251 rctlblk_set_local_flags.3c          := LINKSRC = rctlblk_set_value.3c
2252 rctlblk_set_privilege.3c            := LINKSRC = rctlblk_set_value.3c
2253 rctlblk_set_recipient_pid.3c        := LINKSRC = rctlblk_set_value.3c
2254 rctlblk_size.3c                     := LINKSRC = rctlblk_set_value.3c

2256 re_exec.3c                          := LINKSRC = re_comp.3c

2258 readdir_r.3c                        := LINKSRC = readdir.3c

2260 regex.3c                           := LINKSRC = regcmp.3c

2262 regerror.3c                        := LINKSRC = regcomp.3c
2263 regexec.3c                         := LINKSRC = regcomp.3c
2264 regfree.3c                         := LINKSRC = regcomp.3c

2266 rw_rdlock.3c                       := LINKSRC = rwlock.3c
2267 rw_tryrdlock.3c                    := LINKSRC = rwlock.3c
2268 rw_trywrlock.3c                    := LINKSRC = rwlock.3c
2269 rw_unlock.3c                       := LINKSRC = rwlock.3c
2270 rw_wrlock.3c                       := LINKSRC = rwlock.3c
2271 rwlock_destroy.3c                  := LINKSRC = rwlock.3c
2272 rwlock_init.3c                     := LINKSRC = rwlock.3c

2274 alphasort.3c                       := LINKSRC = scandir.3c

2276 fscanf.3c                          := LINKSRC = scanf.3c
2277 sscanf.3c                          := LINKSRC = scanf.3c
2278 vfscanf.3c                         := LINKSRC = scanf.3c
2279 vscanf.3c                          := LINKSRC = scanf.3c
2280 vsscanf.3c                         := LINKSRC = scanf.3c

2282 sched_get_priority_min.3c          := LINKSRC = sched_get_priority_max.3c

2284 schedctl_exit.3c                   := LINKSRC = schedctl_init.3c
2285 schedctl_lookup.3c                 := LINKSRC = schedctl_init.3c
2286 schedctl_start.3c                  := LINKSRC = schedctl_init.3c
2287 schedctl_stop.3c                   := LINKSRC = schedctl_init.3c

2289 FD_CLR.3c                          := LINKSRC = select.3c
2290 FD_ISSET.3c                        := LINKSRC = select.3c
2291 FD_SET.3c                          := LINKSRC = select.3c
2292 FD_ZERO.3c                         := LINKSRC = select.3c
2293 pselect.3c                         := LINKSRC = select.3c

2295 sem_reltimedwait_np.3c             := LINKSRC = sem_timedwait.3c

2297 sem_trywait.3c                     := LINKSRC = sem_wait.3c

2299 sema_destroy.3c                    := LINKSRC = semaphore.3c
2300 sema_init.3c                       := LINKSRC = semaphore.3c
2301 sema_post.3c                       := LINKSRC = semaphore.3c
2302 sema_trywait.3c                    := LINKSRC = semaphore.3c
2303 sema_wait.3c                       := LINKSRC = semaphore.3c
```



```
2305 abort_handler_s.3c                 := LINKSRC = set_constraint_handler_s.3c
2306 ignore_handler_s.3c                := LINKSRC = set_constraint_handler_s.3c

2308 setvbuf.3c                         := LINKSRC = setbuf.3c

2310 setlinebuf.3c                      := LINKSRC = setbuffer.3c

2312 longjmp.3c                         := LINKSRC = setjmp.3c
2313 siglongjmp.3c                      := LINKSRC = setjmp.3c
2314 sigsetjmp.3c                       := LINKSRC = setjmp.3c

2316 sighold.3c                         := LINKSRC = signal.3c
2317 sigignore.3c                       := LINKSRC = signal.3c
2318 sigpause.3c                        := LINKSRC = signal.3c
2319 sigrelse.3c                        := LINKSRC = signal.3c
2320 sigset.3c                          := LINKSRC = signal.3c

2322 sigaddset.3c                       := LINKSRC = sigsetops.3c
2323 sigdelset.3c                       := LINKSRC = sigsetops.3c
2324 sigemptyset.3c                     := LINKSRC = sigsetops.3c
2325 sigfillset.3c                      := LINKSRC = sigsetops.3c
2326 sigismember.3c                     := LINKSRC = sigsetops.3c

2328 sigtimedwait.3c                    := LINKSRC = sigwaitinfo.3c

2330 gsignal.3c                         := LINKSRC = ssignal.3c

2332 stderr.3c                          := LINKSRC = stdio.3c
2333 stdin.3c                           := LINKSRC = stdio.3c
2334 stdout.3c                          := LINKSRC = stdio.3c

2336 sig2str.3c                         := LINKSRC = str2sig.3c

2338 strcoll_l.3c                       := LINKSRC = strcoll.3c

2340 strerror_l.3c                      := LINKSRC = strerror.3c
2341 strerror_r.3c                      := LINKSRC = strerror.3c

2343 strfmon_l.3c                       := LINKSRC = strfmon.3c

2345 ascftime.3c                        := LINKSRC = strftime.3c
2346 cftime.3c                          := LINKSRC = strftime.3c
2347 strftime_l.3c                      := LINKSRC = strftime.3c

2349 stpcpy.3c                          := LINKSRC = string.3c
2350 stpncpy.3c                         := LINKSRC = string.3c
2351 strcasecmp.3c                      := LINKSRC = string.3c
2352 strcasecmp_l.3c                    := LINKSRC = string.3c
2353 strcasestr.3c                      := LINKSRC = string.3c
2354 strcasestr_l.3c                    := LINKSRC = string.3c
2355 strcat.3c                          := LINKSRC = string.3c
2356 strchr.3c                          := LINKSRC = string.3c
2357 strchrnul.3c                       := LINKSRC = string.3c
2358 strcmp.3c                          := LINKSRC = string.3c
2359 strcpy.3c                          := LINKSRC = string.3c
2360 strcspn.3c                         := LINKSRC = string.3c
2361 strdup.3c                          := LINKSRC = string.3c
2362 strdupa.3c                         := LINKSRC = string.3c
2363 strlcat.3c                         := LINKSRC = string.3c
2364 strlcpy.3c                         := LINKSRC = string.3c
2365 strlen.3c                          := LINKSRC = string.3c
2366 strncasecmp.3c                     := LINKSRC = string.3c
2367 strncasecmp_l.3c                   := LINKSRC = string.3c
2368 strncat.3c                         := LINKSRC = string.3c
2369 strncmp.3c                         := LINKSRC = string.3c
2370 strncpy.3c                         := LINKSRC = string.3c
```

```
2371 strndup.3c                          := LINKSRC = string.3c
2372 strndupa.3c                         := LINKSRC = string.3c
2373 strnlen.3c                          := LINKSRC = string.3c
2374 strnstr.3c                          := LINKSRC = string.3c
2375 strpbrk.3c                          := LINKSRC = string.3c
2376 strrchr.3c                          := LINKSRC = string.3c
2377 strsep.3c                           := LINKSRC = string.3c
2378 strspn.3c                           := LINKSRC = string.3c
2379 strstr.3c                           := LINKSRC = string.3c
2380 strtok.3c                           := LINKSRC = string.3c
2381 strtok_r.3c                         := LINKSRC = string.3c

2383 file_to_decimal.3c                  := LINKSRC = string_to_decimal.3c
2384 func_to_decimal.3c                  := LINKSRC = string_to_decimal.3c

2386 strptime_l.3c                       := LINKSRC = strptime.3c

2388 atof.3c                            := LINKSRC = strtod.3c
2389 strtof.3c                           := LINKSRC = strtod.3c
2390 strtold.3c                          := LINKSRC = strtod.3c

2392 strtoumax.3c                        := LINKSRC = strtoimax.3c

2394 atoi.3c                            := LINKSRC = strtol.3c
2395 atol.3c                            := LINKSRC = strtol.3c
2396 atoll.3c                           := LINKSRC = strtol.3c
2397 lltostr.3c                         := LINKSRC = strtol.3c
2398 strtoll.3c                         := LINKSRC = strtol.3c
2399 ulltostr.3c                        := LINKSRC = strtol.3c

2401 strtoull.3c                        := LINKSRC = strtoul.3c

2403 wstostr.3c                         := LINKSRC = strtows.3c

2405 strxfrm_l.3c                       := LINKSRC = strxfrm.3c

2407 closelog.3c                        := LINKSRC = syslog.3c
2408 openlog.3c                         := LINKSRC = syslog.3c
2409 setlogmask.3c                      := LINKSRC = syslog.3c

2411 thr_setconcurrency.3c              := LINKSRC = thr_getconcurrency.3c

2413 thr_setprio.3c                     := LINKSRC = thr_getprio.3c

2415 thr_getspecific.3c                 := LINKSRC = thr_keycreate.3c
2416 thr_keycreate_once.3c              := LINKSRC = thr_keycreate.3c
2417 thr_setspecific.3c                 := LINKSRC = thr_keycreate.3c

2419 thr_continue.3c                    := LINKSRC = thr_suspend.3c

2421 thr_setname.3c                     := LINKSRC = thr_getname.3c

2423 timegm.3c                          := LINKSRC = mktime.3c

2425 timer_getoverrun.3c                := LINKSRC = timer_settime.3c
2426 timer_gettime.3c                   := LINKSRC = timer_settime.3c

2428 timerclear.3c                      := LINKSRC = timeradd.3c
2429 timercmp.3c                        := LINKSRC = timeradd.3c
2430 timerisset.3c                      := LINKSRC = timeradd.3c
2431 timersub.3c                        := LINKSRC = timeradd.3c

2433 timerfd_gettime.3c                 := LINKSRC = timerfd_create.3c
2434 timerfd_settime.3c                 := LINKSRC = timerfd_create.3c

2436 tempnam.3c                         := LINKSRC = tmpnam.3c
```



```
2437 tmpnam_r.3c                        := LINKSRC = tmpnam.3c

2439 tolower_l.3c                       := LINKSRC = tolower.3c

2441 toupper_l.3c                       := LINKSRC = toupper.3c

2443 towlower_l.3c                      := LINKSRC = towlower.3c

2445 towupper_l.3c                      := LINKSRC = towupper.3c

2447 ftruncate.3c                       := LINKSRC = truncate.3c

2449 tdelete.3c                         := LINKSRC = tsearch.3c
2450 tfind.3c                           := LINKSRC = tsearch.3c
2451 twalk.3c                           := LINKSRC = tsearch.3c

2453 tss_create.3c                      := LINKSRC = tss.3c
2454 tss_delete.3c                      := LINKSRC = tss.3c
2455 tss_get.3c                         := LINKSRC = tss.3c
2456 tss_set.3c                         := LINKSRC = tss.3c

2458 ttyname_r.3c                       := LINKSRC = ttyname.3c

2460 uconv_u16tou8.3c                   := LINKSRC = uconv_u16tou32.3c
2461 uconv_u32tou16.3c                  := LINKSRC = uconv_u16tou32.3c
2462 uconv_u32tou8.3c                   := LINKSRC = uconv_u16tou32.3c
2463 uconv_u8tou16.3c                   := LINKSRC = uconv_u16tou32.3c
2464 uconv_u8tou32.3c                   := LINKSRC = uconv_u16tou32.3c

2466 ucred_free.3c                      := LINKSRC = ucred.3c
2467 ucred_get.3c                       := LINKSRC = ucred.3c
2468 ucred_getegid.3c                   := LINKSRC = ucred.3c
2469 ucred_geteuid.3c                   := LINKSRC = ucred.3c
2470 ucred_getgroups.3c                 := LINKSRC = ucred.3c
2471 ucred_getlabel.3c                  := LINKSRC = ucred.3c
2472 ucred_getpflags.3c                 := LINKSRC = ucred.3c
2473 ucred_getpid.3c                    := LINKSRC = ucred.3c
2474 ucred_getprivset.3c                := LINKSRC = ucred.3c
2475 ucred_getprojid.3c                 := LINKSRC = ucred.3c
2476 ucred_getrgid.3c                   := LINKSRC = ucred.3c
2477 ucred_getruid.3c                   := LINKSRC = ucred.3c
2478 ucred_getsgid.3c                   := LINKSRC = ucred.3c
2479 ucred_getsuid.3c                   := LINKSRC = ucred.3c
2480 ucred_getzoneid.3c                 := LINKSRC = ucred.3c
2481 ucred_size.3c                      := LINKSRC = ucred.3c

2483 vswprintf.3c                       := LINKSRC = vfwprintf.3c
2484 vwprintf.3c                        := LINKSRC = vfwprintf.3c

2486 vasprintf.3c                       := LINKSRC = vprintf.3c
2487 vfprintf.3c                        := LINKSRC = vprintf.3c
2488 vsnprintf.3c                       := LINKSRC = vprintf.3c
2489 vsprintf.3c                        := LINKSRC = vprintf.3c

2491 wait4.3c                           := LINKSRC = wait3.3c

2493 addrtosymstr.3c                    := LINKSRC = walkcontext.3c
2494 backtrace.3c                       := LINKSRC = walkcontext.3c
2495 backtrace_symbols.3c               := LINKSRC = walkcontext.3c
2496 backtrace_symbols_fd.3c            := LINKSRC = walkcontext.3c
2497 printstack.3c                      := LINKSRC = walkcontext.3c

2499 wcpncpy.3c                         := LINKSRC = wcpcpy.3c

2501 wcrtomb_l.3c                       := LINKSRC = wcrtomb.3c
```

```
2503 wcscasecmp_l.3c                    := LINKSRC = wcscasecmp.3c
2504 wcsncasecmp.3c                    := LINKSRC = wcscasecmp.3c
2505 wcsncasecmp_l.3c                  := LINKSRC = wcscasecmp.3c

2507 wcscoll_l.3c                      := LINKSRC = wcscoll.3c
2508 wscoll.3c                         := LINKSRC = wcscoll.3c

2510 wcsnlen.3c                        := LINKSRC = wcslen.3c

2512 wcsnrtombs.3c                     := LINKSRC = wcsrtombs.3c
2513 wcsnrtombs_l.3c                   := LINKSRC = wcsrtombs.3c
2514 wcsrtombs_l.3c                    := LINKSRC = wcsrtombs.3c

2516 watof.3c                          := LINKSRC = wcstod.3c
2517 wcstof.3c                         := LINKSRC = wcstod.3c
2518 wcstold.3c                        := LINKSRC = wcstod.3c
2519 wstod.3c                          := LINKSRC = wcstod.3c

2521 wcstoumax.3c                      := LINKSRC = wcstoimax.3c

2523 watoi.3c                          := LINKSRC = wcstol.3c
2524 watol.3c                          := LINKSRC = wcstol.3c
2525 watoll.3c                         := LINKSRC = wcstol.3c
2526 wcstoll.3c                        := LINKSRC = wcstol.3c
2527 wstol.3c                          := LINKSRC = wcstol.3c

2529 wcstombs_l.3c                     := LINKSRC = wcstombs.3c

2531 wcstoull.3c                       := LINKSRC = wcstoul.3c

2533 wcscat.3c                         := LINKSRC = wcstring.3c
2534 wcschr.3c                         := LINKSRC = wcstring.3c
2535 wcscmp.3c                         := LINKSRC = wcstring.3c
2536 wcscpy.3c                         := LINKSRC = wcstring.3c
2537 wcscspn.3c                        := LINKSRC = wcstring.3c
2538 wcsncat.3c                        := LINKSRC = wcstring.3c
2539 wcsncmp.3c                        := LINKSRC = wcstring.3c
2540 wcsncpy.3c                        := LINKSRC = wcstring.3c
2541 wcspbrk.3c                        := LINKSRC = wcstring.3c
2542 wcsrchr.3c                        := LINKSRC = wcstring.3c
2543 wcsspn.3c                         := LINKSRC = wcstring.3c
2544 wcstok.3c                         := LINKSRC = wcstring.3c
2545 wcswcs.3c                         := LINKSRC = wcstring.3c
2546 windex.3c                         := LINKSRC = wcstring.3c
2547 wrindex.3c                        := LINKSRC = wcstring.3c
2548 wscat.3c                          := LINKSRC = wcstring.3c
2549 wschr.3c                          := LINKSRC = wcstring.3c
2550 wscmp.3c                          := LINKSRC = wcstring.3c
2551 wscpy.3c                          := LINKSRC = wcstring.3c
2552 wscspn.3c                         := LINKSRC = wcstring.3c
2553 wslen.3c                          := LINKSRC = wcstring.3c
2554 wsncat.3c                         := LINKSRC = wcstring.3c
2555 wsncmp.3c                         := LINKSRC = wcstring.3c
2556 wsncpy.3c                         := LINKSRC = wcstring.3c
2557 wspbrk.3c                         := LINKSRC = wcstring.3c
2558 wsrchr.3c                         := LINKSRC = wcstring.3c
2559 wsspn.3c                          := LINKSRC = wcstring.3c
2560 wstok.3c                          := LINKSRC = wcstring.3c
2561 wscasecmp.3c                      := LINKSRC = wstring.3c
2562 wscol.3c                          := LINKSRC = wstring.3c
2563 wsdup.3c                          := LINKSRC = wstring.3c
2564 wsncasecmp.3c                     := LINKSRC = wstring.3c

2566 wcswidth_l.3c                     := LINKSRC = wcswidth.3c

2568 wsxfrm.3c                         := LINKSRC = wcsxfrm.3c
```



```
2570 wctob_l.3c                       := LINKSRC = wctob.3c

2572 wctomb_l.3c                      := LINKSRC = wctomb.3c

2574 towctrans.3c                     := LINKSRC = wctrans.3c
2575 towctrans_l.3c                   := LINKSRC = wctrans.3c
2576 wctrans_l.3c                     := LINKSRC = wctrans.3c

2578 wctype_l.3c                      := LINKSRC = wctype.3c

2580 wcwidth_l.3c                     := LINKSRC = wcwidth.3c

2582 wordfree.3c                      := LINKSRC = wordexp.3c

2584 .KEY_STATE:

2586 include         $(SRC)/man/Makefile.man

2588 install:        $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
**********************************************************
    2162 Mon Oct 15 13:25:39 2018
new/usr/src/man/man3c/pthread_attr_getname_np.3c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2018 Joyent, Inc.
.\"
.Dd "August 22, 2018"
.Dt PTHREAD_ATTR_GETNAME_NP 3C
.Os
.Sh NAME
.Nm pthread_attr_getname_np ,
.Nm pthread_attr_setname_np
.Nd get or set thread name attribute
.Sh SYNOPSIS
.In pthread.h
.
.Ft int
.Fo pthread_attr_getname_np
.Fa "pthread_attr_t *restrict attr"
.Fa "char *name"
.Fa "size_t len"
.Fc
.
.Ft int
.Fo pthread_attr_setname_np
.Fa "pthread_attr_t *restrict attr"
.Fa "const char *name"
.Fc
.
.Sh DESCRIPTION
The
.Fn pthread_attr_setname_np
and
.Fn pthread_attr_getname_np
functions, respectively, set and get the thread name attribute in
.Fa attr
to
.Fa name .
For
.Fn pthread_attr_getname_np ,
.Fa len
is the size of
.Fa name .
Any threads created with
.Xr pthread_create 3c
using
.Fa attr
will have their name set to
.Fa name
upon creation.
.Pp
Thread names are limited to
.Dv PTHREAD_MAX_NAMELEN_NP
```



```
including the terminating NUL.
They may only contain printable ASCII characters.
.Sh RETURN VALUES
Upon successful completion, the
.Fn pthread_attr_getname_np
and
.Fn pthread_attr_setname_np
functions return
.Sy 0 .
Otherwise, an error number is returned to indicate the error.
.Sh ERRORS
The
.Fn pthread_attr_getname_np
function may fail with:
.Bl -tag -width Er
.It Er EINVAL
The
.Fa name
argument is
.Sy NULL .
.It Er ERANGE
The size of
.Fa name
as indicated by
.Fa len
is too small to contain the thread name.
The buffer may be over-written with partial contents of the thread name.
.El
.Pp
The
.Fn pthread_attr_setname_np
function may fail with:
.Bl -tag -width Er
.It Er ERANGE
The length of name given in
.Fa name
exceeds the maximum size allowed.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
.Sy MT-Safe
.Sh SEE ALSO
.Xr pthread_create 3c ,
.Xr pthread_getname_np 3c
```

```
**********************************************************
    2422 Mon Oct 15 13:25:41 2018
new/usr/src/man/man3c/pthread_getname_np.3c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2018 Joyent, Inc.
.\"
.Dd "August 22, 2018"
.Dt PTHREAD_GETNAME_NP 3C
.Os
.Sh NAME
.Nm pthread_getname_np ,
.Nm pthread_setname_np
.Nd get or set the name of a thread
.Sh SYNOPSIS
.In pthread.h
.
.Ft int
.Fo pthread_getname_np
.Fa "pthread_t tid"
.Fa "char *name"
.Fa "size_t len"
.Fc
.
.Ft int
.Fo pthread_setname_np
.Fa "pthread_t tid"
.Fa "const char *name"
.Fc
.
.Sh DESCRIPTION
The
.Fn pthread_getname_np
and
.Fn pthread_setname_np
functions, respectively, get and set the names of the thread whose id is given
by the
.Fa tid
parameter.
For
.Fn pthread_getname_np ,
.Fa len
indicates the size of
.Fa name .
.Pp
Thread names are limited to
.Dv PTHREAD_MAX_NAMELEN_NP
including the terminating NUL.
They may only contain printable ASCII characters.
.Pp
To clear a thread name, call
.Fn pthread_setname_np
with
.Sy NULL .
```



```
.Pp
Unlike some other systems, threads do not inherit the process name by default.
.Sh RETURN VALUES
Upon successful completion, the
.Fn pthread_getname_np
and
.Fn pthread_setname_np
functions return
.Sy 0 .
Otherwise, an error number is returned to indicate the error.
If the thread identified by
.Fa tid
does not have a name set,
.Fa pthread_getname_np
will be set to an empty string (length = 0).
.Sh ERRORS
On failure, the contents of the buffer are undefined.
Errors from
.Xr open 2 ,
.Xr read 2 ,
or
.Xr write 2
are possible.
In addition, the
.Fn pthread_getname_np
function will fail with:
.Bl -tag -width Er
.It Er EINVAL
The
.Fa name
argument is
.Sy NULL .
.It Er ERANGE
The size of
.Fa name
as given by
.Fa len
was not large enough to contain the name of the thread.
.It Er ESRCH
The thread
.Fa tid
was not found.
.El
.Pp
The
.Fn pthread_setname_np
function will fail with:
.Bl -tag -width Er
.It Er ERANGE
The length of
.Fa name
exceeds the maximum allowed size.
.It Er ESRCH
The thread
.Fa tid
was not found.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
.Sy MT-Safe
.Sh SEE ALSO
.Xr pthread_attr_getname_np 3c ,
.Xr pthread_create 3c
```

```
**********************************************************
    2305 Mon Oct 15 13:25:43 2018
new/usr/src/man/man3c/thr_getname.3c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2018 Joyent, Inc.
.\"
.Dd "August 22, 2018"
.Dt THR_GETNAME 3C
.Os
.Sh NAME
.Nm thr_getname ,
.Nm thr_setname
.Nd get or set the name of a thread
.Sh SYNOPSIS
.In thread.h
.
.Ft int
.Fo thr_getname
.Fa "thread_t tid"
.Fa "char *name"
.Fa "size_t len"
.Fc
.
.Ft int
.Fo thr_setname
.Fa "thread_t tid"
.Fa "const char *name"
.Fc
.
.Sh DESCRIPTION
The
.Fn thr_getname
and
.Fn thr_setname
functions, respectively, get and set the names of the thread whose id is given
by the
.Fa tid
parameter.
For
.Fn thr_getname ,
.Fa len
indicates the size of
.Fa name .
.Pp
Thread names are limited to
.Dv THREAD_NAME_MAX
including the terminating NUL.
They may only contain printable ASCII characters.
.Pp
To clear a thread name, call
.Fn thr_setname
with
.Sy NULL .
```



```
.Pp
Unlike some other systems, threads do not inherit the process name by default.
.Sh RETURN VALUES
Upon successful completion, the
.Fn thr_getname
and
.Fn thr_setname
functions return
.Sy 0 .
Otherwise, an error number is returned to indicate the error.
If the thread identified by
.Fa tid
does not have a name set,
.Fa thr_getname
will be set to an empty string (length = 0).
.Sh ERRORS
On failure, the contents of the buffer are undefined.
Errors from
.Xr open 2 ,
.Xr read 2 ,
or
.Xr write 2
are possible.
In addition, the
.Fn thr_getname
function will fail with:
.Bl -tag -width Er
.It Er EINVAL
The
.Fa name
argument is
.Sy NULL .
.It Er ERANGE
The size of
.Fa name
as given by
.Fa len
was not large enough to contain the name of the thread.
.It Er ESRCH
The thread
.Fa tid
was not found.
.El
.Pp
The
.Fn thr_setname
function will fail with:
.Bl -tag -width Er
.It Er ERANGE
The length of
.Fa name
exceeds the maximum allowed size.
.It Er ESRCH
The thread
.Fa tid
was not found.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
.Sy MT-Safe
.Sh SEE ALSO
.Xr pthread_setname_np 3c ,
.Xr thr_create 3c
```

```
**********************************************************
   34612 Mon Oct 15 13:25:44 2018
new/usr/src/man/man3lib/libproc.3lib
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2018 Joyent, Inc.
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd August 31, 2018
.Dd June 06, 2016
.Dt LIBPROC 3LIB
.Os
.Sh NAME
.Nm libproc
.Nd process control library
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Sh DESCRIPTION
The
.Nm
library provides consumers a general series of interfaces to inspect
and control both live processes and core files.
It is intended for introspection tools such as debuggers by providing a
high-level interface to the /proc file system
.Pf ( Xr proc 4 ) .
.Pp
The
.Nm
library provides interfaces that focus on:
.Bl -bullet -offset indent
.It
Creating and attaching to live process, core files, and arbitrary ELF
objects.
.It
Interrogating the state of a process or core file.
.It
Manipulating the current state of a process or thread.
.It
Interrogating the state of threads of a process or core file.
.It
Running system calls in the context of another process.
.It
Various utilities for iterating process and core file file descriptors,
mappings, symbols, and more.
.It
Various utilities to support debugging tools.
.El
.Ss Live Processes
The
.Nm
library can be used to manipulate running processes and to create new
ones.
To manipulate an existing process first
```



```
.Em grab
it with the
.Em Pgrab
function.
A process is generally stopped as a side effect of grabbing it.
Callers must exercise caution, as if they do not use the library correctly, or
they terminate unexpectedly, a process may remain stopped.
.Pp
Unprivileged users may only grab their own processes.
Users with the privilege
.Sy PRIV_PROC_OWNER
may manipulate processes that they do not own; however, additional
restrictions as described in
.Xr privileges 5
apply.
.Pp
In addition, the
.Fn Pcreate
and
.Fn Pxcreate
functions may be used to create processes which are always controlled by
the library.
.Ss Core Files
The
.Nm
library has the ability to open and interpret core files produced by
processes on the system.
Process core dump generation is controlled by the
.Xr coreadm 1M
command.
In addition, the library has the ability to understand and interpret core dumps
generated by Linux kernel and can provide a subset of its functionality on such
core files, provided the original binary is also present.
.Pp
Not all functions in the
.Nm
library are valid for core files.
In general, none of the commands which manipulate the current state of a process
or thread or that try to force system calls on a victim process will work.
Furthermore several of the information and iteration interfaces are limited
based on the data that is available in the core file.
For example, if the core file is of a process that omits the frame pointer, the
ability to iterate the stack will be limited.
.Pp
Use the
.Fn Pgrab_core
or
.Fn Pfgrab_core
function to open a core file.
Use the
.Fn Pgrab_file
function to open an ELF object file.
This is useful for obtaining information stored in ELF headers and
sections.
.Ss Debug Information
Many of the operations in the library rely on debug information being
present in a process and its associated libraries.
The library leverages symbol table information, CTF data
.Pf ( Xr CTF 4 )
sections, and frame unwinding information based on the use of an ABI
defined frame pointer, eg.
.Sy %ebp
and
.Sy %rbp
on x86 systems.
.Pp
```

```
125 Some software providers strip programs of this information or build
126 their executables such that the information will not be present in a
127 core dump.
128 To deal with this fact, the library is able to consume information that is not
129 present in the core file or the running process.
130 It can both consume it from the underlying executable and it also supports
131 finding it from related ELF objects that are linked to it via the
132 .Sy .gnu_debuglink
133 and the
134 .Sy .note.gnu.build-id
135 ELF sections.
136 .Ss Iteration Interfaces
137 The
138 .Nm
139 library provides the ability to iterate over the following aspects of a
140 process or core file:
141 .Bl -bullet -offset indent
142 .It
143 Active threads
144 .It
145 Active and zombie threads
146 .It
147 All non-system processes
148 .It
149 All process mappings
150 .It
151 All objects in a process
152 .It
153 The environment
154 .It
155 The symbol table
156 .It
157 Stack frames
158 .It
159 File Descriptors
160 .El
161 .Ss System Call Injection
162 The
163 .Nm
164 library allows the caller to force system calls to be executed in the
165 context of the running process.
166 This can be used both as a tool for introspection, allowing one to get
167 information outside its current context as well as performing modifications to a
168 process.
169 .Pp
170 These functions run in the context of the calling process.
171 This is often an easier way of getting non-exported information about a
172 process from the system.
173 For example, the
174 .Xr pfiles 1
175 command uses this interface to get more detailed information about a
176 process's open file descriptors, which it would not have access to
177 otherwise.
178 .Sh INTERFACES
179 The shared object
180 .Sy libproc.so.1
181 provides the public interfaces defined below.
182 See
183 .Xr Intro 3
184 for additional information on shared object interfaces.
185 Functions are organized into categories that describe their purpose.
186 Individual functions are documented in their own manual pages.
187 .Ss Creation, Grabbing, and Releasing
188 The following routines are related to creating library handles,
189 grabbing cores, processes, and threads, and releasing those resources.
190 .Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
```



```
191 .It Sy Lfree Ta Sy Lgrab
192 .It Sy Lgrab_error Ta Sy Pcreate
193 .It Sy Pcreate_agent Ta Sy Pcreate_callback
194 .It Sy Pcreate_error Ta Sy Pdestroy_agent
195 .It Sy Pfgrab_core Ta Sy Pfree
196 .It Sy Pgrab Ta Sy Pgrab_core
197 .It Sy Pgrab_error Ta Sy Pgrab_file
198 .It Sy Pgrab_ops Ta Sy Prelease
199 .It Sy Preopen Ta Sy Pxcreate
200 .El
201 .Ss Process interrogation and manipulation
202 The following routines obtain information about a process and allow
203 manipulation of the process itself.
204 .Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
205 .It Sy Paddr_to_ctf Ta Sy Paddr_to_loadobj
206 .It Sy Paddr_to_map Ta Sy Paddr_to_text_map
207 .It Sy Pasfd Ta Sy Pclearfault
208 .It Sy Pclearsig Ta Sy Pcontent
209 .It Sy Pcred Ta Sy Pctlfd
210 .It Sy Pdelbkpt Ta Sy Pdelwapt
211 .It Sy Pdstop Ta Sy Pexecname
212 .It Sy Pfault Ta Sy Pfgcore
213 .It Sy Pgcore Ta Sy Pgetareg
214 .It Sy Pgetauxval Ta Sy Pgetauxvec
215 .It Sy Pgetenv Ta Sy Pisprocdir
216 .It Sy Pissyscall_prev Ta Sy Plmid
217 .It Sy Plmid_to_loadobj Ta Sy Plmid_to_map
218 .It Sy Plookup_by_addr Ta Sy Plookup_by_name
219 .It Sy Plwp_alt_stack Ta Sy Plwp_getfpregs
220 .It Sy Plwp_getname Ta Sy Plwp_getpsinfo
221 .It Sy Plwp_getregs Ta Sy Plwp_getspymaster
222 .It Sy Plwp_main_stack Ta Sy Plwp_setfpregs
223 .It Sy Plwp_setregs Ta Sy Plwp_stack
224 .It Sy Pname_to_ctf Ta Sy Pname_to_loadobj
225 .It Sy Pname_to_map Ta Sy Pobjname
226 .It Sy Pobjname_resolved Ta Sy Pplatform
227 .It Sy Ppltdest Ta Sy Ppriv
228 .It Sy Ppsinfo Ta Sy Pputareg
229 .It Sy Prd_agent Ta Sy Pread
230 .It Sy Pread_string Ta Sy Preset_maps
231 .It Sy Psetbkpt Ta Sy Psecflags
232 .It Sy Psetcred Ta Sy Psetfault
233 .It Sy Psetflags Ta Sy Psetpriv
234 .It Sy Psetrun Ta Sy Psetsignal
235 .It Sy Psetsysentry Ta Sy Psetsysexit
236 .It Sy Psetwapt Ta Sy Psetzoneid
237 .It Sy Psignal Ta Sy Pstate
238 .It Sy Pstatus Ta Sy Pstop
239 .It Sy Pstopstatus Ta Sy Psync
240 .It Sy Psysentry Ta Sy Psysexit
241 .It Sy Puname Ta Sy Punsetflags
242 .It Sy Pupdate_maps Ta Sy Pupdate_syms
243 .It Sy Pwait Ta Sy Pwrite
244 .It Sy Pxecbkpt Ta Sy Pxecwapt
245 .It Sy Pxlookup_by_addr Ta Sy Pxlookup_by_addr_resolved
246 .It Sy Pxlookup_by_name Ta Sy Pzonename
247 .It Sy Pzonepath Ta Sy Pzoneroot Ta
220 .It Sy Plwp_getpsinfo Ta Sy Plwp_getregs
221 .It Sy Plwp_getspymaster Ta Sy Plwp_main_stack
222 .It Sy Plwp_setfpregs Ta Sy Plwp_setregs
223 .It Sy Plwp_stack Ta Sy Pname_to_ctf
224 .It Sy Pname_to_loadobj Ta Sy Pname_to_map
225 .It Sy Pobjname Ta Sy Pobjname_resolved
226 .It Sy Pplatform Ta Sy Ppltdest
227 .It Sy Ppriv Ta Sy Ppsinfo
228 .It Sy Pputareg Ta Sy Prd_agent
```

```
.It Sy Pread Ta Sy Pread_string
.It Sy Preset_maps Ta Sy Psetbkpt
.It Sy Psecflags Ta Sy Psetcred
.It Sy Psetfault Ta Sy Psetflags
.It Sy Psetpriv Ta Sy Psetrun
.It Sy Psetsignal Ta Sy Psetsysentry
.It Sy Psetsysexit Ta Sy Psetwapt
.It Sy Psetzoneid Ta Sy Psignal
.It Sy Pstate Ta Sy Pstatus
.It Sy Pstop Ta Sy Pstopstatus
.It Sy Psync Ta Sy Psysentry
.It Sy Psysexit Ta Sy Puname
.It Sy Punsetflags Ta Sy Pupdate_maps
.It Sy Pupdate_syms Ta Sy Pwait
.It Sy Pwrite Ta Sy Pxecbkpt
.It Sy Pxecwapt Ta Sy Pxlookup_by_addr
.It Sy Pxlookup_by_addr_resolved Ta Sy Pxlookup_by_name
.It Sy Pzonename Ta Sy Pzonepath
.It Sy Pzoneroot Ta
.El
.Ss Thread interrogation and manipulation
The following routines obtain information about a thread and allow
manipulation of the thread itself.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Lalt_stack Ta Sy Lclearfault
.It Sy Lclearsig Ta Sy Lctlfd
.It Sy Ldstop Ta Sy Lgetareg
.It Sy Lmain_stack Ta Sy Lprochandle
.It Sy Lpsinfo Ta Sy Lputareg
.It Sy Lsetrun Ta Sy Lstack
.It Sy Lstate Ta Sy Lstatus
.It Sy Lstop Ta Sy Lsync
.It Sy Lwait Ta Sy Lxecbkpt
.It Sy Lxecwapt Ta ""
.El
.Ss System Call Injection
The following routines are used to inject specific system calls and have
them run in the context of a process.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy pr_access Ta Sy pr_close
.It Sy pr_creat Ta Sy pr_door_info
.It Sy pr_exit Ta Sy pr_fcntl
.It Sy pr_fstat Ta Sy pr_fstat64
.It Sy pr_fstatvfs Ta Sy pr_getitimer
.It Sy pr_getpeername Ta Sy pr_getpeerucred
.It Sy pr_getprojid Ta Sy pr_getrctl
.It Sy pr_getrlimit Ta Sy pr_getrlimit64
.It Sy pr_getsockname Ta Sy pr_getsockopt
.It Sy pr_gettaskid Ta Sy pr_getzoneid
.It Sy pr_ioctl Ta Sy pr_link
.It Sy pr_llseek Ta Sy pr_lseek
.It Sy pr_lstat Ta Sy pr_lstat64
.It Sy pr_memcntl Ta Sy pr_meminfo
.It Sy pr_mmap Ta Sy pr_munmap
.It Sy pr_open Ta Sy pr_processor_bind
.It Sy pr_rename Ta Sy pr_setitimer
.It Sy pr_setrctl Ta Sy pr_setrlimit
.It Sy pr_setrlimit64 Ta Sy pr_settaskid
.It Sy pr_sigaction Ta Sy pr_stat
.It Sy pr_stat64 Ta Sy pr_statvfs
.It Sy pr_unlink Ta Sy pr_waitid
.El
.Ss Iteration routines
These routines are used to iterate over the contents of a process.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Penv_iter Ta Sy Plwp_iter
```



```
.It Sy Plwp_iter_all Ta Sy Pmapping_iter
.It Sy Pmapping_iter_resolved Ta Sy Pobject_iter
.It Sy Pobject_iter_resolved Ta Sy Pstack_iter
.It Sy Psymbol_iter Ta Sy Psymbol_iter_by_addr
.It Sy Psymbol_iter_by_lmid Ta Sy Psymbol_iter_by_name
.It Sy Pxsymbol_iter Ta Sy Pfdinfo_iter
.El
.Ss Utility routines
The following routines are utilities that are useful to consumers of the
library.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Perror_printf Ta Sy proc_arg_grab
.It Sy proc_arg_psinfo Ta Sy proc_arg_xgrab
.It Sy proc_arg_xpsinfo Ta Sy proc_content2str
.It Sy proc_finistdio Ta Sy proc_fltname
.It Sy proc_fltset2str Ta Sy proc_flushstdio
.It Sy proc_get_auxv Ta Sy proc_get_cred
.It Sy proc_get_priv Ta Sy proc_get_psinfo
.It Sy proc_get_status Ta Sy proc_initstdio
.It Sy proc_lwp_in_set Ta Sy proc_lwp_range_valid
.It Sy proc_signame Ta Sy proc_sigset2str
.It Sy proc_str2content Ta Sy proc_str2flt
.It Sy proc_str2fltset Ta Sy proc_str2sig
.It Sy proc_str2sigset Ta Sy proc_str2sys
.It Sy proc_str2sysset Ta Sy proc_sysname
.It Sy proc_sysset2str Ta Sy proc_unctrl_psinfo
.It Sy proc_walk Ta ""
.El
.Ss x86 Specific Routines
The following routines are specific to the x86, 32-bit and 64-bit,
versions of the
.Nm
library.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Pldt Ta Sy proc_get_ldt
.El
.Ss SPARC specific Routines
The following functions are specific to the SPARC, 32-bit and 64-bit,
versions of the
.Nm
library.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Plwp_getgwindows Ta Sy Plwp_getxregs
.It Sy Plwp_setxregs Ta Sy ""
.El
.Pp
The following functions are specific to the 64-bit SPARC version of the
.Nm
library.
.Bl -column -offset indent ".Sy Pmapping_iter_resolved" ".Sy Psymbol_iter_by_add
.It Sy Plwp_getasrs Ta Sy Plwp_setasrs
.El
.Sh PROCESS STATES
Every process handle that exists in
.Nm
has a state.
In some cases, such as for core files, these states are static.
In other cases, such as handles that correspond to a running process or a
created process, these states are dynamic and change based on actions taken in
the library.
The state can be obtained with the
.Xr Pstate 3PROC
function.
.Pp
The various states are:
.Bl -tag -width Dv -offset indent
```

```
.It Dv PS_RUN
An actively running process.
This may be a process that was obtained by creating it with functions such as
.Xr Pcreate 3PROC
or by grabbing an existing process such as
.Xr Pgrab 3PROC .
.It Dv PS_STOP
An active process that is no longer executing.
A process may stop for many reasons such as an explicit stop request (through
.Xr pstop 1
for example) or if a tracing event is hit.
.Pp
The reason a process is stopped may be obtained through the thread's
.Sy lwpstatus_t
structure read directly from /proc or obtained through the
.Xr Lstatus 3PROC
function.
.It Dv PS_LOST
Control over the process has been lost.
This may happen when the process executes a new image requiring a different set
of privileges.
To resume control call
.Xr Preopen 3PROC .
For more information on losing control of a process, see
.Xr proc 4 .
.It DV PS_UNDEAD
A zombie process.
It has terminated, but it has not been cleaned up yet by its parent.
For more on the conditions of becoming a zombie, see
.Xr exec 2 .
.It DV_PS_DEAD
Processes in this state are always core files.
See the earlier section
.Sx Core Files
for more information on working with core files.
.It Dv PS_IDLE
A process that has never been run.
This is always the case for handles that refer to files as the files cannot be
executed.
Those process handles are obtained through calling
.Xr Pgrab_file 3PROC .
.El
.Pp
Many functions relating to tracing processes, for example
.Xr Psignal 3PROC ,
.Xr Psetsignal 3PROC ,
.Xr Psetfault 3PROC ,
.Xr Psysentry 3PROC ,
and others, mention that they only act upon
.Em Active Processes .
This specifically refers to processes whose state are in
.Dv PS_RUN
and
.Dv PS_STOP .
Process handles in the other states have no notion of settable tracing
flags, though core files
.Pf ( type Dv PS_DEAD )
may have a read-only snapshot of their tracing settings available.
.Sh TYPES
The
.Nm
library uses many types that come from the /proc file system
.Pf ( Xr proc 4 )
and the ELF format
.Pf ( Xr elf 3ELF ) .
However, it also defines the following types:
```



```
.Pp
.Sy struct ps_prochandle
.Pp
The
.Sy struct ps_prochandle
is an opaque handle to the library and the core element of control for a
process.
Consumers obtain pointers to a handle through the use of the
.Fn Pcreate ,
.Fn Pgrab ,
and related functions.
When a caller is done with a handle, then it should call one of the
.Fn Pfree
and
.Fn Prelease
functions to relinquish the handle, release associated resources, and
potentially set the process to run again.
.Pp
.Sy struct ps_lwphandle
.Pp
The
.Sy struct ps_lwphandle
is analogous to the
.Sy struct ps_prochandle ,
but it represents the control of an individual thread, rather than a
process.
Consumers obtain pointers to a handle through the
.Fn Lgrab
function and relinquish it with the
.Fn Lfree
function.
.Pp
.Sy core_content_t
.Pp
The
.Sy core_content_t
is a value which describes the various content types of core files.
These are used in functions such as
.Xr Pcontent 3PROC
and
.Xr Pgcore 3PROC
to describe and control the types of content that get included.
Various content types may be included together through a bitwise-inclusive-OR.
The default system core contents are controlled with the
.Xr coreadm 1M
tool.
The following table lists the current set of core contents in the system, though
the set may increase over time.
The string after the macro is the human readable string that corresponds with
the constant and is used by
.Xr coreadm 1M ,
.Xr proc_content2str 3PROC ,
and
.Xr proc_str2content 3PROC .
.Bl -tag -offset indent -width indent
.It Dv CC_CONTENT_STACK ("stack")
The contents include the process stack.
Note, this only covers the main thread's stack.
The stack of other threads is covered by
.Dv CC_CONTENT_ANON .
.It Dv CC_CONTENT_HEAP ("heap")
The contents include the process heap.
.It Dv CC_CONTENT_SHFILE ("shfile")
The contents include shared mappings that are backed by files (e.g.
mapped through
.Xr mmap 2
```

```
493 with the
494 .Dv MAP_SHARED
495 flag).
496 .It Dv CC_CONTENT_SHANNON ("shannon")
497 The contents include shared mappings that are backed by anonymous memory
498 (e.g. mapped through
499 .Xr mmap 2
500 with the
501 .Dv MAP_SHARED
502 and
503 .Dv MAP_ANON
504 flags).
505 .It Dv CC_CONTENT_RODATA ("rodata")
506 The contents include private read-only file mappings, such as shared
507 library text.
508 .It Dv CC_CONTENT_ANON ("anon")
509 The contents include private anonymous mappings.
510 This includes the stacks of threads which are not the main thread.
511 .It Dv CC_CONTENT_SHM ("shm")
512 The contents include system V shared memory.
513 .It Dv CC_CONTENT_ISM ("ism")
514 The contents include ISM (intimate shared memory) mappings.
515 .It Dv CC_CONTENT_DISM ("dism")
516 The contents include DISM (dynamic shared memory) mappings.
517 .It Dv CC_CONTENT_CTF ("ctf")
518 The contents include
519 .Xr ctf 4
520 (Compact C Type Format) information.
521 Note, not all objects in the process may have CTF information available.
522 .It Dv CC_CONTENT_SYMTAB ("symtab")
523 The contents include the symbol table.
524 Note, not all objects in the process may have a symbol table available.
525 .It Dv CC_CONTENT_ALL ("all")
526 This value indicates that all of the above content values are present.
527 Note that additional values may be added in the future, in which case
528 the value of the symbol will be updated to include them.
529 Comparisons with
530 .Dv CC_CONTENT_ALL
531 should validate all the expected bits are set by an expression such as
532 .Li (c & CC_CONTENT_ALL) == CC_CONTENT_ALL .
533 .It Dv CC_CONTENT_NONE ("none")
534 This value indicates that there is no content present.
535 .It Dv CC_CONTENT_DEFAULT ("default")
536 The content includes the following set of default values:
537 .Dv CC_CONTENT_STACK ,
538 .Dv CC_CONTENT_HEAP ,
539 .Dv CC_CONTENT_ISM ,
540 .Dv CC_CONTENT_DISM ,
541 .Dv CC_CONTENT_SHM ,
542 .Dv CC_CONTENT_SHANON ,
543 .Dv CC_CONTENT_TEXT ,
544 .Dv CC_CONTENT_DATA ,
545 .Dv CC_CONTENT_RODATA ,
546 .Dv CC_CONTENT_ANON ,
547 .Dv CC_CONTENT_CTF ,
548 and
549 .Dv CC_CONTENT_SYMTAB.
550 Note that the default may change.
551 Comparisons with CC_CONTENT_DEFAULT should validate that all of the expected
552 bits are set with an expression such as
553 .Li (c\ &\ CC_CONTENT_DEFAULT)\ ==\ CC_CONTENT_DEFAULT.
554 .It Dv CC_CONTENT_INVALID
555 This indicates that the contents are invalid.
556 .El
557 .Pp
558 .Sy prfdinfo_t
```



```
559 .Pp
560 The
561 .Sy prfdinfo_t
562 structure is used with the
563 .Fn Pfdinfo_iter
564 function which describes information about a file descriptor.
565 The structure is defined as follows:
566 .Bd -literal
567 typedef struct prfdinfo {
568         int             pr_fd;
569         mode_t          pr_mode;
570         uid_t           pr_uid;
571         gid_t           pr_gid;
572         major_t         pr_major;       /* think stat.st_dev */
573         minor_t         pr_minor;
574         major_t         pr_rmajor;      /* think stat.st_rdev */
575         minor_t         pr_rminor;
576         ino64_t         pr_ino;
577         off64_t         pr_offset;
578         off64_t         pr_size;
579         int             pr_fileflags;   /* fcntl(F_GETXFL), etc */
580         int             pr_fdflags;     /* fcntl(F_GETFD), etc. */
581         char            pr_path[MAXPATHLEN];
582 } prfdinfo_t;
______unchanged_portion_omitted_
```

```
**********************************************************
   11441 Mon Oct 15 13:25:51 2018
new/usr/src/man/man3proc/Makefile
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  15 # Copyright 2018 Joyent, Inc.
  15 # Copyright 2015 Joyent, Inc.
  16 #

  18 include         $(SRC)/Makefile.master

  20 MANSECT=        3proc

  22 MANFILES=                                        \
  23                 Lctlfd.3proc                    \
  24                 Lfree.3proc                     \
  25                 Lgrab_error.3proc               \
  26                 Lgrab.3proc                     \
  27                 Lprochandle.3proc               \
  28                 Lpsinfo.3proc                   \
  29                 Lstate.3proc                    \
  30                 Lstatus.3proc                   \
  31                 Paddr_to_ctf.3proc              \
  32                 Paddr_to_loadobj.3proc          \
  33                 Paddr_to_map.3proc              \
  34                 Pasfd.3proc                     \
  35                 Pclearfault.3proc               \
  36                 Pclearsig.3proc                 \
  37                 Pcontent.3proc                  \
  38                 Pcreate_agent.3proc             \
  39                 Pcreate_error.3proc             \
  40                 Pcreate.3proc                   \
  41                 Pcred.3proc                     \
  42                 Pctlfd.3proc                    \
  43                 Pdelbkpt.3proc                  \
  44                 Pdelwapt.3proc                  \
  45                 Pdestroy_agent.3proc            \
  46                 Penv_iter.3proc                 \
  47                 Perror_printf.3proc             \
  48                 Pexecname.3proc                 \
  49                 Pfault.3proc                    \
  50                 Pfdinfo_iter.3proc              \
  51                 Pgcore.3proc                    \
  52                 Pgetareg.3proc                  \
  53                 Pgetauxval.3proc                \
  54                 Pgetauxvec.3proc                \
  55                 Pgetenv.3proc                   \
  56                 Pgrab_core.3proc                \
  57                 Pgrab_error.3proc               \
  58                 Pgrab_file.3proc                \
  59                 Pgrab.3proc                     \
```



```
  60                 Pisprocdir.3proc                \
  61                 Pissyscall.3proc                \
  62                 Pldt.3proc                      \
  63                 Plmid.3proc                     \
  64                 Plookup_by_addr.3proc           \
  65                 Plwp_getasrs.3proc              \
  66                 Plwp_getgwindows.3proc          \
  67                 Plwp_getname.3proc              \
  68                 Plwp_getpsinfo.3proc            \
  69                 Plwp_getregs.3proc              \
  70                 Plwp_getspymaster.3proc         \
  71                 Plwp_getxregs.3proc             \
  72                 Plwp_iter.3proc                 \
  73                 Plwp_stack.3proc                \
  74                 Pmapping_iter.3proc             \
  75                 Pobjname.3proc                  \
  76                 Pplatform.3proc                 \
  77                 Ppltdest.3proc                  \
  78                 Ppriv.3proc                     \
  79                 Ppsinfo.3proc                   \
  80                 pr_access.3proc                 \
  81                 pr_close.3proc                  \
  82                 pr_creat.3proc                  \
  83                 pr_door_info.3proc              \
  84                 pr_exit.3proc                   \
  85                 pr_fcntl.3proc                  \
  86                 pr_fstatvfs.3proc               \
  87                 pr_getitimer.3proc              \
  88                 pr_getpeername.3proc            \
  89                 pr_getpeerucred.3proc           \
  90                 pr_getprojid.3proc              \
  91                 pr_getrctl.3proc                \
  92                 pr_getrlimit.3proc              \
  93                 pr_getsockname.3proc            \
  94                 pr_getsockopt.3proc             \
  95                 pr_gettaskid.3proc              \
  96                 pr_getzoneid.3proc              \
  97                 pr_ioctl.3proc                  \
  98                 pr_link.3proc                   \
  99                 pr_llseek.3proc                 \
 100                 pr_lseek.3proc                  \
 101                 pr_memcntl.3proc                \
 102                 pr_meminfo.3proc                \
 103                 pr_mmap.3proc                   \
 104                 pr_munmap.3proc                 \
 105                 pr_open.3proc                   \
 106                 pr_processor_bind.3proc         \
 107                 pr_rename.3proc                 \
 108                 pr_setitimer.3proc              \
 109                 pr_setrctl.3proc                \
 110                 pr_setrlimit.3proc              \
 111                 pr_settaskid.3proc              \
 112                 pr_sigaction.3proc              \
 113                 pr_stat.3proc                   \
 114                 pr_statvfs.3proc                \
 115                 pr_unlink.3proc                 \
 116                 pr_waitid.3proc                 \
 117                 Prd_agent.3proc                 \
 118                 Pread.3proc                     \
 119                 Prelease.3proc                  \
 120                 Preopen.3proc                   \
 121                 Preset_maps.3proc               \
 122                 proc_arg_grab.3proc             \
 123                 proc_arg_psinfo.3proc           \
 124                 proc_content2str.3proc          \
 125                 proc_fltname.3proc              \
```

```
126                 proc_fltset2str.3proc           \
127                 proc_get_auxv.3proc             \
128                 proc_get_cred.3proc             \
129                 proc_get_priv.3proc             \
130                 proc_get_psinfo.3proc           \
131                 proc_get_status.3proc           \
132                 proc_initstdio.3proc            \
133                 proc_lwp_in_set.3proc           \
134                 proc_service.3proc              \
135                 proc_str2flt.3proc              \
136                 proc_str2fltset.3proc           \
137                 proc_unctrl_psinfo.3proc        \
138                 proc_walk.3proc                 \
139                 Psecflags.3proc                 \
140                 Psetbkpt.3proc                  \
141                 Psetcred.3proc                  \
142                 Psetfault.3proc                 \
143                 Psetflags.3proc                 \
144                 Psetpriv.3proc                  \
145                 Psetrun.3proc                   \
146                 Psetsignal.3proc                \
147                 Psetsysentry.3proc              \
148                 Psetwapt.3proc                  \
149                 Psetzoneid.3proc                \
150                 Psignal.3proc                   \
151                 Pstack_iter.3proc               \
152                 Pstate.3proc                    \
153                 Pstatus.3proc                   \
154                 Pstopstatus.3proc               \
155                 Psymbol_iter.3proc              \
156                 Psync.3proc                     \
157                 Psysentry.3proc                 \
158                 Puname.3proc                    \
159                 Pupdate_maps.3proc              \
160                 Pupdate_syms.3proc              \
161                 Pwrite.3proc                    \
162                 Pxecbkpt.3proc                  \
163                 Pzonename.3proc                 \
164                 ps_lgetregs.3proc               \
165                 ps_pglobal_lookup.3proc         \
166                 ps_pread.3proc                  \
167                 ps_pstop.3proc


170 MANLINKS=                                       \
171                 Lalt_stack.3proc                \
172                 Lclearfault.3proc               \
173                 Lclearsig.3proc                 \
174                 Ldstop.3proc                    \
175                 Lgetareg.3proc                  \
176                 Lmain_stack.3proc               \
177                 Lputareg.3proc                  \
178                 Lsetrun.3proc                   \
179                 Lstack.3proc                    \
180                 Lstop.3proc                     \
181                 Lsync.3proc                     \
182                 Lwait.3proc                     \
183                 Lxecbkpt.3proc                  \
184                 Lxecwapt.3proc                  \
185                 Paddr_to_text_map.3proc         \
186                 Pcreate_callback.3proc          \
187                 Pdstop.3proc                    \
188                 Pfgcore.3proc                   \
189                 Pfgrab_core.3proc               \
190                 Pfree.3proc                     \
191                 Pissyscall_prev.3proc           \
```



```
192                 Plmid_to_ctf.3proc              \
193                 Plmid_to_loadobj.3proc          \
194                 Plmid_to_map.3proc              \
195                 Plookup_by_name.3proc           \
196                 Plwp_alt_stack.3proc            \
197                 Plwp_getfpregs.3proc            \
198                 Plwp_iter_all.3proc             \
199                 Plwp_main_stack.3proc           \
200                 Plwp_setasrs.3proc              \
201                 Plwp_setfpregs.3proc            \
202                 Plwp_setregs.3proc              \
203                 Plwp_setxregs.3proc             \
204                 Pmapping_iter_resolved.3proc    \
205                 Pname_to_ctf.3proc              \
206                 Pname_to_loadobj.3proc          \
207                 Pname_to_map.3proc              \
208                 Pobject_iter_resolved.3proc     \
209                 Pobject_iter.3proc              \
210                 Pobjname_resolved.3proc         \
211                 Ppriv_free.3proc                \
212                 Pputareg.3proc                  \
213                 pr_fstat.3proc                  \
214                 pr_fstat64.3proc                \
215                 pr_getrlimit64.3proc            \
216                 pr_lstat.3proc                  \
217                 pr_lstat64.3proc                \
218                 pr_setrlimit64.3proc            \
219                 pr_stat64.3proc                 \
220                 Pread_string.3proc              \
221                 proc_arg_xgrab.3proc            \
222                 proc_arg_xpsinfo.3proc          \
223                 proc_finistdio.3proc            \
224                 proc_flushstdio.3proc           \
225                 proc_free_priv.3proc            \
226                 proc_get_ldt.3proc              \
227                 proc_lwp_range_valid.3proc      \
228                 proc_signame.3proc              \
229                 proc_sigset2str.3proc           \
230                 proc_str2content.3proc          \
231                 proc_str2sig.3proc              \
232                 proc_str2sigset.3proc           \
233                 proc_str2sys.3proc              \
234                 proc_str2sysset.3proc           \
235                 proc_sysname.3proc              \
236                 proc_sysset2str.3proc           \
237                 ps_kill.3proc                   \
238                 ps_lcontinue.3proc              \
239                 ps_lgetfpregs.3proc             \
240                 ps_lgetxregs.3proc              \
241                 ps_lgetxregsize.3proc           \
242                 ps_lrolltoaddr.3proc            \
243                 ps_lsetfpregs.3proc             \
244                 ps_lsetregs.3proc               \
245                 ps_lsetxregs.3proc              \
246                 ps_lstop.3proc                  \
247                 ps_pcontinue.3proc              \
248                 ps_pdread.3proc                 \
249                 ps_pdwrite.3proc                \
250                 ps_pglobal_sym.3proc            \
251                 ps_ptread.3proc                 \
252                 ps_ptwrite.3proc                \
253                 ps_pwrite.3proc                 \
254                 Psetsysexit.3proc               \
255                 Pstop.3proc                     \
256                 Psymbol_iter_by_addr.3proc      \
257                 Psymbol_iter_by_lmid.3proc      \
```

```
258                 Psymbol_iter_by_name.3proc      \
259                 Psysexit.3proc                  \
260                 Punsetflags.3proc               \
261                 Pwait.3proc                     \
262                 Pxcreate.3proc                  \
263                 Pxecwapt.3proc                  \
264                 Pxlookup_by_addr_resolved.3proc \
265                 Pxlookup_by_addr.3proc          \
266                 Pxlookup_by_name.3proc          \
267                 Pxsymbol_iter.3proc             \
268                 Pzonepath.3proc                 \
269                 Pzoneroot.3proc


272 ps_lgetfpregs.3proc            := LINKSRC = ps_lgetregs.3proc
273 ps_lgetxregs.3proc             := LINKSRC = ps_lgetregs.3proc
274 ps_lgetxregsize.3proc         := LINKSRC = ps_lgetregs.3proc
275 ps_lsetfpregs.3proc            := LINKSRC = ps_lgetregs.3proc
276 ps_lsetregs.3proc              := LINKSRC = ps_lgetregs.3proc
277 ps_lsetxregs.3proc             := LINKSRC = ps_lgetregs.3proc

279 ps_pglobal_sym.3proc          := LINKSRC = ps_pglobal_lookup.3proc

281 ps_pdread.3proc               := LINKSRC = ps_pread.3proc
282 ps_pdwrite.3proc              := LINKSRC = ps_pread.3proc
283 ps_ptread.3proc               := LINKSRC = ps_pread.3proc
284 ps_ptwrite.3proc              := LINKSRC = ps_pread.3proc
285 ps_pwrite.3proc               := LINKSRC = ps_pread.3proc

287 ps_kill.3proc                 := LINKSRC = ps_pstop.3proc
288 ps_lcontinue.3proc            := LINKSRC = ps_pstop.3proc
289 ps_lrolltoaddr.3proc         := LINKSRC = ps_pstop.3proc
290 ps_lstop.3proc                := LINKSRC = ps_pstop.3proc
291 ps_pcontinue.3proc            := LINKSRC = ps_pstop.3proc


294 Pxcreate.3proc               := LINKSRC = Pcreate.3proc
295 Pcreate_callback.3proc       := LINKSRC = Pcreate.3proc

297 Pfgrab_core.3proc            := LINKSRC = Pgrab_core.3proc

299 Pfree.3proc                  := LINKSRC = Prelease.3proc

301 Plwp_iter_all.3proc          := LINKSRC = Plwp_iter.3proc

303 Pmapping_iter_resolved.3proc := LINKSRC = Pmapping_iter.3proc
304 Pobject_iter.3proc           := LINKSRC = Pmapping_iter.3proc
305 Pobject_iter_resolved.3proc  := LINKSRC = Pmapping_iter.3proc

307 Psymbol_iter_by_addr.3proc   := LINKSRC = Psymbol_iter.3proc
308 Psymbol_iter_by_lmid.3proc   := LINKSRC = Psymbol_iter.3proc
309 Psymbol_iter_by_name.3proc   := LINKSRC = Psymbol_iter.3proc
310 Pxsymbol_iter.3proc          := LINKSRC = Psymbol_iter.3proc

312 Plmid_to_ctf.3proc           := LINKSRC = Paddr_to_ctf.3proc
313 Pname_to_ctf.3proc           := LINKSRC = Paddr_to_ctf.3proc

315 Plmid_to_loadobj.3proc      := LINKSRC = Paddr_to_loadobj.3proc
316 Pname_to_loadobj.3proc      := LINKSRC = Paddr_to_loadobj.3proc

318 Paddr_to_text_map.3proc     := LINKSRC = Paddr_to_map.3proc
319 Plmid_to_map.3proc          := LINKSRC = Paddr_to_map.3proc
320 Pname_to_map.3proc          := LINKSRC = Paddr_to_map.3proc

322 Pdstop.3proc                := LINKSRC = Pstopstatus.3proc
323 Pstop.3proc                 := LINKSRC = Pstopstatus.3proc
```



```
324 Pwait.3proc                 := LINKSRC = Pstopstatus.3proc
325 Ldstop.3proc                := LINKSRC = Pstopstatus.3proc
326 Lstop.3proc                 := LINKSRC = Pstopstatus.3proc
327 Lwait.3proc                 := LINKSRC = Pstopstatus.3proc

329 Lsync.3proc                 := LINKSRC = Psync.3proc

331 Pfgcore.3proc               := LINKSRC = Pgcore.3proc

333 Pputareg.3proc              := LINKSRC = Pgetareg.3proc
334 Lgetareg.3proc              := LINKSRC = Pgetareg.3proc
335 Lputareg.3proc              := LINKSRC = Pgetareg.3proc

337 Pissyscall_prev.3proc       := LINKSRC = Pissyscall.3proc

339 Pxlookup_by_addr.3proc      := LINKSRC = Plookup_by_addr.3proc
340 Pxlookup_by_addr_resolved.3proc := LINKSRC = Plookup_by_addr.3proc
341 Plookup_by_name.3proc       := LINKSRC = Plookup_by_addr.3proc
342 Pxlookup_by_name.3proc      := LINKSRC = Plookup_by_addr.3proc

344 Plwp_setregs.3proc          := LINKSRC = Plwp_getregs.3proc
345 Plwp_getfpregs.3proc        := LINKSRC = Plwp_getregs.3proc
346 Plwp_setfpregs.3proc        := LINKSRC = Plwp_getregs.3proc

348 Plwp_alt_stack.3proc        := LINKSRC = Plwp_stack.3proc
349 Plwp_main_stack.3proc       := LINKSRC = Plwp_stack.3proc
350 Lalt_stack.3proc            := LINKSRC = Plwp_stack.3proc
351 Lmain_stack.3proc           := LINKSRC = Plwp_stack.3proc
352 Lstack.3proc                := LINKSRC = Plwp_stack.3proc

354 Pobjname_resolved.3proc     := LINKSRC = Pobjname.3proc

356 Ppriv_free.3proc            := LINKSRC = Ppriv.3proc

358 Pread_string.3proc          := LINKSRC = Pread.3proc

360 Punsetflags.3proc          := LINKSRC = Psetflags.3proc

362 Psetsysexit.3proc          := LINKSRC = Psetsysentry.3proc

364 Psysexit.3proc             := LINKSRC = Psysentry.3proc

366 Pxecwapt.3proc             := LINKSRC = Pxecbkpt.3proc
367 Lxecbkpt.3proc             := LINKSRC = Pxecbkpt.3proc
368 Lxecwapt.3proc             := LINKSRC = Pxecbkpt.3proc

370 Lclearfault.3proc          := LINKSRC = Pclearfault.3proc

372 Lclearsig.3proc           := LINKSRC = Pclearsig.3proc

374 Lsetrun.3proc             := LINKSRC = Psetrun.3proc

376 Pzonepath.3proc           := LINKSRC = Pzonename.3proc
377 Pzoneroot.3proc           := LINKSRC = Pzonename.3proc

379 pr_fstat.3proc            := LINKSRC = pr_stat.3proc
380 pr_fstat64.3proc          := LINKSRC = pr_stat.3proc
381 pr_lstat.3proc            := LINKSRC = pr_stat.3proc
382 pr_lstat64.3proc          := LINKSRC = pr_stat.3proc
383 pr_stat64.3proc           := LINKSRC = pr_stat.3proc

385 pr_getrlimit64.3proc      := LINKSRC = pr_getrlimit.3proc

387 pr_setrlimit64.3proc      := LINKSRC = pr_setrlimit.3proc

389 proc_arg_xgrab.3proc      := LINKSRC = proc_arg_grab.3proc
```

```
391 proc_arg_xpsinfo.3proc          := LINKSRC = proc_arg_psinfo.3proc

393 proc_str2content.3proc          := LINKSRC = proc_content2str.3proc

395 proc_flushstdio.3proc           := LINKSRC = proc_initstdio.3proc
396 proc_finistdio.3proc            := LINKSRC = proc_initstdio.3proc

398 proc_signame.3proc              := LINKSRC = proc_fltname.3proc
399 proc_sysname.3proc              := LINKSRC = proc_fltname.3proc

401 proc_sigset2str.3proc           := LINKSRC = proc_fltset2str.3proc
402 proc_sysset2str.3proc           := LINKSRC = proc_fltset2str.3proc

404 proc_free_priv.3proc            := LINKSRC = proc_get_priv.3proc

406 proc_lwp_range_valid.3proc      := LINKSRC = proc_lwp_in_set.3proc

408 proc_str2sig.3proc              := LINKSRC = proc_str2flt.3proc
409 proc_str2sys.3proc              := LINKSRC = proc_str2flt.3proc

411 proc_str2sigset.3proc           := LINKSRC = proc_str2fltset.3proc
412 proc_str2sysset.3proc           := LINKSRC = proc_str2fltset.3proc

414 proc_get_ldt.3proc              := LINKSRC = Pldt.3proc

416 Plwp_setxregs.3proc             := LINKSRC = Plwp_getxregs.3proc

418 Plwp_setasrs.3proc             := LINKSRC = Plwp_getasrs.3proc

420 .KEEP_STATE:

422 include         $(SRC)/man/Makefile.man

424 install:        $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
**********************************************************
    2787 Mon Oct 15 13:25:54 2018
new/usr/src/man/man3proc/Paddr_to_ctf.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PADDR_TO_CTF 3PROC
.Os
.Sh NAME
.Nm Paddr_to_ctf ,
.Nm Plmid_to_ctf ,
.Nm Pname_to_ctf
.Nd lookup CTF data
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "ctf_file_t *"
.Fo Paddr_to_ctf
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fc
.Ft "ctf_file_t *"
.Fo Plmid_to_ctf
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *name"
.Fc
.Ft "ctf_file_t *"
.Fo Pname_to_ctf
.Fa "struct ps_prochandle *P"
.Fa "const char *name"
.Fc
.Sh DESCRIPTION
The
.Fn Paddr_to_ctf ,
.Fn Plmid_to_ctf ,
and
.Fn Pname_to_ctf
functions lookup CTF (Compact C Type Format) data, for use with
.Sy libctf ,
from the process represented by the handle
.Fa P .
In all cases, the CTF sections of both the running executable and its
shared libraries are searched.
.Pp
The CTF container returned is valid as long as the process handle
.Fa P
is valid.
That is, until a call to
.Xr Prelease 3PROC
```



```
is made.
Further, consumers must not close the CTF container.
.Pp
The
.Fn Paddr_to_ctf
function attempts to find the CTF section, if any, that exists for the
address
.Fa addr .
Note, not all addresses correspond to memory regions that have CTF
data.
For example, if a user creates a region of anonymous memory through the
.Xr mmap 2
function, then it will not have any corresponding CTF information.
.Pp
The
.Fn Pname_to_ctf
function looks up the object named
.Fa name
and returns the corresponding CTF section, if any exists.
Two special values may be used for name.
The macro
.Dv PR_OBJ_EXEC
refers to the executable object itself and the macro
.Dv PR_OBJ_LDSO refers to the object ld.so.1 .
.Pp
The
.Fn Plmid_to_ctf
function is similar to
.Fn Pname_to_ctf .
It allows the passing of a link-map identifier,
.Fa lmid ,
which constricts the search of the object named with
.Fa name
to that link-map.
The special value of
.Dv PR_LMID_EVERY
indicates that every link-map should be searched, which is equivalent
in behavior to the
.Fn Pname_to_ctf
function.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Paddr_to_ctf ,
.Fn Plmid_to_ctf ,
and
.Fn Pname_to_ctf
functions return a pointer to the corresponding CTF container.
Otherwise, if none exists then
.Dv NULL
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr ctf 4
```

```
**********************************************************
    3023 Mon Oct 15 13:25:57 2018
new/usr/src/man/man3proc/Paddr_to_loadobj.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PADDR_TO_LOADOBJ 3PROC
.Os
.Sh NAME
.Nm Paddr_to_loadobj ,
.Nm Plmid_to_loadobj ,
.Nm Pname_to_loadobj
.Nd lookup loaded object information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.In libproc.h
.Ft "const rd_loadobj_t *"
.Fo Paddr_to_loadobj
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fc
.Ft "const rd_loadobj_t *"
.Fo Plmid_to_loadobj
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *name"
.Fc
.Ft "const rd_loadobj_t *"
.Fo Pname_to_loadobj
.Fa "struct ps_prochandle *P"
.Fa "const char *name"
.Fc
.Sh DESCRIPTION
The
.Fn Paddr_to_loadobj ,
.Fn Plmid_to_loadobj ,
and
.Fn Pname_to_loadobj
functions lookup loaded object information from the process handle
.Fa P .
This information is provided by the run-time link-editor,
.Xr ld.so.1 1 ,
and provides information about the loaded object such as the link-map
identifier, the TLS module ID, and the address of various sections.
.Pp
The pointer to the data returned by the library will only be valid for
as long as the handle
.Fa P
is valid.
Any calls to
```



```
.Xr Prelease 3PROC
will invalidate the data.
.Pp
The
.Fn Paddr_to_loadobj
function attempts to find the loaded object information, if any, that exists for
the address
.Fa addr .
Not all address correspond to memory regions that were loaded by the
run-time link-editor.
For example, if a user creates a region of anonymous memory through the
.Xr mmap 2
function, then it will not have any corresponding loaded module.
.Pp
The
.Fn Pname_to_loadobj
function looks up the object named
.Fa name
and returns the corresponding loaded object information.
Two special values may be used for name.
The macro
.Dv PR_OBJ_EXEC
refers to the executable object itself and the macro
.Dv PR_OBJ_LDSO refers to the object ld.so.1 .
.Pp
The
.Fn Plmid_to_loadobj
function is similar to
.Fn Pname_to_loadobj .
It allows the use of a link-map identifier,
.Fa lmid ,
which constricts the search of the object named with
.Fa name
to that link-map.
The special value of
.Dv PR_LMID_EVERY
may be passed to indicate that every link-map should be searched, which
is equivalent in behavior to the
.Fn Pname_to_loadobj
function.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Paddr_to_loadobj ,
.Fn Plmid_to_loadobj ,
and
.Fn Pname_to_loadobj
functions return a pointer to the corresponding loadable object
information.
Otherwise, if none exists then
.Dv NULL
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr librtld_db 3LIB ,
.Xr Prelease 3PROC
```

```
**********************************************************
    3055 Mon Oct 15 13:26:01 2018
new/usr/src/man/man3proc/Paddr_to_map.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PADDR_TO_MAP 3PROC
.Os
.Sh NAME
.Nm Paddr_to_map ,
.Nm Paddr_to_text_map ,
.Nm Plmid_to_map ,
.Nm Pname_to_map
.Nd lookup memory map information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const prmap_t *"
.Fo Paddr_to_map
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fc
.Ft "const prmap_t *"
.Fo Paddr_to_text_map
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fc
.Ft "const prmap_t *"
.Fo Plmid_to_map
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *name"
.Fc
.Ft "const prmap_t *"
.Fo Pname_to_map
.Fa "struct ps_prochandle *P"
.Fa "const char *name"
.Fc
.Sh DESCRIPTION
The
.Fn Paddr_to_map ,
.Fn Paddr_to_text_map ,
.Fn Plmid_to_map ,
and
.Fn Pname_to_map
functions lookup memory map information in the process handle
.Fa P .
The
.Sy prmap_t
structure provides information such as the size, offset, and object of
```



```
the mapping and is defined in
.Xr proc 4 .
.Pp
The pointer to the data returned by the library will only be valid for
as long as the handle
.Fa P
is valid.
Any calls to
.Xr Prelease 3PROC
will invalidate the data.
.Pp
The
.Fn Paddr_to_map
function attempts to find the mapping information corresponding to the
address
.Fa addr .
.Pp
The
.Fn Paddr_to_text_map
function is similar to the
.Fn Paddr_to_map
function; however, it only returns successfully if the specified address
corresponds to a text mapping as identified by the run-time link-editor.
One use of this is to ensure that a mapping is actually a text-mapping
before inserting a breakpoint in it.
.Pp
The
.Fn Pname_to_map
function looks up the object named
.Fa name
and returns the corresponding mapping information.
Two special values may be used for name.
The macro
.Dv PR_OBJ_EXEC
refers to the executable object itself and the macro
.Dv PR_OBJ_LDSO refers to the object ld.so.1 .
.Pp
The
.Fn Plmid_to_map
function is similar to
.Fn Pname_to_map .
It allows passing a link-map identifier,
.Fa lmid ,
which constricts the search of the object named with
.Fa name
to that link-map.
The special value of
.Dv PR_LMID_EVERY
may be passed to indicate that every link-map should be searched, which
is equivalent in behavior to the
.Fn Pname_to_map
function.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Paddr_to_map ,
.Fn Paddr_to_text_map ,
.Fn Plmid_to_map ,
and
.Fn Pname_to_map
functions return a pointer to the corresponding mapping information.
If none exists then
.Dv NULL
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
```

```
126 See
127 .Sy LOCKING
128 in
129 .Xr libproc 3LIB .
130 .Sh SEE ALSO
131 .Xr libproc 3LIB ,
132 .Xr Prelease 3PROC ,
133 .Xr proc 4
```

```
**********************************************************
    1839 Mon Oct 15 13:26:07 2018
new/usr/src/man/man3proc/Pasfd.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PASFD 3PROC
.Os
.Sh NAME
.Nm Pasfd
.Nd obtain the process address space file descriptor
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pasfd
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pasfd
funtion returns a file descriptor that allows direct access to the
address space of the process handle
.Fa P .
A defined file descriptor is provided when using the
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_file 3PROC ,
and
.Xr Pgrab_core 3PROC
functions.
Note that the address space may be different in each of these cases and doesn’t
necessarily correspond to the /proc
.Sy as
file, except for live processes.
Other means of obtaining a
.Sy libproc
process handle may not define a file descriptor that contains the
address space.
.Pp
The returned file descriptor must not be closed and is only valid for
as long as the corresponding process handle
.Fa P
is valid.
After a call to
.Xr Prelease 3PROC
the file descriptor is invalidated.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pasfd
```



```
function returns a valid file descriptor.
Otherwise, if none exists, then
.Sy -1
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_file 3PROC ,
.Xr Prelease 3PROC ,
.Xr proc 4
```

```
**********************************************************
    2149 Mon Oct 15 13:26:11 2018
new/usr/src/man/man3proc/Pclearfault.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PCLEARFAULT 3PROC
.Os
.Sh NAME
.Nm Pclearfault ,
.Nm Lclearfault
.Nd clear process and thread faults
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pclearfault
.Fa "struct ps_prochandle *P"
.Fc
.Ft int
.Fo Lclearfault
.Fa "struct ps_lwphandle *L"
.Fc
.Sh DESCRIPTION
During normal operation a process may encounter a
.Sy fault ,
due to a hardware exception, identifying a problem with the running process.
Hardware faults include things like executing illegal instructions, encountering
a breakpoint, and arithmetic exceptions.
Faults are discussed further in
.Xr proc 4 .
.Pp
The
.Fn Pclearfault
function instructs the system to clear any fault pending delivery to a
thread in the process represented by the process handle
.Fa P .
The pending fault will never be delivered to process represented by
.Fa P .
.Pp
The
.Fn Lclearfault
function is identical to the
.Fn Pclearfault
function, except rather than operating on the representative thread of
the process it operates on the thread handle
.Fa L .
.Pp
The
.Fn Pclearfault
```



```
and
.Fn Lclearfault
functions only have meaning for active processes.
They will fail on process handles corresponding to zombie processes, ELF
objects, and cores.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pclearfault
and
.Fn Lclearfault
functions clear pending faults and return
.Sy 0 .
Otherwise,
.Sy -1
is returned,
.Sy errno
is set to indicate the error,
and no faults are cleared.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2100 Mon Oct 15 13:26:15 2018
new/usr/src/man/man3proc/Pclearsig.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PCLEARSIG 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pclearsig ,
  19 .Nm Lclearsig
  20 .Nd clear process signals
  21 .Sh LIBRARY
  22 .Lb libproc
  23 .Sh SYNOPSIS
  22 .Lb libproc
  24 .In libproc.h
  25 .Ft int
  26 .Fo Pclearsig
  27 .Fa "struct ps_prochandle *P"
  28 .Fc
  29 .Ft int
  30 .Fo Lclearsig
  31 .Fa "struct ps_lwphandle *L"
  32 .Fc
  33 .Sh DESCRIPTION
  34 During normal operation a process may receive a signal.
  35 Signals may indicate an error, for example referencing unmapped memory, an alarm
  36 firing, requests for information, and users requesting an interruption.
  37 For more information on the generation and usage of signals, see
  38 .Xr signal.h 3HEAD .
  39 .Pp
  40 The
  41 .Fn Pclearsig
  42 function instructs the system to clear any signal pending delivery to
  43 a thread in the process represented by the process handle
  44 .Fa P .
  45 The pending signal will never be delivered to process represented by
  46 .Fa P .
  47 .Pp
  48 The
  49 .Fn Lclearsig
  50 function is identical to the
  51 .Fn Pclearsig
  52 function, except rather than operating on the process and its
  53 representive thread, it instead operates on the thread handle
  54 .Fa L .
  55 .Pp
  56 The
  57 .Fn Pclearsig
  58 function only has meaning for active processes.
  59 It will fail on process handles corresponding to core files, zombie processes
```



```
  60 and ELF objects.
  61 .Sh RETURN VALUES
  62 Upon successful completion, the
  63 .Fn Pclearsig
  64 function clears pending faults and
  65 returns
  66 .Sy 0 .
  67 Otherwise,
  68 .Sy -1
  69 is returned,
  70 .Sy errno
  71 is set to indicate the error,
  72 and no faults are cleared.
  73 .Sh ERRORS
  74 For a full list of possible errors see the
  75 .Sy DIAGNOSTICS
  76 section in
  77 .Xr proc 4 .
  78 .Sh INTERFACE STABILITY
  79 .Sy Uncommitted
  80 .Sh MT-LEVEL
  81 See
  82 .Sy LOCKING
  83 in
  84 .Xr libproc 3LIB .
  85 .Sh SEE ALSO
  86 .Xr signal.h 3HEAD ,
  87 .Xr libproc 3LIB ,
  88 .Xr proc 4
```

```
**********************************************************
    1800 Mon Oct 15 13:26:18 2018
new/usr/src/man/man3proc/Pcontent.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PCONTENT 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pconent
  19 .Nd obtain process content types available
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft "core_content_t"
  25 .Fo Pcontent
  26 .Fa "struct ps_prochandle *P"
  27 .Fc
  28 .Sh DESCRIPTION
  29 The
  30 .Fn Pcontent
  31 function describes information available from the process handle
  32 .Fa P .
  33 .Pp
  34 Different types of process handles have different kinds of content
  35 available to them.
  36 For example, handles to active and running processes have more information
  37 available than various core files, as the core file retains a subset of
  38 information available in the running process.
  39 Handles that refer to ELF objects, obtained through
  40 .Xr Pgrab_file 3PROC ,
  41 will not have information such as a stack available.
  42 The content of core files is controlled by
  43 .Xr coreadm 1M .
  44 .Pp
  45 The symbols that may be returned are listed in the
  46 .Sy core_content_t
  47 heading in the
  48 .Sy TYPES
  49 section in
  50 .Xr libproc 3LIB .
  51 .Sh RETURN VALUES
  52 Upon successful completion, the
  53 .Fn Pcontent
  54 function returns the bitwise-inclusive-OR of content types.
  55 Otherwise,
  56 .Dv CC_CONTENT_INVALID
  57 is returned to indicate an error.
  58 .Sh INTERFACE STABILITY
  59 .Sy Uncommitted
```



```
  60 .Sh MT-LEVEL
  61 See
  62 .Sy LOCKING
  63 in
  64 .Xr libproc 3LIB .
  65 .Sh SEE ALSO
  66 .Xr libproc 3LIB ,
  67 .Xr Pcreate 3PROC ,
  68 .Xr Pgrab 3PROC ,
  69 .Xr Pgrab_core 3PROC ,
  70 .Xr Pgrab_file 3PROC ,
  71 .Xr proc 4
```

```
**********************************************************
    5102 Mon Oct 15 13:26:22 2018
new/usr/src/man/man3proc/Pcreate.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PCREATE 3PROC
.Os
.Sh NAME
.Nm Pcreate ,
.Nm Pxcreate ,
.Nm Pcreate_callback
.Nd create and control a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "struct ps_prochandle *"
.Fo Pcreate
.Fa "const char *file"
.Fa "char *const *argv"
.Fa "int *perr"
.Fa "char *path"
.Fa "size_t len"
.Fc
.Ft "struct ps_prochandle *"
.Fo Pxcreate
.Fa "const char *file"
.Fa "char *const *argv"
.Fa "char *const *envp"
.Fa "int *perr"
.Fa "char *path"
.Fa "size_t len"
.Fc
.Ft void
.Fo Pcreate_callback
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pcreate
function creates a process controlled by the
.Sy libproc
library.
The
.Fn Pxcreate
function does the same while also allowing the replacement of the
environment via
.Fa envp .
.Pp
Both functions cause the caller to
```



```
.Xr fork 2 .
Followed by the child calling
.Xr exec 2
to load the new process image specified by
.Fa file .
The
.Ev PATH
is searched for
.Fa file
if it is not an absolute path, similar to
.Xr execvp 2 .
.Pp
The process image will be invoked with the arguments specified by
.Fa argv ,
which should be a
.Dv NULL Ns -terminated
array of character strings.
Each entry in the array is an individual argument.
The environment of the process image will be inherited from the running process
if the
.Fn Pcreate
function is called or if the
.Fn Pxcreate
function is called and the value of
.Fa envp
is
.Dv NULL .
Otherwise,
.Fa envp
should be a
.Dv NULL Ns -terminated
array of character strings whose entries are in the form of
.Em key=value .
For more on the process environment, see
.Xr environ 5 .
.Pp
The
.Fn Pcreate_callback
function allows a way for callers to inject a callback into the child
process before the call to
.Xr exec 2 .
The symbol
.Sy Pcreate_callback
is a symbol that may be interposed on by consumers.
It allows the chance for the modification of signal dispositions or any other
changes that a caller may wish to make.
.Pp
If the caller's real user or group ID is not their effective user or
group ID, then the child process's user and group IDs will all be reset
to the real user and group id.
.Pp
The
.Fa perr
argument must be a
.Pf non- Dv NULL
pointer.
If the
.Fn Pcreate
or
.Fn Pxcreate
functions fail, the value pointed to will be filled in with a more
detailed error code listed in
.Sx ERRORS .
A human-readable error message is obtained with
.Xr Pcreate_error 3PROC .
.Pp
```

```
126 Multiple executables named
127 .Fa file
128 may exist on the
129 .Fa PATH .
130 To determine the full path of the executable pass a non-NULL
131 .Fa path
132 pointer.
133 Upon successful completion of
134 .Fn Pcreate
135 or
136 .Fn Pxcreate
137 the
138 .Fa path
139 pointer will contain the full path up to
140 .Fa len
141 bytes, including the NUL character.
142 .Pp
143 Upon successful completion of the
144 .Fn Pcreate
145 or
146 .Fn Pxcreate
147 function, a handle to the process is returned.
148 This handle is usable with other
149 .Sy libproc
150 routines and will persist until either
151 .Xr Pfree 3PROC
152 or
153 .Xr Prelease 3PROC
154 is called on the resulting handle.
155 The process created is stopped just after return from the
156 .Xr exec 2
157 family of system calls.
158 The process will not run, unless the caller sets it running or releases its
159 handle to the process.
160 .Pp
161 A 32-bit process cannot use this interface to launch or control a
162 64-bit process.
163 However, a 64-bit process can create and control both 32-bit and 64-bit
164 processes.
165 .Sh RETURN VALUES
166 Upon successful completion, both the
167 .Fn Pcreate
168 and
169 .Fn Pxcreate
170 functions create a new process and return a
171 .Sy libproc
172 handle to it.
173 Otherwise,
174 .Sy NULL
175 is returned and
176 .Fa perr
177 is filled in with the corresponding error.
178 .Sh ERRORS
179 The
180 .Fn Pcreate
181 and
182 .Fn Pxcreate
183 functions will fail if:
184 .Bl -tag -width Er
185 .It Er C_FORK
186 The call to
187 .Xr fork 2
188 failed.
189 .It Er C_INTR
190 The operation was interrupted by a signal.
191 .It Er C_LP64
```



```
192 The calling process is 32-bit, but it attempted to control a 64-bit
193 process.
194 .It Er C_NOEXEC
195 The specified
196 .Fa file
197 or the one found by searching
198 .Dv PATH
199 cannot be executed.
200 .It Er C_NOENT
201 The specified
202 .Fa file
203 does not exist or it could not be found by searching
204 .Dv PATH .
205 .It Er C_PERM
206 The specified
207 .Fa file
208 or the one found by searching
209 .Dv PATH
210 is set-id or unreadable.
211 .It Er C_STRANGE
212 An unanticipated system error occurred while trying to create the
213 process and its handle.
214 When this occurs, then the value of
215 .Sy errno
216 is meaningful.
217 See
218 .Xr errno 3C
219 for more information and
220 .Xr Intro 2
221 for the list of possible errors.
222 .El
223 .Sh INTERFACE STABILITY
224 .Sy Uncommitted
225 .Sh MT-LEVEL
226 .Sy MT-Safe
227 .Sh SEE ALSO
228 .Xr exec 2 ,
229 .Xr execvp 2 ,
230 .Xr fork 2 ,
231 .Xr Intro 2 ,
232 .Xr errno 3C ,
233 .Xr libproc 3LIB ,
234 .Xr Pcreate_error 3PROC ,
235 .Xr Pfree 3PROC ,
236 .Xr Prelease 3PROC
```

```
**********************************************************
    2046 Mon Oct 15 13:26:25 2018
new/usr/src/man/man3proc/Pcreate_agent.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PCREATE_AGENT 3PROC
.Os
.Sh NAME
.Nm Pcreate_agent
.Nd create the agent LWP
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pcreate_agent
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pcreate_agent
function creates the agent LWP in the process represented by the handle
.Fa P .
The agent LWP is used as a means to force system calls to be invoked on
the controlled process.
For more information on the agent LWP, see
.Xr proc 4 .
.Pp
The agent LWP cannot be created for process handles corresponding to
core files, zombie processes, processes that have yet to run, and ELF
objects.
.Pp
The
.Fn Pcreate_agent
function is reentrant.
It may be entered recursively.
The act of creating the agent LWP will cause the process to be stopped.
For every call to the
.Fn Pcreate_agent
function, a corresponding call to
.Xr Pdestroy_agent 3PROC
is required.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pcreate_agent
function returns
.Sy 0
and creates the agent LWP.
Otherwise,
.Sy -1
```



```
is returned,
.Sy errno
is set to indicate the error, and the agent LWP is not created.
.Sh ERRORS
The
.Fn Pcreate_agent
function will fail if:
.Bl -tag -width Er
.It Er ENOENT
The process referred to by
.Fa P
is a core file, zombie, ELF object, or has not begun execution.
.El
.Pp
Note, it is possible for other error numbers to be returned.
If they are, they represent unanticipated failure.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pdestroy_agent 3PROC ,
.Xr proc 4
```

```
***********************************************************
    1287 Mon Oct 15 13:26:26 2018
new/usr/src/man/man3proc/Pcreate_error.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PCREATE_ERROR 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pcreate_error
  19 .Nd get Pcreate, Pxcreate error message string
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft "const char *"
  25 .Fo Pcreate_error
  26 .Fa "int error"
  27 .Fc
  28 .Sh DESCRIPTION
  29 The
  30 .Fn Pcreate_error
  31 function returns a pointer to a human-readable character string
  32 describing the error that occurred.
  33 The
  34 .Fn Pcreate_error
  35 function translates errors produced by the
  36 .Xr Pcreate 3PROC
  37 and
  38 .Xr Pxcreate 3PROC
  39 functions only (passed as the
  40 .Fa perr
  41 argument).
  42 .Sh RETURN VALUES
  43 The
  44 .Fn Pcreate_error
  45 function always returns a pointer to a character string that describes
  46 the error that occurred, even if it is an unknown error.
  47 .Sh INTERFACE STABILITY
  48 .Sy Uncommitted
  49 .Sh MT-LEVEL
  50 .Sy MT-Safe
  51 .Sh SEE ALSO
  52 .Xr libproc 3LIB ,
  53 .Xr Pcreate 3PROC ,
  54 .Xr Pxcreate 3PROC
```

```
**********************************************************
    1972 Mon Oct 15 13:26:28 2018
new/usr/src/man/man3proc/Pcred.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PCRED 3PROC
.Os
.Sh NAME
.Nm Pcred
.Nd obtain process credentials
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pcred
.Fa "struct ps_prochandle *P"
.Fa "prcred_t *pcrp"
.Fa "int ngroups"
.Fc
.Sh DESCRIPTION
The
.Fn Pcred
function obtains the credentials of the process from the handle
.Fa P .
.Pp
The credentials are read into the buffer pointed to by
.Fa pcrp .
The
.Sy prcred_t
type is defined in
.Xr proc 4 .
It contains information about the current effective, saved, and real
user and group IDs.
It also allows for supplemental groups to be obtained.
The
.Fn Pcred
function will read a number of supplemental groups based on the value of
.Fa ngroups .
The
.Sy prcred_t
structure only contains the space for one supplemental group by default.
Callers should ensure that the buffer pointed to by
.Fa pcrp
contains enough space to include all of the required supplemental
groups that are desired.
.Pp
Not all process handles have credential information available to them.
For example, the handles that come from
.Xr Pgrab_file 3PROC
```



```
have no processes associated with them and thus have no credentials
associated with them.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pcred
function returns
.Sy 0
and updates the memory at
.Fa pcrp
with the credentials.
Otherwise,
.Sy -1
is returned to indicate an error.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Psetcred 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1757 Mon Oct 15 13:26:31 2018
new/usr/src/man/man3proc/Pctlfd.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PCTFLD 3PROC
.Os
.Sh NAME
.Nm Pctlfd
.Nd obtain the process control file descriptor
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pctlfd
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pctlfd
function returns a file descriptor to the underlying /proc file system
.Sy ctl
file for the process identified by the handle
.Fa P .
This may be used for injecting control operations manually;
however, many interfaces for using it are provided by
.Xr libproc 3LIB
itself.
.Pp
Only live processes have a control file descriptor.
Process handles that correspond to files and cores, created through
.Xr Pgrab_file 3PROC
and
.Xr Pgrab_core 3PROC ,
do not have a corresponding file descriptor.
.Pp
The file descriptor is invalidated when the process handle is released
through
.Xr Prelease 3PROC
or if control is lost and the handle is reopened.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pctlfd
function returns a valid file descriptor.
Otherwise, if none exists, then
.Sy -1
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
```



```
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_file 3PROC ,
.Xr Prelease 3PROC ,
.Xr Preopen 3PROC ,
.Xr proc 4
```

```
**********************************************************
    2105 Mon Oct 15 13:26:35 2018
new/usr/src/man/man3proc/Pdelbkpt.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PDELBKPT 3PROC
.Os
.Sh NAME
.Nm Pdelbkpt
.Nd clear a breakpoint in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pdelbkpt
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t address"
.Fa "ulong_t saved"
.Fc
.Sh DESCRIPTION
The
.Fn Pdelbkpt
function removes the breakpoint installed at
.Fa address
from process
.Fa P .
Restoring the instruction present in
.Fa saved .
.Pp
If the instruction at
.Fa address
is no longer the architecture-specific breakpoint instruction, then
.Fa saved
is not restored, but the function still returns successfully.
This behavior is done due to the presence of setting breakpoints in
self-modifying code, e.g. procedure linkage tables.
.Pp
The
.Fn Pdelbkpt
function only works on running processes, such as those created through
.Xr Pgrab 3PROC
and
.Xr Pcreate 3PROC .
Attempting to remove a breakpoint from process handles to core files,
zombie processes, or ELF objects will fail.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pdelbkpt
function removes the breakpoint and
```



```
returns
.Sy 0 .
Otherwise,
.Sy -1
is returned, and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Pdelbkpt
function will fail with:
.Bl -tag -width Er
.It Er ENOENT
The handle
.Fa P
refers to a grabbed core file, a zombie process, or an ELF object.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_file 3PROC ,
.Xr Psetbkpt 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1894 Mon Oct 15 13:26:38 2018
new/usr/src/man/man3proc/Pdelwapt.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PDELWAPT 3PROC
.Os
.Sh NAME
.Nm Pdelwapt
.Nd remove a watchpoint in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pdelwapt
.Fa "struct ps_prochandle *P"
.Fa "const prwatch_t *wp"
.Fc
.Sh DESCRIPTION
The
.Fn Pdelwapt
function removes the watchpoint specified by
.Fa wp
from the process represented by the handle
.Fa P .
The watchpoint should have been installed with either
.Xr Psetwapt 3PROC
or the underlying /proc file system
.Sy PCWATCH
operation.
.Pp
The
.Fn Pdelwapt
function only works on running processes, such as those created through
.Xr Pgrab 3PROC
and
.Xr Pcreate 3PROC .
Attempting to remove a watchpoint from process handles to core files,
zombie processes, or ELF objects will fail.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pdelwapt
function removes the watchpoint and returns
.Sy 0 .
Otherwise,
.Sy -1
is returned,
.Sy errno
is set to indicate the error.
```



```
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Pdelwapt
function will fail with:
.Bl -tag -width Er
.It Er ENOENT
The handle
.Fa P
refers to a grabbed core file, a zombie process, or an ELF object.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_file 3PROC ,
.Xr Psetwapt 3PROC ,
.Xr proc 4
```

```
***********************************************************
    1260 Mon Oct 15 13:26:41 2018
new/usr/src/man/man3proc/Pdestroy_agent.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PDESTROY_AGENT 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pdestroy_agent
  19 .Nd destroy the agent LWP
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft void
  25 .Fo Pdestroy_agent
  26 .Fa "struct ps_prochandle *P"
  27 .Fc
  28 .Sh DESCRIPTION
  29 The
  30 .Fn Pdestroy_agent
  31 function
  32 removes the agent LWP in
  33 .Fa P .
  34 The existence of the agent LWP is reference counted by the library and
  35 therefore if
  36 .Xr Pcreate_agent 3PROC
  37 has been called multiple times, the
  38 .Fn Pdestroy_agent
  39 function must be called an equal number of times.
  40 Upon the last time, it will destroy the agent LWP.
  41 .Pp
  42 Destroying the agent LWP does not change the state of the process
  43 represented by
  44 .Fa P .
  45 .Sh INTERFACE STABILITY
  46 .Sy Uncommitted
  47 .Sh MT-LEVEL
  48 See
  49 .Sy LOCKING
  50 in
  51 .Xr libproc 3LIB .
  52 .Sh SEE ALSO
  53 .Xr libproc 3LIB ,
  54 .Xr Pcreate_agent 3PROC
```

```
**********************************************************
    1833 Mon Oct 15 13:26:44 2018
new/usr/src/man/man3proc/Penv_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd March 2, 2017
.Dt PENV_ITER 3PROC
.Os
.Sh NAME
.Nm Penv_iter
.Nd iterate process environment
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Penv_iter
.Fa "struct ps_prochandle *P"
.Fa "proc_env_f *func"
.Fa "void *data"
.Fc
.Sh DESCRIPTION
The
.Fn Penv_iter
function iterates over the environment of the process represented by
.Fa P .
For each environment variable,
.Fa func
is passed the caller argument
.Fa data
along with the address of the environment variable and the key-value pair.
For the full signature of the
.Ft proc_env_f
callback, see
.Xr libproc 3LIB .
.Pp
The callback's return value controls whether or not iteration proceeds.
If
.Fa func
returns zero, then iteration continues.
Otherwise, iteration is terminated and the value is returned.
It is recommended that callback functions do not return
.Sy -1
so as to distinguish between the failure of the
.Fn Penv_iter
function and the callback function.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Penv_iter
function returns
.Sy 0 .
```



```
Otherwise, if there was an internal error, for example due to a
corrupted environment, then
.Sy -1
is returned.
Otherwise, if the callback function
.Fa func
returns non-zero, then its return value will be returned instead.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB
```

```
**********************************************************
    1523 Mon Oct 15 13:26:47 2018
new/usr/src/man/man3proc/Pexecname.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PEXECNAME 3PROC
.Os
.Sh NAME
.Nm Pexecname
.Nd obtain full path to process executable
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "char *"
.Fo Pexecname
.Fa "struct ps_prochandle *P"
.Fa "char *buf"
.Fa "size_t buflen"
.Fc
.Sh DESCRIPTION
The
.Fn Pexecname
function attempts to determine the full path to the process executable
referred to by the handle
.Fa P .
.Pp
If found,
.Fa buf
will be filled in with the full path for up to
.Fa buflen
bytes, including the null terminator.
.Pp
For a handle grabbed with
.Xr Pgrab_file 3PROC ,
the executable refers to the path of the file itself.
For a core file, the system attempts to determine the original path of the
executable and return that.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pexecname
function returns the value of
.Fa buf ,
and up to
.Fa buflen
bytes of
.Fa buf
are filled in with a null-terminated path.
Otherwise,
.Dv NULL
```



```
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB
```

```
**********************************************************
    2506 Mon Oct 15 13:26:51 2018
new/usr/src/man/man3proc/Pfault.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PFAULT 3PROC
.Os
.Sh NAME
.Nm Pfault
.Nd enable and disable the tracing of faults
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "int"
.Fo Pfault
.Fa "struct ps_prochandle *P"
.Fa "int which"
.Fa "int stop"
.Fc
.Sh DESCRIPTION
The
.Fn Pfault
function controls what the process
.Fa P
should do on faults.
.Pp
A fault is a hardware event that occurs in the context of a running
process and thread.
A hardware fault may occur because an illegal instruction was executed,
a breakpoint or watchpoint was encountered, or an arithmetic exception occurred,
among others.
The full list of faults is available in both
.Xr proc 4
and
.In sys/fault.h .
.Pp
For each hardware fault, a process may be configured to stop the thread
that encountered it when it occurs.
The value of the
.Fa stop
parameter controls whether or not the listed fault in
.Fa which
will cause the thread to trap.
A value of 1 indicates the thread should stop; a value of 0 indicates it should
not.
.Pp
The value of
.Fa which
indicates which hardware fault the change applies to.
```



```
However, if the value of
.Fa which
is zero, then it applies to all faults.
.Pp
The
.Fn Pfault
function only applies to actively running processes.
It does not function on handles that refer to core files, zombie processes, or
ELF objects.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pfault
function returns the old disposition of the fault --
.Sy 0
if it was not set to stop and
.Sy 1
if it was --
and the fault state is updated.
Otherwise,
.Sy -1
is returned,
.Dv errno
is updated with the error that occurred, and the fault state is not
updated.
.Sh ERRORS
The
.Fn Pfault
function will fail if:
.Bl -tag -width Er
.It Er EINVAL
The value of
.Fa which
is invalid, e.g. it is less than zero or greater than the largest defined
fault.
.It Er ENOENT
The handle
.Fa P
refers to a process that is a zombie, a core file, or a file.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2084 Mon Oct 15 13:26:56 2018
new/usr/src/man/man3proc/Pfdinfo_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PFDINFO_ITER 3PROC
.Os
.Sh NAME
.Nm Pfdinfo_iter
.Nd iterate open files in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pfdinfo_iter
.Fa "struct ps_prochandle *P"
.Fa "proc_fdinfo_f *func"
.Fa "void *data"
.Fc
.Sh DESCRIPTION
The
.Fn Pfdinfo_iter
function iterates over the open files in the process represented by
.Fa P .
.Pp
For each open file, the callback function
.Fa func
will be invoked and it will be passed the
.Fa data
argument as well as a pointer to a
.Sy prfdinfo_t
structure defined in
.Xr libproc 3LIB .
For the full signature of the
.Vt proc_fdinfo_f
see
.Xr libproc 3LIB .
.Pp
The return value of
.Fa func
controls whether or not iteration continues.
If
.Fa func
returns
.Sy 0 ,
then iteration will continue.
However, if
.Fa func
instead returns a non-zero value, then iteration will halt and that
```



```
value will be used as the return value of the
.Fn Pfdinfo_iter
function.
Because the
.Fn Pfdinfo_iter
function returns
.Sy -1
to an indicate its own errors, it is recommended that the callback
function does not return
.Sy -1
to indicate an error so that the caller may distinguish between the
failure of the callback function and the failure of the
.Fn Pfdinfo_iter
function.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pfdinfo_iter
returns
.Sy 0 .
Otherwise, if there was an internal error then
.Sy -1
is returned.
Otherwise, if the callback function
.Fa func
returns non-zero, then its return value will be returned instead.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB
```

```
**********************************************************
    3541 Mon Oct 15 13:26:59 2018
new/usr/src/man/man3proc/Pgetareg.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGETAREG 3PROC
.Os
.Sh NAME
.Nm Pgetareg ,
.Nm Pputareg ,
.Nm Lgetareg ,
.Nm Lputareg
.Nd set and get a register from a stopped process or thread
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pgetareg
.Fa "struct ps_prochandle *P"
.Fa "int regno"
.Fa "prgreg_t *preg"
.Fc
.Ft int
.Fo Pputareg
.Fa "struct ps_prochandle *P"
.Fa "int regno"
.Fa "prgreg_t preg"
.Fc
.Ft int
.Fo Lgetareg
.Fa "struct ps_lwphandle *L"
.Fa "int regno"
.Fa "prgreg_t *preg"
.Fc
.Ft int
.Fo Lputareg
.Fa "struct ps_lwphandle *L"
.Fa "int regno"
.Fa "prgreg_t preg"
.Fc
.Sh DESCRIPTION
The
.Fn Pgetareg
and
.Fn Pputareg
functions read and update the registers of the process handle referred
to by
.Fa P .
The getting and setting of registers of the process operates on the
```



```
representative thread (LWP).
For more information on how the representative is chosen, see
.Xr proc 4 .
.Pp
To change the registers of a specific thread, use the
.Fn Lgetareg
and
.Fn Lputareg
functions.
.Pp
The getting and setting of registers only applies to stopped processes.
In addition, one may obtain registers from core files, but not set them.
To stop a process, see the
.Xr Pstop 3PROC
function.
.Pp
The register to get or set is indicated by the
.Fa regno
argument.
For a list of registers, see
.In sys/regset.h .
The set of registers is specific to each architecture of the system.
The
.Fn Pgetareg
function will fill in the value of
.Fa preg
with the value of the register
.Fa regno ,
while the
.Fn Pputareg
function will update the value of the register
.Fa regno
with the value in
.Fa preg .
Updated registers will be set when the process resumes execution.
.Pp
The
.Fn Lgetareg
and
.Fn Lputareg
functions are equivalent to the
.Fn Pgetareg
and
.Fn Psetareg
functions, except rather than operating on the process and its
representative thread, they instead operate on the thread handle
.Fa L .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pgetareg
and
.Fn Pputareg
function return
.Sy 0 .
Otherwise,
.Sy -1
is returned,
.Sy errno
is set, and no registers will have been gotten or updated.
.Sh ERRORS
The
.Fn Pgetareg
and
.Fn Lgetareg
functions will fail if:
.Bl -tag -width Er
```

```
 126 .It Er EINVAL
 127 The value of
 128 .Fa regno
 129 is invalid.
 130 This means it is less than
 131 .Sy 0
 132 and greater than
 133 .Sy NPRGREG .
 134 Note,
 135 .Sy NPRGREG Ns ’s
 136 value varies based on process architecture.
 137 .It Er EBUSY
 138 The handle
 139 .Fa P
 140 is neither stopped nor a core file.
 141 .It Er ENODATA
 142 The handle
 143 .Fa P
 144 refers to a file obtained through
 145 .Xr Pgrab_file 3PROC .
 146 .El
 147 .Pp
 148 The
 149 .Fn Pputareg
 150 and
 151 .Fn Lputareg
 152 functions will fail if:
 153 .Bl -tag -width Er
 154 .It Er EINVAL
 155 The value of
 156 .Fa regno
 157 is invalid.
 158 This means it is less than
 159 .Sy 0
 160 and greater than
 161 .Sy NPRGREG .
 162 Note,
 163 .Sy NPRGREG Ns ’s
 164 value varies based on process architecture.
 165 .It Er EBUSY
 166 The handle
 167 .Fa P
 168 is not stopped or refers to a non-active process.
 169 .El
 170 .Sh INTERFACE STABILITY
 171 .Sy Uncommitted
 172 .Sh MT-LEVEL
 173 See
 174 .Sy LOCKING
 175 in
 176 .Xr libproc 3LIB .
 177 .Sh SEE ALSO
 178 .Xr errno 3C ,
 179 .Xr libproc 3LIB ,
 180 .Xr Lgrab 3PROC ,
 181 .Xr Pgrab_file 3PROC ,
 182 .Xr Pstop 3PROC ,
 183 .Xr proc 4
```

```
**********************************************************
    1362 Mon Oct 15 13:27:02 2018
new/usr/src/man/man3proc/Pgetauxval.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PGETAUXVAL 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pgetauxval
  19 .Nd obtain auxiliary vector value
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft int
  25 .Fo Pgetauxval
  26 .Fa "struct ps_prochandle *P"
  27 .Fa "int type"
  28 .Fc
  29 .Sh DESCRIPTION
  30 The
  31 .Fn Pgetauxval
  32 function looks up the entry
  33 .Fa type
  34 in the auxiliary vector of the process handle
  35 .Fa P
  36 and returns its value.
  37 The
  38 .Fa type
  39 argument should be the entry of the auxiliary vector.
  40 The list of such types may be found in
  41 .In sys/auxv.h .
  42 .Sh RETURN VALUES
  43 Upon successful completion, the
  44 .Fn Pgetauxval
  45 function returns the value of the auxiliary vector entry
  46 .Fa type .
  47 Otherwise,
  48 .Sy -1
  49 is returned to indicate that either the auxiliary vector or the
  50 .Fa type
  51 entry could not be found.
  52 .Sh INTERFACE STABILITY
  53 .Sy Uncommitted
  54 .Sh MT-LEVEL
  55 See
  56 .Sy LOCKING
  57 in
  58 .Xr libproc 3LIB .
  59 .Sh SEE ALSO
```



```
  60 .Xr libproc 3LIB ,
  61 .Xr Pgetauxvec 3PROC ,
  62 .Xr proc 4
```

```
***********************************************************
    1426 Mon Oct 15 13:27:04 2018
new/usr/src/man/man3proc/Pgetauxvec.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGETAUXVEC 3PROC
.Os
.Sh NAME
.Nm Pgetauxvec
.Nd obtain process auxiliary vector
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const auxv_t *"
.Fo Pgetauxvec
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pgetauxvec
function returns a pointer to a copy of the auxiliary vector for the
process handle
.Fa P .
If the process handle does not represent an actively running process or
the auxiliary vector could not be found, then it instead returns an
empty auxiliary vector.
The definitions of the
.Sy auxv_t
may be found in
.In sys/auxv.h .
.Pp
The returned auxiliary vector should not be used across any subsequent
calls to
.Xr libproc 3LIB .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pgetauxvec
function always returns a pointer to an auxiliary vector.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pgetauxval 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1365 Mon Oct 15 13:27:07 2018
new/usr/src/man/man3proc/Pgetenv.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGETENV 3PROC
.Os
.Sh NAME
.Nm Pgetenv
.Nd get process environment variable
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "char *"
.Fo Pgetenv
.Fa "struct ps_prochandle *P"
.Fa "const char *name"
.Fa "char *buf"
.Fa "size_t buflen"
.Fc
.Sh DESCRIPTION
The
.Fa Pgetenv
function searches the environment of the process handle
.Fa P
for the environment variable named
.Fa name .
If found it copies up to
.Fa buflen
characters of the value, including the null terminator, into the buffer
.Fa buf .
Everything after the
.Sy =
sign is copied.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pgetenv
function returns the pointer
.Fa buf .
Otherwise, if the environment variable was not found,
.Dv NULL
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
```



```
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Penv_iter 3PROC ,
.Xr environ 5
```

```
**********************************************************
    6479 Mon Oct 15 13:27:09 2018
new/usr/src/man/man3proc/Pgrab.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGRAB 3PROC
.Os
.Sh NAME
.Nm Pgrab
.Nd grab and control a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "struct ps_prochandle *"
.Fo Pgrab
.Fa "pid_t pid"
.Fa "int flags"
.Fa "int *perr"
.Fc
.Sh DESCRIPTION
The
.Fn Pgrab
function attempts to grab the process identified by
.Fa pid
and returns a handle to it that allows the process to be controlled,
interrogated, and manipulated.
This interface only works with processes that already exist.
Use
.Xr Pgrab_core 3PROC
for core files and
.Xr Pcreate 3PROC
to create processes.
.Pp
A grabbed process undergoes the following changes unless
.Fa flags
is set to the contrary:
.Bl -bullet -offset indent
.It
The process is stopped
.It
All other tracing flags are cleared
.It
The grab is exclusive.
If any existing handles to this process exist or anyone else is using the
underlying facilities of the /proc file system to control this process,
it will fail.
.It
Unless the process is already stopped, the
.Dv PR_RLC
```



```
flag is set indicating the process should run-on-last-close.
Allowing the process to resume running if its controlling process dies.
.El
.Pp
Grabbing a process is a
.Em destructive
action.
Stopping a process stops execution of all its threads.
The impact of stopping a process depends on the purpose of that process.
For example, if one stops a process that's primarily doing
computation, then its computation is delayed the entire time that it
is stopped.
However, if instead this is an active TCP server, then the accept backlog may
fill causing connection errors and potentially connection time out errors.
.Pp
Special care must be taken to ensure that a stopped process continues,
even if the controlling process terminates.
If the controlling process disables the
.Dv PR_RLC
flag or the process was already stopped, then the process remains
stopped after the controlling process terminates.
Exercise caution when changing this behavior.
.Pp
Many of these default behaviors can be controlled by passing values to
the
.Fa flags
argument.
Values for
.Fa flags
are constructed by a bitwise-inclusive-OR of flags from the following
list:
.Bl -tag -width Dv -offset indent
.It Dv PGRAB_RETAIN
Indicates that any existing tracing flags on
.Fa pid
should be retained.
If this flag is not specified, they will be cleared as part of creating the
.Sy libproc
handle for this process.
.Pp
Normally extant tracing flags are cleared when a process is grabbed.
.It Dv PGRAB_FORCE
Indicates that the process should not be grabbed exclusively.
Care should be taken with this option.
If other consumers are manipulating the process, then this may result in
surprising behavior as the process is being manipulated from multiple points of
control at the same time.
.Pp
Normally an attempt will be made to grab the process exclusively and
fail if it is already in use.
.It Dv PGRAB_RDONLY
Indicates that the process should be grabbed in a read-only fashion.
This implies that both the
.Dv PGRAB_RETAIN
and
.Dv PGRAB_NOSTOP
flags should be set.
If a process is opened read-only, then a caller can only read information about
a process and cannot manipulate it, change its current state, or inject systems
calls into it.
.Pp
Normally when a process is grabbed, it does so for both reading and writing.
.It Dv PGRAB_NOSTOP
Do not stop a process as it is grabbed.
Note, any extant tracing flags on the process will still be cleared unless the
.Dv PGRAB_RETAIN
```

```
126 flag has been set.
127 .Pp
128 Normally a process is stopped as a result of grabbing the process.
129 .El
130 .Pp
131 The
132 .Fa perr
133 argument must be a
134 .Pf non- Dv NULL
135 pointer which will store a more detailed error in the event that the
136 .Fn Pgrab
137 function fails.
138 A human-readable form of the error can be obtained with
139 .Xr Pgrab_error 3PROC .
140 .Pp
141 Once a caller is done with the library handle it should call
142 .Xr Prelease 3PROC
143 to release the grabbed process.
144 Failure to properly release the handle may leave a process stopped and interfere
145 with the ability of other software to obtain a handle.
146 .Ss Permissions
147 Unprivileged users may grab and control their own processes only if both
148 the user and group IDs of the target process match those of the calling
149 process.
150 In addition, the caller must have a super set of the target’s privileges.
151 Processes with the
152 .Sy PRIV_PROC_OWNER
153 privilege may manipulate any process on the system, as long as it has an
154 equal privilege set.
155 For more details on the security and programming considerations, please see the
156 section
157 .Sy PROGRAMMING NOTES
158 in
159 .Xr proc 4 .
160 .Sh RETURN VALUES
161 Upon successful completion, the
162 .Fn Pgrab
163 function returns a control handle to the process.
164 Otherwise,
165 .Dv NULL
166 is returned with
167 .Fa perr
168 containing the error code.
169 .Sh ERRORS
170 The
171 .Fn Pgrab
172 function will fail if:
173 .Bl -tag -width Er
174 .It Er G_BUSY
175 The process
176 .Fa pid
177 is already being traced and the
178 .Dv PGRAB_FORCE
179 flag was not passed in
180 .Fa flags .
181 .It Er G_LP64
182 The calling process is a 32-bit process and process
183 .Fa pid
184 is 64-bit.
185 .It Er G_NOFD
186 Too many files are open.
187 This is logically equivalent to receiving
188 .Er EMFILE .
189 .It Er G_NOPROC
190 The process referred to by
191 .Fa pid
```



```
192 does not exist.
193 .It Er G_PERM
194 The calling process has insufficient permissions or privileges to open
195 the specified process.
196 See
197 .Sx Permissions
198 for more information.
199 .It Er G_SYS
200 The process referred to by
201 .Fa pid
202 is a system process and cannot be grabbed.
203 .It Er G_SELF
204 The process referred to by
205 .Fa pid
206 is the process ID of the caller and the
207 .Dv PGRAB_RDONLY
208 was not passed.
209 A process may only grab itself if it’s read-only.
210 .It Er G_STRANGE
211 An unanticipated system error occurred while trying to grab the process
212 file and create the handle.
213 The value of
214 .Sy errno
215 indicates the system failure.
216 .It Er G_ZOMB
217 The process referred to by
218 .Fa pid
219 is a zombie and cannot be grabbed.
220 .El
221 .Sh INTERFACE STABILITY
222 .Sy Uncommitted
223 .Sh MT-LEVEL
224 .Sy MT-Safe
225 .Sh SEE ALSO
226 .Xr errno 3C ,
227 .Xr libproc 3LIB ,
228 .Xr Pfree 3PROC ,
229 .Xr Pgrab_core 3PROC ,
230 .Xr Pgrab_error 3PROC ,
231 .Xr Pgrab_file 3PROC ,
232 .Xr Prelease 3PROC
```

```
**********************************************************
    4103 Mon Oct 15 13:27:11 2018
new/usr/src/man/man3proc/Pgrab_core.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGRAB_CORE 3PROC
.Os
.Sh NAME
.Nm Pgrab_core ,
.Nm Pfgrab_core
.Nd grab a core file
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "struct ps_prochandle *"
.Fo Pgrab_core
.Fa "const char *core"
.Fa "const char *aout"
.Fa "int gflag"
.Fa "int *perr"
.Fc
.Ft "struct ps_prochandle *"
.Fo Pfgrab_core
.Fa "int core_fd"
.Fa "const char *aout"
.Fa "int *perr"
.Fc
.Sh DESCRIPTION
The
.Fn Pgrab_core
and
.Fn Pfgrab_core
functions open a core file for introspection.
Unlike live processes, core files cannot have their state modified;
however, all of the functions that iterate or query state will work.
These functions work on all illumos core files and the core files of some other
operating systems.
See both
.Xr core 4
and the
.Em Core Files
section of
.Xr libproc 3LIB
for more information.
.Pp
The
.Fn Pgrab_core
function attempts to open the core file specified by
.Fa core .
```



```
The system attempts to determine the path of the original executable.
The argument
.Fa aout
may either be the path to that file, a path to a directory to search, or the
.Dv NULL
pointer, if neither is known.
The system will search for it and will supplement information in the core file
with that.
.Pp
The
.Fa gflag
argument to the
.Fn Pgrab_core
function controls how the file is opened.
If the
.Dv PGRAB_RDONLY
flag is specified, then the core file will be opened with the
.Xr open 2
flag
.Dv O_RDONLY .
Otherwise, it will be opened
.Dv O_RDWR .
.Pp
The
.Fa perr
argument must be a
.Pf non- Dv NULL
pointer which will store a more detailed error in the event that the
.Fn Pgrab_core
function fails.
A human-readable form of the error can be obtained through the routine
.Xr Pgrab_error 3PROC .
.Pp
The
.Fn Pfgrab_core
is similar to the
.Fn Pgrab_core
function.
Except, instead of operating on a path, it opens a handle to the core file
referenced by
.Fa core_fd .
The
.Fa aout
and
.Fa perr
arguments are identical to those in the
.Fn Pgrab_core
function.
.Pp
The handle returned, from either function, is valid until it is closed
with
.Xr Prelease 3PROC
or
.Xr Pfree 3PROC .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pgrab_core
and
.Fn Pfgrab_core
functions return a
.Sy libproc
handle to the core file.
Otherwise,
.Dv NULL
is returned and
.Fa perr
```

```
126 is filled in with a more detailed error message.
127 .Sh ERRORS
128 The
129 .Fn Pgrab_core
130 function will fail if:
131 .Bl -tag -width Er
132 .It Er G_NOCORE
133 The file
134 .Fa core
135 does not exist.
136 .It Er G_STRANGE
137 An unexpected system error occurred while trying to open
138 .Fa core .
139 The value of
140 .Sy errno
141 indicates the system failure.
142 .El
143 .Pp
144 The
145 .Fn Pgrab_core
146 and
147 .Fn Pfgrab_core
148 functions will fail if:
149 .Bl -tag -width Er
150 .It Dv G_ELF
151 An unexpected
152 .Xr libelf 3LIB
153 failure occurred.
154 .It Dv G_FORMAT
155 The core file referred to by either
156 .Fa core
157 or
158 .Fa core_fd
159 is not a valid ELF core file.
160 .It Dv G_ISAINVAL
161 The architecture of the core file referred to by either
162 .Fa core
163 or
164 .Fa core_fd
165 does not match the current running system.
166 .It Dv G_LP64
167 The calling process is a 32-bit process and the core file referenced by
168 either
169 .Fa core
170 or
171 .Fa core_fd
172 refers to a 64-bit process.
173 .It Dv G_NOTE
174 The ELF notes present in the core file referred to by either
175 .Fa core
176 or
177 .Fa core_fd
178 are corrupt or missing required data.
179 .It Dv G_STRANGE
180 An unanticipated system error occurred while trying to open the core
181 file and create the handle.
182 The value of
183 .Sy errno
184 indicates the system failure.
185 .El
186 .Sh INTERFACE STABILITY
187 .Sy Uncommitted
188 .Sh MT-LEVEL
189 .Sy MT-Safe
190 .Sh SEE ALSO
191 .Xr gcore 1 ,
```



```
192 .Xr open 2 ,
193 .Xr errno 3C ,
194 .Xr libproc 3LIB ,
195 .Xr Pfree 3PROC ,
196 .Xr Pgrab_error 3PROC ,
197 .Xr Prelease 3PROC ,
198 .Xr core 4
```

```
***********************************************************
    1323 Mon Oct 15 13:27:12 2018
new/usr/src/man/man3proc/Pgrab_error.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGRAB_ERROR 3PROC
.Os
.Sh NAME
.Nm Pgrab_error
.Nd get Pgrab error message string
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const char *"
.Fo Pgrab_error
.Fa "int error"
.Fc
.Sh DESCRIPTION
The
.Fn Pgrab_error
function returns a pointer to a human-readable character string
describing the error that occurred.
This function only knows how to translate errors that are stored in
.Fa perr
during a failed call to
.Xr Pgrab 3PROC ,
.Xr Pfgrab_core 3PROC,
.Xr Pgrab_core 3PROC ,
or
.Xr Pgrab_file 3PROC .
.Sh RETURN VALUES
The
.Fn Pgrab_error
function always returns a pointer to a character string that describes
the error that occurred, even if it is an unknown error.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
.Sy MT-Safe
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_file 3PROC
```

```
**********************************************************
    2442 Mon Oct 15 13:27:16 2018
new/usr/src/man/man3proc/Pgrab_file.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PGRAB_FILE 3PROC
.Os
.Sh NAME
.Nm Pgrab_file
.Nd grab and inspect an ELF object
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft struct ps_prochandle
.Fo Pgrab_file
.Fa "const char *fname"
.Fa "int *perr"
.Fc
.Sh DESCRIPTION
The
.Fn Pgrab_file
function creates a handle to the ELF object contained in file
.Fa fname .
This handle is considered an
.Em idle
handle, it allows one to inspect aspects of the ELF contents present in
the handle, for example obtaining CTF information and looking up
symbols.
.Pp
There is no running state associated with this handle nor can there be.
If one intends to control a running process or create a process, see
.Xr Pgrab 3PROC
and
.Xr Pcreate 3PROC
respectively.
To inspect a core file use
.Xr Pgrab_core 3PROC .
.Pp
The
.Fa perr
argument must be a
.Pf non- Dv NULL
pointer which will store a more detailed error in the event that
.Fn Pgrab_file
fails.
A human-readable form of the error can be obtained with
.Xr Pgrab_error 3PROC .
.Pp
When finished with the returned handle,
```



```
.Xr Prelease 3PROC
must be called to clean up resources associated with it.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pgrab_file
function returns a control handle to the process.
Otherwise,
.Dv NULL
is returned and
.Fa perr
is filled in with an error code.
.Sh ERRORS
.Bl -tag -width Er -offset indent
.It Er G_ELF
An unexpected
.Xr libelf 3LIB
failure occurred while processing the file named by
.Fa fname .
.It Er G_FORMAT
The file named by
.Fa fname
is not a valid ELF file.
.It Er G_NOEXEC
The file named by
.Fa fname
does not exist.
.It Er G_STRANGE
An unanticipated system error occurred while trying to grab the
file
.Fa fname
and create the handle.
The value of
.Sy errno
indicates the system failure.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
.Sy MT-Safe
.Sh SEE ALSO
.Xr errno 3C ,
.Xr libelf 3LIB ,
.Xr libproc 3LIB ,
.Xr Pfree 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr Pgrab_error 3PROC ,
.Xr Prelease 3PROC
```

```
***********************************************************
    1216 Mon Oct 15 13:27:21 2018
new/usr/src/man/man3proc/Pisprocdir.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PISPROCIDR 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pisprocdir
  19 .Nd determine if a directory is the /proc directory
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft int
  25 .Fo Pisprocdir
  26 .Fa "struct ps_prochandle *P"
  27 .Fa "const char *dir"
  28 .Fc
  29 .Sh DESECRIPTION
  30 The
  31 .Fn Pisprocdir
  32 function determines whether or not the directory
  33 .Fa dir
  34 is the root of the /proc file-system.
  35 This works across loopback file system (lofs) mounts and chroots.
  36 .Sh RETURN VALUES
  37 Upon successful completion, the
  38 .Fn Pisprocdir
  39 function returns
  40 .Sy 1
  41 if
  42 .Fa dir
  43 is the /proc directory, otherwise if not, it returns
  44 .Sy 0 .
  45 .Sh INTERFACE STABILITY
  46 .Sy Uncommitted
  47 .Sh MT-LEVEL
  48 See
  49 .Sy LOCKING
  50 in
  51 .Xr libproc 3LIB .
  52 .Sh SEE ALSO
  53 .Xr libproc 3LIB ,
  54 .Xr proc 4 ,
  55 .Xr lofs 7FS
```

```
**********************************************************
    1912 Mon Oct 15 13:27:25 2018
new/usr/src/man/man3proc/Pissyscall.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PISSYSCALL 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pissyscall ,
  19 .Nm Pissyscall_prev
  20 .Nd determine if instructions are system call instructions
  21 .Sh LIBRARY
  22 .Lb libproc
  23 .Sh SYNOPSIS
  22 .Lb libproc
  24 .In libproc.h
  25 .Ft int
  26 .Fo Pissyscall
  27 .Fa "struct ps_prochandle *P"
  28 .Fa "uintptr_t addr"
  29 .Fc
  30 .Ft int
  31 .Fo Pissyscall_prev
  32 .Fa "struct ps_prochandle *P"
  33 .Fa "uintptr_t addr"
  34 .Fa "uintptr_t *dst"
  35 .Fc
  36 .Sh DESCRIPTION
  37 The
  38 .Fn Pissyscall
  39 function determines whether or not the instructions at
  40 .Fa addr
  41 in the process handle
  42 .Fa P
  43 corresponds to one of the architecture’s system call instructions.
  44 .Pp
  45 the
  46 .Fn Pissyscall_prev
  47 function determines whether or not the instruction before
  48 .Fa addr
  49 in the process handle
  50 .Fa P
  51 corresponds to one of the architecture’s system call instructions.
  52 If it does, and
  53 .Fa dst
  54 is a
  55 .Pf non- Dv NULL
  56 pointer, then the address of the system call instruction will be copied
  57 into the location pointed to by
  58 .Fa dst .
  59 .Sh RETURN VALUES
```



```
  60 Upon successful completion, the
  61 .Fn Pissyscall
  62 function returns
  63 .Sy non-zero
  64 if
  65 .Fa addr
  66 corresponds to a system call instruction.
  67 Otherwise,
  68 .Sy 0
  69 is returned.
  70 .Pp
  71 Upon successful completion, the
  72 .Fn Pissyscall_prev
  73 function returns
  74 .Sy non-zero
  75 if
  76 .Fa addr
  77 corresponds to a system call instruction and if
  78 .Fa dst
  79 is
  80 .Pf non- Dv NULL ,
  81 .Fa dst
  82 is updated.
  83 Otherwise,
  84 .Sy 0
  85 is returned.
  86 .Sh INTERFACE STABILITY
  87 .Sy Uncommitted
  88 .Sh MT-LEVEL
  89 See
  90 .Sy LOCKING
  91 in
  92 .Xr libproc 3LIB .
  93 .Sh SEE ALSO
  94 .Xr libproc 3LIB
```

```
**********************************************************
    2382 Mon Oct 15 13:27:29 2018
new/usr/src/man/man3proc/Pldt.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLDT 3PROC
.Os
.Sh NAME
.Nm Pldt ,
.Nm proc_get_ldt
.Nd obtain local descriptor table of a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pldt
.Fa "struct ps_prochandle *P"
.Fa "struct ssd *pldt"
.Fa "int nldt"
.Fc
.Ft int
.Fo proc_get_ldt
.Fa "pid_t pid"
.Fa "struct ssd *pldt"
.Fa "int nldt"
.Fc
.Sh DESCRIPTION
The
.Fn Pldt
function reads the local descriptor table (LDT) of the process handle
.Fa P
into the buffer
.Fa pldt .
Up to
.Fa nldt
entries will be read.
.Pp
If either
.Fa pldt
is
.Dv NULL
or
.Fa nldt
is
.Fa 0 ,
then rather than filling in
.Fa pldt ,
only the number of entries currently in the LDT is returned.
.Pp
```



```
The buffer
.Fa pldt
should contain sufficient space for
.Fa nldt
entries.
For example, callers could allocate space as:
.Pp
.Dl pldt = malloc(sizeof (struct ssd) * nldt);
.Pp
For more information on the LDT and the
.Sy struct ssd ,
see
.Xr proc 4 .
.Pp
The
.Fn proc_get_ldt
function is similar to the
.Fn Pldt
function; however, rather than reading from a process handle, it reads
the
.Sy ldt
file from the /proc file system for the process
.Fa pid .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pldt
and
.Fn proc_get_ldt
functions return the number of LDT entries written to
.Fa pldt .
If
.Fa pldt
is
.Dv NULL
or
.Fa nldt
is zero, then no data will be written.
Otherwise,
.Sy -1
is returned.
The
.Fn Pldt
function sets
.Sy errno
to indicate the error that occurred.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Pldt
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
No LDT information is available in the process handle
.Fa P .
.El
.Sh ARCHITECTURE
The
.Fn Pldt
and
.Fn proc_get_ldt
functions are only available on
.Sy x86
```

```
 126 platforms.
 127 .Sh INTERFACE STABILITY
 128 .Sy Uncommitted
 129 .Sh MT-LEVEL
 130 See
 131 .Sy LOCKING
 132 in
 133 .Xr libproc 3LIB .
 134 .Sh SEE ALSO
 135 .Xr libproc 3LIB ,
 136 .Xr proc 4
```

```
**********************************************************
    1609 Mon Oct 15 13:27:35 2018
new/usr/src/man/man3proc/Plmid.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLMID 3PROC
.Os
.Sh NAME
.Nm Plmid
.Nd get the link-map identifier of an address
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plmid
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "Lmid_t *lmidp"
.Fc
.Sh DESCRIPTION
The
.Fn Plmid
function attempts to determine the link-map identifier that corresponds
to the address
.Fa addr
in the process
.Fa P .
If
.Fa addr
does not correspond to an address from an object mapped in by the
run-time link editor, such as anonymous mapping created with
.Xr mmap 2 ,
then there will be no corresponding link-map identifier.
.Pp
.Fa lmidp
must be a
.Pf non- Dv NULL
pointer that will be filled in with the link-map identifier when it is
successfully determined by the run-time link-editor.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plmid
function returns
.Sy 0
and updates
.Fa lmidp
with the link-map identifier of
.Fa addr .
Otherwise,
```



```
.Sy -1
is returned to indicate that the link-map identifier could not be
determined.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    4458 Mon Oct 15 13:27:40 2018
new/usr/src/man/man3proc/Plookup_by_addr.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLOOKUP_BY_ADDR 3PROC
.Os
.Sh NAME
.Nm Plookup_by_addr ,
.Nm Pxlookup_by_addr ,
.Nm Pxlookup_by_addr_resolved ,
.Nm Plookup_by_name ,
.Nm Pxlookup_by_name
.Nd lookup symbols in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plookup_by_addr
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "char *buf"
.Fa "size_t size"
.Fa "Gelf_Sym *symp"
.Fc
.Ft int
.Fo Pxlookup_by_addr
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "char *buf"
.Fa "size_t size"
.Fa "Gelf_Sym *symp"
.Fa "prsyminfo_t *sip"
.Fc
.Ft int
.Fo Pxlookup_by_addr_resolved
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "char *buf"
.Fa "size_t size"
.Fa "Gelf_Sym *symp"
.Fa "prsyminfo_t *sip"
.Fc
.Ft int
.Fo Plookup_by_name
.Fa "struct ps_prochandle *P"
.Fa "const char *object"
.Fa "const char *symbol"
.Fa "GElf_Sym *symp"
```



```
.Fc
.Ft int
.Fo Pxlookup_by_name
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *object"
.Fa "const char *symbol"
.Fa "GElf_Sym *symp"
.Fa "prsyminfo_t *sip"
.Fc
.Sh DESCRIPTION
The
.Fn Plookup_by_addr ,
.Fn Pxlookup_by_addr ,
.Fn Pxlookup_by_addr_resolved ,
.Fn Plookup_by_name ,
and
.Fn Pxlookup_by_name
functions look up symbol information in the process handle
.Fa P
and fill in the ELF symbol information in
.Fa symp
with the found symbol.
Symbols may be looked up both by address and name.
.Pp
The
.Fn Plookup_by_addr
function looks up symbol information corresponding to the address
.Fa addr .
If found, up to
.Fa size
bytes of the symbol's name, including the null terminator will be filled
in to the buffer
.Fa buf .
.Pp
The
.Fn Pxlookup_by_addr
function is identical to the
.Fn Plookup_by_addr
function, except that it also fills in the structure
.Fa sip
with additional information.
The definition of the
.Sy prsyminfo_t
is found in
.Xr libproc 3LIB .
.Pp
The
.Fn Pxlookup_by_addr_resolved
function is similar to the
.Fn Pxlookup_by_addr
function; however, it attempts to resolve the paths present in the
.Sy prsyminfo_t
to an absolute path on the file system.
.Pp
The
.Fn Plookup_by_name
function attempts to look up a symbol based on its name.
The
.Fa object
argument allows the caller to specify a specific object that was mapped
in by the run-time link-editor to search for
.Fa symbol
in.
The system provides three special values which may be passed in for
.Fa object .
```

```
126 The value
127 .Dv PR_OBJ_EXEC
128 refers to the executable’s object (a.out).
129 The value
130 .Dv PR_OBJ_LDSO
131 refers to the object
132 .Sy ld.so.1 .
133 The value
134 .Dv PR_OBJ_EVERY
135 indicates that every object should be searched.
136 .Pp
137 The
138 .Fn Pxlookup_by_name
139 function is similar to the
140 .Fn Plookup_by_name
141 function; however, it allows a link-map identifier,
142 .Fa lmid ,
143 to be specified and also provides additional information about the
144 symbol in the form of the
145 .Sy prsyminfo_t
146 .Fa sip .
147 The specification of
148 .Fa lmid
149 restricts the search for the object named
150 .Fa object
151 and symbol named
152 .Fa symbol
153 to the specified link-map.
154 .Pp
155 There are three special link-map identifiers that may be passed in.
156 The symbol
157 .Dv PR_LMID_EVERY
158 indicates that every link-map should be searched.
159 The symbol
160 .Dv LM_ID_BASE
161 indicates that the base link-map, the one that is used for the
162 executable should be searched.
163 Finally, the symbol
164 .Dv LM_ID_LDSO
165 refers to the link-map that is used by the run-time link editor, ld.so.1.
166 The
167 .Fn Plookup_by_name
168 function behaves like
169 .Fn Pxlookup_by_name
170 when the
171 .Dv PR_LMID_EVERY
172 argument is passed to
173 .Fa lmid ,
174 indicating that every link-map should be searched.
175 .Sh RETURN VALUES
176 Upon successful completion, the
177 .Fn Plookup_by_addr ,
178 .Fn Pxlookup_by_addr ,
179 .Fn Pxlookup_by_addr_resolved ,
180 .Fn Plookup_by_name ,
181 and
182 .Fn Pxlookup_by_name
183 functions return
184 .Sy 0
185 and fill in the symbol information.
186 Otherwise,
187 .Sy -1
188 is returned to indicate that the symbol could not be found.
189 .Sh INTERFACE STABILITY
190 .Sy Uncommitted
191 .Sh MT-LEVEL
```



```
192 See
193 .Sy LOCKING
194 in
195 .Xr libproc 3LIB .
196 .Sh SEE ALSO
197 .Xr elf 3ELF ,
198 .Xr gelf 3ELF ,
199 .Xr libproc 3LIB ,
200 .Xr proc 4
```

```
**********************************************************
    3070 Mon Oct 15 13:27:46 2018
new/usr/src/man/man3proc/Plwp_getasrs.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_GETASRS 3PROC
.Os
.Sh NAME
.Nm Plwp_getasrs ,
.Nm Plwp_setasrs
.Nd get and set SPARCv9 ancillary state registers
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_getasrs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "asrset_t asrs"
.Fc
.Ft int
.Fo Plwp_setasrs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "const asrset_t asrs"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getasrs
and
.Fn Plwp_setasrs
functions get and set the ancillary thread-specific register set of the
thread
.Fa lwpid
in the process handle
.Fa P .
.Pp
The ancillary state registers are only present on 64-bit
.Sy SPARCv9
processes.
They contain information that is specific to the platform and are not included
in the information obtained through functions such as
.Xr Plwp_getregs 3PROC ,
.Xr Plwp_getfpregs 3PROC ,
and
.Xr Plwp_getxregs 3PROC .
.Pp
The
.Fn Plwp_getasrs
```



```
function reads the ancillary registers into
.Fa asrs ,
while the
.Fn Plwp_setasrs
sets the thread's ancillary registers to the values provided by
.Fa asrs .
.Pp
Processes should be stopped prior to obtaining the register state of
individual threads.
Processes may be stopped with
.Xr Pstop 3PROC .
.Pp
The
.Sy asrset_t
structure is described in
.Xr proc 4 .
.Pp
One may not set the register values of a process that is not an active
process, e.g. a process handle that refers to a file or a core file.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getasrs
and
.Fn Plwp_setasrs
functions return
.Sy 0
and get or set the register state.
Otherwise,
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_getasrs
and
.Fn Plwp_setasrs
function will fail if:
.Bl -tag -width Er
.It Er EDATA
The process handle
.Fa P
does not have any ancillary register state information.
.It Er EBUSY
The process handle
.Fa P
refers to a live process and it is not stopped.
.It Er ENOENT
The process handle
.Fa P
refers to a live process and there is no thread with id
.Fa lwpid
or it is not a 64-bit SPARCv9 process.
.It Er EINVAL
The process handle
.Fa P
refers to a core file and there is no thread with id
.Fa lwpid .
.El
.Sh ARCHITECTURE
The
```

```
 126 .Fn Plwp_getasrs
 127 and
 128 .Fn Plwp_setasrs
 129 functions are only available on
 130 64-bit
 131 .Sy SPARCv9
 132 platforms.
 133 .Sh INTERFACE STABILITY
 134 .Sy Uncommitted
 135 .Sh MT-LEVEL
 136 See
 137 .Sy LOCKING
 138 in
 139 .Xr libproc 3LIB .
 140 .Sh SEE ALSO
 141 .Xr libproc 3LIB ,
 142 .Xr Plwp_getfpregs 3PROC ,
 143 .Xr Plwp_getregs 3PROC ,
 144 .Xr Plwp_setregs 3PROC ,
 145 .Xr Plwp_setregs 3PROC ,
 146 .Xr Pstop 3PROC ,
 147 .Xr proc 4
```

```
**********************************************************
    2073 Mon Oct 15 13:27:51 2018
new/usr/src/man/man3proc/Plwp_getname.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2018 Joyent, Inc.
.\"
.Dd August 31, 2018
.Dt PLWP_GETNAME 3PROC
.Os
.Sh NAME
.Nm Plwp_getname
.Nd get thread name
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.In libproc.h
.Ft int
.Fo Plwp_getname
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "char *buf"
.Fa "size_t bufsize"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getname
function returns the thread name in the supplied buffer, from the
thread
.Fa lwpid
in the process handle
.Fa P .
.Pp
If no thread name is set, the buffer is set to the empty string.
The buffer should be at least
.Dv THREAD_NAME_MAX
in size.
.Pp
The
.Fn Plwp_getname
function only works on process handles that refer to active processes
and core files, it does not work on process handles that refer to
individual files.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getname
function returns 0.
Otherwise,
.Sy -1
is returned,
.Sy errno
is set to indicate the error.
The buffer may be modified even if an error is returned.
.Sh ERRORS
```



```
For a full list of possible errors also see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_getname
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
refers to a file handle obtained through
.Xr Pgrab_file 3PROC .
.It Er EINVAL
The process handle
.Fa P
refers to a core file and the specified thread does not exist.
.It Er ENOENT
The process handle
.Fa P
refers to an active process and the specified thread does not exist.
.Fa P
.It Er ENAMETOOLONG
The buffer is not sufficient to hold the thread name.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2152 Mon Oct 15 13:27:53 2018
new/usr/src/man/man3proc/Plwp_getpsinfo.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_GETPSINFO 3PROC
.Os
.Sh NAME
.Nm Plwp_getpsinfo
.Nd get thread specific ps information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_getpsinfo
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "lwpsinfo_t *lps"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getpsinfo
function
looks up the thread-specific
.Xr ps 1
information for the thread specified by
.Fa lwpid
in the process handle
.Fa P .
The caller should provide a pointer to an
.Sy lwpsinfo_t ,
.Fa lps ,
whose definition can be found in
.Xr proc 4 .
.Fa lps
will be filled in with information such as the thread’s id, its current
state, priority, and run-time.
.Pp
The
.Fn Plwp_getpsinfo
function only works on process handles that refer to active processes
and core files, it does not work on process handles that refer to
individual files.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getpsinfo
function returns
.Sy 0
and
```



```
.Fa lps
is filled in with the
thread-specific
.Xr ps 1
information.
Otherwise,
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors also see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_getpsinfo
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
refers to a file handle obtained through
.Xr Pgrab_file 3PROC .
.It Er EINVAL
The process handle
.Fa P
refers to a core file and the specified thread does not exist.
.It Er ENOENT
The process handle
.Fa P
refers to an active process and the specified thread does not exist.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr ps 1 ,
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    3354 Mon Oct 15 13:27:57 2018
new/usr/src/man/man3proc/Plwp_getregs.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_GETREGS 3PROC
.Os
.Sh NAME
.Nm Plwp_getfpregs ,
.Nm Plwp_setfpregs ,
.Nm Plwp_getregs ,
.Nm Plwp_setregs
.Nd get and set thread registers
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_getfpregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "prfpregset_t *fpregs"
.Fc
.Ft int
.Fo Plwp_setfpregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "const prfpregset_t fpregs"
.Fc
.Ft int
.Fo Plwp_getregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "prpregset_t *gregs"
.Fc
.Ft int
.Fo Plwp_setregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "const prpregset_t gregs"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getregs ,
.Fn Plwp_setregs ,
.Fn Plwp_getfpregs ,
and
.Fn Plwp_setfpregs
functions allow one to get and set the general purpose and floating
point registers from the thread
```



```
.Fa lwpid
in the process handle
.Fa P .
.Pp
The
.Fn Plwp_getfpregs
function updates the structure pointed to by
.Fa fpregs
with the state and values of the floating point registers of the thread
specified by
.Fa lwpid .
.Pp
The
.Fn Plwp_setfpregs
function updates the floating point registers of the thread specified by
.Fa lwpid
to the register state contained in
.Fa fpregs .
.Pp
The
.Fn Plwp_getregs
function updates the structure pointed to by
.Fa gregs
with the state and values of the general purpose registers of the thread
specified by
.Fa lwpid.
.Pp
The
.Fn Plwp_setregs
function updates the general purpose registers of the thread specified
by
.Fa lwpid
to the register state contained in
.Fa gregs .
.Pp
Processes must be stopped before obtaining the register state of
individual threads.
Processes may be stopped with
.Xr Pstop 3PROC .
The structures used for registers are described in
.Xr proc 4
and their defintions may be found in
.In sys/regset.h .
The definitions of these structures varies based on the architecture of
the system and the running process.
.Pp
One may not set the register values of a process that is not an active
process, e.g. a process handle that refers to a file or a core file.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getregs ,
.Fn Plwp_setregs ,
.Fn Plwp_getfpregs ,
and
.Fn Plwp_setfpregs
functions return
.Sy 0
and obtain or set the register state.
Otherwise,
.Sy -1
is returned,
.Sy errno
is set to indicate the error, and the register state is not updated nor
are the data pointers changed.
.Sh ERRORS
For a full list of possible errors also see the
```

```
 126 .Sy DIAGNOSTICS
 127 section in
 128 .Xr proc 4 .
 129 .Pp
 130 The
 131 .Fn Plwp_getregs ,
 132 .Fn Plwp_setregs ,
 133 .Fn Plwp_getfpregs ,
 134 and
 135 .Fn Plwp_setfpregs
 136 will fail if:
 137 .Bl -tag -width Er
 138 .It Er EBUSY
 139 The process handle
 140 .Fa P
 141 is not currently stopped.
 142 .It Er ENOENT
 143 There is no thread in
 144 .Fa P
 145 with id
 146 .Fa lwpid .
 147 .El
 148 .Sh INTERFACE STABILITY
 149 .Sy Uncommitted
 150 .Sh MT-LEVEL
 151 See
 152 .Sy LOCKING
 153 in
 154 .Xr libproc 3LIB .
 155 .Sh SEE ALSO
 156 .Xr libproc 3LIB ,
 157 .Xr proc 4
```

```
**********************************************************
    2673 Mon Oct 15 13:28:01 2018
new/usr/src/man/man3proc/Plwp_getspymaster.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_GETSPYMASTER 3PROC
.Os
.Sh NAME
.Nm Plwp_getspymaster
.Nd get agent LWP spy master information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_getspymaster
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "psinfo_t *ps"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getspymaster
function returns information about the spy master that corresponds to
the agent LWP for the thread
.Fa lwpid
in the process handle
.Fa P .
.Pp
The agent LWP allows another process to inject actions into the target process.
When an agent LWP is created, it leverages an existing thread in the process and
it also creates a record of whom created the agent, which is called the spy
master.
For more information on the agent LWP and the spy master, see
.Xr proc 4 .
.Pp
If the thread identified
.Fa lwpid
has an agent LWP, the corresponding ps information
.Po see
.Xr proc 4
for the definition of the
.Sy psinfo_t
.Pc
will be filled into
.Fa ps .
.Pp
Note, process handles that correspond to a file, created by
.Xr Pgrab_file 3PROC ,
cannot have an agent LWP created for them and thus cannot have any spy
```



```
master information.
In addition, core files from older releases may not have any data on the spy
master.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getspymaster
returns
.Sy 0
and updates
.Fa ps .
Otherwise, it returns
.Sy -1 ,
sets
.Sy errno ,
and
.Fa ps
is not modified.
.Sh ERRORS
For a full list of possible errors also see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_getpsinfo
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
refers to a file handle obtained through
.Xr Pgrab_file 3PROC
or
.Fa P
does not have any information about the spy master.
.It Er EINVAL
The process handle
.Fa P
refers to a core file and the specified thread does not exist.
.Pp
The thread,
.Fa lwpid
does not have an active agent,
.Dv PR_AGENT
is not set in the
.Sy pr_flags
member of the thread’s status information.
.It Er ENOENT
The process handle
.Fa P
refers to an active process and the specified thread does not exist.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    3069 Mon Oct 15 13:28:04 2018
new/usr/src/man/man3proc/Plwp_getxregs.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_GETXREGS 3PROC
.Os
.Sh NAME
.Nm Plwp_getxregs ,
.Nm Plwp_setxregs
.Nd get and set extended register state
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_getxregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "prxregset_t *xregs"
.Fc
.Ft int
.Fo Plwp_setxregs
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "const prxregset_t *xregs"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_getxregs
and
.Fn Plwp_setxregs
functions get and set the extended register state of the thread
.Fa lwpid
in the process handle
.Fa P .
.Pp
The extended register state is defined by the architecture.
These registers may refer to optional registers that have become common on the
platform, but are not part of the standard ABI and thus not covered by
functions such as
.Xr Plwp_getregs 3PROC
and
.Xr Plwp_getfpregs 3PROC .
.Pp
The
.Fn Plwp_getxregs
function gets the extended register state information and places it into
.Fa xregs .
Where as the
```



```
.Fn Plwp_setxregs
function sets the extended register state information in
.Fa xregs
for the process handle
.Fa P .
.Pp
Processes must be stopped prior to obtaining the register state of
individual threads.
Processes may be stopped with
.Xr Pstop 3PROC .
.Pp
The
.Sy prxregset_t
structure is described in
.Xr proc 4 .
.Pp
One may not set the register values of a process that is not an active
process, e.g. a process handle that refers to a file or a core file.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_getxregs
and
.Fn Plwp_setxregs
functions return
.Sy 0
and get or set the register state.
Otherwise,
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_getxregs
and
.Fn Plwp_setxregs
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
The process handle
.Fa P
does not have any extended register state information.
.It Er EBUSY
The process handle
.Fa P
refers to a live process and it is not stopped.
.It Er ENOENT
The process handle
.Fa P
refers to a live process and there is no thread with id
.Fa lwpid .
.It Er EINVAL
The process handle
.Fa P
refers to a core file and there is no thread with id
.Fa lwpid .
.El
.Sh ARCHITECTURE
The
.Fn Plwp_getxregs
and
```

```
 126 .Fn Plwp_setxregs
 127 functions are only available on
 128 .Sy SPARC
 129 platforms.
 130 .Sh INTERFACE STABILITY
 131 .Sy Uncommitted
 132 .Sh MT-LEVEL
 133 See
 134 .Sy LOCKING
 135 in
 136 .Xr libproc 3LIB .
 137 .Sh SEE ALSO
 138 .Xr libproc 3LIB ,
 139 .Xr Plwp_getfpregs 3PROC ,
 140 .Xr Plwp_getregs 3PROC ,
 141 .Xr Plwp_setfpregs 3PROC ,
 142 .Xr Plwp_setregs 3PROC ,
 143 .Xr Pstop 3PROC ,
 144 .Xr proc 4
```

```
**********************************************************
    2443 Mon Oct 15 13:28:08 2018
new/usr/src/man/man3proc/Plwp_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_ITER 3PROC
.Os
.Sh NAME
.Nm Plwp_iter ,
.Nm Plwp_iter_all
.Nd iterate over threads
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_iter
.Fa "struct ps_prochandle *P"
.Fa "proc_lwp_f *func",
.Fa "void *data"
.Fc
.Ft int
.Fo Plwp_iter_all
.Fa "struct ps_prochandle *P"
.Fa "proc_lwp_all_f *func"
.Fa "void *data"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_iter
and
.Fn Plwp_iter_all
functions iterates over threads in the given process handle
.Fa P .
The
.Fn Plwp_iter
function iterates over all active threads, where as the
.Fn Plwp_iter_all
function iterates over both active threads and zombie threads -- threads
waiting to be reaped.
.Pp
For each thread, the callback function
.Fa func
is called with the pointer to the private data argument,
.Fa data ,
and the thread's
.Sy lwpstatus_t
structure.
In the case of the
.Fn Plwp_iter_all
```



```
function, the thread's
.Sy lwpsinfo_t
is also included.
.Pp
The return value of
.Fa func
controls whether or not iteration continues.
If
.Fa func
returns
.Sy 0 ,
then both functions will continue iteration.
However, if
.Fa func
returns non-zero, then iteration will halt and that value will be used
as the return value of the
.Fn Plwp_iter
and
.Fn Plwp_iter_all
functions.
Because both functions return
.Sy -1
on internal failure, it is recommended that the callback function does
not return
.Sy -1
to indicate an error so that the caller may distinguish between the
failure of the callback function and the failure of the
.Fn Plwp_iter
and
.Fn Plwp_iter_all
functions.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_iter
and
.Fn Plwp_iter_all
functions return
.Sy 0 .
Otherwise, if there was an internal error or there is no thread data, then
.Sy -1
is returned.
Otherwise, if the callback function
.Fa func
returns non-zero, then its return value will be returned instead.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB
```

```
**********************************************************
    3993 Mon Oct 15 13:28:10 2018
new/usr/src/man/man3proc/Plwp_stack.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PLWP_STACK 3PROC
.Os
.Sh NAME
.Nm Plwp_stack ,
.Nm Plwp_alt_stack ,
.Nm Plwp_main_stack ,
.Nm Lstack ,
.Nm Lalt_stack ,
.Nm Lmain_stack
.Nd get thread stack information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Plwp_stack
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "stack_t *stkp"
.Fc
.Ft int
.Fo Plwp_alt_stack
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "stack_t *stkp"
.Fc
.Ft int
.Fo Plwp_main_stack
.Fa "struct ps_prochandle *P"
.Fa "lwpid_t lwpid"
.Fa "stack_t *stkp"
.Fc
.Ft int
.Fo Lalt_stack
.Fa "struct ps_lwphandle *L"
.Fa "stack_t *stkp"
.Fc
.Ft int
.Fo Lmain_stack
.Fa "struct ps_lwphandle *L"
.Fa "stack_t *stkp"
.Fc
.Ft int
.Fo Lstack
.Fa "struct ps_lwphandle *L"
```



```
.Fa "stack_t *stkp"
.Fc
.Sh DESCRIPTION
The
.Fn Plwp_stack ,
.Fn Plwp_alt_stack ,
and
.Fn Plwp_main_stack
functions obtain information about the size and address of the stacks
for the thread identified by
.Fa lwpid
in the process handle
.Fa P .
.Pp
Each thread in a process has its own stack which is used both for
maintaining function call sequences and the storing of local variables.
A thread may also configure a different stack to handle specific
signals.
This stack is often called the
.Em alternate stack .
Whether or not the alternate stack is used may be controlled through the
.Xr sigaction 2
and
.Xr sigaltstack 2
functions .
.Pp
The
.Fn Plwp_stack
function fills in
.Fa stkp
with the information about the thread’s currently executing stack,
whether the alternate or main one.
.Pp
The
.Fn Plwp_alt_stack
function fills in
.Fa stkp
with the information about the thread’s alternate stack, if it’s
configured.
.Pp
The
.Fn Plwp_main_stack
function fills in
.Fa stkp
with the information about the thread’s main stack, regardless of if there
is an alternate stack or it is executing one.
.Pp
Process handles that refer to files, obtained through
.Xr Pgrab_file 3PROC ,
do not have any stack information and these functions will always fail
on them.
.Pp
The
.Fn Lalt_stack ,
.Fn Lmain_stack ,
and
.Fn Lstack
functions are identical to the
.Fn Plwp_alt_stack ,
.Fn Plwp_main_stack ,
and
.Fn Plwp_main_stack
functions, except rather than specifying a thread to operate on, they
operate on the thread handle
.Fa L ,
which specifies the thread to operate on.
```

```
.Sh RETURN VALUES
Upon successful completion, the
.Fn Plwp_stack ,
.Fn Plwp_alt_stack ,
.Fn Plwp_main_stack ,
.Fn Lalt_stack ,
.Fn Lmain_stack ,
and
.Fn Lstack
functions return
.Sy 0
and fills in
.Fa stkp
with information about the appropriate stack.
Otherwise,
.Sy -1
is returned,
.Sy errno
is updated with the error, and
.Fa stkp
is not modified.
.Sh ERRORS
For a full list of possible errors also see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Plwp_stack ,
.Fn Plwp_alt_stack ,
and
.Fn Plwp_main_stack
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
The process handle
.Fa P
refers to a grabbed file, not an active process or core.
.It Er EINVAL
The process handle
.Fa P
refers to a core file and the specified thread does not exist.
.It Er ENOENT
The process handle
.Fa P
refers to an active process and the specified thread does not exist.
.El
.Pp
The
.Fn Plwp_alt_stack
and
.Fn Lalt_stack
functions will fail if:
.Bl -tag -width Er
.It Er ENODATA
The thread identified by
.Fa lwpid
did not have an alternate stack enabled.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
```



```
.Sh SEE ALSO
.Xr sigaction 2 ,
.Xr sigaltstack 2 ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    3924 Mon Oct 15 13:28:15 2018
new/usr/src/man/man3proc/Pmapping_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PMAPPING_ITER 3PROC
.Os
.Sh NAME
.Nm Pmapping_iter ,
.Nm Pmapping_iter_resolved ,
.Nm Pobject_iter ,
.Nm Pobject_iter_resolved
.Nd iterate over process mappings and objects
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pmapping_iter
.Fa "struct ps_prochandle *P"
.Fa "proc_map_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Pmapping_iter_resolved
.Fa "struct ps_prochandle *P"
.Fa "proc_map_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Pobject_iter
.Fa "struct ps_prochandle *P"
.Fa "proc_map_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Pobject_iter_resolved
.Fa "struct ps_prochandle *P"
.Fa "proc_map_f *func"
.Fa "void *data"
.Fc
.Sh DESCRIPTION
The
.Fn Pmapping_iter
and
.Fn Pmapping_iter_resolved
functions iterate over the memory mappings in the process represented by
.Fa P.
.Pp
For each memory mapping, the callback function
```



```
.Fa func
will be invoked and it will be passed the
.Fa data
argument,
the
.Sy prmap_t
structure defined from
.Xr proc 4 ,
and a name of the mapping.
The way that the name is obtained varies based on whether one calls
.Fn Pmapping_iter
or
.Fn Pmapping_iter_resolved .
In both cases, the dynamic linker is consulted to determine the file
name for the mapping, if it's known.
If the name is unknown, for example an anonymous mapping, then the
.Dv NULL
pointer is passed in for the name.
In the case of the
.Fn Pmapping_iter_resolved
function the system tries to resolve it to a complete file system path.
If that fails, it falls back to the information from the dynamic linker,
before returning
.Dv NULL
in the case of not being able to find any name.
For more information on the
signature of the
.Ft proc_map_f ,
see
.Xr libproc 3LIB .
.Pp
The return value of
.Fa func
controls whether or not iteration continues.
If
.Fa func
returns
.Sy 0
then iteration continues.
If
.Fa func
returns non-zero then iteration will halt and the value will be
returned to the caller.
Because
.Sy -1
indicates internal failure, it is recommended that the callback function not
return
.Sy -1
to indicate an error itself.
This allows the caller to distinguish between failure of the callback function
versus failure of the
.Fn Pmapping_iter
and
.Fn Pmapping_iter_resolved
functions.
.Pp
The
.Fn Pobject_iter
and
.Fn Pobject_iter_resolved
functions are similar to the
.Fn Pmapping_iter
and
.Fn Pmapping_iter_resolved
functions.
Except, rather than iterating over every mapping, they iterate over the objects
```

```
126 that the process has loaded by the dynamic linker.
127 For example, an anonymous mapping will show up when iterating mappings, but will
128 not show up when iterating objects.
129 Further, while most dynamic shared objects have multiple mappings for the text
130 and data sections, there will only be a single object that is iterated over.
131 .Pp
132 The distinction between the
133 .Fn Pobject_iter
134 and
135 .Fn Pobject_iter_resolved
136 functions is identical to the difference in name resolution between the
137 .Fn Pmapping_iter
138 and
139 .Fn Pmapping_iter_resolved
140 functions.
141 .Sh RETURN VALUES
142 Upon successful completion, the
143 .Fn Pmapping_iter ,
144 .Fn Pmapping_iter_resolved
145 .Fn Pobject_iter ,
146 and
147 .Fn Pobject_iter_resolved
148 functions return
149 .Sy 0.
150 Otherwise, if there was an internal error then
151 .Sy -1
152 is returned.
153 Otherwise, if the callback function
154 .Fa func
155 returns non-zero, then its return value will be returned instead.
156 .Sh INTERFACE STABILITY
157 .Sy Uncommitted
158 .Sh MT-LEVEL
159 See
160 .Sy LOCKING
161 in
162 .Xr libproc 3LIB .
163 .Sh SEE ALSO
164 .Xr libproc 3LIB ,
165 .Xr proc 4
```

```
**********************************************************
    2194 Mon Oct 15 13:28:20 2018
new/usr/src/man/man3proc/Pobjname.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt POBJNAME 3PROC
.Os
.Sh NAME
.Nm Pobjname ,
.Nm Pobjname_resolved
.Nd turn a virtual address into its mapped object
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "char *"
.Fo Pobjname
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "char *buffer"
.Fa "size_t bufsize"
.Fc
.Ft char *
.Fo Pobjname_resolved
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fa "char *buffer"
.Fa "size_t bufsize"
.Fc
.Sh DESCRIPTION
The
.Fn Pobjname
and
.Fn Pobjname_resolved
functions attempt to determine the underlying mapped object that
contains the virtual address
.Fa addr
in the process handle
.Fa P .
.Pp
A program consists of multiple memory mappings.
Some are provided by the system, such as the stack and the heap.
While others are created through explicit calls to
.Xr mmap 2
or brought in by the run-time link-editor due to dependencies
specified in binaries and libraries.
.Pp
If
.Fa addr
is contained in a mapping, then up to
```



```
.Fa bufsize
characters, including the null terminator,
of the name of the corresponding object will be written into
.Fa buffer .
The
.Fn Pobjname_resolved
function attempts to resolve the object to a full file system path.
If the full file-system path cannot be determined, then it will fall back
to the name that the run-time link-editor has for that mapping, which is
the behavior of
.Fn Pobjname .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pobjname
and
.Fn Pobjname_resolved
functions return
.Fa buffer .
Otherwise,
.Dv NULL
is returned to indicate the object name could not be found.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr mmap 2 ,
.Xr libproc 3LIB ,
.Xr Pobject_iter 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1791 Mon Oct 15 13:28:24 2018
new/usr/src/man/man3proc/Pplatform.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PPLATFORM 3PROC
.Os
.Sh NAME
.Nm Pplatform
.Nd get platform string
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "char *"
.Fo Pplatform
.Fa "struct ps_prochandle *P"
.Fa "char *buffer"
.Fa "size_t bufsize"
.Fc
.Sh DESCRIPTION
The
.Fn Pplatform
function determines the name of the platform for the process handle
.Fa P .
Up to
.Fa bufsize
characters, including the null terminator, will be copied into
.Fa buffer .
.Pp
The platform is the value reported by the
.Sy -s
option to
.Xr uname 1 .
For example, on x86 systems, the value will be
.Sy i86pc .
.Pp
Note, process handles that correspond to a file, created by
.Xr Pgrab_file 3PROC ,
will not report a platform and the
.Fn Pplatform
function will fail.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pplatform
function returns
.Fa buffer .
Otherwise,
.Dv NULL
is returned,
```



```
.Sy errno
is set, and
.Fa buffer
is not updated.
.Sh ERRORS
The
.Fn Pplatform
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
refers to a core file and there is no
.Sy NT_PLATFORM
ELF note available.
.It Er EFAULT
.Fa P
refers to a live process and
.Fa buffer
is an invalid address.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr uname 1 ,
.Xr sysinfo 2 ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
***********************************************************
    1471 Mon Oct 15 13:28:27 2018
new/usr/src/man/man3proc/Ppltdest.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PPLTDEST 3PROC
.Os
.Sh NAME
.Nm Ppltdest
.Nd determine PLT destination symbol
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const char *"
.Fo Ppltdest
.Fa "struct ps_prochandle *P"
.Fa "uintptr_t addr"
.Fc
.Sh DESCRIPTION
The
.Fn Ppltdest
function determines if the address at
.Fa addr
in the process handle
.Fa P
corresponds to an entry in the procedure linkage table (PLT) and if so,
returns a pointer to a null-terminated character string that contains
the symbol’s name.
The returned pointer is not valid after any other calls to function in
.Xr libproc 3LIB .
The
.Fn Ppltdest
is also a useful way to determine if
.Fa addr
corresponds to an address in the PLT.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Ppltdest
returns a pointer to a character string with the name.
Otherwise,
.Dv NULL
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
```



```
.Sh SEE ALSO
.Xr libproc 3LIB
.Rs
.%T Linkers and Libraries Guide
.Re
```

```
**********************************************************
    1596 Mon Oct 15 13:28:29 2018
new/usr/src/man/man3proc/Ppriv.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PPRIV 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Ppriv ,
  19 .Nm Ppriv_free
  20 .Nd get and free process privilege sets
  21 .Sh LIBRARY
  22 .Lb libproc
  23 .Sh SYNOPSIS
  22 .Lb libproc
  24 .In libproc.h
  25 .Ft int
  26 .Fo Ppriv
  27 .Fa "struct ps_prochandle *P"
  28 .Fa "prpriv_t **pprv"
  29 .Fc
  30 .Ft void
  31 .Fo Ppriv_free
  32 .Fa "struct ps_prochandle *P"
  33 .Fa "prpriv_t *prv"
  34 .Fc
  35 .Sh DESCRIPTION
  36 The
  37 .Fn Ppriv
  38 function obtains the privilege set of the process handle
  39 .Fa P .
  40 The privilege set, if it exists, will be dynamically allocated and a
  41 pointer to it will be placed in
  42 .Fa pprv .
  43 It must be released with a call to
  44 .Fn Ppriv_free .
  45 The definition of the
  46 .Sy prpriv_t
  47 structure is documented in
  48 .Xr proc 4 .
  49 .Pp
  50 The
  51 .Fn Ppriv_free
  52 function releases the storage in
  53 .Fa prv
  54 that was allocated as a result of calling
  55 .Fn Ppriv .
  56 .Sh RETURN VALUES
  57 Upon successful completion, the
  58 .Fn Ppriv
  59 function returns
```



```
  60 .Sy 0
  61 and
  62 .Fa pprv
  63 is updated with a pointer to the allocated privilege set.
  64 Otherwise,
  65 .Sy -1
  66 is returned and
  67 .Fa pprv
  68 is not updated.
  69 .Sh INTERFACE STABILITY
  70 .Sy Uncommitted
  71 .Sh MT-LEVEL
  72 See
  73 .Sy LOCKING
  74 in
  75 .Xr libproc 3LIB .
  76 .Sh SEE ALSO
  77 .Xr libproc 3LIB ,
  78 .Xr proc 4 ,
  79 .Xr privileges 5
```

```
***********************************************************
    1263 Mon Oct 15 13:28:30 2018
new/usr/src/man/man3proc/Ppsinfo.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PPSINFO 3PROC
.Os
.Sh NAME
.Nm Ppsinfo
.Nd get process ps information
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const psinfo_t *"
.Fo Ppsinfo
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Ppsinfo
function returns the process handle
.Fa P Ns ’s
.Xr ps 1
information.
The
.Sy psinfo_t
structure that is returned is defined in
.Xr proc 4
and is valid until a subsequent call to
.Xr Prelease 3PROC .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Ppsinfo
function returns a pointer to the process ps information.
Otherwise,
.Dv NULL
is returned to indicate that it could not be found.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr ps 1 ,
.Xr libproc 3LIB ,
.Xr Prelease 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1422 Mon Oct 15 13:28:31 2018
new/usr/src/man/man3proc/Prd_agent.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PRD_AGENT 3PROC
.Os
.Sh NAME
.Nm Prd_agent
.Nd get librtld_db agent
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "rd_agent_t *"
.Fo Prd_agent
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Prd_agent
function returns a pointer to an agent suitable for use with the
run-time link editing database library,
.Xr librtld_db 3LIB ,
corresponding to the process handle
.Fa P .
.Pp
The returned agent is valid until either the process handle executes a
new process image through the
.Xr exec 2
family of calls or the process handle
.Fa P
is released through a call to
.Xr Prelease 3PROC .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Prd_agent
function returns a pointer to the librtld_db agent.
Otherwise, it returns
.Dv NULL
to indicate failure.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr exec 2 ,
```



```
.Xr libproc 3LIB ,
.Xr librtld_db 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2574 Mon Oct 15 13:28:31 2018
new/usr/src/man/man3proc/Pread.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PREAD 3PROC
.Os
.Sh NAME
.Nm Pread ,
.Nm Pread_string
.Nd read data from a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft ssize_t
.Fo Pread
.Fa "struct ps_prochandle *P"
.Fa "void *buf"
.Fa "size_t nbytes"
.Fa "uintptr_t address"
.Fc
.Ft ssize_t
.Fo Pread_string
.Fa "struct ps_prochandle *P"
.Fa "char *buf"
.Fa "size_t nbytes"
.Fa "uintptr_t address"
.Fc
.Sh DESCRIPTION
The
.Fn Pread
function reads data from the process handle
.Fa P
starting at
.Fa address
in the address space of the process and reads at most
.Fa nbytes
of data into
.Fa buf
and is logically analogous to the
.Xr pread 2
function.
.Pp
For live processes, this function is equivalent to reading from the /proc file
system
.Sy as
file for the process.
For core files and file handles, it reads and writes from the logical address
space and not the corresponding offset of the file itself.
```



```
For example, a core file contains a sparse representation of the address space
of a crashed process and unmapped regions are not present in the file.
However,
.Fa address
still refers to the virtual addresses that were present at run-time and
not those in the core file.
.Pp
The
.Fn Pread_string
function is similar to the
.Fn Pread
function, except that it attempts to interpret
.Fa address
as a null terminated character string and will stop reading characters
into
.Fa buf
if either
.Fa nbytes
has been read or a null terminator is encountered.
The resulting data in
.Fa buf
will always be null terminated, even if no null terminator was found in
the first
.Fa nbytes
of data.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pread
and
.Fn Pread_string
functions return a non-negative integer indicating the number of bytes
actually read.
Otherwise, the functions return
.Sy -1
and set
.Sy errno
to indicate the error.
.Sh ERRORS
For a full list of possible errors also see the
.Sy DIAGNOSTICS
section in
.Xr proc 4
and
the
.Sy ERRORS
section in
.Xr pread 2 .
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr pread 2 ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2792 Mon Oct 15 13:28:31 2018
new/usr/src/man/man3proc/Prelease.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PRELEASE 3PROC
.Os
.Sh NAME
.Nm Prelease ,
.Nm Pfree
.Nd release a process control handle
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Prelease
.Fa "struct ps_prochandle *P"
.Fa "int flags"
.Fc
.Ft void
.Fo Pfree
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Prelease
function is used to release all of the resources associated with a
.Nm libproc
handle.
It is suitable for handles to core files, created processes, and grabbed
processes from the
.Xr Pgrab_core 3PROC ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
and
.Xr Pgrab_file 3PROC
functions.
.Pp
After calling the
.Fn Prelease
function, all data that was returned via the handle will no longer be
valid.
For example, the data from calls to
.Xr Pctlfd 3PROC ,
.Xr Pgetauxvec 3PROC ,
.Xr Pstatus 3PROC ,
and others.
.Pp
The behavior of the released process is controlled by the
```



```
.Fa flags
argument.
By default, if no flags are passed, then the process represented by
.Fa P
will be set running if it was created by
.Xr Pcreate 3PROC
or if it was not originally stopped or set to stop in /proc.
The following values may be passed in to the
.Fa flags
argument.
Multiple flags should be be combined with a bitwise-inclusive-OR.
.Bl -tag -width Er -offset indent
.It Dv PRELEASE_CLEAR
When releasing the process, clear all tracing flags that are set on the
process.
.It Dv PRELEASE_RETAIN
When releasing the process, retain all tracing flags that are currently
active on the process.
.It Dv PRELEASE_HANG
Leave the process stopped.
It will not resume execution unless it is explicitly enabled with
.Xr prun 1
or another process explicitly enables it.
.It Dv PRELEASE_KILL
Release the process and terminate it with
.Dv SIGKILL .
This option takes precedence over all other values that may be passed in to
.Fa flags .
.El
.Pp
The
.Fn Pfree
function is similar to the
.Fn Prelease
function in that it frees the resources associated with the process
handle
.Fa P ;
however, unlike the
.Fn Prelease
function, it does not handle any logic to change or set the grabbed processes
state.
In general, prefer
.Fn Prelease
to
.Fn Pfree .
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr prun 1 ,
.Xr libproc 3LIB ,
.Xr Pcreate 3PROC ,
.Xr Pgrab 3PROC ,
.Xr Pgrab_core 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1699 Mon Oct 15 13:28:31 2018
new/usr/src/man/man3proc/Preopen.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PREOPEN 3PROC
.Os
.Sh NAME
.Nm Preopen
.Nd reopen a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Preopen
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Preopen
function is used to regain control of the process represented by the
handle
.Fa P .
A loss of control is indicated by the
.Xr Pstatus 3PROC
function returning the value
.Dv PS_LOST .
This may occur when the controlled process performs an
.Xr exec 2
of a setuid or setgid binary or one where the controlling process cannot
read the object file.
For more information on this, see the
.Sy PROGRAMMING NOTES
section of
.Xr proc 4 .
.Pp
If successful, the controlling process will obtain control of the
process specified by
.Fa P .
If it fails, the controlling process should release the handle with
.Xr Prelease 3PROC .
Note there are occasions where due to permissions it may not be possible
to obtain control again.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Preopen
function returns
.Sy 0.
Otherwise,
```



```
.Sy -1
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr exec 2 ,
.Xr libproc 3LIB ,
.Xr Prelease 3PROC ,
.Xr Pstatus 3PROC ,
.Xr proc 4
```

```
***********************************************************
    1397 Mon Oct 15 13:28:32 2018
new/usr/src/man/man3proc/Preset_maps.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PRESET_MAPS 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Preset_maps
  19 .Nd reset memory mapping data after exec
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft void
  25 .Fo Preset_maps
  26 .Fa "struct ps_prochandle *P"
  27 .Fc
  28 .Sh DESCRIPTION
  29 The
  30 .Fn Preset_maps
  31 function resets all of the mapping data, cached symbol tables, debug
  32 information, CTF, and more in the process handle
  33 .Fa P .
  34 The map information will rebuild itself on the execution of another
  35 .Xr libproc 3LIB
  36 function that requires the information.
  37 This function should be called any time after a process handle performs
  38 an
  39 .Xr exec 2 .
  40 As such, this interface is only relevant to the manipulation of live
  41 processes and not core files or ELF files.
  42 For more information, see the
  43 .Sy PROGRAMMING NOTES
  44 section in
  45 .Xr libproc 3LIB .
  46 .Sh INTERFACE STABILITY
  47 .Sy Uncommitted
  48 .Sh MT-LEVEL
  49 See
  50 .Sy LOCKING
  51 in
  52 .Xr libproc 3LIB .
  53 .Sh SEE ALSO
  54 .Xr exec 2 ,
  55 .Xr libproc 3LIB ,
  56 .Xr proc 4
```

```
**********************************************************
    1649 Mon Oct 15 13:28:33 2018
new/usr/src/man/man3proc/Psecflags.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2016, Richard Lowe.
.\"
.Dd June 06, 2016
.Dt PSECFLAGS 3PROC
.Os
.Sh NAME
.Nm Psecflags ,
.Nm Psecflags_free
.Nd get and free process security flags
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psecflags
.Fa "struct ps_prochandle *P"
.Fa "prsecflags_t **psf"
.Fc
.Ft void
.Fo Psecflags_free
.Fa "struct ps_prochandle *P"
.Fa "prsecflags_t *psf"
.Fc
.Sh DESCRIPTION
The
.Fn Psecflags
function obtains the security flags of the process handle
.Fa P .
The security flags structure will be dynamically allocated and a pointer to it
will be placed in
.Fa psf .
It must be released with a call to
.Fn Psecflags_free .
The definition of the
.Sy prsecflags_t
structure is documented in
.Xr proc 4 .
.Pp
The
.Fn Psecflags_free
function releases the storage in
.Fa psf
that was allocated as a result of calling
.Fn Psecflags .
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psecflags
function returns
```



```
.Sy 0
and
.Fa psf
is updated with a pointer to the allocated security flags.
Otherwise,
.Sy -1
is returned and
.Fa psf
is not updated.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4 ,
.Xr security-flags 5
```

```
**********************************************************
    2040 Mon Oct 15 13:28:33 2018
new/usr/src/man/man3proc/Psetbkpt.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PSETBKPT 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Psetbkpt
  19 .Nd set a breakpoint trap in a process
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft int
  25 .Fo Psetbkpt
  26 .Fa "struct ps_prochandle *P"
  27 .Fa "uintptr_t address"
  28 .Fa "ulong_t *saved"
  29 .Fc
  30 .Sh DESCRIPTION
  31 The
  32 .Fn Psetbkpt
  33 function sets a breakpoint instruction at the address
  34 .Fa address
  35 in the process handle
  36 .Fa P .
  37 The instruction that used to be set will be set in
  38 .Fa saved
  39 and should be retained.
  40 .Pp
  41 A breakpoint will remain in place until a subsequent call to
  42 .Xr Pdelbkpt 3PROC .
  43 The value stored in
  44 .Fa saved
  45 should be passed as the third argument to
  46 .Xr Pdelbkpt 3PROC .
  47 .Pp
  48 When a process executes an instruction that has been replaced with a
  49 breakpoint it generates a
  50 .Sy FLTBPT
  51 trap
  52 causing the thread to stop.
  53 .Pp
  54 Note, breakpoints may only be set in active processes.
  55 They may not be set in process handles that refer to core files, zombie
  56 processes, or files.
  57 .Sh RETURN VALUES
  58 Upon successful completion, the
  59 .Fn Psetbkpt
```



```
  60 function sets the breakpoint and
  61 returns
  62 .Sy 0 .
  63 Otherwise,
  64 .Sy -1
  65 is returned and
  66 .Sy errno
  67 is set to indicate the error.
  68 .Sh ERRORS
  69 For a full list of possible errors see the
  70 .Sy DIAGNOSTICS
  71 section in
  72 .Xr proc 4 .
  73 .Pp
  74 The
  75 .Fn Psetbkpt
  76 function will fail if:
  77 .Bl -tag -width Er
  78 .It Er ENOENT
  79 .Fa P
  80 does not refer to an active process.
  81 .It Er EBUSY
  82 A breakpoint instruction was already written by another debugger.
  83 .El
  84 .Sh INTERFACE STABILITY
  85 .Sy Uncommitted
  86 .Sh MT-LEVEL
  87 See
  88 .Sy LOCKING
  89 in
  90 .Xr libproc 3LIB .
  91 .Sh SEE ALSO
  92 .Xr libproc 3LIB ,
  93 .Xr Pdelbkpt 3PROC ,
  94 .Xr proc 4
```

```
**********************************************************
    1414 Mon Oct 15 13:28:33 2018
new/usr/src/man/man3proc/Psetcred.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETCRED 3PROC
.Os
.Sh NAME
.Nm Psetcred
.Nd set process credentials
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetcred
.Fa "struct ps_prochandle *P"
.Fa "const prcred_t *credp"
.Fc
.Sh DESCRIPTION
The
.Fn Psetcred
function updates the credentials of the process handle
.Fa P
to the values set in
.Fa credp .
.Fa credp
must be fully initialized.
The definition of the
.Sy prcred_t
structure may be found in
.Xr proc 4 .
.Pp
Note, the credentials may only be updated for an active process.
If the process handle refers to a zombie process, core file, or a file, then
.Fn Psetcred
will fail.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetcred
function returns
.Sy 0
and updates the credentials of the process
.Fa P .
Otherwise,
.Sy -1
is returned.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
```



```
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pcred 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1562 Mon Oct 15 13:28:34 2018
new/usr/src/man/man3proc/Psetfault.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETFAULT 3PROC
.Os
.Sh NAME
.Nm Psetfault
.Nd set fault tracing flags
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Psetfault
.Fa "struct ps_prochandle *P"
.Fa "const fltset_t *set"
.Fc
.Sh DESCRIPTION
The
.Fn Psetfault
function sets the fault tracing flags on the process handle
.Fa P
to
.Fa set .
It replaces any existing fault tracing flags on the process.
These flags indicate which faults cause execution of the thread to stop.
Allowing another tool, such as a debugger, to act upon the process.
For more information on faults and the
.Sy fltset_t
structure see the
.Sy PCSFAULT
section in
.Xr proc 4 .
The current fault set for the process may be obtained through the
.Xr Pfault 3PROC
function.
.Pp
Note, only active processes may have their fault set updated.
Process handles that refer to core files, zombie processes, and files do not
have fault tracing flags and this function is a no-op on them.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
```



```
.Xr libproc 3LIB ,
.Xr Pfault 3PROC ,
.Xr proc 4
```

```
**********************************************************
    2338 Mon Oct 15 13:28:34 2018
new/usr/src/man/man3proc/Psetflags.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETFLAGS 3PROC
.Os
.Sh NAME
.Nm Psetflags ,
.Nm Punsetflags
.Nd set and unset process flags
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetflags
.Fa "struct ps_prochandle *P"
.Fa "long flags"
.Fc
.Ft int
.Fo Punsetflags
.Fa "struct ps_prochandle *P"
.Fa "long flags"
.Fc
.Sh DESCRIPTION
The
.Fn Psetflags
and
.Fn Punsetflags
functions manipulate the process flags for the process handle
.Fa P .
The process flags determine how the process behaves in the face of
various actions.
For example, setting the
.Sy PR_FORK
flag indicates that the tracing flags of the process and the
inherit-on-fork mode should be set on children.
A full list of the process flags is available in the
.Sy PCSET
section in
.Xr proc 4 .
.Pp
The
.Fn Psetflags
function sets the flags specified in
.Fa flags
by doing a bitwise-inclusive-OR with the previously set flags.
.Pp
The
```



```
.Fn Punsetflags
function removes the flags specified in
.Fa flags
from the tracing flags of the process.
Items not listed in
.Fa flags
will remain.
.Pp
To see the current set of flags active on the process, check the
.Sy pr_flags
member of the
.Sy pstatus_t
for the process.
It can be obtained through the
.Xr Pstatus 3PROC
function.
.Pp
Note, attempting to modify the process flags only works on active
processes.
Attempting to call these functions of process handles corresponding to core
files, zombie processes, or files, will result in an error.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetflags
and
.Fn Punsetflags
functions return
.Sy 0 .
Otherwise,
.Sy -1
is returned
and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pstatus 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1954 Mon Oct 15 13:28:34 2018
new/usr/src/man/man3proc/Psetpriv.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETPRIV 3PROC
.Os
.Sh NAME
.Nm Psetpriv
.Nd set process privileges
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetpriv
.Fa "struct ps_prochandle *P"
.Fa "prpriv_t *pprv"
.Fc
.Sh DESCRIPTION
The
.Fn Psetpriv
function updates the privileges of the process handle
.Fa P
to the set described by
.Fa pprv .
The updated privilege set has restrictions on what it may change for a
process which are documented in the
.Sy PCSPRIV
section of
.Xr proc 4 .
In addition, the definition of the
.Sy prpriv_t
structure is described in
.Xr proc 4 .
.Pp
The current privilege set for the process may be obtained through a call
to
.Xr Ppriv 3PROC .
.Pp
Note, the privilege set may only be updated for active processes.
Process handles which correspond to core files, zombie processes, and
files, may not have their privilege sets updated.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetpriv
function returns
.Sy 0
and updates the privilege sets of the process.
Otherwise,
```



```
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Psetpriv
function will fail if:
.Bl -tag -width Er
.It Er EBADF
.Fa P
doesn’t refer to an active process, but a core file, zombie, or a file.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Ppriv 3PROC ,
.Xr proc 4 ,
.Xr privileges 5
```

```
**********************************************************
    3360 Mon Oct 15 13:28:34 2018
new/usr/src/man/man3proc/Psetrun.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETRUN 3PROC
.Os
.Sh NAME
.Nm Psetrun ,
.Nm Lsetrun
.Nd run a stopped process or thread
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetrun
.Fa "struct ps_prochandle *P"
.Fa "int sig"
.Fa "int flags"
.Fc
.Ft int
.Fo Lsetrun
.Fa "struct ps_lwphandle *L"
.Fa "int sig"
.Fa "int flags"
.Fc
.Sh DESCRIPTION
The
.Fn Psetrun
function resumes the stopped process handle
.Fa P
and transitions the process to running.
If
.Fa sig
is non-zero, then the
.Fn Psetrun
function causes the signal to be delivered.
See
.Xr signal.h 3HEAD
for a list of valid signal identifiers.
.Pp
The
.Fa flags
member modifies the behavior of the resumed handle.
The following values may be combined by a bitwise-inclusive-OR:
.Bl -tag -width Dv -offset indent
.It Dv PRCSIG
Clears the current signal, if any.
.It Dv PRCFAULT
```



```
Clears the current fault, if any.
.It Dv PRSTEP
Indicates that the thread should single-step over the next machine
instruction and upon completion, inject a trap.
For the specific mechanics of single-stepping and what traps or signals will be
injected, see the
.Sy PRSTEP
section of
.Xr proc 4 .
.It Dv PRSABORT
Indicates that the thread should abort the system call that it is
currently executing.
This is only valid if the thread is stopped because it is asleep or right before
a system call.
This will cause the system call to return
.Er EINTR .
.El
.Pp
If
both
.Dv PRCSIG
is specified and
.Fa sig
is non-zero, then the
.Dv PRCSIG
request takes priority, and it will be treated as though
.Fa sig
was passed the argument
.Sy 0 .
.Pp
When the process is resumed all extent tracing flags and register
changes will be synchronized with the process.
For more information on resuming a thread see the
.Sy PCRUN
section in
.Xr proc 4 .
.Pp
Note, the
.Fn Psetrun
function is only valid for active processes.
It will fail on process handles that refer to core files, zombie processes, and
ELF objects.
.Pp
The
.Fn Lsetrun
function is equivalent to the
.Fn Psetrun
function, except rather than operating on a process it operates on a
thread.
.Fn Lsetrun
causes the specified thread,
.Fa L ,
to resume execution.
Whereas
.Fn Psetrun
causes all threads in the process to resume.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetrun
and
.Fn Lsetrun
functions return
.Sy 0 .
Otherwise,
.Sy -1
is returned and
```

```
 126 .Sy errno
 127 is set to indicate the error.
 128 .Sh ERRORS
 129 For a full list of possible errors see the
 130 .Sy DIAGNOSTICS
 131 section in
 132 .Xr proc 4 .
 133 .Pp
 134 The
 135 .Fn Psetrun
 136 and
 137 .Fn Lsetrun
 138 functions will fail if:
 139 .Bl -tag -width Er
 140 .It Er EBUSY
 141 The process handle
 142 .Fa P
 143 is not currently stopped or it is not stopped due to an event of
 144 interest, a directed stop, or it is asleep in a system call.
 145 .El
 146 .Sh INTERFACE STABILITY
 147 .Sy Uncommitted
 148 .Sh MT-LEVEL
 149 See
 150 .Sy LOCKING
 151 in
 152 .Xr libproc 3LIB .
 153 .Sh SEE ALSO
 154 .Xr signal.h 3HEAD ,
 155 .Xr libproc 3LIB ,
 156 .Xr Pstatus 3PROC ,
 157 .Xr proc 4
```

```
**********************************************************
    1794 Mon Oct 15 13:28:35 2018
new/usr/src/man/man3proc/Psetsignal.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETSIGNAL 3PROC
.Os
.Sh NAME
.Nm Psetsignal
.Nd set signal tracing flags
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Psetsignal
.Fa "struct ps_prochandle *P"
.Fa "const sigset_t *set"
.Fc
.Sh DESCRIPTION
The
.Fn Psetsignal
function sets the signal tracing flags for the process handle
.Fa P
to
.Fa set .
The call to
.Fn Psetsignal
replaces any existing signal tracing flags entirely with
.Fa set .
The signal tracing flags determine which signals, when received by a
thread in the process, will cause that thread to stop.
For more information on the behavior of the signal tracing flags, including
which signals may be traced this way, see the
.Sy PCSTRACE
section in
.Xr proc 4 .
.Pp
The
.Fa set
argument may be manipulated with the standard signal set manipulation
functions such as
.Xr sigaddset 3C ,
.Xr sigdelset 3C ,
and others which may all be found in
.Xr sigsetops 3C .
.Pp
Note, only active processes may have their signal tracing flags updated.
Process handles that refer to core files, zombie processes, and files do
not have fault tracing flags and this function is a no-op on them.
```



```
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr sigsetops 3C ,
.Xr signal.h 3HEAD ,
.Xr libproc 3LIB ,
.Xr Psignal 3PROC ,
.Xr proc 4
```

```
**********************************************************
    1818 Mon Oct 15 13:28:35 2018
new/usr/src/man/man3proc/Psetsysentry.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETSYSENTRY 3PROC
.Os
.Sh NAME
.Nm Psetsysentry ,
.Nm Psetsysexit
.Nd set system call tracing flags
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Psetsysentry
.Fa "struct ps_prochandle *P"
.Fa "const sysset_t *set"
.Fc
.Ft void
.Fo Psetsysexit
.Fa "struct ps_prochandle *P"
.Fa "const sysset_t *set"
.Fc
.Sh DESCRIPTION
The
.Fn Psetsysentry
and
.Fn Psetsysexit
set the system call entry and exit tracing flags respectively in the
process handle
.Fa P
to
.Fa set .
The call to
.Fn Psetsysentry
or
.Fn Psetsysexit
replaces the corresponding set of system call tracing flags entirely
with the new set.
The system call entry tracing flags cause a thread to stop on entry to the
system call and the exit tracing flags cause a thread to stop on return from the
system call, before control returns back to the user land process.
For more information on the state of the thread and for information on
manipulating the
.Sy sysset_t ,
see
.Xr proc 4 .
.Pp
```



```
Note that only active processes may have their system call tracing flags
updated.
Process handles that refer to core files, zombie processes, and files do not
have fault tracing flags and this function is a no-op on them.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr Intro 2 ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    1821 Mon Oct 15 13:28:35 2018
new/usr/src/man/man3proc/Psetwapt.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETWAPT 3PROC
.Os
.Sh NAME
.Nm Psetwapt
.Nd set a watchpoint in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetwapt
.Fa "struct ps_prochandle *P"
.Fa "const prwatch_t *wp"
.Fc
.Sh DESCRIPTION
The
.Fn Psetwapt
function adds a watchpoint to the process handle
.Fa P .
Allowing the hardware to generate a trap when the specified area is
accessed.
The watchpoint’s parameters are described in
.Fa wp .
For more information on watchpoints and the
.Sy prwatch_t
structure, see the
.Sy PCWATCH
section in
.Xr proc 4 .
The watched area will persist until a subsequent call to
.Xr Pdelwapt 3PROC .
.Pp
Note, only active processes support watchpoints.
It is an error to call this function on process handles that correspond to core
files, zombie processes, or files.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetwapt
function returns
.Sy 0
and installs the watchpoint in
.Fa P .
Otherwise,
.Sy -1
is returned and
```



```
.Sy errno
is set.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Psetwapt
function will fail if:
.Bl -tag -width Er
.It Er ENOENT
.Fa P
does not refer to an active process.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pdelwapt 3PROC ,
.Xr proc 4
```

```
**********************************************************
    2437 Mon Oct 15 13:28:35 2018
new/usr/src/man/man3proc/Psetzoneid.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSETZONEID 3PROC
.Os
.Sh NAME
.Nm Psetzoneid
.Nd change processes zone id
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psetzoneid
.Fa "struct ps_prochandle *P"
.Fa "zoneid_t zoneid"
.Fc
.Sh DESCRIPTION
The
.Fn Psetzoneid
function moves the process handle
.Fa P
into the zone specified by
.Fa zoneid .
A process that is in the non-global zone may only move between the
global zone and its original zone.
A process that is in the global zone may not use this interface to enter a
non-global zone.
This function will fail if called from a non-global zone.
This function only manipulates the processes credentials.
.Pp
Care should be taken when moving a process around temporarily, such that
if the process that is manipulating
.Fa P
dies, it does not cause
.Fa P
to resume running while still in the global zone.
It is suggested that the
.Sy PR_KLC
flag is set with
.Xr Psetflags 3PROC
which will cause the process to terminate if the process that holds
.Fa P
unexpectedly terminates.
See
.Xr proc 4
for more information on the
.Sy PR_KLC
```



```
flag.
.Pp
Note, only active processes may change their zone.
It is an error to call this function on process handles that correspond to core
files, zombie processes, or files.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psetzoneid
function returns
.Sy 0
and changes the zone for
.Fa P .
Otherwise,
.Sy -1
is returned and
.Sy errno
is set.
.Sh ERRORS
For a full list of possible errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4 .
.Pp
The
.Fn Psetzoneid
function will fail if:
.Bl -tag -width Er
.It Er EINVAL
.Fa zoneid
does not correspond to an existing zone or the zone id is not the global
zone or the original zone of
.Fa P .
.It Er EPERM
The caller does not hold the required privileges for zone configuration.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr proc 4 ,
.Xr privileges 5 ,
.Xr zones 5
```

```
**********************************************************
    2018 Mon Oct 15 13:28:35 2018
new/usr/src/man/man3proc/Psignal.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSIGNAL 3PROC
.Os
.Sh NAME
.Nm Psignal
.Nd set signal tracing action
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psignal
.Fa "struct ps_prochandle *P"
.Fa "int which"
.Fa "int stop"
.Fc
.Sh DESCRIPTION
The
.Fn Psignal
function sets the signal tracing flag for the process handle
.Fa P .
If
.Fa stop
is
.Sy non-zero
it causes the process handle to stop threads that encounter the signal
.Fa which .
If
.Fa stop
is
.Sy zero ,
then it disables tracing for the signal
.Fa which .
.Pp
The signal constants, representing valid values for
.Fa which ,
can be found in
.Xr signal.h 3HEAD.
The signal
.Dv SIGKILL
may not be stopped.
.Pp
Note, only active processes may have their signal tracing flags updated.
Process handles that refer to core files, zombie processes, and files do
not have signal tracing flags.
Calling this function on them is an error.
```



```
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psignal
function returns the previous disposition of the signal
.Fa which .
It returns
.Sy 1
if it was set and
.Sy 0
if not.
Otherwise,
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
The
.Fn Psignal
function will fail if:
.Bl -tag -width Er
.It Er EINVAL
.Fa which
is
.Dv SIGKILL
and
.Fa stop
is non-zero .
.Pp
.Fa which
is not a valid signal.
.It Er ENOENT
.Fa P
does not correspond to an active process.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr signal.h 3HEAD ,
.Xr libproc 3LIB ,
.Xr Psetsignal 3PROC ,
.Xr proc 4
```

```
**********************************************************
    3124 Mon Oct 15 13:28:36 2018
new/usr/src/man/man3proc/Pstack_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PSTACK_ITER 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pstack_iter
  19 .Nd iterate process stack frames
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft int
  25 .Fo Pstack_iter
  26 .Fa "struct ps_prochandle *P"
  27 .Fa "const prgregset_t regs"
  28 .Fa "proc_stack_f *func"
  29 .Fa "void *data"
  30 .Fc
  31 .Sh DESCRIPTION
  32 The
  33 .Fn Pstack_iter
  34 function iterates over the stack frames in the process
  35 .Fa P
  36 starting at the point defined by
  37 .Fa regs .
  38 .Pp
  39 For each valid stack frame encountered, the callback function
  40 .Fa func
  41 is invoked with
  42 .Fa data
  43 passed as argument.
  44 The full signature of
  45 .Ft proc_stack_f
  46 is defined in
  47 .Xr libproc 3LIB .
  48 With each callback, a register set, argument set, and argument count
  49 will be provided.
  50 In that register set, only a subset of the registers will be valid, which
  51 include the frame pointer, program counter, and on SPARC systems, the next
  52 program counter.
  53 These registers can be accessed with the constants
  54 .Sy R_FP ,
  55 .Sy R_PC ,
  56 and
  57 .Sy R_nPC
  58 respectively.
  59 These correspond to the registers
```



```
  60 .Em %ebp
  61 and
  62 .Em %eip
  63 on i386,
  64 .Em %rbp
  65 and
  66 .Em %rip
  67 on amd64,
  68 .Em %fp ,
  69 .Em %pc ,
  70 and
  71 .Em %npc
  72 on both SPARC and SPARCv9.
  73 .Pp
  74 Callers will receive a callback for the first stack frame indicated by
  75 .Fa regs
  76 and then will receive a subsequent callback for each caller of that
  77 frame until no such frame can be found.
  78 Stack frames that logically come after the frame indicated by
  79 .Fa regs
  80 will not receive callbacks.
  81 .Pp
  82 The compiler can either facilitate or stymie the iteration of the stack.
  83 Programs that have been compiled in such a way as to omit the frame pointer will
  84 result in truncated stacks.
  85 Similarly, if the initial set of registers passed in via
  86 .Fa regs
  87 is invalid, then the ability to iterate the stack will be limited.
  88 The return value of
  89 .Fa func
  90 controls whether or not iteration continues.
  91 If
  92 .Fa func
  93 returns
  94 .Sy 0
  95 then iteration continues.
  96 However, if
  97 .Fa func
  98 returns non-zero, then iteration will halt and that value will be used
  99 as the return value of the
 100 .Fn Pstack_iter
 101 function.
 102 Because
 103 .Fn Pstack_iter
 104 returns
 105 .Sy -1
 106 on internal failure it is recommended the callback function not return
 107 .Sy -1
 108 to indicate an error.
 109 Thus the caller may distinguish between the failure of the callback function and
 110 the failure of the
 111 .Fn Pstack_iter
 112 function.
 113 .Sh RETURN VALUES
 114 Upon successful completion, the
 115 .Fn Pstack_iter
 116 function returns
 117 .Sy 0.
 118 If there was an internal error then
 119 .Sy -1
 120 is returned.
 121 Otherwise, if the callback function
 122 .Fa func
 123 returns non-zero, then its return value will be returned instead.
 124 .Sh INTERFACE STABILITY
 125 .Sy Uncommitted
```

```
 126 .Sh MT-LEVEL
 127 See
 128 .Sy LOCKING
 129 in
 130 .Xr libproc 3LIB .
 131 .Sh SEE ALSO
 132 .Xr libproc 3LIB ,
 133 .Xr proc 4
```

```
***********************************************************
    1030 Mon Oct 15 13:28:36 2018
new/usr/src/man/man3proc/Pstate.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2015 Joyent, Inc.
  13 .\"
  14 .Dd May 11, 2016
  15 .Dt PSTATE 3PROC
  16 .Os
  17 .Sh NAME
  18 .Nm Pstate
  19 .Nd obtain process handle state
  20 .Sh LIBRARY
  21 .Lb libproc
  22 .Sh SYNOPSIS
  21 .Lb libproc
  23 .In libproc.h
  24 .Ft int
  25 .Fo Pstate
  26 .Fa "struct ps_prochandle *P"
  27 .Fc
  28 .Sh DESCRIPTION
  29 The
  30 .Fn Pstate
  31 function returns the state of the process handle
  32 .Fa P .
  33 The list of states is available in the
  34 .Sy PROCESS STATES
  35 section in
  36 .Xr libproc 3LIB .
  37 .Sh RETURN VALUES
  38 Upon successful completion, the current state is returned.
  39 .Sh INTERFACE STABILITY
  40 .Sy Uncommitted
  41 .Sh MT-LEVEL
  42 See
  43 .Sy LOCKING
  44 in
  45 .Xr libproc 3LIB .
  46 .Sh SEE ALSO
  47 .Xr libproc 3LIB
```

```
**********************************************************
    1365 Mon Oct 15 13:28:36 2018
new/usr/src/man/man3proc/Pstatus.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSTATUS 3PROC
.Os
.Sh NAME
.Nm Pstatus
.Nd obtain process status structure
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft "const pstatus_t *"
.Fo Pstatus
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pstatus
function returns the status information about the process handle
.Fa P .
The
.Sy pstatus_t
structure is defined in
.Xr proc 4
and contains information about the process including its flags, the
number of threads, the size of the stack, and more.
.Pp
The returned pointer is only valid as long as the process handle
.Fa P
is valid.
After a call to
.Xr Prelease 3PROC ,
the returned data pointer is invalid.
.Sh RETURN VALUES
The
.Fn Pstatus
function always returns a pointer to the process's status.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Prelease 3PROC ,
```



```
.Xr proc 4
```

```
**********************************************************
    4787 Mon Oct 15 13:28:36 2018
new/usr/src/man/man3proc/Pstopstatus.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSTOPSTATUS 3PROC
.Os
.Sh NAME
.Nm Pdstop ,
.Nm Pstopstatus ,
.Nm Pstop ,
.Nm Pwait ,
.Nm Ldstop ,
.Nm Lstop ,
.Nm Lwait
.Nd process and thread stop operations
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Pdstop
.Fa "struct ps_prochandle *P"
.Fc
.Ft int
.Fo Pstopstatus
.Fa "struct ps_prochandle *P"
.Fa "long request"
.Fa "uint_t msec"
.Fc
.Ft int
.Fo Pstop
.Fa "struct ps_prochandle *P"
.Fc
.Ft int
.Fo Pwait
.Fa "struct ps_prochandle *P"
.Fc
.Ft int
.Fo Ldstop
.Fa "struct ps_lwphandle *L"
.Fc
.Ft int
.Fo Lstop
.Fa "struct ps_lwphandle *L"
.Fc
.Ft int
.Fo Lwait
.Fa "struct ps_lwphandle *L"
.Fc
```



```
.Sh DESCRIPTION
The
.Fn Pstopstatus
function allows the caller to stop and optionally wait for the process
handle referred to by
.Fa P
to be stopped.
Stopping a process causes all of its threads to stop execution.
Where in their execution the threads will halt is not defined.
Threads may be resumed with
.Xr Psetrun 3PROC
and
.Xr prun 1 .
.Pp
The
.Fa request
argument should be one of the following symbols:
.Bl -tag -width Dv -offset indent
.It Dv PCSTOP
Stop the process; wait for completion before returning.
.It Dv PCDSTOP
Stop the process; do not wait for completion before returning.
That is, the stopping of the process is performed asynchronously in
relation to the caller.
.It Dv PCWSTOP
Do not direct the process to stop; simply wait for it to stop.
.It Dv PCNULL
Do not direct the process to stop; simply refreshes the state of the
process.
.El
.Pp
Both the
.Dv PCSTOP
and
.Dv PCWSTOP
requests allow an upper bound on the amount of time to wait for the
process to stop.
The
.Fa msec
argument indicates the number of milliseconds to wait for the stop to
complete.
If the value of
.Fa msec
is
.Sy 0 ,
then it will wait forever.
Callers should pass
.Sy 0
for
.Fa msec
when the request is
.Dv PCDSTOP
or
.Dv PCNULL .
.Pp
When a non-zero timeout is specified, the process may or may not be
stopped upon return.
The return value does not reflect the current state of the process.
For example, if the timeout expires during a
.Fa PCWSTOP
request, the return value will be
.Sy 0
regardless of the actual state of the process.
.Pp
Only active processes may be stopped.
Handles that refer to core files, zombie processes, or files cannot be used;
```

```
126 unless the value of
127 .Fa request
128 is set to
129 .Dv PCNULL .
130 .Pp
131 The
132 .Fn Pstop
133 function is is equivalent to calling the
134 .Fn Pstopstatus
135 function with the request set to
136 .Dv PCSTOP
137 and an infinite timeout.
138 .Pp
139 The
140 .Fn Pwait
141 function is is equivalent to calling the
142 .Fn Pstopstatus
143 function with the request set to
144 .Dv PCWSTOP
145 and an infinite timeout.
146 .Pp
147 The
148 .Fn Pdstop
149 function is is equivalent to calling the
150 .Fn Pstopstatus
151 function with the request set to
152 .Dv PCDSTOP .
153 .Pp
154 The
155 .Fn Ldstop ,
156 .Fn Lstop ,
157 and
158 .Fn Lwait
159 functions are equivalent to the
160 .Fn Pdstop ,
161 .Fn Pstop ,
162 and
163 .Fn Pwait
164 functions, respectively.
165 Except, rather than operating on a process, they operate on the thread handle
166 .Fa L .
167 A call to
168 .Fn Lstop
169 stops only a single thread; whereas
170 .Fn Pstop
171 stops every thread in the process.
172 .Sh RETURN VALUES
173 Upon successful completion, the
174 .Fn Pdstop ,
175 .Fn Pstopstatus ,
176 .Fn Pstop ,
177 .Fn Pwait ,
178 .Fn Ldstop ,
179 .Fn Lstop ,
180 and
181 .Fn Lwait
182 functions return
183 .Sy 0 .
184 Otherwise,
185 .Sy -1
186 is returned and
187 .Dv errno
188 is set to indicate the error that occurred.
189 .Sh ERRORS
190 For a full list of possible errors see the
191 .Sy DIAGNOSTICS
```



```
192 section in
193 .Xr proc 4 .
194 .Pp
195 The
196 .Fn Pdstop ,
197 .Fn Pstopstatus ,
198 .Fn Pstop ,
199 .Fn Pwait ,
200 .Fn Ldstop ,
201 .Fn Lstop ,
202 and
203 .Fn Lwait
204 functions will fail if:
205 .Bl -tag -width Er
206 .It Er EAGAIN
207 Control over the handle
208 .Fa P
209 was lost.
210 Callers should call
211 .Xr Preopen 3PROC .
212 For more information on losing control, see
213 .Sy PROGRAMMING NOTES
214 in
215 .Xr proc 4 .
216 .It Er ENOENT
217 The request was not
218 .Dv PCNULL
219 and the process handle
220 .Fa P
221 does not refer to an active process, but refers to a core file, a zombie
222 process, or a file.
223 .It Er EINVAL
224 .Fa request
225 is not valid or the process is in an unknown state.
226 .It Er EPROTO
227 A fatal protocol error occurred and the process could not be stopped.
228 .El
229 .Sh INTERFACE STABILITY
230 .Sy Uncommitted
231 .Sh MT-LEVEL
232 See
233 .Sy LOCKING
234 in
235 .Xr libproc 3LIB .
236 .Sh SEE ALSO
237 .Xr libproc 3LIB ,
238 .Xr Lgrab 3PROC ,
239 .Xr Pcreate 3PROC ,
240 .Xr Pgrab 3PROC ,
241 .Xr Pgrab_core 3PROC ,
242 .Xr Pgrab_file 3PROC ,
243 .Xr proc 4
```

```
**********************************************************
    7152 Mon Oct 15 13:28:37 2018
new/usr/src/man/man3proc/Psymbol_iter.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSYMBOL_ITER 3PROC
.Os
.Sh NAME
.Nm Psymbol_iter ,
.Nm Psymbol_iter_by_addr ,
.Nm Psymbol_iter_by_lmid ,
.Nm Psymbol_iter_by_name ,
.Nm Pxsymbol_iter
.Nd iterate symbols in a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psymbol_iter
.Fa "struct ps_prochandle *P"
.Fa "const char *object_name"
.Fa "int which"
.Fa "int mask"
.Fa "proc_sym_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Psymbol_iter_by_addr
.Fa "struct ps_prochandle *P"
.Fa "const char *object_name"
.Fa "int which"
.Fa "int mask"
.Fa "proc_sym_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Psymbol_iter_by_lmid
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *object_name"
.Fa "int which"
.Fa "int mask"
.Fa "proc_sym_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Psymbol_iter_by_name
.Fa "struct ps_prochandle *P"
.Fa "const char *object_name"
```



```
.Fa "int which"
.Fa "int mask"
.Fa "proc_sym_f *func"
.Fa "void *data"
.Fc
.Ft int
.Fo Pxsymbol_iter
.Fa "struct ps_prochandle *P"
.Fa "Lmid_t lmid"
.Fa "const char *object_name"
.Fa "int which"
.Fa "int mask"
.Fa "proc_xsym_f *func"
.Fa "void *data"
.Fc
.Sh DESCRIPTION
The
.Fn Psymbol_iter ,
.Fn Psymbol_iter_by_addr ,
.Fn Psymbol_iter_by_lmid ,
.Fn Psymbol_iter_by_name ,
and
.Fn Pxsymbol_iter
functions are used to iterate over the symbols present in the process
referred to by the handle
.Fa P .
For each symbol found, the callback function
.Fa func
will be called once and the argument
.Fa data
will be passed to it along with an ELF symbol entry in the form of the
.Sy GElf_Sym
along with the name of the symbol, if known.
In the case of the
.Fn Pxsymbol_iter
function an additional
.Sy prsyminfo_t
argument will be provided to the callback.
The definitions of
.Sy proc_sym_f ,
.Sy proc_xsym_f ,
and
.Sy prsyminfo_t
are found in
.Xr libproc 3LIB .
.Pp
The
.Fa object_name
argument names the object that is a part of the controlled process which
will be searched for symbols.
Only one object may be searched at any given time.
Valid object names may be obtained through the
.Xr Pobjname 3PROC
and
.Xr Pobject_iter 3PROC
functions, among others.
The system also has two special object names that may be passed in to refer to
the objects of the executable file and for ld.so.1.
The symbol
.Dv PR_OBJ_EXEC
refers to the executables object and the symbol
.Dv PR_OBJ_LDSO
refers to the object ld.so.1.
.Pp
The
.Fa which
```

```
126 argument controls which of two possible symbol tables will be searched.
127 If the argument is
128 .Dv PR_SYMTAB
129 then the ELF symbol table will be searched.
130 Otherwise, if it is
131 .Dv PR_DYNSYM
132 then the symbol table associated with the dynamic section will be
133 searched instead.
134 If any other value is specified for
135 .Fa which ,
136 then an error will be returned.
137 .Pp
138 The
139 .Fa mask
140 argument controls which symbols will be included.
141 The
142 .Fa mask
143 argument allows for control over both the symbol's binding and the
144 symbol's type.
145 These flags logically correspond to the various ELF symbol bindings and types.
146 The following values may be passed as a bitwise-inclusive-OR into the
147 .Fa flags
148 argument:
149 .Bl -tag -width Dv -offset indent
150 .It Dv BIND_LOCAL
151 The symbol is a local symbol.
152 Local symbols are not visible outside of their object file.
153 .It Dv BIND_GLOBAL
154 The symbol is a global symbol.
155 Global symbols are visible outside of their object file and may be referred to
156 by other ELF objects.
157 .It Dv BIND_WEAK
158 The symbol is a weak symbol.
159 Weak symbols are visible outside of their object file, but another definition of
160 the symbol may be used instead.
161 .It Dv BIND_ANY
162 This is a combination of
163 .Dv BIND_LOCAL ,
164 .Dv BIND_GLOBAL ,
165 and
166 .Dv BIND_WEAK .
167 Every symbol's binding will match this value.
168 .It Dv TYPE_NOTYPE
169 The symbol's type is not specified.
170 .It Dv TYPE_OBJECT
171 The symbol refers to a data object.
172 For example, variables.
173 .It Dv TYPE_FUNC
174 The symbol refers to a function.
175 .It Dv TYPE_SECTION
176 The symbol refers to an ELF section.
177 .It Dv TYPE_FILE
178 The symbol refers to the name of a source file associated with an object
179 file.
180 .It Dv TYPE_ANY
181 This is a combination of
182 .Dv TYPE_NOTYPE ,
183 .Dv TYPE_OBJECT ,
184 .Dv TYPE_FUNC ,
185 .Dv TYPE_SECTION ,
186 and
187 .Dv TYPE_FILE .
188 Every symbol's type will match this value.
189 .El
190 .Pp
191 To obtain all of the symbols in an object, the caller would pass the
```



```
192 expression
193 .Dv BIND_ANY |
194 .Dv TYPE_ANY
195 in as the value of
196 .Fa mask.
197 .Pp
198 The
199 .Fn Psymbol_iter_by_lmid
200 and
201 .Fn Pxsymbol_iter
202 functions allow for a link-map identifier to be specified in the
203 .Fa lmid
204 argument.
205 This will restrict the search for the object specified in
206 .Fa object_name
207 to the specified link-map.
208 There are three special link-map identifiers that may be passed in.
209 The symbol
210 .Dv PR_LMID_EVERY
211 indicates that every link-map should be searched.
212 The symbol
213 .Dv LM_ID_BASE
214 indicates that the base link-map, the one that is used for the
215 executable should be searched.
216 Finally, the symbol
217 .Dv LM_ID_LDSO
218 refers to the link-map that is used by the run-time link editor, ld.so.1.
219 The functions which do not allow a link-map identifier to be specified always
220 search every link-map.
221 .Pp
222 By default, symbols are iterated based on the order of the symbol
223 table being searched.
224 However, it is also possible to iterate based on the name of the symbol and
225 based on the address of the symbol.
226 To iterate by name use the
227 .Fn Psymbol_iter_by_name
228 function.
229 To iterate by address use the
230 .Fn Psymbol_iter_by_addr
231 function.
232 The
233 .Fn Psymbol_iter ,
234 .Fn Psymbol_iter_by_lmid ,
235 and
236 .Fn Pxsymbol_iter
237 functions all sort based on the order of the symbol table.
238 .Pp
239 The return value of the callback function
240 .Fa func
241 determines whether or not iteration continues.
242 If
243 .Fa func
244 returns
245 .Sy 0,
246 then iteration will continue.
247 However, if
248 .Fa func
249 returns non-zero, then iteration will halt and that value will be used
250 as the return value of the
251 .Fn Psymbol_iter ,
252 .Fn Psymbol_iter_by_addr ,
253 .Fn Psymbol_iter_by_lmid ,
254 .Fn Psymbol_iter_by_name ,
255 and
256 .Fn Pxsymbol_iter
257 functions.
```

```
Because these functions return
.Sy -1
on internal failure, it is recommended that the callback function not return
.Sy -1
to indicate an error so that the caller may distinguish between the
failure of the callback function and the failure of these functions.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psymbol_iter ,
.Fn Psymbol_iter_by_addr ,
.Fn Psymbol_iter_by_lmid ,
.Fn Psymbol_iter_by_name ,
and
.Fn Pxsymbol_iter
functions return
.Sy 0 .
If there was an internal error then
.Sy -1
is returned.
Otherwise, if the callback function
.Fa func
returns non-zero, then its return value will be returned instead.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr elf 3ELF ,
.Xr gelf 3ELF ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2033 Mon Oct 15 13:28:37 2018
new/usr/src/man/man3proc/Psync.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSYNC 3PROC
.Os
.Sh NAME
.Nm Psync ,
.Nm Lsync
.Nd synchronize cached tracing flags and modifications
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Psync
.Fa "struct ps_prochandle *P"
.Fc
.Ft void
.Fo Lsync
.Fa "struct ps_lwphandle *L"
.Fc
.Sh DESCRIPTION
The
.Fn Psync
function synchronizes modifications to the process handle
.Fa P
back to the underlying active process.
The
.Fn Psync
function ensures that any outstanding process holds, register
modifications, signal injections, and modifications to the various fault
and system call tracing flags are taken care of.
.Pp
Normally this function is not required as other library routines take
care of synchronizing this state out to the process when it is required.
If the underlying /proc file system routines are used outside of the
library, calling this function may be required.
.Pp
The
.Fn Psync
function is only meaningful for active processes.
It will do nothing on process handles that refer to core files, zombie
processes, and ELF objects.
.Pp
The
.Fn Lsync
function is equivalent to the
.Fn Psync
```



```
function, except rather than operating on the entire process and its
representative thread, it instead operates on the thread handle
.Fa L .
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pfault 3PROC ,
.Xr Pputareg 3PROC ,
.Xr Psetfault 3PROC ,
.Xr Psetsignal 3PROC ,
.Xr Psetsysentry 3PROC ,
.Xr Psetsysexit 3PROC ,
.Xr Psignal 3PROC ,
.Xr Psysentry 3PROC ,
.Xr Psysexit 3PROC ,
.Xr proc 4
```

```
**********************************************************
    2795 Mon Oct 15 13:28:37 2018
new/usr/src/man/man3proc/Psysentry.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PSYSENTRY 3PROC
.Os
.Sh NAME
.Nm Psysentry ,
.Nm Psysexit
.Nd set system call entry and exit actions
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Psysentry
.Fa "struct ps_prochandle *P"
.Fa "int which"
.Fa "int stop"
.Fc
.Ft int
.Fo Psysexit
.Fa "struct ps_prochandle *P"
.Fa "int which"
.Fa "int stop"
.Fc
.Sh DESCRIPTION
The
.Fn Psysentry
and
.Fn Psysexit
functions controls what actions the process handle
.Fa P
should take upon executing a system call.
.Pp
The system allows a process to be stopped on both entry and exit of a
system call.
For information on the state of the process when it is stopped due to system
call tracing, see the
.Sy PCSENTRY
and
.Sy PCSEXIT
sections of
.Xr proc 4 .
.Pp
The value of the
.Fa stop
parameter controls whether or not the system call listed in
.Fa which
```



```
causes the process to stop.
A value of non-zero indicates the process should stop;
a value of 0 indicates it should not.
.Pp
The value of
.Fa which
indicates which system call the change applies to.
A value of 0 applies to all system calls.
Note, the system does not supply a stable mapping from system call names to
identifiers.
.Pp
These functions only apply to actively running processes.
They do not function on handles that refer to core files, zombie processes,
or ELF objects.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Psysentry
and
.Fn Psysexit
functions return the previous disposition of the system call --
.Sy 0
if it was not set to stop and
.Sy 1
if it was --
and the system call state is updated.
Otherwise,
.Sy -1
is returned,
.Dv errno
is updated with the error that occurred, and the system call state is
not updated.
.Sh ERRORS
The
.Fn Psysentry
and
.Fn Psysexit
functions will fail if:
.Bl -tag -width Er
.It Er EINVAL
The value of
.Fa which
is invalid, e.g. it is less than zero or greater than the largest defined
system call.
.It Er ENOENT
The handle
.Fa P
refers to a process that is a zombie, a core file, or an ELF object.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Pp
Note, while the
.Fn Psysentry
and
.Fn Psysexit
functions are uncommitted, the mapping of system calls to system call
numbers is
.Sy Not-an-Interface
and may change at any time.
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
```

```
126 .Xr proc 4 ,
127 .Xr attributes 5
```

```
**********************************************************
    1775 Mon Oct 15 13:28:37 2018
new/usr/src/man/man3proc/Puname.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PUNAME 3PROC
.Os
.Sh NAME
.Nm Puname
.Nd get uname information from a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft int
.Fo Puname
.Fa "struct ps_prochandle *P"
.Fa "struct utsname *u"
.Fc
.Sh DESCRIPTION
The
.Fn Puname
function copies the operating system information from the process
handle
.Fa P
into
.Fa u .
.Pp
For an active process or zombie process, this is the same information
obtained from
.Xr uname 2 .
For core files, if available, it is the system information at the time
the core was dumped.
.Pp
Handles that correspond to ELF objects do not contain uname
information.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Puname
function returns
.Sy 0
and updates the information at
.Fa u .
Otherwise,
.Sy -1
is returned and
.Sy errno
is set to indicate the error.
.Sh ERRORS
The
```



```
.Fn Puname
function will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
is a handle that corresponds to an ELF object or
.Fa P
is a handle that corresponds to a core file and that information is not
available in the core file.
.It Er EFAULT
.Fa P
is a handle that corresponds to an active process and
.Fa u
is a bad address.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr uname 1 ,
.Xr uname 2 ,
.Xr libproc 3LIB
```

```
***********************************************************
    1498 Mon Oct 15 13:28:37 2018
new/usr/src/man/man3proc/Pupdate_maps.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PUPDATE_MAPS 3PROC
.Os
.Sh NAME
.Nm Pupdate_maps
.Nd update address space mappings
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Pupdate_maps
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pupdate_maps
function is used to verify that all of the cached mappings in the
process
.Fa P
are still valid and update the cached data with any new information.
This is generally called in response to activity by the run-time
link-editor.
In general, the
.Sy libproc
library takes care of managing the need to call this function; however,
debuggers, introspection tools, and others that are interposing on rtld
activity or other actions, may need to call this function.
Note that the
.Fn Pupdate_maps
function is only meaningful for active processes.
It will do nothing on process handles that refer to core files, zombie
processes, and ELF objects.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pupdate_syms 3PROC
```

```
***********************************************************
    1534 Mon Oct 15 13:28:38 2018
new/usr/src/man/man3proc/Pupdate_syms.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
***********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PUPDATE_SYMS 3PROC
.Os
.Sh NAME
.Nm Pupdate_syms
.Nd update cached symbol tables
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft void
.Fo Pupdate_syms
.Fa "struct ps_prochandle *P"
.Fc
.Sh DESCRIPTION
The
.Fn Pupdate_syms
function is used to update all of the cached symbol tables in the
process handle
.Fa P :
updating, invalidating, and caching new symbol tables as appropriate for
functions such as
.Xr Psymbol_iter 3PROC .
This is generally called in response to activity by the run-time
link-editor.
In general, the
.Sy libproc
library takes care of managing the need to call this function;
however, debuggers, introspection tools, and others that are
interposing on rtld activity may need to call this function.
Note that the
.Fn Pupdate_syms
function is only meaningful for active processes.
It will do nothing on process handles that refer to core files, zombie
processes, and ELF objects.
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr libproc 3LIB ,
.Xr Pupdate_maps 3PROC
```

```
**********************************************************
    2128 Mon Oct 15 13:28:38 2018
new/usr/src/man/man3proc/Pwrite.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PWRITE 3PROC
.Os
.Sh NAME
.Nm Pwrite
.Nd write data to the address space of a process
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft ssize_t
.Fo Pwrite
.Fa "struct ps_prochandle *P"
.Fa "const void *buf"
.Fa "size_t nbyte"
.Fa "uintptr_t address"
.Fc
.Sh DESCRIPTION
The
.Fn Pwrite
function writes data from
.Fa buf
to the process handle
.Fa P
starting at the address
.Fa address .
It writes at most
.Fa nbyte
of data.
The
.Fn Pwrite
function is logically analogous to the
.Xr pwrite 2
function.
.Pp
For live processes, this function is equivalent to writing to the
/proc file system
.Sy as
file for the process.
For core files, it writes to the logical address space of what was once the
process and not the corresponding offset in the on-disk file.
ELF objects grabbed through
.Xr Pgrab_file 3PROC
do not support being written to.
.Pp
The
```



```
.Fn Pwrite
function cannot be used to
.Em extend
the size of a mapping; writing to an unmapped region generates an
error.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pwrite
function returns the number of bytes successfully written to
.Fa P .
This number is never greater than
.Fa nbyte .
Otherwise, it returns
.Sy -1
and
.Sy errno
is set to indicate an error.
For the full list of errors see the
.Sy DIAGNOSTICS
section in
.Xr proc 4
and
the
.Sy ERRORS
section in
.Xr pwrite 2 .
.Pp
In addition, the
.Fn Pwrite
function will fail if:
.Bl -tag -width Er
.It Er EIO
.Fa P
refers to an ELF object and not a core file or active process.
.El
.Sh INTERFACE STABILITY
.Sy Uncommitted
.Sh MT-LEVEL
See
.Sy LOCKING
in
.Xr libproc 3LIB .
.Sh SEE ALSO
.Xr pwrite 2 ,
.Xr libproc 3LIB ,
.Xr proc 4
```

```
**********************************************************
    2747 Mon Oct 15 13:28:38 2018
new/usr/src/man/man3proc/Pzonename.3proc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2015 Joyent, Inc.
.\"
.Dd May 11, 2016
.Dt PZONENAME 3PROC
.Os
.Sh NAME
.Nm Pzonename ,
.Nm Pzoneroot ,
.Nm Pzonepath
.Nd get zone name, root, and full object path
.Sh LIBRARY
.Lb libproc
.Sh SYNOPSIS
.Lb libproc
.In libproc.h
.Ft char *
.Fo Pzonename
.Fa "struct ps_prochandle *P"
.Fa "char *buf"
.Fa "size_t nbyte"
.Fc
.Ft char *
.Fo Pzoneroot
.Fa "struct ps_prochandle *P"
.Fa "char *buf"
.Fa "size_t nbyte"
.Fc
.Ft char *
.Fo Pzonepath
.Fa "struct ps_prochandle *P"
.Fa "const char *path"
.Fa "char *buf"
.Fa "size_t nbyte"
.Fc
.Sh DESCRIPTION
The
.Fn Pzonename
function attempts to determine the name of the zone for the process
handle
.Fa P .
If found, up to
.Fa nbytes ,
including a null terminator, will be written into
.Fa buf .
.Pp
The
.Fn Pzoneroot
function attempts to determine the root of the zone corresponding to the
process handle
```



```
.Fa P .
If found, up to
.Fa nbytes ,
including a null terminator, will be written into
.Fa buf .
If the root cannot be found, for example a core file that did not
originate on the current system, then the empty string will be written
into
.Fa buf .
.Pp
The
.Fn Pzonepath
function attempts to derive the full path of the object
.Fa path
in a zone relative to the root associated with the current process
handle
.Fa P .
If found, up to
.Fa nbytes ,
including a null terminator, will be written into
.Fa buf .
It is legal to use the same buffer for both
.Fa path
and
.Fa buf ;
it will not be updated unless the function completes successfully.
.Sh RETURN VALUES
Upon successful completion, the
.Fn Pzonename ,
.Fn Pzoneroot ,
and
.Fn Pzonepath
functions return
.Sy buf .
Otherwise, if an error occurred,
.Dv NULL
is returned and
.Sy errno
is set.
.Sh ERRORS
The
.Fn Pzonename
and
.Fn Pzoneroot
functions will fail if:
.Bl -tag -width Er
.It Er ENODATA
.Fa P
refers to a core file and zone information was not available in the core
dump or
.Fa P
refers to an ELF object grabbed through
.Xr Pgrab_file 3PROC .
.It Er EFAULT
.Fa P
refers to an active process and
.Fa buf
is invalid.
.El
.Pp
The
.Fn Pzoneroot
function will fail if:
.Bl -tag -width Er
.It Er ENOMEM
Insufficient memory was available on the system.
```

```
 126 .El
 127 .Sh INTERFACE STABILITY
 128 .Sy Uncommitted
 129 .Sh MT-LEVEL
 130 See
 131 .Sy LOCKING
 132 in
 133 .Xr libproc 3LIB .
 134 .Sh SEE ALSO
 135 .Xr getzoneidbyname 3C ,
 136 .Xr libproc 3LIB ,
 137 .Xr proc 4
```

```
**********************************************************
  100203 Mon Oct 15 13:28:38 2018
new/usr/src/man/man4/proc.4
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
'\" te
.\" Copyright 1989 AT&T
.\" Copyright (c) 2006, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 2018, Joyent, Inc.
.\" Copyright (c) 2013, Joyent, Inc. All rights reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH PROC 4 "August 22, 2018"
.TH PROC 4 "Jun 6, 2016"
.SH NAME
proc \- /proc, the process file system
.SH DESCRIPTION
.LP
\fB/proc\fR is a file system that provides access to the state of each process
and light-weight process (lwp) in the system. The name of each entry in the
\fB/proc\fR directory is a decimal number corresponding to a process-ID. These
entries are themselves subdirectories. Access to process state is provided by
additional files contained within each subdirectory; the hierarchy is described
more completely below. In this document, ``\fB/proc\fR file'' refers to a
non-directory file within the hierarchy rooted at \fB/proc\fR. The owner of
each \fB/proc\fR file and subdirectory is determined by the user-ID of the
process.
.sp
.LP
\fB/proc\fR can be mounted on any mount point, in addition to the standard
\fB/proc\fR mount point, and can be mounted several places at once. Such
additional mounts are allowed in order to facilitate the confinement of
processes to subtrees of the file system via \fBchroot\fR(1M) and yet allow
such processes access to commands like \fBps\fR(1).
.sp
.LP
Standard system calls are used to access \fB/proc\fR files: \fBopen\fR(2),
\fBclose\fR(2), \fBread\fR(2), and \fBwrite\fR(2) (including \fBreadv\fR(2),
\fBwritev\fR(2), \fBpread\fR(2), and \fBpwrite\fR(2)). Most files describe
process state and can only be opened for reading. \fBctl\fR and \fBlwpctl\fR
(control) files permit manipulation of process state and can only be opened for
writing. \fBas\fR (address space) files contain the image of the running
process and can be opened for both reading and writing. An open for writing
allows process control; a read-only open allows inspection but not control. In
this document, we refer to the process as open for reading or writing if any of
its associated \fB/proc\fR files is open for reading or writing.
.sp
.LP
In general, more than one process can open the same \fB/proc\fR file at the
same time. \fIExclusive\fR \fIopen\fR is an advisory mechanism provided to
allow controlling processes to avoid collisions with each other. A process can
obtain exclusive control of a target process, with respect to other cooperating
processes, if it successfully opens any \fB/proc\fR file in the target process
for writing (the \fBas\fR or \fBctl\fR files, or the \fBlwpctl\fR file of any
lwp) while specifying \fBO_EXCL\fR in the \fBopen\fR(2). Such an open will fail
if the target process is already open for writing (that is, if an \fBas\fR,
\fBctl\fR, or \fBlwpctl\fR file is already open for writing). There can be any
number of concurrent read-only opens; \fBO_EXCL\fR is ignored on opens for
reading. It is recommended that the first open for writing by a controlling
process use the \fBO_EXCL\fR flag; multiple controlling processes usually
result in chaos.
.sp
.LP
If a process opens one of its own \fB/proc\fR files for writing, the open
```



```
succeeds regardless of \fBO_EXCL\fR and regardless of whether some other
process has the process open for writing. Self-opens do not count when another
process attempts an exclusive open. (A process cannot exclude a debugger by
opening itself for writing and the application of a debugger cannot prevent a
process from opening itself.) All self-opens for writing are forced to be
close-on-exec (see the \fBF_SETFD\fR operation of \fBfcntl\fR(2)).
.sp
.LP
Data may be transferred from or to any locations in the address space of the
traced process by applying \fBlseek\fR(2) to position the \fBas\fR file at the
virtual address of interest followed by \fBread\fR(2) or \fBwrite\fR(2) (or by
using \fBpread\fR(2) or \fBpwrite\fR(2) for the combined operation). The
address-map files \fB/proc/\fR\fIpid\fR\fB/map\fR and
\fB/proc/\fR\fIpid\fR\fB/xmap\fR can be read to determine the accessible areas
(mappings) of the address space. \fBI/O\fR transfers may span contiguous
mappings. An \fBI/O\fR request extending into an unmapped area is truncated at
the boundary. A write request beginning at an unmapped virtual address fails
with \fBEIO\fR; a read request beginning at an unmapped virtual address returns
zero (an end-of-file indication).
.sp
.LP
Information and control operations are provided through additional files.
\fB<procfs.h>\fR contains definitions of data structures and message formats
used with these files. Some of these definitions involve the use of sets of
flags. The set types \fBsigset_t\fR, \fBfltset_t\fR, and \fBsysset_t\fR
correspond, respectively, to signal, fault, and system call enumerations
defined in \fB<sys/signal.h>\fR, \fB<sys/fault.h>\fR, and
\fB<sys/syscall.h>\fR\&. Each set type is large enough to hold flags for its
own enumeration. Although they are of different sizes, they have a common
structure and can be manipulated by these macros:
.sp
.in +2
.nf
prfillset(&set);             /* turn on all flags in set */
premptyset(&set);            /* turn off all flags in set */
praddset(&set, flag);        /* turn on the specified flag */
prdelset(&set, flag);        /* turn off the specified flag */
r = prismember(&set, flag);  /* != 0 iff flag is turned on */
.fi
.in -2

.sp
.LP
One of \fBprfillset()\fR or \fBpremptyset()\fR must be used to initialize
\fBset\fR before it is used in any other operation. \fBflag\fR must be a member
of the enumeration corresponding to \fBset\fR.
.sp
.LP
Every process contains at least one \fIlight-weight process\fR, or \fIlwp\fR.
Each lwp represents a flow of execution that is independently scheduled by the
operating system. All lwps in a process share its address space as well as many
other attributes. Through the use of \fBlwpctl\fR and \fBctl\fR files as
described below, it is possible to affect individual lwps in a process or to
affect all of them at once, depending on the operation.
.sp
.LP
When the process has more than one lwp, a representative lwp is chosen by the
system for certain process status files and control operations. The
representative lwp is a stopped lwp only if all of the process's lwps are
stopped; is stopped on an event of interest only if all of the lwps are so
stopped (excluding \fBPR_SUSPENDED\fR lwps); is in a \fBPR_REQUESTED\fR stop
only if there are no other events of interest to be found; or, failing
everything else, is in a \fBPR_SUSPENDED\fR stop (implying that the process is
deadlocked). See the description of the \fBstatus\fR file for definitions of
stopped states. See the \fBPCSTOP\fR control operation for the definition of
``event of interest''.
```

```
.sp
.LP
The representative lwp remains fixed (it will be chosen again on the next
operation) as long as all of the lwps are stopped on events of interest or are
in a \fBPR_SUSPENDED\fR stop and the \fBPCRUN\fR control operation is not
applied to any of them.
.sp
.LP
When applied to the process control file, every \fB/proc\fR control operation
that must act on an lwp uses the same algorithm to choose which lwp to act
upon. Together with synchronous stopping (see \fBPCSET\fR), this enables a
debugger to control a multiple-lwp process using only the process-level status
and control files if it so chooses. More fine-grained control can be achieved
using the lwp-specific files.
.sp
.LP
The system supports two process data models, the traditional 32-bit data model
in which ints, longs and pointers are all 32 bits wide (the ILP32 data model),
and on some platforms the 64-bit data model in which longs and pointers, but
not ints, are 64 bits in width (the LP64 data model). In the LP64 data model
some system data types, notably \fBsize_t\fR, \fBoff_t\fR, \fBtime_t\fR and
\fBdev_t\fR, grow from 32 bits to 64 bits as well.
.sp
.LP
The \fB/proc\fR interfaces described here are available to both 32-bit and
64-bit controlling processes. However, many operations attempted by a 32-bit
controlling process on a 64-bit target process will fail with \fBEOVERFLOW\fR
because the address space range of a 32-bit process cannot encompass a 64-bit
process or because the data in some 64-bit system data type cannot be
compressed to fit into the corresponding 32-bit type without loss of
information. Operations that fail in this circumstance include reading and
writing the address space, reading the address-map files, and setting the
target process's registers. There is no restriction on operations applied by a
64-bit process to either a 32-bit or a 64-bit target processes.
.sp
.LP
The format of the contents of any \fB/proc\fR file depends on the data model of
the observer (the controlling process), not on the data model of the target
process. A 64-bit debugger does not have to translate the information it reads
from a \fB/proc\fR file for a 32-bit process from 32-bit format to 64-bit
format. However, it usually has to be aware of the data model of the target
process. The \fBpr_dmodel\fR field of the \fBstatus\fR files indicates the
target process's data model.
.sp
.LP
To help deal with system data structures that are read from 32-bit processes, a
64-bit controlling program can be compiled with the C preprocessor symbol
\fB_SYSCALL32\fR defined before system header files are included. This makes
explicit 32-bit fixed-width data structures (like \fBcstruct stat32\fR) visible
to the 64-bit program. See \fBtypes32.h\fR(3HEAD).
.SH DIRECTORY STRUCTURE
.LP
At the top level, the directory \fB/proc\fR contains entries each of which
names an existing process in the system. These entries are themselves
directories. Except where otherwise noted, the files described below can be
opened for reading only. In addition, if a process becomes a \fIzombie\fR (one
that has exited but whose parent has not yet performed a \fBwait\fR(3C) upon
it), most of its associated \fB/proc\fR files disappear from the hierarchy;
subsequent attempts to open them, or to read or write files opened before the
process exited, will elicit the error \fBENOENT\fR.
.sp
.LP
Although process state and consequently the contents of \fB/proc\fR files can
change from instant to instant, a single \fBread\fR(2) of a \fB/proc\fR file is
guaranteed to return a sane representation of state; that is, the read will be
atomic with respect to the state of the process. No such guarantee applies to
```



```
successive reads applied to a \fB/proc\fR file for a running process. In
addition, atomicity is not guaranteed for \fBI/O\fR applied to the \fBas\fR
(address-space) file for a running process or for a process whose address space
contains memory shared by another running process.
.sp
.LP
A number of structure definitions are used to describe the files. These
structures may grow by the addition of elements at the end in future releases
of the system and it is not legitimate for a program to assume that they will
not.
.SH STRUCTURE OF \fB/proc/\fR\fIpid\fR
.LP
A given directory \fB/proc/\fR\fIpid\fR contains the following entries. A
process can use the invisible alias \fB/proc/self\fR if it wishes to open one
of its own \fB/proc\fR files (invisible in the sense that the name ''self''
does not appear in a directory listing of \fB/proc\fR obtained from
\fBls\fR(1), \fBgetdents\fR(2), or \fBreaddir\fR(3C)).
.SS "contracts"
.LP
A directory containing references to the contracts held by the process. Each
entry is a symlink to the contract's directory under \fB/system/contract\fR.
See \fBcontract\fR(4).
.SS "as"
.LP
Contains the address-space image of the process; it can be opened for both
reading and writing. \fBlseek\fR(2) is used to position the file at the virtual
address of interest and then the address space can be examined or changed
through \fBread\fR(2) or \fBwrite\fR(2) (or by using \fBpread\fR(2) or
\fBpwrite\fR(2) for the combined operation).
.SS "ctl"
.LP
A write-only file to which structured messages are written directing the system
to change some aspect of the process's state or control its behavior in some
way. The seek offset is not relevant when writing to this file. Individual lwps
also have associated \fBlwpctl\fR files in the lwp subdirectories. A control
message may be written either to the process's \fBctl\fR file or to a specific
\fBlwpctl\fR file with operation-specific effects. The effect of a control
message is immediately reflected in the state of the process visible through
appropriate status and information files. The types of control messages are
described in detail later. See \fBCONTROL MESSAGES\fR.
.SS "status"
.LP
Contains state information about the process and the representative lwp. The
file contains a \fBpstatus\fR structure which contains an embedded
\fBlwpstatus\fR structure for the representative lwp, as follows:
.sp
.in +2
.nf
typedef struct pstatus {
     int pr_flags;          /* flags (see below) */
     int pr_nlwp;           /* number of active lwps in the process */
     int pr_nzomb;          /* number of zombie lwps in the process */
     pid_tpr_pid;           /* process id */
     pid_tpr_ppid;          /* parent process id */
     pid_tpr_pgid;          /* process group id */
     pid_tpr_sid;           /* session id */
     id_t pr_aslwpid;       /* obsolete */
     id_t pr_agentid;       /* lwp-id of the agent lwp, if any */
     sigset_t pr_sigpend;   /* set of process pending signals */
     uintptr_t pr_brkbase;  /* virtual address of the process heap */
     size_t pr_brksize;     /* size of the process heap, in bytes */
     uintptr_t pr_stkbase;  /* virtual address of the process stack */
     size_tpr_stksize;      /* size of the process stack, in bytes */
     timestruc_t pr_utime;  /* process user cpu time */
     timestruc_t pr_stime;  /* process system cpu time */
     timestruc_t pr_cutime; /* sum of children's user times */
```

```
257      timestruc_t pr_cstime;   /* sum of children's system times */
258      sigset_t pr_sigtrace;    /* set of traced signals */
259      fltset_t pr_flttrace;    /* set of traced faults */
260      sysset_t pr_sysentry;    /* set of system calls traced on entry */
261      sysset_t pr_sysexit;     /* set of system calls traced on exit */
262      char pr_dmodel;          /* data model of the process */
263      taskid_t pr_taskid;      /* task id */
264      projid_t pr_projid;      /* project id */
265      zoneid_t pr_zoneid;      /* zone id */
266      lwpstatus_t pr_lwp;      /* status of the representative lwp */
267 } pstatus_t;
_____unchanged_portion_omitted_
493 .fi
494 .in -2

496 .sp
497 .LP
498 \fBpr_flags\fR is a bit-mask holding the following lwp flags. For convenience,
499 it also contains the process flags, described previously.
500 .sp
501 .ne 2
502 .na
503 \fB\fBPR_STOPPED\fR\fR
504 .ad
505 .RS 14n
506 The lwp is stopped.
507 .RE

509 .sp
510 .ne 2
511 .na
512 \fB\fBPR_ISTOP\fR\fR
513 .ad
514 .RS 14n
515 The lwp is stopped on an event of interest (see \fBPCSTOP\fR).
516 .RE

518 .sp
519 .ne 2
520 .na
521 \fB\fBPR_DSTOP\fR\fR
522 .ad
523 .RS 14n
524 The lwp has a stop directive in effect (see \fBPCSTOP\fR).
525 .RE

527 .sp
528 .ne 2
529 .na
530 \fB\fBPR_STEP\fR\fR
531 .ad
532 .RS 14n
533 The lwp has a single-step directive in effect (see \fBPCRUN\fR).
534 .RE

536 .sp
537 .ne 2
538 .na
539 \fB\fBPR_ASLEEP\fR\fR
540 .ad
541 .RS 14n
542 The lwp is in an interruptible sleep within a system call.
543 .RE

545 .sp
546 .ne 2
```



```
547 .na
548 \fB\fBPR_PCINVAL\fR\fR
549 .ad
550 .RS 14n
551 The lwp's current instruction (\fBpr_instr\fR) is undefined.
552 .RE

554 .sp
555 .ne 2
556 .na
557 \fB\fBPR_DETACH\fR\fR
558 .ad
559 .RS 14n
560 This is a detached lwp (see \fBpthread_create\fR(3C) and
561 \fBpthread_join\fR(3C)).
562 .RE

564 .sp
565 .ne 2
566 .na
567 \fB\fBPR_DAEMON\fR\fR
568 .ad
569 .RS 14n
570 This is a daemon lwp (see \fBpthread_create\fR(3C)).
571 .RE

573 .sp
574 .ne 2
575 .na
576 \fB\fBPR_ASLWP\fR\fR
577 .ad
578 .RS 14n
579 This flag is obsolete and is never set.
580 .RE

582 .sp
583 .ne 2
584 .na
585 \fB\fBPR_AGENT\fR\fR
586 .ad
587 .RS 14n
588 This is the \fB/proc\fR agent lwp for the process.
589 .RE

591 .sp
592 .LP
593 \fBpr_lwpid\fR names the specific lwp.
594 .sp
595 .LP
596 \fBpr_why\fR and \fBpr_what\fR together describe, for a stopped lwp, the reason
597 for the stop. Possible values of \fBpr_why\fR and the associated \fBpr_what\fR
598 are:
599 .sp
600 .ne 2
601 .na
602 \fB\fBPR_REQUESTED\fR\fR
603 .ad
604 .RS 17n
605 indicates that the stop occurred in response to a stop directive, normally
606 because \fBPCSTOP\fR was applied or because another lwp stopped on an event of
607 interest and the asynchronous-stop flag (see \fBPCSET\fR) was not set for the
608 process. \fBpr_what\fR is unused in this case.
609 .RE

611 .sp
612 .ne 2
```

```
.na
\fB\fBPR_SIGNALLED\fR\fR
.ad
.RS 17n
indicates that the lwp stopped on receipt of a signal (see \fBPCSTRACE\fR);
\fBpr_what\fR holds the signal number that caused the stop (for a newly-stopped
lwp, the same value is in \fBpr_cursig\fR).
.RE

.sp
.ne 2
.na
\fB\fBPR_FAULTED\fR\fR
.ad
.RS 17n
indicates that the lwp stopped on incurring a hardware fault (see
\fBPCSFAULT\fR); \fBpr_what\fR holds the fault number that caused the stop.
.RE

.sp
.ne 2
.na
\fB\fBPR_SYSENTRY\fR\fR
.ad
.br
.na
\fB\fBPR_SYSEXIT\fR\fR
.ad
.RS 17n
indicate a stop on entry to or exit from a system call (see \fBPCSENTRY\fR and
\fBPCSEXIT\fR); \fBpr_what\fR holds the system call number.
.RE

.sp
.ne 2
.na
\fB\fBPR_JOBCONTROL\fR\fR
.ad
.RS 17n
indicates that the lwp stopped due to the default action of a job control stop
signal (see \fBsigaction\fR(2)); \fBpr_what\fR holds the stopping signal
number.
.RE

.sp
.ne 2
.na
\fB\fBPR_SUSPENDED\fR\fR
.ad
.RS 17n
indicates that the lwp stopped due to internal synchronization of lwps within
the process. \fBpr_what\fR is unused in this case.
.RE

.sp
.LP
\fBpr_cursig\fR names the current signal, that is, the next signal to be
delivered to the lwp, if any. \fBpr_info\fR, when the lwp is in a
\fBPR_SIGNALLED\fR or \fBPR_FAULTED\fR stop, contains additional information
pertinent to the particular signal or fault (see \fB<sys/siginfo.h>\fR).
.sp
.LP
\fBpr_lwppend\fR identifies any synchronous or directed signals pending for the
lwp. \fBpr_lwphold\fR identifies those signals whose delivery is being blocked
by the lwp (the signal mask).
.sp
```



```
.LP
\fBpr_action\fR contains the signal action information pertaining to the
current signal (see \fBsigaction\fR(2)); it is undefined if \fBpr_cursig\fR is
zero. \fBpr_altstack\fR contains the alternate signal stack information for the
lwp (see \fBsigaltstack\fR(2)).
.sp
.LP
\fBpr_oldcontext\fR, if not zero, contains the address on the lwp stack of a
\fBucontext\fR structure describing the previous user-level context (see
\fBucontext.h\fR(3HEAD)). It is non-zero only if the lwp is executing in the
context of a signal handler.
.sp
.LP
\fBpr_syscall\fR is the number of the system call, if any, being executed by
the lwp; it is non-zero if and only if the lwp is stopped on \fBPR_SYSENTRY\fR
or \fBPR_SYSEXIT\fR, or is asleep within a system call ( \fBPR_ASLEEP\fR is
set). If \fBpr_syscall\fR is non-zero, \fBpr_nsysarg\fR is the number of
arguments to the system call and \fBpr_sysarg\fR contains the actual arguments.
.sp
.LP
\fBpr_rval1\fR, \fBpr_rval2\fR, and \fBpr_errno\fR are defined only if the lwp
is stopped on \fBPR_SYSEXIT\fR or if the \fBPR_VFORKP\fR flag is set. If
\fBpr_errno\fR is zero, \fBpr_rval1\fR and \fBpr_rval2\fR contain the return
values from the system call. Otherwise, \fBpr_errno\fR contains the error
number for the failing system call (see \fB<sys/errno.h>\fR).
.sp
.LP
\fBpr_clname\fR contains the name of the lwp's scheduling class.
.sp
.LP
\fBpr_tstamp\fR, if the lwp is stopped, contains a time stamp marking when the
lwp stopped, in real time seconds and nanoseconds since an arbitrary time in
the past.
.sp
.LP
\fBpr_utime\fR is the amount of user level CPU time used by this LWP.
.sp
.LP
\fBpr_stime\fR is the amount of system level CPU time used by this LWP.
.sp
.LP
\fBpr_ustack\fR is the virtual address of the \fBstack_t\fR that contains the
stack boundaries for this LWP. See \fBgetustack\fR(2) and
\fB_stack_grow\fR(3C).
.sp
.LP
\fBpr_instr\fR contains the machine instruction to which the lwp's program
counter refers. The amount of data retrieved from the process is
machine-dependent. On SPARC based machines, it is a 32-bit word. On x86-based
machines, it is a single byte. In general, the size is that of the machine's
smallest instruction. If \fBPR_PCINVAL\fR is set, \fBpr_instr\fR is undefined;
this occurs whenever the lwp is not stopped or when the program counter refers
to an invalid virtual address.
.sp
.LP
\fBpr_reg\fR is an array holding the contents of a stopped lwp's general
registers.
.sp
.ne 2
.na
\fBSPARC\fR
.ad
.RS 21n
On SPARC-based machines, the predefined constants \fBR_G0\fR ... \fBR_G7\fR,
\fBR_O0\fR ... \fBR_O7\fR, \fBR_L0\fR ... \fBR_L7\fR, \fBR_I0\fR ...
\fBR_I7\fR, \fBR_PC\fR, \fBR_nPC\fR, and \fBR_Y\fR can be used as indices to
```

```
refer to the corresponding registers; previous register windows can be read
from their overflow locations on the stack (however, see the \fBgwindows\fR
file in the \fB/proc/\fR\fIpid\fR\fB/lwp/\fR\fIlwpid\fR subdirectory).
.RE

.sp
.ne 2
.na
\fBSPARC V8 (32-bit)\fR
.ad
.RS 21n
For SPARC V8 (32-bit) controlling processes, the predefined constants
\fBR_PSR\fR, \fBR_WIM\fR, and \fBR_TBR\fR can be used as indices to refer to
the corresponding special registers. For SPARC V9 (64-bit) controlling
processes, the predefined constants \fBR_CCR\fR, \fBR_ASI\fR, and \fBR_FPRS\fR
can be used as indices to refer to the corresponding special registers.
.RE

.sp
.ne 2
.na
\fBx86 (32-bit)\fR
.ad
.RS 21n
For 32-bit x86 processes, the predefined constants listed belowcan be used as
indices to refer to the corresponding registers.
.sp
.in +2
.nf
SS
UESP
EFL
CS
EIP
ERR
TRAPNO
EAX
ECX
EDX
EBX
ESP
EBP
ESI
EDI
DS
ES
GS
.fi
.in -2

The preceding constants are listed in \fB<sys/regset.h>\fR\&.
.sp
Note that a 32-bit process can run on an x86 64-bit system, using the constants
listed above.
.RE

.sp
.ne 2
.na
\fBx86 (64-bit)\fR
.ad
.RS 21n
To read the registers of a 32- \fBor\fR a 64-bit process, a 64-bit x86 process
should use the predefined constants listed below.
.sp
.in +2
```



```
.nf
REG_GSBASE
REG_FSBASE
REG_DS
REG_ES
REG_GS
REG_FS
REG_SS
REG_RSP
REG_RFL
REG_CS
REG_RIP
REG_ERR
REG_TRAPNO
REG_RAX
REG_RCX
REG_RDX
REG_RBX
REG_RBP
REG_RSI
REG_RDI
REG_R8
REG_R9
REG_R10
REG_R11
REG_R12
REG_R13
REG_R14
REG_R15
.fi
.in -2

The preceding constants are listed in \fB<sys/regset.h>\fR\&.
.RE

.sp
.LP
\fBpr_fpreg\fR is a structure holding the contents of the floating-point
registers.
.sp
.LP
SPARC registers, both general and floating-point, as seen by a 64-bit
controlling process are the V9 versions of the registers, even if the target
process is a 32-bit (V8) process. V8 registers are a subset of the V9
registers.
.sp
.LP
If the lwp is not stopped, all register values are undefined.
.SS "psinfo"
.LP
Contains miscellaneous information about the process and the representative lwp
needed by the \fBps\fR(1) command. \fBpsinfo\fR remains accessible after a
process becomes a \fIzombie\fR. The file contains a \fBpsinfo\fR structure
which contains an embedded \fBlwpsinfo\fR structure for the representative lwp,
as follows:
.sp
.in +2
.nf
typedef struct psinfo {
    int pr_flag;              /* process flags (DEPRECATED: see below) */
    int pr_nlwp;              /* number of active lwps in the process */
    int pr_nzomb;             /* number of zombie lwps in the process */
    pid_t pr_pid;             /* process id */
    pid_t pr_ppid;            /* process id of parent */
    pid_t pr_pgid;            /* process id of process group leader */
    pid_t pr_sid;             /* session id */
```

```
 877     uid_t pr_uid;           /* real user id */
 878     uid_t pr_euid;          /* effective user id */
 879     gid_t pr_gid;           /* real group id */
 880     gid_t pr_egid;          /* effective group id */
 881     uintptr_t pr_addr;      /* address of process */
 882     size_t pr_size;         /* size of process image in Kbytes */
 883     size_t pr_rssize;       /* resident set size in Kbytes */
 884     dev_t pr_ttydev;        /* controlling tty device (or PRNODEV) */
 885     ushort_t pr_pctcpu;     /* % of recent cpu time used by all lwps */
 886     ushort_t pr_pctmem;     /* % of system memory used by process */
 887     timestruc_t pr_start;   /* process start time, from the epoch */
 888     timestruc_t pr_time;    /* cpu time for this process */
 889     timestruc_t pr_ctime;   /* cpu time for reaped children */
 890     char pr_fname[PRFNSZ];  /* name of exec'ed file */
 891     char pr_psargs[PRARGSZ]; /* initial characters of arg list */
 892     int pr_wstat;           /* if zombie, the wait() status */
 893     int pr_argc;            /* initial argument count */
 894     uintptr_t pr_argv;      /* address of initial argument vector */
 895     uintptr_t pr_envp;      /* address of initial environment vector */
 896     char pr_dmodel;         /* data model of the process */
 897     lwpsinfo_t pr_lwp;      /* information for representative lwp */
 897     taskid_t pr_taskid;     /* task id */
 898     projid_t pr_projid;     /* project id */
 899     poolid_t pr_poolid;     /* pool id */
 900     zoneid_t pr_zoneid;     /* zone id */
 901     ctid_t pr_contract;     /* process contract id */
 902     lwpsinfo_t pr_lwp;      /* information for representative lwp */
 903 } psinfo_t;
_____unchanged_portion_omitted_
1447 .fi
1448 .in -2

1450 .sp
1451 .LP
1452 The \fBlwpstatus\fR structure may grow by the addition of elements at the end
1453 in future releases of the system. Programs must use \fBpr_entsize\fR in the
1454 file header to index through the array. These comments apply to all \fB/proc\fR
1455 files that include a \fBprheader\fR structure (\fBlpsinfo\fR and \fBlusage\fR,
1456 below).
1457 .SS "lpsinfo"
1458 .LP
1459 Contains a \fBprheader\fR structure followed by an array of \fBlwpsinfo\fR
1460 structures, one for eachactive and zombie lwp in the process. See also
1461 \fB/proc/\fR\fIpid\fR\fB/lwp/\fR\fIlwpid\fR/\fBlwpsinfo\fR, below.
1462 .SS "lusage"
1463 .LP
1464 Contains a \fBprheader\fR structure followed by an array of \fBprusage\fR
1465 structures, one for each active lwp in the process, plus an additional element
1466 at the beginning that contains the summation over all defunct lwps (lwps that
1467 once existed but no longer exist in the process). Excluding the \fBpr_lwpid\fR,
1468 \fBpr_tstamp\fR, \fBpr_create\fR, and \fBpr_term\fR entries, the entry-by-entry
1469 summation over all these structures is the definition of the process usage
1470 information obtained from the \fBusage\fR file. (See also
1471 \fB/proc/\fR\fIpid\fR\fB/lwp/\fR\fIlwpid\fR/\fBlwpusage\fR, below.)
1472 .SS "lwp"
1473 .LP
1474 A directory containing entries each of which names an active or zombie lwp
1475 within the process. These entries are themselves directories containing
1476 additional files as described below. Only the \fBlwpsinfo\fR file exists in the
1477 directory of a zombie lwp.
1478 .SH STRUCTURE OF \fB/proc/\fR\fIpid\fR\fB/lwp/\fR\fIlwpid\fR
1479 .LP
1480 A given directory \fB/proc/\fR\fIpid\fR\fB/lwp/\fR\fIlwpid\fR contains the
1481 following entries:
1482 .SS "lwpctl"
1483 .LP
```



```
1484 Write-only control file. The messages written to this file affect the specific
1485 lwp rather than the representative lwp, as is the case for the process's
1486 \fBctl\fR file.
1487 .SS lwpname
1488 A buffer of \fBTHREAD_NAME_MAX\fR bytes representing the LWP name; the buffer is
1489 zero-filled if the thread name is shorter than the buffer. If no thread name is
1490 set, the buffer contains the empty string. A read with a buffer shorter than
1491 \fBTHREAD_NAME_MAX\fR bytes is not guaranteed to be NUL-terminated.  Writing to
1492 this file will set the LWP name for the specific lwp.  This file may not be
1493 present in older operating system versions. \fBTHREAD_NAME_MAX\fR may increase
1494 in the future; clients should be prepared for this.
1495 .SS "lwpstatus"
1496 .LP
1497 lwp-specific state information. This file contains the \fBlwpstatus\fR
1498 structure for the specific lwp as described above for the representative lwp in
1499 the process's \fBstatus\fR file.
1500 .SS "lwpsinfo"
1501 .LP
1502 lwp-specific \fBps\fR(1) information. This file contains the \fBlwpsinfo\fR
1503 structure for the specific lwp as described above for the representative lwp in
1504 the process's \fBpsinfo\fR file. The \fBlwpsinfo\fR file remains accessible
1505 after an lwp becomes a zombie.
1506 .SS "lwpusage"
1507 .LP
1508 This file contains the \fBprusage\fR structure for the specific lwp as
1509 described above for the process's \fBusage\fR file.
1510 .SS "gwindows"
1511 .LP
1512 This file exists only on SPARC based machines. If it is non-empty, it contains
1513 a \fBgwindows_t\fR structure, defined in \fB<sys/regset.h>\fR, with the values
1514 of those SPARC register windows that could not be stored on the stack when the
1515 lwp stopped. Conditions under which register windows are not stored on the
1516 stack are: the stack pointer refers to nonexistent process memory or the stack
1517 pointer is improperly aligned. If the lwp is not stopped or if there are no
1518 register windows that could not be stored on the stack, the file is empty (the
1519 usual case).
1520 .SS "xregs"
1521 .LP
1522 Extra state registers. The extra state register set is architecture dependent;
1523 this file is empty if the system does not support extra state registers. If the
1524 file is non-empty, it contains an architecture dependent structure of type
1525 \fBprxregset_t\fR, defined in \fB<procfs.h>\fR, with the values of the lwp's
1526 extra state registers. If the lwp is not stopped, all register values are
1527 undefined. See also the \fBPCSXREG\fR control operation, below.
1528 .SS "asrs"
1529 .LP
1530 This file exists only for 64-bit SPARC V9 processes. It contains an
1531 \fBasrset_t\fR structure, defined in <\fBsys/regset.h\fR>, containing the
1532 values of the lwp's platform-dependent ancillary state registers. If the lwp is
1533 not stopped, all register values are undefined. See also the \fBPCSASRS\fR
1534 control operation, below.
1535 .SS "spymaster"
1536 .LP
1537 For an agent lwp (see \fBPCAGENT\fR), this file contains a \fBpsinfo_t\fR
1538 structure that corresponds to the process that created the agent lwp at the
1539 time the agent was created. This structure is identical to that retrieved via
1540 the \fBpsinfo\fR file, with one modification: the \fBpr_time\fR field does not
1541 correspond to the CPU time for the process, but rather to the creation time of
1542 the agent lwp.
1543 .SS "templates"
1544 .LP
1545 A directory which contains references to the active templates for the lwp,
1546 named by the contract type. Changes made to an active template descriptor do
1547 not affect the original template which was activated, though they do affect the
1548 active template. It is not possible to activate an active template descriptor.
1549 See \fBcontract\fR(4).
```

```
.SH CONTROL MESSAGES
.LP
Process state changes are effected through messages written to a process's
\fBctl\fR file or to an individual lwp's \fBlwpctl\fR file. All control
messages consist of a \fBlong\fR that names the specific operation followed by
additional data containing the operand, if any.
.sp
.LP
Multiple control messages may be combined in a single \fBwrite\fR(2) (or
\fBwritev\fR(2)) to a control file, but no partial writes are permitted. That
is, each control message, operation code plus operand, if any, must be
presented in its entirety to the \fBwrite\fR(2) and not in pieces over several
system calls. If a control operation fails, no subsequent operations contained
in the same \fBwrite\fR(2) are attempted.
.sp
.LP
Descriptions of the allowable control messages follow. In all cases, writing a
message to a control file for a process or lwp that has terminated elicits the
error \fBENOENT\fR.
.SS "PCSTOP PCDSTOP PCWSTOP PCTWSTOP"
.LP
When applied to the process control file, \fBPCSTOP\fR directs all lwps to stop
and waits for them to stop, \fBPCDSTOP\fR directs all lwps to stop without
waiting for them to stop, and \fBPCWSTOP\fR simply waits for all lwps to stop.
When applied to an lwp control file, \fBPCSTOP\fR directs the specific lwp to
stop and waits until it has stopped, \fBPCDSTOP\fR directs the specific lwp to
stop without waiting for it to stop, and \fBPCWSTOP\fR simply waits for the
specific lwp to stop. When applied to an lwp control file, \fBPCSTOP\fR and
\fBPCWSTOP\fR complete when the lwp stops on an event of interest, immediately
if already so stopped; when applied to the process control file, they complete
when every lwp has stopped either on an event of interest or on a
\fBPR_SUSPENDED\fR stop.
.sp
.LP
\fBPCTWSTOP\fR is identical to \fBPCWSTOP\fR except that it enables the
operation to time out, to avoid waiting forever for a process or lwp that may
never stop on an event of interest. \fBPCTWSTOP\fR takes a \fBlong\fR operand
specifying a number of milliseconds; the wait will terminate successfully after
the specified number of milliseconds even if the process or lwp has not
stopped; a timeout value of zero makes the operation identical to
\fBPCWSTOP\fR.
.sp
.LP
An ''event of interest'' is either a \fBPR_REQUESTED\fR stop or a stop that has
been specified in the process's tracing flags (set by \fBPCSTRACE\fR,
\fBPCSFAULT\fR, \fBPCSENTRY\fR, and \fBPCSEXIT\fR). \fBPR_JOBCONTROL\fR and
\fBPR_SUSPENDED\fR stops are specifically not events of interest. (An lwp may
stop twice due to a stop signal, first showing \fBPR_SIGNALLED\fR if the signal
is traced and again showing \fBPR_JOBCONTROL\fR if the lwp is set running
without clearing the signal.) If \fBPCSTOP\fR or \fBPCDSTOP\fR is applied to an
lwp that is stopped, but not on an event of interest, the stop directive takes
effect when the lwp is restarted by the competing mechanism. At that time, the
lwp enters a \fBPR_REQUESTED\fR stop before executing any user-level code.
.sp
.LP
A write of a control message that blocks is interruptible by a signal so that,
for example, an \fBalarm\fR(2) can be set to avoid waiting forever for a
process or lwp that may never stop on an event of interest. If \fBPCSTOP\fR is
interrupted, the lwp stop directives remain in effect even though the
\fBwrite\fR(2) returns an error. (Use of \fBPCTWSTOP\fR with a non-zero timeout
is recommended over \fBPCWSTOP\fR with an \fBalarm\fR(2).)
.sp
.LP
A system process (indicated by the \fBPR_ISSYS\fR flag) never executes at user
level, has no user-level address space visible through \fB/proc\fR, and cannot
be stopped. Applying one of these operations to a system process or any of its
```



```
lwps elicits the error \fBEBUSY\fR.
.SS "PCRUN"
.LP
Make an lwp runnable again after a stop. This operation takes a \fBlong\fR
operand containing zero or more of the following flags:
.sp
.ne 2
.na
\fB\fBPRCSIG\fR\fR
.ad
.RS 12n
clears the current signal, if any (see \fBPCCSIG\fR).
.RE

.sp
.ne 2
.na
\fB\fBPRCFAULT\fR\fR
.ad
.RS 12n
clears the current fault, if any (see \fBPCCFAULT\fR).
.RE

.sp
.ne 2
.na
\fB\fBPRSTEP\fR\fR
.ad
.RS 12n
directs the lwp to execute a single machine instruction. On completion of the
instruction, a trace trap occurs. If \fBFLTTRACE\fR is being traced, the lwp
stops; otherwise, it is sent \fBSIGTRAP\fR. If \fBSIGTRAP\fR is being traced
and is not blocked, the lwp stops. When the lwp stops on an event of interest,
the single-step directive is cancelled, even if the stop occurs before the
instruction is executed. This operation requires hardware and operating system
support and may not be implemented on all processors. It is implemented on
SPARC and x86-based machines.
.RE

.sp
.ne 2
.na
\fB\fBPRSABORT\fR\fR
.ad
.RS 12n
is meaningful only if the lwp is in a \fBPR_SYSENTRY\fR stop or is marked
\fBPR_ASLEEP\fR; it instructs the lwp to abort execution of the system call
(see \fBPCSENTRY\fR and \fBPCSEXIT\fR).
.RE

.sp
.ne 2
.na
\fB\fBPRSTOP\fR\fR
.ad
.RS 12n
directs the lwp to stop again as soon as possible after resuming execution (see
\fBPCDSTOP\fR). In particular, if the lwp is stopped on \fBPR_SIGNALLED\fR or
\fBPR_FAULTED\fR, the next stop will show \fBPR_REQUESTED\fR, no other stop
will have intervened, and the lwp will not have executed any user-level code.
.RE

.sp
.LP
When applied to an lwp control file, \fBPCRUN\fR clears any outstanding
directed-stop request and makes the specific lwp runnable. The operation fails
```

with \fBEBUSY\fR if the specific lwp is not stopped on an event of interest or has not been directed to stop or if the agent lwp exists and this is not the agent lwp (see \fBPCAGENT\fR).

.LP

When applied to the process control file, a representative lwp is chosen for the operation as described for \fB/proc/\fR\fIpid\fR\fB/status\fR. The operation fails with \fBEBUSY\fR if the representative lwp is not stopped on an event of interest or has not been directed to stop or if the agent lwp exists. If \fBPRSTEP\fR or \fBPRSTOP\fR was requested, the representative lwp is made runnable and its outstanding directed-stop request is cleared; otherwise all outstanding directed-stop requests are cleared and, if it was stopped on an event of interest, the representative lwp is marked \fBPR_REQUESTED\fR. If, as a consequence, all lwps are in the \fBPR_REQUESTED\fR or \fBPR_SUSPENDED\fR stop state, all lwps showing \fBPR_REQUESTED\fR are made runnable.

.SS "PCSTRACE"

.LP

Define a set of signals to be traced in the process. The receipt of one of these signals by an lwp causes the lwp to stop. The set of signals is defined using an operand \fBsigset_t\fR contained in the control message. Receipt of \fBSIGKILL\fR cannot be traced; if specified, it is silently ignored.

.sp

.LP

If a signal that is included in an lwp’s held signal set (the signal mask) is sent to the lwp, the signal is not received and does not cause a stop until it is removed from the held signal set, either by the lwp itself or by setting the held signal set with \fBPCSHOLD\fR.

.SS "PCCSIG"

.LP

The current signal, if any, is cleared from the specific or representative lwp.

.SS "PCSSIG"

.LP

The current signal and its associated signal information for the specific or representative lwp are set according to the contents of the operand \fBsiginfo\fR structure (see \fB<sys/siginfo.h>\fR). If the specified signal number is zero, the current signal is cleared. The semantics of this operation are different from those of \fBkill\fR(2) in that the signal is delivered to the lwp immediately after execution is resumed (even if it is being blocked) and an additional \fBPR_SIGNALLED\fR stop does not intervene even if the signal is traced. Setting the current signal to \fBSIGKILL\fR terminates the process immediately.

.SS "PCKILL"

.LP

If applied to the process control file, a signal is sent to the process with semantics identical to those of \fBkill\fR(2). If applied to an lwp control file, a directed signal is sent to the specific lwp. The signal is named in a \fBlong\fR operand contained in the message. Sending \fBSIGKILL\fR terminates the process immediately.

.SS "PCUNKILL"

.LP

A signal is deleted, that is, it is removed from the set of pending signals. If applied to the process control file, the signal is deleted from the process’s pending signals. If applied to an lwp control file, the signal is deleted from the lwp’s pending signals. The current signal (if any) is unaffected. The signal is named in a \fBlong\fR operand in the control message. It is an error (\fBEINVAL\fR) to attempt to delete \fBSIGKILL\fR.

.SS "PCSHOLD"

.LP

Set the set of held signals for the specific or representative lwp (signals whose delivery will be blocked if sent to the lwp). The set of signals is specified with a \fBsigset_t\fR operand. \fBSIGKILL\fR and \fBSIGSTOP\fR cannot be held; if specified, they are silently ignored.

.SS "PCSFAULT"

.LP

Define a set of hardware faults to be traced in the process. On incurring one of these faults, an lwp stops. The set is defined via the operand

\fBfltset_t\fR structure. Fault names are defined in \fB<sys/fault.h>\fR and include the following. Some of these may not occur on all processors; there may be processor-specific faults in addition to these.

.sp

.ne 2

.na

\fB\fBFLTILL\fR\fR

.ad

.RS 13n

illegal instruction

.RE

.sp

.ne 2

.na

\fB\fBFLTPRIV\fR\fR

.ad

.RS 13n

privileged instruction

.RE

.sp

.ne 2

.na

\fB\fBFLTBPT\fR\fR

.ad

.RS 13n

breakpoint trap

.RE

.sp

.ne 2

.na

\fB\fBFLTTRACE\fR\fR

.ad

.RS 13n

trace trap (single-step)

.RE

.sp

.ne 2

.na

\fB\fBFLTWATCH\fR\fR

.ad

.RS 13n

watchpoint trap

.RE

.sp

.ne 2

.na

\fB\fBFLTACCESS\fR\fR

.ad

.RS 13n

memory access fault (bus error)

.RE

.sp

.ne 2

.na

\fB\fBFLTBOUNDS\fR\fR

.ad

.RS 13n

memory bounds violation

.RE

```
.sp
.ne 2
.na
\fB\fBFLTIOVF\fR\fR
.ad
.RS 13n
integer overflow
.RE

.sp
.ne 2
.na
\fB\fBFLTIZDIV\fR\fR
.ad
.RS 13n
integer zero divide
.RE

.sp
.ne 2
.na
\fB\fBFLTFPE\fR\fR
.ad
.RS 13n
floating-point exception
.RE

.sp
.ne 2
.na
\fB\fBFLTSTACK\fR\fR
.ad
.RS 13n
unrecoverable stack fault
.RE

.sp
.ne 2
.na
\fB\fBFLTPAGE\fR\fR
.ad
.RS 13n
recoverable page fault
.RE

.sp
.LP
When not traced, a fault normally results in the posting of a signal to the lwp
that incurred the fault. If an lwp stops on a fault, the signal is posted to
the lwp when execution is resumed unless the fault is cleared by \fBPCCFAULT\fR
or by the \fBPRCFAULT\fR option of \fBPCRUN\fR. \fBFLTPAGE\fR is an exception;
no signal is posted. The \fBpr_info\fR field in the \fBlwpstatus\fR structure
identifies the signal to be sent and contains machine-specific information
about the fault.
.SS "PCCFAULT"
.LP
The current fault, if any, is cleared; the associated signal will not be sent
to the specific or representative lwp.
.SS "PCSENTRY PCSEXIT"
.LP
These control operations instruct the process's lwps to stop on entry to or
exit from specified system calls. The set of system calls to be traced is
defined via an operand \fBsysset_t\fR structure.
.sp
.LP
When entry to a system call is being traced, an lwp stops after having begun
```



```
the call to the system but before the system call arguments have been fetched
from the lwp. When exit from a system call is being traced, an lwp stops on
completion of the system call just prior to checking for signals and returning
to user level. At this point, all return values have been stored into the lwp's
registers.
.sp
.LP
If an lwp is stopped on entry to a system call (\fBPR_SYSENTRY\fR) or when
sleeping in an interruptible system call (\fBPR_ASLEEP\fR is set), it may be
instructed to go directly to system call exit by specifying the \fBPRSABORT\fR
flag in a \fBPCRUN\fR control message. Unless exit from the system call is
being traced, the lwp returns to user level showing \fBEINTR\fR.
.SS "PCWATCH"
.LP
Set or clear a watched area in the controlled process from a \fBprwatch\fR
structure operand:
.sp
.in +2
.nf
typedef struct prwatch {
    uintptr_t pr_vaddr;  /* virtual address of watched area */
    size_t pr_size;      /* size of watched area in bytes */
    int pr_wflags;       /* watch type flags */
} prwatch_t;
______unchanged_portion_omitted_
.fi
.in -2

.sp
.LP
These operations have the same effect as \fBpread\fR(2) and \fBpwrite\fR(2),
respectively, of the target process's address space file. The difference is
that more than one \fBPCREAD\fR or \fBPCWRITE\fR control operation can be
written to the control file at once, and they can be interspersed with other
control operations in a single write to the control file. This is useful, for
example, when planting many breakpoint instructions in the process's address
space, or when stepping over a breakpointed instruction. Unlike \fBpread\fR(2)
and \fBpwrite\fR(2), no provision is made for partial reads or writes; if the
operation cannot be performed completely, it fails with \fBEIO\fR.
.SS "PCNICE"
.LP
The traced process's \fBnice\fR(2) value is incremented by the amount in the
operand \fBlong\fR. Only a process with the {\fBPRIV_PROC_PRIOCNTL\fR}
privilege asserted in its effective set can better a process's priority in this
way, but any user may lower the priority. This operation is not meaningful for
all scheduling classes.
.SS "PCSCRED"
.LP
Set the target process credentials to the values contained in the
\fBprcred_t\fR structure operand (see \fB/proc/\fR\fIpid\fR\fB/cred\fR). The
effective, real, and saved user-IDs and group-IDs of the target process are
set. The target process's supplementary groups are not changed; the
\fBpr_ngroups\fR and \fBpr_groups\fR members of the structure operand are
ignored. Only the privileged processes can perform this operation; for all
others it fails with \fBEPERM\fR.
.SS "PCSCREDX"
.LP
Operates like \fBPCSCRED\fR but also sets the supplementary groups; the length
of the data written with this control operation should be "sizeof
(\fBprcred_t\fR) + sizeof (\fBgid_t)\fR * (#groups - 1)".
.SS "PCSPRIV"
.LP
Set the target process privilege to the values contained in the \fBprpriv_t\fR
operand (see \fB/proc/pid/priv\fR). The effective, permitted, inheritable, and
limit sets are all changed. Privilege flags can also be set. The process is
made privilege aware unless it can relinquish privilege awareness. See
```

\fBprivileges\fR(5).
.sp
.LP
The limit set of the target process cannot be grown. The other privilege sets must be subsets of the intersection of the effective set of the calling process with the new limit set of the target process or subsets of the original values of the sets in the target process.
.sp
.LP
If any of the above restrictions are not met, \fBEPERM\fR is returned. If the structure written is improperly formatted, \fBEINVAL\fR is returned.
.SH PROGRAMMING NOTES
.LP
For security reasons, except for the \fBpsinfo\fR, \fBusage\fR, \fBlpsinfo\fR, \fBlusage\fR, \fBlwpsinfo\fR, and \fBlwpusage\fR files, which are world-readable, and except for privileged processes, an open of a \fB/proc\fR file fails unless both the user-ID and group-ID of the caller match those of the traced process and the process's object file is readable by the caller. The effective set of the caller is a superset of both the inheritable and the permitted set of the target process. The limit set of the caller is a superset of the limit set of the target process. Except for the world-readable files just mentioned, files corresponding to setuid and setgid processes can be opened only by the appropriately privileged process.
.sp
.LP
A process that is missing the basic privilege {\fBPRIV_PROC_INFO\fR} cannot see any processes under \fB/proc\fR that it cannot send a signal to.
.sp
.LP
A process that has {\fBPRIV_PROC_OWNER\fR} asserted in its effective set can open any file for reading. To manipulate or control a process, the controlling process must have at least as many privileges in its effective set as the target process has in its effective, inheritable, and permitted sets. The limit set of the controlling process must be a superset of the limit set of the target process. Additional restrictions apply if any of the uids of the target process are 0. See \fBprivileges\fR(5).
.sp
.LP
Even if held by a privileged process, an open process or lwp file descriptor (other than file descriptors for the world-readable files) becomes invalid if the traced process performs an \fBexec\fR(2) of a setuid/setgid object file or an object file that the traced process cannot read. Any operation performed on an invalid file descriptor, except \fBclose\fR(2), fails with \fBEAGAIN\fR. In this situation, if any tracing flags are set and the process or any lwp file descriptor is open for writing, the process will have been directed to stop and its run-on-last-close flag will have been set (see \fBPCSET\fR). This enables a controlling process (if it has permission) to reopen the \fB/proc\fR files to get new valid file descriptors, close the invalid file descriptors, unset the run-on-last-close flag (if desired), and proceed. Just closing the invalid file descriptors causes the traced process to resume execution with all tracing flags cleared. Any process not currently open for writing via \fB/proc\fR, but that has left-over tracing flags from a previous open, and that executes a setuid/setgid or unreadable object file, will not be stopped but will have all its tracing flags cleared.
.sp
.LP
To wait for one or more of a set of processes or lwps to stop or terminate, \fB/proc\fR file descriptors (other than those obtained by opening the \fBcwd\fR or \fBroot\fR directories or by opening files in the \fBfd\fR or \fBobject\fR directories) can be used in a \fBpoll\fR(2) system call. When requested and returned, either of the polling events \fBPOLLPRI\fR or \fBPOLLWRNORM\fR indicates that the process or lwp stopped on an event of interest. Although they cannot be requested, the polling events \fBPOLLHUP\fR, \fBPOLLERR\fR, and \fBPOLLNVAL\fR may be returned. \fBPOLLHUP\fR indicates that the process or lwp has terminated. \fBPOLLERR\fR indicates that the file descriptor has become invalid. \fBPOLLNVAL\fR is returned immediately if

\fBPOLLPRI\fR or \fBPOLLWRNORM\fR is requested on a file descriptor referring to a system process (see \fBPCSTOP\fR). The requested events may be empty to wait simply for termination.
.SH FILES
.ne 2
.na
\fB\fB/proc\fR\fR
.ad
.sp .6
.RS 4n
directory (list of processes)
.RE

.sp
.ne 2
.na
\fB\fB/proc/\fIpid\fR\fR\fR
.ad
.sp .6
.RS 4n
specific process directory
.RE

.sp
.ne 2
.na
\fB\fB/proc/self\fR\fR
.ad
.sp .6
.RS 4n
alias for a process's own directory
.RE

.sp
.ne 2
.na
\fB\fB/proc/\fIpid\fR/as\fR\fR
.ad
.sp .6
.RS 4n
address space file
.RE

.sp
.ne 2
.na
\fB\fB/proc/\fIpid\fR/ctl\fR\fR
.ad
.sp .6
.RS 4n
process control file
.RE

.sp
.ne 2
.na
\fB\fB/proc/\fIpid\fR/status\fR\fR
.ad
.sp .6
.RS 4n
process status
.RE

.sp
.ne 2
.na

```
2449 \fB\fB/proc/\fIpid\fR/lstatus\fR\fR
2450 .ad
2451 .sp .6
2452 .RS 4n
2453 array of lwp status structs
2454 .RE

2456 .sp
2457 .ne 2
2458 .na
2459 \fB\fB/proc/\fIpid\fR/psinfo\fR\fR
2460 .ad
2461 .sp .6
2462 .RS 4n
2463 process \fBps\fR(1) info
2464 .RE

2466 .sp
2467 .ne 2
2468 .na
2469 \fB\fB/proc/\fIpid\fR/lpsinfo\fR\fR
2470 .ad
2471 .sp .6
2472 .RS 4n
2473 array of lwp \fBps\fR(1) info structs
2474 .RE

2476 .sp
2477 .ne 2
2478 .na
2479 \fB\fB/proc/\fIpid\fR/map\fR\fR
2480 .ad
2481 .sp .6
2482 .RS 4n
2483 address space map
2484 .RE

2486 .sp
2487 .ne 2
2488 .na
2489 \fB\fB/proc/\fIpid\fR/xmap\fR\fR
2490 .ad
2491 .sp .6
2492 .RS 4n
2493 extended address space map
2494 .RE

2496 .sp
2497 .ne 2
2498 .na
2499 \fB\fB/proc/\fIpid\fR/rmap\fR\fR
2500 .ad
2501 .sp .6
2502 .RS 4n
2503 reserved address map
2504 .RE

2506 .sp
2507 .ne 2
2508 .na
2509 \fB\fB/proc/\fIpid\fR/cred\fR\fR
2510 .ad
2511 .sp .6
2512 .RS 4n
2513 process credentials
2514 .RE
```



```
2516 .sp
2517 .ne 2
2518 .na
2519 \fB\fB/proc/\fIpid\fR/priv\fR\fR
2520 .ad
2521 .sp .6
2522 .RS 4n
2523 process privileges
2524 .RE

2526 .sp
2527 .ne 2
2528 .na
2529 \fB\fB/proc/\fIpid\fR/sigact\fR\fR
2530 .ad
2531 .sp .6
2532 .RS 4n
2533 process signal actions
2534 .RE

2536 .sp
2537 .ne 2
2538 .na
2539 \fB\fB/proc/\fIpid\fR/auxv\fR\fR
2540 .ad
2541 .sp .6
2542 .RS 4n
2543 process aux vector
2544 .RE

2546 .sp
2547 .ne 2
2548 .na
2549 \fB\fB/proc/\fIpid\fR/ldt\fR\fR
2550 .ad
2551 .sp .6
2552 .RS 4n
2553 process \fBLDT\fR (x86 only)
2554 .RE

2556 .sp
2557 .ne 2
2558 .na
2559 \fB\fB/proc/\fIpid\fR/usage\fR\fR
2560 .ad
2561 .sp .6
2562 .RS 4n
2563 process usage
2564 .RE

2566 .sp
2567 .ne 2
2568 .na
2569 \fB\fB/proc/\fIpid\fR/lusage\fR\fR
2570 .ad
2571 .sp .6
2572 .RS 4n
2573 array of lwp usage structs
2574 .RE

2576 .sp
2577 .ne 2
2578 .na
2579 \fB\fB/proc/\fIpid\fR/path\fR\fR
2580 .ad
```

```
2581 .sp .6
2582 .RS 4n
2583 symbolic links to process open files
2584 .RE

2586 .sp
2587 .ne 2
2588 .na
2589 \fB\fB/proc/\fIpid\fR/pagedata\fR\fR
2590 .ad
2591 .sp .6
2592 .RS 4n
2593 process page data
2594 .RE

2596 .sp
2597 .ne 2
2598 .na
2599 \fB\fB/proc/\fIpid\fR/watch\fR\fR
2600 .ad
2601 .sp .6
2602 .RS 4n
2603 active watchpoints
2604 .RE

2606 .sp
2607 .ne 2
2608 .na
2609 \fB\fB/proc/\fIpid\fR/cwd\fR\fR
2610 .ad
2611 .sp .6
2612 .RS 4n
2613 alias for the current working directory
2614 .RE

2616 .sp
2617 .ne 2
2618 .na
2619 \fB\fB/proc/\fIpid\fR/root\fR\fR
2620 .ad
2621 .sp .6
2622 .RS 4n
2623 alias for the root directory
2624 .RE

2626 .sp
2627 .ne 2
2628 .na
2629 \fB\fB/proc/\fIpid\fR/fd\fR\fR
2630 .ad
2631 .sp .6
2632 .RS 4n
2633 directory (list of open files)
2634 .RE

2636 .sp
2637 .ne 2
2638 .na
2639 \fB\fB/proc/\fIpid\fR/fd/*\fR\fR
2640 .ad
2641 .sp .6
2642 .RS 4n
2643 aliases for process's open files
2644 .RE

2646 .sp
```



```
2647 .ne 2
2648 .na
2649 \fB\fB/proc/\fIpid\fR/object\fR\fR
2650 .ad
2651 .sp .6
2652 .RS 4n
2653 directory (list of mapped files)
2654 .RE

2656 .sp
2657 .ne 2
2658 .na
2659 \fB\fB/proc/\fIpid\fR/object/a.out\fR\fR
2660 .ad
2661 .sp .6
2662 .RS 4n
2663 alias for process's executable file
2664 .RE

2666 .sp
2667 .ne 2
2668 .na
2669 \fB\fB/proc/\fIpid\fR/object/*\fR\fR
2670 .ad
2671 .sp .6
2672 .RS 4n
2673 aliases for other mapped files
2674 .RE

2676 .sp
2677 .ne 2
2678 .na
2679 \fB\fB/proc/\fIpid\fR/lwp\fR\fR
2680 .ad
2681 .sp .6
2682 .RS 4n
2683 directory (list of lwps)
2684 .RE

2686 .sp
2687 .ne 2
2688 .na
2689 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR\fR\fR
2690 .ad
2691 .sp .6
2692 .RS 4n
2693 specific lwp directory
2694 .RE

2696 .sp
2697 .ne 2
2698 .na
2699 \fB\fB/proc/\fIpid\fR/lwp/agent\fR\fR
2700 .ad
2701 .sp .6
2702 .RS 4n
2703 alias for the agent lwp directory
2704 .RE

2706 .sp
2707 .ne 2
2708 .na
2709 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/lwpctl\fR\fR
2710 .ad
2711 .sp .6
2712 .RS 4n
```

```
2713 lwp control file
2714 .RE

2716 .sp
2717 .ne 2
2718 .na
2719 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/lwpstatus\fR\fR
2720 .ad
2721 .sp .6
2722 .RS 4n
2723 lwp status
2724 .RE

2726 .sp
2727 .ne 2
2728 .na
2729 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/lwpsinfo\fR\fR
2730 .ad
2731 .sp .6
2732 .RS 4n
2733 lwp \fBps\fR(1) info
2734 .RE

2736 .sp
2737 .ne 2
2738 .na
2739 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/lwpusage\fR\fR
2740 .ad
2741 .sp .6
2742 .RS 4n
2743 lwp usage
2744 .RE

2746 .sp
2747 .ne 2
2748 .na
2749 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/gwindows\fR\fR
2750 .ad
2751 .sp .6
2752 .RS 4n
2753 register windows (SPARC only)
2754 .RE

2756 .sp
2757 .ne 2
2758 .na
2759 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/xregs\fR\fR
2760 .ad
2761 .sp .6
2762 .RS 4n
2763 extra state registers
2764 .RE

2766 .sp
2767 .ne 2
2768 .na
2769 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/asrs\fR\fR
2770 .ad
2771 .sp .6
2772 .RS 4n
2773 ancillary state registers (SPARC V9 only)
2774 .RE

2776 .sp
2777 .ne 2
2778 .na
```



```
2779 \fB\fB/proc/\fIpid\fR/lwp/\fIlwpid\fR/spymaster\fR\fR
2780 .ad
2781 .sp .6
2782 .RS 4n
2783 For an agent LWP, the controlling process
2784 .RE

2786 .SH SEE ALSO
2787 .LP
2788 \fBls\fR(1), \fBps\fR(1), \fBchroot\fR(1M), \fBalarm\fR(2), \fBbrk\fR(2),
2789 \fBchdir\fR(2), \fBchroot\fR(2), \fBclose\fR(2), \fBcreat\fR(2), \fBdup\fR(2),
2790 \fBexec\fR(2), \fBfcntl\fR(2), \fBfork\fR(2), \fBfork1\fR(2), \fBfstat\fR(2),
2791 \fBgetdents\fR(2), \fBgetustack\fR(2), \fBkill\fR(2), \fBlseek\fR(2),
2792 \fBmmap\fR(2), \fBnice\fR(2), \fBopen\fR(2), \fBpoll\fR(2), \fBpread\fR(2),
2793 \fBptrace\fR(3C), \fBpwrite\fR(2), \fBread\fR(2), \fBreadlink\fR(2),
2794 \fBreadv\fR(2), \fBshmget\fR(2), \fBsigaction\fR(2), \fBsigaltstack\fR(2),
2795 \fBvfork\fR(2), \fBwrite\fR(2), \fBwritev\fR(2), \fB_stack_grow\fR(3C),
2796 \fBreaddir\fR(3C), \fBpthread_create\fR(3C), \fBpthread_join\fR(3C),
2797 \fBsiginfo.h\fR(3HEAD), \fBsignal.h\fR(3HEAD), \fBthr_create\fR(3C),
2798 \fBthr_join\fR(3C), \fBtypes32.h\fR(3HEAD), \fBucontext.h\fR(3HEAD),
2799 \fBwait\fR(3C), \fBcontract\fR(4), \fBcore\fR(4), \fBprocess\fR(4),
2800 \fBlfcompile\fR(5), \fBprivileges\fR(5), \fBsecurity-flags\fR(5)
2801 .SH DIAGNOSTICS
2802 .LP
2803 Errors that can occur in addition to the errors normally associated with file
2804 system access:
2805 .sp
2806 .ne 2
2807 .na
2808 \fB\fBE2BIG\fR\fR
2809 .ad
2810 .RS 13n
2811 Data to be returned in a \fBread\fR(2) of the page data file exceeds the size
2812 of the read buffer provided by the caller.
2813 .RE

2815 .sp
2816 .ne 2
2817 .na
2818 \fB\fBEACCES\fR\fR
2819 .ad
2820 .RS 13n
2821 An attempt was made to examine a process that ran under a different uid than
2822 the controlling process and {\fBPRIV_PROC_OWNER\fR} was not asserted in the
2823 effective set.
2824 .RE

2826 .sp
2827 .ne 2
2828 .na
2829 \fB\fBEAGAIN\fR\fR
2830 .ad
2831 .RS 13n
2832 The traced process has performed an \fBexec\fR(2) of a setuid/setgid object
2833 file or of an object file that it cannot read; all further operations on the
2834 process or lwp file descriptor (except \fBclose\fR(2)) elicit this error.
2835 .RE

2837 .sp
2838 .ne 2
2839 .na
2840 \fB\fBEBUSY\fR\fR
2841 .ad
2842 .RS 13n
2843 \fBPCSTOP\fR, \fBPCDSTOP\fR, \fBPCWSTOP\fR, or \fBPCTWSTOP\fR was applied to a
2844 system process; an exclusive \fBopen\fR(2) was attempted on a \fB/proc\fR file
```

for a process already open for writing; \fBPCRUN\fR, \fBPCSREG\fR,
\fBPCSVADDR\fR, \fBPCSFPREG\fR, or \fBPCSXREG\fR was applied to a process or
lwp not stopped on an event of interest; an attempt was made to mount
\fB/proc\fR when it was already mounted; \fBPCAGENT\fR was applied to a process
that was not fully stopped or that already had an agent lwp.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 13n
In general, this means that some invalid argument was supplied to a system
call. A non-exhaustive list of conditions eliciting this error includes: a
control message operation code is undefined; an out-of-range signal number was
specified with \fBPCSSIG\fR, \fBPCKILL\fR, or \fBPCUNKILL\fR; \fBSIGKILL\fR was
specified with \fBPCUNKILL\fR; \fBPCSFPREG\fR was applied on a system that does
not support floating-point operations; \fBPCSXREG\fR was applied on a system
that does not support extra state registers.
.RE

.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 13n
A signal was received by the controlling process while waiting for the traced
process or lwp to stop via \fBPCSTOP\fR, \fBPCWSTOP\fR, or \fBPCTWSTOP\fR.
.RE

.sp
.ne 2
.na
\fB\fBEIO\fR\fR
.ad
.RS 13n
A \fBwrite\fR(2) was attempted at an illegal address in the traced process.
.RE

.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 13n
The traced process or lwp has terminated after being opened. The basic
privilege {\fBPRIV_PROC_INFO\fR} is not asserted in the effective set of the
calling process and the calling process cannot send a signal to the target
process.
.RE

.sp
.ne 2
.na
\fB\fBENOMEM\fR\fR
.ad
.RS 13n
The system-imposed limit on the number of page data file descriptors was
reached on an open of \fB/proc/\fR\fIpid\fR\fB/pagedata\fR; an attempt was made
with \fBPCWATCH\fR to establish more watched areas than the system can support;
the \fBPCAGENT\fR operation was issued when the system was out of resources for
creating lwps.
.RE

.sp
.ne 2
.na
\fB\fBENOSYS\fR\fR
.ad
.RS 13n
An attempt was made to perform an unsupported operation (such as
\fBcreat\fR(2), \fBlink\fR(2), or \fBunlink\fR(2)) on an entry in \fB/proc\fR.
.RE

.sp
.ne 2
.na
\fB\fBEOVERFLOW\fR\fR
.ad
.RS 13n
A 32-bit controlling process attempted to read or write the \fBas\fR file or
attempted to read the \fBmap\fR, \fBrmap\fR, or \fBpagedata\fR file of a 64-bit
target process. A 32-bit controlling process attempted to apply one of the
control operations \fBPCSREG\fR, \fBPCSXREG\fR, \fBPCSVADDR\fR, \fBPCWATCH\fR,
\fBPCAGENT\fR, \fBPCREAD\fR, \fBPCWRITE\fR to a 64-bit target process.
.RE

.sp
.ne 2
.na
\fB\fBEPERM\fR\fR
.ad
.RS 13n
The process that issued the \fBPCSCRED\fR or \fBPCSCREDX\fR operation did not
have the {\fBPRIV_PROC_SETID\fR} privilege asserted in its effective set, or
the process that issued the \fBPCNICE\fR operation did not have the
{\fBPRIV_PROC_PRIOCNTL\fR} in its effective set.
.sp
An attempt was made to control a process of which the E, P, and I privilege
sets were not a subset of the effective set of the controlling process or the
limit set of the controlling process is not a superset of limit set of the
controlled process.
.sp
Any of the uids of the target process are 0 or an attempt was made to change
any of the uids to 0 using PCSCRED and the security policy imposed additional
restrictions. See \fBprivileges\fR(5).
.RE

.SH NOTES
.LP
Descriptions of structures in this document include only interesting structure
elements, not filler and padding fields, and may show elements out of order for
descriptive clarity. The actual structure definitions are contained in
\fB<procfs.h>\fR\&.
.SH BUGS
.LP
Because the old \fBioctl\fR(2)-based version of \fB/proc\fR is currently
supported for binary compatibility with old applications, the top-level
directory for a process, \fB/proc/\fR\fIpid\fR, is not world-readable, but it
is world-searchable. Thus, anyone can open \fB/proc/\fR\fIpid\fR\fB/psinfo\fR
even though \fBls\fR(1) applied to \fB/proc/\fR\fIpid\fR will fail for anyone
but the owner or an appropriately privileged process. Support for the old
\fBioctl\fR(2)-based version of \fB/proc\fR will be dropped in a future
release, at which time the top-level directory for a process will be made
world-readable.
.sp
.LP
On SPARC based machines, the types \fBgregset_t\fR and \fBfpregset_t\fR defined
in <\fBsys/regset.h\fR> are similar to but not the same as the types
\fBprgregset_t\fR and \fBprfpregset_t\fR defined in <\fBprocfs.h\fR>.

```
**********************************************************
  126087 Mon Oct 15 13:28:39 2018
new/usr/src/pkg/manifests/system-dtrace-tests.mf
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012, 2016 by Delphix. All rights reserved.
# Copyright 2018 Joyent, Inc.
#

set name=pkg.fmri value=pkg:/system/dtrace/tests@$(PKGVERS)
set name=pkg.description value="DTrace Test Suite Internal Distribution"
set name=pkg.summary value="DTrace Test Suite"
set name=info.classification \
    value=org.opensolaris.category.2008:Development/System
set name=variant.arch value=$(ARCH)
dir path=opt/SUNWdtrt group=sys
dir path=opt/SUNWdtrt/bin
dir path=opt/SUNWdtrt/bin/$(ARCH32)
dir path=opt/SUNWdtrt/bin/$(ARCH64)
dir path=opt/SUNWdtrt/lib
dir path=opt/SUNWdtrt/lib/java
dir path=opt/SUNWdtrt/tst
dir path=opt/SUNWdtrt/tst/$(ARCH)
dir path=opt/SUNWdtrt/tst/$(ARCH)/arrays
$(i386_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/error
$(i386_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/funcs
dir path=opt/SUNWdtrt/tst/$(ARCH)/pid
$(sparc_ONLY)dir path=opt/SUNWdtrt/tst/$(ARCH)/usdt
dir path=opt/SUNWdtrt/tst/$(ARCH)/ustack
dir path=opt/SUNWdtrt/tst/common
dir path=opt/SUNWdtrt/tst/common/aggs
dir path=opt/SUNWdtrt/tst/common/arithmetic
dir path=opt/SUNWdtrt/tst/common/arrays
dir path=opt/SUNWdtrt/tst/common/assocs
dir path=opt/SUNWdtrt/tst/common/begin
dir path=opt/SUNWdtrt/tst/common/bitfields
dir path=opt/SUNWdtrt/tst/common/buffering
dir path=opt/SUNWdtrt/tst/common/builtinvar
dir path=opt/SUNWdtrt/tst/common/cg
dir path=opt/SUNWdtrt/tst/common/clauses
dir path=opt/SUNWdtrt/tst/common/cpc
dir path=opt/SUNWdtrt/tst/common/decls
```



```
dir path=opt/SUNWdtrt/tst/common/drops
dir path=opt/SUNWdtrt/tst/common/dtraceUtil
dir path=opt/SUNWdtrt/tst/common/end
dir path=opt/SUNWdtrt/tst/common/enum
dir path=opt/SUNWdtrt/tst/common/env
dir path=opt/SUNWdtrt/tst/common/error
dir path=opt/SUNWdtrt/tst/common/exit
dir path=opt/SUNWdtrt/tst/common/fbtprovider
dir path=opt/SUNWdtrt/tst/common/funcs
dir path=opt/SUNWdtrt/tst/common/grammar
dir path=opt/SUNWdtrt/tst/common/include
dir path=opt/SUNWdtrt/tst/common/inline
dir path=opt/SUNWdtrt/tst/common/io
dir path=opt/SUNWdtrt/tst/common/ip
dir path=opt/SUNWdtrt/tst/common/java_api
dir path=opt/SUNWdtrt/tst/common/json
dir path=opt/SUNWdtrt/tst/common/lexer
dir path=opt/SUNWdtrt/tst/common/llquantize
dir path=opt/SUNWdtrt/tst/common/mdb
dir path=opt/SUNWdtrt/tst/common/mib
dir path=opt/SUNWdtrt/tst/common/misc
dir path=opt/SUNWdtrt/tst/common/multiaggs
dir path=opt/SUNWdtrt/tst/common/nfs
dir path=opt/SUNWdtrt/tst/common/offsetof
dir path=opt/SUNWdtrt/tst/common/operators
dir path=opt/SUNWdtrt/tst/common/pid
dir path=opt/SUNWdtrt/tst/common/plockstat
dir path=opt/SUNWdtrt/tst/common/pointers
dir path=opt/SUNWdtrt/tst/common/pragma
dir path=opt/SUNWdtrt/tst/common/predicates
dir path=opt/SUNWdtrt/tst/common/preprocessor
dir path=opt/SUNWdtrt/tst/common/print
dir path=opt/SUNWdtrt/tst/common/printa
dir path=opt/SUNWdtrt/tst/common/printf
dir path=opt/SUNWdtrt/tst/common/privs
dir path=opt/SUNWdtrt/tst/common/probes
dir path=opt/SUNWdtrt/tst/common/proc
dir path=opt/SUNWdtrt/tst/common/profile-n
dir path=opt/SUNWdtrt/tst/common/providers
dir path=opt/SUNWdtrt/tst/common/raise
dir path=opt/SUNWdtrt/tst/common/rates
dir path=opt/SUNWdtrt/tst/common/safety
dir path=opt/SUNWdtrt/tst/common/scalars
dir path=opt/SUNWdtrt/tst/common/sched
dir path=opt/SUNWdtrt/tst/common/scripting
dir path=opt/SUNWdtrt/tst/common/sdt
dir path=opt/SUNWdtrt/tst/common/sizeof
dir path=opt/SUNWdtrt/tst/common/speculation
dir path=opt/SUNWdtrt/tst/common/stability
dir path=opt/SUNWdtrt/tst/common/stack
dir path=opt/SUNWdtrt/tst/common/stackdepth
dir path=opt/SUNWdtrt/tst/common/stop
dir path=opt/SUNWdtrt/tst/common/strlen
dir path=opt/SUNWdtrt/tst/common/strtoll
dir path=opt/SUNWdtrt/tst/common/struct
dir path=opt/SUNWdtrt/tst/common/sugar
dir path=opt/SUNWdtrt/tst/common/syscall
dir path=opt/SUNWdtrt/tst/common/sysevent
dir path=opt/SUNWdtrt/tst/common/threadname
dir path=opt/SUNWdtrt/tst/common/tick-n
dir path=opt/SUNWdtrt/tst/common/trace
dir path=opt/SUNWdtrt/tst/common/tracemem
dir path=opt/SUNWdtrt/tst/common/translators
dir path=opt/SUNWdtrt/tst/common/typedef
dir path=opt/SUNWdtrt/tst/common/types
dir path=opt/SUNWdtrt/tst/common/uctf
```

```
dir path=opt/SUNWdtrt/tst/common/union
dir path=opt/SUNWdtrt/tst/common/usdt
dir path=opt/SUNWdtrt/tst/common/ustack
dir path=opt/SUNWdtrt/tst/common/vars
dir path=opt/SUNWdtrt/tst/common/version
$(i386_ONLY)dir path=opt/SUNWdtrt/tst/i86xpv
$(i386_ONLY)dir path=opt/SUNWdtrt/tst/i86xpv/xdt
file path=opt/SUNWdtrt/README mode=0444
file path=opt/SUNWdtrt/bin/$(ARCH32)/chkargs mode=0555
file path=opt/SUNWdtrt/bin/$(ARCH64)/chkargs mode=0555
file path=opt/SUNWdtrt/bin/baddof mode=0555
file path=opt/SUNWdtrt/bin/badioctl mode=0555
file path=opt/SUNWdtrt/bin/chkargs mode=0555
file path=opt/SUNWdtrt/bin/dstyle mode=0555
file path=opt/SUNWdtrt/bin/dtest mode=0555
file path=opt/SUNWdtrt/bin/dtfailures mode=0555
file path=opt/SUNWdtrt/bin/exception.lst mode=0444
file path=opt/SUNWdtrt/bin/jdtrace mode=0555
file path=opt/SUNWdtrt/lib/java/jdtrace.jar
file path=opt/SUNWdtrt/tst/$(ARCH)/arrays/tst.uregsarray.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/error/tst.DTRACEFLT_DIVZERO.d \
    mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyin.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyinstr.d \
    mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyout.d \
    mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/funcs/tst.badcopyoutstr.d \
    mode=0444
$(sparc_ONLY)file \
    path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.D_PROC_ALIGN.misaligned.d mode=0444
$(sparc_ONLY)file \
    path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.D_PROC_ALIGN.misaligned.exe \
    mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite1.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite1.exe \
    mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite2.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/err.uregswrite2.exe \
    mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.badinstr.exe mode=0555
$(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d mode=0444
$(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.d.out mode=0444
$(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.br.exe mode=0555
file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.d mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.branch.exe mode=0555
file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.d mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.embedded.exe mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.ret.exe mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.exe mode=0555
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.retlist.ksh mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.uregswrite.d mode=0444
$(i386_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/pid/tst.uregswrite.exe \
    mode=0555
$(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/usdt/tst.tailcall.ksh \
    mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.d.out mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.annotated.exe mode=0555
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.d mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.circstack.exe mode=0555
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.d.out mode=0444
file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.helper.exe mode=0555
```



```
$(sparc_ONLY)file path=opt/SUNWdtrt/tst/$(ARCH)/ustack/tst.trapstat.ksh \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_FUNC.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_MDIM.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_NULL.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_REDEF.redef.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.avgtoofew.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.maxnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.mintoofew.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.quantizetoofew.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.stddevtoofew.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_AGG_SCALAR.sumtoofew.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_AGGARG.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_CLEAR_PROTO.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_IDENT.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_FUNC_UNDEF.badaggfunc.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badexpr.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.badkey3.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_IDENT_UNDEF.noeffect.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey1.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey2.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_KEY_TYPE.badkey4.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqbad1.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASETYPE.lqshort.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_BASEVAL.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMTYPE.lqbad1.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_LIMVAL.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHBASE.order.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHLIM.order.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MATCHSTEP.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_MISMATCH.lqbadarg.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPLARGE.lqtoofew.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPSMALL.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPTYPE.lqbadinc.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_LQUANT_STEPVAL.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_AGGARG.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_PROTO.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_NORMALIZE_SCALAR.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_ARG.lquantizetoofew.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.avgtoomany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.counttoomany.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizenoarg.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.lquantizetoomany.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.maxtoomany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.minnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.mintoomany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizenoarg.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.quantizetoomany.d \
    mode=0444
```

```
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.stddevnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.stddevtoomany.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.sumnoarg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_PROTO_LEN.sumtoomany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_AGGARG.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_PROTO.badmany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_PROTO.badnone.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/err.D_TRUNC_SCALAR.bad.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggencoding.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggencoding.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.agghist.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.agghist.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpack.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpack.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackbanner.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackbanner.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackzoom.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggpackzoom.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggzoom.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.aggzoom.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.allquant.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.allquant.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.avg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.avg.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.avg_neg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.avg_neg.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clear.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clear.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg2.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearavg2.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.cleardenormalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.cleardenormalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearlquantize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearlquantize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearnormalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearnormalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearstddev.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.clearstddev.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.count.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.count.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.count2.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.count2.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.count3.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalizeonly.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.denormalizeonly.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.fmtnormalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.fmtnormalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.forms.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.forms.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.goodkey.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.keysort.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.keysort.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantnormal.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantnormal.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantrange.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantrange.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantround.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantround.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantzero.d mode=0444
```



```
file path=opt/SUNWdtrt/tst/common/aggs/tst.lquantzero.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.max.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.max.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.max_neg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.max_neg.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.min.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.min.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.min_neg.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.min_neg.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs1.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs2.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs2.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs3.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multiaggs3.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multinormalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.multinormalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.neglquant.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.neglquant.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negorder.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negorder.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negquant.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negtrunc.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negtrunc.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negtruncquant.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.negtruncquant.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.normalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.normalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.order.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.order.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantmany.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantmany.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantround.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantround.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantzero.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.quantzero.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.signature.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeys.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeys.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeyspos.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.signedkeyspos.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.sizedkeys.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.sizedkeys.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.normalize.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.stddev.normalize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.subr.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.sum.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.sum.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc0.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.trunc0.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.truncquant.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.truncquant.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.valsortkeypos.d mode=0444
file path=opt/SUNWdtrt/tst/common/aggs/tst.valsortkeypos.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0.d mode=0444
file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0_1.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.divby0_2.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_DIV_ZERO.modby0.d mode=0444
```

```
391 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.addmin.d mode=0444
392 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.divmin.d mode=0444
393 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muladd.d mode=0444
394 file path=opt/SUNWdtrt/tst/common/arithmetic/err.D_SYNTAX.muldiv.d mode=0444
395 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d mode=0444
396 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.basics.d.out mode=0444
397 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d mode=0444
398 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compcast.d.out mode=0444
399 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compnarrowassign.d mode=0444
400 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.compnarrowassign.d.out \
401     mode=0444
402 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d mode=0444
403 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.execcast.d.out mode=0444
404 file path=opt/SUNWdtrt/tst/common/arithmetic/tst.nonarrow.ksh mode=0444
405 file path=opt/SUNWdtrt/tst/common/arrays/err.D_ARR_BADREF.bad.d mode=0444
406 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRBIG.toobig.d mode=0444
407 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRNULL.bad.d mode=0444
408 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_ARRSUB.bad.d mode=0444
409 file path=opt/SUNWdtrt/tst/common/arrays/err.D_DECL_PROTO_TYPE.badtuple.d \
410     mode=0444
411 file path=opt/SUNWdtrt/tst/common/arrays/err.D_IDENT_UNDEF.badureg.d mode=0444
412 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic1.d mode=0444
413 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic2.d mode=0444
414 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic3.d mode=0444
415 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic4.d mode=0444
416 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic5.d mode=0444
417 file path=opt/SUNWdtrt/tst/common/arrays/tst.basic6.d mode=0444
418 file path=opt/SUNWdtrt/tst/common/arrays/tst.uregsarray.d mode=0444
419 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupgtype.d \
420     mode=0444
421 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.dupttype.d \
422     mode=0444
423 file path=opt/SUNWdtrt/tst/common/assocs/err.D_OP_INCOMPAT.this.d mode=0444
424 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_ARG.badsig.d mode=0444
425 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toofew.d mode=0444
426 file path=opt/SUNWdtrt/tst/common/assocs/err.D_PROTO_LEN.toomany.d mode=0444
427 file path=opt/SUNWdtrt/tst/common/assocs/err.D_SYNTAX.errassign.d mode=0444
428 file path=opt/SUNWdtrt/tst/common/assocs/err.tupoflow.d mode=0444
429 file path=opt/SUNWdtrt/tst/common/assocs/tst.cpyarray.d mode=0444
430 file path=opt/SUNWdtrt/tst/common/assocs/tst.diffprofile.d mode=0444
431 file path=opt/SUNWdtrt/tst/common/assocs/tst.initialize.d mode=0444
432 file path=opt/SUNWdtrt/tst/common/assocs/tst.invalidref.d mode=0444
433 file path=opt/SUNWdtrt/tst/common/assocs/tst.misc.d mode=0444
434 file path=opt/SUNWdtrt/tst/common/assocs/tst.orthogonality.d mode=0444
435 file path=opt/SUNWdtrt/tst/common/assocs/tst.this.d mode=0444
436 file path=opt/SUNWdtrt/tst/common/assocs/tst.valassign.d.out mode=0444
437 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.begin.d mode=0444
438 file path=opt/SUNWdtrt/tst/common/begin/err.D_PDESC_ZERO.tick.d mode=0444
439 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d mode=0444
440 file path=opt/SUNWdtrt/tst/common/begin/tst.begin.d.out mode=0444
441 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d mode=0444
442 file path=opt/SUNWdtrt/tst/common/begin/tst.multibegin.d.out mode=0444
443 file \
444     path=opt/SUNWdtrt/tst/common/bitfields/err.D_ADDROF_BITFIELD.BitfieldAddress
445     mode=0444
446 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.NegBitField.d \
447     mode=0444
448 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFCONST.ZeroBitField.d \
449     mode=0444
450 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.ExceedBaseType.d \
451     mode=0444
452 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFSIZE.GreaterThan64.d \
453     mode=0444
454 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_DECL_BFTYPE.badtype.d \
455     mode=0444
456 file path=opt/SUNWdtrt/tst/common/bitfields/err.D_OFFSETOF_BITFIELD.d \
```



```
457     mode=0444
458 file \
459     path=opt/SUNWdtrt/tst/common/bitfields/err.D_SIZEOF_BITFIELD.SizeofBitfield.
460     mode=0444
461 file path=opt/SUNWdtrt/tst/common/bitfields/tst.BitFieldPromotion.d mode=0444
462 file path=opt/SUNWdtrt/tst/common/bitfields/tst.SizeofBitField.d mode=0444
463 file path=opt/SUNWdtrt/tst/common/buffering/err.end.d mode=0444
464 file path=opt/SUNWdtrt/tst/common/buffering/err.resize1.d mode=0444
465 file path=opt/SUNWdtrt/tst/common/buffering/err.resize2.d mode=0444
466 file path=opt/SUNWdtrt/tst/common/buffering/err.resize3.d mode=0444
467 file path=opt/SUNWdtrt/tst/common/buffering/err.zerobuf.d mode=0444
468 file path=opt/SUNWdtrt/tst/common/buffering/tst.alignring.d mode=0444
469 file path=opt/SUNWdtrt/tst/common/buffering/tst.cputime.ksh mode=0444
470 file path=opt/SUNWdtrt/tst/common/buffering/tst.dynvarsize.d mode=0444
471 file path=opt/SUNWdtrt/tst/common/buffering/tst.fill1.d mode=0444
472 file path=opt/SUNWdtrt/tst/common/buffering/tst.fill1.d.out mode=0444
473 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize1.d mode=0444
474 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize2.d mode=0444
475 file path=opt/SUNWdtrt/tst/common/buffering/tst.resize3.d mode=0444
476 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring1.d mode=0444
477 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring2.d mode=0444
478 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring2.d.out mode=0444
479 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring3.d mode=0444
480 file path=opt/SUNWdtrt/tst/common/buffering/tst.ring3.d.out mode=0444
481 file path=opt/SUNWdtrt/tst/common/buffering/tst.smallring.d mode=0444
482 file path=opt/SUNWdtrt/tst/common/buffering/tst.switch1.d mode=0444
483 file path=opt/SUNWdtrt/tst/common/buffering/tst.switch1.d.out mode=0444
484 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.cpuusage.d \
485     mode=0444
486 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.nice.d \
487     mode=0444
488 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.priority.d \
489     mode=0444
490 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.prsize.d \
491     mode=0444
492 file path=opt/SUNWdtrt/tst/common/builtinvar/err.D_XLATE_NOCONV.rssize.d \
493     mode=0444
494 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg0.d mode=0444
495 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg0clause.d mode=0444
496 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1.d mode=0444
497 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1to8.d mode=0444
498 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.arg1to8clause.d mode=0444
499 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.caller.d mode=0444
500 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.caller1.d mode=0444
501 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.epid.d mode=0444
502 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.epid1.d mode=0444
503 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.errno.d mode=0444
504 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.errno1.d mode=0444
505 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.execname.d mode=0444
506 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.hpriority.d mode=0444
507 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.id.d mode=0444
508 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.id1.d mode=0444
509 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.ipl.d mode=0444
510 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.ipl1.d mode=0444
511 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.lwpsinfo.d mode=0444
512 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.lwpsinfo1.d mode=0444
513 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.pid.d mode=0444
514 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.pid1.d mode=0444
515 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.psinfo.d mode=0444
516 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.psinfo1.d mode=0444
517 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.tid.d mode=0444
518 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.tid1.d mode=0444
519 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.timestamp.d mode=0444
520 file path=opt/SUNWdtrt/tst/common/builtinvar/tst.vtimestamp.d mode=0444
521 file path=opt/SUNWdtrt/tst/common/cg/err.D_NOREG.noreg.d mode=0444
522 file path=opt/SUNWdtrt/tst/common/cg/err.baddif.d mode=0444
```

```
523 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.aggfun.d mode=0444
524 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.aggtup.d mode=0444
525 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.arrtup.d mode=0444
526 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.body.d mode=0444
527 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.both.d mode=0444
528 file path=opt/SUNWdtrt/tst/common/clauses/err.D_IDENT_UNDEF.pred.d mode=0444
529 file path=opt/SUNWdtrt/tst/common/clauses/tst.nopred.d mode=0444
530 file path=opt/SUNWdtrt/tst/common/clauses/tst.pred.d mode=0444
531 file path=opt/SUNWdtrt/tst/common/clauses/tst.predfirst.d mode=0444
532 file path=opt/SUNWdtrt/tst/common/clauses/tst.predlast.d mode=0444
533 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.lowfrequency.d \
534     mode=0444
535 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.malformedoverflow.d \
536     mode=0444
537 file path=opt/SUNWdtrt/tst/common/cpc/err.D_PDESC_ZERO.nonexistentevent.d \
538     mode=0444
539 file path=opt/SUNWdtrt/tst/common/cpc/err.cpcvscpustatpart1.ksh mode=0444
540 file path=opt/SUNWdtrt/tst/common/cpc/err.cpcvscpustatpart2.ksh mode=0444
541 file path=opt/SUNWdtrt/tst/common/cpc/err.cputrackfailtostart.ksh mode=0444
542 file path=opt/SUNWdtrt/tst/common/cpc/err.cputrackterminates.ksh mode=0444
543 file path=opt/SUNWdtrt/tst/common/cpc/err.toomanyenablings.d mode=0444
544 file path=opt/SUNWdtrt/tst/common/cpc/tst.allcpus.ksh mode=0444
545 file path=opt/SUNWdtrt/tst/common/cpc/tst.genericevent.d mode=0444
546 file path=opt/SUNWdtrt/tst/common/cpc/tst.platformevent.ksh mode=0444
547 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_LOCASSC.NonLocalAssoc.d \
548     mode=0444
549 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_LONGINT.LongStruct.d \
550     mode=0444
551 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PARMCLASS.BadStorageClass.d \
552     mode=0444
553 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_NAME.VoidName.d \
554     mode=0444
555 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_TYPE.Dyn.d mode=0444
556 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_VARARGS.VarLenArgs.d \
557     mode=0444
558 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_PROTO_VOID.NonSoleVoid.d \
559     mode=0444
560 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_SIGNINT.UnsignedStruct.d \
561     mode=0444
562 file path=opt/SUNWdtrt/tst/common/decls/err.D_DECL_VOIDATTR.ShortVoidDecl.d \
563     mode=0444
564 file path=opt/SUNWdtrt/tst/common/decls/tst.arrays.d mode=0444
565 file path=opt/SUNWdtrt/tst/common/decls/tst.basics.d mode=0444
566 file path=opt/SUNWdtrt/tst/common/decls/tst.funcs.d mode=0444
567 file path=opt/SUNWdtrt/tst/common/decls/tst.pointers.d mode=0444
568 file path=opt/SUNWdtrt/tst/common/decls/tst.varargsfuncs.d mode=0444
569 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_AGGREGATION.d mode=0444
570 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_DBLERROR.d mode=0444
571 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_DYNAMIC.d mode=0444
572 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_PRINCIPAL.d mode=0444
573 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_PRINCIPAL.end.d \
574     mode=0444
575 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_SPEC.d mode=0444
576 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_SPECUNAVAIL.d mode=0444
577 file path=opt/SUNWdtrt/tst/common/drops/drp.DTRACEDROP_STKSTROVERFLOW.d \
578     mode=0444
579 file \
580     path=opt/SUNWdtrt/tst/common/dtraceUtil/err.D_PDESC_ZERO.InvalidDescription1
581     mode=0444
582 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.AddSearchPath.d.ksh mode=0444
583 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeGiga.d.ksh mode=0444
584 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeKilo.d.ksh mode=0444
585 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeMega.d.ksh mode=0444
586 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.BufsizeTera.d.ksh mode=0444
587 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DataModel32.d.ksh mode=0444
588 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DataModel64.d.ksh mode=0444
```



```
589 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh \
590     mode=0444
591 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DefineNameWithCPP.d.ksh.out \
592     mode=0444
593 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh \
594     mode=0444
595 file \
596     path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithFunction.d.ksh.out \
597     mode=0444
598 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh \
599     mode=0444
600 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithID.d.ksh.out \
601     mode=0444
602 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh \
603     mode=0444
604 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithModule.d.ksh.out \
605     mode=0444
606 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh \
607     mode=0444
608 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithName.d.ksh.out \
609     mode=0444
610 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh \
611     mode=0444
612 file \
613     path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithProvider.d.ksh.out \
614     mode=0444
615 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.DestructWithoutW.d.ksh \
616     mode=0444
617 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationOut.d.ksh \
618     mode=0444
619 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ELFGenerationWithO.d.ksh \
620     mode=0444
621 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus1.d.ksh mode=0444
622 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExitStatus2.d.ksh mode=0444
623 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ExtraneousProbeIds.d.ksh \
624     mode=0444
625 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName1.d.ksh \
626     mode=0444
627 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidFuncName2.d.ksh \
628     mode=0444
629 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId1.d.ksh mode=0444
630 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId2.d.ksh mode=0444
631 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidId3.d.ksh mode=0444
632 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule1.d.ksh \
633     mode=0444
634 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule2.d.ksh \
635     mode=0444
636 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule3.d.ksh \
637     mode=0444
638 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidModule4.d.ksh \
639     mode=0444
640 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProbeIdentifier.d.ksh \
641     mode=0444
642 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider1.d.ksh \
643     mode=0444
644 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider2.d.ksh \
645     mode=0444
646 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider3.d.ksh \
647     mode=0444
648 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidProvider4.d.ksh \
649     mode=0444
650 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc1.d.ksh \
651     mode=0444
652 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc2.d.ksh \
653     mode=0444
654 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc3.d.ksh \
```

```
655     mode=0444
656 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc4.d.ksh \
657     mode=0444
658 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc5.d.ksh \
659     mode=0444
660 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc6.d.ksh \
661     mode=0444
662 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc7.d.ksh \
663     mode=0444
664 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc8.d.ksh \
665     mode=0444
666 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceFunc9.d.ksh \
667     mode=0444
668 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID1.d.ksh \
669     mode=0444
670 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID2.d.ksh \
671     mode=0444
672 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID3.d.ksh \
673     mode=0444
674 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID4.d.ksh \
675     mode=0444
676 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID5.d.ksh \
677     mode=0444
678 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID6.d.ksh \
679     mode=0444
680 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceID7.d.ksh \
681     mode=0444
682 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule1.d.ksh \
683     mode=0444
684 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule2.d.ksh \
685     mode=0444
686 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule3.d.ksh \
687     mode=0444
688 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule4.d.ksh \
689     mode=0444
690 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule5.d.ksh \
691     mode=0444
692 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule6.d.ksh \
693     mode=0444
694 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule7.d.ksh \
695     mode=0444
696 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceModule8.d.ksh \
697     mode=0444
698 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName1.d.ksh \
699     mode=0444
700 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName2.d.ksh \
701     mode=0444
702 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName3.d.ksh \
703     mode=0444
704 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName4.d.ksh \
705     mode=0444
706 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName5.d.ksh \
707     mode=0444
708 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName6.d.ksh \
709     mode=0444
710 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName7.d.ksh \
711     mode=0444
712 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName8.d.ksh \
713     mode=0444
714 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceName9.d.ksh \
715     mode=0444
716 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider1.d.ksh \
717     mode=0444
718 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider2.d.ksh \
719     mode=0444
720 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider3.d.ksh \
```



```
721     mode=0444
722 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider4.d.ksh \
723     mode=0444
724 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.InvalidTraceProvider5.d.ksh \
725     mode=0444
726 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.MultipleInvalidProbeId.d.ksh \
727     mode=0444
728 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.PreprocessorStatement.d.ksh \
729     mode=0444
730 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh mode=0444
731 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.QuietMode.d.ksh.out mode=0444
732 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh mode=0444
733 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.TestCompile.d.ksh.out \
734     mode=0444
735 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.UnDefineNameWithCPP.d.ksh \
736     mode=0444
737 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh \
738     mode=0444
739 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroFunctionProbes.d.ksh.out \
740     mode=0444
741 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh \
742     mode=0444
743 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroModuleProbes.d.ksh.out \
744     mode=0444
745 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh \
746     mode=0444
747 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroNameProbes.d.ksh.out \
748     mode=0444
749 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbeIdentfier.d.ksh \
750     mode=0444
751 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProbesWithoutZ.d.ksh \
752     mode=0444
753 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh \
754     mode=0444
755 file path=opt/SUNWdtrt/tst/common/dtraceUtil/tst.ZeroProviderProbes.d.ksh.out \
756     mode=0444
757 file path=opt/SUNWdtrt/tst/common/end/err.D_IDENT_UNDEF.timespent.d mode=0444
758 file path=opt/SUNWdtrt/tst/common/end/tst.end.d mode=0444
759 file path=opt/SUNWdtrt/tst/common/end/tst.endwithoutbegin.d mode=0444
760 file path=opt/SUNWdtrt/tst/common/end/tst.multibeginend.d mode=0444
761 file path=opt/SUNWdtrt/tst/common/end/tst.multiend.d mode=0444
762 file path=opt/SUNWdtrt/tst/common/enum/err.D_DECL_IDRED.EnumSameName.d \
763     mode=0444
764 file path=opt/SUNWdtrt/tst/common/enum/err.D_UNKNOWN.RepeatIdentifiers.d \
765     mode=0444
766 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumEquality.d mode=0444
767 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumSameValue.d mode=0444
768 file path=opt/SUNWdtrt/tst/common/enum/tst.EnumValAssign.d mode=0444
769 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.setfromscript.d \
770     mode=0444
771 file path=opt/SUNWdtrt/tst/common/env/err.D_PRAGMA_OPTSET.unsetfromscript.d \
772     mode=0444
773 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh mode=0444
774 file path=opt/SUNWdtrt/tst/common/env/tst.ld_nolazyload.ksh.out mode=0444
775 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh mode=0444
776 file path=opt/SUNWdtrt/tst/common/env/tst.setenv1.ksh.out mode=0444
777 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh mode=0444
778 file path=opt/SUNWdtrt/tst/common/env/tst.setenv2.ksh.out mode=0444
779 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh mode=0444
780 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv1.ksh.out mode=0444
781 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh mode=0444
782 file path=opt/SUNWdtrt/tst/common/env/tst.unsetenv2.ksh.out mode=0444
783 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_BADADDR.d mode=0444
784 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_DIVZERO.d mode=0444
785 file path=opt/SUNWdtrt/tst/common/error/tst.DTRACEFLT_UNKNOWN.d mode=0444
786 file path=opt/SUNWdtrt/tst/common/error/tst.error.d mode=0444
```

```
787 file path=opt/SUNWdtrt/tst/common/error/tst.errorend.d mode=0444
788 file path=opt/SUNWdtrt/tst/common/exit/err.D_PROTO_LEN.noarg.d mode=0444
789 file path=opt/SUNWdtrt/tst/common/exit/err.exitarg1.d mode=0444
790 file path=opt/SUNWdtrt/tst/common/exit/tst.basic1.d mode=0444
791 file path=opt/SUNWdtrt/tst/common/fbtprovider/err.D_PDESC_ZERO.notreturn.d \
792     mode=0444
793 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.basic.d mode=0444
794 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionentry.d mode=0444
795 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.functionreturnvalue.d \
796     mode=0444
797 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.ioctlargs.d mode=0444
798 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offset.d mode=0444
799 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.offsetzero.d mode=0444
800 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return.d mode=0444
801 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.return0.d mode=0444
802 file path=opt/SUNWdtrt/tst/common/fbtprovider/tst.tailcall.d mode=0444
803 file path=opt/SUNWdtrt/tst/common/funcs/err.D_FUNC_UNDEF.progenyofbad1.d \
804     mode=0444
805 file path=opt/SUNWdtrt/tst/common/funcs/err.D_OP_VFPTR.badop.d mode=0444
806 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.chillbadarg.d \
807     mode=0444
808 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.copyoutbadarg.d \
809     mode=0444
810 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.mobadarg.d mode=0444
811 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.raisebadarg.d \
812     mode=0444
813 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.tolower.d mode=0444
814 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_ARG.toupper.d mode=0444
815 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.allocanoarg.d \
816     mode=0444
817 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.badbreakpoint.d \
818     mode=0444
819 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoofew.d \
820     mode=0444
821 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.chilltoomany.d \
822     mode=0444
823 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrbadarg.d \
824     mode=0444
825 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyoutstrtoofew.d \
826     mode=0444
827 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoofew.d \
828     mode=0444
829 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.copyouttoomany.d \
830     mode=0444
831 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoofew.d mode=0444
832 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.motoomany.d mode=0444
833 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtabadarg.d mode=0444
834 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoofew.d mode=0444
835 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.mtatoomany.d mode=0444
836 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.panicbadarg.d \
837     mode=0444
838 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.progenyofbad2.d \
839     mode=0444
840 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.stopbadarg.d mode=0444
841 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolower.d mode=0444
842 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.tolowertoomany.d \
843     mode=0444
844 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.toupper.d mode=0444
845 file path=opt/SUNWdtrt/tst/common/funcs/err.D_PROTO_LEN.touppertoomany.d \
846     mode=0444
847 file path=opt/SUNWdtrt/tst/common/funcs/err.D_STRINGOF_TYPE.badstringof.d \
848     mode=0444
849 file path=opt/SUNWdtrt/tst/common/funcs/err.D_VAR_UNDEF.badvar.d mode=0444
850 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca.d mode=0444
851 file path=opt/SUNWdtrt/tst/common/funcs/err.badalloca2.d mode=0444
852 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy.d mode=0444
```



```
853 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy1.d mode=0444
854 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy2.d mode=0444
855 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy3.d mode=0444
856 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy4.d mode=0444
857 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy5.d mode=0444
858 file path=opt/SUNWdtrt/tst/common/funcs/err.badbcopy6.d mode=0444
859 file path=opt/SUNWdtrt/tst/common/funcs/err.badchill.d mode=0444
860 file path=opt/SUNWdtrt/tst/common/funcs/err.chillbadarg.ksh mode=0444
861 file path=opt/SUNWdtrt/tst/common/funcs/err.copyout.d mode=0444
862 file path=opt/SUNWdtrt/tst/common/funcs/err.copyoutbadaddr.ksh mode=0444
863 file path=opt/SUNWdtrt/tst/common/funcs/err.copyoutstrbadaddr.ksh mode=0444
864 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoa6badaddr.d mode=0444
865 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntoabadaddr.d mode=0444
866 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadaddr.d mode=0444
867 file path=opt/SUNWdtrt/tst/common/funcs/err.inet_ntopbadarg.d mode=0444
868 file path=opt/SUNWdtrt/tst/common/funcs/tst.badfreopen.ksh mode=0444
869 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d mode=0444
870 file path=opt/SUNWdtrt/tst/common/funcs/tst.basename.d.out mode=0444
871 file path=opt/SUNWdtrt/tst/common/funcs/tst.bcopy.d mode=0444
872 file path=opt/SUNWdtrt/tst/common/funcs/tst.chill.ksh mode=0444
873 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d mode=0444
874 file path=opt/SUNWdtrt/tst/common/funcs/tst.cleanpath.d.out mode=0444
875 file path=opt/SUNWdtrt/tst/common/funcs/tst.copyin.d mode=0444
876 file path=opt/SUNWdtrt/tst/common/funcs/tst.copyinto.d mode=0444
877 file path=opt/SUNWdtrt/tst/common/funcs/tst.ddi_pathname.d mode=0444
878 file path=opt/SUNWdtrt/tst/common/funcs/tst.default.d mode=0444
879 file path=opt/SUNWdtrt/tst/common/funcs/tst.freopen.ksh mode=0444
880 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh mode=0444
881 file path=opt/SUNWdtrt/tst/common/funcs/tst.ftruncate.ksh.out mode=0444
882 file path=opt/SUNWdtrt/tst/common/funcs/tst.hton.d mode=0444
883 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d mode=0444
884 file path=opt/SUNWdtrt/tst/common/funcs/tst.index.d.out mode=0444
885 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d mode=0444
886 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa.d.out mode=0444
887 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d mode=0444
888 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntoa6.d.out mode=0444
889 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d mode=0444
890 file path=opt/SUNWdtrt/tst/common/funcs/tst.inet_ntop.d.out mode=0444
891 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d mode=0444
892 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostr.d.out mode=0444
893 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d mode=0444
894 file path=opt/SUNWdtrt/tst/common/funcs/tst.lltostrbase.d.out mode=0444
895 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owned.d mode=0444
896 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_owner.d mode=0444
897 file path=opt/SUNWdtrt/tst/common/funcs/tst.mutex_type_adaptive.d mode=0444
898 file path=opt/SUNWdtrt/tst/common/funcs/tst.progenyof.d mode=0444
899 file path=opt/SUNWdtrt/tst/common/funcs/tst.rand.d mode=0444
900 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d mode=0444
901 file path=opt/SUNWdtrt/tst/common/funcs/tst.strchr.d.out mode=0444
902 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d mode=0444
903 file path=opt/SUNWdtrt/tst/common/funcs/tst.strjoin.d.out mode=0444
904 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d mode=0444
905 file path=opt/SUNWdtrt/tst/common/funcs/tst.strstr.d.out mode=0444
906 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok.d mode=0444
907 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok.d.out mode=0444
908 file path=opt/SUNWdtrt/tst/common/funcs/tst.strtok_null.d mode=0444
909 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d mode=0444
910 file path=opt/SUNWdtrt/tst/common/funcs/tst.substr.d.out mode=0444
911 file path=opt/SUNWdtrt/tst/common/funcs/tst.substrminate.d mode=0444
912 file path=opt/SUNWdtrt/tst/common/funcs/tst.substrminate.d.out mode=0444
913 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d mode=0444
914 file path=opt/SUNWdtrt/tst/common/funcs/tst.system.d.out mode=0444
915 file path=opt/SUNWdtrt/tst/common/funcs/tst.tolower.d mode=0444
916 file path=opt/SUNWdtrt/tst/common/funcs/tst.toupper.d mode=0444
917 file path=opt/SUNWdtrt/tst/common/grammar/err.D_ADDROF_LVAL.d mode=0444
918 file path=opt/SUNWdtrt/tst/common/grammar/err.D_EMPTY.empty.d mode=0444
```

```
file path=opt/SUNWdtrt/tst/common/grammar/tst.clauses.d mode=0444
file path=opt/SUNWdtrt/tst/common/grammar/tst.stmts.d mode=0444
file path=opt/SUNWdtrt/tst/common/include/tst.includefirst.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef1.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_DECL_IDRED.redef2.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_IDENT_UNDEF.recur.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef1.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.baddef2.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/err.D_OP_INCOMPAT.badxlate.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineDataAssign.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineExpression.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineKinds.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineTypedef.d mode=0444
file path=opt/SUNWdtrt/tst/common/inline/tst.InlineWritableAssign.d mode=0444
file path=opt/SUNWdtrt/tst/common/io/tst.fds.d mode=0444
file path=opt/SUNWdtrt/tst/common/io/tst.fds.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/io/tst.fds.exe mode=0555
file path=opt/SUNWdtrt/tst/common/ip/get.ipv4remote.pl mode=0555
file path=opt/SUNWdtrt/tst/common/ip/get.ipv6remote.pl mode=0555
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localicmp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localtcp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4localudp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteicmp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remotetcp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv4remoteudp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6localicmp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remoteicmp.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.ipv6remoteicmp.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.localtcpstate.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/ip/tst.remotetcpstate.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/test.jar
file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Abort.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Bean.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Close.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Drop.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.Enable.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.exe mode=0555
file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.FunctionLookup.ksh.out \
    mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.GetAggregate.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.MaxConsumers.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.MultiAggPrinta.ksh.out \
    mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.exe mode=0555
file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeData.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh mode=0444
```



```
file path=opt/SUNWdtrt/tst/common/java_api/tst.ProbeDescription.ksh.out \
    mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.StateMachine.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.StopLock.ksh.out mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d mode=0444
file path=opt/SUNWdtrt/tst/common/java_api/tst.printa.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.general.d mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.general.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.strsize.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.usdt.d.out mode=0444
file path=opt/SUNWdtrt/tst/common/json/tst.usdt.exe mode=0555
file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NL.char.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_CHR_NULL.char.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_DIGIT.InvalidDigit.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_INT_OFLOW.BigInt.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_STR_NL.string.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brace1.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brace2.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack1.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack2.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.brack3.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren1.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren2.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/err.D_SYNTAX.paren3.d mode=0444
file path=opt/SUNWdtrt/tst/common/lexer/tst.D_MACRO_OFLOW.ParIntOvflow.d.ksh \
    mode=0444
file \
    path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOREVEN.nodivide.d
    mode=0444
file \
    path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTOREVEN.notfactor.d
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTORMATCH.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTORNSTEPS.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTORSMALL.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTORTYPE.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_FACTORVAL.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGHMATCH.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGHTYPE.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_HIGHVAL.d mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOWMATCH.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOWTYPE.d mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_LOWVAL.d mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAGRANGE.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_MAGTOOBIG.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEPMATCH.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEPTYPE.d \
    mode=0444
file path=opt/SUNWdtrt/tst/common/llquantize/err.D_LLQUANT_NSTEPVAL.d \
    mode=0444
```

```
1051 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d mode=0444
1052 file path=opt/SUNWdtrt/tst/common/llquantize/tst.bases.d.out mode=0444
1053 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d mode=0444
1054 file path=opt/SUNWdtrt/tst/common/llquantize/tst.basic.d.out mode=0444
1055 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d mode=0444
1056 file path=opt/SUNWdtrt/tst/common/llquantize/tst.clear.d.out mode=0444
1057 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d mode=0444
1058 file path=opt/SUNWdtrt/tst/common/llquantize/tst.multiaggs.d.out mode=0444
1059 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d mode=0444
1060 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negorder.d.out mode=0444
1061 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d mode=0444
1062 file path=opt/SUNWdtrt/tst/common/llquantize/tst.negvalue.d.out mode=0444
1063 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d mode=0444
1064 file path=opt/SUNWdtrt/tst/common/llquantize/tst.normal.d.out mode=0444
1065 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d mode=0444
1066 file path=opt/SUNWdtrt/tst/common/llquantize/tst.range.d.out mode=0444
1067 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d mode=0444
1068 file path=opt/SUNWdtrt/tst/common/llquantize/tst.steps.d.out mode=0444
1069 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d mode=0444
1070 file path=opt/SUNWdtrt/tst/common/llquantize/tst.trunc.d.out mode=0444
1071 file path=opt/SUNWdtrt/tst/common/mdb/tst.dtracedcmd.ksh mode=0444
1072 file path=opt/SUNWdtrt/tst/common/mib/tst.icmp.ksh mode=0444
1073 file path=opt/SUNWdtrt/tst/common/mib/tst.tcp.ksh mode=0444
1074 file path=opt/SUNWdtrt/tst/common/mib/tst.udp.ksh mode=0444
1075 file path=opt/SUNWdtrt/tst/common/misc/err.D_PRAGMA_OPTSET.d mode=0444
1076 file path=opt/SUNWdtrt/tst/common/misc/tst.badopt.d mode=0444
1077 file path=opt/SUNWdtrt/tst/common/misc/tst.boolopt.d mode=0444
1078 file path=opt/SUNWdtrt/tst/common/misc/tst.boolopt.d.out mode=0444
1079 file path=opt/SUNWdtrt/tst/common/misc/tst.dofmax.ksh mode=0444
1080 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d mode=0444
1081 file path=opt/SUNWdtrt/tst/common/misc/tst.dynopt.d.out mode=0444
1082 file path=opt/SUNWdtrt/tst/common/misc/tst.enablerace.ksh mode=0444
1083 file path=opt/SUNWdtrt/tst/common/misc/tst.haslam.d mode=0444
1084 file path=opt/SUNWdtrt/tst/common/misc/tst.include.ksh mode=0444
1085 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh mode=0444
1086 file path=opt/SUNWdtrt/tst/common/misc/tst.macroglob.ksh.out mode=0444
1087 file path=opt/SUNWdtrt/tst/common/misc/tst.roch.d mode=0444
1088 file path=opt/SUNWdtrt/tst/common/misc/tst.schrock.ksh mode=0444
1089 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGKEY.d mode=0444
1090 file path=opt/SUNWdtrt/tst/common/multiaggs/err.D_PRINTA_AGGPROTO.d mode=0444
1091 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d mode=0444
1092 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.many.d.out mode=0444
1093 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d mode=0444
1094 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.same.d.out mode=0444
1095 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d mode=0444
1096 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sort.d.out mode=0444
1097 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d mode=0444
1098 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.sortpos.d.out mode=0444
1099 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d mode=0444
1100 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.tuplecompat.d.out mode=0444
1101 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d mode=0444
1102 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero.d.out mode=0444
1103 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d mode=0444
1104 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero2.d.out mode=0444
1105 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d mode=0444
1106 file path=opt/SUNWdtrt/tst/common/multiaggs/tst.zero3.d.out mode=0444
1107 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.d mode=0444
1108 file path=opt/SUNWdtrt/tst/common/nfs/tst.call.exe mode=0555
1109 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.d mode=0444
1110 file path=opt/SUNWdtrt/tst/common/nfs/tst.call3.exe mode=0555
1111 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_BITFIELD.bitfield.d \
1112     mode=0444
1113 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.badtype.d \
1114     mode=0444
1115 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_OFFSETOF_TYPE.notsou.d \
1116     mode=0444
```



```
1117 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.OffsetofNULL.d \
1118     mode=0444
1119 file path=opt/SUNWdtrt/tst/common/offsetof/err.D_UNKNOWN.badmemb.d mode=0444
1120 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofAlias.d mode=0444
1121 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofArith.d mode=0444
1122 file path=opt/SUNWdtrt/tst/common/offsetof/tst.OffsetofUnion.d mode=0444
1123 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d mode=0444
1124 file path=opt/SUNWdtrt/tst/common/offsetof/tst.struct.d.out mode=0444
1125 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d mode=0444
1126 file path=opt/SUNWdtrt/tst/common/offsetof/tst.union.d.out mode=0444
1127 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d mode=0444
1128 file path=opt/SUNWdtrt/tst/common/operators/tst.ternary.d.out mode=0444
1129 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.d mode=0444
1130 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badlib.exe mode=0555
1131 file path=opt/SUNWdtrt/tst/common/pid/err.D_PDESC_ZERO.badproc1.d mode=0444
1132 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_BADPID.badproc2.d mode=0444
1133 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.d mode=0444
1134 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_CREATEFAIL.many.exe mode=0555
1135 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.d mode=0444
1136 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_FUNC.badfunc.exe mode=0555
1137 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.d mode=0444
1138 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_LIB.libdash.exe mode=0555
1139 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.d mode=0444
1140 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.alldash.exe mode=0555
1141 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.d mode=0444
1142 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.badname.exe mode=0555
1143 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.d mode=0444
1144 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_NAME.globdash.exe mode=0555
1145 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.d mode=0444
1146 file path=opt/SUNWdtrt/tst/common/pid/err.D_PROC_OFF.toobig.exe mode=0555
1147 file path=opt/SUNWdtrt/tst/common/pid/tst.addprobes.ksh mode=0444
1148 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.d mode=0444
1149 file path=opt/SUNWdtrt/tst/common/pid/tst.args1.exe mode=0555
1150 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.d mode=0444
1151 file path=opt/SUNWdtrt/tst/common/pid/tst.coverage.exe mode=0555
1152 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d mode=0444
1153 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.d.out mode=0444
1154 file path=opt/SUNWdtrt/tst/common/pid/tst.emptystack.exe mode=0555
1155 file path=opt/SUNWdtrt/tst/common/pid/tst.float.d mode=0444
1156 file path=opt/SUNWdtrt/tst/common/pid/tst.float.exe mode=0555
1157 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.d mode=0444
1158 file path=opt/SUNWdtrt/tst/common/pid/tst.fork.exe mode=0555
1159 file path=opt/SUNWdtrt/tst/common/pid/tst.killonerror.ksh mode=0444
1160 file path=opt/SUNWdtrt/tst/common/pid/tst.main.ksh mode=0444
1161 file path=opt/SUNWdtrt/tst/common/pid/tst.manypids.ksh mode=0444
1162 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh mode=0444
1163 file path=opt/SUNWdtrt/tst/common/pid/tst.newprobes.ksh.out mode=0444
1164 file path=opt/SUNWdtrt/tst/common/pid/tst.probemod.ksh mode=0444
1165 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex1.ksh mode=0444
1166 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh mode=0444
1167 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex2.ksh.out mode=0444
1168 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh mode=0444
1169 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex3.ksh.out mode=0444
1170 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh mode=0444
1171 file path=opt/SUNWdtrt/tst/common/pid/tst.provregex4.ksh.out mode=0444
1172 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.d mode=0444
1173 file path=opt/SUNWdtrt/tst/common/pid/tst.ret1.exe mode=0555
1174 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.d mode=0444
1175 file path=opt/SUNWdtrt/tst/common/pid/tst.ret2.exe mode=0555
1176 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh mode=0444
1177 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probefunc.ksh.out mode=0444
1178 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probemod.ksh mode=0444
1179 file path=opt/SUNWdtrt/tst/common/pid/tst.utf8probemod.ksh.out mode=0444
1180 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.d mode=0444
1181 file path=opt/SUNWdtrt/tst/common/pid/tst.vfork.exe mode=0555
1182 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.d mode=0444
```

```
1183 file path=opt/SUNWdtrt/tst/common/pid/tst.weak1.exe mode=0555
1184 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.d mode=0444
1185 file path=opt/SUNWdtrt/tst/common/pid/tst.weak2.exe mode=0555
1186 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.d mode=0444
1187 file path=opt/SUNWdtrt/tst/common/plockstat/tst.available.exe mode=0555
1188 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.d mode=0444
1189 file path=opt/SUNWdtrt/tst/common/plockstat/tst.libmap.exe mode=0555
1190 file path=opt/SUNWdtrt/tst/common/pointers/err.BadAlign.d mode=0444
1191 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.ArrayVar.d \
1192     mode=0444
1193 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.DynamicVar.d \
1194     mode=0444
1195 file path=opt/SUNWdtrt/tst/common/pointers/err.D_ADDROF_VAR.agg.d mode=0444
1196 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_NONPTR.noptr.d \
1197     mode=0444
1198 file path=opt/SUNWdtrt/tst/common/pointers/err.D_DEREF_VOID.VoidPointerDeref.d \
1199     mode=0444
1200 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_ARRFUN.ArrayAssignment.d \
1201     mode=0444
1202 file \
1203     path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_INCOMPAT.VoidPointerArith.d \
1204     mode=0444
1205 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_LVAL.AddressChange.d \
1206     mode=0444
1207 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.NonPointerAccess.d \
1208     mode=0444
1209 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_PTR.badpointer.d mode=0444
1210 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.BadPointerAccess.d \
1211     mode=0444
1212 file path=opt/SUNWdtrt/tst/common/pointers/err.D_OP_SOU.badpointer.d mode=0444
1213 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress1.d mode=0444
1214 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress2.d mode=0444
1215 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress3.d mode=0444
1216 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress4.d mode=0444
1217 file path=opt/SUNWdtrt/tst/common/pointers/err.InvalidAddress5.d mode=0444
1218 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer1.d mode=0444
1219 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer2.d mode=0444
1220 file path=opt/SUNWdtrt/tst/common/pointers/tst.ArrayPointer3.d mode=0444
1221 file path=opt/SUNWdtrt/tst/common/pointers/tst.GlobalVar.d mode=0444
1222 file path=opt/SUNWdtrt/tst/common/pointers/tst.IntegerArithmetic1.d mode=0444
1223 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic1.d mode=0444
1224 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic2.d mode=0444
1225 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerArithmetic3.d mode=0444
1226 file path=opt/SUNWdtrt/tst/common/pointers/tst.PointerAssignment.d mode=0444
1227 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer1.d mode=0444
1228 file path=opt/SUNWdtrt/tst/common/pointers/tst.ValidPointer2.d mode=0444
1229 file path=opt/SUNWdtrt/tst/common/pointers/tst.VoidCast.d mode=0444
1230 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast1.d mode=0444
1231 file path=opt/SUNWdtrt/tst/common/pointers/tst.assigncast2.d mode=0444
1232 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic1.d mode=0444
1233 file path=opt/SUNWdtrt/tst/common/pointers/tst.basic2.d mode=0444
1234 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGERR.d mode=0444
1235 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_DEPEND.main.d mode=0444
1236 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_INVAL.d mode=0444
1237 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_MALFORM.d mode=0444
1238 file path=opt/SUNWdtrt/tst/common/pragma/err.D_PRAGMA_UNUSED.UnusedPragma.d \
1239     mode=0444
1240 file path=opt/SUNWdtrt/tst/common/pragma/err.circlibdep.ksh mode=0444
1241 file path=opt/SUNWdtrt/tst/common/pragma/err.invalidlibdep.ksh mode=0444
1242 file path=opt/SUNWdtrt/tst/common/pragma/tst.libchain.ksh mode=0444
1243 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdep.ksh mode=0444
1244 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepfullyconnected.ksh \
1245     mode=0444
1246 file path=opt/SUNWdtrt/tst/common/pragma/tst.libdepsepdir.ksh mode=0444
1247 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal.ksh mode=0444
1248 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal2.ksh mode=0444
```



```
1249 file path=opt/SUNWdtrt/tst/common/pragma/tst.temporal3.d mode=0444
1250 file path=opt/SUNWdtrt/tst/common/predicates/err.D_PRED_SCALAR.NonScalarPred.d \
1251     mode=0444
1252 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.invalid.d mode=0444
1253 file path=opt/SUNWdtrt/tst/common/predicates/err.D_SYNTAX.operr.d mode=0444
1254 file path=opt/SUNWdtrt/tst/common/predicates/tst.argsnotcached.d mode=0444
1255 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d mode=0444
1256 file path=opt/SUNWdtrt/tst/common/predicates/tst.basics.d.out mode=0444
1257 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d mode=0444
1258 file path=opt/SUNWdtrt/tst/common/predicates/tst.complex.d.out mode=0444
1259 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_IDENT_UNDEF.afterprobe.d \
1260     mode=0444
1261 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_PRAGCTL_INVAL.tabdefine.d \
1262     mode=0444
1263 file path=opt/SUNWdtrt/tst/common/preprocessor/err.D_SYNTAX.withoutpound.d \
1264     mode=0444
1265 file path=opt/SUNWdtrt/tst/common/preprocessor/err.defincomp.d mode=0444
1266 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefelsenotendif.d \
1267     mode=0444
1268 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefincomp.d mode=0444
1269 file path=opt/SUNWdtrt/tst/common/preprocessor/err.ifdefnotendif.d mode=0444
1270 file path=opt/SUNWdtrt/tst/common/preprocessor/err.incompelse.d mode=0444
1271 file path=opt/SUNWdtrt/tst/common/preprocessor/err.mulelse.d mode=0444
1272 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d mode=0444
1273 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifdef.d.out mode=0444
1274 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d mode=0444
1275 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifndef.d.out mode=0444
1276 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d mode=0444
1277 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.ifnotdef.d.out mode=0444
1278 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d mode=0444
1279 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicaland.d.out mode=0444
1280 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d mode=0444
1281 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalandor.d.out \
1282     mode=0444
1283 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d mode=0444
1284 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.logicalor.d.out mode=0444
1285 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d mode=0444
1286 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.muland.d.out mode=0444
1287 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d mode=0444
1288 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.mulor.d.out mode=0444
1289 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d mode=0444
1290 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.precondi.d.out mode=0444
1291 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.predicatedeclare.d \
1292     mode=0444
1293 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d mode=0444
1294 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexp.d.out mode=0444
1295 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d mode=0444
1296 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpelse.d.out mode=0444
1297 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d mode=0444
1298 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpif.d.out mode=0444
1299 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d mode=0444
1300 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.preexpifelse.d.out \
1301     mode=0444
1302 file path=opt/SUNWdtrt/tst/common/preprocessor/tst.withinprobe.d mode=0444
1303 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_AGG.bad.d mode=0444
1304 file path=opt/SUNWdtrt/tst/common/print/err.D_PRINT_VOID.bad.d mode=0444
1305 file path=opt/SUNWdtrt/tst/common/print/err.D_PROTO_LEN.bad.d mode=0444
1306 file path=opt/SUNWdtrt/tst/common/print/tst.array.d mode=0444
1307 file path=opt/SUNWdtrt/tst/common/print/tst.array.d.out mode=0444
1308 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d mode=0444
1309 file path=opt/SUNWdtrt/tst/common/print/tst.bitfield.d.out mode=0444
1310 file path=opt/SUNWdtrt/tst/common/print/tst.dyn.d mode=0444
1311 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d mode=0444
1312 file path=opt/SUNWdtrt/tst/common/print/tst.enum.d.out mode=0444
1313 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d mode=0444
1314 file path=opt/SUNWdtrt/tst/common/print/tst.primitive.d.out mode=0444
```

```
1315 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d mode=0444
1316 file path=opt/SUNWdtrt/tst/common/print/tst.struct.d.out mode=0444
1317 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d mode=0444
1318 file path=opt/SUNWdtrt/tst/common/print/tst.xlate.d.out mode=0444
1319 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badagg.d \
1320     mode=0444
1321 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badfmt.d \
1322     mode=0444
1323 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_AGGARG.badval.d \
1324     mode=0444
1325 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTA_PROTO.bad.d mode=0444
1326 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.jstack.d \
1327     mode=0444
1328 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.stack.d \
1329     mode=0444
1330 file path=opt/SUNWdtrt/tst/common/printa/err.D_PRINTF_ARG_TYPE.ustack.d \
1331     mode=0444
1332 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d mode=0444
1333 file path=opt/SUNWdtrt/tst/common/printa/tst.basics.d.out mode=0444
1334 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d mode=0444
1335 file path=opt/SUNWdtrt/tst/common/printa/tst.def.d.out mode=0444
1336 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d mode=0444
1337 file path=opt/SUNWdtrt/tst/common/printa/tst.dynwidth.d.out mode=0444
1338 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d mode=0444
1339 file path=opt/SUNWdtrt/tst/common/printa/tst.fmt.d.out mode=0444
1340 file path=opt/SUNWdtrt/tst/common/printa/tst.largeusersym.ksh mode=0444
1341 file path=opt/SUNWdtrt/tst/common/printa/tst.many.d mode=0444
1342 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d mode=0444
1343 file path=opt/SUNWdtrt/tst/common/printa/tst.manyval.d.out mode=0444
1344 file path=opt/SUNWdtrt/tst/common/printa/tst.stack.d mode=0444
1345 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d mode=0444
1346 file path=opt/SUNWdtrt/tst/common/printa/tst.tuple.d.out mode=0444
1347 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh mode=0444
1348 file path=opt/SUNWdtrt/tst/common/printa/tst.walltimestamp.ksh.out mode=0444
1349 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_AGG_CONV.aggfmt.d \
1350     mode=0444
1351 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.toomany.d \
1352     mode=0444
1353 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_EXTRA.widths.d \
1354     mode=0444
1355 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_FMT.badfmt.d \
1356     mode=0444
1357 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_PROTO.novalue.d \
1358     mode=0444
1359 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.aggarg.d \
1360     mode=0444
1361 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_ARG_TYPE.recursive.d \
1362     mode=0444
1363 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.noprec.d \
1364     mode=0444
1365 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_PROTO.nowidth.d \
1366     mode=0444
1367 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badprec.d \
1368     mode=0444
1369 file path=opt/SUNWdtrt/tst/common/printf/err.D_PRINTF_DYN_TYPE.badwidth.d \
1370     mode=0444
1371 file path=opt/SUNWdtrt/tst/common/printf/err.D_PROTO_LEN.toofew.d mode=0444
1372 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv1.d mode=0444
1373 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv2.d mode=0444
1374 file path=opt/SUNWdtrt/tst/common/printf/err.D_SYNTAX.badconv3.d mode=0444
1375 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d mode=0444
1376 file path=opt/SUNWdtrt/tst/common/printf/tst.basics.d.out mode=0444
1377 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d mode=0444
1378 file path=opt/SUNWdtrt/tst/common/printf/tst.flags.d.out mode=0444
1379 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d mode=0444
1380 file path=opt/SUNWdtrt/tst/common/printf/tst.hello.d.out mode=0444
```



```
1381 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d mode=0444
1382 file path=opt/SUNWdtrt/tst/common/printf/tst.ints.d.out mode=0444
1383 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d mode=0444
1384 file path=opt/SUNWdtrt/tst/common/printf/tst.precs.d.out mode=0444
1385 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d mode=0444
1386 file path=opt/SUNWdtrt/tst/common/printf/tst.print-f.d.out mode=0444
1387 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh mode=0444
1388 file path=opt/SUNWdtrt/tst/common/printf/tst.printT.ksh.out mode=0444
1389 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh mode=0444
1390 file path=opt/SUNWdtrt/tst/common/printf/tst.printY.ksh.out mode=0444
1391 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d mode=0444
1392 file path=opt/SUNWdtrt/tst/common/printf/tst.printcont.d.out mode=0444
1393 file path=opt/SUNWdtrt/tst/common/printf/tst.printeE.d mode=0444
1394 file path=opt/SUNWdtrt/tst/common/printf/tst.printeE.d.out mode=0444
1395 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d mode=0444
1396 file path=opt/SUNWdtrt/tst/common/printf/tst.printgG.d.out mode=0444
1397 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d mode=0444
1398 file path=opt/SUNWdtrt/tst/common/printf/tst.rawfmt.d.out mode=0444
1399 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d mode=0444
1400 file path=opt/SUNWdtrt/tst/common/printf/tst.signs.d.out mode=0444
1401 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d mode=0444
1402 file path=opt/SUNWdtrt/tst/common/printf/tst.str.d.out mode=0444
1403 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d mode=0444
1404 file path=opt/SUNWdtrt/tst/common/printf/tst.sym.d.out mode=0444
1405 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d mode=0444
1406 file path=opt/SUNWdtrt/tst/common/printf/tst.uints.d.out mode=0444
1407 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d mode=0444
1408 file path=opt/SUNWdtrt/tst/common/printf/tst.widths.d.out mode=0444
1409 file path=opt/SUNWdtrt/tst/common/printf/tst.widths1.d mode=0444
1410 file path=opt/SUNWdtrt/tst/common/printf/tst.wp.d mode=0444
1411 file path=opt/SUNWdtrt/tst/common/printf/tst.wp.d.out mode=0444
1412 file path=opt/SUNWdtrt/tst/common/privs/tst.fds.ksh mode=0444
1413 file path=opt/SUNWdtrt/tst/common/privs/tst.func_access.ksh mode=0444
1414 file path=opt/SUNWdtrt/tst/common/privs/tst.getf.ksh mode=0444
1415 file path=opt/SUNWdtrt/tst/common/privs/tst.kpriv.ksh mode=0444
1416 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivdrop.ksh mode=0444
1417 file path=opt/SUNWdtrt/tst/common/privs/tst.noprivrestrict.ksh mode=0444
1418 file path=opt/SUNWdtrt/tst/common/privs/tst.op_access.ksh mode=0444
1419 file path=opt/SUNWdtrt/tst/common/privs/tst.procpriv.ksh mode=0444
1420 file path=opt/SUNWdtrt/tst/common/privs/tst.providers.ksh mode=0444
1421 file path=opt/SUNWdtrt/tst/common/privs/tst.tick.ksh mode=0444
1422 file path=opt/SUNWdtrt/tst/common/privs/tst.unpriv_funcs.ksh mode=0444
1423 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probeqtn.d mode=0444
1424 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.probestar.d \
1425     mode=0444
1426 file path=opt/SUNWdtrt/tst/common/probes/err.D_PDESC_ZERO.tickstar.d mode=0444
1427 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.assign.d mode=0444
1428 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declare.d mode=0444
1429 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.declarein.d mode=0444
1430 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.lbraces.d mode=0444
1431 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.probespec.d mode=0444
1432 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.rbraces.d mode=0444
1433 file path=opt/SUNWdtrt/tst/common/probes/err.D_SYNTAX.recdec.d mode=0444
1434 file path=opt/SUNWdtrt/tst/common/probes/tst.basic1.d mode=0444
1435 file path=opt/SUNWdtrt/tst/common/probes/tst.check.d mode=0444
1436 file path=opt/SUNWdtrt/tst/common/probes/tst.declare.d mode=0444
1437 file path=opt/SUNWdtrt/tst/common/probes/tst.declareafter.d mode=0444
1438 file path=opt/SUNWdtrt/tst/common/probes/tst.emptyprobe.d mode=0444
1439 file path=opt/SUNWdtrt/tst/common/probes/tst.pragma.d mode=0444
1440 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaaftertab.d mode=0444
1441 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmainside.d mode=0444
1442 file path=opt/SUNWdtrt/tst/common/probes/tst.pragmaoutside.d mode=0444
1443 file path=opt/SUNWdtrt/tst/common/probes/tst.probestar.d mode=0444
1444 file path=opt/SUNWdtrt/tst/common/proc/tst.create.ksh mode=0444
1445 file path=opt/SUNWdtrt/tst/common/proc/tst.discard.ksh mode=0444
1446 file path=opt/SUNWdtrt/tst/common/proc/tst.exec.ksh mode=0444
```

```
1447 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ENOENT.ksh mode=0444
1448 file path=opt/SUNWdtrt/tst/common/proc/tst.execfail.ksh mode=0444
1449 file path=opt/SUNWdtrt/tst/common/proc/tst.exitcore.ksh mode=0444
1450 file path=opt/SUNWdtrt/tst/common/proc/tst.exitexit.ksh mode=0444
1451 file path=opt/SUNWdtrt/tst/common/proc/tst.exitkilled.ksh mode=0444
1452 file path=opt/SUNWdtrt/tst/common/proc/tst.signal.ksh mode=0444
1453 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.d mode=0444
1454 file path=opt/SUNWdtrt/tst/common/proc/tst.sigwait.exe mode=0555
1455 file path=opt/SUNWdtrt/tst/common/proc/tst.startexit.ksh mode=0444
1456 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZERO.profile.d \
1457     mode=0444
1458 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonens.d mode=0444
1459 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROonensec.d \
1460     mode=0444
1461 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROoneus.d mode=0444
1462 file path=opt/SUNWdtrt/tst/common/profile-n/err.D_PDESC_ZEROoneusec.d \
1463     mode=0444
1464 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d mode=0444
1465 file path=opt/SUNWdtrt/tst/common/profile-n/tst.argtest.d.out mode=0444
1466 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d mode=0444
1467 file path=opt/SUNWdtrt/tst/common/profile-n/tst.basic.d.out mode=0444
1468 file path=opt/SUNWdtrt/tst/common/profile-n/tst.func.ksh mode=0444
1469 file path=opt/SUNWdtrt/tst/common/profile-n/tst.mod.ksh mode=0444
1470 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilehz.d mode=0444
1471 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilehz.d.out mode=0444
1472 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilems.d mode=0444
1473 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilems.d.out mode=0444
1474 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilemsec.d mode=0444
1475 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilemsec.d.out mode=0444
1476 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d mode=0444
1477 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilenhz.d.out mode=0444
1478 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d mode=0444
1479 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilens.d.out mode=0444
1480 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d mode=0444
1481 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilensec.d.out mode=0444
1482 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d mode=0444
1483 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profiles.d.out mode=0444
1484 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilesec.d mode=0444
1485 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profilesec.d.out mode=0444
1486 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileus.d mode=0444
1487 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileus.d.out mode=0444
1488 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileusec.d mode=0444
1489 file path=opt/SUNWdtrt/tst/common/profile-n/tst.profileusec.d.out mode=0444
1490 file path=opt/SUNWdtrt/tst/common/profile-n/tst.sym.ksh mode=0444
1491 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufunc.ksh mode=0444
1492 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.exe mode=0555
1493 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh mode=0444
1494 file path=opt/SUNWdtrt/tst/common/profile-n/tst.ufuncsort.ksh.out mode=0444
1495 file path=opt/SUNWdtrt/tst/common/profile-n/tst.umod.ksh mode=0444
1496 file path=opt/SUNWdtrt/tst/common/profile-n/tst.usym.ksh mode=0444
1497 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_INVAL.wrongdec4.d \
1498     mode=0444
1499 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.nonprofile.d \
1500     mode=0444
1501 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec1.d \
1502     mode=0444
1503 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec2.d \
1504     mode=0444
1505 file path=opt/SUNWdtrt/tst/common/providers/err.D_PDESC_ZERO.wrongdec3.d \
1506     mode=0444
1507 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d mode=0444
1508 file path=opt/SUNWdtrt/tst/common/providers/tst.basics.d.out mode=0444
1509 file path=opt/SUNWdtrt/tst/common/providers/tst.beginexit.d mode=0444
1510 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d mode=0444
1511 file path=opt/SUNWdtrt/tst/common/providers/tst.beginprof.d.out mode=0444
1512 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d mode=0444
```



```
1513 file path=opt/SUNWdtrt/tst/common/providers/tst.probattrs.d.out mode=0444
1514 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d mode=0444
1515 file path=opt/SUNWdtrt/tst/common/providers/tst.probefunc.d.out mode=0444
1516 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d mode=0444
1517 file path=opt/SUNWdtrt/tst/common/providers/tst.probemod.d.out mode=0444
1518 file path=opt/SUNWdtrt/tst/common/providers/tst.probename.d mode=0444
1519 file path=opt/SUNWdtrt/tst/common/providers/tst.probename.d.out mode=0444
1520 file path=opt/SUNWdtrt/tst/common/providers/tst.probprov.d mode=0444
1521 file path=opt/SUNWdtrt/tst/common/providers/tst.probprov.d.out mode=0444
1522 file path=opt/SUNWdtrt/tst/common/providers/tst.profend.d mode=0444
1523 file path=opt/SUNWdtrt/tst/common/providers/tst.profend.d.out mode=0444
1524 file path=opt/SUNWdtrt/tst/common/providers/tst.profexit.d mode=0444
1525 file path=opt/SUNWdtrt/tst/common/providers/tst.profexit.d.out mode=0444
1526 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d mode=0444
1527 file path=opt/SUNWdtrt/tst/common/providers/tst.trace.d.out mode=0444
1528 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d mode=0444
1529 file path=opt/SUNWdtrt/tst/common/providers/tst.twoprof.d.out mode=0444
1530 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.d mode=0444
1531 file path=opt/SUNWdtrt/tst/common/raise/tst.raise1.exe mode=0555
1532 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.d mode=0444
1533 file path=opt/SUNWdtrt/tst/common/raise/tst.raise2.exe mode=0555
1534 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.d mode=0444
1535 file path=opt/SUNWdtrt/tst/common/raise/tst.raise3.exe mode=0555
1536 file path=opt/SUNWdtrt/tst/common/rates/tst.aggrate.d mode=0444
1537 file path=opt/SUNWdtrt/tst/common/rates/tst.aggrate.d.out mode=0444
1538 file path=opt/SUNWdtrt/tst/common/rates/tst.statusrate.d mode=0444
1539 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d mode=0444
1540 file path=opt/SUNWdtrt/tst/common/rates/tst.switchrate.d.out mode=0444
1541 file path=opt/SUNWdtrt/tst/common/safety/tst.basename.d mode=0444
1542 file path=opt/SUNWdtrt/tst/common/safety/tst.caller.d mode=0444
1543 file path=opt/SUNWdtrt/tst/common/safety/tst.cleanpath.d mode=0444
1544 file path=opt/SUNWdtrt/tst/common/safety/tst.copyin.d mode=0444
1545 file path=opt/SUNWdtrt/tst/common/safety/tst.copyin2.d mode=0444
1546 file path=opt/SUNWdtrt/tst/common/safety/tst.ddi_pathname.d mode=0444
1547 file path=opt/SUNWdtrt/tst/common/safety/tst.dirname.d mode=0444
1548 file path=opt/SUNWdtrt/tst/common/safety/tst.errno.d mode=0444
1549 file path=opt/SUNWdtrt/tst/common/safety/tst.execname.d mode=0444
1550 file path=opt/SUNWdtrt/tst/common/safety/tst.gid.d mode=0444
1551 file path=opt/SUNWdtrt/tst/common/safety/tst.hton.d mode=0444
1552 file path=opt/SUNWdtrt/tst/common/safety/tst.index.d mode=0444
1553 file path=opt/SUNWdtrt/tst/common/safety/tst.msgdsize.d mode=0444
1554 file path=opt/SUNWdtrt/tst/common/safety/tst.msgsize.d mode=0444
1555 file path=opt/SUNWdtrt/tst/common/safety/tst.null.d mode=0444
1556 file path=opt/SUNWdtrt/tst/common/safety/tst.pid.d mode=0444
1557 file path=opt/SUNWdtrt/tst/common/safety/tst.ppid.d mode=0444
1558 file path=opt/SUNWdtrt/tst/common/safety/tst.progenyof.d mode=0444
1559 file path=opt/SUNWdtrt/tst/common/safety/tst.random.d mode=0444
1560 file path=opt/SUNWdtrt/tst/common/safety/tst.rw.d mode=0444
1561 file path=opt/SUNWdtrt/tst/common/safety/tst.shortstr.d mode=0444
1562 file path=opt/SUNWdtrt/tst/common/safety/tst.stack.d mode=0444
1563 file path=opt/SUNWdtrt/tst/common/safety/tst.stackdepth.d mode=0444
1564 file path=opt/SUNWdtrt/tst/common/safety/tst.stddev.d mode=0444
1565 file path=opt/SUNWdtrt/tst/common/safety/tst.strchr.d mode=0444
1566 file path=opt/SUNWdtrt/tst/common/safety/tst.strjoin.d mode=0444
1567 file path=opt/SUNWdtrt/tst/common/safety/tst.strstr.d mode=0444
1568 file path=opt/SUNWdtrt/tst/common/safety/tst.strtok.d mode=0444
1569 file path=opt/SUNWdtrt/tst/common/safety/tst.substr.d mode=0444
1570 file path=opt/SUNWdtrt/tst/common/safety/tst.ucaller.d mode=0444
1571 file path=opt/SUNWdtrt/tst/common/safety/tst.uid.d mode=0444
1572 file path=opt/SUNWdtrt/tst/common/safety/tst.unalign.d mode=0444
1573 file path=opt/SUNWdtrt/tst/common/safety/tst.uregs.d mode=0444
1574 file path=opt/SUNWdtrt/tst/common/safety/tst.ustack.d mode=0444
1575 file path=opt/SUNWdtrt/tst/common/safety/tst.ustackdepth.d mode=0444
1576 file path=opt/SUNWdtrt/tst/common/safety/tst.vahole.d mode=0444
1577 file path=opt/SUNWdtrt/tst/common/safety/tst.violentdeath.ksh mode=0444
1578 file path=opt/SUNWdtrt/tst/common/safety/tst.zonename.d mode=0444
```

```
1579 file path=opt/SUNWdtrt/tst/common/scalars/err.D_ARR_LOCAL.thisarray.d \
1580     mode=0444
1581 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.selfthis.d \
1582     mode=0444
1583 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_CLASS.thisself.d \
1584     mode=0444
1585 file path=opt/SUNWdtrt/tst/common/scalars/err.D_DECL_IDRED.errval.d mode=0444
1586 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dec.err.d \
1587     mode=0444
1588 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupgtype.d \
1589     mode=0444
1590 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupltype.d \
1591     mode=0444
1592 file path=opt/SUNWdtrt/tst/common/scalars/err.D_OP_INCOMPAT.dupttype.d \
1593     mode=0444
1594 file path=opt/SUNWdtrt/tst/common/scalars/err.D_SYNTAX.declare.d mode=0444
1595 file path=opt/SUNWdtrt/tst/common/scalars/err.bigglobal.d mode=0444
1596 file path=opt/SUNWdtrt/tst/common/scalars/err.biglocal.d mode=0444
1597 file path=opt/SUNWdtrt/tst/common/scalars/tst.16kglobal.d mode=0444
1598 file path=opt/SUNWdtrt/tst/common/scalars/tst.16klocal.d mode=0444
1599 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d mode=0444
1600 file path=opt/SUNWdtrt/tst/common/scalars/tst.basicvar.d.out mode=0444
1601 file path=opt/SUNWdtrt/tst/common/scalars/tst.localvar.d mode=0444
1602 file path=opt/SUNWdtrt/tst/common/scalars/tst.misc.d mode=0444
1603 file path=opt/SUNWdtrt/tst/common/scalars/tst.self.d mode=0444
1604 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray.d mode=0444
1605 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfarray2.d mode=0444
1606 file path=opt/SUNWdtrt/tst/common/scalars/tst.selfthis.d mode=0444
1607 file path=opt/SUNWdtrt/tst/common/scalars/tst.this.d mode=0444
1608 file path=opt/SUNWdtrt/tst/common/scalars/tst.thisself.d mode=0444
1609 file path=opt/SUNWdtrt/tst/common/sched/tst.enqueue.d mode=0444
1610 file path=opt/SUNWdtrt/tst/common/sched/tst.oncpu.d mode=0444
1611 file path=opt/SUNWdtrt/tst/common/sched/tst.stackdepth.d mode=0444
1612 file path=opt/SUNWdtrt/tst/common/scripting/err.D_MACRO_UNDEF.invalidargs.d \
1613     mode=0444
1614 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_LVAL.rdonly.d mode=0444
1615 file path=opt/SUNWdtrt/tst/common/scripting/err.D_OP_WRITE.usepidmacro.d \
1616     mode=0444
1617 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.concat.d mode=0444
1618 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.desc.d mode=0444
1619 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.inval.d mode=0444
1620 file path=opt/SUNWdtrt/tst/common/scripting/err.D_SYNTAX.pid.d mode=0444
1621 file path=opt/SUNWdtrt/tst/common/scripting/tst.D_MACRO_UNUSED.overflow.ksh \
1622     mode=0444
1623 file path=opt/SUNWdtrt/tst/common/scripting/tst.arg0.d mode=0444
1624 file path=opt/SUNWdtrt/tst/common/scripting/tst.arguments.ksh mode=0444
1625 file path=opt/SUNWdtrt/tst/common/scripting/tst.assign.d mode=0444
1626 file path=opt/SUNWdtrt/tst/common/scripting/tst.basic.d mode=0444
1627 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.d mode=0444
1628 file path=opt/SUNWdtrt/tst/common/scripting/tst.egid.ksh mode=0444
1629 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.d mode=0444
1630 file path=opt/SUNWdtrt/tst/common/scripting/tst.euid.ksh mode=0444
1631 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.d mode=0444
1632 file path=opt/SUNWdtrt/tst/common/scripting/tst.gid.ksh mode=0444
1633 file path=opt/SUNWdtrt/tst/common/scripting/tst.pgid.d mode=0444
1634 file path=opt/SUNWdtrt/tst/common/scripting/tst.pid.d mode=0444
1635 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.d mode=0444
1636 file path=opt/SUNWdtrt/tst/common/scripting/tst.ppid.ksh mode=0444
1637 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.d mode=0444
1638 file path=opt/SUNWdtrt/tst/common/scripting/tst.projid.ksh mode=0444
1639 file path=opt/SUNWdtrt/tst/common/scripting/tst.quite.d mode=0444
1640 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.d mode=0444
1641 file path=opt/SUNWdtrt/tst/common/scripting/tst.sid.ksh mode=0444
1642 file path=opt/SUNWdtrt/tst/common/scripting/tst.stringmacro.ksh mode=0444
1643 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.d mode=0444
1644 file path=opt/SUNWdtrt/tst/common/scripting/tst.taskid.ksh mode=0444
```



```
1645 file path=opt/SUNWdtrt/tst/common/scripting/tst.trace.d mode=0444
1646 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.d mode=0444
1647 file path=opt/SUNWdtrt/tst/common/scripting/tst.uid.ksh mode=0444
1648 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargs.d mode=0444
1649 file path=opt/SUNWdtrt/tst/common/sdt/tst.sdtargs.exe mode=0555
1650 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_BADREF.SizeofAssoc.d \
1651     mode=0444
1652 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_IDENT_UNDEF.UnknownSymbol.d \
1653     mode=0444
1654 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.badstruct.d \
1655     mode=0444
1656 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SIZEOF_TYPE.d mode=0444
1657 file path=opt/SUNWdtrt/tst/common/sizeof/err.D_SYNTAX.SizeofBadType.d \
1658     mode=0444
1659 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofArray.d mode=0444
1660 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofDataTypes.d mode=0444
1661 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofExpression.d mode=0444
1662 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofNULL.d mode=0444
1663 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d mode=0444
1664 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofStrConst.d.out mode=0444
1665 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d mode=0444
1666 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString1.d.out mode=0444
1667 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d mode=0444
1668 file path=opt/SUNWdtrt/tst/common/sizeof/tst.SizeofString2.d.out mode=0444
1669 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations1.d \
1670     mode=0444
1671 file path=opt/SUNWdtrt/tst/common/speculation/err.BufSizeVariations2.d \
1672     mode=0444
1673 file \
1674     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithBreakPo
1675     mode=0444
1676 file \
1677     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithChill.d
1678     mode=0444
1679 file \
1680     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1681     mode=0444
1682 file \
1683     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithCopyOut
1684     mode=0444
1685 file \
1686     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithPanic.d
1687     mode=0444
1688 file \
1689     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithRaise.d
1690     mode=0444
1691 file \
1692     path=opt/SUNWdtrt/tst/common/speculation/err.D_ACT_SPEC.SpeculateWithStop.d
1693     mode=0444
1694 file path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_COMM.AggAftCommit.d \
1695     mode=0444
1696 file \
1697     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithAvg.d \
1698     mode=0444
1699 file \
1700     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithCount.d
1701     mode=0444
1702 file \
1703     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithLquant.
1704     mode=0444
1705 file \
1706     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMax.d \
1707     mode=0444
1708 file \
1709     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithMin.d \
1710     mode=0444
```

```
1711 file \
1712     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithQuant.d
1713     mode=0444
1714 file \
1715     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithStddev.
1716     mode=0444
1717 file \
1718     path=opt/SUNWdtrt/tst/common/speculation/err.D_AGG_SPEC.SpeculateWithSum.d \
1719     mode=0444
1720 file \
1721     path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.CommitAftCommit.d \
1722     mode=0444
1723 file path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_COMM.DisjointCommit.d \
1724     mode=0444
1725 file \
1726     path=opt/SUNWdtrt/tst/common/speculation/err.D_COMM_DREC.CommitAftDataRec.d
1727     mode=0444
1728 file \
1729     path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.DataRecAftCommit.d
1730     mode=0444
1731 file \
1732     path=opt/SUNWdtrt/tst/common/speculation/err.D_DREC_COMM.ExitAfterCommit.d \
1733     mode=0444
1734 file path=opt/SUNWdtrt/tst/common/speculation/err.D_EXIT_SPEC.ExitAftSpec.d \
1735     mode=0444
1736 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_MALFORM.NspecExpr.d \
1737     mode=0444
1738 file \
1739     path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.HugeNspecValue.
1740     mode=0444
1741 file \
1742     path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.InvalidSpecSize
1743     mode=0444
1744 file \
1745     path=opt/SUNWdtrt/tst/common/speculation/err.D_PRAGMA_OPTSET.NegSpecSize.d \
1746     mode=0444
1747 file path=opt/SUNWdtrt/tst/common/speculation/err.D_PROTO_LEN.SpecNoId.d \
1748     mode=0444
1749 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_COMM.SpecAftCommit.d \
1750     mode=0444
1751 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_DREC.SpecAftDataRec.d \
1752     mode=0444
1753 file path=opt/SUNWdtrt/tst/common/speculation/err.D_SPEC_SPEC.SpecAftSpec.d \
1754     mode=0444
1755 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeBufSize.d mode=0444
1756 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeNspec.d mode=0444
1757 file path=opt/SUNWdtrt/tst/common/speculation/err.NegativeSpecSize.d mode=0444
1758 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations1.d \
1759     mode=0444
1760 file path=opt/SUNWdtrt/tst/common/speculation/err.SpecSizeVariations2.d \
1761     mode=0444
1762 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitAfterDiscard.d \
1763     mode=0444
1764 file path=opt/SUNWdtrt/tst/common/speculation/tst.CommitWithZero.d mode=0444
1765 file path=opt/SUNWdtrt/tst/common/speculation/tst.DataRecAftDiscard.d \
1766     mode=0444
1767 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftCommit.d mode=0444
1768 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDataRec.d \
1769     mode=0444
1770 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardAftDiscard.d \
1771     mode=0444
1772 file path=opt/SUNWdtrt/tst/common/speculation/tst.DiscardWithZero.d mode=0444
1773 file path=opt/SUNWdtrt/tst/common/speculation/tst.ExitAftDiscard.d mode=0444
1774 file path=opt/SUNWdtrt/tst/common/speculation/tst.NoSpecBuffer.d mode=0444
1775 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations1.d \
1776     mode=0444
```



```
1777 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations2.d \
1778     mode=0444
1779 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpecSizeVariations3.d \
1780     mode=0444
1781 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculateWithRandom.d \
1782     mode=0444
1783 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationCommit.d \
1784     mode=0444
1785 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationDiscard.d \
1786     mode=0444
1787 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationID.d mode=0444
1788 file path=opt/SUNWdtrt/tst/common/speculation/tst.SpeculationWithZero.d \
1789     mode=0444
1790 file path=opt/SUNWdtrt/tst/common/speculation/tst.TwoSpecBuffers.d mode=0444
1791 file path=opt/SUNWdtrt/tst/common/speculation/tst.negcommit.d mode=0444
1792 file path=opt/SUNWdtrt/tst/common/speculation/tst.negspec.d mode=0444
1793 file path=opt/SUNWdtrt/tst/common/speculation/tst.zerosize.d mode=0444
1794 file path=opt/SUNWdtrt/tst/common/stability/err.D_ATTR_MIN.MinAttributes.d \
1795     mode=0444
1796 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_PROTO.bad.d mode=0444
1797 file path=opt/SUNWdtrt/tst/common/stack/err.D_STACK_SIZE.d mode=0444
1798 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_FRAMES.bad.d mode=0444
1799 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_PROTO.bad.d mode=0444
1800 file path=opt/SUNWdtrt/tst/common/stack/err.D_USTACK_STRSIZE.bad.d mode=0444
1801 file path=opt/SUNWdtrt/tst/common/stack/tst.default.d mode=0444
1802 file path=opt/SUNWdtrt/tst/common/stackdepth/tst.default.d mode=0444
1803 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.d mode=0444
1804 file path=opt/SUNWdtrt/tst/common/stop/tst.stop1.exe mode=0555
1805 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.d mode=0444
1806 file path=opt/SUNWdtrt/tst/common/stop/tst.stop2.exe mode=0555
1807 file path=opt/SUNWdtrt/tst/common/strlen/tst.strlen1.d mode=0444
1808 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooLarge.d mode=0444
1809 file path=opt/SUNWdtrt/tst/common/strtoll/err.BaseTooSmall.d mode=0444
1810 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d mode=0444
1811 file path=opt/SUNWdtrt/tst/common/strtoll/tst.strtoll.d.out mode=0444
1812 file path=opt/SUNWdtrt/tst/common/struct/err.D_ADDROF_VAR.StructPointer.d \
1813     mode=0444
1814 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon.d \
1815     mode=0444
1816 file \
1817     path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_COMBO.StructWithoutColon1.d \
1818     mode=0444
1819 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.circular.d \
1820     mode=0444
1821 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order.d \
1822     mode=0444
1823 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.order2.d \
1824     mode=0444
1825 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.recursive.d \
1826     mode=0444
1827 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_INCOMPLETE.simple.d \
1828     mode=0444
1829 file path=opt/SUNWdtrt/tst/common/struct/err.D_DECL_VOIDOBJ.baddec.d mode=0444
1830 file path=opt/SUNWdtrt/tst/common/struct/err.D_PROTO_ARG.DupStructAssoc.d \
1831     mode=0444
1832 file path=opt/SUNWdtrt/tst/common/struct/tst.StructAssoc.d mode=0444
1833 file path=opt/SUNWdtrt/tst/common/struct/tst.StructDataTypes.d mode=0444
1834 file path=opt/SUNWdtrt/tst/common/struct/tst.StructInside.d mode=0444
1835 file path=opt/SUNWdtrt/tst/common/struct/tst.clauselocal.d mode=0444
1836 file path=opt/SUNWdtrt/tst/common/struct/tst.clauselocal.d.out mode=0444
1837 file path=opt/SUNWdtrt/tst/common/sugar/tst.else.d mode=0444
1838 file path=opt/SUNWdtrt/tst/common/sugar/tst.if.d mode=0444
1839 file path=opt/SUNWdtrt/tst/common/sugar/tst.if2.d mode=0444
1840 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_before_after.d mode=0444
1841 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_nested.d mode=0444
1842 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon.d mode=0444
```

```
1843 file path=opt/SUNWdtrt/tst/common/sugar/tst.if_trailing_semicolon2.d mode=0444
1844 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.d mode=0444
1845 file path=opt/SUNWdtrt/tst/common/syscall/tst.args.exe mode=0555
1846 file path=opt/SUNWdtrt/tst/common/syscall/tst.openret.ksh mode=0444
1847 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.d mode=0444
1848 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post.exe mode=0555
1849 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.d mode=0444
1850 file path=opt/SUNWdtrt/tst/common/sysevent/tst.post_chan.exe mode=0555
1851 file path=opt/SUNWdtrt/tst/common/threadname/tst.threadname.d mode=0444
1852 file path=opt/SUNWdtrt/tst/common/threadname/tst.threadname.exe mode=0555
1853 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZERO.tick.d mode=0444
1854 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonens.d mode=0444
1855 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROonensec.d mode=0444
1856 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneus.d mode=0444
1857 file path=opt/SUNWdtrt/tst/common/tick-n/err.D_PDESC_ZEROoneusec.d mode=0444
1858 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickarg0.d mode=0444
1859 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d mode=0444
1860 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickms.d.out mode=0444
1861 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d mode=0444
1862 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickmsec.d.out mode=0444
1863 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickns.d mode=0444
1864 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickns.d.out mode=0444
1865 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticknsec.d mode=0444
1866 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticknsec.d.out mode=0444
1867 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d mode=0444
1868 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticks.d.out mode=0444
1869 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d mode=0444
1870 file path=opt/SUNWdtrt/tst/common/tick-n/tst.ticksec.d.out mode=0444
1871 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickus.d mode=0444
1872 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickus.d.out mode=0444
1873 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d mode=0444
1874 file path=opt/SUNWdtrt/tst/common/tick-n/tst.tickusec.d.out mode=0444
1875 file path=opt/SUNWdtrt/tst/common/trace/err.D_PROTO_LEN.bad.d mode=0444
1876 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_AGG.bad.d mode=0444
1877 file path=opt/SUNWdtrt/tst/common/trace/err.D_TRACE_VOID.bad.d mode=0444
1878 file path=opt/SUNWdtrt/tst/common/trace/tst.dyn.d mode=0444
1879 file path=opt/SUNWdtrt/tst/common/trace/tst.misc.d mode=0444
1880 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d mode=0444
1881 file path=opt/SUNWdtrt/tst/common/trace/tst.qstring.d.out mode=0444
1882 file path=opt/SUNWdtrt/tst/common/trace/tst.string.d mode=0444
1883 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_ARG.badsize.d mode=0444
1884 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_PROTO_LEN.toofew.d mode=0444
1885 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ADDR.badaddr.d \
1886     mode=0444
1887 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_ARGS.d mode=0444
1888 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_DYNSIZE.d mode=0444
1889 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.negsize.d \
1890     mode=0444
1891 file path=opt/SUNWdtrt/tst/common/tracemem/err.D_TRACEMEM_SIZE.zerosize.d \
1892     mode=0444
1893 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d mode=0444
1894 file path=opt/SUNWdtrt/tst/common/tracemem/tst.dynsize.d.out mode=0444
1895 file path=opt/SUNWdtrt/tst/common/tracemem/tst.rootvp.d mode=0444
1896 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d mode=0444
1897 file path=opt/SUNWdtrt/tst/common/tracemem/tst.smallsize.d.out mode=0444
1898 file \
1899     path=opt/SUNWdtrt/tst/common/translators/err.D_DECL_TYPERED.BadTransDecl.d \
1900     mode=0444
1901 file \
1902     path=opt/SUNWdtrt/tst/common/translators/err.D_OP_INCOMPLETE.NonExistentInpu
1903     mode=0444
1904 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl1.d \
1905     mode=0444
1906 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl3.d \
1907     mode=0444
1908 file path=opt/SUNWdtrt/tst/common/translators/err.D_SYNTAX.BadTransDecl4.d \
```



```
1909     mode=0444
1910 file \
1911     path=opt/SUNWdtrt/tst/common/translators/err.D_TYPE_MEMBER.NonExistentInput2
1912     mode=0444
1913 file \
1914     path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_INCOMPAT.BadInputType1.
1915     mode=0444
1916 file \
1917     path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_MEMB.NonExistentOutput2
1918     mode=0444
1919 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_NONE.BadTransDecl6.d \
1920     mode=0444
1921 file \
1922     path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_REDECL.RepeatTransDecl.
1923     mode=0444
1924 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransDecl8.d \
1925     mode=0444
1926 file path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.BadTransInt.d \
1927     mode=0444
1928 file \
1929     path=opt/SUNWdtrt/tst/common/translators/err.D_XLATE_SOU.NonExistentOutput1.
1930     mode=0444
1931 file path=opt/SUNWdtrt/tst/common/translators/tst.CircularTransDecl.d \
1932     mode=0444
1933 file path=opt/SUNWdtrt/tst/common/translators/tst.EmptyTransDecl.d mode=0444
1934 file path=opt/SUNWdtrt/tst/common/translators/tst.ForwardTag.d mode=0444
1935 file path=opt/SUNWdtrt/tst/common/translators/tst.InputAliasTrans.d mode=0444
1936 file path=opt/SUNWdtrt/tst/common/translators/tst.InputIntTrans.d mode=0444
1937 file path=opt/SUNWdtrt/tst/common/translators/tst.OutputAliasTrans.d mode=0444
1938 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialDereferencing.d \
1939     mode=0444
1940 file path=opt/SUNWdtrt/tst/common/translators/tst.PartialOutputTransDefn.d \
1941     mode=0444
1942 file path=opt/SUNWdtrt/tst/common/translators/tst.ProcModelTrans.d mode=0444
1943 file path=opt/SUNWdtrt/tst/common/translators/tst.RepeatDeclaration.d \
1944     mode=0444
1945 file path=opt/SUNWdtrt/tst/common/translators/tst.SimultaneousTranslators.d \
1946     mode=0444
1947 file path=opt/SUNWdtrt/tst/common/translators/tst.StructureAssignment.d \
1948     mode=0444
1949 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh \
1950     mode=0444
1951 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability1.ksh.out \
1952     mode=0444
1953 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh \
1954     mode=0444
1955 file path=opt/SUNWdtrt/tst/common/translators/tst.TestTransStability2.ksh.out \
1956     mode=0444
1957 file path=opt/SUNWdtrt/tst/common/translators/tst.TransNonPointer.d mode=0444
1958 file path=opt/SUNWdtrt/tst/common/translators/tst.TransOutputPointer.d \
1959     mode=0444
1960 file path=opt/SUNWdtrt/tst/common/translators/tst.TransPointer.d mode=0444
1961 file path=opt/SUNWdtrt/tst/common/translators/tst.TranslateSelf.d mode=0444
1962 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionInputTrans.d mode=0444
1963 file path=opt/SUNWdtrt/tst/common/translators/tst.UnionOutputTrans.d mode=0444
1964 file path=opt/SUNWdtrt/tst/common/typedef/err.D_DECL_IDRED.DupTypeDef.d \
1965     mode=0444
1966 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.BadExistingTypedef.d \
1967     mode=0444
1968 file path=opt/SUNWdtrt/tst/common/typedef/err.D_SYNTAX.TypedefInClause.d \
1969     mode=0444
1970 file path=opt/SUNWdtrt/tst/common/typedef/tst.ChainTypedef.d mode=0444
1971 file path=opt/SUNWdtrt/tst/common/typedef/tst.TypedefDataAssign.d mode=0444
1972 file path=opt/SUNWdtrt/tst/common/types/err.D_CAST_INVAL.badcast.d mode=0444
1973 file path=opt/SUNWdtrt/tst/common/types/err.D_CG_DYN.ResultDynType.d mode=0444
1974 file path=opt/SUNWdtrt/tst/common/types/err.D_CHR_OFLOW.charconst.d mode=0444
```

```
1975 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_BADCLASS.bad.d mode=0444
1976 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_CHARATTR.badtype3.d \
1977     mode=0444
1978 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype4.d mode=0444
1979 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.badtype5.d mode=0444
1980 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattr1.d mode=0444
1981 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattr2.d mode=0444
1982 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattr3.d mode=0444
1983 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattr4.d mode=0444
1984 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_COMBO.rptdattr5.d mode=0444
1985 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENCONST.badeval.d mode=0444
1986 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enoflow.d mode=0444
1987 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_ENOFLOW.enuflow.d mode=0444
1988 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_SCOPE.scopeop.d mode=0444
1989 file path=opt/SUNWdtrt/tst/common/types/err.D_DECL_USELESS.baddec.d mode=0444
1990 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ACT.badcond.d mode=0444
1991 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_ARITH.badoperand.d mode=0444
1992 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INCOMPAT.badassign.d \
1993     mode=0444
1994 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badbitop.d mode=0444
1995 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_INT.badshift.d mode=0444
1996 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badcond.d mode=0444
1997 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badincop.d mode=0444
1998 file path=opt/SUNWdtrt/tst/common/types/err.D_OP_SCALAR.badlogop.d mode=0444
1999 file path=opt/SUNWdtrt/tst/common/types/err.D_PROTO_LEN.badcond1.d mode=0444
2000 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badenum.d mode=0444
2001 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badid.d mode=0444
2002 file path=opt/SUNWdtrt/tst/common/types/err.D_SYNTAX.badstruct.d mode=0444
2003 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype1.d mode=0444
2004 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.badtype2.d mode=0444
2005 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupenum.d mode=0444
2006 file path=opt/SUNWdtrt/tst/common/types/err.D_UNKNOWN.dupstruct.d mode=0444
2007 file path=opt/SUNWdtrt/tst/common/types/err.D_XLATE_REDECL.ResultDynType.d \
2008     mode=0444
2009 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt1.d mode=0444
2010 file path=opt/SUNWdtrt/tst/common/types/err.EDT_NOTYPE.cmbdatt2.d mode=0444
2011 file path=opt/SUNWdtrt/tst/common/types/tst.assignops.d mode=0444
2012 file path=opt/SUNWdtrt/tst/common/types/tst.badshiftops.d mode=0444
2013 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d mode=0444
2014 file path=opt/SUNWdtrt/tst/common/types/tst.basics.d.out mode=0444
2015 file path=opt/SUNWdtrt/tst/common/types/tst.bitops.d mode=0444
2016 file path=opt/SUNWdtrt/tst/common/types/tst.charconstants.d mode=0444
2017 file path=opt/SUNWdtrt/tst/common/types/tst.complex.d mode=0444
2018 file path=opt/SUNWdtrt/tst/common/types/tst.condexpr.d mode=0444
2019 file path=opt/SUNWdtrt/tst/common/types/tst.const.d mode=0444
2020 file path=opt/SUNWdtrt/tst/common/types/tst.constants.d mode=0444
2021 file path=opt/SUNWdtrt/tst/common/types/tst.conv.d mode=0444
2022 file path=opt/SUNWdtrt/tst/common/types/tst.enum.d mode=0444
2023 file path=opt/SUNWdtrt/tst/common/types/tst.intincop.d mode=0444
2024 file path=opt/SUNWdtrt/tst/common/types/tst.intops.d mode=0444
2025 file path=opt/SUNWdtrt/tst/common/types/tst.inttypes.d mode=0444
2026 file path=opt/SUNWdtrt/tst/common/types/tst.ptrincop.d mode=0444
2027 file path=opt/SUNWdtrt/tst/common/types/tst.ptrops.d mode=0444
2028 file path=opt/SUNWdtrt/tst/common/types/tst.relenum.d mode=0444
2029 file path=opt/SUNWdtrt/tst/common/types/tst.relstring.d mode=0444
2030 file path=opt/SUNWdtrt/tst/common/types/tst.shiftops.d mode=0444
2031 file path=opt/SUNWdtrt/tst/common/types/tst.stringconstants.d mode=0444
2032 file path=opt/SUNWdtrt/tst/common/types/tst.struct.d mode=0444
2033 file path=opt/SUNWdtrt/tst/common/types/tst.typedef.d mode=0444
2034 file path=opt/SUNWdtrt/tst/common/types/tst.unaryop.d mode=0444
2035 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid.d mode=0444
2036 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid2.d mode=0444
2037 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidpid3.d mode=0444
2038 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype.ksh mode=0444
2039 file path=opt/SUNWdtrt/tst/common/uctf/err.invalidtype2.ksh mode=0444
2040 file path=opt/SUNWdtrt/tst/common/uctf/err.user64mode.ksh mode=0444
```



```
2041 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.exe mode=0555
2042 file path=opt/SUNWdtrt/tst/common/uctf/tst.aouttype.ksh mode=0444
2043 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.exe mode=0555
2044 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh mode=0444
2045 file path=opt/SUNWdtrt/tst/common/uctf/tst.chasestrings.ksh.out mode=0444
2046 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.exe mode=0555
2047 file path=opt/SUNWdtrt/tst/common/uctf/tst.libtype.ksh mode=0444
2048 file path=opt/SUNWdtrt/tst/common/uctf/tst.linkmap.ksh mode=0444
2049 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprint.ksh mode=0444
2050 file path=opt/SUNWdtrt/tst/common/uctf/tst.pidprinttarg.ksh mode=0444
2051 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.exe mode=0555
2052 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh mode=0444
2053 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtype.ksh.out mode=0444
2054 file path=opt/SUNWdtrt/tst/common/uctf/tst.printtypetarg.ksh mode=0444
2055 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh mode=0444
2056 file path=opt/SUNWdtrt/tst/common/uctf/tst.userlandkey.ksh.out mode=0444
2057 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh mode=0444
2058 file path=opt/SUNWdtrt/tst/common/uctf/tst.userstrings.ksh.out mode=0444
2059 file path=opt/SUNWdtrt/tst/common/union/err.D_ADDROF_VAR.UnionPointer.d \
2060     mode=0444
2061 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon.d \
2062     mode=0444
2063 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_COMBO.UnionWithoutColon1.d \
2064     mode=0444
2065 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.circular.d \
2066     mode=0444
2067 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.order.d \
2068     mode=0444
2069 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.recursive.d \
2070     mode=0444
2071 file path=opt/SUNWdtrt/tst/common/union/err.D_DECL_INCOMPLETE.simple.d \
2072     mode=0444
2073 file path=opt/SUNWdtrt/tst/common/union/err.D_PROTO_ARG.DupUnionAssoc.d \
2074     mode=0444
2075 file path=opt/SUNWdtrt/tst/common/union/tst.UnionAssoc.d mode=0444
2076 file path=opt/SUNWdtrt/tst/common/union/tst.UnionDataTypes.d mode=0444
2077 file path=opt/SUNWdtrt/tst/common/union/tst.UnionInside.d mode=0444
2078 file path=opt/SUNWdtrt/tst/common/usdt/tst.andpid.ksh mode=0444
2079 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.d mode=0444
2080 file path=opt/SUNWdtrt/tst/common/usdt/tst.argmap.exe mode=0555
2081 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.d mode=0444
2082 file path=opt/SUNWdtrt/tst/common/usdt/tst.args.exe mode=0555
2083 file path=opt/SUNWdtrt/tst/common/usdt/tst.badguess.ksh mode=0444
2084 file path=opt/SUNWdtrt/tst/common/usdt/tst.corruptenv.ksh mode=0444
2085 file path=opt/SUNWdtrt/tst/common/usdt/tst.deadstub.ksh mode=0444
2086 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh mode=0444
2087 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose1.ksh.out mode=0444
2088 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh mode=0444
2089 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose2.ksh.out mode=0444
2090 file path=opt/SUNWdtrt/tst/common/usdt/tst.dlclose3.ksh mode=0444
2091 file path=opt/SUNWdtrt/tst/common/usdt/tst.eliminate.ksh mode=0444
2092 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh mode=0444
2093 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled.ksh.out mode=0444
2094 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh mode=0444
2095 file path=opt/SUNWdtrt/tst/common/usdt/tst.enabled2.ksh.out mode=0444
2096 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh mode=0444
2097 file path=opt/SUNWdtrt/tst/common/usdt/tst.entryreturn.ksh.out mode=0444
2098 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh mode=0444
2099 file path=opt/SUNWdtrt/tst/common/usdt/tst.fork.ksh.out mode=0444
2100 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.exe mode=0555
2101 file path=opt/SUNWdtrt/tst/common/usdt/tst.forker.ksh mode=0444
2102 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess32.ksh mode=0444
2103 file path=opt/SUNWdtrt/tst/common/usdt/tst.guess64.ksh mode=0444
2104 file path=opt/SUNWdtrt/tst/common/usdt/tst.header.ksh mode=0444
2105 file path=opt/SUNWdtrt/tst/common/usdt/tst.include.ksh mode=0444
2106 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe.exe mode=0555
```

```
2107 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe1.ksh mode=0444
2108 file path=opt/SUNWdtrt/tst/common/usdt/tst.lazyprobe2.ksh mode=0444
2109 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkpriv.ksh mode=0444
2110 file path=opt/SUNWdtrt/tst/common/usdt/tst.linkunpriv.ksh mode=0444
2111 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh mode=0444
2112 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiple.ksh.out mode=0444
2113 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh mode=0444
2114 file path=opt/SUNWdtrt/tst/common/usdt/tst.multiprov.ksh.out mode=0444
2115 file path=opt/SUNWdtrt/tst/common/usdt/tst.nodtrace.ksh mode=0444
2116 file path=opt/SUNWdtrt/tst/common/usdt/tst.noprobes.ksh mode=0444
2117 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreap.ksh mode=0444
2118 file path=opt/SUNWdtrt/tst/common/usdt/tst.noreapring.ksh mode=0444
2119 file path=opt/SUNWdtrt/tst/common/usdt/tst.onlyenabled.ksh mode=0444
2120 file path=opt/SUNWdtrt/tst/common/usdt/tst.reap.ksh mode=0444
2121 file path=opt/SUNWdtrt/tst/common/usdt/tst.reeval.ksh mode=0444
2122 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh mode=0444
2123 file path=opt/SUNWdtrt/tst/common/usdt/tst.sameprovmulti.ksh.out mode=0444
2124 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh mode=0444
2125 file path=opt/SUNWdtrt/tst/common/usdt/tst.static.ksh.out mode=0444
2126 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh mode=0444
2127 file path=opt/SUNWdtrt/tst/common/usdt/tst.static2.ksh.out mode=0444
2128 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh mode=0444
2129 file path=opt/SUNWdtrt/tst/common/usdt/tst.user.ksh.out mode=0444
2130 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.d mode=0444
2131 file path=opt/SUNWdtrt/tst/common/ustack/tst.bigstack.exe mode=0555
2132 file path=opt/SUNWdtrt/tst/common/ustack/tst.depth.ksh mode=0444
2133 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.exe mode=0555
2134 file path=opt/SUNWdtrt/tst/common/ustack/tst.spin.ksh mode=0444
2135 file path=opt/SUNWdtrt/tst/common/vars/tst.gid.d mode=0444
2136 file path=opt/SUNWdtrt/tst/common/vars/tst.nullassign.d mode=0444
2137 file path=opt/SUNWdtrt/tst/common/vars/tst.ppid.d mode=0444
2138 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh mode=0444
2139 file path=opt/SUNWdtrt/tst/common/vars/tst.ucaller.ksh.out mode=0444
2140 file path=opt/SUNWdtrt/tst/common/vars/tst.uid.d mode=0444
2141 file path=opt/SUNWdtrt/tst/common/vars/tst.walltimestamp.d mode=0444
2142 file path=opt/SUNWdtrt/tst/common/version/tst.1.0.d mode=0444
2143 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.basic.ksh mode=0444
2144 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.hvmenable.ksh mode=0444
2145 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.memenable.ksh mode=0444
2146 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedargs.ksh mode=0444
2147 $(i386_ONLY)file path=opt/SUNWdtrt/tst/i86xpv/xdt/tst.schedenable.ksh \
2148     mode=0444
2149 legacy pkg=SUNWdtrt category=internal \
2150     desc="DTrace Test Suite Internal Distribution" \
2151     hotline="Contact the DTrace discussion forum" name="DTrace Test Suite"
2152 license cr_Sun license=cr_Sun
2153 license lic_CDDL license=lic_CDDL
2154 depend fmri=runtime/java type=require
2155 depend fmri=runtime/java/runtime64 type=require
```

```
**********************************************************
   82045 Mon Oct 15 13:28:39 2018
new/usr/src/pkg/manifests/system-library.man3c.inc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2018 Nexenta Systems, Inc.
  15 # Copyright 2013 OmniTI Computer Consulting, Inc. All rights reserved.
  16 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  17 # Copyright 2018 Jason King
  18 # Copyright 2018, Joyent, Inc.
  19 #

  21 file path=usr/share/man/man3c/__fbufsize.3c
  22 file path=usr/share/man/man3c/_longjmp.3c
  23 file path=usr/share/man/man3c/_stack_grow.3c
  24 file path=usr/share/man/man3c/a64l.3c
  25 file path=usr/share/man/man3c/abort.3c
  26 file path=usr/share/man/man3c/abs.3c
  27 file path=usr/share/man/man3c/addsev.3c
  28 file path=usr/share/man/man3c/addseverity.3c
  29 file path=usr/share/man/man3c/aio_cancel.3c
  30 file path=usr/share/man/man3c/aio_error.3c
  31 file path=usr/share/man/man3c/aio_fsync.3c
  32 file path=usr/share/man/man3c/aio_read.3c
  33 file path=usr/share/man/man3c/aio_return.3c
  34 file path=usr/share/man/man3c/aio_suspend.3c
  35 file path=usr/share/man/man3c/aio_waitn.3c
  36 file path=usr/share/man/man3c/aio_write.3c
  37 file path=usr/share/man/man3c/aiocancel.3c
  38 file path=usr/share/man/man3c/aioread.3c
  39 file path=usr/share/man/man3c/aiowait.3c
  40 file path=usr/share/man/man3c/aligned_alloc.3c
  41 file path=usr/share/man/man3c/arc4random.3c
  42 file path=usr/share/man/man3c/assert.3c
  43 file path=usr/share/man/man3c/atexit.3c
  44 file path=usr/share/man/man3c/atomic_add.3c
  45 file path=usr/share/man/man3c/atomic_and.3c
  46 file path=usr/share/man/man3c/atomic_bits.3c
  47 file path=usr/share/man/man3c/atomic_cas.3c
  48 file path=usr/share/man/man3c/atomic_dec.3c
  49 file path=usr/share/man/man3c/atomic_inc.3c
  50 file path=usr/share/man/man3c/atomic_ops.3c
  51 file path=usr/share/man/man3c/atomic_or.3c
  52 file path=usr/share/man/man3c/atomic_swap.3c
  53 file path=usr/share/man/man3c/attropen.3c
  54 file path=usr/share/man/man3c/basename.3c
  55 file path=usr/share/man/man3c/bsd_signal.3c
  56 file path=usr/share/man/man3c/bsearch.3c
  57 file path=usr/share/man/man3c/bstring.3c
  58 file path=usr/share/man/man3c/btowc.3c
  59 file path=usr/share/man/man3c/byteorder.3c
  60 file path=usr/share/man/man3c/call_once.3c
```



```
  61 file path=usr/share/man/man3c/catgets.3c
  62 file path=usr/share/man/man3c/catopen.3c
  63 file path=usr/share/man/man3c/cfgetispeed.3c
  64 file path=usr/share/man/man3c/cfsetispeed.3c
  65 file path=usr/share/man/man3c/clearenv.3c
  66 file path=usr/share/man/man3c/clock.3c
  67 file path=usr/share/man/man3c/clock_nanosleep.3c
  68 file path=usr/share/man/man3c/clock_settime.3c
  69 file path=usr/share/man/man3c/closedir.3c
  70 file path=usr/share/man/man3c/closefrom.3c
  71 file path=usr/share/man/man3c/cnd.3c
  72 file path=usr/share/man/man3c/cond_init.3c
  73 file path=usr/share/man/man3c/confstr.3c
  74 file path=usr/share/man/man3c/crypt.3c
  75 file path=usr/share/man/man3c/crypt_genhash_impl.3c
  76 file path=usr/share/man/man3c/crypt_gensalt.3c
  77 file path=usr/share/man/man3c/crypt_gensalt_impl.3c
  78 file path=usr/share/man/man3c/cset.3c
  79 file path=usr/share/man/man3c/ctermid.3c
  80 file path=usr/share/man/man3c/ctime.3c
  81 file path=usr/share/man/man3c/ctype.3c
  82 file path=usr/share/man/man3c/cuserid.3c
  83 file path=usr/share/man/man3c/daemon.3c
  84 file path=usr/share/man/man3c/decimal_to_floating.3c
  85 file path=usr/share/man/man3c/difftime.3c
  86 file path=usr/share/man/man3c/directio.3c
  87 file path=usr/share/man/man3c/dirfd.3c
  88 file path=usr/share/man/man3c/dirname.3c
  89 file path=usr/share/man/man3c/div.3c
  90 file path=usr/share/man/man3c/dladdr.3c
  91 file path=usr/share/man/man3c/dlclose.3c
  92 file path=usr/share/man/man3c/dldump.3c
  93 file path=usr/share/man/man3c/dlerror.3c
  94 file path=usr/share/man/man3c/dlinfo.3c
  95 file path=usr/share/man/man3c/dlopen.3c
  96 file path=usr/share/man/man3c/dlsym.3c
  97 file path=usr/share/man/man3c/door_bind.3c
  98 file path=usr/share/man/man3c/door_call.3c
  99 file path=usr/share/man/man3c/door_create.3c
 100 file path=usr/share/man/man3c/door_cred.3c
 101 file path=usr/share/man/man3c/door_getparam.3c
 102 file path=usr/share/man/man3c/door_info.3c
 103 file path=usr/share/man/man3c/door_return.3c
 104 file path=usr/share/man/man3c/door_revoke.3c
 105 file path=usr/share/man/man3c/door_server_create.3c
 106 file path=usr/share/man/man3c/door_ucred.3c
 107 file path=usr/share/man/man3c/drand48.3c
 108 file path=usr/share/man/man3c/dup2.3c
 109 file path=usr/share/man/man3c/econvert.3c
 110 file path=usr/share/man/man3c/ecvt.3c
 111 file path=usr/share/man/man3c/enable_extended_FILE_stdio.3c
 112 file path=usr/share/man/man3c/encrypt.3c
 113 file path=usr/share/man/man3c/end.3c
 114 file path=usr/share/man/man3c/endian.3c
 115 file path=usr/share/man/man3c/epoll_create.3c
 116 file path=usr/share/man/man3c/epoll_ctl.3c
 117 file path=usr/share/man/man3c/epoll_wait.3c
 118 file path=usr/share/man/man3c/err.3c
 119 file path=usr/share/man/man3c/euclen.3c
 120 file path=usr/share/man/man3c/eventfd.3c
 121 file path=usr/share/man/man3c/exit.3c
 122 file path=usr/share/man/man3c/fattach.3c
 123 file path=usr/share/man/man3c/fclose.3c
 124 file path=usr/share/man/man3c/fcloseall.3c
 125 file path=usr/share/man/man3c/fdatasync.3c
 126 file path=usr/share/man/man3c/fdetach.3c
```

```
 127 file path=usr/share/man/man3c/fdopen.3c
 128 file path=usr/share/man/man3c/ferror.3c
 129 file path=usr/share/man/man3c/fflush.3c
 130 file path=usr/share/man/man3c/ffs.3c
 131 file path=usr/share/man/man3c/fgetattr.3c
 132 file path=usr/share/man/man3c/fgetc.3c
 133 file path=usr/share/man/man3c/fgetpos.3c
 134 file path=usr/share/man/man3c/fgetwc.3c
 135 file path=usr/share/man/man3c/floating_to_decimal.3c
 136 file path=usr/share/man/man3c/flock.3c
 137 file path=usr/share/man/man3c/flockfile.3c
 138 file path=usr/share/man/man3c/fmtmsg.3c
 139 file path=usr/share/man/man3c/fnmatch.3c
 140 file path=usr/share/man/man3c/fopen.3c
 141 file path=usr/share/man/man3c/fpgetround.3c
 142 file path=usr/share/man/man3c/fputc.3c
 143 file path=usr/share/man/man3c/fputwc.3c
 144 file path=usr/share/man/man3c/fputws.3c
 145 file path=usr/share/man/man3c/fread.3c
 146 file path=usr/share/man/man3c/freopen.3c
 147 file path=usr/share/man/man3c/fseek.3c
 148 file path=usr/share/man/man3c/fsetpos.3c
 149 file path=usr/share/man/man3c/fsync.3c
 150 file path=usr/share/man/man3c/ftell.3c
 151 file path=usr/share/man/man3c/ftime.3c
 152 file path=usr/share/man/man3c/ftok.3c
 153 file path=usr/share/man/man3c/fts.3c
 154 file path=usr/share/man/man3c/ftw.3c
 155 file path=usr/share/man/man3c/fwide.3c
 156 file path=usr/share/man/man3c/fwprintf.3c
 157 file path=usr/share/man/man3c/fwrite.3c
 158 file path=usr/share/man/man3c/fwscanf.3c
 159 file path=usr/share/man/man3c/get_nprocs.3c
 160 file path=usr/share/man/man3c/getcpuid.3c
 161 file path=usr/share/man/man3c/getcwd.3c
 162 file path=usr/share/man/man3c/getdate.3c
 163 file path=usr/share/man/man3c/getdtablesize.3c
 164 file path=usr/share/man/man3c/getentropy.3c
 165 file path=usr/share/man/man3c/getenv.3c
 166 file path=usr/share/man/man3c/getexecname.3c
 167 file path=usr/share/man/man3c/getgrnam.3c
 168 file path=usr/share/man/man3c/gethostid.3c
 169 file path=usr/share/man/man3c/gethostname.3c
 170 file path=usr/share/man/man3c/gethrtime.3c
 171 file path=usr/share/man/man3c/getline.3c
 172 file path=usr/share/man/man3c/getloadavg.3c
 173 file path=usr/share/man/man3c/getlogin.3c
 174 file path=usr/share/man/man3c/getmntent.3c
 175 file path=usr/share/man/man3c/getnetgrent.3c
 176 file path=usr/share/man/man3c/getopt.3c
 177 file path=usr/share/man/man3c/getopt_long.3c
 178 file path=usr/share/man/man3c/getpagesize.3c
 179 file path=usr/share/man/man3c/getpagesizes.3c
 180 file path=usr/share/man/man3c/getpass.3c
 181 file path=usr/share/man/man3c/getpeerucred.3c
 182 file path=usr/share/man/man3c/getpriority.3c
 183 file path=usr/share/man/man3c/getprogname.3c
 184 file path=usr/share/man/man3c/getpw.3c
 185 file path=usr/share/man/man3c/getpwnam.3c
 186 file path=usr/share/man/man3c/getrusage.3c
 187 file path=usr/share/man/man3c/gets.3c
 188 file path=usr/share/man/man3c/getspnam.3c
 189 file path=usr/share/man/man3c/getsubopt.3c
 190 file path=usr/share/man/man3c/gettext.3c
 191 file path=usr/share/man/man3c/gettimeofday.3c
 192 file path=usr/share/man/man3c/gettxt.3c
```



```
 193 file path=usr/share/man/man3c/getusershell.3c
 194 file path=usr/share/man/man3c/getutent.3c
 195 file path=usr/share/man/man3c/getutxent.3c
 196 file path=usr/share/man/man3c/getvfsent.3c
 197 file path=usr/share/man/man3c/getwc.3c
 198 file path=usr/share/man/man3c/getwchar.3c
 199 file path=usr/share/man/man3c/getwd.3c
 200 file path=usr/share/man/man3c/getwidth.3c
 201 file path=usr/share/man/man3c/getws.3c
 202 file path=usr/share/man/man3c/getzoneid.3c
 203 file path=usr/share/man/man3c/glob.3c
 204 file path=usr/share/man/man3c/grantpt.3c
 205 file path=usr/share/man/man3c/hsearch.3c
 206 file path=usr/share/man/man3c/iconv.3c
 207 file path=usr/share/man/man3c/iconv_close.3c
 208 file path=usr/share/man/man3c/iconv_open.3c
 209 file path=usr/share/man/man3c/imaxabs.3c
 210 file path=usr/share/man/man3c/imaxdiv.3c
 211 file path=usr/share/man/man3c/index.3c
 212 file path=usr/share/man/man3c/inet.3c
 213 file path=usr/share/man/man3c/initgroups.3c
 214 file path=usr/share/man/man3c/insque.3c
 215 file path=usr/share/man/man3c/is_system_labeled.3c
 216 file path=usr/share/man/man3c/isaexec.3c
 217 file path=usr/share/man/man3c/isastream.3c
 218 file path=usr/share/man/man3c/isatty.3c
 219 file path=usr/share/man/man3c/isnand.3c
 220 file path=usr/share/man/man3c/iswalpha.3c
 221 file path=usr/share/man/man3c/iswctype.3c
 222 file path=usr/share/man/man3c/killpg.3c
 223 file path=usr/share/man/man3c/lckpwdf.3c
 224 file path=usr/share/man/man3c/lfmt.3c
 225 file path=usr/share/man/man3c/lio_listio.3c
 226 file path=usr/share/man/man3c/localeconv.3c
 227 file path=usr/share/man/man3c/lockf.3c
 228 file path=usr/share/man/man3c/lsearch.3c
 229 file path=usr/share/man/man3c/madvise.3c
 230 file path=usr/share/man/man3c/makecontext.3c
 231 file path=usr/share/man/man3c/makedev.3c
 232 file path=usr/share/man/man3c/malloc.3c
 233 file path=usr/share/man/man3c/mblen.3c
 234 file path=usr/share/man/man3c/mbrlen.3c
 235 file path=usr/share/man/man3c/mbrtowc.3c
 236 file path=usr/share/man/man3c/mbsinit.3c
 237 file path=usr/share/man/man3c/mbsrtowcs.3c
 238 file path=usr/share/man/man3c/mbstowcs.3c
 239 file path=usr/share/man/man3c/mbtowc.3c
 240 file path=usr/share/man/man3c/membar_ops.3c
 241 file path=usr/share/man/man3c/memory.3c
 242 file path=usr/share/man/man3c/memset_s.3c
 243 file path=usr/share/man/man3c/mkfifo.3c
 244 file path=usr/share/man/man3c/mkstemp.3c
 245 file path=usr/share/man/man3c/mktemp.3c
 246 file path=usr/share/man/man3c/mktime.3c
 247 file path=usr/share/man/man3c/mlock.3c
 248 file path=usr/share/man/man3c/mlockall.3c
 249 file path=usr/share/man/man3c/monitor.3c
 250 file path=usr/share/man/man3c/mq_close.3c
 251 file path=usr/share/man/man3c/mq_getattr.3c
 252 file path=usr/share/man/man3c/mq_notify.3c
 253 file path=usr/share/man/man3c/mq_open.3c
 254 file path=usr/share/man/man3c/mq_receive.3c
 255 file path=usr/share/man/man3c/mq_send.3c
 256 file path=usr/share/man/man3c/mq_setattr.3c
 257 file path=usr/share/man/man3c/mq_unlink.3c
 258 file path=usr/share/man/man3c/msync.3c
```

```
259 file path=usr/share/man/man3c/mtx.3c
260 file path=usr/share/man/man3c/mutex_init.3c
261 file path=usr/share/man/man3c/nanosleep.3c
262 file path=usr/share/man/man3c/ndbm.3c
263 file path=usr/share/man/man3c/newlocale.3c
264 file path=usr/share/man/man3c/nl_langinfo.3c
265 file path=usr/share/man/man3c/offsetof.3c
266 file path=usr/share/man/man3c/opendir.3c
267 file path=usr/share/man/man3c/perror.3c
268 file path=usr/share/man/man3c/pfmt.3c
269 file path=usr/share/man/man3c/plock.3c
270 file path=usr/share/man/man3c/popen.3c
271 file path=usr/share/man/man3c/port_alert.3c
272 file path=usr/share/man/man3c/port_associate.3c
273 file path=usr/share/man/man3c/port_create.3c
274 file path=usr/share/man/man3c/port_get.3c
275 file path=usr/share/man/man3c/port_send.3c
276 file path=usr/share/man/man3c/posix_fadvise.3c
277 file path=usr/share/man/man3c/posix_fallocate.3c
278 file path=usr/share/man/man3c/posix_madvise.3c
279 file path=usr/share/man/man3c/posix_memalign.3c
280 file path=usr/share/man/man3c/posix_openpt.3c
281 file path=usr/share/man/man3c/posix_spawn.3c
282 file path=usr/share/man/man3c/posix_spawn_file_actions_addclose.3c
283 file path=usr/share/man/man3c/posix_spawn_file_actions_addclosefrom_np.3c
284 file path=usr/share/man/man3c/posix_spawn_file_actions_adddup2.3c
285 file path=usr/share/man/man3c/posix_spawn_file_actions_destroy.3c
286 file path=usr/share/man/man3c/posix_spawn_pipe_np.3c
287 file path=usr/share/man/man3c/posix_spawnattr_destroy.3c
288 file path=usr/share/man/man3c/posix_spawnattr_getflags.3c
289 file path=usr/share/man/man3c/posix_spawnattr_getpgroup.3c
290 file path=usr/share/man/man3c/posix_spawnattr_getschedparam.3c
291 file path=usr/share/man/man3c/posix_spawnattr_getschedpolicy.3c
292 file path=usr/share/man/man3c/posix_spawnattr_getsigdefault.3c
293 file path=usr/share/man/man3c/posix_spawnattr_getsigignore_np.3c
294 file path=usr/share/man/man3c/posix_spawnattr_getsigmask.3c
295 file path=usr/share/man/man3c/printf.3c
296 file path=usr/share/man/man3c/priv_addset.3c
297 file path=usr/share/man/man3c/priv_set.3c
298 file path=usr/share/man/man3c/priv_str_to_set.3c
299 file path=usr/share/man/man3c/pset_getloadavg.3c
300 file path=usr/share/man/man3c/psignal.3c
301 file path=usr/share/man/man3c/pthread_atfork.3c
302 file path=usr/share/man/man3c/pthread_attr_get_np.3c
303 file path=usr/share/man/man3c/pthread_attr_getdetachstate.3c
304 file path=usr/share/man/man3c/pthread_attr_getguardsize.3c
305 file path=usr/share/man/man3c/pthread_attr_getinheritsched.3c
306 file path=usr/share/man/man3c/pthread_attr_getname_np.3c
307 file path=usr/share/man/man3c/pthread_attr_getschedparam.3c
308 file path=usr/share/man/man3c/pthread_attr_getschedpolicy.3c
309 file path=usr/share/man/man3c/pthread_attr_getscope.3c
310 file path=usr/share/man/man3c/pthread_attr_getstack.3c
311 file path=usr/share/man/man3c/pthread_attr_getstackaddr.3c
312 file path=usr/share/man/man3c/pthread_attr_getstacksize.3c
313 file path=usr/share/man/man3c/pthread_attr_init.3c
314 file path=usr/share/man/man3c/pthread_barrier_destroy.3c
315 file path=usr/share/man/man3c/pthread_barrier_wait.3c
316 file path=usr/share/man/man3c/pthread_barrierattr_destroy.3c
317 file path=usr/share/man/man3c/pthread_barrierattr_getpshared.3c
318 file path=usr/share/man/man3c/pthread_cancel.3c
319 file path=usr/share/man/man3c/pthread_cleanup_pop.3c
320 file path=usr/share/man/man3c/pthread_cleanup_push.3c
321 file path=usr/share/man/man3c/pthread_cond_init.3c
322 file path=usr/share/man/man3c/pthread_cond_signal.3c
323 file path=usr/share/man/man3c/pthread_cond_wait.3c
324 file path=usr/share/man/man3c/pthread_condattr_getclock.3c
```



```
325 file path=usr/share/man/man3c/pthread_condattr_getpshared.3c
326 file path=usr/share/man/man3c/pthread_condattr_init.3c
327 file path=usr/share/man/man3c/pthread_create.3c
328 file path=usr/share/man/man3c/pthread_detach.3c
329 file path=usr/share/man/man3c/pthread_equal.3c
330 file path=usr/share/man/man3c/pthread_exit.3c
331 file path=usr/share/man/man3c/pthread_getconcurrency.3c
332 file path=usr/share/man/man3c/pthread_getname_np.3c
333 file path=usr/share/man/man3c/pthread_getschedparam.3c
334 file path=usr/share/man/man3c/pthread_getspecific.3c
335 file path=usr/share/man/man3c/pthread_join.3c
336 file path=usr/share/man/man3c/pthread_key_create.3c
337 file path=usr/share/man/man3c/pthread_key_delete.3c
338 file path=usr/share/man/man3c/pthread_kill.3c
339 file path=usr/share/man/man3c/pthread_mutex_consistent.3c
340 file path=usr/share/man/man3c/pthread_mutex_getprioceiling.3c
341 file path=usr/share/man/man3c/pthread_mutex_init.3c
342 file path=usr/share/man/man3c/pthread_mutex_lock.3c
343 file path=usr/share/man/man3c/pthread_mutex_timedlock.3c
344 file path=usr/share/man/man3c/pthread_mutexattr_getprioceiling.3c
345 file path=usr/share/man/man3c/pthread_mutexattr_getprotocol.3c
346 file path=usr/share/man/man3c/pthread_mutexattr_getpshared.3c
347 file path=usr/share/man/man3c/pthread_mutexattr_getrobust.3c
348 file path=usr/share/man/man3c/pthread_mutexattr_gettype.3c
349 file path=usr/share/man/man3c/pthread_mutexattr_init.3c
350 file path=usr/share/man/man3c/pthread_once.3c
351 file path=usr/share/man/man3c/pthread_rwlock_init.3c
352 file path=usr/share/man/man3c/pthread_rwlock_rdlock.3c
353 file path=usr/share/man/man3c/pthread_rwlock_timedrdlock.3c
354 file path=usr/share/man/man3c/pthread_rwlock_timedwrlock.3c
355 file path=usr/share/man/man3c/pthread_rwlock_unlock.3c
356 file path=usr/share/man/man3c/pthread_rwlock_wrlock.3c
357 file path=usr/share/man/man3c/pthread_rwlockattr_getpshared.3c
358 file path=usr/share/man/man3c/pthread_rwlockattr_init.3c
359 file path=usr/share/man/man3c/pthread_self.3c
360 file path=usr/share/man/man3c/pthread_setcancelstate.3c
361 file path=usr/share/man/man3c/pthread_setcanceltype.3c
362 file path=usr/share/man/man3c/pthread_setschedprio.3c
363 file path=usr/share/man/man3c/pthread_sigmask.3c
364 file path=usr/share/man/man3c/pthread_spin_destroy.3c
365 file path=usr/share/man/man3c/pthread_spin_lock.3c
366 file path=usr/share/man/man3c/pthread_spin_unlock.3c
367 file path=usr/share/man/man3c/pthread_testcancel.3c
368 file path=usr/share/man/man3c/ptrace.3c
369 file path=usr/share/man/man3c/ptsname.3c
370 file path=usr/share/man/man3c/putenv.3c
371 file path=usr/share/man/man3c/putpwent.3c
372 file path=usr/share/man/man3c/puts.3c
373 file path=usr/share/man/man3c/putspent.3c
374 file path=usr/share/man/man3c/putws.3c
375 file path=usr/share/man/man3c/qsort.3c
376 file path=usr/share/man/man3c/quick_exit.3c
377 file path=usr/share/man/man3c/raise.3c
378 file path=usr/share/man/man3c/rand.3c
379 file path=usr/share/man/man3c/random.3c
380 file path=usr/share/man/man3c/rctl_walk.3c
381 file path=usr/share/man/man3c/rctlblk_set_value.3c
382 file path=usr/share/man/man3c/re_comp.3c
383 file path=usr/share/man/man3c/readdir.3c
384 file path=usr/share/man/man3c/realpath.3c
385 file path=usr/share/man/man3c/reboot.3c
386 file path=usr/share/man/man3c/regcmp.3c
387 file path=usr/share/man/man3c/regcomp.3c
388 file path=usr/share/man/man3c/remove.3c
389 file path=usr/share/man/man3c/rewind.3c
390 file path=usr/share/man/man3c/rewinddir.3c
```

```
391 file path=usr/share/man/man3c/rwlock.3c
392 file path=usr/share/man/man3c/scandir.3c
393 file path=usr/share/man/man3c/scanf.3c
394 file path=usr/share/man/man3c/sched_get_priority_max.3c
395 file path=usr/share/man/man3c/sched_getparam.3c
396 file path=usr/share/man/man3c/sched_getscheduler.3c
397 file path=usr/share/man/man3c/sched_rr_get_interval.3c
398 file path=usr/share/man/man3c/sched_setparam.3c
399 file path=usr/share/man/man3c/sched_setscheduler.3c
400 file path=usr/share/man/man3c/sched_yield.3c
401 file path=usr/share/man/man3c/schedctl_init.3c
402 file path=usr/share/man/man3c/seekdir.3c
403 file path=usr/share/man/man3c/select.3c
404 file path=usr/share/man/man3c/sem_close.3c
405 file path=usr/share/man/man3c/sem_destroy.3c
406 file path=usr/share/man/man3c/sem_getvalue.3c
407 file path=usr/share/man/man3c/sem_init.3c
408 file path=usr/share/man/man3c/sem_open.3c
409 file path=usr/share/man/man3c/sem_post.3c
410 file path=usr/share/man/man3c/sem_timedwait.3c
411 file path=usr/share/man/man3c/sem_unlink.3c
412 file path=usr/share/man/man3c/sem_wait.3c
413 file path=usr/share/man/man3c/semaphore.3c
414 file path=usr/share/man/man3c/set_constraint_handler_s.3c
415 file path=usr/share/man/man3c/setbuf.3c
416 file path=usr/share/man/man3c/setbuffer.3c
417 file path=usr/share/man/man3c/setcat.3c
418 file path=usr/share/man/man3c/setenv.3c
419 file path=usr/share/man/man3c/setjmp.3c
420 file path=usr/share/man/man3c/setkey.3c
421 file path=usr/share/man/man3c/setlabel.3c
422 file path=usr/share/man/man3c/setlocale.3c
423 file path=usr/share/man/man3c/shm_open.3c
424 file path=usr/share/man/man3c/shm_unlink.3c
425 file path=usr/share/man/man3c/sigfpe.3c
426 file path=usr/share/man/man3c/siginterrupt.3c
427 file path=usr/share/man/man3c/signal.3c
428 file path=usr/share/man/man3c/signalfd.3c
429 file path=usr/share/man/man3c/sigqueue.3c
430 file path=usr/share/man/man3c/sigsetops.3c
431 file path=usr/share/man/man3c/sigstack.3c
432 file path=usr/share/man/man3c/sigwaitinfo.3c
433 file path=usr/share/man/man3c/sleep.3c
434 file path=usr/share/man/man3c/smt_pause.3c
435 file path=usr/share/man/man3c/ssignal.3c
436 file path=usr/share/man/man3c/stack_getbounds.3c
437 file path=usr/share/man/man3c/stack_inbounds.3c
438 file path=usr/share/man/man3c/stack_setbounds.3c
439 file path=usr/share/man/man3c/stack_violation.3c
440 file path=usr/share/man/man3c/stdio.3c
441 file path=usr/share/man/man3c/str2sig.3c
442 file path=usr/share/man/man3c/strcoll.3c
443 file path=usr/share/man/man3c/strerror.3c
444 file path=usr/share/man/man3c/strfmon.3c
445 file path=usr/share/man/man3c/strftime.3c
446 file path=usr/share/man/man3c/string.3c
447 file path=usr/share/man/man3c/string_to_decimal.3c
448 file path=usr/share/man/man3c/strptime.3c
449 file path=usr/share/man/man3c/strsignal.3c
450 file path=usr/share/man/man3c/strtod.3c
451 file path=usr/share/man/man3c/strtoimax.3c
452 file path=usr/share/man/man3c/strtol.3c
453 file path=usr/share/man/man3c/strtonum.3c
454 file path=usr/share/man/man3c/strtoul.3c
455 file path=usr/share/man/man3c/strtows.3c
456 file path=usr/share/man/man3c/strxfrm.3c
```



```
457 file path=usr/share/man/man3c/swab.3c
458 file path=usr/share/man/man3c/sync_instruction_memory.3c
459 file path=usr/share/man/man3c/sysconf.3c
460 file path=usr/share/man/man3c/syslog.3c
461 file path=usr/share/man/man3c/system.3c
462 file path=usr/share/man/man3c/tcdrain.3c
463 file path=usr/share/man/man3c/tcflow.3c
464 file path=usr/share/man/man3c/tcflush.3c
465 file path=usr/share/man/man3c/tcgetattr.3c
466 file path=usr/share/man/man3c/tcgetpgrp.3c
467 file path=usr/share/man/man3c/tcgetsid.3c
468 file path=usr/share/man/man3c/tcsendbreak.3c
469 file path=usr/share/man/man3c/tcsetattr.3c
470 file path=usr/share/man/man3c/tcsetpgrp.3c
471 file path=usr/share/man/man3c/tell.3c
472 file path=usr/share/man/man3c/telldir.3c
473 file path=usr/share/man/man3c/termios.3c
474 file path=usr/share/man/man3c/thr_create.3c
475 file path=usr/share/man/man3c/thr_exit.3c
476 file path=usr/share/man/man3c/thr_getconcurrency.3c
477 file path=usr/share/man/man3c/thr_getname.3c
478 file path=usr/share/man/man3c/thr_getprio.3c
479 file path=usr/share/man/man3c/thr_join.3c
480 file path=usr/share/man/man3c/thr_keycreate.3c
481 file path=usr/share/man/man3c/thr_kill.3c
482 file path=usr/share/man/man3c/thr_main.3c
483 file path=usr/share/man/man3c/thr_min_stack.3c
484 file path=usr/share/man/man3c/thr_self.3c
485 file path=usr/share/man/man3c/thr_sigsetmask.3c
486 file path=usr/share/man/man3c/thr_stksegment.3c
487 file path=usr/share/man/man3c/thr_suspend.3c
488 file path=usr/share/man/man3c/thr_yield.3c
489 file path=usr/share/man/man3c/thrd_create.3c
490 file path=usr/share/man/man3c/thrd_current.3c
491 file path=usr/share/man/man3c/thrd_detach.3c
492 file path=usr/share/man/man3c/thrd_equal.3c
493 file path=usr/share/man/man3c/thrd_exit.3c
494 file path=usr/share/man/man3c/thrd_join.3c
495 file path=usr/share/man/man3c/thrd_yield.3c
496 file path=usr/share/man/man3c/timer_create.3c
497 file path=usr/share/man/man3c/timer_delete.3c
498 file path=usr/share/man/man3c/timer_settime.3c
499 file path=usr/share/man/man3c/timeradd.3c
500 file path=usr/share/man/man3c/timerfd_create.3c
501 file path=usr/share/man/man3c/timespec_get.3c
502 file path=usr/share/man/man3c/tmpfile.3c
503 file path=usr/share/man/man3c/tmpnam.3c
504 file path=usr/share/man/man3c/toascii.3c
505 file path=usr/share/man/man3c/tolower.3c
506 file path=usr/share/man/man3c/toupper.3c
507 file path=usr/share/man/man3c/towlower.3c
508 file path=usr/share/man/man3c/towupper.3c
509 file path=usr/share/man/man3c/truncate.3c
510 file path=usr/share/man/man3c/tsearch.3c
511 file path=usr/share/man/man3c/tss.3c
512 file path=usr/share/man/man3c/ttyname.3c
513 file path=usr/share/man/man3c/ttyslot.3c
514 file path=usr/share/man/man3c/u8_strcmp.3c
515 file path=usr/share/man/man3c/u8_textprep_str.3c
516 file path=usr/share/man/man3c/u8_validate.3c
517 file path=usr/share/man/man3c/ualarm.3c
518 file path=usr/share/man/man3c/uconv_u16tou32.3c
519 file path=usr/share/man/man3c/ucred.3c
520 file path=usr/share/man/man3c/ungetc.3c
521 file path=usr/share/man/man3c/ungetwc.3c
522 file path=usr/share/man/man3c/unlockpt.3c
```

```
523 file path=usr/share/man/man3c/unsetenv.3c
524 file path=usr/share/man/man3c/uselocale.3c
525 file path=usr/share/man/man3c/usleep.3c
526 file path=usr/share/man/man3c/vfwprintf.3c
527 file path=usr/share/man/man3c/vlfmt.3c
528 file path=usr/share/man/man3c/vpfmt.3c
529 file path=usr/share/man/man3c/vprintf.3c
530 file path=usr/share/man/man3c/vsyslog.3c
531 file path=usr/share/man/man3c/wait.3c
532 file path=usr/share/man/man3c/wait3.3c
533 file path=usr/share/man/man3c/waitpid.3c
534 file path=usr/share/man/man3c/walkcontext.3c
535 file path=usr/share/man/man3c/wcpcpy.3c
536 file path=usr/share/man/man3c/wcrtomb.3c
537 file path=usr/share/man/man3c/wcscasecmp.3c
538 file path=usr/share/man/man3c/wcscoll.3c
539 file path=usr/share/man/man3c/wcsdup.3c
540 file path=usr/share/man/man3c/wcsftime.3c
541 file path=usr/share/man/man3c/wcslen.3c
542 file path=usr/share/man/man3c/wcsrtombs.3c
543 file path=usr/share/man/man3c/wcsstr.3c
544 file path=usr/share/man/man3c/wcstod.3c
545 file path=usr/share/man/man3c/wcstoimax.3c
546 file path=usr/share/man/man3c/wcstol.3c
547 file path=usr/share/man/man3c/wcstombs.3c
548 file path=usr/share/man/man3c/wcstoul.3c
549 file path=usr/share/man/man3c/wcstring.3c
550 file path=usr/share/man/man3c/wcswidth.3c
551 file path=usr/share/man/man3c/wcsxfrm.3c
552 file path=usr/share/man/man3c/wctob.3c
553 file path=usr/share/man/man3c/wctomb.3c
554 file path=usr/share/man/man3c/wctrans.3c
555 file path=usr/share/man/man3c/wctype.3c
556 file path=usr/share/man/man3c/wcwidth.3c
557 file path=usr/share/man/man3c/wmemchr.3c
558 file path=usr/share/man/man3c/wmemcmp.3c
559 file path=usr/share/man/man3c/wmemcpy.3c
560 file path=usr/share/man/man3c/wmemmove.3c
561 file path=usr/share/man/man3c/wmemset.3c
562 file path=usr/share/man/man3c/wordexp.3c
563 file path=usr/share/man/man3c/wsprintf.3c
564 file path=usr/share/man/man3c/wsscanf.3c
565 file path=usr/share/man/man3c/wstring.3c
566 link path=usr/share/man/man3c/FD_CLR.3c target=select.3c
567 link path=usr/share/man/man3c/FD_ISSET.3c target=select.3c
568 link path=usr/share/man/man3c/FD_SET.3c target=select.3c
569 link path=usr/share/man/man3c/FD_ZERO.3c target=select.3c
570 link path=usr/share/man/man3c/__flbf.3c target=__fbufsize.3c
571 link path=usr/share/man/man3c/__fpending.3c target=__fbufsize.3c
572 link path=usr/share/man/man3c/__fpurge.3c target=__fbufsize.3c
573 link path=usr/share/man/man3c/__freadable.3c target=__fbufsize.3c
574 link path=usr/share/man/man3c/__freading.3c target=__fbufsize.3c
575 link path=usr/share/man/man3c/__fsetlocking.3c target=__fbufsize.3c
576 link path=usr/share/man/man3c/__fwritable.3c target=__fbufsize.3c
577 link path=usr/share/man/man3c/__fwriting.3c target=__fbufsize.3c
578 link path=usr/share/man/man3c/_edata.3c target=end.3c
579 link path=usr/share/man/man3c/_end.3c target=end.3c
580 link path=usr/share/man/man3c/_etext.3c target=end.3c
581 link path=usr/share/man/man3c/_exithandle.3c target=exit.3c
582 link path=usr/share/man/man3c/_flushlbf.3c target=__fbufsize.3c
583 link path=usr/share/man/man3c/_setjmp.3c target=_longjmp.3c
584 link path=usr/share/man/man3c/abort_handler_s.3c \
585     target=set_constraint_handler_s.3c
586 link path=usr/share/man/man3c/addrtosymstr.3c target=walkcontext.3c
587 link path=usr/share/man/man3c/aiowrite.3c target=aioread.3c
588 link path=usr/share/man/man3c/alloca.3c target=malloc.3c
```



```
589 link path=usr/share/man/man3c/alphasort.3c target=scandir.3c
590 link path=usr/share/man/man3c/arc4random_buf.3c target=arc4random.3c
591 link path=usr/share/man/man3c/arc4random_uniform.3c target=arc4random.3c
592 link path=usr/share/man/man3c/ascftime.3c target=strftime.3c
593 link path=usr/share/man/man3c/asctime.3c target=ctime.3c
594 link path=usr/share/man/man3c/asctime_r.3c target=ctime.3c
595 link path=usr/share/man/man3c/asprintf.3c target=printf.3c
596 link path=usr/share/man/man3c/at_quick_exit.3c target=quick_exit.3c
597 link path=usr/share/man/man3c/atof.3c target=strtod.3c
598 link path=usr/share/man/man3c/atoi.3c target=strtol.3c
599 link path=usr/share/man/man3c/atol.3c target=strtol.3c
600 link path=usr/share/man/man3c/atoll.3c target=strtol.3c
601 link path=usr/share/man/man3c/atomic_add_16.3c target=atomic_add.3c
602 link path=usr/share/man/man3c/atomic_add_16_nv.3c target=atomic_add.3c
603 link path=usr/share/man/man3c/atomic_add_32.3c target=atomic_add.3c
604 link path=usr/share/man/man3c/atomic_add_32_nv.3c target=atomic_add.3c
605 link path=usr/share/man/man3c/atomic_add_64.3c target=atomic_add.3c
606 link path=usr/share/man/man3c/atomic_add_64_nv.3c target=atomic_add.3c
607 link path=usr/share/man/man3c/atomic_add_8.3c target=atomic_add.3c
608 link path=usr/share/man/man3c/atomic_add_8_nv.3c target=atomic_add.3c
609 link path=usr/share/man/man3c/atomic_add_char.3c target=atomic_add.3c
610 link path=usr/share/man/man3c/atomic_add_char_nv.3c target=atomic_add.3c
611 link path=usr/share/man/man3c/atomic_add_int.3c target=atomic_add.3c
612 link path=usr/share/man/man3c/atomic_add_int_nv.3c target=atomic_add.3c
613 link path=usr/share/man/man3c/atomic_add_long.3c target=atomic_add.3c
614 link path=usr/share/man/man3c/atomic_add_long_nv.3c target=atomic_add.3c
615 link path=usr/share/man/man3c/atomic_add_ptr.3c target=atomic_add.3c
616 link path=usr/share/man/man3c/atomic_add_ptr_nv.3c target=atomic_add.3c
617 link path=usr/share/man/man3c/atomic_add_short.3c target=atomic_add.3c
618 link path=usr/share/man/man3c/atomic_add_short_nv.3c target=atomic_add.3c
619 link path=usr/share/man/man3c/atomic_and_16.3c target=atomic_and.3c
620 link path=usr/share/man/man3c/atomic_and_16_nv.3c target=atomic_and.3c
621 link path=usr/share/man/man3c/atomic_and_32.3c target=atomic_and.3c
622 link path=usr/share/man/man3c/atomic_and_32_nv.3c target=atomic_and.3c
623 link path=usr/share/man/man3c/atomic_and_64.3c target=atomic_and.3c
624 link path=usr/share/man/man3c/atomic_and_64_nv.3c target=atomic_and.3c
625 link path=usr/share/man/man3c/atomic_and_8.3c target=atomic_and.3c
626 link path=usr/share/man/man3c/atomic_and_8_nv.3c target=atomic_and.3c
627 link path=usr/share/man/man3c/atomic_and_uchar.3c target=atomic_and.3c
628 link path=usr/share/man/man3c/atomic_and_uchar_nv.3c target=atomic_and.3c
629 link path=usr/share/man/man3c/atomic_and_uint.3c target=atomic_and.3c
630 link path=usr/share/man/man3c/atomic_and_uint_nv.3c target=atomic_and.3c
631 link path=usr/share/man/man3c/atomic_and_ulong.3c target=atomic_and.3c
632 link path=usr/share/man/man3c/atomic_and_ulong_nv.3c target=atomic_and.3c
633 link path=usr/share/man/man3c/atomic_and_ushort.3c target=atomic_and.3c
634 link path=usr/share/man/man3c/atomic_and_ushort_nv.3c target=atomic_and.3c
635 link path=usr/share/man/man3c/atomic_cas_16.3c target=atomic_cas.3c
636 link path=usr/share/man/man3c/atomic_cas_32.3c target=atomic_cas.3c
637 link path=usr/share/man/man3c/atomic_cas_64.3c target=atomic_cas.3c
638 link path=usr/share/man/man3c/atomic_cas_8.3c target=atomic_cas.3c
639 link path=usr/share/man/man3c/atomic_cas_ptr.3c target=atomic_cas.3c
640 link path=usr/share/man/man3c/atomic_cas_uchar.3c target=atomic_cas.3c
641 link path=usr/share/man/man3c/atomic_cas_uint.3c target=atomic_cas.3c
642 link path=usr/share/man/man3c/atomic_cas_ulong.3c target=atomic_cas.3c
643 link path=usr/share/man/man3c/atomic_cas_ushort.3c target=atomic_cas.3c
644 link path=usr/share/man/man3c/atomic_clear_long_excl.3c target=atomic_bits.3c
645 link path=usr/share/man/man3c/atomic_dec_16.3c target=atomic_dec.3c
646 link path=usr/share/man/man3c/atomic_dec_16_nv.3c target=atomic_dec.3c
647 link path=usr/share/man/man3c/atomic_dec_32.3c target=atomic_dec.3c
648 link path=usr/share/man/man3c/atomic_dec_32_nv.3c target=atomic_dec.3c
649 link path=usr/share/man/man3c/atomic_dec_64.3c target=atomic_dec.3c
650 link path=usr/share/man/man3c/atomic_dec_64_nv.3c target=atomic_dec.3c
651 link path=usr/share/man/man3c/atomic_dec_8.3c target=atomic_dec.3c
652 link path=usr/share/man/man3c/atomic_dec_8_nv.3c target=atomic_dec.3c
653 link path=usr/share/man/man3c/atomic_dec_ptr.3c target=atomic_dec.3c
654 link path=usr/share/man/man3c/atomic_dec_ptr_nv.3c target=atomic_dec.3c
```

```
link path=usr/share/man/man3c/atomic_dec_uchar.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_uchar_nv.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_uint.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_uint_nv.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_ulong.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_ulong_nv.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_ushort.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_dec_ushort_nv.3c target=atomic_dec.3c
link path=usr/share/man/man3c/atomic_inc_16.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_16_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_32.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_32_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_64.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_64_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_8.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_8_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ptr.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ptr_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_uchar.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_uchar_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_uint.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_uint_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ulong.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ulong_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ushort.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_inc_ushort_nv.3c target=atomic_inc.3c
link path=usr/share/man/man3c/atomic_or_16.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_16_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_32.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_32_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_64.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_64_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_8.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_8_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_uchar.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_uchar_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_uint.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_uint_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_ulong.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_ulong_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_ushort.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_or_ushort_nv.3c target=atomic_or.3c
link path=usr/share/man/man3c/atomic_set_long_excl.3c target=atomic_bits.3c
link path=usr/share/man/man3c/atomic_swap_16.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_32.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_64.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_8.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_ptr.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_uchar.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_uint.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_ulong.3c target=atomic_swap.3c
link path=usr/share/man/man3c/atomic_swap_ushort.3c target=atomic_swap.3c
link path=usr/share/man/man3c/backtrace.3c target=walkcontext.3c
link path=usr/share/man/man3c/backtrace_symbols.3c target=walkcontext.3c
link path=usr/share/man/man3c/backtrace_symbols_fd.3c target=walkcontext.3c
link path=usr/share/man/man3c/bcmp.3c target=bstring.3c
link path=usr/share/man/man3c/bcopy.3c target=bstring.3c
link path=usr/share/man/man3c/be16toh.3c target=endian.3c
link path=usr/share/man/man3c/be32toh.3c target=endian.3c
link path=usr/share/man/man3c/be64toh.3c target=endian.3c
link path=usr/share/man/man3c/betoh16.3c target=endian.3c
link path=usr/share/man/man3c/betoh32.3c target=endian.3c
link path=usr/share/man/man3c/betoh64.3c target=endian.3c
link path=usr/share/man/man3c/bind_textdomain_codeset.3c target=gettext.3c
link path=usr/share/man/man3c/bindtextdomain.3c target=gettext.3c
link path=usr/share/man/man3c/btowc_l.3c target=btowc.3c
```



```
link path=usr/share/man/man3c/bzero.3c target=bstring.3c
link path=usr/share/man/man3c/calloc.3c target=malloc.3c
link path=usr/share/man/man3c/canonicalize_file_name.3c target=realpath.3c
link path=usr/share/man/man3c/catclose.3c target=catopen.3c
link path=usr/share/man/man3c/cfgetospeed.3c target=cfgetispeed.3c
link path=usr/share/man/man3c/cfsetospeed.3c target=cfsetispeed.3c
link path=usr/share/man/man3c/cftime.3c target=strftime.3c
link path=usr/share/man/man3c/clearerr.3c target=ferror.3c
link path=usr/share/man/man3c/clock_getres.3c target=clock_settime.3c
link path=usr/share/man/man3c/clock_gettime.3c target=clock_settime.3c
link path=usr/share/man/man3c/closelog.3c target=syslog.3c
link path=usr/share/man/man3c/cnd_broadcast.3c target=cnd.3c
link path=usr/share/man/man3c/cnd_destroy.3c target=cnd.3c
link path=usr/share/man/man3c/cnd_init.3c target=cnd.3c
link path=usr/share/man/man3c/cnd_signal.3c target=cnd.3c
link path=usr/share/man/man3c/cnd_timedwait.3c target=cnd.3c
link path=usr/share/man/man3c/cnd_wait.3c target=cnd.3c
link path=usr/share/man/man3c/cond_broadcast.3c target=cond_init.3c
link path=usr/share/man/man3c/cond_destroy.3c target=cond_init.3c
link path=usr/share/man/man3c/cond_reltimedwait.3c target=cond_init.3c
link path=usr/share/man/man3c/cond_signal.3c target=cond_init.3c
link path=usr/share/man/man3c/cond_timedwait.3c target=cond_init.3c
link path=usr/share/man/man3c/cond_wait.3c target=cond_init.3c
link path=usr/share/man/man3c/csetcol.3c target=cset.3c
link path=usr/share/man/man3c/csetlen.3c target=cset.3c
link path=usr/share/man/man3c/csetno.3c target=cset.3c
link path=usr/share/man/man3c/ctermid_r.3c target=ctermid.3c
link path=usr/share/man/man3c/ctime_r.3c target=ctime.3c
link path=usr/share/man/man3c/dbm_clearerr.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_close.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_delete.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_error.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_fetch.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_firstkey.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_nextkey.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_open.3c target=ndbm.3c
link path=usr/share/man/man3c/dbm_store.3c target=ndbm.3c
link path=usr/share/man/man3c/dcgettext.3c target=gettext.3c
link path=usr/share/man/man3c/dcngettext.3c target=gettext.3c
link path=usr/share/man/man3c/decimal_to_double.3c \
    target=decimal_to_floating.3c
link path=usr/share/man/man3c/decimal_to_extended.3c \
    target=decimal_to_floating.3c
link path=usr/share/man/man3c/decimal_to_quadruple.3c \
    target=decimal_to_floating.3c
link path=usr/share/man/man3c/decimal_to_single.3c \
    target=decimal_to_floating.3c
link path=usr/share/man/man3c/dgettext.3c target=gettext.3c
link path=usr/share/man/man3c/dladdr1.3c target=dladdr.3c
link path=usr/share/man/man3c/dlmopen.3c target=dlopen.3c
link path=usr/share/man/man3c/dngettext.3c target=gettext.3c
link path=usr/share/man/man3c/door_setparam.3c target=door_getparam.3c
link path=usr/share/man/man3c/door_unbind.3c target=door_bind.3c
link path=usr/share/man/man3c/double_to_decimal.3c \
    target=floating_to_decimal.3c
link path=usr/share/man/man3c/dup3.3c target=dup2.3c
link path=usr/share/man/man3c/duplocale.3c target=newlocale.3c
link path=usr/share/man/man3c/edata.3c target=end.3c
link path=usr/share/man/man3c/endgrent.3c target=getgrnam.3c
link path=usr/share/man/man3c/endnetgrent.3c target=getnetgrent.3c
link path=usr/share/man/man3c/endpwent.3c target=getpwnam.3c
link path=usr/share/man/man3c/endspent.3c target=getspnam.3c
link path=usr/share/man/man3c/endusershell.3c target=getusershell.3c
link path=usr/share/man/man3c/endutent.3c target=getutent.3c
link path=usr/share/man/man3c/endutxent.3c target=getutxent.3c
link path=usr/share/man/man3c/epoll_create1.3c target=epoll_create.3c
```

```
787 link path=usr/share/man/man3c/epoll_pwait.3c target=epoll_wait.3c
788 link path=usr/share/man/man3c/erand48.3c target=drand48.3c
789 link path=usr/share/man/man3c/errno.3c target=perror.3c
790 link path=usr/share/man/man3c/errx.3c target=err.3c
791 link path=usr/share/man/man3c/etext.3c target=end.3c
792 link path=usr/share/man/man3c/euccol.3c target=euclen.3c
793 link path=usr/share/man/man3c/eucscol.3c target=euclen.3c
794 link path=usr/share/man/man3c/explicit_bzero.3c target=bstring.3c
795 link path=usr/share/man/man3c/extended_to_decimal.3c \
796     target=floating_to_decimal.3c
797 link path=usr/share/man/man3c/fconvert.3c target=econvert.3c
798 link path=usr/share/man/man3c/fcvt.3c target=ecvt.3c
799 link path=usr/share/man/man3c/fdopendir.3c target=opendir.3c
800 link path=usr/share/man/man3c/fdwalk.3c target=closefrom.3c
801 link path=usr/share/man/man3c/feof.3c target=ferror.3c
802 link path=usr/share/man/man3c/ffsl.3c target=ffs.3c
803 link path=usr/share/man/man3c/ffsll.3c target=ffs.3c
804 link path=usr/share/man/man3c/fgetgrent.3c target=getgrnam.3c
805 link path=usr/share/man/man3c/fgetgrent_r.3c target=getgrnam.3c
806 link path=usr/share/man/man3c/fgetpwent.3c target=getpwnam.3c
807 link path=usr/share/man/man3c/fgetpwent_r.3c target=getpwnam.3c
808 link path=usr/share/man/man3c/fgets.3c target=gets.3c
809 link path=usr/share/man/man3c/fgetspent.3c target=getspnam.3c
810 link path=usr/share/man/man3c/fgetspent_r.3c target=getspnam.3c
811 link path=usr/share/man/man3c/fgetwc_l.3c target=fgetwc.3c
812 link path=usr/share/man/man3c/fgetws.3c target=getws.3c
813 link path=usr/share/man/man3c/file_to_decimal.3c target=string_to_decimal.3c
814 link path=usr/share/man/man3c/fileno.3c target=ferror.3c
815 link path=usr/share/man/man3c/finite.3c target=isnand.3c
816 link path=usr/share/man/man3c/fls.3c target=ffs.3c
817 link path=usr/share/man/man3c/flsl.3c target=ffs.3c
818 link path=usr/share/man/man3c/flsll.3c target=ffs.3c
819 link path=usr/share/man/man3c/fpclass.3c target=isnand.3c
820 link path=usr/share/man/man3c/fpgetmask.3c target=fpgetround.3c
821 link path=usr/share/man/man3c/fpgetsticky.3c target=fpgetround.3c
822 link path=usr/share/man/man3c/fprintf.3c target=printf.3c
823 link path=usr/share/man/man3c/fpsetmask.3c target=fpgetround.3c
824 link path=usr/share/man/man3c/fpsetround.3c target=fpgetround.3c
825 link path=usr/share/man/man3c/fpsetsticky.3c target=fpgetround.3c
826 link path=usr/share/man/man3c/fputs.3c target=puts.3c
827 link path=usr/share/man/man3c/free.3c target=malloc.3c
828 link path=usr/share/man/man3c/freelocale.3c target=newlocale.3c
829 link path=usr/share/man/man3c/freezero.3c target=malloc.3c
830 link path=usr/share/man/man3c/fscanf.3c target=scanf.3c
831 link path=usr/share/man/man3c/fseeko.3c target=fseek.3c
832 link path=usr/share/man/man3c/fsetattr.3c target=fgetattr.3c
833 link path=usr/share/man/man3c/ftello.3c target=ftell.3c
834 link path=usr/share/man/man3c/ftruncate.3c target=truncate.3c
835 link path=usr/share/man/man3c/ftrylockfile.3c target=flockfile.3c
836 link path=usr/share/man/man3c/fts_children.3c target=fts.3c
837 link path=usr/share/man/man3c/fts_close.3c target=fts.3c
838 link path=usr/share/man/man3c/fts_open.3c target=fts.3c
839 link path=usr/share/man/man3c/fts_read.3c target=fts.3c
840 link path=usr/share/man/man3c/fts_set.3c target=fts.3c
841 link path=usr/share/man/man3c/func_to_decimal.3c target=string_to_decimal.3c
842 link path=usr/share/man/man3c/funlockfile.3c target=flockfile.3c
843 link path=usr/share/man/man3c/gconvert.3c target=econvert.3c
844 link path=usr/share/man/man3c/gcvt.3c target=ecvt.3c
845 link path=usr/share/man/man3c/get_nprocs_conf.3c target=get_nprocs.3c
846 link path=usr/share/man/man3c/getattrat.3c target=fgetattr.3c
847 link path=usr/share/man/man3c/getc.3c target=fgetc.3c
848 link path=usr/share/man/man3c/getc_unlocked.3c target=fgetc.3c
849 link path=usr/share/man/man3c/getchar.3c target=fgetc.3c
850 link path=usr/share/man/man3c/getchar_unlocked.3c target=fgetc.3c
851 link path=usr/share/man/man3c/getdelim.3c target=getline.3c
852 link path=usr/share/man/man3c/getextmntent.3c target=getmntent.3c
```



```
853 link path=usr/share/man/man3c/getgrent.3c target=getgrnam.3c
854 link path=usr/share/man/man3c/getgrent_r.3c target=getgrnam.3c
855 link path=usr/share/man/man3c/getgrgid.3c target=getgrnam.3c
856 link path=usr/share/man/man3c/getgrgid_r.3c target=getgrnam.3c
857 link path=usr/share/man/man3c/getgrnam_r.3c target=getgrnam.3c
858 link path=usr/share/man/man3c/gethomelgroup.3c target=getcpuid.3c
859 link path=usr/share/man/man3c/gethrvtime.3c target=gethrtime.3c
860 link path=usr/share/man/man3c/getlogin_r.3c target=getlogin.3c
861 link path=usr/share/man/man3c/getmntany.3c target=getmntent.3c
862 link path=usr/share/man/man3c/getnetgrent_r.3c target=getnetgrent.3c
863 link path=usr/share/man/man3c/getopt_long_clip.3c target=getopt_long.3c
864 link path=usr/share/man/man3c/getopt_long_only.3c target=getopt_long.3c
865 link path=usr/share/man/man3c/getpassphrase.3c target=getpass.3c
866 link path=usr/share/man/man3c/getpwent.3c target=getpwnam.3c
867 link path=usr/share/man/man3c/getpwent_r.3c target=getpwnam.3c
868 link path=usr/share/man/man3c/getpwnam_r.3c target=getpwnam.3c
869 link path=usr/share/man/man3c/getpwuid.3c target=getpwnam.3c
870 link path=usr/share/man/man3c/getpwuid_r.3c target=getpwnam.3c
871 link path=usr/share/man/man3c/getspent.3c target=getspnam.3c
872 link path=usr/share/man/man3c/getspent_r.3c target=getspnam.3c
873 link path=usr/share/man/man3c/getspnam_r.3c target=getspnam.3c
874 link path=usr/share/man/man3c/getutid.3c target=getutent.3c
875 link path=usr/share/man/man3c/getutline.3c target=getutent.3c
876 link path=usr/share/man/man3c/getutmp.3c target=getutxent.3c
877 link path=usr/share/man/man3c/getutmpx.3c target=getutxent.3c
878 link path=usr/share/man/man3c/getutxid.3c target=getutxent.3c
879 link path=usr/share/man/man3c/getutxline.3c target=getutxent.3c
880 link path=usr/share/man/man3c/getvfsany.3c target=getvfsent.3c
881 link path=usr/share/man/man3c/getvfsfile.3c target=getvfsent.3c
882 link path=usr/share/man/man3c/getvfsspec.3c target=getvfsent.3c
883 link path=usr/share/man/man3c/getw.3c target=fgetc.3c
884 link path=usr/share/man/man3c/getwc_l.3c target=getwc.3c
885 link path=usr/share/man/man3c/getwchar_l.3c target=getwchar.3c
886 link path=usr/share/man/man3c/getzoneidbyname.3c target=getzoneid.3c
887 link path=usr/share/man/man3c/getzonenamebyid.3c target=getzoneid.3c
888 link path=usr/share/man/man3c/globfree.3c target=glob.3c
889 link path=usr/share/man/man3c/gmtime.3c target=ctime.3c
890 link path=usr/share/man/man3c/gmtime_r.3c target=ctime.3c
891 link path=usr/share/man/man3c/gsignal.3c target=ssignal.3c
892 link path=usr/share/man/man3c/hasmntopt.3c target=getmntent.3c
893 link path=usr/share/man/man3c/hcreate.3c target=hsearch.3c
894 link path=usr/share/man/man3c/hdestroy.3c target=hsearch.3c
895 link path=usr/share/man/man3c/htobe16.3c target=endian.3c
896 link path=usr/share/man/man3c/htobe32.3c target=endian.3c
897 link path=usr/share/man/man3c/htobe64.3c target=endian.3c
898 link path=usr/share/man/man3c/htole16.3c target=endian.3c
899 link path=usr/share/man/man3c/htole32.3c target=endian.3c
900 link path=usr/share/man/man3c/htole64.3c target=endian.3c
901 link path=usr/share/man/man3c/htonl.3c target=byteorder.3c
902 link path=usr/share/man/man3c/htonll.3c target=byteorder.3c
903 link path=usr/share/man/man3c/htons.3c target=byteorder.3c
904 link path=usr/share/man/man3c/ignore_handler_s.3c \
905     target=set_constraint_handler_s.3c
906 link path=usr/share/man/man3c/inet6.3c target=inet.3c
907 link path=usr/share/man/man3c/inet_addr.3c target=inet.3c
908 link path=usr/share/man/man3c/inet_aton.3c target=inet.3c
909 link path=usr/share/man/man3c/inet_lnaof.3c target=inet.3c
910 link path=usr/share/man/man3c/inet_makeaddr.3c target=inet.3c
911 link path=usr/share/man/man3c/inet_netof.3c target=inet.3c
912 link path=usr/share/man/man3c/inet_network.3c target=inet.3c
913 link path=usr/share/man/man3c/inet_ntoa.3c target=inet.3c
914 link path=usr/share/man/man3c/inet_ntop.3c target=inet.3c
915 link path=usr/share/man/man3c/inet_pton.3c target=inet.3c
916 link path=usr/share/man/man3c/initstate.3c target=random.3c
917 link path=usr/share/man/man3c/innetgr.3c target=getnetgrent.3c
918 link path=usr/share/man/man3c/isalnum.3c target=ctype.3c
```

```
 919 link path=usr/share/man/man3c/isalnum_l.3c target=ctype.3c
 920 link path=usr/share/man/man3c/isalpha.3c target=ctype.3c
 921 link path=usr/share/man/man3c/isalpha_l.3c target=ctype.3c
 922 link path=usr/share/man/man3c/isascii.3c target=ctype.3c
 923 link path=usr/share/man/man3c/isblank.3c target=ctype.3c
 924 link path=usr/share/man/man3c/isblank_l.3c target=ctype.3c
 925 link path=usr/share/man/man3c/iscntrl.3c target=ctype.3c
 926 link path=usr/share/man/man3c/iscntrl_l.3c target=ctype.3c
 927 link path=usr/share/man/man3c/isdigit.3c target=ctype.3c
 928 link path=usr/share/man/man3c/isdigit_l.3c target=ctype.3c
 929 link path=usr/share/man/man3c/isenglish.3c target=iswalpha.3c
 930 link path=usr/share/man/man3c/isgraph.3c target=ctype.3c
 931 link path=usr/share/man/man3c/isgraph_l.3c target=ctype.3c
 932 link path=usr/share/man/man3c/isideogram.3c target=iswalpha.3c
 933 link path=usr/share/man/man3c/islower.3c target=ctype.3c
 934 link path=usr/share/man/man3c/islower_l.3c target=ctype.3c
 935 link path=usr/share/man/man3c/isnanf.3c target=isnand.3c
 936 link path=usr/share/man/man3c/isnumber.3c target=iswalpha.3c
 937 link path=usr/share/man/man3c/isphonogram.3c target=iswalpha.3c
 938 link path=usr/share/man/man3c/isprint.3c target=ctype.3c
 939 link path=usr/share/man/man3c/isprint_l.3c target=ctype.3c
 940 link path=usr/share/man/man3c/ispunct.3c target=ctype.3c
 941 link path=usr/share/man/man3c/ispunct_l.3c target=ctype.3c
 942 link path=usr/share/man/man3c/isspace.3c target=ctype.3c
 943 link path=usr/share/man/man3c/isspace_l.3c target=ctype.3c
 944 link path=usr/share/man/man3c/isspecial.3c target=iswalpha.3c
 945 link path=usr/share/man/man3c/isupper.3c target=ctype.3c
 946 link path=usr/share/man/man3c/isupper_l.3c target=ctype.3c
 947 link path=usr/share/man/man3c/iswalnum.3c target=iswalpha.3c
 948 link path=usr/share/man/man3c/iswalnum_l.3c target=iswalpha.3c
 949 link path=usr/share/man/man3c/iswalpha_l.3c target=iswalpha.3c
 950 link path=usr/share/man/man3c/iswascii.3c target=iswalpha.3c
 951 link path=usr/share/man/man3c/iswblank.3c target=iswalpha.3c
 952 link path=usr/share/man/man3c/iswblank_l.3c target=iswalpha.3c
 953 link path=usr/share/man/man3c/iswcntrl.3c target=iswalpha.3c
 954 link path=usr/share/man/man3c/iswcntrl_l.3c target=iswalpha.3c
 955 link path=usr/share/man/man3c/iswctype_l.3c target=iswctype.3c
 956 link path=usr/share/man/man3c/iswdigit.3c target=iswalpha.3c
 957 link path=usr/share/man/man3c/iswdigit_l.3c target=iswalpha.3c
 958 link path=usr/share/man/man3c/iswgraph.3c target=iswalpha.3c
 959 link path=usr/share/man/man3c/iswgraph_l.3c target=iswalpha.3c
 960 link path=usr/share/man/man3c/iswhexnumber.3c target=iswalpha.3c
 961 link path=usr/share/man/man3c/iswhexnumber_l.3c target=iswalpha.3c
 962 link path=usr/share/man/man3c/iswideogram.3c target=iswalpha.3c
 963 link path=usr/share/man/man3c/iswideogram_l.3c target=iswalpha.3c
 964 link path=usr/share/man/man3c/iswlower.3c target=iswalpha.3c
 965 link path=usr/share/man/man3c/iswlower_l.3c target=iswalpha.3c
 966 link path=usr/share/man/man3c/iswnumber.3c target=iswalpha.3c
 967 link path=usr/share/man/man3c/iswnumber_l.3c target=iswalpha.3c
 968 link path=usr/share/man/man3c/iswphonogram.3c target=iswalpha.3c
 969 link path=usr/share/man/man3c/iswphonogram_l.3c target=iswalpha.3c
 970 link path=usr/share/man/man3c/iswprint.3c target=iswalpha.3c
 971 link path=usr/share/man/man3c/iswprint_l.3c target=iswalpha.3c
 972 link path=usr/share/man/man3c/iswpunct.3c target=iswalpha.3c
 973 link path=usr/share/man/man3c/iswpunct_l.3c target=iswalpha.3c
 974 link path=usr/share/man/man3c/iswspace.3c target=iswalpha.3c
 975 link path=usr/share/man/man3c/iswspace_l.3c target=iswalpha.3c
 976 link path=usr/share/man/man3c/iswspecial.3c target=iswalpha.3c
 977 link path=usr/share/man/man3c/iswspecial_l.3c target=iswalpha.3c
 978 link path=usr/share/man/man3c/iswupper.3c target=iswalpha.3c
 979 link path=usr/share/man/man3c/iswupper_l.3c target=iswalpha.3c
 980 link path=usr/share/man/man3c/iswxdigit.3c target=iswalpha.3c
 981 link path=usr/share/man/man3c/iswxdigit_l.3c target=iswalpha.3c
 982 link path=usr/share/man/man3c/isxdigit.3c target=ctype.3c
 983 link path=usr/share/man/man3c/isxdigit_l.3c target=ctype.3c
 984 link path=usr/share/man/man3c/jrand48.3c target=drand48.3c
```



```
 985 link path=usr/share/man/man3c/l64a.3c target=a64l.3c
 986 link path=usr/share/man/man3c/labs.3c target=abs.3c
 987 link path=usr/share/man/man3c/lcong48.3c target=drand48.3c
 988 link path=usr/share/man/man3c/ldiv.3c target=div.3c
 989 link path=usr/share/man/man3c/le16toh.3c target=endian.3c
 990 link path=usr/share/man/man3c/le32toh.3c target=endian.3c
 991 link path=usr/share/man/man3c/le64toh.3c target=endian.3c
 992 link path=usr/share/man/man3c/letoh16.3c target=endian.3c
 993 link path=usr/share/man/man3c/letoh32.3c target=endian.3c
 994 link path=usr/share/man/man3c/letoh64.3c target=endian.3c
 995 link path=usr/share/man/man3c/lfind.3c target=lsearch.3c
 996 link path=usr/share/man/man3c/llabs.3c target=abs.3c
 997 link path=usr/share/man/man3c/lldiv.3c target=div.3c
 998 link path=usr/share/man/man3c/lltostr.3c target=strtol.3c
 999 link path=usr/share/man/man3c/localtime.3c target=ctime.3c
1000 link path=usr/share/man/man3c/localtime_r.3c target=ctime.3c
1001 link path=usr/share/man/man3c/longjmp.3c target=setjmp.3c
1002 link path=usr/share/man/man3c/lrand48.3c target=drand48.3c
1003 link path=usr/share/man/man3c/major.3c target=makedev.3c
1004 link path=usr/share/man/man3c/mblen_l.3c target=mblen.3c
1005 link path=usr/share/man/man3c/mbrlen_l.3c target=mbrlen.3c
1006 link path=usr/share/man/man3c/mbrtowc_l.3c target=mbrtowc.3c
1007 link path=usr/share/man/man3c/mbsinit_l.3c target=mbsinit.3c
1008 link path=usr/share/man/man3c/mbsnrtowcs.3c target=mbsrtowcs.3c
1009 link path=usr/share/man/man3c/mbsnrtowcs_l.3c target=mbsrtowcs.3c
1010 link path=usr/share/man/man3c/mbsrtowcs_l.3c target=mbsrtowcs.3c
1011 link path=usr/share/man/man3c/mbstowcs_l.3c target=mbstowcs.3c
1012 link path=usr/share/man/man3c/mbtowc_l.3c target=mbtowc.3c
1013 link path=usr/share/man/man3c/memalign.3c target=malloc.3c
1014 link path=usr/share/man/man3c/membar_consumer.3c target=membar_ops.3c
1015 link path=usr/share/man/man3c/membar_enter.3c target=membar_ops.3c
1016 link path=usr/share/man/man3c/membar_exit.3c target=membar_ops.3c
1017 link path=usr/share/man/man3c/membar_producer.3c target=membar_ops.3c
1018 link path=usr/share/man/man3c/memccpy.3c target=memory.3c
1019 link path=usr/share/man/man3c/memchr.3c target=memory.3c
1020 link path=usr/share/man/man3c/memcmp.3c target=memory.3c
1021 link path=usr/share/man/man3c/memcpy.3c target=memory.3c
1022 link path=usr/share/man/man3c/memmem.3c target=memory.3c
1023 link path=usr/share/man/man3c/memmove.3c target=memory.3c
1024 link path=usr/share/man/man3c/memset.3c target=memory.3c
1025 link path=usr/share/man/man3c/minor.3c target=makedev.3c
1026 link path=usr/share/man/man3c/mkdtemp.3c target=mkstemp.3c
1027 link path=usr/share/man/man3c/mkfifoat.3c target=mkfifo.3c
1028 link path=usr/share/man/man3c/mkostemp.3c target=mkstemp.3c
1029 link path=usr/share/man/man3c/mkostemps.3c target=mkstemp.3c
1030 link path=usr/share/man/man3c/mkstemps.3c target=mkstemp.3c
1031 link path=usr/share/man/man3c/mq_reltimedreceive_np.3c target=mq_receive.3c
1032 link path=usr/share/man/man3c/mq_reltimedsend_np.3c target=mq_send.3c
1033 link path=usr/share/man/man3c/mq_timedreceive.3c target=mq_receive.3c
1034 link path=usr/share/man/man3c/mq_timedsend.3c target=mq_send.3c
1035 link path=usr/share/man/man3c/mrand48.3c target=drand48.3c
1036 link path=usr/share/man/man3c/mtx_destroy.3c target=mtx.3c
1037 link path=usr/share/man/man3c/mtx_init.3c target=mtx.3c
1038 link path=usr/share/man/man3c/mtx_lock.3c target=mtx.3c
1039 link path=usr/share/man/man3c/mtx_timedlock.3c target=mtx.3c
1040 link path=usr/share/man/man3c/mtx_trylock.3c target=mtx.3c
1041 link path=usr/share/man/man3c/mtx_unlock.3c target=mtx.3c
1042 link path=usr/share/man/man3c/munlock.3c target=mlock.3c
1043 link path=usr/share/man/man3c/munlockall.3c target=mlockall.3c
1044 link path=usr/share/man/man3c/mutex_consistent.3c target=mutex_init.3c
1045 link path=usr/share/man/man3c/mutex_destroy.3c target=mutex_init.3c
1046 link path=usr/share/man/man3c/mutex_lock.3c target=mutex_init.3c
1047 link path=usr/share/man/man3c/mutex_trylock.3c target=mutex_init.3c
1048 link path=usr/share/man/man3c/mutex_unlock.3c target=mutex_init.3c
1049 link path=usr/share/man/man3c/nftw.3c target=ftw.3c
1050 link path=usr/share/man/man3c/ngettext.3c target=gettext.3c
```

```
1051 link path=usr/share/man/man3c/nl_langinfo_l.3c target=nl_langinfo.3c
1052 link path=usr/share/man/man3c/nrand48.3c target=drand48.3c
1053 link path=usr/share/man/man3c/ntohl.3c target=byteorder.3c
1054 link path=usr/share/man/man3c/ntohll.3c target=byteorder.3c
1055 link path=usr/share/man/man3c/ntohs.3c target=byteorder.3c
1056 link path=usr/share/man/man3c/openlog.3c target=syslog.3c
1057 link path=usr/share/man/man3c/pclose.3c target=popen.3c
1058 link path=usr/share/man/man3c/port_dissociate.3c target=port_associate.3c
1059 link path=usr/share/man/man3c/port_getn.3c target=port_get.3c
1060 link path=usr/share/man/man3c/port_sendn.3c target=port_send.3c
1061 link path=usr/share/man/man3c/posix_spawn_file_actions_addopen.3c \
1062     target=posix_spawn_file_actions_addclose.3c
1063 link path=usr/share/man/man3c/posix_spawn_file_actions_init.3c \
1064     target=posix_spawn_file_actions_destroy.3c
1065 link path=usr/share/man/man3c/posix_spawnattr_init.3c \
1066     target=posix_spawnattr_destroy.3c
1067 link path=usr/share/man/man3c/posix_spawnattr_setflags.3c \
1068     target=posix_spawnattr_getflags.3c
1069 link path=usr/share/man/man3c/posix_spawnattr_setpgroup.3c \
1070     target=posix_spawnattr_getpgroup.3c
1071 link path=usr/share/man/man3c/posix_spawnattr_setschedparam.3c \
1072     target=posix_spawnattr_getschedparam.3c
1073 link path=usr/share/man/man3c/posix_spawnattr_setschedpolicy.3c \
1074     target=posix_spawnattr_getschedpolicy.3c
1075 link path=usr/share/man/man3c/posix_spawnattr_setsigdefault.3c \
1076     target=posix_spawnattr_getsigdefault.3c
1077 link path=usr/share/man/man3c/posix_spawnattr_setsigignore_np.3c \
1078     target=posix_spawnattr_getsigignore_np.3c
1079 link path=usr/share/man/man3c/posix_spawnattr_setsigmask.3c \
1080     target=posix_spawnattr_getsigmask.3c
1081 link path=usr/share/man/man3c/posix_spawnp.3c target=posix_spawn.3c
1082 link path=usr/share/man/man3c/printstack.3c target=walkcontext.3c
1083 link path=usr/share/man/man3c/priv_allocset.3c target=priv_addset.3c
1084 link path=usr/share/man/man3c/priv_basicset.3c target=priv_addset.3c
1085 link path=usr/share/man/man3c/priv_copyset.3c target=priv_addset.3c
1086 link path=usr/share/man/man3c/priv_delset.3c target=priv_addset.3c
1087 link path=usr/share/man/man3c/priv_emptyset.3c target=priv_addset.3c
1088 link path=usr/share/man/man3c/priv_fillset.3c target=priv_addset.3c
1089 link path=usr/share/man/man3c/priv_freeset.3c target=priv_addset.3c
1090 link path=usr/share/man/man3c/priv_getbyname.3c target=priv_str_to_set.3c
1091 link path=usr/share/man/man3c/priv_getbynum.3c target=priv_str_to_set.3c
1092 link path=usr/share/man/man3c/priv_getsetbyname.3c target=priv_str_to_set.3c
1093 link path=usr/share/man/man3c/priv_getsetbynum.3c target=priv_str_to_set.3c
1094 link path=usr/share/man/man3c/priv_gettext.3c target=priv_str_to_set.3c
1095 link path=usr/share/man/man3c/priv_ineffect.3c target=priv_set.3c
1096 link path=usr/share/man/man3c/priv_intersect.3c target=priv_addset.3c
1097 link path=usr/share/man/man3c/priv_inverse.3c target=priv_addset.3c
1098 link path=usr/share/man/man3c/priv_isemptyset.3c target=priv_addset.3c
1099 link path=usr/share/man/man3c/priv_isequalset.3c target=priv_addset.3c
1100 link path=usr/share/man/man3c/priv_isfullset.3c target=priv_addset.3c
1101 link path=usr/share/man/man3c/priv_ismember.3c target=priv_addset.3c
1102 link path=usr/share/man/man3c/priv_issubset.3c target=priv_addset.3c
1103 link path=usr/share/man/man3c/priv_set_to_str.3c target=priv_str_to_set.3c
1104 link path=usr/share/man/man3c/priv_union.3c target=priv_addset.3c
1105 link path=usr/share/man/man3c/pselect.3c target=select.3c
1106 link path=usr/share/man/man3c/psiginfo.3c target=psignal.3c
1107 link path=usr/share/man/man3c/pthread_attr_destroy.3c \
1108     target=pthread_attr_init.3c
1109 link path=usr/share/man/man3c/pthread_attr_setdetachstate.3c \
1110     target=pthread_attr_getdetachstate.3c
1111 link path=usr/share/man/man3c/pthread_attr_setguardsize.3c \
1112     target=pthread_attr_getguardsize.3c
1113 link path=usr/share/man/man3c/pthread_attr_setinheritsched.3c \
1114     target=pthread_attr_getinheritsched.3c
1115 link path=usr/share/man/man3c/pthread_attr_setname_np.3c \
1116     target=pthread_attr_getname_np.3c
```



```
1117 link path=usr/share/man/man3c/pthread_attr_setschedparam.3c \
1118     target=pthread_attr_getschedparam.3c
1119 link path=usr/share/man/man3c/pthread_attr_setschedpolicy.3c \
1120     target=pthread_attr_getschedpolicy.3c
1121 link path=usr/share/man/man3c/pthread_attr_setscope.3c \
1122     target=pthread_attr_getscope.3c
1123 link path=usr/share/man/man3c/pthread_attr_setstack.3c \
1124     target=pthread_attr_getstack.3c
1125 link path=usr/share/man/man3c/pthread_attr_setstackaddr.3c \
1126     target=pthread_attr_getstackaddr.3c
1127 link path=usr/share/man/man3c/pthread_attr_setstacksize.3c \
1128     target=pthread_attr_getstacksize.3c
1129 link path=usr/share/man/man3c/pthread_barrier_init.3c \
1130     target=pthread_barrier_destroy.3c
1131 link path=usr/share/man/man3c/pthread_barrierattr_init.3c \
1132     target=pthread_barrierattr_destroy.3c
1133 link path=usr/share/man/man3c/pthread_barrierattr_setpshared.3c \
1134     target=pthread_barrierattr_getpshared.3c
1135 link path=usr/share/man/man3c/pthread_cond_broadcast.3c \
1136     target=pthread_cond_signal.3c
1137 link path=usr/share/man/man3c/pthread_cond_destroy.3c \
1138     target=pthread_cond_init.3c
1139 link path=usr/share/man/man3c/pthread_cond_reltimedwait_np.3c \
1140     target=pthread_cond_wait.3c
1141 link path=usr/share/man/man3c/pthread_cond_timedwait.3c \
1142     target=pthread_cond_wait.3c
1143 link path=usr/share/man/man3c/pthread_condattr_destroy.3c \
1144     target=pthread_condattr_init.3c
1145 link path=usr/share/man/man3c/pthread_condattr_setclock.3c \
1146     target=pthread_condattr_getclock.3c
1147 link path=usr/share/man/man3c/pthread_condattr_setpshared.3c \
1148     target=pthread_condattr_getpshared.3c
1149 link path=usr/share/man/man3c/pthread_key_create_once_np.3c \
1150     target=pthread_key_create.3c
1151 link path=usr/share/man/man3c/pthread_mutex_destroy.3c \
1152     target=pthread_mutex_init.3c
1153 link path=usr/share/man/man3c/pthread_mutex_reltimedlock_np.3c \
1154     target=pthread_mutex_timedlock.3c
1155 link path=usr/share/man/man3c/pthread_mutex_setprioceiling.3c \
1156     target=pthread_mutex_getprioceiling.3c
1157 link path=usr/share/man/man3c/pthread_mutex_trylock.3c \
1158     target=pthread_mutex_lock.3c
1159 link path=usr/share/man/man3c/pthread_mutex_unlock.3c \
1160     target=pthread_mutex_lock.3c
1161 link path=usr/share/man/man3c/pthread_mutexattr_destroy.3c \
1162     target=pthread_mutexattr_init.3c
1163 link path=usr/share/man/man3c/pthread_mutexattr_setprioceiling.3c \
1164     target=pthread_mutexattr_getprioceiling.3c
1165 link path=usr/share/man/man3c/pthread_mutexattr_setprotocol.3c \
1166     target=pthread_mutexattr_getprotocol.3c
1167 link path=usr/share/man/man3c/pthread_mutexattr_setpshared.3c \
1168     target=pthread_mutexattr_getpshared.3c
1169 link path=usr/share/man/man3c/pthread_mutexattr_setrobust.3c \
1170     target=pthread_mutexattr_getrobust.3c
1171 link path=usr/share/man/man3c/pthread_mutexattr_settype.3c \
1172     target=pthread_mutexattr_gettype.3c
1173 link path=usr/share/man/man3c/pthread_rwlock_destroy.3c \
1174     target=pthread_rwlock_init.3c
1175 link path=usr/share/man/man3c/pthread_rwlock_reltimedrdlock_np.3c \
1176     target=pthread_rwlock_timedrdlock.3c
1177 link path=usr/share/man/man3c/pthread_rwlock_reltimedwrlock_np.3c \
1178     target=pthread_rwlock_timedwrlock.3c
1179 link path=usr/share/man/man3c/pthread_rwlock_tryrdlock.3c \
1180     target=pthread_rwlock_rdlock.3c
1181 link path=usr/share/man/man3c/pthread_rwlock_trywrlock.3c \
1182     target=pthread_rwlock_wrlock.3c
```

```
1183 link path=usr/share/man/man3c/pthread_rwlockattr_destroy.3c \
1184     target=pthread_rwlockattr_init.3c
1185 link path=usr/share/man/man3c/pthread_rwlockattr_setpshared.3c \
1186     target=pthread_rwlockattr_getpshared.3c
1187 link path=usr/share/man/man3c/pthread_setconcurrency.3c \
1188     target=pthread_getconcurrency.3c
1189 link path=usr/share/man/man3c/pthread_setname_np.3c \
1190     target=pthread_getname_np.3c
1191 link path=usr/share/man/man3c/pthread_setschedparam.3c \
1192     target=pthread_getschedparam.3c
1193 link path=usr/share/man/man3c/pthread_setspecific.3c \
1194     target=pthread_getspecific.3c
1195 link path=usr/share/man/man3c/pthread_spin_init.3c \
1196     target=pthread_spin_destroy.3c
1197 link path=usr/share/man/man3c/pthread_spin_trylock.3c \
1198     target=pthread_spin_lock.3c
1199 link path=usr/share/man/man3c/putc.3c target=fputc.3c
1200 link path=usr/share/man/man3c/putc_unlocked.3c target=fputc.3c
1201 link path=usr/share/man/man3c/putchar.3c target=fputc.3c
1202 link path=usr/share/man/man3c/putchar_unlocked.3c target=fputc.3c
1203 link path=usr/share/man/man3c/putmntent.3c target=getmntent.3c
1204 link path=usr/share/man/man3c/pututline.3c target=getutent.3c
1205 link path=usr/share/man/man3c/pututxline.3c target=getutxent.3c
1206 link path=usr/share/man/man3c/putw.3c target=fputc.3c
1207 link path=usr/share/man/man3c/putwc.3c target=fputwc.3c
1208 link path=usr/share/man/man3c/putwchar.3c target=fputwc.3c
1209 link path=usr/share/man/man3c/qeconvert.3c target=econvert.3c
1210 link path=usr/share/man/man3c/qfconvert.3c target=econvert.3c
1211 link path=usr/share/man/man3c/qgconvert.3c target=econvert.3c
1212 link path=usr/share/man/man3c/quadruple_to_decimal.3c \
1213     target=floating_to_decimal.3c
1214 link path=usr/share/man/man3c/rand_r.3c target=rand.3c
1215 link path=usr/share/man/man3c/rctlblk_get_enforced_value.3c \
1216     target=rctlblk_set_value.3c
1217 link path=usr/share/man/man3c/rctlblk_get_firing_time.3c \
1218     target=rctlblk_set_value.3c
1219 link path=usr/share/man/man3c/rctlblk_get_global_action.3c \
1220     target=rctlblk_set_value.3c
1221 link path=usr/share/man/man3c/rctlblk_get_global_flags.3c \
1222     target=rctlblk_set_value.3c
1223 link path=usr/share/man/man3c/rctlblk_get_local_action.3c \
1224     target=rctlblk_set_value.3c
1225 link path=usr/share/man/man3c/rctlblk_get_local_flags.3c \
1226     target=rctlblk_set_value.3c
1227 link path=usr/share/man/man3c/rctlblk_get_privilege.3c \
1228     target=rctlblk_set_value.3c
1229 link path=usr/share/man/man3c/rctlblk_get_recipient_pid.3c \
1230     target=rctlblk_set_value.3c
1231 link path=usr/share/man/man3c/rctlblk_get_value.3c target=rctlblk_set_value.3c
1232 link path=usr/share/man/man3c/rctlblk_set_local_action.3c \
1233     target=rctlblk_set_value.3c
1234 link path=usr/share/man/man3c/rctlblk_set_local_flags.3c \
1235     target=rctlblk_set_value.3c
1236 link path=usr/share/man/man3c/rctlblk_set_privilege.3c \
1237     target=rctlblk_set_value.3c
1238 link path=usr/share/man/man3c/rctlblk_set_recipient_pid.3c \
1239     target=rctlblk_set_value.3c
1240 link path=usr/share/man/man3c/rctlblk_size.3c target=rctlblk_set_value.3c
1241 link path=usr/share/man/man3c/re_exec.3c target=re_comp.3c
1242 link path=usr/share/man/man3c/readdir_r.3c target=readdir.3c
1243 link path=usr/share/man/man3c/realloc.3c target=malloc.3c
1244 link path=usr/share/man/man3c/reallocarray.3c target=malloc.3c
1245 link path=usr/share/man/man3c/recallocarray.3c target=malloc.3c
1246 link path=usr/share/man/man3c/regerror.3c target=regcomp.3c
1247 link path=usr/share/man/man3c/regex.3c target=regcmp.3c
1248 link path=usr/share/man/man3c/regexec.3c target=regcomp.3c
```



```
1249 link path=usr/share/man/man3c/regfree.3c target=regcomp.3c
1250 link path=usr/share/man/man3c/remque.3c target=insque.3c
1251 link path=usr/share/man/man3c/resetmnttab.3c target=getmntent.3c
1252 link path=usr/share/man/man3c/rindex.3c target=index.3c
1253 link path=usr/share/man/man3c/rw_rdlock.3c target=rwlock.3c
1254 link path=usr/share/man/man3c/rw_tryrdlock.3c target=rwlock.3c
1255 link path=usr/share/man/man3c/rw_trywrlock.3c target=rwlock.3c
1256 link path=usr/share/man/man3c/rw_unlock.3c target=rwlock.3c
1257 link path=usr/share/man/man3c/rw_wrlock.3c target=rwlock.3c
1258 link path=usr/share/man/man3c/rwlock_destroy.3c target=rwlock.3c
1259 link path=usr/share/man/man3c/rwlock_init.3c target=rwlock.3c
1260 link path=usr/share/man/man3c/sched_get_priority_min.3c \
1261     target=sched_get_priority_max.3c
1262 link path=usr/share/man/man3c/schedctl_exit.3c target=schedctl_init.3c
1263 link path=usr/share/man/man3c/schedctl_lookup.3c target=schedctl_init.3c
1264 link path=usr/share/man/man3c/schedctl_start.3c target=schedctl_init.3c
1265 link path=usr/share/man/man3c/schedctl_stop.3c target=schedctl_init.3c
1266 link path=usr/share/man/man3c/seconvert.3c target=econvert.3c
1267 link path=usr/share/man/man3c/seed48.3c target=drand48.3c
1268 link path=usr/share/man/man3c/sem_reltimedwait_np.3c target=sem_timedwait.3c
1269 link path=usr/share/man/man3c/sem_trywait.3c target=sem_wait.3c
1270 link path=usr/share/man/man3c/sema_destroy.3c target=semaphore.3c
1271 link path=usr/share/man/man3c/sema_init.3c target=semaphore.3c
1272 link path=usr/share/man/man3c/sema_post.3c target=semaphore.3c
1273 link path=usr/share/man/man3c/sema_trywait.3c target=semaphore.3c
1274 link path=usr/share/man/man3c/sema_wait.3c target=semaphore.3c
1275 link path=usr/share/man/man3c/setattrat.3c target=fgetattr.3c
1276 link path=usr/share/man/man3c/setgrent.3c target=getgrnam.3c
1277 link path=usr/share/man/man3c/sethostname.3c target=gethostname.3c
1278 link path=usr/share/man/man3c/setlinebuf.3c target=setbuffer.3c
1279 link path=usr/share/man/man3c/setlogmask.3c target=syslog.3c
1280 link path=usr/share/man/man3c/setnetgrent.3c target=getnetgrent.3c
1281 link path=usr/share/man/man3c/setpriority.3c target=getpriority.3c
1282 link path=usr/share/man/man3c/setprogname.3c target=getprogname.3c
1283 link path=usr/share/man/man3c/setpwent.3c target=getpwnam.3c
1284 link path=usr/share/man/man3c/setspent.3c target=getspnam.3c
1285 link path=usr/share/man/man3c/setstate.3c target=random.3c
1286 link path=usr/share/man/man3c/settimeofday.3c target=gettimeofday.3c
1287 link path=usr/share/man/man3c/setusershell.3c target=getusershell.3c
1288 link path=usr/share/man/man3c/setutent.3c target=getutent.3c
1289 link path=usr/share/man/man3c/setutxent.3c target=getutxent.3c
1290 link path=usr/share/man/man3c/setvbuf.3c target=setbuf.3c
1291 link path=usr/share/man/man3c/sfconvert.3c target=econvert.3c
1292 link path=usr/share/man/man3c/sgconvert.3c target=econvert.3c
1293 link path=usr/share/man/man3c/sig2str.3c target=str2sig.3c
1294 link path=usr/share/man/man3c/sigaddset.3c target=sigsetops.3c
1295 link path=usr/share/man/man3c/sigdelset.3c target=sigsetops.3c
1296 link path=usr/share/man/man3c/sigemptyset.3c target=sigsetops.3c
1297 link path=usr/share/man/man3c/sigfillset.3c target=sigsetops.3c
1298 link path=usr/share/man/man3c/sighold.3c target=signal.3c
1299 link path=usr/share/man/man3c/sigignore.3c target=signal.3c
1300 link path=usr/share/man/man3c/sigismember.3c target=sigsetops.3c
1301 link path=usr/share/man/man3c/siglongjmp.3c target=setjmp.3c
1302 link path=usr/share/man/man3c/sigpause.3c target=signal.3c
1303 link path=usr/share/man/man3c/sigrelse.3c target=signal.3c
1304 link path=usr/share/man/man3c/sigset.3c target=signal.3c
1305 link path=usr/share/man/man3c/sigsetjmp.3c target=setjmp.3c
1306 link path=usr/share/man/man3c/sigtimedwait.3c target=sigwaitinfo.3c
1307 link path=usr/share/man/man3c/single_to_decimal.3c \
1308     target=floating_to_decimal.3c
1309 link path=usr/share/man/man3c/snprintf.3c target=printf.3c
1310 link path=usr/share/man/man3c/sprintf.3c target=printf.3c
1311 link path=usr/share/man/man3c/srand.3c target=rand.3c
1312 link path=usr/share/man/man3c/srand48.3c target=drand48.3c
1313 link path=usr/share/man/man3c/srandom.3c target=random.3c
1314 link path=usr/share/man/man3c/sscanf.3c target=scanf.3c
```

```
1315 link path=usr/share/man/man3c/stderr.3c target=stdio.3c
1316 link path=usr/share/man/man3c/stdin.3c target=stdio.3c
1317 link path=usr/share/man/man3c/stdout.3c target=stdio.3c
1318 link path=usr/share/man/man3c/stpcpy.3c target=string.3c
1319 link path=usr/share/man/man3c/stpncpy.3c target=string.3c
1320 link path=usr/share/man/man3c/strcasecmp.3c target=string.3c
1321 link path=usr/share/man/man3c/strcasecmp_l.3c target=string.3c
1322 link path=usr/share/man/man3c/strcasestr.3c target=string.3c
1323 link path=usr/share/man/man3c/strcasestr_l.3c target=string.3c
1324 link path=usr/share/man/man3c/strcat.3c target=string.3c
1325 link path=usr/share/man/man3c/strchr.3c target=string.3c
1326 link path=usr/share/man/man3c/strchrnul.3c target=string.3c
1327 link path=usr/share/man/man3c/strcmp.3c target=string.3c
1328 link path=usr/share/man/man3c/strcoll_l.3c target=strcoll.3c
1329 link path=usr/share/man/man3c/strcpy.3c target=string.3c
1330 link path=usr/share/man/man3c/strcspn.3c target=string.3c
1331 link path=usr/share/man/man3c/strdup.3c target=string.3c
1332 link path=usr/share/man/man3c/strdupa.3c target=string.3c
1333 link path=usr/share/man/man3c/strerror_l.3c target=strerror.3c
1334 link path=usr/share/man/man3c/strerror_r.3c target=strerror.3c
1335 link path=usr/share/man/man3c/strfmon_l.3c target=strfmon.3c
1336 link path=usr/share/man/man3c/strftime_l.3c target=strftime.3c
1337 link path=usr/share/man/man3c/strlcat.3c target=string.3c
1338 link path=usr/share/man/man3c/strlcpy.3c target=string.3c
1339 link path=usr/share/man/man3c/strlen.3c target=string.3c
1340 link path=usr/share/man/man3c/strncasecmp.3c target=string.3c
1341 link path=usr/share/man/man3c/strncasecmp_l.3c target=string.3c
1342 link path=usr/share/man/man3c/strncat.3c target=string.3c
1343 link path=usr/share/man/man3c/strncmp.3c target=string.3c
1344 link path=usr/share/man/man3c/strncpy.3c target=string.3c
1345 link path=usr/share/man/man3c/strndup.3c target=string.3c
1346 link path=usr/share/man/man3c/strndupa.3c target=string.3c
1347 link path=usr/share/man/man3c/strnlen.3c target=string.3c
1348 link path=usr/share/man/man3c/strnstr.3c target=string.3c
1349 link path=usr/share/man/man3c/strpbrk.3c target=string.3c
1350 link path=usr/share/man/man3c/strptime_l.3c target=strptime.3c
1351 link path=usr/share/man/man3c/strrchr.3c target=string.3c
1352 link path=usr/share/man/man3c/strsep.3c target=string.3c
1353 link path=usr/share/man/man3c/strspn.3c target=string.3c
1354 link path=usr/share/man/man3c/strstr.3c target=string.3c
1355 link path=usr/share/man/man3c/strtof.3c target=strtod.3c
1356 link path=usr/share/man/man3c/strtok.3c target=string.3c
1357 link path=usr/share/man/man3c/strtok_r.3c target=string.3c
1358 link path=usr/share/man/man3c/strtold.3c target=strtod.3c
1359 link path=usr/share/man/man3c/strtoll.3c target=strtol.3c
1360 link path=usr/share/man/man3c/strtoull.3c target=strtoul.3c
1361 link path=usr/share/man/man3c/strtoumax.3c target=strtoimax.3c
1362 link path=usr/share/man/man3c/strxfrm_l.3c target=strxfrm.3c
1363 link path=usr/share/man/man3c/swapcontext.3c target=makecontext.3c
1364 link path=usr/share/man/man3c/swprintf.3c target=fwprintf.3c
1365 link path=usr/share/man/man3c/swscanf.3c target=fwscanf.3c
1366 link path=usr/share/man/man3c/tdelete.3c target=tsearch.3c
1367 link path=usr/share/man/man3c/tempnam.3c target=tmpnam.3c
1368 link path=usr/share/man/man3c/textdomain.3c target=gettext.3c
1369 link path=usr/share/man/man3c/tfind.3c target=tsearch.3c
1370 link path=usr/share/man/man3c/thr_continue.3c target=thr_suspend.3c
1371 link path=usr/share/man/man3c/thr_getspecific.3c target=thr_keycreate.3c
1372 link path=usr/share/man/man3c/thr_keycreate_once.3c target=thr_keycreate.3c
1373 link path=usr/share/man/man3c/thr_setconcurrency.3c \
1374     target=thr_getconcurrency.3c
1375 link path=usr/share/man/man3c/thr_setname.3c \
1376     target=thr_getname.3c
1377 link path=usr/share/man/man3c/thr_setprio.3c target=thr_getprio.3c
1378 link path=usr/share/man/man3c/thr_setspecific.3c target=thr_keycreate.3c
1379 link path=usr/share/man/man3c/thrd_sleep.3c target=nanosleep.3c
1380 link path=usr/share/man/man3c/timegm.3c target=mktime.3c
```



```
1381 link path=usr/share/man/man3c/timer_getoverrun.3c target=timer_settime.3c
1382 link path=usr/share/man/man3c/timer_gettime.3c target=timer_settime.3c
1383 link path=usr/share/man/man3c/timerclear.3c target=timeradd.3c
1384 link path=usr/share/man/man3c/timercmp.3c target=timeradd.3c
1385 link path=usr/share/man/man3c/timerfd_gettime.3c target=timerfd_create.3c
1386 link path=usr/share/man/man3c/timerfd_settime.3c target=timerfd_create.3c
1387 link path=usr/share/man/man3c/timerisset.3c target=timeradd.3c
1388 link path=usr/share/man/man3c/timersub.3c target=timeradd.3c
1389 link path=usr/share/man/man3c/tmpnam_r.3c target=tmpnam.3c
1390 link path=usr/share/man/man3c/tolower_l.3c target=tolower.3c
1391 link path=usr/share/man/man3c/toupper_l.3c target=toupper.3c
1392 link path=usr/share/man/man3c/towctrans.3c target=wctrans.3c
1393 link path=usr/share/man/man3c/towctrans_l.3c target=wctrans.3c
1394 link path=usr/share/man/man3c/towlower_l.3c target=towlower.3c
1395 link path=usr/share/man/man3c/towupper_l.3c target=towupper.3c
1396 link path=usr/share/man/man3c/tss_create.3c target=tss.3c
1397 link path=usr/share/man/man3c/tss_delete.3c target=tss.3c
1398 link path=usr/share/man/man3c/tss_get.3c target=tss.3c
1399 link path=usr/share/man/man3c/tss_set.3c target=tss.3c
1400 link path=usr/share/man/man3c/ttyname_r.3c target=ttyname.3c
1401 link path=usr/share/man/man3c/twalk.3c target=tsearch.3c
1402 link path=usr/share/man/man3c/tzset.3c target=ctime.3c
1403 link path=usr/share/man/man3c/uconv_u16tou8.3c target=uconv_u16tou32.3c
1404 link path=usr/share/man/man3c/uconv_u32tou16.3c target=uconv_u16tou32.3c
1405 link path=usr/share/man/man3c/uconv_u32tou8.3c target=uconv_u16tou32.3c
1406 link path=usr/share/man/man3c/uconv_u8tou16.3c target=uconv_u16tou32.3c
1407 link path=usr/share/man/man3c/uconv_u8tou32.3c target=uconv_u16tou32.3c
1408 link path=usr/share/man/man3c/ucred_free.3c target=ucred.3c
1409 link path=usr/share/man/man3c/ucred_get.3c target=ucred.3c
1410 link path=usr/share/man/man3c/ucred_getegid.3c target=ucred.3c
1411 link path=usr/share/man/man3c/ucred_geteuid.3c target=ucred.3c
1412 link path=usr/share/man/man3c/ucred_getgroups.3c target=ucred.3c
1413 link path=usr/share/man/man3c/ucred_getlabel.3c target=ucred.3c
1414 link path=usr/share/man/man3c/ucred_getpflags.3c target=ucred.3c
1415 link path=usr/share/man/man3c/ucred_getpid.3c target=ucred.3c
1416 link path=usr/share/man/man3c/ucred_getprivset.3c target=ucred.3c
1417 link path=usr/share/man/man3c/ucred_getprojid.3c target=ucred.3c
1418 link path=usr/share/man/man3c/ucred_getrgid.3c target=ucred.3c
1419 link path=usr/share/man/man3c/ucred_getruid.3c target=ucred.3c
1420 link path=usr/share/man/man3c/ucred_getsgid.3c target=ucred.3c
1421 link path=usr/share/man/man3c/ucred_getsuid.3c target=ucred.3c
1422 link path=usr/share/man/man3c/ucred_getzoneid.3c target=ucred.3c
1423 link path=usr/share/man/man3c/ucred_size.3c target=ucred.3c
1424 link path=usr/share/man/man3c/ulckpwdf.3c target=lckpwdf.3c
1425 link path=usr/share/man/man3c/ulltostr.3c target=strtol.3c
1426 link path=usr/share/man/man3c/unordered.3c target=isnand.3c
1427 link path=usr/share/man/man3c/updwtmp.3c target=getutxent.3c
1428 link path=usr/share/man/man3c/updwtmpx.3c target=getutxent.3c
1429 link path=usr/share/man/man3c/utmpname.3c target=getutent.3c
1430 link path=usr/share/man/man3c/utmpxname.3c target=getutxent.3c
1431 link path=usr/share/man/man3c/valloc.3c target=malloc.3c
1432 link path=usr/share/man/man3c/vasprintf.3c target=vprintf.3c
1433 link path=usr/share/man/man3c/verr.3c target=err.3c
1434 link path=usr/share/man/man3c/verrx.3c target=err.3c
1435 link path=usr/share/man/man3c/vfprintf.3c target=vprintf.3c
1436 link path=usr/share/man/man3c/vfscanf.3c target=scanf.3c
1437 link path=usr/share/man/man3c/vfwscanf.3c target=fwscanf.3c
1438 link path=usr/share/man/man3c/vscanf.3c target=scanf.3c
1439 link path=usr/share/man/man3c/vsnprintf.3c target=vprintf.3c
1440 link path=usr/share/man/man3c/vsprintf.3c target=vprintf.3c
1441 link path=usr/share/man/man3c/vsscanf.3c target=scanf.3c
1442 link path=usr/share/man/man3c/vswprintf.3c target=vfwprintf.3c
1443 link path=usr/share/man/man3c/vswscanf.3c target=fwscanf.3c
1444 link path=usr/share/man/man3c/vwarn.3c target=err.3c
1445 link path=usr/share/man/man3c/vwarnx.3c target=err.3c
1446 link path=usr/share/man/man3c/vwprintf.3c target=vfwprintf.3c
```

```
1447 link path=usr/share/man/man3c/vwscanf.3c target=fwscanf.3c
1448 link path=usr/share/man/man3c/wait4.3c target=wait3.3c
1449 link path=usr/share/man/man3c/warn.3c target=err.3c
1450 link path=usr/share/man/man3c/warnx.3c target=err.3c
1451 link path=usr/share/man/man3c/watof.3c target=wcstod.3c
1452 link path=usr/share/man/man3c/watoi.3c target=wcstol.3c
1453 link path=usr/share/man/man3c/watol.3c target=wcstol.3c
1454 link path=usr/share/man/man3c/watoll.3c target=wcstol.3c
1455 link path=usr/share/man/man3c/wcpncpy.3c target=wcpcpy.3c
1456 link path=usr/share/man/man3c/wcrtomb_l.3c target=wcrtomb.3c
1457 link path=usr/share/man/man3c/wcscasecmp_l.3c target=wcscasecmp.3c
1458 link path=usr/share/man/man3c/wcscat.3c target=wcstring.3c
1459 link path=usr/share/man/man3c/wcschr.3c target=wcstring.3c
1460 link path=usr/share/man/man3c/wcscmp.3c target=wcstring.3c
1461 link path=usr/share/man/man3c/wcscoll_l.3c target=wcscoll.3c
1462 link path=usr/share/man/man3c/wcscpy.3c target=wcstring.3c
1463 link path=usr/share/man/man3c/wcscspn.3c target=wcstring.3c
1464 link path=usr/share/man/man3c/wcsetno.3c target=cset.3c
1465 link path=usr/share/man/man3c/wcsncasecmp.3c target=wcscasecmp.3c
1466 link path=usr/share/man/man3c/wcsncasecmp_l.3c target=wcscasecmp.3c
1467 link path=usr/share/man/man3c/wcsncat.3c target=wcstring.3c
1468 link path=usr/share/man/man3c/wcsncmp.3c target=wcstring.3c
1469 link path=usr/share/man/man3c/wcsncpy.3c target=wcstring.3c
1470 link path=usr/share/man/man3c/wcsnlen.3c target=wcslen.3c
1471 link path=usr/share/man/man3c/wcsnrtombs.3c target=wcsrtombs.3c
1472 link path=usr/share/man/man3c/wcsnrtombs_l.3c target=wcsrtombs.3c
1473 link path=usr/share/man/man3c/wcspbrk.3c target=wcstring.3c
1474 link path=usr/share/man/man3c/wcsrchr.3c target=wcstring.3c
1475 link path=usr/share/man/man3c/wcsrtombs_l.3c target=wcsrtombs.3c
1476 link path=usr/share/man/man3c/wcsspn.3c target=wcstring.3c
1477 link path=usr/share/man/man3c/wcstof.3c target=wcstod.3c
1478 link path=usr/share/man/man3c/wcstok.3c target=wcstring.3c
1479 link path=usr/share/man/man3c/wcstold.3c target=wcstod.3c
1480 link path=usr/share/man/man3c/wcstoll.3c target=wcstol.3c
1481 link path=usr/share/man/man3c/wcstombs_l.3c target=wcstombs.3c
1482 link path=usr/share/man/man3c/wcstoull.3c target=wcstoul.3c
1483 link path=usr/share/man/man3c/wcstoumax.3c target=wcstoimax.3c
1484 link path=usr/share/man/man3c/wcswcs.3c target=wcstring.3c
1485 link path=usr/share/man/man3c/wcswidth_l.3c target=wcswidth.3c
1486 link path=usr/share/man/man3c/wctob_l.3c target=wctob.3c
1487 link path=usr/share/man/man3c/wctomb_l.3c target=wctomb.3c
1488 link path=usr/share/man/man3c/wctrans_l.3c target=wctrans.3c
1489 link path=usr/share/man/man3c/wctype_l.3c target=wctype.3c
1490 link path=usr/share/man/man3c/wcwidth_l.3c target=wcwidth.3c
1491 link path=usr/share/man/man3c/windex.3c target=wcstring.3c
1492 link path=usr/share/man/man3c/wordfree.3c target=wordexp.3c
1493 link path=usr/share/man/man3c/wprintf.3c target=fwprintf.3c
1494 link path=usr/share/man/man3c/wrindex.3c target=wcstring.3c
1495 link path=usr/share/man/man3c/wscanf.3c target=fwscanf.3c
1496 link path=usr/share/man/man3c/wscasecmp.3c target=wstring.3c
1497 link path=usr/share/man/man3c/wscat.3c target=wcstring.3c
1498 link path=usr/share/man/man3c/wschr.3c target=wcstring.3c
1499 link path=usr/share/man/man3c/wscmp.3c target=wcstring.3c
1500 link path=usr/share/man/man3c/wscol.3c target=wstring.3c
1501 link path=usr/share/man/man3c/wscoll.3c target=wcscoll.3c
1502 link path=usr/share/man/man3c/wscpy.3c target=wcstring.3c
1503 link path=usr/share/man/man3c/wscspn.3c target=wcstring.3c
1504 link path=usr/share/man/man3c/wsdup.3c target=wstring.3c
1505 link path=usr/share/man/man3c/wslen.3c target=wcstring.3c
1506 link path=usr/share/man/man3c/wsncasecmp.3c target=wstring.3c
1507 link path=usr/share/man/man3c/wsncat.3c target=wcstring.3c
1508 link path=usr/share/man/man3c/wsncmp.3c target=wcstring.3c
1509 link path=usr/share/man/man3c/wsncpy.3c target=wcstring.3c
1510 link path=usr/share/man/man3c/wspbrk.3c target=wcstring.3c
1511 link path=usr/share/man/man3c/wsrchr.3c target=wcstring.3c
1512 link path=usr/share/man/man3c/wsspn.3c target=wcstring.3c
```



```
1513 link path=usr/share/man/man3c/wstod.3c target=wcstod.3c
1514 link path=usr/share/man/man3c/wstok.3c target=wcstring.3c
1515 link path=usr/share/man/man3c/wstol.3c target=wcstol.3c
1516 link path=usr/share/man/man3c/wstostr.3c target=strtows.3c
1517 link path=usr/share/man/man3c/wsxfrm.3c target=wcsxfrm.3c
```

```
**********************************************************
   15294 Mon Oct 15 13:28:39 2018
new/usr/src/pkg/manifests/system-library.man3proc.inc
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2018 Joyent, Inc.
  13 # Copyright 2015 Joyent, Inc.
  14 #

  16 file path=usr/share/man/man3proc/proc_service.3proc
  17 file path=usr/share/man/man3proc/ps_lgetregs.3proc
  18 file path=usr/share/man/man3proc/ps_pglobal_lookup.3proc
  19 file path=usr/share/man/man3proc/ps_pread.3proc
  20 file path=usr/share/man/man3proc/ps_pstop.3proc
  21 link path=usr/share/man/man3proc/ps_kill.3proc target=ps_pstop.3proc
  22 link path=usr/share/man/man3proc/ps_lcontinue.3proc target=ps_pstop.3proc
  23 link path=usr/share/man/man3proc/ps_lgetfpregs.3proc target=ps_lgetregs.3proc
  24 link path=usr/share/man/man3proc/ps_lgetxregs.3proc target=ps_lgetregs.3proc
  25 link path=usr/share/man/man3proc/ps_lgetxregsize.3proc \
  26     target=ps_lgetregs.3proc
  27 link path=usr/share/man/man3proc/ps_lrolltoaddr.3proc target=ps_pstop.3proc
  28 link path=usr/share/man/man3proc/ps_lsetfpregs.3proc target=ps_lgetregs.3proc
  29 link path=usr/share/man/man3proc/ps_lsetregs.3proc target=ps_lgetregs.3proc
  30 link path=usr/share/man/man3proc/ps_lsetxregs.3proc target=ps_lgetregs.3proc
  31 link path=usr/share/man/man3proc/ps_lstop.3proc target=ps_pstop.3proc
  32 link path=usr/share/man/man3proc/ps_pcontinue.3proc target=ps_pstop.3proc
  33 link path=usr/share/man/man3proc/ps_pdread.3proc target=ps_pread.3proc
  34 link path=usr/share/man/man3proc/ps_pdwrite.3proc target=ps_pread.3proc
  35 link path=usr/share/man/man3proc/ps_pglobal_sym.3proc \
  36     target=ps_pglobal_lookup.3proc
  37 link path=usr/share/man/man3proc/ps_ptread.3proc target=ps_pread.3proc
  38 link path=usr/share/man/man3proc/ps_ptwrite.3proc target=ps_pread.3proc
  39 link path=usr/share/man/man3proc/ps_pwrite.3proc target=ps_pread.3proc
  40 file path=usr/share/man/man3proc/Lctlfd.3proc
  41 file path=usr/share/man/man3proc/Lfree.3proc
  42 file path=usr/share/man/man3proc/Lgrab_error.3proc
  43 file path=usr/share/man/man3proc/Lgrab.3proc
  44 file path=usr/share/man/man3proc/Lprochandle.3proc
  45 file path=usr/share/man/man3proc/Lpsinfo.3proc
  46 file path=usr/share/man/man3proc/Lstate.3proc
  47 file path=usr/share/man/man3proc/Lstatus.3proc
  48 file path=usr/share/man/man3proc/Paddr_to_ctf.3proc
  49 file path=usr/share/man/man3proc/Paddr_to_loadobj.3proc
  50 file path=usr/share/man/man3proc/Paddr_to_map.3proc
  51 file path=usr/share/man/man3proc/Pasfd.3proc
  52 file path=usr/share/man/man3proc/Pclearfault.3proc
  53 file path=usr/share/man/man3proc/Pclearsig.3proc
  54 file path=usr/share/man/man3proc/Pcontent.3proc
  55 file path=usr/share/man/man3proc/Pcreate_agent.3proc
  56 file path=usr/share/man/man3proc/Pcreate_error.3proc
  57 file path=usr/share/man/man3proc/Pcreate.3proc
  58 file path=usr/share/man/man3proc/Pcred.3proc
  59 file path=usr/share/man/man3proc/Pctlfd.3proc
```



```
  60 file path=usr/share/man/man3proc/Pdelbkpt.3proc
  61 file path=usr/share/man/man3proc/Pdelwapt.3proc
  62 file path=usr/share/man/man3proc/Pdestroy_agent.3proc
  63 file path=usr/share/man/man3proc/Penv_iter.3proc
  64 file path=usr/share/man/man3proc/Perror_printf.3proc
  65 file path=usr/share/man/man3proc/Pexecname.3proc
  66 file path=usr/share/man/man3proc/Pfault.3proc
  67 file path=usr/share/man/man3proc/Pfdinfo_iter.3proc
  68 file path=usr/share/man/man3proc/Pgcore.3proc
  69 file path=usr/share/man/man3proc/Pgetareg.3proc
  70 file path=usr/share/man/man3proc/Pgetauxval.3proc
  71 file path=usr/share/man/man3proc/Pgetauxvec.3proc
  72 file path=usr/share/man/man3proc/Pgetenv.3proc
  73 file path=usr/share/man/man3proc/Pgrab_core.3proc
  74 file path=usr/share/man/man3proc/Pgrab_error.3proc
  75 file path=usr/share/man/man3proc/Pgrab_file.3proc
  76 file path=usr/share/man/man3proc/Pgrab.3proc
  77 file path=usr/share/man/man3proc/Pisprocdir.3proc
  78 file path=usr/share/man/man3proc/Pissyscall.3proc
  79 file path=usr/share/man/man3proc/Pldt.3proc
  80 file path=usr/share/man/man3proc/Plmid.3proc
  81 file path=usr/share/man/man3proc/Plookup_by_addr.3proc
  82 file path=usr/share/man/man3proc/Plwp_getasrs.3proc
  83 file path=usr/share/man/man3proc/Plwp_getgwindows.3proc
  84 file path=usr/share/man/man3proc/Plwp_getname.3proc
  85 file path=usr/share/man/man3proc/Plwp_getpsinfo.3proc
  86 file path=usr/share/man/man3proc/Plwp_getregs.3proc
  87 file path=usr/share/man/man3proc/Plwp_getspymaster.3proc
  88 file path=usr/share/man/man3proc/Plwp_getxregs.3proc
  89 file path=usr/share/man/man3proc/Plwp_iter.3proc
  90 file path=usr/share/man/man3proc/Plwp_stack.3proc
  91 file path=usr/share/man/man3proc/Pmapping_iter.3proc
  92 file path=usr/share/man/man3proc/Pobjname.3proc
  93 file path=usr/share/man/man3proc/Pplatform.3proc
  94 file path=usr/share/man/man3proc/Ppltdest.3proc
  95 file path=usr/share/man/man3proc/Ppriv.3proc
  96 file path=usr/share/man/man3proc/Ppsinfo.3proc
  97 file path=usr/share/man/man3proc/pr_access.3proc
  98 file path=usr/share/man/man3proc/pr_close.3proc
  99 file path=usr/share/man/man3proc/pr_creat.3proc
 100 file path=usr/share/man/man3proc/pr_door_info.3proc
 101 file path=usr/share/man/man3proc/pr_exit.3proc
 102 file path=usr/share/man/man3proc/pr_fcntl.3proc
 103 file path=usr/share/man/man3proc/pr_fstatvfs.3proc
 104 file path=usr/share/man/man3proc/pr_getitimer.3proc
 105 file path=usr/share/man/man3proc/pr_getpeername.3proc
 106 file path=usr/share/man/man3proc/pr_getpeerucred.3proc
 107 file path=usr/share/man/man3proc/pr_getprojid.3proc
 108 file path=usr/share/man/man3proc/pr_getrctl.3proc
 109 file path=usr/share/man/man3proc/pr_getrlimit.3proc
 110 file path=usr/share/man/man3proc/pr_getsockname.3proc
 111 file path=usr/share/man/man3proc/pr_getsockopt.3proc
 112 file path=usr/share/man/man3proc/pr_gettaskid.3proc
 113 file path=usr/share/man/man3proc/pr_getzoneid.3proc
 114 file path=usr/share/man/man3proc/pr_ioctl.3proc
 115 file path=usr/share/man/man3proc/pr_link.3proc
 116 file path=usr/share/man/man3proc/pr_llseek.3proc
 117 file path=usr/share/man/man3proc/pr_lseek.3proc
 118 file path=usr/share/man/man3proc/pr_memcntl.3proc
 119 file path=usr/share/man/man3proc/pr_meminfo.3proc
 120 file path=usr/share/man/man3proc/pr_mmap.3proc
 121 file path=usr/share/man/man3proc/pr_munmap.3proc
 122 file path=usr/share/man/man3proc/pr_open.3proc
 123 file path=usr/share/man/man3proc/pr_processor_bind.3proc
 124 file path=usr/share/man/man3proc/pr_rename.3proc
 125 file path=usr/share/man/man3proc/pr_setitimer.3proc
```

```
126 file path=usr/share/man/man3proc/pr_setrctl.3proc
127 file path=usr/share/man/man3proc/pr_setrlimit.3proc
128 file path=usr/share/man/man3proc/pr_settaskid.3proc
129 file path=usr/share/man/man3proc/pr_sigaction.3proc
130 file path=usr/share/man/man3proc/pr_stat.3proc
131 file path=usr/share/man/man3proc/pr_statvfs.3proc
132 file path=usr/share/man/man3proc/pr_unlink.3proc
133 file path=usr/share/man/man3proc/pr_waitid.3proc
134 file path=usr/share/man/man3proc/Prd_agent.3proc
135 file path=usr/share/man/man3proc/Pread.3proc
136 file path=usr/share/man/man3proc/Prelease.3proc
137 file path=usr/share/man/man3proc/Preopen.3proc
138 file path=usr/share/man/man3proc/Preset_maps.3proc
139 file path=usr/share/man/man3proc/proc_arg_grab.3proc
140 file path=usr/share/man/man3proc/proc_arg_psinfo.3proc
141 file path=usr/share/man/man3proc/proc_content2str.3proc
142 file path=usr/share/man/man3proc/proc_fltname.3proc
143 file path=usr/share/man/man3proc/proc_fltset2str.3proc
144 file path=usr/share/man/man3proc/proc_get_auxv.3proc
145 file path=usr/share/man/man3proc/proc_get_cred.3proc
146 file path=usr/share/man/man3proc/proc_get_priv.3proc
147 file path=usr/share/man/man3proc/proc_get_psinfo.3proc
148 file path=usr/share/man/man3proc/proc_get_status.3proc
149 file path=usr/share/man/man3proc/proc_initstdio.3proc
150 file path=usr/share/man/man3proc/proc_lwp_in_set.3proc
151 file path=usr/share/man/man3proc/proc_str2flt.3proc
152 file path=usr/share/man/man3proc/proc_str2fltset.3proc
153 file path=usr/share/man/man3proc/proc_unctrl_psinfo.3proc
154 file path=usr/share/man/man3proc/proc_walk.3proc
155 file path=usr/share/man/man3proc/Psecflags.3proc
156 file path=usr/share/man/man3proc/Psetbkpt.3proc
157 file path=usr/share/man/man3proc/Psetcred.3proc
158 file path=usr/share/man/man3proc/Psetfault.3proc
159 file path=usr/share/man/man3proc/Psetflags.3proc
160 file path=usr/share/man/man3proc/Psetpriv.3proc
161 file path=usr/share/man/man3proc/Psetrun.3proc
162 file path=usr/share/man/man3proc/Psetsignal.3proc
163 file path=usr/share/man/man3proc/Psetsysentry.3proc
164 file path=usr/share/man/man3proc/Psetwapt.3proc
165 file path=usr/share/man/man3proc/Psetzoneid.3proc
166 file path=usr/share/man/man3proc/Psignal.3proc
167 file path=usr/share/man/man3proc/Pstack_iter.3proc
168 file path=usr/share/man/man3proc/Pstate.3proc
169 file path=usr/share/man/man3proc/Pstatus.3proc
170 file path=usr/share/man/man3proc/Pstopstatus.3proc
171 file path=usr/share/man/man3proc/Psymbol_iter.3proc
172 file path=usr/share/man/man3proc/Psync.3proc
173 file path=usr/share/man/man3proc/Psysentry.3proc
174 file path=usr/share/man/man3proc/Puname.3proc
175 file path=usr/share/man/man3proc/Pupdate_maps.3proc
176 file path=usr/share/man/man3proc/Pupdate_syms.3proc
177 file path=usr/share/man/man3proc/Pwrite.3proc
178 file path=usr/share/man/man3proc/Pxecbkpt.3proc
179 file path=usr/share/man/man3proc/Pzonename.3proc
180 link path=usr/share/man/man3proc/Lalt_stack.3proc target=Plwp_stack.3proc
181 link path=usr/share/man/man3proc/Lclearfault.3proc target=Pclearfault.3proc
182 link path=usr/share/man/man3proc/Lclearsig.3proc target=Pclearsig.3proc
183 link path=usr/share/man/man3proc/Ldstop.3proc target=Pstopstatus.3proc
184 link path=usr/share/man/man3proc/Lgetareg.3proc target=Pgetareg.3proc
185 link path=usr/share/man/man3proc/Lmain_stack.3proc target=Plwp_stack.3proc
186 link path=usr/share/man/man3proc/Lputareg.3proc target=Pgetareg.3proc
187 link path=usr/share/man/man3proc/Lsetrun.3proc target=Psetrun.3proc
188 link path=usr/share/man/man3proc/Lstack.3proc target=Plwp_stack.3proc
189 link path=usr/share/man/man3proc/Lstop.3proc target=Pstopstatus.3proc
190 link path=usr/share/man/man3proc/Lsync.3proc target=Psync.3proc
191 link path=usr/share/man/man3proc/Lwait.3proc target=Pstopstatus.3proc
```



```
192 link path=usr/share/man/man3proc/Lxecbkpt.3proc target=Pxecbkpt.3proc
193 link path=usr/share/man/man3proc/Lxecwapt.3proc target=Pxecbkpt.3proc
194 link path=usr/share/man/man3proc/Paddr_to_text_map.3proc target=Paddr_to_map.3pr
195 link path=usr/share/man/man3proc/Pcreate_callback.3proc target=Pcreate.3proc
196 link path=usr/share/man/man3proc/Pdstop.3proc target=Pstopstatus.3proc
197 link path=usr/share/man/man3proc/Pfgcore.3proc target=Pgcore.3proc
198 link path=usr/share/man/man3proc/Pfgrab_core.3proc target=Pgrab_core.3proc
199 link path=usr/share/man/man3proc/Pfree.3proc target=Prelease.3proc
200 link path=usr/share/man/man3proc/Pissyscall_prev.3proc target=Pissyscall.3proc
201 link path=usr/share/man/man3proc/Plmid_to_ctf.3proc target=Paddr_to_ctf.3proc
202 link path=usr/share/man/man3proc/Plmid_to_loadobj.3proc target=Paddr_to_loadobj.
203 link path=usr/share/man/man3proc/Plmid_to_map.3proc target=Paddr_to_map.3proc
204 link path=usr/share/man/man3proc/Plookup_by_name.3proc target=Plookup_by_addr.3p
205 link path=usr/share/man/man3proc/Plwp_alt_stack.3proc target=Plwp_stack.3proc
206 link path=usr/share/man/man3proc/Plwp_getfpregs.3proc target=Plwp_getregs.3proc
207 link path=usr/share/man/man3proc/Plwp_iter_all.3proc target=Plwp_iter.3proc
208 link path=usr/share/man/man3proc/Plwp_main_stack.3proc target=Plwp_stack.3proc
209 link path=usr/share/man/man3proc/Plwp_setasrs.3proc target=Plwp_getasrs.3proc
210 link path=usr/share/man/man3proc/Plwp_setfpregs.3proc target=Plwp_getregs.3proc
211 link path=usr/share/man/man3proc/Plwp_setregs.3proc target=Plwp_getregs.3proc
212 link path=usr/share/man/man3proc/Plwp_setxregs.3proc target=Plwp_getxregs.3proc
213 link path=usr/share/man/man3proc/Pmapping_iter_resolved.3proc target=Pmapping_it
214 link path=usr/share/man/man3proc/Pname_to_ctf.3proc target=Paddr_to_ctf.3proc
215 link path=usr/share/man/man3proc/Pname_to_loadobj.3proc target=Paddr_to_loadobj.
216 link path=usr/share/man/man3proc/Pname_to_map.3proc target=Paddr_to_map.3proc
217 link path=usr/share/man/man3proc/Pobject_iter_resolved.3proc target=Pmapping_ite
218 link path=usr/share/man/man3proc/Pobject_iter.3proc target=Pmapping_iter.3proc
219 link path=usr/share/man/man3proc/Pobjname_resolved.3proc target=Pobjname.3proc
220 link path=usr/share/man/man3proc/Ppriv_free.3proc target=Ppriv.3proc
221 link path=usr/share/man/man3proc/Pputareg.3proc target=Pgetareg.3proc
222 link path=usr/share/man/man3proc/pr_fstat.3proc target=pr_stat.3proc
223 link path=usr/share/man/man3proc/pr_fstat64.3proc target=pr_stat.3proc
224 link path=usr/share/man/man3proc/pr_getrlimit64.3proc target=pr_getrlimit.3proc
225 link path=usr/share/man/man3proc/pr_lstat.3proc target=pr_stat.3proc
226 link path=usr/share/man/man3proc/pr_lstat64.3proc target=pr_stat.3proc
227 link path=usr/share/man/man3proc/pr_setrlimit64.3proc target=pr_setrlimit.3proc
228 link path=usr/share/man/man3proc/pr_stat64.3proc target=pr_stat.3proc
229 link path=usr/share/man/man3proc/Pread_string.3proc target=Pread.3proc
230 link path=usr/share/man/man3proc/proc_arg_xgrab.3proc target=proc_arg_grab.3proc
231 link path=usr/share/man/man3proc/proc_arg_xpsinfo.3proc target=proc_arg_psinfo.3
232 link path=usr/share/man/man3proc/proc_finistdio.3proc target=proc_initstdio.3pro
233 link path=usr/share/man/man3proc/proc_flushstdio.3proc target=proc_initstdio.3pr
234 link path=usr/share/man/man3proc/proc_free_priv.3proc target=proc_get_priv.3proc
235 link path=usr/share/man/man3proc/proc_get_ldt.3proc target=Pldt.3proc
236 link path=usr/share/man/man3proc/proc_lwp_range_valid.3proc target=proc_lwp_in_s
237 link path=usr/share/man/man3proc/proc_signame.3proc target=proc_fltname.3proc
238 link path=usr/share/man/man3proc/proc_sigset2str.3proc target=proc_fltset2str.3p
239 link path=usr/share/man/man3proc/proc_str2content.3proc target=proc_content2str.
240 link path=usr/share/man/man3proc/proc_str2sig.3proc target=proc_str2flt.3proc
241 link path=usr/share/man/man3proc/proc_str2sigset.3proc target=proc_str2fltset.3p
242 link path=usr/share/man/man3proc/proc_str2sys.3proc target=proc_str2flt.3proc
243 link path=usr/share/man/man3proc/proc_str2sysset.3proc target=proc_str2fltset.3p
244 link path=usr/share/man/man3proc/proc_sysname.3proc target=proc_fltname.3proc
245 link path=usr/share/man/man3proc/proc_sysset2str.3proc target=proc_fltset2str.3p
246 link path=usr/share/man/man3proc/Psetsysexit.3proc target=Psetsysentry.3proc
247 link path=usr/share/man/man3proc/Pstop.3proc target=Pstopstatus.3proc
248 link path=usr/share/man/man3proc/Psymbol_iter_by_addr.3proc target=Psymbol_iter.
249 link path=usr/share/man/man3proc/Psymbol_iter_by_lmid.3proc target=Psymbol_iter.
250 link path=usr/share/man/man3proc/Psymbol_iter_by_name.3proc target=Psymbol_iter.
251 link path=usr/share/man/man3proc/Psysexit.3proc target=Psysentry.3proc
252 link path=usr/share/man/man3proc/Punsetflags.3proc target=Psetflags.3proc
253 link path=usr/share/man/man3proc/Pwait.3proc target=Pstopstatus.3proc
254 link path=usr/share/man/man3proc/Pxcreate.3proc target=Pcreate.3proc
255 link path=usr/share/man/man3proc/Pxecwapt.3proc target=Pxecbkpt.3proc
256 link path=usr/share/man/man3proc/Pxlookup_by_addr_resolved.3proc target=Plookup_
257 link path=usr/share/man/man3proc/Pxlookup_by_addr.3proc target=Plookup_by_addr.3
```

```
 258 link path=usr/share/man/man3proc/Pxlookup_by_name.3proc target=Plookup_by_addr.3
 259 link path=usr/share/man/man3proc/Pxsymbol_iter.3proc target=Psymbol_iter.3proc
 260 link path=usr/share/man/man3proc/Pzonepath.3proc target=Pzonename.3proc
 261 link path=usr/share/man/man3proc/Pzoneroot.3proc target=Pzonename.3proc
```

```
**********************************************************
   12083 Mon Oct 15 13:28:40 2018
new/usr/src/pkg/manifests/system-test-libctest.mf
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet at
   9 # http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright (c) 2012 by Delphix. All rights reserved.
  14 # Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.
  15 # Copyright 2015 Garrett D'Amore <garrett@damore.org>
  16 # Copyright 2018 Joyent, Inc.
  17 #

  19 set name=pkg.fmri value=pkg:/system/test/libctest@$(PKGVERS)
  20 set name=pkg.description value="C library Unit Tests"
  21 set name=pkg.summary value="C Library Unit Test Suite"
  22 set name=info.classification \
  23     value=org.opensolaris.category.2008:Development/System
  24 set name=variant.arch value=$(ARCH)
  25 dir path=opt/libc-tests
  26 dir path=opt/libc-tests/bin
  27 dir path=opt/libc-tests/cfg
  28 dir path=opt/libc-tests/cfg/symbols
  29 dir path=opt/libc-tests/runfiles
  30 dir path=opt/libc-tests/tests
  31 dir path=opt/libc-tests/tests/random
  32 dir path=opt/libc-tests/tests/regex
  33 dir path=opt/libc-tests/tests/regex/data
  34 dir path=opt/libc-tests/tests/select
  35 dir path=opt/libc-tests/tests/symbols
  36 dir path=usr group=sys
  37 dir path=usr/lib
  38 dir path=usr/lib/locale
  39 dir path=usr/lib/locale/zz_AA.UTF-8
  40 dir path=usr/lib/locale/zz_AA.UTF-8/LC_COLLATE
  41 dir path=usr/lib/locale/zz_AA.UTF-8/LC_CTYPE
  42 dir path=usr/lib/locale/zz_AA.UTF-8/LC_MESSAGES
  43 dir path=usr/lib/locale/zz_AA.UTF-8/LC_MONETARY
  44 dir path=usr/lib/locale/zz_AA.UTF-8/LC_NUMERIC
  45 dir path=usr/lib/locale/zz_AA.UTF-8/LC_TIME
  46 file path=opt/libc-tests/README mode=0444
  47 file path=opt/libc-tests/bin/libctest mode=0555
  48 file path=opt/libc-tests/cfg/README mode=0444
  49 file path=opt/libc-tests/cfg/compilation.cfg mode=0444
  50 file path=opt/libc-tests/cfg/symbols/README mode=0444
  51 file path=opt/libc-tests/cfg/symbols/assert_h.cfg mode=0444
  52 file path=opt/libc-tests/cfg/symbols/ctype_h.cfg mode=0444
  53 file path=opt/libc-tests/cfg/symbols/dirent_h.cfg mode=0444
  54 file path=opt/libc-tests/cfg/symbols/fcntl_h.cfg mode=0444
  55 file path=opt/libc-tests/cfg/symbols/locale_h.cfg mode=0444
  56 file path=opt/libc-tests/cfg/symbols/math_h.cfg mode=0444
  57 file path=opt/libc-tests/cfg/symbols/netdb_h.cfg mode=0444
  58 file path=opt/libc-tests/cfg/symbols/pthread_h.cfg mode=0444
  59 file path=opt/libc-tests/cfg/symbols/signal_h.cfg mode=0444
  60 file path=opt/libc-tests/cfg/symbols/stdalign_h.cfg mode=0444
```



```
  61 file path=opt/libc-tests/cfg/symbols/stddef_h.cfg mode=0444
  62 file path=opt/libc-tests/cfg/symbols/stdio_h.cfg mode=0444
  63 file path=opt/libc-tests/cfg/symbols/stdlib_h.cfg mode=0444
  64 file path=opt/libc-tests/cfg/symbols/stdnoreturn_h.cfg mode=0444
  65 file path=opt/libc-tests/cfg/symbols/string_h.cfg mode=0444
  66 file path=opt/libc-tests/cfg/symbols/strings_h.cfg mode=0444
  67 file path=opt/libc-tests/cfg/symbols/sys_stat_h.cfg mode=0444
  68 file path=opt/libc-tests/cfg/symbols/sys_time_h.cfg mode=0444
  69 file path=opt/libc-tests/cfg/symbols/sys_timeb_h.cfg mode=0444
  70 file path=opt/libc-tests/cfg/symbols/threads_h.cfg mode=0444
  71 file path=opt/libc-tests/cfg/symbols/time_h.cfg mode=0444
  72 file path=opt/libc-tests/cfg/symbols/ucontext_h.cfg mode=0444
  73 file path=opt/libc-tests/cfg/symbols/unistd_h.cfg mode=0444
  74 file path=opt/libc-tests/cfg/symbols/wchar_h.cfg mode=0444
  75 file path=opt/libc-tests/cfg/symbols/wctype_h.cfg mode=0444
  76 file path=opt/libc-tests/runfiles/default.run mode=0444
  77 file path=opt/libc-tests/tests/aligned_alloc.32 mode=0555
  78 file path=opt/libc-tests/tests/aligned_alloc.64 mode=0555
  79 file path=opt/libc-tests/tests/c11_threads.32 mode=0555
  80 file path=opt/libc-tests/tests/c11_threads.64 mode=0555
  81 file path=opt/libc-tests/tests/c11_tss.32 mode=0555
  82 file path=opt/libc-tests/tests/c11_tss.64 mode=0555
  83 file path=opt/libc-tests/tests/call_once.32 mode=0555
  84 file path=opt/libc-tests/tests/call_once.64 mode=0555
  85 file path=opt/libc-tests/tests/catopen mode=0555
  86 file path=opt/libc-tests/tests/endian.32 mode=0555
  87 file path=opt/libc-tests/tests/endian.64 mode=0555
  88 file path=opt/libc-tests/tests/env-7076.32 mode=0555
  89 file path=opt/libc-tests/tests/env-7076.64 mode=0555
  90 file path=opt/libc-tests/tests/fnmatch.32 mode=0555
  91 file path=opt/libc-tests/tests/fnmatch.64 mode=0555
  92 file path=opt/libc-tests/tests/fpround_test mode=0555
  93 file path=opt/libc-tests/tests/fpround_test.$(ARCH) mode=0555
  94 file path=opt/libc-tests/tests/fpround_test.$(ARCH64) mode=0555
  95 file path=opt/libc-tests/tests/memset_s.32 mode=0555
  96 file path=opt/libc-tests/tests/memset_s.64 mode=0555
  97 file path=opt/libc-tests/tests/newlocale_test mode=0555
  98 file path=opt/libc-tests/tests/newlocale_test.$(ARCH) mode=0555
  99 file path=opt/libc-tests/tests/newlocale_test.$(ARCH64) mode=0555
 100 file path=opt/libc-tests/tests/nl_langinfo_test mode=0555
 101 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH) mode=0555
 102 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH64) mode=0555
 103 file path=opt/libc-tests/tests/printf-6961.64 mode=0555
 104 file path=opt/libc-tests/tests/printf-9511.32 mode=0555
 105 file path=opt/libc-tests/tests/printf-9511.64 mode=0555
 106 file path=opt/libc-tests/tests/priv_gettext mode=0555
 107 file path=opt/libc-tests/tests/psignal mode=0555
 108 file path=opt/libc-tests/tests/psignal-5097.32 mode=0555
 109 file path=opt/libc-tests/tests/psignal-5097.64 mode=0555
 110 file path=opt/libc-tests/tests/pthread_attr_get_np mode=0555
 111 file path=opt/libc-tests/tests/quick_exit mode=0555
 112 file path=opt/libc-tests/tests/quick_exit_order.32 mode=0555
 113 file path=opt/libc-tests/tests/quick_exit_order.64 mode=0555
 114 file path=opt/libc-tests/tests/quick_exit_status.32 mode=0555
 115 file path=opt/libc-tests/tests/quick_exit_status.64 mode=0555
 116 file path=opt/libc-tests/tests/random/arc4key.ksh mode=0555
 117 file path=opt/libc-tests/tests/random/arc4random mode=0555
 118 file path=opt/libc-tests/tests/random/arc4random_fork mode=0555
 119 file path=opt/libc-tests/tests/random/arc4random_forkall mode=0555
 120 file path=opt/libc-tests/tests/random/arc4random_forksig mode=0555
 121 file path=opt/libc-tests/tests/random/arc4random_prefork mode=0555
 122 file path=opt/libc-tests/tests/random/arc4random_preforkall mode=0555
 123 file path=opt/libc-tests/tests/random/arc4random_preforksig mode=0555
 124 file path=opt/libc-tests/tests/random/arc4random_rekey mode=0555
 125 file path=opt/libc-tests/tests/random/chacha mode=0555
 126 file path=opt/libc-tests/tests/random/getentropy mode=0555
```

```
 127 file path=opt/libc-tests/tests/random/getrandom mode=0555
 128 file path=opt/libc-tests/tests/random/getrandred mode=0555
 129 file path=opt/libc-tests/tests/random/inz_child mode=0555
 130 file path=opt/libc-tests/tests/random/inz_inval mode=0555
 131 file path=opt/libc-tests/tests/random/inz_mlock mode=0555
 132 file path=opt/libc-tests/tests/random/inz_region mode=0555
 133 file path=opt/libc-tests/tests/random/inz_split mode=0555
 134 file path=opt/libc-tests/tests/random/inz_split_vpp mode=0555
 135 file path=opt/libc-tests/tests/random/inz_vpp mode=0555
 136 file path=opt/libc-tests/tests/regex/data/basic.dat mode=0444
 137 file path=opt/libc-tests/tests/regex/data/basic.out mode=0444
 138 file path=opt/libc-tests/tests/regex/data/categorize.dat mode=0444
 139 file path=opt/libc-tests/tests/regex/data/categorize.out mode=0444
 140 file path=opt/libc-tests/tests/regex/data/forcedassoc.dat mode=0444
 141 file path=opt/libc-tests/tests/regex/data/forcedassoc.out mode=0444
 142 file path=opt/libc-tests/tests/regex/data/leftassoc.dat mode=0444
 143 file path=opt/libc-tests/tests/regex/data/leftassoc.out mode=0444
 144 file path=opt/libc-tests/tests/regex/data/nullsubexpr.dat mode=0444
 145 file path=opt/libc-tests/tests/regex/data/nullsubexpr.out mode=0444
 146 file path=opt/libc-tests/tests/regex/data/repetition.dat mode=0444
 147 file path=opt/libc-tests/tests/regex/data/repetition.out mode=0444
 148 file path=opt/libc-tests/tests/regex/data/rightassoc.dat mode=0444
 149 file path=opt/libc-tests/tests/regex/data/rightassoc.out mode=0444
 150 file path=opt/libc-tests/tests/regex/regex_test mode=0555
 151 file path=opt/libc-tests/tests/regex/testregex mode=0555
 152 file path=opt/libc-tests/tests/select/select.sh mode=0555
 153 file path=opt/libc-tests/tests/select/select_test mode=0555
 154 file path=opt/libc-tests/tests/set_constraint_handler_s.32 mode=0555
 155 file path=opt/libc-tests/tests/set_constraint_handler_s.64 mode=0555
 156 file path=opt/libc-tests/tests/strcoll-strxfrm-6907.32 mode=0555
 157 file path=opt/libc-tests/tests/strcoll-strxfrm-6907.64 mode=0555
 158 file path=opt/libc-tests/tests/strerror mode=0555
 159 file path=opt/libc-tests/tests/symbols/setup mode=0555
 160 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH) mode=0555
 161 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH64) mode=0555
 162 file path=opt/libc-tests/tests/thread_name mode=0555
 163 file path=opt/libc-tests/tests/timespec_get.32 mode=0555
 164 file path=opt/libc-tests/tests/timespec_get.64 mode=0555
 165 file path=opt/libc-tests/tests/wcsncasecmp-7344.32 mode=0555
 166 file path=opt/libc-tests/tests/wcsncasecmp-7344.64 mode=0555
 167 file path=opt/libc-tests/tests/wcsncasecmp-7350.32 mode=0555
 168 file path=opt/libc-tests/tests/wcsncasecmp-7350.64 mode=0555
 169 file path=opt/libc-tests/tests/wcsncasecmp.32 mode=0555
 170 file path=opt/libc-tests/tests/wcsncasecmp.64 mode=0555
 171 file path=opt/libc-tests/tests/wcsrtombs_test mode=0555
 172 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH) mode=0555
 173 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH64) mode=0555
 174 file path=opt/libc-tests/tests/wctype_test mode=0555
 175 file path=opt/libc-tests/tests/wctype_test.$(ARCH) mode=0555
 176 file path=opt/libc-tests/tests/wctype_test.$(ARCH64) mode=0555
 177 file path=usr/lib/locale/zz_AA.UTF-8/LC_COLLATE/LCL_DATA mode=0444
 178 file path=usr/lib/locale/zz_AA.UTF-8/LC_CTYPE/LCL_DATA mode=0444
 179 file path=usr/lib/locale/zz_AA.UTF-8/LC_MESSAGES/LCL_DATA mode=0444
 180 file path=usr/lib/locale/zz_AA.UTF-8/LC_MESSAGES/SUNW_OST_OSLIB.mo mode=0444
 181 file path=usr/lib/locale/zz_AA.UTF-8/LC_MESSAGES/priv_names mode=0444
 182 file path=usr/lib/locale/zz_AA.UTF-8/LC_MONETARY/LCL_DATA mode=0444
 183 file path=usr/lib/locale/zz_AA.UTF-8/LC_NUMERIC/LCL_DATA mode=0444
 184 file path=usr/lib/locale/zz_AA.UTF-8/LC_TIME/LCL_DATA mode=0444
 185 hardlink path=opt/libc-tests/tests/symbols/assert_h target=setup
 186 hardlink path=opt/libc-tests/tests/symbols/ctype_h target=setup
 187 hardlink path=opt/libc-tests/tests/symbols/dirent_h target=setup
 188 hardlink path=opt/libc-tests/tests/symbols/fcntl_h target=setup
 189 hardlink path=opt/libc-tests/tests/symbols/locale_h target=setup
 190 hardlink path=opt/libc-tests/tests/symbols/math_h target=setup
 191 hardlink path=opt/libc-tests/tests/symbols/netdb_h target=setup
 192 hardlink path=opt/libc-tests/tests/symbols/pthread_h target=setup
```



```
 193 hardlink path=opt/libc-tests/tests/symbols/signal_h target=setup
 194 hardlink path=opt/libc-tests/tests/symbols/stdalign_h target=setup
 195 hardlink path=opt/libc-tests/tests/symbols/stddef_h target=setup
 196 hardlink path=opt/libc-tests/tests/symbols/stdio_h target=setup
 197 hardlink path=opt/libc-tests/tests/symbols/stdlib_h target=setup
 198 hardlink path=opt/libc-tests/tests/symbols/stdnoreturn_h target=setup
 199 hardlink path=opt/libc-tests/tests/symbols/string_h target=setup
 200 hardlink path=opt/libc-tests/tests/symbols/strings_h target=setup
 201 hardlink path=opt/libc-tests/tests/symbols/sys_stat_h target=setup
 202 hardlink path=opt/libc-tests/tests/symbols/sys_time_h target=setup
 203 hardlink path=opt/libc-tests/tests/symbols/sys_timeb_h target=setup
 204 hardlink path=opt/libc-tests/tests/symbols/threads_h target=setup
 205 hardlink path=opt/libc-tests/tests/symbols/time_h target=setup
 206 hardlink path=opt/libc-tests/tests/symbols/ucontext_h target=setup
 207 hardlink path=opt/libc-tests/tests/symbols/unistd_h target=setup
 208 hardlink path=opt/libc-tests/tests/symbols/wchar_h target=setup
 209 hardlink path=opt/libc-tests/tests/symbols/wctype_h target=setup
 210 license lic_CDDL license=lic_CDDL
 211 license usr/src/test/libc-tests/tests/regex/THIRDPARTYLICENSE \
 212     license=usr/src/test/libc-tests/tests/regex/THIRDPARTYLICENSE
 213 depend fmri=locale/ar type=require
 214 depend fmri=locale/de type=require
 215 depend fmri=locale/en type=require
 216 depend fmri=locale/en-extra type=require
 217 depend fmri=locale/ja type=require
 218 depend fmri=locale/ru type=require
 219 depend fmri=system/test/testrunner type=require
```

```
**********************************************************
    3619 Mon Oct 15 13:28:40 2018
new/usr/src/test/libc-tests/runfiles/default.run
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet at
   9 # http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright (c) 2012 by Delphix. All rights reserved.
  14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  15 # Copyright 2018 Joyent, Inc.
  15 # Copyright 2017 Joyent, Inc.
  16 #

  18 [DEFAULT]
  19 pre =
  20 verbose = False
  21 quiet = False
  22 timeout = 60
  23 post =
  24 outputdir = /var/tmp/test_results

  26 [/opt/libc-tests/tests/fpround_test]
  27 [/opt/libc-tests/tests/newlocale_test]
  28 [/opt/libc-tests/tests/nl_langinfo_test]
  29 [/opt/libc-tests/tests/wcsrtombs_test]
  30 [/opt/libc-tests/tests/wctype_test]

  32 [/opt/libc-tests/tests/strcoll-strxfrm-6907.32]
  33 [/opt/libc-tests/tests/strcoll-strxfrm-6907.64]
  34 [/opt/libc-tests/tests/wcsncasecmp.32]
  35 [/opt/libc-tests/tests/wcsncasecmp.64]
  36 [/opt/libc-tests/tests/wcsncasecmp-7344.32]
  37 [/opt/libc-tests/tests/wcsncasecmp-7344.64]
  38 [/opt/libc-tests/tests/wcsncasecmp-7350.32]
  39 [/opt/libc-tests/tests/wcsncasecmp-7350.64]

  41 [/opt/libc-tests/tests/random/getrandom]
  42 [/opt/libc-tests/tests/random/getentropy]
  43 [/opt/libc-tests/tests/random/chacha]
  44 [/opt/libc-tests/tests/random/inz_child]
  45 [/opt/libc-tests/tests/random/inz_inval]
  46 [/opt/libc-tests/tests/random/inz_mlock]
  47 [/opt/libc-tests/tests/random/inz_region]
  48 [/opt/libc-tests/tests/random/inz_split]
  49 [/opt/libc-tests/tests/random/inz_split_vpp]
  50 [/opt/libc-tests/tests/random/inz_vpp]
  51 [/opt/libc-tests/tests/random/arc4random]
  52 [/opt/libc-tests/tests/random/arc4random_fork]
  53 [/opt/libc-tests/tests/random/arc4random_prefork]
  54 [/opt/libc-tests/tests/random/arc4random_forkall]
  55 [/opt/libc-tests/tests/random/arc4random_preforkall]
  56 [/opt/libc-tests/tests/random/arc4random_forksig]
  57 [/opt/libc-tests/tests/random/arc4random_preforksig]
  58 [/opt/libc-tests/tests/random/arc4key.ksh]
```



```
  60 [/opt/libc-tests/tests/regex/regex_test]

  62 [/opt/libc-tests/tests/select/select.sh]
  63 timeout = 600

  65 [/opt/libc-tests/tests/aligned_alloc.32]
  66 [/opt/libc-tests/tests/aligned_alloc.64]
  67 [/opt/libc-tests/tests/c11_threads.32]
  68 [/opt/libc-tests/tests/c11_threads.64]
  69 [/opt/libc-tests/tests/c11_tss.32]
  70 [/opt/libc-tests/tests/c11_tss.64]
  71 [/opt/libc-tests/tests/call_once.32]
  72 [/opt/libc-tests/tests/call_once.64]
  73 [/opt/libc-tests/tests/catopen]
  74 [/opt/libc-tests/tests/endian.32]
  75 [/opt/libc-tests/tests/endian.64]
  76 [/opt/libc-tests/tests/env-7076.32]
  77 [/opt/libc-tests/tests/env-7076.64]
  78 [/opt/libc-tests/tests/fnmatch.32]
  79 [/opt/libc-tests/tests/fnmatch.64]
  80 [/opt/libc-tests/tests/memset_s.32]
  81 [/opt/libc-tests/tests/memset_s.64]
  82 [/opt/libc-tests/tests/printf-6961.64]
  83 [/opt/libc-tests/tests/printf-9511.32]
  84 [/opt/libc-tests/tests/printf-9511.64]
  85 [/opt/libc-tests/tests/priv_gettext]
  86 [/opt/libc-tests/tests/psignal]
  87 [/opt/libc-tests/tests/quick_exit]
  88 [/opt/libc-tests/tests/set_constraint_handler_s.32]
  89 [/opt/libc-tests/tests/set_constraint_handler_s.64]
  90 [/opt/libc-tests/tests/strerror]
  91 [/opt/libc-tests/tests/thread_name]
  92 [/opt/libc-tests/tests/timespec_get.32]
  93 [/opt/libc-tests/tests/timespec_get.64]

  95 [/opt/libc-tests/tests/pthread_attr_get_np]

  97 [/opt/libc-tests/tests/symbols]
  98 pre = setup
  99 tests = [
 100         'assert_h',
 101         'ctype_h',
 102         'dirent_h',
 103         'fcntl_h',
 104         'locale_h',
 105         'math_h',
 106         'netdb_h',
 107         'pthread_h',
 108         'signal_h',
 109         'stdalign_h',
 110         'stddef_h',
 111         'stdio_h',
 112         'stdlib_h',
 113         'stdnoreturn_h',
 114         'string_h',
 115         'strings_h',
 116         'sys_stat_h',
 117         'sys_time_h',
 118         'sys_timeb_h',
 119         'time_h',
 120         'threads_h',
 121         'ucontext_h',
 122         'unistd_h',
 123         'wchar_h',
 124         'wctype_h'
 125          ]
```

****
940 Mon Oct 15 13:28:40 2018
new/usr/src/test/libc-tests/tests/threads/Makefile
8158 Want named threads API
9857 proc manpages should have LIBRARY section
****

```
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2018 Joyent, Inc.
# Copyright 2016 Joyent, Inc.
#

include $(SRC)/Makefile.master

ROOTOPTPKG = $(ROOT)/opt/libc-tests
TESTDIR = $(ROOTOPTPKG)/tests

PROGS = pthread_attr_get_np thread_name
PROGS = pthread_attr_get_np

include $(SRC)/cmd/Makefile.cmd
include $(SRC)/test/Makefile.com

CPPFLAGS += -D_REENTRANT

CMDS = $(PROGS:%=$(TESTDIR)/%)
$(CMDS) := FILEMODE = 0555

all: $(PROGS)

install: all $(CMDS)

lint:

clobber: clean
        -$(RM) $(PROGS)

clean:
        -$(RM) *.o

$(CMDS): $(TESTDIR) $(PROGS)

$(TESTDIR):
        $(INS.dir)

$(TESTDIR)/%: %
        $(INS.file)

$(TESTDIR)/%: %
        $(INS.file)
```

```
**********************************************************
    6508 Mon Oct 15 13:28:40 2018
new/usr/src/test/libc-tests/tests/threads/thread_name.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2018 Joyent, Inc.
 */

/*
 * Some basic pthread name API tests.
 */

#include <sys/stat.h>
#include <pthread.h>
#include <limits.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <thread.h>
#include <fcntl.h>
#include <stdio.h>
#include <errno.h>
#include <err.h>


/*ARGSUSED*/
static void *
thr(void *unused)
{
        (void) sleep(100);
        return (NULL);
}

/*ARGSUSED*/
int
main(int argc, char *argv[])
{
        char name[PTHREAD_MAX_NAMELEN_NP];
        pthread_attr_t attr;
        char path[PATH_MAX];
        pthread_t tid;
        ssize_t n;
        int test;
        int rc;
        int fd;

        /* Default thread name is empty string. */
        test = 1;

        rc = pthread_getname_np(pthread_self(), name, sizeof (name));

        if (rc != 0 || strcmp(name, "") != 0)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);
```



```
        /* Can set name. */
        test = 2;

        (void) strlcpy(name, "main", sizeof (name));
        rc = pthread_setname_np(pthread_self(), name);

        if (rc != 0)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        rc = pthread_getname_np(pthread_self(), name, sizeof (name));

        if (rc != 0 || strcmp(name, "main") != 0)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        /* ERANGE check. */
        test = 3;

        rc = pthread_getname_np(pthread_self(), name, 2);

        if (rc != ERANGE)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        /* EINVAL check. */
        test = 4;

        rc = pthread_getname_np(pthread_self(), NULL, sizeof (name));

        if (rc != EINVAL)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        /* can clear thread name. */
        test = 5;

        rc = pthread_setname_np(pthread_self(), NULL);

        if (rc != 0)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        rc = pthread_getname_np(pthread_self(), name, sizeof (name));

        if (rc != 0 || strcmp(name, "") != 0)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        /* non-existent thread check. */
        test = 6;

        rc = pthread_getname_np(808, name, sizeof (name));

        if (rc != ESRCH)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        rc = pthread_setname_np(808, "state");

        if (rc != ESRCH)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

        /* too long a name. */
        test = 7;

        rc = pthread_setname_np(pthread_self(),
            "12345678901234567890123456789012");

        if (rc != ERANGE)
                errx(EXIT_FAILURE, "test %d failed with %d", test, rc);
```

```
127         /* can name another thread. */
128         test = 8;

130         rc = pthread_create(&tid, NULL, thr, NULL);

132         if (rc != 0)
133                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

135         rc = pthread_setname_np(tid, "otherthread");

137         if (rc != 0)
138                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

140         /* attr tests. */
141         test = 9;

143         (void) pthread_attr_init(&attr);

145         rc = pthread_attr_setname_np(&attr,
146             "12345678901234567890123456789012");

148         if (rc != ERANGE)
149                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

151         rc = pthread_attr_setname_np(&attr, "thread2");

153         if (rc != 0)
154                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

156         rc = pthread_attr_getname_np(&attr, NULL, sizeof (name));

158         if (rc != EINVAL)
159                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

161         rc = pthread_attr_getname_np(&attr, name, 2);

163         if (rc != ERANGE)
164                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

166         /* does the attr actually apply? */
167         test = 10;

169         rc = pthread_create(&tid, &attr, thr, NULL);

171         if (rc != 0)
172                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

174         rc = pthread_getname_np(tid, name, sizeof (name));

176         if (rc != 0 || strcmp(name, "thread2") != 0)
177                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

179         /* proc read tests */
180         test = 11;

182         (void) snprintf(path, sizeof (path),
183             "/proc/self/lwp/%d/lwpname", (int)tid);

185         fd = open(path, O_RDWR);

187         if (fd == -1)
188                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

190         n = read(fd, name, sizeof (name));

192         if (n != sizeof (name) || strcmp(name, "thread2") != 0)
```



```
193                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

195         if (lseek(fd, 0, SEEK_SET) != 0)
196                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

198         n = read(fd, name, PTHREAD_MAX_NAMELEN_NP * 2);

200         if (n != sizeof (name) || strcmp(name, "thread2") != 0)
201                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

203         if (lseek(fd, 0, SEEK_SET) != 0)
204                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

206         n = read(fd, name, 4);

208         if (n != 4 || strncmp(name, "thre", 4) != 0)
209                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

211         /* proc write tests */
212         test = 12;

214         if (lseek(fd, 0, SEEK_SET) != 0)
215                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

217         n = write(fd, "12345678901234567890123456789012345678901",
218             PTHREAD_MAX_NAMELEN_NP);

220         if (n != PTHREAD_MAX_NAMELEN_NP)
221                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

223         if (lseek(fd, 0, SEEK_SET) != 0)
224                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

226         n = write(fd, "foo", sizeof ("foo"));

228         if (n != sizeof ("foo"))
229                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

231         if (lseek(fd, 0, SEEK_SET) != 0)
232                 errx(EXIT_FAILURE, "test %d failed with %d", test, errno);

234         n = read(fd, name, sizeof (name));

236         if (n != sizeof (name) || strcmp(name, "foo") != 0)
237                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

239         (void) close(fd);

241         /* thr_* API. */
242         test = 13;

244         rc = thr_setname(thr_self(), "main");

246         if (rc != 0)
247                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

249         rc = thr_getname(thr_self(), name, sizeof (name));

251         if (rc != 0 || strcmp(name, "main") != 0)
252                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

254         /* badness */
255         test = 14;

257         rc = thr_setname(thr_self(), "\033]0;messeduptitle\a");
```

```
259         if (rc != EINVAL)
260                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

262         rc = thr_setname(thr_self(), "ab\177\177\n");

264         if (rc != EINVAL)
265                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

267         rc = pthread_attr_setname_np(&attr, "\033]0;messeduptitle\a");

269         if (rc != EINVAL)
270                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

272         rc = pthread_attr_setname_np(&attr, "ab\177\177\n");

274         if (rc != EINVAL)
275                 errx(EXIT_FAILURE, "test %d failed with %d", test, rc);

277         return (EXIT_SUCCESS);
278 }
```

```
**********************************************************
   55583 Mon Oct 15 13:28:40 2018
new/usr/src/uts/common/disp/thread.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2018 Joyent, Inc.
 * Copyright (c) 2013, Joyent, Inc.  All rights reserved.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/sysmacros.h>
#include <sys/signal.h>
#include <sys/stack.h>
#include <sys/pcb.h>
#include <sys/user.h>
#include <sys/systm.h>
#include <sys/sysinfo.h>
#include <sys/errno.h>
#include <sys/cmn_err.h>
#include <sys/cred.h>
#include <sys/resource.h>
#include <sys/task.h>
#include <sys/project.h>
#include <sys/proc.h>
#include <sys/debug.h>
#include <sys/disp.h>
#include <sys/class.h>
#include <vm/seg_kmem.h>
#include <vm/seg_kp.h>
#include <sys/machlock.h>
#include <sys/kmem.h>
#include <sys/varargs.h>
#include <sys/turnstile.h>
#include <sys/poll.h>
#include <sys/vtrace.h>
#include <sys/callb.h>
#include <c2/audit.h>
#include <sys/tnf.h>
#include <sys/sobject.h>
#include <sys/cpupart.h>
#include <sys/pset.h>
```



```
#include <sys/door.h>
#include <sys/spl.h>
#include <sys/copyops.h>
#include <sys/rctl.h>
#include <sys/brand.h>
#include <sys/pool.h>
#include <sys/zone.h>
#include <sys/tsol/label.h>
#include <sys/tsol/tndb.h>
#include <sys/cpc_impl.h>
#include <sys/sdt.h>
#include <sys/reboot.h>
#include <sys/kdi.h>
#include <sys/schedctl.h>
#include <sys/waitq.h>
#include <sys/cpucaps.h>
#include <sys/kiconv.h>
#include <sys/ctype.h>

struct kmem_cache *thread_cache;        /* cache of free threads */
struct kmem_cache *lwp_cache;           /* cache of free lwps */
struct kmem_cache *turnstile_cache;     /* cache of free turnstiles */

/*
 * allthreads is only for use by kmem_readers.  All kernel loops can use
 * the current thread as a start/end point.
 */
kthread_t *allthreads = &t0;    /* circular list of all threads */

static kcondvar_t reaper_cv;            /* synchronization var */
kthread_t       *thread_deathrow;       /* circular list of reapable threads */
kthread_t       *lwp_deathrow;          /* circular list of reapable threads */
kmutex_t        reaplock;               /* protects lwp and thread deathrows */
int     thread_reapcnt = 0;             /* number of threads on deathrow */
int     lwp_reapcnt = 0;                /* number of lwps on deathrow */
int     reaplimit = 16;                 /* delay reaping until reaplimit */

thread_free_lock_t      *thread_free_lock;
                                        /* protects tick thread from reaper */

extern int nthread;

/* System Scheduling classes. */
id_t    syscid;                         /* system scheduling class ID */
id_t    sysdccid = CLASS_UNUSED;        /* reset when SDC loads */

void    *segkp_thread;                  /* cookie for segkp pool */

int lwp_cache_sz = 32;
int t_cache_sz = 8;
static kt_did_t next_t_id = 1;

/* Default mode for thread binding to CPUs and processor sets */
int default_binding_mode = TB_ALLHARD;

/*
 * Min/Max stack sizes for stack size parameters
 */
#define MAX_STKSIZE     (32 * DEFAULTSTKSZ)
#define MIN_STKSIZE     DEFAULTSTKSZ

/*
 * default_stksize overrides lwp_default_stksize if it is set.
 */
int     default_stksize;
int     lwp_default_stksize;
```

```
 127 static zone_key_t zone_thread_key;

 129 unsigned int kmem_stackinfo;            /* stackinfo feature on-off */
 130 kmem_stkinfo_t *kmem_stkinfo_log;       /* stackinfo circular log */
 131 static kmutex_t kmem_stkinfo_lock;      /* protects kmem_stkinfo_log */

 133 /*
 134  * forward declarations for internal thread specific data (tsd)
 135  */
 136 static void *tsd_realloc(void *, size_t, size_t);

 138 void thread_reaper(void);

 140 /* forward declarations for stackinfo feature */
 141 static void stkinfo_begin(kthread_t *);
 142 static void stkinfo_end(kthread_t *);
 143 static size_t stkinfo_percent(caddr_t, caddr_t, caddr_t);

 145 /*ARGSUSED*/
 146 static int
 147 turnstile_constructor(void *buf, void *cdrarg, int kmflags)
 148 {
 149         bzero(buf, sizeof (turnstile_t));
 150         return (0);
 151 }
_____unchanged_portion_omitted_

 724 void
 725 thread_free(kthread_t *t)
 726 {
 727         boolean_t allocstk = (t->t_flag & T_TALLOCSTK);
 728         klwp_t *lwp = t->t_lwp;
 729         caddr_t swap = t->t_swap;

 731         ASSERT(t != &t0 && t->t_state == TS_FREE);
 732         ASSERT(t->t_door == NULL);
 733         ASSERT(t->t_schedctl == NULL);
 734         ASSERT(t->t_pollstate == NULL);

 736         t->t_pri = 0;
 737         t->t_pc = 0;
 738         t->t_sp = 0;
 739         t->t_wchan0 = NULL;
 740         t->t_wchan = NULL;
 741         if (t->t_cred != NULL) {
 742                 crfree(t->t_cred);
 743                 t->t_cred = 0;
 744         }
 745         if (t->t_pdmsg) {
 746                 kmem_free(t->t_pdmsg, strlen(t->t_pdmsg) + 1);
 747                 t->t_pdmsg = NULL;
 748         }
 749         if (audit_active)
 750                 audit_thread_free(t);
 751 #ifndef NPROBE
 752         if (t->t_tnf_tpdp)
 753                 tnf_thread_free(t);
 754 #endif /* NPROBE */
 755         if (t->t_cldata) {
 756                 CL_EXITCLASS(t->t_cid, (caddr_t *)t->t_cldata);
 757         }
 758         if (t->t_rprof != NULL) {
 759                 kmem_free(t->t_rprof, sizeof (*t->t_rprof));
 760                 t->t_rprof = NULL;
 761         }
```



```
 762         t->t_lockp = NULL;      /* nothing should try to lock this thread now */
 763         if (lwp)
 764                 lwp_freeregs(lwp, 0);
 765         if (t->t_ctx)
 766                 freectx(t, 0);
 767         t->t_stk = NULL;
 768         if (lwp)
 769                 lwp_stk_fini(lwp);
 770         lock_clear(&t->t_lock);

 772         if (t->t_ts->ts_waiters > 0)
 773                 panic("thread_free: turnstile still active");

 775         kmem_cache_free(turnstile_cache, t->t_ts);

 777         free_afd(&t->t_activefd);

 779         /*
 780          * Barrier for the tick accounting code.  The tick accounting code
 781          * holds this lock to keep the thread from going away while it's
 782          * looking at it.
 783          */
 784         thread_free_barrier(t);

 786         ASSERT(ttoproj(t) == proj0p);
 787         project_rele(ttoproj(t));

 789         lgrp_affinity_free(&t->t_lgrp_affinity);

 791         mutex_enter(&pidlock);
 792         nthread--;
 793         mutex_exit(&pidlock);

 795         if (t->t_name != NULL) {
 796                 kmem_free(t->t_name, THREAD_NAME_MAX);
 797                 t->t_name = NULL;
 798         }

 800         /*
 801          * Free thread, lwp and stack.  This needs to be done carefully, since
 802          * if T_TALLOCSTK is set, the thread is part of the stack.
 803          */
 804         t->t_lwp = NULL;
 805         t->t_swap = NULL;

 807         if (swap) {
 808                 segkp_release(segkp, swap);
 809         }
 810         if (lwp) {
 811                 kmem_cache_free(lwp_cache, lwp);
 812         }
 813         if (!allocstk) {
 814                 kmem_cache_free(thread_cache, t);
 815         }
 816 }
_____unchanged_portion_omitted_

2100 /*
2101  * Tunable kmem_stackinfo is set, compute stack utilization percentage.
2102  */
2103 static size_t
2104 stkinfo_percent(caddr_t t_stk, caddr_t t_stkbase, caddr_t sp)
2105 {
2106         size_t percent;
2107         size_t s;
```

```
2109         if (t_stk > t_stkbase) {
2110                 /* stack grows down */
2111                 if (sp > t_stk) {
2112                         return (0);
2113                 }
2114                 if (sp < t_stkbase) {
2115                         return (100);
2116                 }
2117                 percent = t_stk - sp + 1;
2118                 s = t_stk - t_stkbase + 1;
2119         } else {
2120                 /* stack grows up */
2121                 if (sp < t_stk) {
2122                         return (0);
2123                 }
2124                 if (sp > t_stkbase) {
2125                         return (100);
2126                 }
2127                 percent = sp - t_stk + 1;
2128                 s = t_stkbase - t_stk + 1;
2129         }
2130         percent = ((100 * percent) / s) + 1;
2131         if (percent > 100) {
2132                 percent = 100;
2133         }
2134         return (percent);
2135 }

2137 /*
2138  * NOTE: This will silently truncate a name > THREAD_NAME_MAX - 1 characters
2139  * long.  It is expected that callers (acting on behalf of userland clients)
2140  * will perform any required checks to return the correct error semantics.
2141  * It is also expected callers on behalf of userland clients have done
2142  * any necessary permission checks.
2143  */
2144 int
2145 thread_setname(kthread_t *t, const char *name)
2146 {
2147         char *buf = NULL;

2149         /*
2150          * We optimistically assume that a thread's name will only be set
2151          * once and so allocate memory in preparation of setting t_name.
2152          * If it turns out a name has already been set, we just discard (free)
2153          * the buffer we just allocated and reuse the current buffer
2154          * (as all should be THREAD_NAME_MAX large).
2155          *
2156          * Such an arrangement means over the lifetime of a kthread_t, t_name
2157          * is either NULL or has one value (the address of the buffer holding
2158          * the current thread name).   The assumption is that most kthread_t
2159          * instances will not have a name assigned, so dynamically allocating
2160          * the memory should minimize the footprint of this feature, but by
2161          * having the buffer persist for the life of the thread, it simplifies
2162          * usage in highly constrained situations (e.g. dtrace).
2163          */
2164         if (name != NULL && name[0] != '\0') {
2165                 for (size_t i = 0; name[i] != '\0'; i++) {
2166                         if (!isprint(name[i]))
2167                                 return (EINVAL);
2168                 }

2170                 buf = kmem_zalloc(THREAD_NAME_MAX, KM_SLEEP);
2171                 (void) strlcpy(buf, name, THREAD_NAME_MAX);
2172         }

2174         mutex_enter(&ttoproc(t)->p_lock);
```



```
2175         if (t->t_name == NULL) {
2176                 t->t_name = buf;
2177         } else {
2178                 if (buf != NULL) {
2179                         (void) strlcpy(t->t_name, name, THREAD_NAME_MAX);
2180                         kmem_free(buf, THREAD_NAME_MAX);
2181                 } else {
2182                         bzero(t->t_name, THREAD_NAME_MAX);
2183                 }
2184         }
2185         mutex_exit(&ttoproc(t)->p_lock);
2186         return (0);
2187 }

2189 int
2190 thread_vsetname(kthread_t *t, const char *fmt, ...)
2191 {
2192         char name[THREAD_NAME_MAX];
2193         va_list va;
2194         int rc;

2196         va_start(va, fmt);
2197         rc = vsnprintf(name, sizeof (name), fmt, va);
2198         va_end(va);

2200         if (rc < 0)
2201                 return (EINVAL);

2203         if (rc >= sizeof (name))
2204                 return (ENAMETOOLONG);

2206         return (thread_setname(t, name));
2207 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
  448963 Mon Oct 15 13:28:40 2018
new/usr/src/uts/common/dtrace/dtrace.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2018, Joyent, Inc.
 * Copyright (c) 2017, Joyent, Inc.
 * Copyright (c) 2012, 2014 by Delphix. All rights reserved.
 */

/*
 * DTrace - Dynamic Tracing for Solaris
 *
 * This is the implementation of the Solaris Dynamic Tracing framework
 * (DTrace).  The user-visible interface to DTrace is described at length in
 * the "Solaris Dynamic Tracing Guide".  The interfaces between the libdtrace
 * library, the in-kernel DTrace framework, and the DTrace providers are
 * described in the block comments in the <sys/dtrace.h> header file.  The
 * internal architecture of DTrace is described in the block comments in the
 * <sys/dtrace_impl.h> header file.  The comments contained within the DTrace
 * implementation very much assume mastery of all of these sources; if one has
 * an unanswered question about the implementation, one should consult them
 * first.
 *
 * The functions here are ordered roughly as follows:
 *
 *   - Probe context functions
 *   - Probe hashing functions
 *   - Non-probe context utility functions
 *   - Matching functions
 *   - Provider-to-Framework API functions
 *   - Probe management functions
 *   - DIF object functions
 *   - Format functions
 *   - Predicate functions
 *   - ECB functions
 *   - Buffer functions
 *   - Enabling functions
 *   - DOF functions
 *   - Anonymous enabling functions
 *   - Consumer state functions
 *   - Helper functions
```



```
 *   - Hook functions
 *   - Driver cookbook functions
 *
 * Each group of functions begins with a block comment labelled the "DTrace
 * [Group] Functions", allowing one to find each block by searching forward
 * on capital-f functions.
 */
#include <sys/errno.h>
#include <sys/stat.h>
#include <sys/modctl.h>
#include <sys/conf.h>
#include <sys/systm.h>
#include <sys/ddi.h>
#include <sys/sunddi.h>
#include <sys/cpuvar.h>
#include <sys/kmem.h>
#include <sys/strsubr.h>
#include <sys/sysmacros.h>
#include <sys/dtrace_impl.h>
#include <sys/atomic.h>
#include <sys/cmn_err.h>
#include <sys/mutex_impl.h>
#include <sys/rwlock_impl.h>
#include <sys/ctf_api.h>
#include <sys/panic.h>
#include <sys/priv_impl.h>
#include <sys/policy.h>
#include <sys/cred_impl.h>
#include <sys/procfs_isa.h>
#include <sys/taskq.h>
#include <sys/mkdev.h>
#include <sys/kdi.h>
#include <sys/zone.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include "strtolctype.h"

/*
 * DTrace Tunable Variables
 *
 * The following variables may be tuned by adding a line to /etc/system that
 * includes both the name of the DTrace module ("dtrace") and the name of the
 * variable.  For example:
 *
 *   set dtrace:dtrace_destructive_disallow = 1
 *
 * In general, the only variables that one should be tuning this way are those
 * that affect system-wide DTrace behavior, and for which the default behavior
 * is undesirable.  Most of these variables are tunable on a per-consumer
 * basis using DTrace options, and need not be tuned on a system-wide basis.
 * When tuning these variables, avoid pathological values; while some attempt
 * is made to verify the integrity of these variables, they are not considered
 * part of the supported interface to DTrace, and they are therefore not
 * checked comprehensively.  Further, these variables should not be tuned
 * dynamically via "mdb -kw" or other means; they should only be tuned via
 * /etc/system.
 */
int             dtrace_destructive_disallow = 0;
dtrace_optval_t dtrace_nonroot_maxsize = (16 * 1024 * 1024);
size_t          dtrace_difo_maxsize = (256 * 1024);
dtrace_optval_t dtrace_dof_maxsize = (8 * 1024 * 1024);
size_t          dtrace_statvar_maxsize = (16 * 1024);
size_t          dtrace_actions_max = (16 * 1024);
size_t          dtrace_retain_max = 1024;
dtrace_optval_t dtrace_helper_actions_max = 1024;
dtrace_optval_t dtrace_helper_providers_max = 32;
```

```
126 dtrace_optval_t dtrace_dstate_defsize = (1 * 1024 * 1024);
127 size_t          dtrace_strsize_default = 256;
128 dtrace_optval_t dtrace_cleanrate_default = 9900990;             /* 101 hz */
129 dtrace_optval_t dtrace_cleanrate_min = 200000;                  /* 5000 hz */
130 dtrace_optval_t dtrace_cleanrate_max = (uint64_t)60 * NANOSEC;  /* 1/minute */
131 dtrace_optval_t dtrace_aggrate_default = NANOSEC;               /* 1 hz */
132 dtrace_optval_t dtrace_statusrate_default = NANOSEC;            /* 1 hz */
133 dtrace_optval_t dtrace_statusrate_max = (hrtime_t)10 * NANOSEC;  /* 6/minute */
134 dtrace_optval_t dtrace_switchrate_default = NANOSEC;            /* 1 hz */
135 dtrace_optval_t dtrace_nspec_default = 1;
136 dtrace_optval_t dtrace_specsize_default = 32 * 1024;
137 dtrace_optval_t dtrace_stackframes_default = 20;
138 dtrace_optval_t dtrace_ustackframes_default = 20;
139 dtrace_optval_t dtrace_jstackframes_default = 50;
140 dtrace_optval_t dtrace_jstackstrsize_default = 512;
141 int             dtrace_msgdsize_max = 128;
142 hrtime_t        dtrace_chill_max = MSEC2NSEC(500);              /* 500 ms */
143 hrtime_t        dtrace_chill_interval = NANOSEC;                /* 1000 ms */
144 int             dtrace_devdepth_max = 32;
145 int             dtrace_err_verbose;
146 hrtime_t        dtrace_deadman_interval = NANOSEC;
147 hrtime_t        dtrace_deadman_timeout = (hrtime_t)10 * NANOSEC;
148 hrtime_t        dtrace_deadman_user = (hrtime_t)30 * NANOSEC;
149 hrtime_t        dtrace_unregister_defunct_reap = (hrtime_t)60 * NANOSEC;

151 /*
152  * DTrace External Variables
153  *
154  * As dtrace(7D) is a kernel module, any DTrace variables are obviously
155  * available to DTrace consumers via the backtick (`) syntax.  One of these,
156  * dtrace_zero, is made deliberately so:  it is provided as a source of
157  * well-known, zero-filled memory.  While this variable is not documented,
158  * it is used by some translators as an implementation detail.
159  */
160 const char      dtrace_zero[256] = { 0 };       /* zero-filled memory */

162 /*
163  * DTrace Internal Variables
164  */
165 static dev_info_t       *dtrace_devi;           /* device info */
166 static vmem_t           *dtrace_arena;          /* probe ID arena */
167 static vmem_t           *dtrace_minor;          /* minor number arena */
168 static taskq_t          *dtrace_taskq;          /* task queue */
169 static dtrace_probe_t   **dtrace_probes;        /* array of all probes */
170 static int              dtrace_nprobes;         /* number of probes */
171 static dtrace_provider_t *dtrace_provider;      /* provider list */
172 static dtrace_meta_t    *dtrace_meta_pid;       /* user-land meta provider */
173 static int              dtrace_opens;           /* number of opens */
174 static int              dtrace_helpers;         /* number of helpers */
175 static int              dtrace_getf;            /* number of unpriv getf()s */
176 static void             *dtrace_softstate;      /* softstate pointer */
177 static dtrace_hash_t    *dtrace_bymod;          /* probes hashed by module */
178 static dtrace_hash_t    *dtrace_byfunc;         /* probes hashed by function */
179 static dtrace_hash_t    *dtrace_byname;         /* probes hashed by name */
180 static dtrace_toxrange_t *dtrace_toxrange;      /* toxic range array */
181 static int              dtrace_toxranges;       /* number of toxic ranges */
182 static int              dtrace_toxranges_max;   /* size of toxic range array */
183 static dtrace_anon_t    dtrace_anon;            /* anonymous enabling */
184 static kmem_cache_t     *dtrace_state_cache;    /* cache for dynamic state */
185 static uint64_t         dtrace_vtime_references; /* number of vtimestamp refs */
186 static kthread_t        *dtrace_panicked;       /* panicking thread */
187 static dtrace_ecb_t     *dtrace_ecb_create_cache; /* cached created ECB */
188 static dtrace_genid_t   dtrace_probegen;        /* current probe generation */
189 static dtrace_helpers_t *dtrace_deferred_pid;   /* deferred helper list */
190 static dtrace_enabling_t *dtrace_retained;      /* list of retained enablings */
191 static dtrace_genid_t   dtrace_retained_gen;    /* current retained enab gen */
```



```
192 static dtrace_dynvar_t  dtrace_dynhash_sink;    /* end of dynamic hash chains */
193 static int              dtrace_dynvar_failclean; /* dynvars failed to clean */

195 /*
196  * DTrace Locking
197  * DTrace is protected by three (relatively coarse-grained) locks:
198  *
199  * (1) dtrace_lock is required to manipulate essentially any DTrace state,
200  *     including enabling state, probes, ECBs, consumer state, helper state,
201  *     etc.  Importantly, dtrace_lock is _not_ required when in probe context;
202  *     probe context is lock-free -- synchronization is handled via the
203  *     dtrace_sync() cross call mechanism.
204  *
205  * (2) dtrace_provider_lock is required when manipulating provider state, or
206  *     when provider state must be held constant.
207  *
208  * (3) dtrace_meta_lock is required when manipulating meta provider state, or
209  *     when meta provider state must be held constant.
210  *
211  * The lock ordering between these three locks is dtrace_meta_lock before
212  * dtrace_provider_lock before dtrace_lock.  (In particular, there are
213  * several places where dtrace_provider_lock is held by the framework as it
214  * calls into the providers -- which then call back into the framework,
215  * grabbing dtrace_lock.)
216  *
217  * There are two other locks in the mix:  mod_lock and cpu_lock.  With respect
218  * to dtrace_provider_lock and dtrace_lock, cpu_lock continues its historical
219  * role as a coarse-grained lock; it is acquired before both of these locks.
220  * With respect to dtrace_meta_lock, its behavior is stranger:  cpu_lock must
221  * be acquired _between_ dtrace_meta_lock and any other DTrace locks.
222  * mod_lock is similar with respect to dtrace_provider_lock in that it must be
223  * acquired _between_ dtrace_provider_lock and dtrace_lock.
224  */
225 static kmutex_t         dtrace_lock;            /* probe state lock */
226 static kmutex_t         dtrace_provider_lock;   /* provider state lock */
227 static kmutex_t         dtrace_meta_lock;       /* meta-provider state lock */

229 /*
230  * DTrace Provider Variables
231  *
232  * These are the variables relating to DTrace as a provider (that is, the
233  * provider of the BEGIN, END, and ERROR probes).
234  */
235 static dtrace_pattr_t   dtrace_provider_attr = {
236 { DTRACE_STABILITY_STABLE, DTRACE_STABILITY_STABLE, DTRACE_CLASS_COMMON },
237 { DTRACE_STABILITY_PRIVATE, DTRACE_STABILITY_PRIVATE, DTRACE_CLASS_UNKNOWN },
238 { DTRACE_STABILITY_PRIVATE, DTRACE_STABILITY_PRIVATE, DTRACE_CLASS_UNKNOWN },
239 { DTRACE_STABILITY_STABLE, DTRACE_STABILITY_STABLE, DTRACE_CLASS_COMMON },
240 { DTRACE_STABILITY_STABLE, DTRACE_STABILITY_STABLE, DTRACE_CLASS_COMMON },
241 };
______unchanged_portion_omitted_

3147 /*
3148  * This function implements the DIF emulator's variable lookups.  The emulator
3149  * passes a reserved variable identifier and optional built-in array index.
3150  */
3151 static uint64_t
3152 dtrace_dif_variable(dtrace_mstate_t *mstate, dtrace_state_t *state, uint64_t v,
3153     uint64_t ndx)
3154 {
3155         /*
3156          * If we're accessing one of the uncached arguments, we'll turn this
3157          * into a reference in the args array.
3158          */
3159         if (v >= DIF_VAR_ARG0 && v <= DIF_VAR_ARG9) {
3160                 ndx = v - DIF_VAR_ARG0;
```

```
3161                 v = DIF_VAR_ARGS;
3162         }

3164         switch (v) {
3165         case DIF_VAR_ARGS:
3166                 if (!(mstate->dtms_access & DTRACE_ACCESS_ARGS)) {
3167                         cpu_core[CPU->cpu_id].cpuc_dtrace_flags |=
3168                             CPU_DTRACE_KPRIV;
3169                         return (0);
3170                 }

3172                 ASSERT(mstate->dtms_present & DTRACE_MSTATE_ARGS);
3173                 if (ndx >= sizeof (mstate->dtms_arg) /
3174                     sizeof (mstate->dtms_arg[0])) {
3175                         int aframes = mstate->dtms_probe->dtpr_aframes + 2;
3176                         dtrace_provider_t *pv;
3177                         uint64_t val;

3179                         pv = mstate->dtms_probe->dtpr_provider;
3180                         if (pv->dtpv_pops.dtps_getargval != NULL)
3181                                 val = pv->dtpv_pops.dtps_getargval(pv->dtpv_arg,
3182                                     mstate->dtms_probe->dtpr_id,
3183                                     mstate->dtms_probe->dtpr_arg, ndx, aframes);
3184                         else
3185                                 val = dtrace_getarg(ndx, aframes);

3187                         /*
3188                          * This is regrettably required to keep the compiler
3189                          * from tail-optimizing the call to dtrace_getarg().
3190                          * The condition always evaluates to true, but the
3191                          * compiler has no way of figuring that out a priori.
3192                          * (None of this would be necessary if the compiler
3193                          * could be relied upon to _always_ tail-optimize
3194                          * the call to dtrace_getarg() -- but it can't.)
3195                          */
3196                         if (mstate->dtms_probe != NULL)
3197                                 return (val);

3199                         ASSERT(0);
3200                 }

3202                 return (mstate->dtms_arg[ndx]);

3204         case DIF_VAR_UREGS: {
3205                 klwp_t *lwp;

3207                 if (!dtrace_priv_proc(state, mstate))
3208                         return (0);

3210                 if ((lwp = curthread->t_lwp) == NULL) {
3211                         DTRACE_CPUFLAG_SET(CPU_DTRACE_BADADDR);
3212                         cpu_core[CPU->cpu_id].cpuc_dtrace_illval = NULL;
3213                         return (0);
3214                 }

3216                 return (dtrace_getreg(lwp->lwp_regs, ndx));
3217         }

3219         case DIF_VAR_VMREGS: {
3220                 uint64_t rval;

3222                 if (!dtrace_priv_kernel(state))
3223                         return (0);

3225                 DTRACE_CPUFLAG_SET(CPU_DTRACE_NOFAULT);
```



```
3227                 rval = dtrace_getvmreg(ndx,
3228                     &cpu_core[CPU->cpu_id].cpuc_dtrace_flags);

3230                 DTRACE_CPUFLAG_CLEAR(CPU_DTRACE_NOFAULT);

3232                 return (rval);
3233         }

3235         case DIF_VAR_CURTHREAD:
3236                 if (!dtrace_priv_proc(state, mstate))
3237                         return (0);
3238                 return ((uint64_t)(uintptr_t)curthread);

3240         case DIF_VAR_TIMESTAMP:
3241                 if (!(mstate->dtms_present & DTRACE_MSTATE_TIMESTAMP)) {
3242                         mstate->dtms_timestamp = dtrace_gethrtime();
3243                         mstate->dtms_present |= DTRACE_MSTATE_TIMESTAMP;
3244                 }
3245                 return (mstate->dtms_timestamp);

3247         case DIF_VAR_VTIMESTAMP:
3248                 ASSERT(dtrace_vtime_references != 0);
3249                 return (curthread->t_dtrace_vtime);

3251         case DIF_VAR_WALLTIMESTAMP:
3252                 if (!(mstate->dtms_present & DTRACE_MSTATE_WALLTIMESTAMP)) {
3253                         mstate->dtms_walltimestamp = dtrace_gethrestime();
3254                         mstate->dtms_present |= DTRACE_MSTATE_WALLTIMESTAMP;
3255                 }
3256                 return (mstate->dtms_walltimestamp);

3258         case DIF_VAR_IPL:
3259                 if (!dtrace_priv_kernel(state))
3260                         return (0);
3261                 if (!(mstate->dtms_present & DTRACE_MSTATE_IPL)) {
3262                         mstate->dtms_ipl = dtrace_getipl();
3263                         mstate->dtms_present |= DTRACE_MSTATE_IPL;
3264                 }
3265                 return (mstate->dtms_ipl);

3267         case DIF_VAR_EPID:
3268                 ASSERT(mstate->dtms_present & DTRACE_MSTATE_EPID);
3269                 return (mstate->dtms_epid);

3271         case DIF_VAR_ID:
3272                 ASSERT(mstate->dtms_present & DTRACE_MSTATE_PROBE);
3273                 return (mstate->dtms_probe->dtpr_id);

3275         case DIF_VAR_STACKDEPTH:
3276                 if (!dtrace_priv_kernel(state))
3277                         return (0);
3278                 if (!(mstate->dtms_present & DTRACE_MSTATE_STACKDEPTH)) {
3279                         int aframes = mstate->dtms_probe->dtpr_aframes + 2;

3281                         mstate->dtms_stackdepth = dtrace_getstackdepth(aframes);
3282                         mstate->dtms_present |= DTRACE_MSTATE_STACKDEPTH;
3283                 }
3284                 return (mstate->dtms_stackdepth);

3286         case DIF_VAR_USTACKDEPTH:
3287                 if (!dtrace_priv_proc(state, mstate))
3288                         return (0);
3289                 if (!(mstate->dtms_present & DTRACE_MSTATE_USTACKDEPTH)) {
3290                         /*
3291                          * See comment in DIF_VAR_PID.
3292                          */
```

```
3293                        if (DTRACE_ANCHORED(mstate->dtms_probe) &&
3294                            CPU_ON_INTR(CPU)) {
3295                                mstate->dtms_ustackdepth = 0;
3296                        } else {
3297                                DTRACE_CPUFLAG_SET(CPU_DTRACE_NOFAULT);
3298                                mstate->dtms_ustackdepth =
3299                                    dtrace_getustackdepth();
3300                                DTRACE_CPUFLAG_CLEAR(CPU_DTRACE_NOFAULT);
3301                        }
3302                        mstate->dtms_present |= DTRACE_MSTATE_USTACKDEPTH;
3303                }
3304                return (mstate->dtms_ustackdepth);

3306        case DIF_VAR_CALLER:
3307                if (!dtrace_priv_kernel(state))
3308                        return (0);
3309                if (!(mstate->dtms_present & DTRACE_MSTATE_CALLER)) {
3310                        int aframes = mstate->dtms_probe->dtpr_aframes + 2;

3312                        if (!DTRACE_ANCHORED(mstate->dtms_probe)) {
3313                                /*
3314                                 * If this is an unanchored probe, we are
3315                                 * required to go through the slow path:
3316                                 * dtrace_caller() only guarantees correct
3317                                 * results for anchored probes.
3318                                 */
3319                                pc_t caller[2];

3321                                dtrace_getpcstack(caller, 2, aframes,
3322                                    (uint32_t *)(uintptr_t)mstate->dtms_arg[0]);
3323                                mstate->dtms_caller = caller[1];
3324                        } else if ((mstate->dtms_caller =
3325                            dtrace_caller(aframes)) == -1) {
3326                                /*
3327                                 * We have failed to do this the quick way;
3328                                 * we must resort to the slower approach of
3329                                 * calling dtrace_getpcstack().
3330                                 */
3331                                pc_t caller;

3333                                dtrace_getpcstack(&caller, 1, aframes, NULL);
3334                                mstate->dtms_caller = caller;
3335                        }

3337                        mstate->dtms_present |= DTRACE_MSTATE_CALLER;
3338                }
3339                return (mstate->dtms_caller);

3341        case DIF_VAR_UCALLER:
3342                if (!dtrace_priv_proc(state, mstate))
3343                        return (0);

3345                if (!(mstate->dtms_present & DTRACE_MSTATE_UCALLER)) {
3346                        uint64_t ustack[3];

3348                        /*
3349                         * dtrace_getupcstack() fills in the first uint64_t
3350                         * with the current PID.  The second uint64_t will
3351                         * be the program counter at user-level.  The third
3352                         * uint64_t will contain the caller, which is what
3353                         * we're after.
3354                         */
3355                        ustack[2] = NULL;
3356                        DTRACE_CPUFLAG_SET(CPU_DTRACE_NOFAULT);
3357                        dtrace_getupcstack(ustack, 3);
3358                        DTRACE_CPUFLAG_CLEAR(CPU_DTRACE_NOFAULT);
```



```
3359                        mstate->dtms_ucaller = ustack[2];
3360                        mstate->dtms_present |= DTRACE_MSTATE_UCALLER;
3361                }

3363                return (mstate->dtms_ucaller);

3365        case DIF_VAR_PROBEPROV:
3366                ASSERT(mstate->dtms_present & DTRACE_MSTATE_PROBE);
3367                return (dtrace_dif_varstr(
3368                    (uintptr_t)mstate->dtms_probe->dtpr_provider->dtpv_name,
3369                    state, mstate));

3371        case DIF_VAR_PROBEMOD:
3372                ASSERT(mstate->dtms_present & DTRACE_MSTATE_PROBE);
3373                return (dtrace_dif_varstr(
3374                    (uintptr_t)mstate->dtms_probe->dtpr_mod,
3375                    state, mstate));

3377        case DIF_VAR_PROBEFUNC:
3378                ASSERT(mstate->dtms_present & DTRACE_MSTATE_PROBE);
3379                return (dtrace_dif_varstr(
3380                    (uintptr_t)mstate->dtms_probe->dtpr_func,
3381                    state, mstate));

3383        case DIF_VAR_PROBENAME:
3384                ASSERT(mstate->dtms_present & DTRACE_MSTATE_PROBE);
3385                return (dtrace_dif_varstr(
3386                    (uintptr_t)mstate->dtms_probe->dtpr_name,
3387                    state, mstate));

3389        case DIF_VAR_PID:
3390                if (!dtrace_priv_proc(state, mstate))
3391                        return (0);

3393                /*
3394                 * Note that we are assuming that an unanchored probe is
3395                 * always due to a high-level interrupt.  (And we're assuming
3396                 * that there is only a single high level interrupt.)
3397                 */
3398                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3399                        return (pid0.pid_id);

3401                /*
3402                 * It is always safe to dereference one's own t_procp pointer:
3403                 * it always points to a valid, allocated proc structure.
3404                 * Further, it is always safe to dereference the p_pidp member
3405                 * of one's own proc structure.  (These are truisms becuase
3406                 * threads and processes don't clean up their own state --
3407                 * they leave that task to whomever reaps them.)
3408                 */
3409                return ((uint64_t)curthread->t_procp->p_pidp->pid_id);

3411        case DIF_VAR_PPID:
3412                if (!dtrace_priv_proc(state, mstate))
3413                        return (0);

3415                /*
3416                 * See comment in DIF_VAR_PID.
3417                 */
3418                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3419                        return (pid0.pid_id);

3421                /*
3422                 * It is always safe to dereference one's own t_procp pointer:
3423                 * it always points to a valid, allocated proc structure.
3424                 * (This is true because threads don't clean up their own
```

```
3425                 * state -- they leave that task to whomever reaps them.)
3426                 */
3427                return ((uint64_t)curthread->t_procp->p_ppid);

3429        case DIF_VAR_TID:
3430                /*
3431                 * See comment in DIF_VAR_PID.
3432                 */
3433                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3434                        return (0);

3436                return ((uint64_t)curthread->t_tid);

3438        case DIF_VAR_EXECNAME:
3439                if (!dtrace_priv_proc(state, mstate))
3440                        return (0);

3442                /*
3443                 * See comment in DIF_VAR_PID.
3444                 */
3445                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3446                        return ((uint64_t)(uintptr_t)p0.p_user.u_comm);

3448                /*
3449                 * It is always safe to dereference one's own t_procp pointer:
3450                 * it always points to a valid, allocated proc structure.
3451                 * (This is true because threads don't clean up their own
3452                 * state -- they leave that task to whomever reaps them.)
3453                 */
3454                return (dtrace_dif_varstr(
3455                    (uintptr_t)curthread->t_procp->p_user.u_comm,
3456                    state, mstate));

3458        case DIF_VAR_ZONENAME:
3459                if (!dtrace_priv_proc(state, mstate))
3460                        return (0);

3462                /*
3463                 * See comment in DIF_VAR_PID.
3464                 */
3465                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3466                        return ((uint64_t)(uintptr_t)p0.p_zone->zone_name);

3468                /*
3469                 * It is always safe to dereference one's own t_procp pointer:
3470                 * it always points to a valid, allocated proc structure.
3471                 * (This is true because threads don't clean up their own
3472                 * state -- they leave that task to whomever reaps them.)
3473                 */
3474                return (dtrace_dif_varstr(
3475                    (uintptr_t)curthread->t_procp->p_zone->zone_name,
3476                    state, mstate));

3478        case DIF_VAR_UID:
3479                if (!dtrace_priv_proc(state, mstate))
3480                        return (0);

3482                /*
3483                 * See comment in DIF_VAR_PID.
3484                 */
3485                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3486                        return ((uint64_t)p0.p_cred->cr_uid);

3488                /*
3489                 * It is always safe to dereference one's own t_procp pointer:
3490                 * it always points to a valid, allocated proc structure.
```



```
3491                 * (This is true because threads don't clean up their own
3492                 * state -- they leave that task to whomever reaps them.)
3493                 *
3494                 * Additionally, it is safe to dereference one's own process
3495                 * credential, since this is never NULL after process birth.
3496                 */
3497                return ((uint64_t)curthread->t_procp->p_cred->cr_uid);

3499        case DIF_VAR_GID:
3500                if (!dtrace_priv_proc(state, mstate))
3501                        return (0);

3503                /*
3504                 * See comment in DIF_VAR_PID.
3505                 */
3506                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3507                        return ((uint64_t)p0.p_cred->cr_gid);

3509                /*
3510                 * It is always safe to dereference one's own t_procp pointer:
3511                 * it always points to a valid, allocated proc structure.
3512                 * (This is true because threads don't clean up their own
3513                 * state -- they leave that task to whomever reaps them.)
3514                 *
3515                 * Additionally, it is safe to dereference one's own process
3516                 * credential, since this is never NULL after process birth.
3517                 */
3518                return ((uint64_t)curthread->t_procp->p_cred->cr_gid);

3520        case DIF_VAR_ERRNO: {
3521                klwp_t *lwp;
3522                if (!dtrace_priv_proc(state, mstate))
3523                        return (0);

3525                /*
3526                 * See comment in DIF_VAR_PID.
3527                 */
3528                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3529                        return (0);

3531                /*
3532                 * It is always safe to dereference one's own t_lwp pointer in
3533                 * the event that this pointer is non-NULL.  (This is true
3534                 * because threads and lwps don't clean up their own state --
3535                 * they leave that task to whomever reaps them.)
3536                 */
3537                if ((lwp = curthread->t_lwp) == NULL)
3538                        return (0);

3540                return ((uint64_t)lwp->lwp_errno);
3541        }

3543        case DIF_VAR_THREADNAME:
3544                /*
3545                 * See comment in DIF_VAR_PID.
3546                 */
3547                if (DTRACE_ANCHORED(mstate->dtms_probe) && CPU_ON_INTR(CPU))
3548                        return (0);

3550                if (curthread->t_name == NULL)
3551                        return (0);

3553                /*
3554                 * Once set, ->t_name itself is never changed: any updates are
3555                 * made to the same buffer that we are pointing out.  So we are
3556                 * safe to dereference it here.
```

```
3557                  */
3558                 return (dtrace_dif_varstr((uintptr_t)curthread->t_name,
3559                     state, mstate));

3561         default:
3562                 DTRACE_CPUFLAG_SET(CPU_DTRACE_ILLOP);
3563                 return (0);
3564         }
3565 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   16063 Mon Oct 15 13:28:41 2018
new/usr/src/uts/common/exec/elf/elf_notes.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 * Copyright (c) 2014, Joyent, Inc. All rights reserved.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/thread.h>
#include <sys/sysmacros.h>
#include <sys/signal.h>
#include <sys/cred.h>
#include <sys/priv.h>
#include <sys/user.h>
#include <sys/file.h>
#include <sys/errno.h>
#include <sys/vnode.h>
#include <sys/mode.h>
#include <sys/vfs.h>
#include <sys/mman.h>
#include <sys/kmem.h>
#include <sys/proc.h>
#include <sys/pathname.h>
#include <sys/cmn_err.h>
#include <sys/systm.h>
#include <sys/elf.h>
#include <sys/vmsystm.h>
#include <sys/debug.h>
#include <sys/procfs.h>
#include <sys/regset.h>
#include <sys/auxv.h>
#include <sys/exec.h>
#include <sys/prsystm.h>
#include <sys/utsname.h>
```



```
#include <sys/zone.h>
#include <vm/as.h>
#include <vm/rm.h>
#include <sys/modctl.h>
#include <sys/systeminfo.h>
#include <sys/machelf.h>
#include <sys/sunddi.h>
#include "elf_impl.h"
#if defined(__i386) || defined(__i386_COMPAT)
#include <sys/sysi86.h>
#endif

void
setup_note_header(Phdr *v, proc_t *p)
{
        int nlwp = p->p_lwpcnt;
        int nzomb = p->p_zombcnt;
        int nfd;
        size_t size;
        prcred_t *pcrp;
        uf_info_t *fip;
        uf_entry_t *ufp;
        int fd;

        fip = P_FINFO(p);
        nfd = 0;
        mutex_enter(&fip->fi_lock);
        for (fd = 0; fd < fip->fi_nfiles; fd++) {
                UF_ENTER(ufp, fip, fd);
                if ((ufp->uf_file != NULL) && (ufp->uf_file->f_count > 0))
                        nfd++;
                UF_EXIT(ufp);
        }
        mutex_exit(&fip->fi_lock);

        v[0].p_type = PT_NOTE;
        v[0].p_flags = PF_R;
        v[0].p_filesz = (sizeof (Note) * (10 + 3 * nlwp + nzomb + nfd))
        v[0].p_filesz = (sizeof (Note) * (10 + 2 * nlwp + nzomb + nfd))
            + roundup(sizeof (psinfo_t), sizeof (Word))
            + roundup(sizeof (pstatus_t), sizeof (Word))
            + roundup(prgetprivsize(), sizeof (Word))
            + roundup(priv_get_implinfo_size(), sizeof (Word))
            + roundup(strlen(platform) + 1, sizeof (Word))
            + roundup(strlen(p->p_zone->zone_name) + 1, sizeof (Word))
            + roundup(__KERN_NAUXV_IMPL * sizeof (aux_entry_t), sizeof (Word))
            + roundup(sizeof (utsname), sizeof (Word))
            + roundup(sizeof (core_content_t), sizeof (Word))
            + roundup(sizeof (prsecflags_t), sizeof (Word))
            + (nlwp + nzomb) * roundup(sizeof (lwpsinfo_t), sizeof (Word))
            + nlwp * roundup(sizeof (lwpstatus_t), sizeof (Word))
            + nlwp * roundup(sizeof (prlwpname_t), sizeof (Word))
            + nfd * roundup(sizeof (prfdinfo_t), sizeof (Word));

        if (curproc->p_agenttp != NULL) {
                v[0].p_filesz += sizeof (Note) +
                    roundup(sizeof (psinfo_t), sizeof (Word));
        }

        size = sizeof (prcred_t) + sizeof (gid_t) * (ngroups_max - 1);
        pcrp = kmem_alloc(size, KM_SLEEP);
        prgetcred(p, pcrp);
        if (pcrp->pr_ngroups != 0) {
                v[0].p_filesz += sizeof (Note) + roundup(sizeof (prcred_t) +
                    sizeof (gid_t) * (pcrp->pr_ngroups - 1), sizeof (Word));
        } else {
```

```
125                 v[0].p_filesz += sizeof (Note) +
126                     roundup(sizeof (prcred_t), sizeof (Word));
127         }
128         kmem_free(pcrp, size);


131 #if defined(__i386) || defined(__i386_COMPAT)
132         mutex_enter(&p->p_ldtlock);
133         size = prnldt(p) * sizeof (struct ssd);
134         mutex_exit(&p->p_ldtlock);
135         if (size != 0)
136                 v[0].p_filesz += sizeof (Note) + roundup(size, sizeof (Word));
137 #endif  /* __i386 || __i386_COMPAT */

139         if ((size = prhasx(p)? prgetprxregsize(p) : 0) != 0)
140                 v[0].p_filesz += nlwp * sizeof (Note)
141                     + nlwp * roundup(size, sizeof (Word));

143 #if defined(__sparc)
144         /*
145          * Figure out the number and sizes of register windows.
146          */
147         {
148                 kthread_t *t = p->p_tlist;
149                 do {
150                         if ((size = prnwindows(ttolwp(t))) != 0) {
151                                 size = sizeof (gwindows_t) -
152                                     (SPARC_MAXREGWINDOW - size) *
153                                     sizeof (struct rwindow);
154                                 v[0].p_filesz += sizeof (Note) +
155                                     roundup(size, sizeof (Word));
156                         }
157                 } while ((t = t->t_forw) != p->p_tlist);
158         }
159         /*
160          * Space for the Ancillary State Registers.
161          */
162         if (p->p_model == DATAMODEL_LP64)
163                 v[0].p_filesz += nlwp * sizeof (Note)
164                     + nlwp * roundup(sizeof (asrset_t), sizeof (Word));
165 #endif /* __sparc */
166 }

168 int
169 write_elfnotes(proc_t *p, int sig, vnode_t *vp, offset_t offset,
170     rlim64_t rlimit, cred_t *credp, core_content_t content)
171 {
172         union {
173                 psinfo_t        psinfo;
174                 pstatus_t       pstatus;
175                 lwpsinfo_t      lwpsinfo;
176                 lwpstatus_t     lwpstatus;
177 #if defined(__sparc)
178                 gwindows_t      gwindows;
179                 asrset_t        asrset;
180 #endif /* __sparc */
181                 char            xregs[1];
182                 aux_entry_t     auxv[__KERN_NAUXV_IMPL];
183                 prcred_t        pcred;
184                 prpriv_t        ppriv;
185                 priv_impl_info_t prinfo;
186                 struct utsname  uts;
187                 prsecflags_t    psecflags;
188         } *bigwad;

190         size_t xregsize = prhasx(p)? prgetprxregsize(p) : 0;
```



```
191         size_t crsize = sizeof (prcred_t) + sizeof (gid_t) * (ngroups_max - 1);
192         size_t psize = prgetprivsize();
193         size_t bigsize = MAX(psize, MAX(sizeof (*bigwad),
194             MAX(xregsize, crsize)));

196         priv_impl_info_t *prii;

198         lwpdir_t *ldp;
199         lwpent_t *lep;
200         kthread_t *t;
201         klwp_t *lwp;
202         user_t *up;
203         int i;
204         int nlwp;
205         int nzomb;
206         int error;
207         uchar_t oldsig;
208         uf_info_t *fip;
209         int fd;
210         vnode_t *vroot;

212 #if defined(__i386) || defined(__i386_COMPAT)
213         struct ssd *ssd;
214         size_t ssdsize;
215 #endif  /* __i386 || __i386_COMPAT */

217         bigsize = MAX(bigsize, priv_get_implinfo_size());

219         bigwad = kmem_alloc(bigsize, KM_SLEEP);

221         /*
222          * The order of the elfnote entries should be same here
223          * and in the gcore(1) command.  Synchronization is
224          * needed between the kernel and gcore(1).
225          */

227         /*
228          * Get the psinfo, and set the wait status to indicate that a core was
229          * dumped.  We have to forge this since p->p_wcode is not set yet.
230          */
231         mutex_enter(&p->p_lock);
232         prgetpsinfo(p, &bigwad->psinfo);
233         mutex_exit(&p->p_lock);
234         bigwad->psinfo.pr_wstat = wstat(CLD_DUMPED, sig);

236         error = elfnote(vp, &offset, NT_PSINFO, sizeof (bigwad->psinfo),
237             (caddr_t)&bigwad->psinfo, rlimit, credp);
238         if (error)
239                 goto done;

241         /*
242          * Modify t_whystop and lwp_cursig so it appears that the current LWP
243          * is stopped after faulting on the signal that caused the core dump.
244          * As a result, prgetstatus() will record that signal, the saved
245          * lwp_siginfo, and its signal handler in the core file status.  We
246          * restore lwp_cursig in case a subsequent signal was received while
247          * dumping core.
248          */
249         mutex_enter(&p->p_lock);
250         lwp = ttolwp(curthread);

252         oldsig = lwp->lwp_cursig;
253         lwp->lwp_cursig = (uchar_t)sig;
254         curthread->t_whystop = PR_FAULTED;

256         prgetstatus(p, &bigwad->pstatus, p->p_zone);
```

```
257         bigwad->pstatus.pr_lwp.pr_why = 0;

259         curthread->t_whystop = 0;
260         lwp->lwp_cursig = oldsig;
261         mutex_exit(&p->p_lock);

263         error = elfnote(vp, &offset, NT_PSTATUS, sizeof (bigwad->pstatus),
264             (caddr_t)&bigwad->pstatus, rlimit, credp);
265         if (error)
266                 goto done;

268         error = elfnote(vp, &offset, NT_PLATFORM, strlen(platform) + 1,
269             platform, rlimit, credp);
270         if (error)
271                 goto done;

273         up = PTOU(p);
274         for (i = 0; i < __KERN_NAUXV_IMPL; i++) {
275                 bigwad->auxv[i].a_type = up->u_auxv[i].a_type;
276                 bigwad->auxv[i].a_un.a_val = up->u_auxv[i].a_un.a_val;
277         }
278         error = elfnote(vp, &offset, NT_AUXV, sizeof (bigwad->auxv),
279             (caddr_t)bigwad->auxv, rlimit, credp);
280         if (error)
281                 goto done;

283         bcopy(&utsname, &bigwad->uts, sizeof (struct utsname));
284         if (!INGLOBALZONE(p)) {
285                 bcopy(p->p_zone->zone_nodename, &bigwad->uts.nodename,
286                     _SYS_NMLN);
287         }
288         error = elfnote(vp, &offset, NT_UTSNAME, sizeof (struct utsname),
289             (caddr_t)&bigwad->uts, rlimit, credp);
290         if (error)
291                 goto done;

293         prgetsecflags(p, &bigwad->psecflags);
294         error = elfnote(vp, &offset, NT_SECFLAGS, sizeof (prsecflags_t),
295             (caddr_t)&bigwad->psecflags, rlimit, credp);
296         if (error)
297                 goto done;

299         prgetcred(p, &bigwad->pcred);

301         if (bigwad->pcred.pr_ngroups != 0) {
302                 crsize = sizeof (prcred_t) +
303                     sizeof (gid_t) * (bigwad->pcred.pr_ngroups - 1);
304         } else
305                 crsize = sizeof (prcred_t);

307         error = elfnote(vp, &offset, NT_PRCRED, crsize,
308             (caddr_t)&bigwad->pcred, rlimit, credp);
309         if (error)
310                 goto done;

312         error = elfnote(vp, &offset, NT_CONTENT, sizeof (core_content_t),
313             (caddr_t)&content, rlimit, credp);
314         if (error)
315                 goto done;

317         prgetpriv(p, &bigwad->ppriv);

319         error = elfnote(vp, &offset, NT_PRPRIV, psize,
320             (caddr_t)&bigwad->ppriv, rlimit, credp);
321         if (error)
322                 goto done;
```



```
324         prii = priv_hold_implinfo();
325         error = elfnote(vp, &offset, NT_PRPRIVINFO, priv_get_implinfo_size(),
326             (caddr_t)prii, rlimit, credp);
327         priv_release_implinfo();
328         if (error)
329                 goto done;

331         /* zone can't go away as long as process exists */
332         error = elfnote(vp, &offset, NT_ZONENAME,
333             strlen(p->p_zone->zone_name) + 1, p->p_zone->zone_name,
334             rlimit, credp);
335         if (error)
336                 goto done;


339         /* open file table */
340         vroot = PTOU(p)->u_rdir;
341         if (vroot == NULL)
342                 vroot = rootdir;

344         VN_HOLD(vroot);

346         fip = P_FINFO(p);

348         for (fd = 0; fd < fip->fi_nfiles; fd++) {
349                 uf_entry_t *ufp;
350                 vnode_t *fvp;
351                 struct file *fp;
352                 vattr_t vattr;
353                 prfdinfo_t fdinfo;

355                 bzero(&fdinfo, sizeof (fdinfo));

357                 mutex_enter(&fip->fi_lock);
358                 UF_ENTER(ufp, fip, fd);
359                 if (((fp = ufp->uf_file) == NULL) || (fp->f_count < 1)) {
360                         UF_EXIT(ufp);
361                         mutex_exit(&fip->fi_lock);
362                         continue;
363                 }

365                 fdinfo.pr_fd = fd;
366                 fdinfo.pr_fdflags = ufp->uf_flag;
367                 fdinfo.pr_fileflags = fp->f_flag2;
368                 fdinfo.pr_fileflags <<= 16;
369                 fdinfo.pr_fileflags |= fp->f_flag;
370                 if ((fdinfo.pr_fileflags & (FSEARCH | FEXEC)) == 0)
371                         fdinfo.pr_fileflags += FOPEN;
372                 fdinfo.pr_offset = fp->f_offset;


375                 fvp = fp->f_vnode;
376                 VN_HOLD(fvp);
377                 UF_EXIT(ufp);
378                 mutex_exit(&fip->fi_lock);

380                 /*
381                  * There are some vnodes that have no corresponding
382                  * path.  Its reasonable for this to fail, in which
383                  * case the path will remain an empty string.
384                  */
385                 (void) vnodetopath(vroot, fvp, fdinfo.pr_path,
386                     sizeof (fdinfo.pr_path), credp);

388                 if (VOP_GETATTR(fvp, &vattr, 0, credp, NULL) != 0) {
```

```
389                         /*
390                          * Try to write at least a subset of information
391                          */
392                         fdinfo.pr_major = 0;
393                         fdinfo.pr_minor = 0;
394                         fdinfo.pr_ino = 0;
395                         fdinfo.pr_mode = 0;
396                         fdinfo.pr_uid = (uid_t)-1;
397                         fdinfo.pr_gid = (gid_t)-1;
398                         fdinfo.pr_rmajor = 0;
399                         fdinfo.pr_rminor = 0;
400                         fdinfo.pr_size = -1;

402                         error = elfnote(vp, &offset, NT_FDINFO,
403                             sizeof (fdinfo), &fdinfo, rlimit, credp);
404                         VN_RELE(fvp);
405                         if (error) {
406                                 VN_RELE(vroot);
407                                 goto done;
408                         }
409                         continue;
410                 }

412                 if (fvp->v_type == VSOCK)
413                         fdinfo.pr_fileflags |= sock_getfasync(fvp);

415                 VN_RELE(fvp);

417                 /*
418                  * This logic mirrors fstat(), which we cannot use
419                  * directly, as it calls copyout().
420                  */
421                 fdinfo.pr_major = getmajor(vattr.va_fsid);
422                 fdinfo.pr_minor = getminor(vattr.va_fsid);
423                 fdinfo.pr_ino = (ino64_t)vattr.va_nodeid;
424                 fdinfo.pr_mode = VTTOIF(vattr.va_type) | vattr.va_mode;
425                 fdinfo.pr_uid = vattr.va_uid;
426                 fdinfo.pr_gid = vattr.va_gid;
427                 fdinfo.pr_rmajor = getmajor(vattr.va_rdev);
428                 fdinfo.pr_rminor = getminor(vattr.va_rdev);
429                 fdinfo.pr_size = (off64_t)vattr.va_size;

431                 error = elfnote(vp, &offset, NT_FDINFO,
432                     sizeof (fdinfo), &fdinfo, rlimit, credp);
433                 if (error) {
434                         VN_RELE(vroot);
435                         goto done;
436                 }
437         }

439         VN_RELE(vroot);

441 #if defined(__i386) || defined(__i386_COMPAT)
442         mutex_enter(&p->p_ldtlock);
443         ssdsize = prnldt(p) * sizeof (struct ssd);
444         if (ssdsize != 0) {
445                 ssd = kmem_alloc(ssdsize, KM_SLEEP);
446                 prgetldt(p, ssd);
447                 error = elfnote(vp, &offset, NT_LDT, ssdsize,
448                     (caddr_t)ssd, rlimit, credp);
449                 kmem_free(ssd, ssdsize);
450         }
451         mutex_exit(&p->p_ldtlock);
452         if (error)
453                 goto done;
454 #endif  /* __i386 || defined(__i386_COMPAT) */
```



```
456         nlwp = p->p_lwpcnt;
457         nzomb = p->p_zombcnt;
458         /* for each entry in the lwp directory ... */
459         for (ldp = p->p_lwpdir; nlwp + nzomb != 0; ldp++) {
460                 prlwpname_t name = { 0, };

462                 if ((lep = ldp->ld_entry) == NULL)      /* empty slot */
463                         continue;

465                 if ((t = lep->le_thread) != NULL) {     /* active lwp */
466                         ASSERT(nlwp != 0);
467                         nlwp--;
468                         lwp = ttolwp(t);
469                         mutex_enter(&p->p_lock);
470                         prgetlwpsinfo(t, &bigwad->lwpsinfo);
471                         if (t->t_name != NULL) {
472                                 (void) strlcpy(name.pr_lwpname, t->t_name,
473                                     sizeof (name.pr_lwpname));
474                         }
475                         mutex_exit(&p->p_lock);
476                 } else {                                /* zombie lwp */
477                         ASSERT(nzomb != 0);
478                         nzomb--;
479                         bzero(&bigwad->lwpsinfo, sizeof (bigwad->lwpsinfo));
480                         bigwad->lwpsinfo.pr_lwpid = lep->le_lwpid;
481                         bigwad->lwpsinfo.pr_state = SZOMB;
482                         bigwad->lwpsinfo.pr_sname = 'Z';
483                         bigwad->lwpsinfo.pr_start.tv_sec = lep->le_start;
484                 }

486                 name.pr_lwpid = bigwad->lwpsinfo.pr_lwpid;

488                 error = elfnote(vp, &offset, NT_LWPSINFO,
489                     sizeof (bigwad->lwpsinfo), (caddr_t)&bigwad->lwpsinfo,
490                     rlimit, credp);
491                 if (error)
492                         goto done;

494                 if (t == NULL)          /* nothing more to do for a zombie */
495                         continue;

497                 mutex_enter(&p->p_lock);
498                 if (t == curthread) {
499                         /*
500                          * Modify t_whystop and lwp_cursig so it appears that
501                          * the current LWP is stopped after faulting on the
502                          * signal that caused the core dump.  As a result,
503                          * prgetlwpstatus() will record that signal, the saved
504                          * lwp_siginfo, and its signal handler in the core file
505                          * status.  We restore lwp_cursig in case a subsequent
506                          * signal was received while dumping core.
507                          */
508                         oldsig = lwp->lwp_cursig;
509                         lwp->lwp_cursig = (uchar_t)sig;
510                         t->t_whystop = PR_FAULTED;

512                         prgetlwpstatus(t, &bigwad->lwpstatus, p->p_zone);
513                         bigwad->lwpstatus.pr_why = 0;

515                         t->t_whystop = 0;
516                         lwp->lwp_cursig = oldsig;
517                 } else {
518                         prgetlwpstatus(t, &bigwad->lwpstatus, p->p_zone);
519                 }
520                 mutex_exit(&p->p_lock);
```

```
521                 error = elfnote(vp, &offset, NT_LWPSTATUS,
522                     sizeof (bigwad->lwpstatus), (caddr_t)&bigwad->lwpstatus,
523                     rlimit, credp);
524                 if (error)
525                         goto done;

527                 if ((error = elfnote(vp, &offset, NT_LWPNAME, sizeof (name),
528                     (caddr_t)&name, rlimit, credp)) != 0)
529                         goto done;


532 #if defined(__sparc)
533                 /*
534                  * Unspilled SPARC register windows.
535                  */
536                 {
537                         size_t size = prnwindows(lwp);

539                         if (size != 0) {
540                                 size = sizeof (gwindows_t) -
541                                     (SPARC_MAXREGWINDOW - size) *
542                                     sizeof (struct rwindow);
543                                 prgetwindows(lwp, &bigwad->gwindows);
544                                 error = elfnote(vp, &offset, NT_GWINDOWS,
545                                     size, (caddr_t)&bigwad->gwindows,
546                                     rlimit, credp);
547                                 if (error)
548                                         goto done;
549                         }
550                 }
551                 /*
552                  * Ancillary State Registers.
553                  */
554                 if (p->p_model == DATAMODEL_LP64) {
555                         prgetasregs(lwp, bigwad->asrset);
556                         error = elfnote(vp, &offset, NT_ASRS,
557                             sizeof (asrset_t), (caddr_t)bigwad->asrset,
558                             rlimit, credp);
559                         if (error)
560                                 goto done;
561                 }
562 #endif /* __sparc */

564                 if (xregsize) {
565                         prgetprxregs(lwp, bigwad->xregs);
566                         error = elfnote(vp, &offset, NT_PRXREG,
567                             xregsize, bigwad->xregs, rlimit, credp);
568                         if (error)
569                                 goto done;
570                 }

572                 if (t->t_lwp->lwp_spymaster != NULL) {
573                         void *psaddr = t->t_lwp->lwp_spymaster;
574 #ifdef _ELF32_COMPAT
575                         /*
576                          * On a 64-bit kernel with 32-bit ELF compatibility,
577                          * this file is compiled into two different objects:
578                          * one is compiled normally, and the other is compiled
579                          * with _ELF32_COMPAT set -- and therefore with a
580                          * psinfo_t defined to be a psinfo32_t.  However, the
581                          * psinfo_t denoting our spymaster is always of the
582                          * native type; if we are in the _ELF32_COMPAT case,
583                          * we need to explicitly convert it.
584                          */
585                         if (p->p_model == DATAMODEL_ILP32) {
586                                 psinfo_kto32(psaddr, &bigwad->psinfo);
```



```
587                                 psaddr = &bigwad->psinfo;
588                         }
589 #endif

591                         error = elfnote(vp, &offset, NT_SPYMASTER,
592                             sizeof (psinfo_t), psaddr, rlimit, credp);
593                         if (error)
594                                 goto done;
595                 }
596         }
597         ASSERT(nlwp == 0);

599 done:
600         kmem_free(bigwad, bigsize);
601         return (error);
602 }
______unchanged_portion_omitted_
```

```
**********************************************************
   15031 Mon Oct 15 13:28:41 2018
new/usr/src/uts/common/fs/proc/prdata.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*
 * Copyright 2018 Joyent, Inc.
 * Copyright (c) 2013, Joyent, Inc. All rights reserved.
 */

#ifndef _SYS_PROC_PRDATA_H
#define _SYS_PROC_PRDATA_H

#include <sys/isa_defs.h>
#include <sys/proc.h>
#include <sys/vnode.h>
#include <sys/prsystm.h>
#include <sys/model.h>
#include <sys/poll.h>
#include <sys/list.h>

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * Test for thread being stopped, not on an event of interest,
 * but with a directed stop in effect.
 */
#define DSTOPPED(t)     \
        ((t)->t_state == TS_STOPPED && \
        ((t)->t_proc_flag & TP_PRSTOP))

#define round4(r)       (((r) + 3) & (~3))
#define round8(r)       (((r) + 7) & (~7))
#define round16(r)      (((r) + 15) & (~15))
#define roundlong(r)    (((r) + sizeof (long) - 1) & (~(sizeof (long) - 1)))
```



```
#define PNSIZ   10                      /* max size of /proc name entries */
#define PLNSIZ  10                      /* max size of /proc lwp name entries */

/*
 * Common file object to which all /proc vnodes for a specific process
 * or lwp refer.  One for the process, one for each lwp.
 */
typedef struct prcommon {
        kmutex_t        prc_mutex;      /* to wait for the proc/lwp to stop */
        kcondvar_t      prc_wait;       /* to wait for the proc/lwp to stop */
        ushort_t        prc_flags;      /* flags */
        uint_t          prc_writers;    /* number of write opens of prnodes */
        uint_t          prc_selfopens;  /* number of write opens by self */
        pid_t           prc_pid;        /* process id */
        model_t         prc_datamodel;  /* data model of the process */
        proc_t          *prc_proc;      /* process being traced */
        kthread_t       *prc_thread;    /* thread (lwp) being traced */
        int             prc_slot;       /* procdir slot number */
        id_t            prc_tid;        /* thread (lwp) id */
        int             prc_tslot;      /* lwpdir slot number, -1 if reaped */
        int             prc_refcnt;     /* this structure's reference count */
        struct pollhead prc_pollhead;   /* list of all pollers */
} prcommon_t;

/* prc_flags */
#define PRC_DESTROY     0x01    /* process or lwp is being destroyed */
#define PRC_LWP         0x02    /* structure refers to an lwp */
#define PRC_SYS         0x04    /* process is a system process */
#define PRC_POLL        0x08    /* poll() in progress on this process/lwp */
#define PRC_EXCL        0x10    /* exclusive access granted (old /proc) */

/*
 * Macros for mapping between i-numbers and pids.
 */
#define pmkino(tslot, pslot, nodetype)          \
        (((((ino_t)(tslot) << nproc_highbit) |  \
        (ino_t)(pslot)) << 6) |                 \
        (nodetype) + 2)

/* for old /proc interface */
#define PRBIAS  64
#define ptoi(n) ((int)(((n) + PRBIAS)))         /* pid to i-number */

/*
 * Node types for /proc files (directories and files contained therein).
 */
typedef enum prnodetype {
        PR_PROCDIR,             /* /proc                                */
        PR_SELF,                /* /proc/self                           */
        PR_PIDDIR,              /* /proc/<pid>                          */
        PR_AS,                  /* /proc/<pid>/as                       */
        PR_CTL,                 /* /proc/<pid>/ctl                      */
        PR_STATUS,              /* /proc/<pid>/status                   */
        PR_LSTATUS,             /* /proc/<pid>/lstatus                  */
        PR_PSINFO,              /* /proc/<pid>/psinfo                   */
        PR_LPSINFO,             /* /proc/<pid>/lpsinfo                  */
        PR_MAP,                 /* /proc/<pid>/map                      */
        PR_RMAP,                /* /proc/<pid>/rmap                     */
        PR_XMAP,                /* /proc/<pid>/xmap                     */
        PR_CRED,                /* /proc/<pid>/cred                     */
        PR_SIGACT,              /* /proc/<pid>/sigact                   */
        PR_AUXV,                /* /proc/<pid>/auxv                     */
#if defined(__i386) || defined(__amd64)
        PR_LDT,                 /* /proc/<pid>/ldt                      */
#endif
```

```
126         PR_USAGE,               /* /proc/<pid>/usage                    */
127         PR_LUSAGE,              /* /proc/<pid>/lusage                   */
128         PR_PAGEDATA,            /* /proc/<pid>/pagedata                 */
129         PR_WATCH,               /* /proc/<pid>/watch                    */
130         PR_CURDIR,              /* /proc/<pid>/cwd                      */
131         PR_ROOTDIR,             /* /proc/<pid>/root                     */
132         PR_FDDIR,               /* /proc/<pid>/fd                       */
133         PR_FD,                  /* /proc/<pid>/fd/nn                    */
134         PR_OBJECTDIR,           /* /proc/<pid>/object                   */
135         PR_OBJECT,              /* /proc/<pid>/object/xxx               */
136         PR_LWPDIR,              /* /proc/<pid>/lwp                      */
137         PR_LWPIDDIR,            /* /proc/<pid>/lwp/<lwpid>              */
138         PR_LWPCTL,              /* /proc/<pid>/lwp/<lwpid>/lwpctl       */
139         PR_LWPNAME,             /* /proc/<pid>/lwp/<lwpid>/lwpname      */
140         PR_LWPSTATUS,           /* /proc/<pid>/lwp/<lwpid>/lwpstatus    */
141         PR_LWPSINFO,            /* /proc/<pid>/lwp/<lwpid>/lwpsinfo     */
142         PR_LWPUSAGE,            /* /proc/<pid>/lwp/<lwpid>/lwpusage     */
143         PR_XREGS,               /* /proc/<pid>/lwp/<lwpid>/xregs        */
144         PR_TMPLDIR,             /* /proc/<pid>/lwp/<lwpid>/templates    */
145         PR_TMPL,                /* /proc/<pid>/lwp/<lwpid>/templates/<id> */
146         PR_SPYMASTER,           /* /proc/<pid>/lwp/<lwpid>/spymaster    */
147 #if defined(__sparc)
148         PR_GWINDOWS,            /* /proc/<pid>/lwp/<lwpid>/gwindows     */
149         PR_ASRS,                /* /proc/<pid>/lwp/<lwpid>/asrs         */
150 #endif
151         PR_PRIV,                /* /proc/<pid>/priv                     */
152         PR_PATHDIR,             /* /proc/<pid>/path                     */
153         PR_PATH,                /* /proc/<pid>/path/xxx                 */
154         PR_CTDIR,               /* /proc/<pid>/contracts                */
155         PR_CT,                  /* /proc/<pid>/contracts/<ctid>         */
156         PR_SECFLAGS,            /* /proc/<pid>/secflags                 */
157         PR_PIDFILE,             /* old process file                     */
158         PR_LWPIDFILE,           /* old lwp file                         */
159         PR_OPAGEDATA,           /* old page data file                   */
160         PR_NFILES               /* number of /proc node types           */
161 } prnodetype_t;
```

*_____unchanged_portion_omitted_*

```
**********************************************************
  146787 Mon Oct 15 13:28:42 2018
new/usr/src/uts/common/fs/proc/prvnops.c
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
 * Copyright (c) 2018, Joyent, Inc.
 * Copyright (c) 2017, Joyent, Inc.
 * Copyright (c) 2017 by Delphix. All rights reserved.
 */

/*      Copyright (c) 1984,      1986, 1987, 1988, 1989 AT&T    */
/*        All Rights Reserved   */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/time.h>
#include <sys/cred.h>
#include <sys/policy.h>
#include <sys/debug.h>
#include <sys/dirent.h>
#include <sys/errno.h>
#include <sys/file.h>
#include <sys/inline.h>
#include <sys/kmem.h>
#include <sys/pathname.h>
#include <sys/proc.h>
#include <sys/brand.h>
#include <sys/signal.h>
#include <sys/stat.h>
#include <sys/sysmacros.h>
#include <sys/systm.h>
#include <sys/zone.h>
#include <sys/uio.h>
#include <sys/var.h>
#include <sys/mode.h>
#include <sys/poll.h>
#include <sys/user.h>
#include <sys/vfs.h>
#include <sys/vfs_opreg.h>
#include <sys/gfs.h>
#include <sys/vnode.h>
#include <sys/fault.h>
```



```
#include <sys/syscall.h>
#include <sys/procfs.h>
#include <sys/atomic.h>
#include <sys/cmn_err.h>
#include <sys/contract_impl.h>
#include <sys/ctfs.h>
#include <sys/avl.h>
#include <sys/ctype.h>
#include <fs/fs_subr.h>
#include <vm/rm.h>
#include <vm/as.h>
#include <vm/seg.h>
#include <vm/seg_vn.h>
#include <vm/hat.h>
#include <fs/proc/prdata.h>
#if defined(__sparc)
#include <sys/regset.h>
#endif
#if defined(__x86)
#include <sys/sysi86.h>
#endif

/*
 * Created by prinit.
 */
vnodeops_t *prvnodeops;

/*
 * Directory characteristics (patterned after the s5 file system).
 */
#define PRROOTINO       2

#define PRDIRSIZE       14
struct prdirect {
        ushort_t        d_ino;
        char            d_name[PRDIRSIZE];
};
_____unchanged_portion_omitted_

#define NPIDDIRFILES    (sizeof (piddir) / sizeof (piddir[0]) - 2)

/*
 * Contents of a /proc/<pid>/lwp/<lwpid> directory.
 */
static prdirent_t lwpiddir[] = {
        { PR_LWPIDDIR,   1 * sizeof (prdirent_t), sizeof (prdirent_t),
                "." },
        { PR_LWPDIR,     2 * sizeof (prdirent_t), sizeof (prdirent_t),
                ".." },
        { PR_LWPCTL,     3 * sizeof (prdirent_t), sizeof (prdirent_t),
                "lwpctl" },
        { PR_LWPNAME,    4 * sizeof (prdirent_t), sizeof (prdirent_t),
                "lwpname" },
        { PR_LWPSTATUS,  5 * sizeof (prdirent_t), sizeof (prdirent_t),
        { PR_LWPSTATUS,  4 * sizeof (prdirent_t), sizeof (prdirent_t),
                "lwpstatus" },
        { PR_LWPSINFO,   6 * sizeof (prdirent_t), sizeof (prdirent_t),
        { PR_LWPSINFO,   5 * sizeof (prdirent_t), sizeof (prdirent_t),
                "lwpsinfo" },
        { PR_LWPUSAGE,   7 * sizeof (prdirent_t), sizeof (prdirent_t),
        { PR_LWPUSAGE,   6 * sizeof (prdirent_t), sizeof (prdirent_t),
                "lwpusage" },
        { PR_XREGS,      8 * sizeof (prdirent_t), sizeof (prdirent_t),
        { PR_XREGS,      7 * sizeof (prdirent_t), sizeof (prdirent_t),
                "xregs" },
        { PR_TMPLDIR,    9 * sizeof (prdirent_t), sizeof (prdirent_t),
```

```
194         { PR_TMPLDIR,    8 * sizeof (prdirent_t), sizeof (prdirent_t),
198                 "templates" },
199         { PR_SPYMASTER,  10 * sizeof (prdirent_t), sizeof (prdirent_t),
196         { PR_SPYMASTER,  9 * sizeof (prdirent_t), sizeof (prdirent_t),
200                 "spymaster" },
201 #if defined(__sparc)
202         { PR_GWINDOWS,  11 * sizeof (prdirent_t), sizeof (prdirent_t),
199         { PR_GWINDOWS,  10 * sizeof (prdirent_t), sizeof (prdirent_t),
203                 "gwindows" },
204         { PR_ASRS,      12 * sizeof (prdirent_t), sizeof (prdirent_t),
201         { PR_ASRS,      11 * sizeof (prdirent_t), sizeof (prdirent_t),
205                 "asrs" },
206 #endif
207 };
______unchanged_portion_omitted_

581 /*
582  * Array of read functions, indexed by /proc file type.
583  */
584 static int pr_read_inval(), pr_read_as(), pr_read_status(),
585         pr_read_lstatus(), pr_read_psinfo(), pr_read_lpsinfo(),
586         pr_read_map(), pr_read_rmap(), pr_read_xmap(),
587         pr_read_cred(), pr_read_sigact(), pr_read_auxv(),
588 #if defined(__x86)
589         pr_read_ldt(),
590 #endif
591         pr_read_usage(), pr_read_lusage(), pr_read_pagedata(),
592         pr_read_watch(), pr_read_lwpstatus(), pr_read_lwpsinfo(),
593         pr_read_lwpusage(), pr_read_lwpname(),
594         pr_read_xregs(), pr_read_priv(),
590         pr_read_lwpusage(), pr_read_xregs(), pr_read_priv(),
595         pr_read_spymaster(), pr_read_secflags(),
596 #if defined(__sparc)
597         pr_read_gwindows(), pr_read_asrs(),
598 #endif
599         pr_read_piddir(), pr_read_pidfile(), pr_read_opagedata();

601 static int (*pr_read_function[PR_NFILES])() = {
602         pr_read_inval,          /* /proc                                */
603         pr_read_inval,          /* /proc/self                           */
604         pr_read_piddir,         /* /proc/<pid> (old /proc read())       */
605         pr_read_as,             /* /proc/<pid>/as                       */
606         pr_read_inval,          /* /proc/<pid>/ctl                      */
607         pr_read_status,         /* /proc/<pid>/status                   */
608         pr_read_lstatus,        /* /proc/<pid>/lstatus                  */
609         pr_read_psinfo,         /* /proc/<pid>/psinfo                   */
610         pr_read_lpsinfo,        /* /proc/<pid>/lpsinfo                  */
611         pr_read_map,            /* /proc/<pid>/map                      */
612         pr_read_rmap,           /* /proc/<pid>/rmap                     */
613         pr_read_xmap,           /* /proc/<pid>/xmap                     */
614         pr_read_cred,           /* /proc/<pid>/cred                     */
615         pr_read_sigact,         /* /proc/<pid>/sigact                   */
616         pr_read_auxv,           /* /proc/<pid>/auxv                     */
617 #if defined(__x86)
618         pr_read_ldt,            /* /proc/<pid>/ldt                      */
619 #endif
620         pr_read_usage,          /* /proc/<pid>/usage                    */
621         pr_read_lusage,         /* /proc/<pid>/lusage                   */
622         pr_read_pagedata,       /* /proc/<pid>/pagedata                 */
623         pr_read_watch,          /* /proc/<pid>/watch                    */
624         pr_read_inval,          /* /proc/<pid>/cwd                      */
625         pr_read_inval,          /* /proc/<pid>/root                     */
626         pr_read_inval,          /* /proc/<pid>/fd                       */
627         pr_read_inval,          /* /proc/<pid>/fd/nn                    */
628         pr_read_inval,          /* /proc/<pid>/object                   */
629         pr_read_inval,          /* /proc/<pid>/object/xxx               */
```



```
630         pr_read_inval,          /* /proc/<pid>/lwp                      */
631         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>              */
632         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/lwpctl       */
633         pr_read_lwpname,        /* /proc/<pid>/lwp/<lwpid>/lwpname      */
634         pr_read_lwpstatus,      /* /proc/<pid>/lwp/<lwpid>/lwpstatus    */
635         pr_read_lwpsinfo,       /* /proc/<pid>/lwp/<lwpid>/lwpsinfo     */
636         pr_read_lwpusage,       /* /proc/<pid>/lwp/<lwpid>/lwpusage     */
637         pr_read_xregs,          /* /proc/<pid>/lwp/<lwpid>/xregs        */
638         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/templates    */
639         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/templates/<id> */
640         pr_read_spymaster,      /* /proc/<pid>/lwp/<lwpid>/spymaster    */
641 #if defined(__sparc)
642         pr_read_gwindows,       /* /proc/<pid>/lwp/<lwpid>/gwindows     */
643         pr_read_asrs,           /* /proc/<pid>/lwp/<lwpid>/asrs         */
644 #endif
645         pr_read_priv,           /* /proc/<pid>/priv                     */
646         pr_read_inval,          /* /proc/<pid>/path                     */
647         pr_read_inval,          /* /proc/<pid>/path/xxx                 */
648         pr_read_inval,          /* /proc/<pid>/contracts                */
649         pr_read_inval,          /* /proc/<pid>/contracts/<ctid>         */
650         pr_read_secflags,       /* /proc/<pid>/secflags                 */
651         pr_read_pidfile,        /* old process file                     */
652         pr_read_pidfile,        /* old lwp file                         */
653         pr_read_opagedata,      /* old pagedata file                    */
654 };
______unchanged_portion_omitted_

1074 #if defined(__x86)
1075 /*
1076  * XX64
1077  *      This is almost certainly broken for the amd64 kernel, because
1078  *      we have two kinds of LDT structures to export -- one for compatibility
1079  *      mode, and one for long mode, sigh.
1080  *
1081  *      For now let's just have a ldt of size 0 for 64-bit processes.
1076  *      For now lets just have a ldt of size 0 for 64-bit processes.
1082  */
1083 static int
1084 pr_read_ldt(prnode_t *pnp, uio_t *uiop)
1085 {
1086         proc_t *p;
1087         struct ssd *ssd;
1088         size_t size;
1089         int error;

1091         ASSERT(pnp->pr_type == PR_LDT);

1093         if ((error = prlock(pnp, ZNO)) != 0)
1094                 return (error);
1095         p = pnp->pr_common->prc_proc;

1097         mutex_exit(&p->p_lock);
1098         mutex_enter(&p->p_ldtlock);
1099         size = prnldt(p) * sizeof (struct ssd);
1100         if (uiop->uio_offset >= size) {
1101                 mutex_exit(&p->p_ldtlock);
1102                 mutex_enter(&p->p_lock);
1103                 prunlock(pnp);
1104                 return (0);
1105         }

1107         ssd = kmem_alloc(size, KM_SLEEP);
1108         prgetldt(p, ssd);
1109         mutex_exit(&p->p_ldtlock);
1110         mutex_enter(&p->p_lock);
1111         prunlock(pnp);
```

```
1113         error = pr_uioread(ssd, size, uiop);
1114         kmem_free(ssd, size);
1115         return (error);
1116 }
```

*unchanged_portion_omitted_*

```
1546 static int
1547 pr_read_lwpname(prnode_t *pnp, uio_t *uiop)
1548 {
1549         char lwpname[THREAD_NAME_MAX];
1550         kthread_t *t;
1551         int error;

1553         ASSERT(pnp->pr_type == PR_LWPNAME);

1555         if (uiop->uio_offset >= THREAD_NAME_MAX)
1556                 return (0);

1558         if ((error = prlock(pnp, ZNO)) != 0)
1559                 return (error);

1561         bzero(lwpname, sizeof (lwpname));

1563         t = pnp->pr_common->prc_thread;

1565         if (t->t_name != NULL)
1566                 (void) strlcpy(lwpname, t->t_name, sizeof (lwpname));

1568         prunlock(pnp);

1570         return (pr_uioread(lwpname, sizeof (lwpname), uiop));
1571 }

1573 /* ARGSUSED */
1574 static int
1575 pr_read_xregs(prnode_t *pnp, uio_t *uiop)
1576 {
1577 #if defined(__sparc)
1578         proc_t *p;
1579         kthread_t *t;
1580         int error;
1581         char *xreg;
1582         size_t size;

1584         ASSERT(pnp->pr_type == PR_XREGS);

1586         xreg = kmem_zalloc(sizeof (prxregset_t), KM_SLEEP);

1588         if ((error = prlock(pnp, ZNO)) != 0)
1589                 goto out;

1591         p = pnp->pr_common->prc_proc;
1592         t = pnp->pr_common->prc_thread;

1594         size = prhasx(p)? prgetprxregsize(p) : 0;
1595         if (uiop->uio_offset >= size) {
1596                 prunlock(pnp);
1597                 goto out;
1598         }

1600         /* drop p->p_lock while (possibly) touching the stack */
1601         mutex_exit(&p->p_lock);
1602         prgetprxregs(ttolwp(t), xreg);
1603         mutex_enter(&p->p_lock);
1604         prunlock(pnp);
```



```
1606         error = pr_uioread(xreg, size, uiop);
1607 out:
1608         kmem_free(xreg, sizeof (prxregset_t));
1609         return (error);
1610 #else
1611         return (0);
1612 #endif
1613 }
```

*unchanged_portion_omitted_*

```
1794 #ifdef _SYSCALL32_IMPL

1796 /*
1797  * Array of ILP32 read functions, indexed by /proc file type.
1798  */
1799 static int pr_read_status_32(),
1800         pr_read_lstatus_32(), pr_read_psinfo_32(), pr_read_lpsinfo_32(),
1801         pr_read_map_32(), pr_read_rmap_32(), pr_read_xmap_32(),
1802         pr_read_sigact_32(), pr_read_auxv_32(),
1803         pr_read_usage_32(), pr_read_lusage_32(), pr_read_pagedata_32(),
1804         pr_read_watch_32(), pr_read_lwpstatus_32(), pr_read_lwpsinfo_32(),
1805         pr_read_lwpusage_32(), pr_read_spymaster_32(),
1806 #if defined(__sparc)
1807         pr_read_gwindows_32(),
1808 #endif
1809         pr_read_opagedata_32();

1811 static int (*pr_read_function_32[PR_NFILES])() = {
1812         pr_read_inval,          /* /proc                                */
1813         pr_read_inval,          /* /proc/self                           */
1814         pr_read_piddir,         /* /proc/<pid> (old /proc read())       */
1815         pr_read_as,             /* /proc/<pid>/as                       */
1816         pr_read_inval,          /* /proc/<pid>/ctl                      */
1817         pr_read_status_32,      /* /proc/<pid>/status                   */
1818         pr_read_lstatus_32,     /* /proc/<pid>/lstatus                  */
1819         pr_read_psinfo_32,      /* /proc/<pid>/psinfo                   */
1820         pr_read_lpsinfo_32,     /* /proc/<pid>/lpsinfo                  */
1821         pr_read_map_32,         /* /proc/<pid>/map                      */
1822         pr_read_rmap_32,        /* /proc/<pid>/rmap                     */
1823         pr_read_xmap_32,        /* /proc/<pid>/xmap                     */
1824         pr_read_cred,           /* /proc/<pid>/cred                     */
1825         pr_read_sigact_32,      /* /proc/<pid>/sigact                   */
1826         pr_read_auxv_32,        /* /proc/<pid>/auxv                     */
1827 #if defined(__x86)
1828         pr_read_ldt,            /* /proc/<pid>/ldt                      */
1829 #endif
1830         pr_read_usage_32,       /* /proc/<pid>/usage                    */
1831         pr_read_lusage_32,      /* /proc/<pid>/lusage                   */
1832         pr_read_pagedata_32,    /* /proc/<pid>/pagedata                 */
1833         pr_read_watch_32,       /* /proc/<pid>/watch                    */
1834         pr_read_inval,          /* /proc/<pid>/cwd                      */
1835         pr_read_inval,          /* /proc/<pid>/root                     */
1836         pr_read_inval,          /* /proc/<pid>/fd                       */
1837         pr_read_inval,          /* /proc/<pid>/fd/nn                    */
1838         pr_read_inval,          /* /proc/<pid>/object                   */
1839         pr_read_inval,          /* /proc/<pid>/object/xxx               */
1840         pr_read_inval,          /* /proc/<pid>/lwp                      */
1841         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>              */
1842         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/lwpctl       */
1843         pr_read_lwpname,        /* /proc/<pid>/lwp/<lwpid>/lwpname      */
1844         pr_read_lwpstatus_32,   /* /proc/<pid>/lwp/<lwpid>/lwpstatus    */
1845         pr_read_lwpsinfo_32,    /* /proc/<pid>/lwp/<lwpid>/lwpsinfo     */
1846         pr_read_lwpusage_32,    /* /proc/<pid>/lwp/<lwpid>/lwpusage     */
1847         pr_read_xregs,          /* /proc/<pid>/lwp/<lwpid>/xregs        */
1848         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/templates    */
```

```
1849         pr_read_inval,          /* /proc/<pid>/lwp/<lwpid>/templates/<id> */
1850         pr_read_spymaster_32,   /* /proc/<pid>/lwp/<lwpid>/spymaster    */
1851 #if defined(__sparc)
1852         pr_read_gwindows_32,    /* /proc/<pid>/lwp/<lwpid>/gwindows     */
1853         pr_read_asrs,           /* /proc/<pid>/lwp/<lwpid>/asrs         */
1854 #endif
1855         pr_read_priv,           /* /proc/<pid>/priv                     */
1856         pr_read_inval,          /* /proc/<pid>/path                     */
1857         pr_read_inval,          /* /proc/<pid>/path/xxx                 */
1858         pr_read_inval,          /* /proc/<pid>/contracts                */
1859         pr_read_inval,          /* /proc/<pid>/contracts/<ctid>         */
1860         pr_read_secflags,       /* /proc/<pid>/secflags                 */
1861         pr_read_pidfile,        /* old process file                     */
1862         pr_read_pidfile,        /* old lwp file                         */
1863         pr_read_opagedata_32,   /* old pagedata file                    */
1864 };
```

*unchanged_portion_omitted_*

```
2756 /* Note we intentionally don't handle partial writes/updates. */
2757 static int
2758 pr_write_lwpname(prnode_t *pnp, uio_t *uiop)
2759 {
2760         kthread_t *t = NULL;
2761         char *lwpname;
2762         int error;

2764         lwpname = kmem_zalloc(THREAD_NAME_MAX, KM_SLEEP);

2766         if ((error = uiomove(lwpname, THREAD_NAME_MAX, UIO_WRITE, uiop)) != 0) {
2767                 kmem_free(lwpname, THREAD_NAME_MAX);
2768                 return (error);
2769         }

2771         /* Somebody tried to write too long a thread name... */
2772         if (lwpname[THREAD_NAME_MAX - 1] != '\0' || uiop->uio_resid > 0) {
2773                 kmem_free(lwpname, THREAD_NAME_MAX);
2774                 return (EIO);
2775         }

2777         VERIFY3U(lwpname[THREAD_NAME_MAX - 1], ==, '\0');

2779         for (size_t i = 0; lwpname[i] != '\0'; i++) {
2780                 if (!ISPRINT(lwpname[i])) {
2781                         kmem_free(lwpname, THREAD_NAME_MAX);
2782                         return (EINVAL);
2783                 }
2784         }

2786         /* Equivalent of thread_setname(), but with the ZNO magic. */
2787         if ((error = prlock(pnp, ZNO)) != 0) {
2788                 kmem_free(lwpname, THREAD_NAME_MAX);
2789                 return (error);
2790         }

2792         t = pnp->pr_common->prc_thread;
2793         if (t->t_name == NULL) {
2794                 t->t_name = lwpname;
2795         } else {
2796                 (void) strlcpy(t->t_name, lwpname, THREAD_NAME_MAX);
2797                 kmem_free(lwpname, THREAD_NAME_MAX);
2798         }

2800         prunlock(pnp);
2801         return (0);
2802 }
```



```
2804 /* ARGSUSED */
2805 static int
2806 prwrite(vnode_t *vp, uio_t *uiop, int ioflag, cred_t *cr, caller_context_t *ct)
2807 {
2808         prnode_t *pnp = VTOP(vp);
2809         int old = 0;
2810         int error;
2811         ssize_t resid;

2813         ASSERT(pnp->pr_type < PR_NFILES);

2815         /*
2816          * Only a handful of /proc files are writable, enumerate them here.
2817          */
2818         switch (pnp->pr_type) {
2819         case PR_PIDDIR:         /* directory write()s: visceral revulsion. */
2820                 ASSERT(pnp->pr_pidfile != NULL);
2821                 /* use the underlying PR_PIDFILE to write the process */
2822                 vp = pnp->pr_pidfile;
2823                 pnp = VTOP(vp);
2824                 ASSERT(pnp->pr_type == PR_PIDFILE);
2825                 /* FALLTHROUGH */
2826         case PR_PIDFILE:
2827         case PR_LWPIDFILE:
2828                 old = 1;
2829                 /* FALLTHROUGH */
2830         case PR_AS:
2831                 if ((error = prlock(pnp, ZNO)) == 0) {
2832                         proc_t *p = pnp->pr_common->prc_proc;
2833                         struct as *as = p->p_as;

2835                         if ((p->p_flag & SSYS) || as == &kas) {
2836                                 /*
2837                                  * /proc I/O cannot be done to a system process.
2838                                  */
2839                                 error = EIO;
2840 #ifdef _SYSCALL32_IMPL
2841                         } else if (curproc->p_model == DATAMODEL_ILP32 &&
2842                             PROCESS_NOT_32BIT(p)) {
2843                                 error = EOVERFLOW;
2844 #endif
2845                         } else {
2846                                 /*
2847                                  * See comments above (pr_read_pidfile)
2848                                  * about this locking dance.
2849                                  */
2850                                 mutex_exit(&p->p_lock);
2851                                 error = prusrio(p, UIO_WRITE, uiop, old);
2852                                 mutex_enter(&p->p_lock);
2853                         }
2854                         prunlock(pnp);
2855                 }
2856                 return (error);

2858         case PR_CTL:
2859         case PR_LWPCTL:
2860                 resid = uiop->uio_resid;
2861                 /*
2862                  * Perform the action on the control file
2863                  * by passing curthreads credentials
2864                  * and not target process's credentials.
2865                  */
2866 #ifdef _SYSCALL32_IMPL
2867                 if (curproc->p_model == DATAMODEL_ILP32)
2868                         error = prwritectl32(vp, uiop, CRED());
2869                 else
```

```
2870                        error = prwritectl(vp, uiop, CRED());
2871 #else
2872                error = prwritectl(vp, uiop, CRED());
2873 #endif
2874                /*
2875                 * This hack makes sure that the EINTR is passed
2876                 * all the way back to the caller's write() call.
2877                 */
2878                if (error == EINTR)
2879                        uiop->uio_resid = resid;
2880                return (error);

2882        case PR_LWPNAME:
2883                return (pr_write_lwpname(pnp, uiop));

2885        default:
2886                return ((vp->v_type == VDIR)? EISDIR : EBADF);
2887        }
2888        /* NOTREACHED */
2889 }
```

*unchanged_portion_omitted_*

```
3405 /*
3406  * Array of lookup functions, indexed by /proc file type.
3407  */
3408 static vnode_t *pr_lookup_notdir(), *pr_lookup_procdir(), *pr_lookup_piddir(),
3409        *pr_lookup_objectdir(), *pr_lookup_lwpdir(), *pr_lookup_lwpiddir(),
3410        *pr_lookup_fddir(), *pr_lookup_pathdir(), *pr_lookup_tmpldir(),
3411        *pr_lookup_ctdir();

3413 static vnode_t *(*pr_lookup_function[PR_NFILES])() = {
3414        pr_lookup_procdir,      /* /proc                                */
3415        pr_lookup_notdir,       /* /proc/self                           */
3416        pr_lookup_piddir,       /* /proc/<pid>                          */
3417        pr_lookup_notdir,       /* /proc/<pid>/as                       */
3418        pr_lookup_notdir,       /* /proc/<pid>/ctl                      */
3419        pr_lookup_notdir,       /* /proc/<pid>/status                   */
3420        pr_lookup_notdir,       /* /proc/<pid>/lstatus                  */
3421        pr_lookup_notdir,       /* /proc/<pid>/psinfo                   */
3422        pr_lookup_notdir,       /* /proc/<pid>/lpsinfo                  */
3423        pr_lookup_notdir,       /* /proc/<pid>/map                      */
3424        pr_lookup_notdir,       /* /proc/<pid>/rmap                     */
3425        pr_lookup_notdir,       /* /proc/<pid>/xmap                     */
3426        pr_lookup_notdir,       /* /proc/<pid>/cred                     */
3427        pr_lookup_notdir,       /* /proc/<pid>/sigact                   */
3428        pr_lookup_notdir,       /* /proc/<pid>/auxv                     */
3429 #if defined(__x86)
3430        pr_lookup_notdir,       /* /proc/<pid>/ldt                      */
3431 #endif
3432        pr_lookup_notdir,       /* /proc/<pid>/usage                    */
3433        pr_lookup_notdir,       /* /proc/<pid>/lusage                   */
3434        pr_lookup_notdir,       /* /proc/<pid>/pagedata                 */
3435        pr_lookup_notdir,       /* /proc/<pid>/watch                    */
3436        pr_lookup_notdir,       /* /proc/<pid>/cwd                      */
3437        pr_lookup_notdir,       /* /proc/<pid>/root                     */
3438        pr_lookup_fddir,        /* /proc/<pid>/fd                       */
3439        pr_lookup_notdir,       /* /proc/<pid>/fd/nn                    */
3440        pr_lookup_objectdir,    /* /proc/<pid>/object                   */
3441        pr_lookup_notdir,       /* /proc/<pid>/object/xxx               */
3442        pr_lookup_lwpdir,       /* /proc/<pid>/lwp                      */
3443        pr_lookup_lwpiddir,     /* /proc/<pid>/lwp/<lwpid>              */
3444        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/lwpctl       */
3445        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/lwpname      */
3446        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/lwpstatus    */
3447        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/lwpsinfo     */
3448        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/lwpusage     */
```



```
3449        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/xregs        */
3450        pr_lookup_tmpldir,      /* /proc/<pid>/lwp/<lwpid>/templates    */
3451        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/templates/<id> */
3452        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/spymaster    */
3453 #if defined(__sparc)
3454        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/gwindows     */
3455        pr_lookup_notdir,       /* /proc/<pid>/lwp/<lwpid>/asrs         */
3456 #endif
3457        pr_lookup_notdir,       /* /proc/<pid>/priv                     */
3458        pr_lookup_pathdir,      /* /proc/<pid>/path                     */
3459        pr_lookup_notdir,       /* /proc/<pid>/path/xxx                 */
3460        pr_lookup_ctdir,        /* /proc/<pid>/contracts                */
3461        pr_lookup_notdir,       /* /proc/<pid>/contracts/<ctid>         */
3462        pr_lookup_notdir,       /* /proc/<pid>/secflags                 */
3463        pr_lookup_notdir,       /* old process file                     */
3464        pr_lookup_notdir,       /* old lwp file                         */
3465        pr_lookup_notdir,       /* old pagedata file                    */
3466 };
```

*unchanged_portion_omitted_*

```
4588 #if defined(DEBUG)

4590 static  uint32_t nprnode;
4591 static  uint32_t nprcommon;

4593 #define INCREMENT(x)    atomic_inc_32(&x);
4594 #define DECREMENT(x)    atomic_dec_32(&x);

4596 #else

4598 #define INCREMENT(x)
4599 #define DECREMENT(x)

4601 #endif  /* DEBUG */

4603 /*
4604  * New /proc vnode required; allocate it and fill in most of the fields.
4605  */
4606 prnode_t *
4607 prgetnode(vnode_t *dp, prnodetype_t type)
4608 {
4609        prnode_t *pnp;
4610        prcommon_t *pcp;
4611        vnode_t *vp;
4612        ulong_t nfiles;

4614        INCREMENT(nprnode);
4615        pnp = kmem_zalloc(sizeof (prnode_t), KM_SLEEP);

4617        mutex_init(&pnp->pr_mutex, NULL, MUTEX_DEFAULT, NULL);
4618        pnp->pr_type = type;

4620        pnp->pr_vnode = vn_alloc(KM_SLEEP);

4622        vp = PTOV(pnp);
4623        vp->v_flag = VNOCACHE|VNOMAP|VNOSWAP|VNOMOUNT;
4624        vn_setops(vp, prvnodeops);
4625        vp->v_vfsp = dp->v_vfsp;
4626        vp->v_type = VPROC;
4627        vp->v_data = (caddr_t)pnp;

4629        switch (type) {
4630        case PR_PIDDIR:
4631        case PR_LWPIDDIR:
4632                /*
4633                 * We need a prcommon and a files array for each of these.
```

```
4634                 */
4635                INCREMENT(nprcommon);

4637                pcp = kmem_zalloc(sizeof (prcommon_t), KM_SLEEP);
4638                pcp->prc_refcnt = 1;
4639                pnp->pr_common = pcp;
4640                mutex_init(&pcp->prc_mutex, NULL, MUTEX_DEFAULT, NULL);
4641                cv_init(&pcp->prc_wait, NULL, CV_DEFAULT, NULL);

4643                nfiles = (type == PR_PIDDIR)? NPIDDIRFILES : NLWPIDDIRFILES;
4644                pnp->pr_files =
4645                    kmem_zalloc(nfiles * sizeof (vnode_t *), KM_SLEEP);

4647                vp->v_type = VDIR;
4648                /*
4649                 * Mode should be read-search by all, but we cannot so long
4650                 * as we must support compatibility mode with old /proc.
4651                 * Make /proc/<pid> be read by owner only, search by all.
4652                 * Make /proc/<pid>/lwp/<lwpid> read-search by all.  Also,
4653                 * set VDIROPEN on /proc/<pid> so it can be opened for writing.
4654                 */
4655                if (type == PR_PIDDIR) {
4656                        /* kludge for old /proc interface */
4657                        prnode_t *xpnp = prgetnode(dp, PR_PIDFILE);
4658                        pnp->pr_pidfile = PTOV(xpnp);
4659                        pnp->pr_mode = 0511;
4660                        vp->v_flag |= VDIROPEN;
4661                } else {
4662                        pnp->pr_mode = 0555;
4663                }

4665                break;

4667        case PR_CURDIR:
4668        case PR_ROOTDIR:
4669        case PR_FDDIR:
4670        case PR_OBJECTDIR:
4671        case PR_PATHDIR:
4672        case PR_CTDIR:
4673        case PR_TMPLDIR:
4674                vp->v_type = VDIR;
4675                pnp->pr_mode = 0500;    /* read-search by owner only */
4676                break;

4678        case PR_CT:
4679                vp->v_type = VLNK;
4680                pnp->pr_mode = 0500;    /* read-search by owner only */
4681                break;

4683        case PR_PATH:
4684        case PR_SELF:
4685                vp->v_type = VLNK;
4686                pnp->pr_mode = 0777;
4687                break;

4689        case PR_LWPDIR:
4690                vp->v_type = VDIR;
4691                pnp->pr_mode = 0555;    /* read-search by all */
4692                break;

4694        case PR_AS:
4695        case PR_TMPL:
4696                pnp->pr_mode = 0600;    /* read-write by owner only */
4697                break;

4699        case PR_CTL:
```



```
4700        case PR_LWPCTL:
4701                pnp->pr_mode = 0200;    /* write-only by owner only */
4702                break;

4704        case PR_PIDFILE:
4705        case PR_LWPIDFILE:
4706                pnp->pr_mode = 0600;    /* read-write by owner only */
4707                break;

4709        case PR_LWPNAME:
4710                pnp->pr_mode = 0644;    /* readable by all + owner can write */
4711                break;

4713        case PR_PSINFO:
4714        case PR_LPSINFO:
4715        case PR_LWPSINFO:
4716        case PR_USAGE:
4717        case PR_LUSAGE:
4718        case PR_LWPUSAGE:
4719                pnp->pr_mode = 0444;    /* read-only by all */
4720                break;

4722        default:
4723                pnp->pr_mode = 0400;    /* read-only by owner only */
4724                break;
4725        }
4726        vn_exists(vp);
4727        return (pnp);
4728 }
_____unchanged_portion_omitted_

4797 /*
4798  * Array of readdir functions, indexed by /proc file type.
4799  */
4800 static int pr_readdir_notdir(), pr_readdir_procdir(), pr_readdir_piddir(),
4801        pr_readdir_objectdir(), pr_readdir_lwpdir(), pr_readdir_lwpiddir(),
4802        pr_readdir_fddir(), pr_readdir_pathdir(), pr_readdir_tmpldir(),
4803        pr_readdir_ctdir();

4805 static int (*pr_readdir_function[PR_NFILES])() = {
4806        pr_readdir_procdir,     /* /proc                                */
4807        pr_readdir_notdir,      /* /proc/self                           */
4808        pr_readdir_piddir,      /* /proc/<pid>                          */
4809        pr_readdir_notdir,      /* /proc/<pid>/as                       */
4810        pr_readdir_notdir,      /* /proc/<pid>/ctl                      */
4811        pr_readdir_notdir,      /* /proc/<pid>/status                   */
4812        pr_readdir_notdir,      /* /proc/<pid>/lstatus                  */
4813        pr_readdir_notdir,      /* /proc/<pid>/psinfo                   */
4814        pr_readdir_notdir,      /* /proc/<pid>/lpsinfo                  */
4815        pr_readdir_notdir,      /* /proc/<pid>/map                      */
4816        pr_readdir_notdir,      /* /proc/<pid>/rmap                     */
4817        pr_readdir_notdir,      /* /proc/<pid>/xmap                     */
4818        pr_readdir_notdir,      /* /proc/<pid>/cred                     */
4819        pr_readdir_notdir,      /* /proc/<pid>/sigact                   */
4820        pr_readdir_notdir,      /* /proc/<pid>/auxv                     */
4821 #if defined(__x86)
4822        pr_readdir_notdir,      /* /proc/<pid>/ldt                      */
4823 #endif
4824        pr_readdir_notdir,      /* /proc/<pid>/usage                    */
4825        pr_readdir_notdir,      /* /proc/<pid>/lusage                   */
4826        pr_readdir_notdir,      /* /proc/<pid>/pagedata                 */
4827        pr_readdir_notdir,      /* /proc/<pid>/watch                    */
4828        pr_readdir_notdir,      /* /proc/<pid>/cwd                      */
4829        pr_readdir_notdir,      /* /proc/<pid>/root                     */
4830        pr_readdir_fddir,       /* /proc/<pid>/fd                       */
4831        pr_readdir_notdir,      /* /proc/<pid>/fd/nn                    */
```

```
4832         pr_readdir_objectdir,   /* /proc/<pid>/object                  */
4833         pr_readdir_notdir,      /* /proc/<pid>/object/xxx              */
4834         pr_readdir_lwpdir,      /* /proc/<pid>/lwp                     */
4835         pr_readdir_lwpiddir,    /* /proc/<pid>/lwp/<lwpid>             */
4836         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/lwpctl      */
4837         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/lwpname     */
4838         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/lwpstatus   */
4839         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/lwpsinfo    */
4840         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/lwpusage    */
4841         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/xregs       */
4842         pr_readdir_tmpldir,     /* /proc/<pid>/lwp/<lwpid>/templates   */
4843         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/templates/<id> */
4844         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/spymaster   */
4845 #if defined(__sparc)
4846         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/gwindows    */
4847         pr_readdir_notdir,      /* /proc/<pid>/lwp/<lwpid>/asrs        */
4848 #endif
4849         pr_readdir_notdir,      /* /proc/<pid>/priv                    */
4850         pr_readdir_pathdir,     /* /proc/<pid>/path                    */
4851         pr_readdir_notdir,      /* /proc/<pid>/path/xxx                */
4852         pr_readdir_ctdir,       /* /proc/<pid>/contracts               */
4853         pr_readdir_notdir,      /* /proc/<pid>/contracts/<ctid>        */
4854         pr_readdir_notdir,      /* /proc/<pid>/secflags                */
4855         pr_readdir_notdir,      /* old process file                    */
4856         pr_readdir_notdir,      /* old lwp file                        */
4857         pr_readdir_notdir,      /* old pagedata file                   */
4858 };
______unchanged_portion_omitted_
```

```
**********************************************************
  102808 Mon Oct 15 13:28:42 2018
new/usr/src/uts/common/sys/dtrace.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2018 Joyent, Inc.
 * Copyright 2017 Joyent, Inc.
 * Copyright (c) 2013 by Delphix. All rights reserved.
 */

#ifndef _SYS_DTRACE_H
#define _SYS_DTRACE_H

#ifdef __cplusplus
extern "C" {
#endif

/*
 * DTrace Dynamic Tracing Software: Kernel Interfaces
 *
 * Note: The contents of this file are private to the implementation of the
 * Solaris system and DTrace subsystem and are subject to change at any time
 * without notice.  Applications and drivers using these interfaces will fail
 * to run on future releases.  These interfaces should not be used for any
 * purpose except those expressly outlined in dtrace(7D) and libdtrace(3LIB).
 * Please refer to the "Solaris Dynamic Tracing Guide" for more information.
 */

#ifndef _ASM

#include <sys/types.h>
#include <sys/modctl.h>
#include <sys/processor.h>
#include <sys/systm.h>
#include <sys/ctf_api.h>
#include <sys/cyclic.h>
#include <sys/int_limits.h>
```



```
/*
 * DTrace Universal Constants and Typedefs
 */
#define DTRACE_CPUALL           -1      /* all CPUs */
#define DTRACE_IDNONE           0       /* invalid probe identifier */
#define DTRACE_EPIDNONE         0       /* invalid enabled probe identifier */
#define DTRACE_AGGIDNONE        0       /* invalid aggregation identifier */
#define DTRACE_AGGVARIDNONE     0       /* invalid aggregation variable ID */
#define DTRACE_CACHEIDNONE      0       /* invalid predicate cache */
#define DTRACE_PROVNONE         0       /* invalid provider identifier */
#define DTRACE_METAPROVNONE     0       /* invalid meta-provider identifier */
#define DTRACE_ARGNONE          -1      /* invalid argument index */

#define DTRACE_PROVNAMELEN      64
#define DTRACE_MODNAMELEN       64
#define DTRACE_FUNCNAMELEN      128
#define DTRACE_NAMELEN          64
#define DTRACE_FULLNAMELEN      (DTRACE_PROVNAMELEN + DTRACE_MODNAMELEN + \
                                 DTRACE_FUNCNAMELEN + DTRACE_NAMELEN + 4)
#define DTRACE_ARGTYPELEN       128

typedef uint32_t dtrace_id_t;           /* probe identifier */
typedef uint32_t dtrace_epid_t;         /* enabled probe identifier */
typedef uint32_t dtrace_aggid_t;        /* aggregation identifier */
typedef int64_t dtrace_aggvarid_t;      /* aggregation variable identifier */
typedef uint16_t dtrace_actkind_t;      /* action kind */
typedef int64_t dtrace_optval_t;        /* option value */
typedef uint32_t dtrace_cacheid_t;      /* predicate cache identifier */

typedef enum dtrace_probespec {
        DTRACE_PROBESPEC_NONE = -1,
        DTRACE_PROBESPEC_PROVIDER = 0,
        DTRACE_PROBESPEC_MOD,
        DTRACE_PROBESPEC_FUNC,
        DTRACE_PROBESPEC_NAME
} dtrace_probespec_t;

/*
 * DTrace Intermediate Format (DIF)
 *
 * The following definitions describe the DTrace Intermediate Format (DIF), a
 * a RISC-like instruction set and program encoding used to represent
 * predicates and actions that can be bound to DTrace probes.  The constants
 * below defining the number of available registers are suggested minimums; the
 * compiler should use DTRACEIOC_CONF to dynamically obtain the number of
 * registers provided by the current DTrace implementation.
 */
#define DIF_VERSION_1   1               /* DIF version 1: Solaris 10 Beta */
#define DIF_VERSION_2   2               /* DIF version 2: Solaris 10 FCS */
#define DIF_VERSION     DIF_VERSION_2   /* latest DIF instruction set version */
#define DIF_DIR_NREGS   8               /* number of DIF integer registers */
#define DIF_DTR_NREGS   8               /* number of DIF tuple registers */

#define DIF_OP_OR       1               /* or   r1, r2, rd */
#define DIF_OP_XOR      2               /* xor  r1, r2, rd */
#define DIF_OP_AND      3               /* and  r1, r2, rd */
#define DIF_OP_SLL      4               /* sll  r1, r2, rd */
#define DIF_OP_SRL      5               /* srl  r1, r2, rd */
#define DIF_OP_SUB      6               /* sub  r1, r2, rd */
#define DIF_OP_ADD      7               /* add  r1, r2, rd */
#define DIF_OP_MUL      8               /* mul  r1, r2, rd */
#define DIF_OP_SDIV     9               /* sdiv r1, r2, rd */
#define DIF_OP_UDIV     10              /* udiv r1, r2, rd */
#define DIF_OP_SREM     11              /* srem r1, r2, rd */
#define DIF_OP_UREM     12              /* urem r1, r2, rd */
#define DIF_OP_NOT      13              /* not  r1, rd */
```

```
126 #define DIF_OP_MOV      14              /* mov  r1, rd */
127 #define DIF_OP_CMP      15              /* cmp  r1, r2 */
128 #define DIF_OP_TST      16              /* tst  r1 */
129 #define DIF_OP_BA       17              /* ba   label */
130 #define DIF_OP_BE       18              /* be   label */
131 #define DIF_OP_BNE      19              /* bne  label */
132 #define DIF_OP_BG       20              /* bg   label */
133 #define DIF_OP_BGU      21              /* bgu  label */
134 #define DIF_OP_BGE      22              /* bge  label */
135 #define DIF_OP_BGEU     23              /* bgeu label */
136 #define DIF_OP_BL       24              /* bl   label */
137 #define DIF_OP_BLU      25              /* blu  label */
138 #define DIF_OP_BLE      26              /* ble  label */
139 #define DIF_OP_BLEU     27              /* bleu label */
140 #define DIF_OP_LDSB     28              /* ldsb [r1], rd */
141 #define DIF_OP_LDSH     29              /* ldsh [r1], rd */
142 #define DIF_OP_LDSW     30              /* ldsw [r1], rd */
143 #define DIF_OP_LDUB     31              /* ldub [r1], rd */
144 #define DIF_OP_LDUH     32              /* lduh [r1], rd */
145 #define DIF_OP_LDUW     33              /* lduw [r1], rd */
146 #define DIF_OP_LDX      34              /* ldx  [r1], rd */
147 #define DIF_OP_RET      35              /* ret  rd */
148 #define DIF_OP_NOP      36              /* nop */
149 #define DIF_OP_SETX     37              /* setx intindex, rd */
150 #define DIF_OP_SETS     38              /* sets strindex, rd */
151 #define DIF_OP_SCMP     39              /* scmp r1, r2 */
152 #define DIF_OP_LDGA     40              /* ldga var, ri, rd */
153 #define DIF_OP_LDGS     41              /* ldgs var, rd */
154 #define DIF_OP_STGS     42              /* stgs var, rs */
155 #define DIF_OP_LDTA     43              /* ldta var, ri, rd */
156 #define DIF_OP_LDTS     44              /* ldts var, rd */
157 #define DIF_OP_STTS     45              /* stts var, rs */
158 #define DIF_OP_SRA      46              /* sra  r1, r2, rd */
159 #define DIF_OP_CALL     47              /* call subr, rd */
160 #define DIF_OP_PUSHTR   48              /* pushtr type, rs, rr */
161 #define DIF_OP_PUSHTV   49              /* pushtv type, rs, rv */
162 #define DIF_OP_POPTS    50              /* popts */
163 #define DIF_OP_FLUSHTS  51              /* flushts */
164 #define DIF_OP_LDGAA    52              /* ldgaa var, rd */
165 #define DIF_OP_LDTAA    53              /* ldtaa var, rd */
166 #define DIF_OP_STGAA    54              /* stgaa var, rs */
167 #define DIF_OP_STTAA    55              /* sttaa var, rs */
168 #define DIF_OP_LDLS     56              /* ldls var, rd */
169 #define DIF_OP_STLS     57              /* stls var, rs */
170 #define DIF_OP_ALLOCS   58              /* allocs r1, rd */
171 #define DIF_OP_COPYS    59              /* copys  r1, r2, rd */
172 #define DIF_OP_STB      60              /* stb  r1, [rd] */
173 #define DIF_OP_STH      61              /* sth  r1, [rd] */
174 #define DIF_OP_STW      62              /* stw  r1, [rd] */
175 #define DIF_OP_STX      63              /* stx  r1, [rd] */
176 #define DIF_OP_ULDSB    64              /* uldsb [r1], rd */
177 #define DIF_OP_ULDSH    65              /* uldsh [r1], rd */
178 #define DIF_OP_ULDSW    66              /* uldsw [r1], rd */
179 #define DIF_OP_ULDUB    67              /* uldub [r1], rd */
180 #define DIF_OP_ULDUH    68              /* ulduh [r1], rd */
181 #define DIF_OP_ULDUW    69              /* ulduw [r1], rd */
182 #define DIF_OP_ULDX     70              /* uldx  [r1], rd */
183 #define DIF_OP_RLDSB    71              /* rldsb [r1], rd */
184 #define DIF_OP_RLDSH    72              /* rldsh [r1], rd */
185 #define DIF_OP_RLDSW    73              /* rldsw [r1], rd */
186 #define DIF_OP_RLDUB    74              /* rldub [r1], rd */
187 #define DIF_OP_RLDUH    75              /* rlduh [r1], rd */
188 #define DIF_OP_RLDUW    76              /* rlduw [r1], rd */
189 #define DIF_OP_RLDX     77              /* rldx  [r1], rd */
190 #define DIF_OP_XLATE    78              /* xlate xlrindex, rd */
191 #define DIF_OP_XLARG    79              /* xlarg xlrindex, rd */
```



```
192 #define DIF_OP_STGA     80              /* stga var, ri, rd */

194 #define DIF_INTOFF_MAX          0xffff  /* highest integer table offset */
195 #define DIF_STROFF_MAX          0xffff  /* highest string table offset */
196 #define DIF_REGISTER_MAX        0xff    /* highest register number */
197 #define DIF_VARIABLE_MAX        0xffff  /* highest variable identifier */
198 #define DIF_SUBROUTINE_MAX      0xffff  /* highest subroutine code */

200 #define DIF_VAR_ARRAY_MIN       0x0000  /* lowest numbered array variable */
201 #define DIF_VAR_ARRAY_UBASE     0x0080  /* lowest user-defined array */
202 #define DIF_VAR_ARRAY_MAX       0x00ff  /* highest numbered array variable */

204 #define DIF_VAR_OTHER_MIN       0x0100  /* lowest numbered scalar or assc */
205 #define DIF_VAR_OTHER_UBASE     0x0500  /* lowest user-defined scalar or assc */
206 #define DIF_VAR_OTHER_MAX       0xffff  /* highest numbered scalar or assc */

208 #define DIF_VAR_ARGS            0x0000  /* arguments array */
209 #define DIF_VAR_REGS            0x0001  /* registers array */
210 #define DIF_VAR_UREGS           0x0002  /* user registers array */
211 #define DIF_VAR_VMREGS          0x0003  /* virtual machine registers array */
212 #define DIF_VAR_CURTHREAD       0x0100  /* thread pointer */
213 #define DIF_VAR_TIMESTAMP       0x0101  /* timestamp */
214 #define DIF_VAR_VTIMESTAMP      0x0102  /* virtual timestamp */
215 #define DIF_VAR_IPL             0x0103  /* interrupt priority level */
216 #define DIF_VAR_EPID            0x0104  /* enabled probe ID */
217 #define DIF_VAR_ID              0x0105  /* probe ID */
218 #define DIF_VAR_ARG0            0x0106  /* first argument */
219 #define DIF_VAR_ARG1            0x0107  /* second argument */
220 #define DIF_VAR_ARG2            0x0108  /* third argument */
221 #define DIF_VAR_ARG3            0x0109  /* fourth argument */
222 #define DIF_VAR_ARG4            0x010a  /* fifth argument */
223 #define DIF_VAR_ARG5            0x010b  /* sixth argument */
224 #define DIF_VAR_ARG6            0x010c  /* seventh argument */
225 #define DIF_VAR_ARG7            0x010d  /* eighth argument */
226 #define DIF_VAR_ARG8            0x010e  /* ninth argument */
227 #define DIF_VAR_ARG9            0x010f  /* tenth argument */
228 #define DIF_VAR_STACKDEPTH      0x0110  /* stack depth */
229 #define DIF_VAR_CALLER          0x0111  /* caller */
230 #define DIF_VAR_PROBEPROV       0x0112  /* probe provider */
231 #define DIF_VAR_PROBEMOD        0x0113  /* probe module */
232 #define DIF_VAR_PROBEFUNC       0x0114  /* probe function */
233 #define DIF_VAR_PROBENAME       0x0115  /* probe name */
234 #define DIF_VAR_PID             0x0116  /* process ID */
235 #define DIF_VAR_TID             0x0117  /* (per-process) thread ID */
236 #define DIF_VAR_EXECNAME        0x0118  /* name of executable */
237 #define DIF_VAR_ZONENAME        0x0119  /* zone name associated with process */
238 #define DIF_VAR_WALLTIMESTAMP   0x011a  /* wall-clock timestamp */
239 #define DIF_VAR_USTACKDEPTH     0x011b  /* user-land stack depth */
240 #define DIF_VAR_UCALLER         0x011c  /* user-level caller */
241 #define DIF_VAR_PPID            0x011d  /* parent process ID */
242 #define DIF_VAR_UID             0x011e  /* process user ID */
243 #define DIF_VAR_GID             0x011f  /* process group ID */
244 #define DIF_VAR_ERRNO           0x0120  /* thread errno */
245 #define DIF_VAR_THREADNAME      0x0121  /* thread name */

247 #define DIF_SUBR_RAND                   0
248 #define DIF_SUBR_MUTEX_OWNED            1
249 #define DIF_SUBR_MUTEX_OWNER            2
250 #define DIF_SUBR_MUTEX_TYPE_ADAPTIVE    3
251 #define DIF_SUBR_MUTEX_TYPE_SPIN        4
252 #define DIF_SUBR_RW_READ_HELD           5
253 #define DIF_SUBR_RW_WRITE_HELD          6
254 #define DIF_SUBR_RW_ISWRITER            7
255 #define DIF_SUBR_COPYIN                 8
256 #define DIF_SUBR_COPYINSTR              9
257 #define DIF_SUBR_SPECULATION            10
```

```
 258 #define DIF_SUBR_PROGENYOF              11
 259 #define DIF_SUBR_STRLEN                 12
 260 #define DIF_SUBR_COPYOUT                13
 261 #define DIF_SUBR_COPYOUTSTR             14
 262 #define DIF_SUBR_ALLOCA                 15
 263 #define DIF_SUBR_BCOPY                  16
 264 #define DIF_SUBR_COPYINTO               17
 265 #define DIF_SUBR_MSGDSIZE               18
 266 #define DIF_SUBR_MSGSIZE                19
 267 #define DIF_SUBR_GETMAJOR               20
 268 #define DIF_SUBR_GETMINOR               21
 269 #define DIF_SUBR_DDI_PATHNAME           22
 270 #define DIF_SUBR_STRJOIN                23
 271 #define DIF_SUBR_LLTOSTR                24
 272 #define DIF_SUBR_BASENAME               25
 273 #define DIF_SUBR_DIRNAME                26
 274 #define DIF_SUBR_CLEANPATH              27
 275 #define DIF_SUBR_STRCHR                 28
 276 #define DIF_SUBR_STRRCHR                29
 277 #define DIF_SUBR_STRSTR                 30
 278 #define DIF_SUBR_STRTOK                 31
 279 #define DIF_SUBR_SUBSTR                 32
 280 #define DIF_SUBR_INDEX                  33
 281 #define DIF_SUBR_RINDEX                 34
 282 #define DIF_SUBR_HTONS                  35
 283 #define DIF_SUBR_HTONL                  36
 284 #define DIF_SUBR_HTONLL                 37
 285 #define DIF_SUBR_NTOHS                  38
 286 #define DIF_SUBR_NTOHL                  39
 287 #define DIF_SUBR_NTOHLL                 40
 288 #define DIF_SUBR_INET_NTOP              41
 289 #define DIF_SUBR_INET_NTOA              42
 290 #define DIF_SUBR_INET_NTOA6             43
 291 #define DIF_SUBR_TOUPPER                44
 292 #define DIF_SUBR_TOLOWER                45
 293 #define DIF_SUBR_GETF                   46
 294 #define DIF_SUBR_JSON                   47
 295 #define DIF_SUBR_STRTOLL                48

 297 #define DIF_SUBR_MAX                    48      /* max subroutine value */

 299 typedef uint32_t dif_instr_t;

 301 #define DIF_INSTR_OP(i)                 (((i) >> 24) & 0xff)
 302 #define DIF_INSTR_R1(i)                 (((i) >> 16) & 0xff)
 303 #define DIF_INSTR_R2(i)                 (((i) >>  8) & 0xff)
 304 #define DIF_INSTR_RD(i)                 ((i) & 0xff)
 305 #define DIF_INSTR_RS(i)                 ((i) & 0xff)
 306 #define DIF_INSTR_LABEL(i)              ((i) & 0xffffff)
 307 #define DIF_INSTR_VAR(i)                (((i) >>  8) & 0xffff)
 308 #define DIF_INSTR_INTEGER(i)            (((i) >>  8) & 0xffff)
 309 #define DIF_INSTR_STRING(i)             (((i) >>  8) & 0xffff)
 310 #define DIF_INSTR_SUBR(i)               (((i) >>  8) & 0xffff)
 311 #define DIF_INSTR_TYPE(i)               (((i) >> 16) & 0xff)
 312 #define DIF_INSTR_XLREF(i)              (((i) >>  8) & 0xffff)

 314 #define DIF_INSTR_FMT(op, r1, r2, d) \
 315         (((op) << 24) | ((r1) << 16) | ((r2) << 8) | (d))

 317 #define DIF_INSTR_NOT(r1, d)            (DIF_INSTR_FMT(DIF_OP_NOT, r1, 0, d))
 318 #define DIF_INSTR_MOV(r1, d)            (DIF_INSTR_FMT(DIF_OP_MOV, r1, 0, d))
 319 #define DIF_INSTR_CMP(op, r1, r2)       (DIF_INSTR_FMT(op, r1, r2, 0))
 320 #define DIF_INSTR_TST(r1)               (DIF_INSTR_FMT(DIF_OP_TST, r1, 0, 0))
 321 #define DIF_INSTR_BRANCH(op, label)     (((op) << 24) | (label))
 322 #define DIF_INSTR_LOAD(op, r1, d)       (DIF_INSTR_FMT(op, r1, 0, d))
 323 #define DIF_INSTR_STORE(op, r1, d)      (DIF_INSTR_FMT(op, r1, 0, d))
```



```
 324 #define DIF_INSTR_SETX(i, d)            ((DIF_OP_SETX << 24) | ((i) << 8) | (d))
 325 #define DIF_INSTR_SETS(s, d)            ((DIF_OP_SETS << 24) | ((s) << 8) | (d))
 326 #define DIF_INSTR_RET(d)                (DIF_INSTR_FMT(DIF_OP_RET, 0, 0, d))
 327 #define DIF_INSTR_NOP                   (DIF_OP_NOP << 24)
 328 #define DIF_INSTR_LDA(op, v, r, d)      (DIF_INSTR_FMT(op, v, r, d))
 329 #define DIF_INSTR_LDV(op, v, d)         (((op) << 24) | ((v) << 8) | (d))
 330 #define DIF_INSTR_STV(op, v, rs)        (((op) << 24) | ((v) << 8) | (rs))
 331 #define DIF_INSTR_CALL(s, d)            ((DIF_OP_CALL << 24) | ((s) << 8) | (d))
 332 #define DIF_INSTR_PUSHTS(op, t, r2, rs) (DIF_INSTR_FMT(op, t, r2, rs))
 333 #define DIF_INSTR_POPTS                 (DIF_OP_POPTS << 24)
 334 #define DIF_INSTR_FLUSHTS               (DIF_OP_FLUSHTS << 24)
 335 #define DIF_INSTR_ALLOCS(r1, d)         (DIF_INSTR_FMT(DIF_OP_ALLOCS, r1, 0, d))
 336 #define DIF_INSTR_COPYS(r1, r2, d)      (DIF_INSTR_FMT(DIF_OP_COPYS, r1, r2, d))
 337 #define DIF_INSTR_XLATE(op, r, d)       (((op) << 24) | ((r) << 8) | (d))

 339 #define DIF_REG_R0      0               /* %r0 is always set to zero */

 341 /*
 342  * A DTrace Intermediate Format Type (DIF Type) is used to represent the types
 343  * of variables, function and associative array arguments, and the return type
 344  * for each DIF object (shown below).  It contains a description of the type,
 345  * its size in bytes, and a module identifier.
 346  */
 347 typedef struct dtrace_diftype {
 348         uint8_t dtdt_kind;              /* type kind (see below) */
 349         uint8_t dtdt_ckind;             /* type kind in CTF */
 350         uint8_t dtdt_flags;             /* type flags (see below) */
 351         uint8_t dtdt_pad;               /* reserved for future use */
 352         uint32_t dtdt_size;             /* type size in bytes (unless string) */
 353 } dtrace_diftype_t;
```

*unchanged_portion_omitted_*

```
**********************************************************
   33678 Mon Oct 15 13:28:43 2018
new/usr/src/uts/common/sys/elf.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
______unchanged_portion_omitted_

 893 typedef Elf64_Xword     Elf64_Capinfo;
 894 typedef Elf64_Word      Elf64_Capchain;

 896 /*
 897  *      Macros to compose and decompose values for capabilities info.
 898  *
 899  *      sym = ELF64_C_SYM(info)
 900  *      grp = ELF64_C_GROUP(info)
 901  *      info = ELF64_C_INFO(sym, grp)
 902  */
 903 #define ELF64_C_SYM(info)       ((info)>>32)
 904 #define ELF64_C_GROUP(info)     ((Elf64_Word)(info))
 905 #define ELF64_C_INFO(sym, grp)  (((Elf64_Xword)(sym)<<32)+(Elf64_Xword)(grp))

 907 #endif  /* defined(_LP64) || defined(_LONGLONG_TYPE) */
 908 #endif

 910 /*
 911  * Version numbers for SHT_SUNW_capinfo and SHT_SUNW_capchain.
 912  */
 913 #define CAPINFO_NONE            0
 914 #define CAPINFO_CURRENT         1
 915 #define CAPINFO_NUM             2

 917 #define CAPCHAIN_NONE           0
 918 #define CAPCHAIN_CURRENT        1
 919 #define CAPCHAIN_NUM            2

 921 /*
 922  * A SHT_SUNW_capinfo table mirrors a symbol table.  A capabilities symbol has
 923  * a SHT_SUNW_capinfo table entry that provides an index into the associated
 924  * SHT_SUNW_cap capabilities group, and the symbol index of the associated lead
 925  * symbol.  A capabilities symbol is a local symbol.  A global lead capabilities
 926  * symbol is tagged with a group CAPINFO_SUNW_GLOB.
 927  */
 928 #define CAPINFO_SUNW_GLOB       0xff

 930 /*
 931  * Capabilities values.
 932  */
 933 #define CA_SUNW_NULL    0
 934 #define CA_SUNW_HW_1    1               /* first hardware capabilities entry */
 935 #define CA_SUNW_SF_1    2               /* first software capabilities entry */
 936 #define CA_SUNW_HW_2    3               /* second hardware capabilities entry */
 937 #define CA_SUNW_PLAT    4               /* platform capability entry */
 938 #define CA_SUNW_MACH    5               /* machine capability entry */
 939 #define CA_SUNW_ID      6               /* capability identifier */
 940 #define CA_SUNW_NUM     7

 942 /*
 943  * Define software capabilities (CA_SUNW_SF_1 values).  Note, hardware
 944  * capabilities (CA_SUNW_HW_1 values) are taken directly from sys/auxv_$MACH.h.
 945  */
 946 #define SF1_SUNW_FPKNWN 0x001           /* frame pointer usage is known */
 947 #define SF1_SUNW_FPUSED 0x002           /* frame pointer is in use */
 948 #define SF1_SUNW_ADDR32 0x004           /* 32-bit address space requirement */
 949 #define SF1_SUNW_MASK   0x007           /* known software capabilities mask */
```



```
 951 /*
 952  *      Known values for note entry types (e_type == ET_CORE)
 953  */
 954 #define NT_PRSTATUS     1       /* prstatus_t   <sys/old_procfs.h>      */
 955 #define NT_PRFPREG      2       /* prfpregset_t <sys/old_procfs.h>      */
 956 #define NT_PRPSINFO     3       /* prpsinfo_t   <sys/old_procfs.h>      */
 957 #define NT_PRXREG       4       /* prxregset_t  <sys/procfs.h>          */
 958 #define NT_PLATFORM     5       /* string from sysinfo(SI_PLATFORM)     */
 959 #define NT_AUXV         6       /* auxv_t array <sys/auxv.h>            */
 960 #define NT_GWINDOWS     7       /* gwindows_t   SPARC only              */
 961 #define NT_ASRS         8       /* asrset_t     SPARC V9 only           */
 962 #define NT_LDT          9       /* ssd array    <sys/sysi86.h> IA32 only */
 963 #define NT_PSTATUS      10      /* pstatus_t    <sys/procfs.h>          */
 964 #define NT_PSINFO       13      /* psinfo_t     <sys/procfs.h>          */
 965 #define NT_PRCRED       14      /* prcred_t     <sys/procfs.h>          */
 966 #define NT_UTSNAME      15      /* struct utsname <sys/utsname.h>       */
 967 #define NT_LWPSTATUS    16      /* lwpstatus_t  <sys/procfs.h>          */
 968 #define NT_LWPSINFO     17      /* lwpsinfo_t   <sys/procfs.h>          */
 969 #define NT_PRPRIV       18      /* prpriv_t     <sys/procfs.h>          */
 970 #define NT_PRPRIVINFO   19      /* priv_impl_info_t <sys/priv.h>        */
 971 #define NT_CONTENT      20      /* core_content_t <sys/corectl.h>       */
 972 #define NT_ZONENAME     21      /* string from getzonenamebyid(3C)      */
 973 #define NT_FDINFO       22      /* open fd info                         */
 974 #define NT_SPYMASTER    23      /* psinfo_t for agent LWP spymaster     */
 975 #define NT_SECFLAGS     24      /* process security-flags */
 976 #define NT_LWPNAME      25      /* prlwpname_t */
 977 #define NT_NUM          25
 976 #define NT_NUM          24


 980 #ifdef _KERNEL
 981 /*
 982  * The following routine checks the processor-specific
 983  * fields of an ELF header.
 984  */
 985 int     elfheadcheck(unsigned char, Elf32_Half, Elf32_Word);
 986 #endif

 988 #ifdef  __cplusplus
 989 }
______unchanged_portion_omitted_
```

```
**********************************************************
   35624 Mon Oct 15 13:28:44 2018
new/usr/src/uts/common/sys/procfs.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2012 DEY Storage Systems, Inc.  All rights reserved.
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _SYS_PROCFS_H
#define _SYS_PROCFS_H

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * This definition is temporary.  Structured proc is the preferred API,
 * and the older ioctl-based interface will be removed in a future version
 * of Solaris.  Until then, by default, including <sys/procfs.h> will
 * provide the older ioctl-based /proc definitions.  To get the structured
 * /proc definitions, either include <procfs.h> or define _STRUCTURED_PROC
 * to be 1 before including <sys/procfs.h>.
 */
#ifndef _STRUCTURED_PROC
#define _STRUCTURED_PROC        0
#endif

#if !defined(_KERNEL) && _STRUCTURED_PROC == 0

#include <sys/old_procfs.h>

#else   /* !defined(_KERNEL) && _STRUCTURED_PROC == 0 */

#include <sys/feature_tests.h>
#include <sys/types.h>
#include <sys/time_impl.h>
#include <sys/signal.h>
#include <sys/siginfo.h>
```



```
#include <sys/fault.h>
#include <sys/syscall.h>
#include <sys/pset.h>
#include <sys/procfs_isa.h>
#include <sys/priv.h>
#include <sys/stat.h>
#include <sys/param.h>
#include <sys/secflags.h>
#include <sys/thread.h>

/*
 * System call interfaces for /proc.
 */

/*
 * Control codes (long values) for messages written to ctl and lwpctl files.
 */
#define PCNULL   0L     /* null request, advance to next message */
#define PCSTOP   1L     /* direct process or lwp to stop and wait for stop */
#define PCDSTOP  2L     /* direct process or lwp to stop */
#define PCWSTOP  3L     /* wait for process or lwp to stop, no timeout */
#define PCTWSTOP 4L     /* wait for stop, with long millisecond timeout arg */
#define PCRUN    5L     /* make process/lwp runnable, w/ long flags argument */
#define PCCSIG   6L     /* clear current signal from lwp */
#define PCCFAULT 7L     /* clear current fault from lwp */
#define PCSSIG   8L     /* set current signal from siginfo_t argument */
#define PCKILL   9L     /* post a signal to process/lwp, long argument */
#define PCUNKILL 10L    /* delete a pending signal from process/lwp, long arg */
#define PCSHOLD  11L    /* set lwp signal mask from sigset_t argument */
#define PCSTRACE 12L    /* set traced signal set from sigset_t argument */
#define PCSFAULT 13L    /* set traced fault set from fltset_t argument */
#define PCSENTRY 14L    /* set traced syscall entry set from sysset_t arg */
#define PCSEXIT  15L    /* set traced syscall exit set from sysset_t arg */
#define PCSET    16L    /* set modes from long argument */
#define PCUNSET  17L    /* unset modes from long argument */
#define PCSREG   18L    /* set lwp general registers from prgregset_t arg */
#define PCSFPREG 19L    /* set lwp floating-point registers from prfpregset_t */
#define PCSXREG  20L    /* set lwp extra registers from prxregset_t arg */
#define PCNICE   21L    /* set nice priority from long argument */
#define PCSVADDR 22L    /* set %pc virtual address from long argument */
#define PCWATCH  23L    /* set/unset watched memory area from prwatch_t arg */
#define PCAGENT  24L    /* create agent lwp with regs from prgregset_t arg */
#define PCREAD   25L    /* read from the address space via priovec_t arg */
#define PCWRITE  26L    /* write to the address space via priovec_t arg */
#define PCSCRED  27L    /* set process credentials from prcred_t argument */
#define PCSASRS  28L    /* set ancillary state registers from asrset_t arg */
#define PCSPRIV  29L    /* set process privileges from prpriv_t argument */
#define PCSZONE  30L    /* set zoneid from zoneid_t argument */
#define PCSCREDX 31L    /* as PCSCRED but with supplemental groups */
/*
 * PCRUN long operand flags.
 */
#define PRCSIG          0x01    /* clear current signal, if any */
#define PRCFAULT        0x02    /* clear current fault, if any */
#define PRSTEP          0x04    /* direct the lwp to single-step */
#define PRSABORT        0x08    /* abort syscall, if in syscall */
#define PRSTOP          0x10    /* set directed stop request */

/*
 * lwp status file.  /proc/<pid>/lwp/<lwpid>/lwpstatus
 */
#define PRCLSZ          8       /* maximum size of scheduling class name */
#define PRSYSARGS       8       /* maximum number of syscall arguments */
typedef struct lwpstatus {
        int     pr_flags;       /* flags (see below) */
        id_t    pr_lwpid;       /* specific lwp identifier */
```

```
 127         short   pr_why;         /* reason for lwp stop, if stopped */
 128         short   pr_what;        /* more detailed reason */
 129         short   pr_cursig;      /* current signal, if any */
 130         short   pr_pad1;
 131         siginfo_t pr_info;      /* info associated with signal or fault */
 132         sigset_t pr_lwppend;    /* set of signals pending to the lwp */
 133         sigset_t pr_lwphold;    /* set of signals blocked by the lwp */
 134         struct sigaction pr_action;     /* signal action for current signal */
 135         stack_t pr_altstack;    /* alternate signal stack info */
 136         uintptr_t pr_oldcontext;        /* address of previous ucontext */
 137         short   pr_syscall;     /* system call number (if in syscall) */
 138         short   pr_nsysarg;     /* number of arguments to this syscall */
 139         int     pr_errno;       /* errno for failed syscall, 0 if successful */
 140         long    pr_sysarg[PRSYSARGS];   /* arguments to this syscall */
 141         long    pr_rval1;       /* primary syscall return value */
 142         long    pr_rval2;       /* second syscall return value, if any */
 143         char    pr_clname[PRCLSZ];      /* scheduling class name */
 144         timestruc_t pr_tstamp;  /* real-time time stamp of stop */
 145         timestruc_t pr_utime;   /* lwp user cpu time */
 146         timestruc_t pr_stime;   /* lwp system cpu time */
 147         int     pr_filler[11 - 2 * sizeof (timestruc_t) / sizeof (int)];
 148         int     pr_errpriv;     /* missing privilege */
 149         uintptr_t pr_ustack;    /* address of stack boundary data (stack_t) */
 150         ulong_t pr_instr;       /* current instruction */
 151         prgregset_t pr_reg;     /* general registers */
 152         prfpregset_t pr_fpreg;  /* floating-point registers */
 153 } lwpstatus_t;
_____unchanged_portion_omitted_

 539 /*
 540  * Representation of LWP name in core files.  In /proc, we use a simple char
 541  * array, but in core files we need to make it easy to correlate the note back
 542  * to the right LWP.  For simplicity, we'll use 32/64 consistent types.
 543  */
 544 typedef struct prlwpname {
 545         uint64_t pr_lwpid;
 546         char pr_lwpname[THREAD_NAME_MAX];
 547 } prlwpname_t;

 549 /*
 550  * Header for /proc/<pid>/lstatus /proc/<pid>/lpsinfo /proc/<pid>/lusage
 551  */
 552 typedef struct prheader {
 553         long    pr_nent;        /* number of entries */
 554         long    pr_entsize;     /* size of each entry, in bytes */
 555 } prheader_t;
_____unchanged_portion_omitted_
```

```
**********************************************************
   27191 Mon Oct 15 13:28:45 2018
new/usr/src/uts/common/sys/thread.h
8158 Want named threads API
9857 proc manpages should have LIBRARY section
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2018 Joyent, Inc.
 */

#ifndef _SYS_THREAD_H
#define _SYS_THREAD_H


#include <sys/types.h>
#include <sys/t_lock.h>
#include <sys/klwp.h>
#include <sys/time.h>
#include <sys/signal.h>
#include <sys/kcpc.h>
#if defined(__GNUC__) && defined(_ASM_INLINES) && defined(_KERNEL)
#include <asm/thread.h>
#endif

#ifdef  __cplusplus
extern "C" {
#endif

/*
 * The thread object, its states, and the methods by which it
 * is accessed.
 */

/*
 * Values that t_state may assume. Note that t_state cannot have more
 * than one of these flags set at a time.
 */
#define TS_FREE         0x00    /* Thread at loose ends */
#define TS_SLEEP        0x01    /* Awaiting an event */
#define TS_RUN          0x02    /* Runnable, but not yet on a processor */
```



```
#define TS_ONPROC       0x04    /* Thread is being run on a processor */
#define TS_ZOMB         0x08    /* Thread has died but hasn't been reaped */
#define TS_STOPPED      0x10    /* Stopped, initial state */
#define TS_WAIT         0x20    /* Waiting to become runnable */

typedef struct ctxop {
        void    (*save_op)(void *);     /* function to invoke to save context */
        void    (*restore_op)(void *);  /* function to invoke to restore ctx */
        void    (*fork_op)(void *, void *);     /* invoke to fork context */
        void    (*lwp_create_op)(void *, void *);       /* lwp_create context */
        void    (*exit_op)(void *);     /* invoked during {thread,lwp}_exit() */
        void    (*free_op)(void *, int); /* function which frees the context */
        void    *arg;           /* argument to above functions, ctx pointer */
        struct ctxop *next;     /* next context ops */
} ctxop_t;
```

```
______unchanged_portion_omitted_
```

```
typedef struct _kthread *kthread_id_t;

struct turnstile;
struct panic_trap_info;
struct upimutex;
struct kproject;
struct on_trap_data;
struct waitq;
struct _kcpc_ctx;
struct _kcpc_set;

/* Definition for kernel thread identifier type */
typedef uint64_t kt_did_t;

typedef struct _kthread {
        struct _kthread *t_link; /* dispq, sleepq, and free queue link */

        caddr_t t_stk;          /* base of stack (kernel sp value to use) */
        void    (*t_startpc)(void);     /* PC where thread started */
        struct cpu *t_bound_cpu; /* cpu bound to, or NULL if not bound */
        short   t_affinitycnt;  /* nesting level of kernel affinity-setting */
        short   t_bind_cpu;     /* user-specified CPU binding (-1 if none) */
        ushort_t t_flag;                /* modified only by current thread */
        ushort_t t_proc_flag;   /* modified holding ttproc(t)->p_lock */
        ushort_t t_schedflag;   /* modified holding thread_lock(t) */
        volatile char t_preempt;        /* don't preempt thread if set */
        volatile char t_preempt_lk;
        uint_t  t_state;        /* thread state (protected by thread_lock) */
        pri_t   t_pri;          /* assigned thread priority */
        pri_t   t_epri;         /* inherited thread priority */
        pri_t   t_cpri;         /* thread scheduling class priority */
        char    t_writer;       /* sleeping in lwp_rwlock_lock(RW_WRITE_LOCK) */
        uchar_t t_bindflag;     /* CPU and pset binding type */
        label_t t_pcb;          /* pcb, save area when switching */
        lwpchan_t t_lwpchan;    /* reason for blocking */
#define t_wchan0        t_lwpchan.lc_wchan0
#define t_wchan         t_lwpchan.lc_wchan
        struct _sobj_ops *t_sobj_ops;
        id_t    t_cid;          /* scheduling class id */
        struct thread_ops *t_clfuncs;   /* scheduling class ops vector */
        void    *t_cldata;      /* per scheduling class specific data */
        ctxop_t *t_ctx;         /* thread context */
        uintptr_t t_lofault;    /* ret pc for failed page faults */
        label_t *t_onfault;     /* on_fault() setjmp buf */
        struct on_trap_data *t_ontrap;  /* on_trap() protection data */
        caddr_t t_swap;         /* the bottom of the stack, if from segkp */
        lock_t  t_lock;         /* used to resume() a thread */
        uint8_t t_lockstat;     /* set while thread is in lockstat code */
        uint8_t t_pil;          /* interrupt thread PIL */
```

```
149         disp_lock_t     t_pi_lock;      /* lock protecting t_prioinv list */
150         char    t_nomigrate;    /* do not migrate if set */
151         struct cpu      *t_cpu; /* CPU that thread last ran on */
152         struct cpu      *t_weakbound_cpu;       /* cpu weakly bound to */
153         struct lgrp_ld  *t_lpl; /* load average for home lgroup */
154         void            *t_lgrp_reserv[2];      /* reserved for future */
155         struct _kthread *t_intr; /* interrupted (pinned) thread */
156         uint64_t        t_intr_start;   /* timestamp when time slice began */
157         kt_did_t        t_did;  /* thread id for kernel debuggers */
158         caddr_t t_tnf_tpdp;     /* Trace facility data pointer */
159         struct _kcpc_ctx *t_cpc_ctx;    /* performance counter context */
160         struct _kcpc_set *t_cpc_set;    /* set this thread has bound */

162         /*
163          * non swappable part of the lwp state.
164          */
165         id_t            t_tid;          /* lwp's id */
166         id_t            t_waitfor;      /* target lwp id in lwp_wait() */
167         struct sigqueue *t_sigqueue;    /* queue of siginfo structs */
168         k_sigset_t      t_sig;          /* signals pending to this process */
169         k_sigset_t      t_extsig;       /* signals sent from another contract */
170         k_sigset_t      t_hold;         /* hold signal bit mask */
171         k_sigset_t      t_sigwait;      /* sigtimedwait/sigfd accepting these */
172         struct  _kthread *t_forw;       /* process's forward thread link */
173         struct  _kthread *t_back;       /* process's backward thread link */
174         struct  _kthread *t_thlink;     /* tid (lwpid) lookup hash link */
175         klwp_t  *t_lwp;                 /* thread's lwp pointer */
176         struct  proc    *t_procp;       /* proc pointer */
177         struct  t_audit_data *t_audit_data;     /* per thread audit data */
178         struct  _kthread *t_next;       /* doubly linked list of all threads */
179         struct  _kthread *t_prev;
180         ushort_t t_whystop;             /* reason for stopping */
181         ushort_t t_whatstop;            /* more detailed reason */
182         int     t_dslot;                /* index in proc's thread directory */
183         struct  pollstate *t_pollstate; /* state used during poll(2) */
184         struct  pollcache *t_pollcache; /* to pass a pcache ptr by /dev/poll */
185         struct  cred    *t_cred;        /* pointer to current cred */
186         time_t  t_start;                /* start time, seconds since epoch */
187         clock_t t_lbolt;                /* lbolt at last clock_tick() */
188         hrtime_t t_stoptime;            /* timestamp at stop() */
189         uint_t  t_pctcpu;               /* %cpu at last clock_tick(), binary */
190                                         /* point at right of high-order bit */
191         short   t_sysnum;               /* system call number */
192         kcondvar_t      t_delay_cv;
193         kmutex_t        t_delay_lock;

195         /*
196          * Pointer to the dispatcher lock protecting t_state and state-related
197          * flags.  This pointer can change during waits on the lock, so
198          * it should be grabbed only by thread_lock().
199          */
200         disp_lock_t     *t_lockp;       /* pointer to the dispatcher lock */
201         ushort_t        t_oldspl;       /* spl level before dispatcher locked */
202         volatile char   t_pre_sys;      /* pre-syscall work needed */
203         lock_t          t_lock_flush;   /* for lock_mutex_flush() impl */
204         struct _disp    *t_disp_queue;  /* run queue for chosen CPU */
205         clock_t         t_disp_time;    /* last time this thread was running */
206         uint_t          t_kpri_req;     /* kernel priority required */

208         /*
209          * Post-syscall / post-trap flags.
210          *      No lock is required to set these.
211          *      These must be cleared only by the thread itself.
212          *
213          *      t_astflag indicates that some post-trap processing is required,
214          *              possibly a signal or a preemption.  The thread will not
```



```
215          *              return to user with this set.
216          *      t_post_sys indicates that some unusualy post-system call
217          *              handling is required, such as an error or tracing.
218          *      t_sig_check indicates that some condition in ISSIG() must be
219          *              checked, but doesn't prevent returning to user.
220          *      t_post_sys_ast is a way of checking whether any of these three
221          *              flags are set.
222          */
223         union __tu {
224                 struct __ts {
225                         volatile char   _t_astflag;     /* AST requested */
226                         volatile char   _t_sig_check;   /* ISSIG required */
227                         volatile char   _t_post_sys;    /* post_syscall req */
228                         volatile char   _t_trapret;     /* call CL_TRAPRET */
229                 } _ts;
230                 volatile int    _t_post_sys_ast;        /* OR of these flags */
231         } _tu;
232 #define t_astflag       _tu._ts._t_astflag
233 #define t_sig_check     _tu._ts._t_sig_check
234 #define t_post_sys      _tu._ts._t_post_sys
235 #define t_trapret       _tu._ts._t_trapret
236 #define t_post_sys_ast  _tu._t_post_sys_ast

238         /*
239          * Real time microstate profiling.
240          */
241                                         /* possible 4-byte filler */
242         hrtime_t t_waitrq;              /* timestamp for run queue wait time */
243         int     t_mstate;               /* current microstate */
244         struct rprof {
245                 int     rp_anystate;            /* set if any state non-zero */
246                 uint_t  rp_state[NMSTATES];     /* mstate profiling counts */
247         } *t_rprof;

249         /*
250          * There is a turnstile inserted into the list below for
251          * every priority inverted synchronization object that
252          * this thread holds.
253          */

255         struct turnstile *t_prioinv;

257         /*
258          * Pointer to the turnstile attached to the synchronization
259          * object where this thread is blocked.
260          */

262         struct turnstile *t_ts;

264         /*
265          * kernel thread specific data
266          *      Borrowed from userland implementation of POSIX tsd
267          */
268         struct tsd_thread {
269                 struct tsd_thread *ts_next;     /* threads with TSD */
270                 struct tsd_thread *ts_prev;     /* threads with TSD */
271                 uint_t            ts_nkeys;     /* entries in value array */
272                 void              **ts_value;   /* array of value/key */
273         } *t_tsd;

275         clock_t         t_stime;        /* time stamp used by the swapper */
276         struct door_data *t_door;       /* door invocation data */
277         kmutex_t        *t_plockp;      /* pointer to process's p_lock */

279         struct sc_shared *t_schedctl;   /* scheduler activations shared data */
280         uintptr_t       t_sc_uaddr;     /* user-level address of shared data */
```

```
282         struct cpupart  *t_cpupart;     /* partition containing thread */
283         int             t_bind_pset;    /* processor set binding */

285         struct copyops  *t_copyops;     /* copy in/out ops vector */

287         caddr_t         t_stkbase;      /* base of the the stack */
288         struct page     *t_red_pp;      /* if non-NULL, redzone is mapped */

290         afd_t           t_activefd;     /* active file descriptor table */

292         struct _kthread *t_priforw;     /* sleepq per-priority sublist */
293         struct _kthread *t_priback;

295         struct sleepq   *t_sleepq;      /* sleep queue thread is waiting on */
296         struct panic_trap_info *t_panic_trap;   /* saved data from fatal trap */
297         int             *t_lgrp_affinity;       /* lgroup affinity */
298         struct upimutex *t_upimutex;    /* list of upimutexes owned by thread */
299         uint32_t        t_nupinest;     /* number of nested held upi mutexes */
300         struct kproject *t_proj;        /* project containing this thread */
301         uint8_t         t_unpark;       /* modified holding t_delay_lock */
302         uint8_t         t_release;      /* lwp_release() waked up the thread */
303         uint8_t         t_hatdepth;     /* depth of recursive hat_memloads */
304         uint8_t         t_xpvcntr;      /* see xen_block_migrate() */
305         kcondvar_t      t_joincv;       /* cv used to wait for thread exit */
306         void            *t_taskq;       /* for threads belonging to taskq */
307         hrtime_t        t_anttime;      /* most recent time anticipatory load */
308                                         /*      was added to an lgroup's load */
309                                         /*      on this thread's behalf */
310         char            *t_pdmsg;       /* privilege debugging message */

312         uint_t          t_predcache;    /* DTrace predicate cache */
313         hrtime_t        t_dtrace_vtime; /* DTrace virtual time */
314         hrtime_t        t_dtrace_start; /* DTrace slice start time */

316         uint8_t         t_dtrace_stop;  /* indicates a DTrace-desired stop */
317         uint8_t         t_dtrace_sig;   /* signal sent via DTrace's raise() */

319         union __tdu {
320                 struct __tds {
321                         uint8_t _t_dtrace_on;   /* hit a fasttrap tracepoint */
322                         uint8_t _t_dtrace_step; /* about to return to kernel */
323                         uint8_t _t_dtrace_ret;  /* handling a return probe */
324                         uint8_t _t_dtrace_ast;  /* saved ast flag */
325 #ifdef __amd64
326                         uint8_t _t_dtrace_reg;  /* modified register */
327 #endif
328                 } _tds;
329                 ulong_t _t_dtrace_ft;           /* bitwise or of these flags */
330         } _tdu;
331 #define t_dtrace_ft     _tdu._t_dtrace_ft
332 #define t_dtrace_on     _tdu._tds._t_dtrace_on
333 #define t_dtrace_step   _tdu._tds._t_dtrace_step
334 #define t_dtrace_ret    _tdu._tds._t_dtrace_ret
335 #define t_dtrace_ast    _tdu._tds._t_dtrace_ast
336 #ifdef __amd64
337 #define t_dtrace_reg    _tdu._tds._t_dtrace_reg
338 #endif

340         uintptr_t       t_dtrace_pc;    /* DTrace saved pc from fasttrap */
341         uintptr_t       t_dtrace_npc;   /* DTrace next pc from fasttrap */
342         uintptr_t       t_dtrace_scrpc; /* DTrace per-thread scratch location */
343         uintptr_t       t_dtrace_astpc; /* DTrace return sequence location */
344 #ifdef __amd64
345         uint64_t        t_dtrace_regv;  /* DTrace saved reg from fasttrap */
346         uint64_t        t_useracc;      /* SMAP state saved across swtch() */
```



```
347 #endif
348         hrtime_t        t_hrtime;       /* high-res last time on cpu */
349         kmutex_t        t_ctx_lock;     /* protects t_ctx in removectx() */
350         struct waitq    *t_waitq;       /* wait queue */
351         kmutex_t        t_wait_mutex;   /* used in CV wait functions */

353         char            *t_name;        /* thread name */
354 } kthread_t;
_____unchanged_portion_omitted_

566 extern void     thread_free_prevent(kthread_t *);
567 extern void     thread_free_allow(kthread_t *);

569 /*
570  * Routines to change the priority and effective priority
571  * of a thread-locked thread, whatever its state.
572  */
573 extern int      thread_change_pri(kthread_t *t, pri_t disp_pri, int front);
574 extern void     thread_change_epri(kthread_t *t, pri_t disp_pri);

576 /*
577  * Routines that manipulate the dispatcher lock for the thread.
578  * The locking heirarchy is as follows:
579  *      cpu_lock > sleepq locks > run queue locks
580  */
581 void    thread_transition(kthread_t *); /* move to transition lock */
582 void    thread_stop(kthread_t *);       /* move to stop lock */
583 void    thread_lock(kthread_t *);       /* lock thread and its queue */
584 void    thread_lock_high(kthread_t *);  /* lock thread and its queue */
585 void    thread_onproc(kthread_t *, struct cpu *); /* set onproc state lock */

587 #define thread_unlock(t)                disp_lock_exit((t)->t_lockp)
588 #define thread_unlock_high(t)           disp_lock_exit_high((t)->t_lockp)
589 #define thread_unlock_nopreempt(t)      disp_lock_exit_nopreempt((t)->t_lockp)

591 #define THREAD_LOCK_HELD(t)     (DISP_LOCK_HELD((t)->t_lockp))

593 extern disp_lock_t transition_lock;     /* lock protecting transiting threads */
594 extern disp_lock_t stop_lock;           /* lock protecting stopped threads */

596 caddr_t thread_stk_init(caddr_t);       /* init thread stack */

598 int thread_setname(kthread_t *, const char *);
599 int thread_vsetname(kthread_t *, const char *, ...);

601 extern int default_binding_mode;

603 #endif  /* _KERNEL */

605 #define THREAD_NAME_MAX 32      /* includes terminating NUL */

607 /*
608  * Macros to indicate that the thread holds resources that could be critical
609  * to other kernel threads, so this thread needs to have kernel priority
610  * if it blocks or is preempted.  Note that this is not necessary if the
611  * resource is a mutex or a writer lock because of priority inheritance.
612  *
613  * The only way one thread may legally manipulate another thread's t_kpri_req
614  * is to hold the target thread's thread lock while that thread is asleep.
615  * (The rwlock code does this to implement direct handoff to waiting readers.)
616  */
617 #define THREAD_KPRI_REQUEST()   (curthread->t_kpri_req++)
618 #define THREAD_KPRI_RELEASE()   (curthread->t_kpri_req--)
619 #define THREAD_KPRI_RELEASE_N(n) (curthread->t_kpri_req -= (n))

621 /*
```

```
 622  * Macro to change a thread's priority.
 623  */
 624 #define THREAD_CHANGE_PRI(t, pri) {                                     \
 625         pri_t __new_pri = (pri);                                        \
 626         DTRACE_SCHED2(change__pri, kthread_t *, (t), pri_t, __new_pri); \
 627         (t)->t_pri = __new_pri;                                         \
 628         schedctl_set_cidpri(t);                                         \
 629 }
```

unchanged_portion_omitted_