```
**********************************************************
  115298 Mon Aug 12 18:24:51 2013
new/usr/src/cmd/init/init.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
 */

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

/*
 * University Copyright- Copyright (c) 1982, 1986, 1988
 * The Regents of the University of California
 * All Rights Reserved
 *
 * University Acknowledgment- Portions of this document are derived from
 * software developed by the University of California, Berkeley, and its
 * contributors.
 */

/*
 * init(1M) is the general process spawning program.  Its primary job is to
 * start and restart svc.startd for smf(5).  For backwards-compatibility it also
 * spawns and respawns processes according to /etc/inittab and the current
 * run-level.  It reads /etc/default/inittab for general configuration.
 *
 * To change run-levels the system administrator runs init from the command
 * line with a level name.  init signals svc.startd via libscf and directs the
 * zone's init (pid 1 in the global zone) what to do by sending it a signal;
 * these signal numbers are commonly refered to in the code as 'states'.  Valid
 * run-levels are [sS0123456].  Additionally, init can be given directives
 * [qQabc], which indicate actions to be taken pertaining to /etc/inittab.
 *
 * When init processes inittab entries, it finds processes that are to be
 * spawned at various run-levels.  inittab contains the set of the levels for
 * which each inittab entry is valid.
 *
 * State File and Restartability
 *   Premature exit by init(1M) is handled as a special case by the kernel:
 *   init(1M) will be immediately re-executed, retaining its original PID.  (PID
```

```
 *   1 in the global zone.)  To track the processes it has previously spawned,
 *   as well as other mutable state, init(1M) regularly updates a state file
 *   such that its subsequent invocations have knowledge of its various
 *   dependent processes and duties.
 *
 * Process Contracts
 *   We start svc.startd(1M) in a contract and transfer inherited contracts when
 *   restarting it.  Everything else is started using the legacy contract
 *   template, and the created contracts are abandoned when they become empty.
 *
 * utmpx Entry Handling
 *   Because init(1M) no longer governs the startup process, its knowledge of
 *   when utmpx becomes writable is indirect.  However, spawned processes
 *   expect to be constructed with valid utmpx entries.  As a result, attempts
 *   to write normal entries will be retried until successful.
 *
 * Maintenance Mode
 *   In certain failure scenarios, init(1M) will enter a maintenance mode, in
 *   which it invokes sulogin(1M) to allow the operator an opportunity to
 *   repair the system.  Normally, this operation is performed as a
 *   fork(2)-exec(2)-waitpid(3C) sequence with the parent waiting for repair or
 *   diagnosis to be completed.  In the cases that fork(2) requests themselves
 *   fail, init(1M) will directly execute sulogin(1M), and allow the kernel to
 *   restart init(1M) on exit from the operator session.
 *
 *   One scenario where init(1M) enters its maintenance mode is when
 *   svc.startd(1M) begins to fail rapidly, defined as when the average time
 *   between recent failures drops below a given threshold.
 */

#include <sys/contract/process.h>
#include <sys/ctfs.h>
#include <sys/stat.h>
#include <sys/statvfs.h>
#include <sys/stropts.h>
#include <sys/systeminfo.h>
#include <sys/time.h>
#include <sys/termios.h>
#include <sys/tty.h>
#include <sys/types.h>
#include <sys/utsname.h>

#include <bsm/adt_event.h>
#include <bsm/libbsm.h>
#include <security/pam_appl.h>

#include <assert.h>
#include <ctype.h>
#include <dirent.h>
#include <errno.h>
#include <fcntl.h>
#include <libcontract.h>
#include <libcontract_priv.h>
#include <libintl.h>
#include <libscf.h>
#include <libscf_priv.h>
#include <poll.h>
#include <procfs.h>
#include <signal.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdio_ext.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <syslog.h>
```

```c
#include <time.h>
#include <ulimit.h>
#include <unistd.h>
#include <utmpx.h>
#include <wait.h>
#include <zone.h>
#include <ucontext.h>

#undef  sleep

#define fioctl(p, sptr, cmd)    ioctl(fileno(p), sptr, cmd)
#define min(a, b)               (((a) < (b)) ? (a) : (b))

#define TRUE    1
#define FALSE   0
#define FAILURE -1

#define UT_USER_SZ      32      /* Size of a utmpx ut_user field */
#define UT_LINE_SZ      32      /* Size of a utmpx ut_line field */

/*
 * SLEEPTIME    The number of seconds "init" sleeps between wakeups if
 *              nothing else requires this "init" wakeup.
 */
#define SLEEPTIME       (5 * 60)

/*
 * MAXCMDL      The maximum length of a command string in inittab.
 */
#define MAXCMDL 512

/*
 * EXEC         The length of the prefix string added to all comamnds
 *              found in inittab.
 */
#define EXEC    (sizeof ("exec ") - 1)

/*
 * TWARN        The amount of time between warning signal, SIGTERM,
 *              and the fatal kill signal, SIGKILL.
 */
#define TWARN   5

#define id_eq(x, y)     ((x[0] == y[0] && x[1] == y[1] && x[2] == y[2] &&\
                        x[3] == y[3]) ? TRUE : FALSE)

/*
 * The kernel's default umask is 022 these days; since some processes inherit
 * their umask from init, init will set it from CMASK in /etc/default/init.
 * init gets the default umask from the kernel, it sets it to 022 whenever
 * it wants to create a file and reverts to CMASK afterwards.
 */

static int cmask;

/*
 * The following definitions, concluding with the 'lvls' array, provide a
 * common mapping between level-name (like 'S'), signal number (state),
 * run-level mask, and specific properties associated with a run-level.
 * This array should be accessed using the routines lvlname_to_state(),
 * lvlname_to_mask(), state_to_mask(), and state_to_flags().
 */

/*
 * Correspondence of signals to init actions.
 */
```

```c
#define LVLQ            SIGHUP
#define LVL0            SIGINT
#define LVL1            SIGQUIT
#define LVL2            SIGILL
#define LVL3            SIGTRAP
#define LVL4            SIGIOT
#define LVL5            SIGEMT
#define LVL6            SIGFPE
#define SINGLE_USER     SIGBUS
#define LVLa            SIGSEGV
#define LVLb            SIGSYS
#define LVLc            SIGPIPE

/*
 * Bit Mask for each level.  Used to determine legal levels.
 */
#define MASK0   0x0001
#define MASK1   0x0002
#define MASK2   0x0004
#define MASK3   0x0008
#define MASK4   0x0010
#define MASK5   0x0020
#define MASK6   0x0040
#define MASKSU  0x0080
#define MASKa   0x0100
#define MASKb   0x0200
#define MASKc   0x0400

#define MASK_NUMERIC (MASK0 | MASK1 | MASK2 | MASK3 | MASK4 | MASK5 | MASK6)
#define MASK_abc (MASKa | MASKb | MASKc)

/*
 * Flags to indicate properties of various states.
 */
#define LSEL_RUNLEVEL   0x0001  /* runlevels you can transition to */

typedef struct lvl {
        int     lvl_state;
        int     lvl_mask;
        char    lvl_name;
        int     lvl_flags;
} lvl_t;
```

______***unchanged_portion_omitted_***

```c
/*
 * getcmd() parses lines from inittab.  Each time it finds a command line
 * it will return TRUE as well as fill the passed CMD_LINE structure and
 * the shell command string.  When the end of inittab is reached, FALSE
 * is returned inittab is automatically opened if it is not currently open
 * and is closed when the end of the file is reached.
 */
static FILE *fp_inittab = NULL;

static int
getcmd(struct CMD_LINE *cmd, char *shcmd)
{
        char    *ptr;
        int     c, lastc, state;
        char    *ptr1;
        int     answer, i, proceed;
        struct  stat    sbuf;
        static char *actions[] = {
                "off", "respawn", "ondemand", "once", "wait", "boot",
                "bootwait", "powerfail", "powerwait", "initdefault",
                "sysinit",
        };
	static short act_masks[] = {
		M_OFF, M_RESPAWN, M_ONDEMAND, M_ONCE, M_WAIT, M_BOOT,
		M_BOOTWAIT, M_PF, M_PWAIT, M_INITDEFAULT, M_SYSINIT,
	};
	/*
	 * Only these actions will be allowed for entries which
	 * are specified for single-user mode.
	 */
	short su_acts = M_INITDEFAULT | M_PF | M_PWAIT | M_WAIT;

	if (fp_inittab == NULL) {
		/*
		 * Before attempting to open inittab we stat it to make
		 * sure it currently exists and is not empty.  We try
		 * several times because someone may have temporarily
		 * unlinked or truncated the file.
		 */
		for (i = 0; i < 3; i++) {
			if (stat(INITTAB, &sbuf) == -1) {
				if (i == 2) {
					console(B_TRUE,
					    "Cannot stat %s, errno: %d\n",
					    INITTAB, errno);
					return (FAILURE);
				} else {
					timer(3);
				}
			} else if (sbuf.st_size < 10) {
				if (i == 2) {
					console(B_TRUE,
					    "%s truncated or corrupted\n",
					    INITTAB);
					return (FAILURE);
				} else {
					timer(3);
				}
			} else {
				break;
			}
		}

		/*
		 * If unable to open inittab, print error message and
		 * return FAILURE to caller.
		 */
		if ((fp_inittab = fopen(INITTAB, "r")) == NULL) {
			console(B_TRUE, "Cannot open %s errno: %d\n", INITTAB,
			    errno);
			return (FAILURE);
		}
	}

	/*
	 * Keep getting commands from inittab until you find a
	 * good one or run out of file.
	 */
	for (answer = FALSE; answer == FALSE; ) {
		/*
		 * Zero out the cmd itself before trying next line.
		 */
		bzero(cmd, sizeof (struct CMD_LINE));

		/*
		 * Read in lines of inittab, parsing at colons, until a line is
		 * read in which doesn't end with a backslash.  Do not start if
		 * the first character read is an EOF.  Note that this means
```

```
		 * that lines which don't end in a newline are still processed,
		 * since the "for" will terminate normally once started,
		 * regardless of whether line terminates with a newline or EOF.
		 */
		state = FAILURE;
		if ((c = fgetc(fp_inittab)) == EOF) {
			answer = FALSE;
			(void) fclose(fp_inittab);
			fp_inittab = NULL;
			break;
		}

		for (proceed = TRUE, ptr = shcmd, state = ID, lastc = '\0';
		    proceed && c != EOF;
		    lastc = c, c = fgetc(fp_inittab)) {
			/* If we're not in the FAILURE state and haven't */
			/* yet reached the shell command field, process  */
			/* the line, otherwise just look for a real end  */
			/* of line.                                      */
			if (state != FAILURE && state != COMMAND) {
			/*
			 * Squeeze out spaces and tabs.
			 */
			if (c == ' ' || c == '\t')
				continue;

			/*
			 * Ignore characters in a comment, except for the \n.
			 */
			if (state == COMMENT) {
				if (c == '\n') {
					lastc = ' ';
					break;
				} else {
					continue;
				}
			}

			/*
			 * Detect comments (lines whose first non-whitespace
			 * character is '#') by checking that we're at the
			 * beginning of a line, have seen a '#', and haven't
			 * yet accumulated any characters.
			 */
			if (state == ID && c == '#' && ptr == shcmd) {
				state = COMMENT;
				continue;
			}

			/*
			 * If the character is a ':', then check the
			 * previous field for correctness and advance
			 * to the next field.
			 */
			if (c == ':') {
				switch (state) {

				case ID :
				/*
				 * Check to see that there are only
				 * 1 to 4 characters for the id.
				 */
				if ((i = ptr - shcmd) < 1 || i > 4) {
					state = FAILURE;
				} else {
					bcopy(shcmd, &cmd->c_id[0], i);
```

```
                                ptr = shcmd;
                                state = LEVELS;
                        }
                        break;

                        case LEVELS :
                        /*
                         * Build a mask for all the levels for
                         * which this command will be legal.
                         */
                        for (cmd->c_levels = 0, ptr1 = shcmd;
                            ptr1 < ptr; ptr1++) {
                                int mask;
                                if (lvlname_to_mask(*ptr1,
                                    &mask) == -1) {
                                        state = FAILURE;
                                        break;
                                }
                                cmd->c_levels |= mask;
                        }
                        if (state != FAILURE) {
                                state = ACTION;
                                ptr = shcmd;    /* Reset the buffer */
                        }
                        break;

                        case ACTION :
                        /*
                         * Null terminate the string in shcmd buffer and
                         * then try to match against legal actions.  If
                         * the field is of length 0, then the default of
                         * "RESPAWN" is used if the id is numeric,
                         * otherwise the default is "OFF".
                         */
                        if (ptr == shcmd) {
                                if (isdigit(cmd->c_id[0]) &&
                                    (cmd->c_id[1] == '\0' ||
                                    isdigit(cmd->c_id[1])) &&
                                    (cmd->c_id[2] == '\0' ||
                                    isdigit(cmd->c_id[2])) &&
                                    (cmd->c_id[3] == '\0' ||
                                    isdigit(cmd->c_id[3])))
                                        cmd->c_action = M_RESPAWN;
                                else
                                        cmd->c_action = M_OFF;
                        } else {
                                for (cmd->c_action = 0, i = 0,
                                    *ptr = '\0';
                                    i <
                                    sizeof (actions)/sizeof (char *);
                            for (cmd->c_action = 0, i = 0, *ptr = '\0';
                            i < sizeof (actions)/sizeof (char *);
                                    i++) {
                                if (strcmp(shcmd, actions[i]) == 0) {
                                        if ((cmd->c_levels & MASKSU) &&
                                            !(act_masks[i] & su_acts))
                                                cmd->c_action = 0;
                                        else
                                                cmd->c_action =
                                                    act_masks[i];
                                        cmd->c_action = act_masks[i];
                                        break;
                                }
                                }
                        }
```

```
                        /*
                         * If the action didn't match any legal action,
                         * set state to FAILURE.
                         */
                        if (cmd->c_action == 0) {
                                state = FAILURE;
                        } else {
                                state = COMMAND;
                                (void) strcpy(shcmd, "exec ");
                        }
                        ptr = shcmd + EXEC;
                        break;
                        }
                        continue;
                }
        }

        /* If the character is a '\n', then this is the end of a */
        /* line.  If the '\n' wasn't preceded by a backslash, */
        /* it is also the end of an inittab command.  If it was */
        /* preceded by a backslash then the next line is a */
        /* continuation.  Note that the continuation '\n' falls */
        /* through and is treated like other characters and is */
        /* stored in the shell command line. */
        if (c == '\n' && lastc != '\\') {
                proceed = FALSE;
                *ptr = '\0';
                break;
        }

        /* For all other characters just stuff them into the */
        /* command as long as there aren't too many of them. */
        /* Make sure there is room for a terminating '\0' also. */
        if (ptr >= shcmd + MAXCMDL - 1)
                state = FAILURE;
        else
                *ptr++ = (char)c;

        /* If the character we just stored was a quoted */
        /* backslash, then change "c" to '\0', so that this     */
        /* backslash will not cause a subsequent '\n' to appear */
        /* quoted.  In otherwords '\' '\' '\n' is the real end */
        /* of a command, while '\' '\n' is a continuation. */
        if (c == '\\' && lastc == '\\')
                c = '\0';
        }

        /*
         * Make sure all the fields are properly specified
         * for a good command line.
         */
        if (state == COMMAND) {
                answer = TRUE;
                cmd->c_command = shcmd;

                /*
                 * If no default level was supplied, insert
                 * all numerical levels.
                 */
                if (cmd->c_levels == 0)
                        cmd->c_levels = MASK_NUMERIC;

                /*
                 * If no action has been supplied, declare this
                 * entry to be OFF.
                 */
```

```
                        if (cmd->c_action == 0)
                                cmd->c_action = M_OFF;

                        /*
                         * If no shell command has been supplied, make sure
                         * there is a null string in the command field.
                         */
                        if (ptr == shcmd + EXEC)
                                *shcmd = ’\0’;
                } else
                        answer = FALSE;

                /*
                 * If we have reached the end of inittab, then close it
                 * and quit trying to find a good command line.
                 */
                if (c == EOF) {
                        (void) fclose(fp_inittab);
                        fp_inittab = NULL;
                        break;
                }
        }
        return (answer);
}
______unchanged_portion_omitted_

/*
 * boot_init(): Do initialization things that should be done at boot.
 */
void
boot_init()
{
        int i;
        struct PROC_TABLE *process, *oprocess;
        struct CMD_LINE cmd;
        char    line[MAXCMDL];
        char    svc_aux[SVC_AUX_SIZE];
        char    init_svc_fmri[SVC_FMRI_SIZE];
        char *old_path;
        int maxfiles;

        /* Use INIT_PATH for sysinit cmds */
        old_path = glob_envp[0];
        glob_envp[0] = malloc((unsigned)(strlen(INIT_PATH)+2));
        (void) strcpy(glob_envp[0], INIT_PATH);

        /*
         * Scan inittab(4) and process the special svc.startd entry, initdefault
         * and sysinit entries.
         */
        while (getcmd(&cmd, &line[0]) == TRUE) {
                if (startd_tmpl >= 0 && id_eq(cmd.c_id, "smf")) {
                        process_startd_line(&cmd, line);
                        (void) snprintf(startd_svc_aux, SVC_AUX_SIZE,
                            INITTAB_ENTRY_ID_STR_FORMAT, cmd.c_id);
                } else if (cmd.c_action == M_INITDEFAULT) {
                        /*
                         * initdefault is no longer meaningful, as the SMF
                         * milestone controls what (legacy) run level we
                         * boot to.
                         */
                        console(B_TRUE,
                            "Ignoring legacy \"initdefault\" entry.\n");
                } else if (cmd.c_action == M_SYSINIT) {
                        /*
                         * Execute the "sysinit" entry and wait for it to
```

```
                         * complete.  No bookkeeping is performed on these
                         * entries because we avoid writing to the file system
                         * until after there has been an chance to check it.
                         */
                        if (process = findpslot(&cmd)) {
                                (void) sighold(SIGCLD);
                                (void) snprintf(svc_aux, SVC_AUX_SIZE,
                                    INITTAB_ENTRY_ID_STR_FORMAT, cmd.c_id);
                                (void) snprintf(init_svc_fmri, SVC_FMRI_SIZE,
                                    SVC_INIT_PREFIX INITTAB_ENTRY_ID_STR_FORMAT,
                                    cmd.c_id);
                                if (legacy_tmpl >= 0) {
                                        (void) ct_pr_tmpl_set_svc_fmri(
                                            legacy_tmpl, init_svc_fmri);
                                        (void) ct_pr_tmpl_set_svc_aux(
                                            legacy_tmpl, svc_aux);
                                }

                                for (oprocess = process;
                                    (process = efork(M_OFF, oprocess,
                                    (NAMED|NOCLEANUP))) == NO_ROOM;
                                    /* CSTYLED */)
                                        ;
                                (void) sigrelse(SIGCLD);

                                if (process == NULLPROC) {
                                        maxfiles = ulimit(UL_GDESLIM, 0);

                                        for (i = 0; i < maxfiles; i++)
                                                (void) fcntl(i, F_SETFD,
                                                    FD_CLOEXEC);
                                        (void) execle(SH, "INITSH", "-c",
                                            cmd.c_command,
                                            (char *)0, glob_envp);
                                        console(B_TRUE,
"Command\n\"%s\"\n failed to execute.  errno = %d (exec of shell failed)\n",
                                            cmd.c_command, errno);
                                        exit(1);
                                } else
                                        while (waitproc(process) == FAILURE)
                                                ;
                                } else while (waitproc(process) == FAILURE);
                                process->p_flags = 0;
                                st_write();
                        }
                }
        }

        /* Restore the path. */
        free(glob_envp[0]);
        glob_envp[0] = old_path;

        /*
         * This will enable st_write() to complain about init_state_file.
         */
        booting = 0;

        /*
         * If the /etc/ioctl.syscon didn’t exist or had invalid contents write
         * out a correct version.
         */
        if (write_ioctl)
                write_ioctl_syscon();

        /*
         * Start svc.startd(1M), which does most of the work.
         */
        if (startd_cline[0] != ’\0’ && startd_tmpl >= 0) {
                /* Start svc.startd. */
                if (startd_run(startd_cline, startd_tmpl, 0) == -1)
                        cur_state = SINGLE_USER;
        } else {
                console(B_TRUE, "Absent svc.startd entry or bad "
                    "contract template.  Not starting svc.startd.\n");
                enter_maintenance();
        }
}
```
______***unchanged_portion_omitted_***

```
/*
 * prog_name() searches for the word or unix path name and
 * returns a pointer to the last element of the pathname.
 */
static char *
prog_name(char *string)
{
        char    *ptr, *ptr2;
        static char word[UT_USER_SZ + 1];
        /* XXX - utmp - fix name length */
        static char word[_POSIX_LOGIN_NAME_MAX];

        /*
         * Search for the first word skipping leading spaces and tabs.
         */
        while (*string == ’ ’ || *string == ’\t’)
                string++;

        /*
         * If the first non-space non-tab character is not one allowed in
         * a word, return a pointer to a null string, otherwise parse the
         * pathname.
         */
        if (*string != ’.’ && *string != ’/’ && *string != ’_’ &&
            (*string < ’a’ || *string > ’z’) &&
            (*string < ’A’ || * string > ’Z’) &&
            (*string < ’0’ || *string > ’9’))
                return ("");

        /*
         * Parse the pathname looking forward for ’/’, ’ ’, ’\t’, ’\n’ or
         * ’\0’.  Each time a ’/’ is found, move "ptr" to one past the
         * ’/’, thus when a ’ ’, ’\t’, ’\n’, or ’\0’ is found, "ptr" will
         * point to the last element of the pathname.
         */
        for (ptr = string; *string != ’ ’ && *string != ’\t’ &&
            *string != ’\n’ && *string != ’\0’; string++) {
                if (*string == ’/’)
                        ptr = string+1;
        }

        /*
         * Copy out up to the size of the "ut_user" array into "word",
         * null terminate it and return a pointer to it.
         */
        for (ptr2 = &word[0]; ptr2 < &word[UT_USER_SZ] &&
        /* XXX - utmp - fix name length */
        for (ptr2 = &word[0]; ptr2 < &word[_POSIX_LOGIN_NAME_MAX - 1] &&
            ptr < string; /* CSTYLED */)
                *ptr2++ = *ptr++;

        *ptr2 = ’\0’;
        return (&word[0]);
```

```
}
```
______***unchanged_portion_omitted_***

```
/*
 * /etc/inittab has more entries and we have run out of room in the proc_table
 * array. Double the size of proc_table to accomodate the extra entries.
 */
static void
increase_proc_table_size()
{
        sigset_t block, unblock;
        void *ptr;
        size_t delta = num_proc * sizeof (struct PROC_TABLE);


        /*
         * Block signals for realloc.
         */
        (void) sigfillset(&block);
        (void) sigprocmask(SIG_BLOCK, &block, &unblock);


        /*
         * On failure we just return because callers of this function check
         * for failure.
         */
        do
                ptr = realloc(g_state, g_state_sz + delta);
        while (ptr == NULL && errno == EAGAIN)
                ;
        while (ptr == NULL && errno == EAGAIN);

        if (ptr != NULL) {
                /* ensure that the new part is initialized to zero */
                bzero((caddr_t)ptr + g_state_sz, delta);

                g_state = ptr;
                g_state_sz += delta;
                num_proc <<= 1;
        }


        /* unblock our signals before returning */
        (void) sigprocmask(SIG_SETMASK, &unblock, NULL);
}
```
______***unchanged_portion_omitted_***

```
/*
 * Initialize our state.
 *
 * If the system just booted, then init_state_file, which is located on an
 * everpresent tmpfs filesystem, should not exist.
 *
 * If we were restarted, then init_state_file should exist, in
 * which case we’ll read it in, sanity check it, and use it.
 *
 * Note: You can’t call console() until proc_table is ready.
 */
void
st_init()
{
        struct stat stb;
        int ret, st_fd, insane = 0;
        size_t to_be_read;
        char *ptr;
        booting = 1;

        do {
                /*
                 * If we can exclusively create the file, then we're the
                 * initial invocation of init(1M).
                 */
                st_fd = open(init_state_file, O_RDWR | O_CREAT | O_EXCL,
                    S_IRUSR | S_IWUSR);
        } while (st_fd == -1 && errno == EINTR);
        if (st_fd != -1)
                goto new_state;

        booting = 0;

        do {
                st_fd = open(init_state_file, O_RDWR, S_IRUSR | S_IWUSR);
        } while (st_fd == -1 && errno == EINTR);
        if (st_fd == -1)
                goto new_state;

        /* Get the size of the file. */
        do
                ret = fstat(st_fd, &stb);
        while (ret == -1 && errno == EINTR)
                ;
        while (ret == -1 && errno == EINTR);
        if (ret == -1)
                goto new_state;

        do
                g_state = malloc(stb.st_size);
        while (g_state == NULL && errno == EAGAIN)
                ;
        while (g_state == NULL && errno == EAGAIN);
        if (g_state == NULL)
                goto new_state;

        to_be_read = stb.st_size;
        ptr = (char *)g_state;
        while (to_be_read > 0) {
                ssize_t read_ret;

                read_ret = read(st_fd, ptr, to_be_read);
                if (read_ret < 0) {
                        if (errno == EINTR)
                                continue;

                        goto new_state;
                }

                to_be_read -= read_ret;
                ptr += read_ret;
        }

        (void) close(st_fd);

        g_state_sz = stb.st_size;

        if (st_sane()) {
                console(B_TRUE, "Restarting.\n");
                return;
        }
```

```
                insane = 1;

new_state:
        if (st_fd >= 0)
                (void) close(st_fd);
        else
                (void) unlink(init_state_file);

        if (g_state != NULL)
                free(g_state);

        /* Something went wrong, so allocate new state. */
        g_state_sz = sizeof (struct init_state) +
            ((init_num_proc - 1) * sizeof (struct PROC_TABLE));
        do
                g_state = calloc(1, g_state_sz);
        while (g_state == NULL && errno == EAGAIN)
                ;
        while (g_state == NULL && errno == EAGAIN);
        if (g_state == NULL) {
                /* Fatal error! */
                exit(errno);
        }

        g_state->ist_runlevel = -1;
        num_proc = init_num_proc;

        if (!booting) {
                console(B_TRUE, "Restarting.\n");

                /* Overwrite the bad state file. */
                st_write();

                if (!insane) {
                        console(B_TRUE,
                            "Error accessing persistent state file '%s'.  "
                            "Ignored.\n", init_state_file);
                } else {
                        console(B_TRUE,
                            "Persistent state file '%s' is invalid and was "
                            "ignored.\n", init_state_file);
                }
        }
}
______unchanged_portion_omitted_

/*
 * Create a contract with these parameters.
 */
static int
contract_make_template(uint_t info, uint_t critical, uint_t fatal,
    uint64_t cookie)
{
        int fd, err;

        char *ioctl_tset_emsg =
            "Couldn't set \"%s\" contract template parameter: %s.\n";

        do
                fd = open64(CTFS_ROOT "/process/template", O_RDWR);
        while (fd < 0 && errno == EINTR)
                ;
        while (fd < 0 && errno == EINTR);
        if (fd < 0) {
                console(B_TRUE, "Couldn't create process template: %s.\n",
                    strerror(errno));
                return (-1);
        }

        if (err = ct_pr_tmpl_set_param(fd, CT_PR_INHERIT | CT_PR_REGENT))
                console(B_TRUE, "Contract set template inherit, regent "
                    "failed: %s.\n", strerror(err));

        /*
         * These errors result in a misconfigured template, which is better
         * than no template at all, so warn but don't abort.
         */
        if (err = ct_tmpl_set_informative(fd, info))
                console(B_TRUE, ioctl_tset_emsg, "informative", strerror(err));

        if (err = ct_tmpl_set_critical(fd, critical))
                console(B_TRUE, ioctl_tset_emsg, "critical", strerror(err));

        if (err = ct_pr_tmpl_set_fatal(fd, fatal))
                console(B_TRUE, ioctl_tset_emsg, "fatal", strerror(err));

        if (err = ct_tmpl_set_cookie(fd, cookie))
                console(B_TRUE, ioctl_tset_emsg, "cookie", strerror(err));

        (void) fcntl(fd, F_SETFD, FD_CLOEXEC);

        return (fd);
}

/*
 * Create the templates and open an event file descriptor.  We use dup2(2) to
 * get these descriptors away from the stdin/stdout/stderr group.
 */
static void
contracts_init()
{
        int err, fd;

        /*
         * Create & configure a legacy template.  We only want empty events so
         * we know when to abandon them.
         */
        legacy_tmpl = contract_make_template(0, CT_PR_EV_EMPTY, CT_PR_EV_HWERR,
            ORDINARY_COOKIE);
        if (legacy_tmpl >= 0) {
                err = ct_tmpl_activate(legacy_tmpl);
                if (err != 0) {
                        (void) close(legacy_tmpl);
                        legacy_tmpl = -1;
                        console(B_TRUE,
                            "Couldn't activate legacy template (%s); "
                            "legacy services will be in init's contract.\n",
                            strerror(err));
                }
        } else
                console(B_TRUE,
                    "Legacy services will be in init's contract.\n");

        if (dup2(legacy_tmpl, 255) == -1) {
                console(B_TRUE, "Could not duplicate legacy template: %s.\n",
                    strerror(errno));
        } else {
                (void) close(legacy_tmpl);
                legacy_tmpl = 255;
        }

        (void) fcntl(legacy_tmpl, F_SETFD, FD_CLOEXEC);
```

```
        startd_tmpl = contract_make_template(0, CT_PR_EV_EMPTY,
            CT_PR_EV_HWERR | CT_PR_EV_SIGNAL | CT_PR_EV_CORE, STARTD_COOKIE);

        if (dup2(startd_tmpl, 254) == -1) {
                console(B_TRUE, "Could not duplicate startd template: %s.\n",
                    strerror(errno));
        } else {
                (void) close(startd_tmpl);
                startd_tmpl = 254;
        }

        (void) fcntl(startd_tmpl, F_SETFD, FD_CLOEXEC);

        if (legacy_tmpl < 0 && startd_tmpl < 0) {
                /* The creation errors have already been reported. */
                console(B_TRUE,
                    "Ignoring contract events.  Core smf(5) services will not "
                    "be restarted.\n");
                return;
        }

        /*
         * Open an event endpoint.
         */
        do
                fd = open64(CTFS_ROOT "/process/pbundle", O_RDONLY);
        while (fd < 0 && errno == EINTR)
                ;
        while (fd < 0 && errno == EINTR);
        if (fd < 0) {
                console(B_TRUE,
                    "Couldn't open process pbundle: %s.  Core smf(5) services "
                    "will not be restarted.\n", strerror(errno));
                return;
        }

        if (dup2(fd, 253) == -1) {
                console(B_TRUE, "Could not duplicate process bundle: %s.\n",
                    strerror(errno));
        } else {
                (void) close(fd);
                fd = 253;
        }

        (void) fcntl(fd, F_SETFD, FD_CLOEXEC);

        /* Reset in case we've been restarted. */
        (void) ct_event_reset(fd);

        poll_fds[0].fd = fd;
        poll_fds[0].events = POLLIN;
        poll_nfds = 1;
}

static int
contract_getfile(ctid_t id, const char *name, int oflag)
{
        int fd;

        do
                fd = contract_open(id, "process", name, oflag);
        while (fd < 0 && errno == EINTR)
                ;
        while (fd < 0 && errno == EINTR);
```

```
	if (fd < 0)
		console(B_TRUE, "Couldn’t open %s for contract %ld: %s.\n",
		    name, id, strerror(errno));

	return (fd);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
   13232 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/last/last.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 2013 Gary Mills
 *
 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

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

/*
 * University Copyright- Copyright (c) 1982, 1986, 1988
 * The Regents of the University of California
 * All Rights Reserved
 *
 * University Acknowledgment- Portions of this document are derived from
 * software developed by the University of California, Berkeley, and its
 * contributors.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

/*
 * last
 */
#include <sys/types.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <strings.h>
#include <signal.h>
#include <sys/stat.h>
#include <pwd.h>
#include <fcntl.h>
#include <utmpx.h>
#include <locale.h>
#include <ctype.h>
```

```
/*
 * Use the full lengths from utmpx for NMAX, LMAX and HMAX .
 * NMAX, LMAX and HMAX are set to these values for now. They
 * should be much higher because of the max allowed limit in
 * utmpx.h
 */
#define NMAX    (sizeof (((struct utmpx *)0)->ut_user))
#define LMAX    (sizeof (((struct utmpx *)0)->ut_line))
#define NMAX    8
#define LMAX    12
#define HMAX    (sizeof (((struct utmpx *)0)->ut_host))

/* Print minimum field widths. */
#define LOGIN_WIDTH     8
#define LINE_WIDTH      12

#define SECDAY  (24*60*60)
#define CHUNK_SIZE 256

#define lineq(a, b)     (strncmp(a, b, LMAX) == 0)
#define nameq(a, b)     (strncmp(a, b, NMAX) == 0)
#define hosteq(a, b)    (strncmp(a, b, HMAX) == 0)
#define linehostnameq(a, b, c, d) \
            (lineq(a, b)&&hosteq(a+LMAX+1, c)&&nameq(a+LMAX+HMAX+2, d))

#define USAGE   "usage: last [-n number] [-f filename] [-a ] [name | tty] ...\n"

/* Beware: These are set in main() to exclude the executable name.  */
static char     **argv;
static int      argc;
static char     **names;
static int      names_num;

static struct   utmpx buf[128];

/*
 * ttnames and logouts are allocated in the blocks of
 * CHUNK_SIZE lines whenever needed. The count of the
 * current size is maintained in the variable "lines"
 * The variable bootxtime is used to hold the time of
 * the last BOOT_TIME
 * All elements of the logouts are initialised to bootxtime
 * everytime the buffer is reallocated.
 */

static char     **ttnames;
static time_t   *logouts;
static time_t   bootxtime;
static int      lines;
static char     timef[128];
static char     hostf[HMAX + 1];

static char *strspl(char *, char *);
static void onintr(int);
static void reallocate_buffer();
static void memory_alloc(int);
static int want(struct utmpx *, char **, char **);
static void record_time(time_t *, int *, int, struct utmpx *);

int
main(int ac, char **av)
{
        int i, j;
        int aflag = 0;
        int fpos;       /* current position in time format buffer */
        int chrcnt;     /* # of chars formatted by current sprintf */
        int bl, wtmp;
        char *ct;
        char *ut_host;
        char *ut_user;
        struct utmpx *bp;
        time_t otime;
        struct stat stb;
        int print = 0;
        char *crmsg = (char *)0;
        long outrec = 0;
        long maxrec = 0x7fffffffL;
        char *wtmpfile = "/var/adm/wtmpx";
        size_t hostf_len;

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

        (void) time(&buf[0].ut_xtime);
        ac--, av++;
        argc = ac;
        argv = av;
        names = malloc(argc * sizeof (char *));
        if (names == NULL) {
                perror("last");
                exit(2);
        }
        names_num = 0;
        for (i = 0; i < argc; i++) {
                if (argv[i][0] == '-') {

                        /* -[0-9]*   sets max # records to print */
                        if (isdigit(argv[i][1])) {
                                maxrec = atoi(argv[i]+1);
                                continue;
                        }

                        for (j = 1; argv[i][j] != '\0'; ++j) {
                                switch (argv[i][j]) {

                                /* -f name sets filename of wtmp file */
                                case 'f':
                                        if (argv[i][j+1] != '\0') {
                                                wtmpfile = &argv[i][j+1];
                                        } else if (i+1 < argc) {
                                                wtmpfile = argv[++i];
                                        } else {
                                                (void) fprintf(stderr,
                                                    gettext("last: argument to "
                                                    "-f is missing\n"));
                                                (void) fprintf(stderr,
                                                    gettext(USAGE));
                                                exit(1);
                                        }
                                        goto next_word;

                                /* -n number sets max # records to print */
                                case 'n': {
                                        char *arg;

                                        if (argv[i][j+1] != '\0') {
                                                arg = &argv[i][j+1];
                                        } else if (i+1 < argc) {
```

```
                                                arg = argv[++i];
                                        } else {
                                                (void) fprintf(stderr,
                                                    gettext("last: argument to "
                                                    "-n is missing\n"));
                                                (void) fprintf(stderr,
                                                    gettext(USAGE));
                                                exit(1);
                                        }

                                        if (!isdigit(*arg)) {
                                                (void) fprintf(stderr,
                                                    gettext("last: argument to "
                                                    "-n is not a number\n"));
                                                (void) fprintf(stderr,
                                                    gettext(USAGE));
                                                exit(1);
                                        }
                                        maxrec = atoi(arg);
                                        goto next_word;
                                }

                                /* -a displays hostname last on the line */
                                case 'a':
                                        aflag++;
                                        break;

                                default:
                                        (void) fprintf(stderr, gettext(USAGE));
                                        exit(1);
                                }
                        }

next_word:
                        continue;
                }

                if (strlen(argv[i]) > 2 || strcmp(argv[i], "~") == 0 ||
                    getpwnam(argv[i]) != NULL) {
                        /* Not a tty number. */
                        names[names_num] = argv[i];
                        ++names_num;
                } else {
                        /* tty number.  Prepend "tty". */
                        names[names_num] = strspl("tty", argv[i]);
                        ++names_num;
                }
        }

        wtmp = open(wtmpfile, 0);
        if (wtmp < 0) {
                perror(wtmpfile);
                exit(1);
        }
        (void) fstat(wtmp, &stb);
        bl = (stb.st_size + sizeof (buf)-1) / sizeof (buf);
        if (signal(SIGINT, SIG_IGN) != SIG_IGN) {
                (void) signal(SIGINT, onintr);
                (void) signal(SIGQUIT, onintr);
        }
        lines = CHUNK_SIZE;
        ttnames = calloc(lines, sizeof (char *));
        logouts = calloc(lines, sizeof (time_t));
        if (ttnames == NULL || logouts == NULL) {
                (void) fprintf(stderr, gettext("Out of memory \n "));
                exit(2);
        }
                for (bl--; bl >= 0; bl--) {
                (void) lseek(wtmp, (off_t)(bl * sizeof (buf)), 0);
                bp = &buf[read(wtmp, buf, sizeof (buf)) / sizeof (buf[0]) - 1];
                for (; bp >= buf; bp--) {
                        if (want(bp, &ut_host, &ut_user)) {
                                for (i = 0; i <= lines; i++) {
                                if (i == lines)
                                        reallocate_buffer();
                                if (ttnames[i] == NULL) {
                                        memory_alloc(i);
                                        /*
                                         * LMAX+HMAX+NMAX+3 bytes have been
                                         * allocated for ttnames[i].
                                         * If bp->ut_line is longer than LMAX,
                                         * ut_host is longer than HMAX,
                                         * and ut_user is longer than NMAX,
                                         * truncate it to fit ttnames[i].
                                         */
                                        (void) strlcpy(ttnames[i], bp->ut_line,
                                            LMAX+1);
                                        (void) strlcpy(ttnames[i]+LMAX+1,
                                            ut_host, HMAX+1);
                                        (void) strlcpy(ttnames[i]+LMAX+HMAX+2,
                                            ut_user, NMAX+1);
                                                record_time(&otime, &print,
                                                    i, bp);
                                                break;
                                        } else if (linehostnameq(ttnames[i],
                                            bp->ut_line, ut_host, ut_user)) {
                                                record_time(&otime,
                                                    &print, i, bp);
                                                break;
                                        }
                                }
                        }
                        if (print) {
                                if (strncmp(bp->ut_line, "ftp", 3) == 0)
                                        bp->ut_line[3] = '\0';
                                if (strncmp(bp->ut_line, "uucp", 4) == 0)
                                        bp->ut_line[4] = '\0';

                                ct = ctime(&bp->ut_xtime);
                                (void) printf(gettext("%-*.*s  %-*.*s "),
                                    LOGIN_WIDTH, NMAX, bp->ut_name,
                                    LINE_WIDTH, LMAX, bp->ut_line);
                                    NMAX, NMAX, bp->ut_name,
                                    LMAX, LMAX, bp->ut_line);
                                hostf_len = strlen(bp->ut_host);
                                (void) snprintf(hostf, sizeof (hostf),
                                    "%-*.*s", hostf_len, hostf_len,
                                    bp->ut_host);
                                fpos = snprintf(timef, sizeof (timef),
                                    "%10.10s %5.5s ",
                                    ct, 11 + ct);
                                if (!lineq(bp->ut_line, "system boot") &&
                                    !lineq(bp->ut_line, "system down")) {
                                        if (otime == 0 &&
                                            bp->ut_type == USER_PROCESS) {

        if (fpos < sizeof (timef)) {
                /* timef still has room */
                (void) snprintf(timef + fpos, sizeof (timef) - fpos,
                    gettext("  still logged in"));
        }
```

```
                                        } else {
                                        time_t delta;
                                        if (otime < 0) {
                                                otime = -otime;
                                                /*
                                                 * TRANSLATION_NOTE
                                                 * See other notes on "down"
                                                 * and "- %5.5s".
                                                 * "-" means "until".  This
                                                 * is displayed after the
                                                 * starting time as in:
                                                 *      16:20 - down
                                                 * You probably don't want to
                                                 * translate this.  Should you
                                                 * decide to translate this,
                                                 * translate "- %5.5s" too.
                                                 */

        if (fpos < sizeof (timef)) {
                /* timef still has room */
                chrcnt = snprintf(timef + fpos, sizeof (timef) - fpos,
                    gettext("- %s"), crmsg);
                fpos += chrcnt;
        }

                                        } else {

        if (fpos < sizeof (timef)) {
                /* timef still has room */
                chrcnt = snprintf(timef + fpos, sizeof (timef) - fpos,
                    gettext("- %5.5s"), ctime(&otime) + 11);
                fpos += chrcnt;
        }

                                        }
                                        delta = otime - bp->ut_xtime;
                                        if (delta < SECDAY) {

        if (fpos < sizeof (timef)) {
                /* timef still has room */
                (void) snprintf(timef + fpos, sizeof (timef) - fpos,
                    gettext("  (%5.5s)"), asctime(gmtime(&delta)) + 11);
        }

                                        } else {

        if (fpos < sizeof (timef)) {
                /* timef still has room */
                (void) snprintf(timef + fpos, sizeof (timef) - fpos,
                    gettext(" (%ld+%5.5s)"), delta / SECDAY,
                    asctime(gmtime(&delta)) + 11);
        }

                                        }
                                }
                                }
                                if (aflag)
                                        (void) printf("%-35.35s %-.*s\n",
                                            timef, strlen(hostf), hostf);
                                else
                                        (void) printf("%-16.16s %-.35s\n",
                                            hostf, timef);
                                (void) fflush(stdout);
                                if (++outrec >= maxrec)
                                        exit(0);
                        }
```

```
			/*
			 * when the system is down or crashed.
			 */
			if (bp->ut_type == BOOT_TIME) {
				for (i = 0; i < lines; i++)
					logouts[i] = -bp->ut_xtime;
				bootxtime = -bp->ut_xtime;
				/*
				 * TRANSLATION_NOTE
				 * Translation of this "down " will replace
				 * the %s in "- %s".  "down" is used instead
				 * of the real time session was ended, probably
				 * because the session ended by a sudden crash.
				 */
				crmsg = gettext("down ");
			}
			print = 0;	/* reset the print flag */
		}
	}
	ct = ctime(&buf[0].ut_xtime);
	(void) printf(gettext("\nwtmp begins %10.10s %5.5s \n"), ct, ct + 11);

	/* free() called to prevent lint warning about names */
	free(names);

	return (0);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
   21455 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/newtask/newtask.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 2013 Gary Mills
 *
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

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

#include <alloca.h>
#include <libproc.h>
#include <libintl.h>
#include <libgen.h>
#include <limits.h>
#include <project.h>
#include <pwd.h>
#include <secdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/varargs.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <priv_utils.h>

#include "utils.h"

#define OPTIONS_STRING  "Fc:lp:v"
#define NENV            8
#define ENVSIZE         255
#define PATH            "PATH=/usr/bin"
#define SUPATH          "PATH=/usr/sbin:/usr/bin"
#define SHELL           "/usr/bin/sh"
#define SHELL2          "/sbin/sh"
#define TIMEZONEFILE    "/etc/default/init"
```

```
#define LOGINFILE       "/etc/default/login"
#define GLOBAL_ERR_SZ   1024
#define GRAB_RETRY_MAX  100

static const char *pname;
extern char **environ;
static char *supath = SUPATH;
static char *path = PATH;
static char global_error[GLOBAL_ERR_SZ];
static int verbose = 0;

static priv_set_t *nset;

/* Private definitions for libproject */
extern projid_t setproject_proc(const char *, const char *, int, pid_t,
    struct ps_prochandle *, struct project *);
extern priv_set_t *setproject_initpriv(void);

static void usage(void);

static void preserve_error(const char *format, ...);

static int update_running_proc(int, char *, char *);
static int set_ids(struct ps_prochandle *, struct project *,
    struct passwd *);
static struct passwd *match_user(uid_t, char *, int);
static void setproject_err(char *, char *, int, struct project *);

static void
usage(void)
{
        (void) fprintf(stderr, gettext("usage: \n\t%s [-v] [-p project] "
            "[-c pid | [-Fl] [command [args ...]]]\n"), pname);
        exit(2);
}
______unchanged_portion_omitted_

/*
 * Given the input arguments, return the passwd structure that matches best.
 * Also, since we use getpwnam() and friends, subsequent calls to this
 * function will re-use the memory previously returned.
 */
static struct passwd *
match_user(uid_t uid, char *projname, int is_my_uid)
{
        char prbuf[PROJECT_BUFSZ], username[LOGNAME_MAX+1];
        struct project prj;
        char *tmp_name;
        struct passwd *pw = NULL;

        /*
         * In order to allow users with the same UID but distinguishable
         * user names to be in different projects we play a guessing
         * game of which username is most appropriate. If we're checking
         * for the uid of the calling process, the login name is a
         * good starting point.
         */
        if (is_my_uid) {
                if ((tmp_name = getlogin()) == NULL ||
                    (pw = getpwnam(tmp_name)) == NULL || (pw->pw_uid != uid) ||
                    (pw->pw_name == NULL))
                        pw = NULL;
        }

        /*
         * If the login name doesn't work,  we try the first match for
         * the current uid in the password file.
         */
        if (pw == NULL) {
                if (((pw = getpwuid(uid)) == NULL) || pw->pw_name == NULL) {
                        preserve_error(gettext("cannot find username "
                            "for uid %d"), uid);
                        return (NULL);
                }
        }

        /*
         * If projname wasn't supplied, we've done our best, so just return
         * what we've got now. Alternatively, if newtask's invoker has
         * superuser privileges, return the pw structure we've got now, with
         * no further checking from inproj(). Superuser should be able to
         * join any project, and the subsequent call to setproject() will
         * allow this.
         */
        if (projname == NULL || getuid() == (uid_t)0)
                return (pw);

        (void) strlcpy(username, pw->pw_name, sizeof (username));
        (void) strcpy(username, pw->pw_name);

        if (inproj(username, projname, prbuf, PROJECT_BUFSZ) == 0) {
                char **u;
                tmp_name = NULL;

                /*
                 * If the previous guesses didn't work, walk through all
                 * project members and test for UID-equivalence.
                 */

                if (getprojbyname(projname, &prj, prbuf,
                    PROJECT_BUFSZ) == NULL) {
                        preserve_error(gettext("unknown project \"%s\""),
                            projname);
                        return (NULL);
                }

                for (u = prj.pj_users; *u; u++) {
                        if ((pw = getpwnam(*u)) == NULL)
                                continue;

                        if (pw->pw_uid == uid) {
                                tmp_name = pw->pw_name;
                                break;
                        }
                }

                if (tmp_name == NULL) {
                        preserve_error(gettext("user \"%s\" is not a member of "
                            "project \"%s\""), username, projname);
                        return (NULL);
                }
        }

        return (pw);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    2641 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/inc/users.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
 */

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


#ifndef _USERS_H
#define _USERS_H


#include <pwd.h>
#include <grp.h>
#include <project.h>

#define GROUP           "/etc/group"

/* max number of projects that can be specified when adding a user */
#define NPROJECTS_MAX   1024

/* validation returns */
#define NOTUNIQUE       0       /* not unique */
#define RESERVED        1       /* reserved */
#define UNIQUE          2       /* is unique */
#define TOOBIG          3       /* number too big */
#define INVALID         4
#define LONGNAME        5       /* string too long */

/*
 * Note: constraints checking for warning (release 2.6),
 * and these may be enforced in the future releases.
 */
#define WARN_NAME_TOO_LONG      0x1
#define WARN_BAD_GROUP_NAME     0x2
#define WARN_BAD_LOGNAME_CHAR   0x4
#define WARN_BAD_LOGNAME_FIRST  0x8
#define WARN_NO_LOWERCHAR       0x10
```

```
#define WARN_BAD_PROJ_NAME      0x20
#define WARN_LOGGED_IN          0x40

/* Exit codes from passmgmt */
#define PEX_SUCCESS     0
#define PEX_NO_PERM     1
#define PEX_SYNTAX      2
#define PEX_BADARG      3
#define PEX_BADUID      4
#define PEX_HOSED_FILES 5
#define PEX_FAILED      6
#define PEX_MISSING     7
#define PEX_BUSY        8
#define PEX_BADNAME     9

#define REL_PATH(x)     (x && *x != '/')

/*
 * interfaces available from the library
 */
extern int valid_login(char *, struct passwd **, int *);
extern int valid_gname(char *, struct group **, int *);
extern int valid_group(char *, struct group **, int *);
extern int valid_project(char *, struct project *, void *buf, size_t, int *);
extern int valid_projname(char *, struct project *, void *buf, size_t, int *);
extern void warningmsg(int, char *);
extern void putgrent(struct group *, FILE *);

/* passmgmt */
#define PASSMGMT        "/usr/lib/passmgmt";
#endif  /* _USERS_H */
```

```
**********************************************************
    2682 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/lib/vlogin.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 2013 Gary Mills
 *
 * Copyright (c) 1997, by Sun Microsystems, Inc.
 * All rights reserved.
 */

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


#pragma ident   "%Z%%M% %I%     %E% SMI"        /* SVr4.0 1.3 */

/*LINTLIBRARY*/

#include        <sys/types.h>
#include        <stdio.h>
#include        <ctype.h>
#include        <userdefs.h>
#include        <users.h>
#include        <deflt.h>
#include        <limits.h>

/* Defaults file */
#define DEFAULT_USERADD "/etc/default/useradd"

/*
 * validate string given as login name.
 */
int
valid_login(char *login, struct passwd **pptr, int *warning)
{
        struct passwd *t_pptr;
        char *ptr = login;
        int bad1char, badc, clower, len;
        char c;
        char action;

        len = 0; clower = 0; badc = 0; bad1char = 0;
```

```
        *warning = 0;
        if (!login || !*login)
                return (INVALID);

        c = *ptr;
        if (!isalpha(c))
                bad1char++;
        for (; c != NULL; ptr++, c = *ptr) {
                len++;
                if (!isprint(c) || (c == ':') || (c == '\n'))
                        return (INVALID);
                if (!isalnum(c) && c != '_' && c != '-' && c != '.')
                        badc++;
                if (islower(c))
                        clower++;
        }

        action = 'w';
        if (defopen(DEFAULT_USERADD) == 0) {
                char *defptr;

                if ((defptr = defread("EXCEED_TRAD=")) != NULL) {
                        char let = tolower(*defptr);

                        switch (let) {
                        case 'w':       /* warning */
                        case 'e':       /* error */
                        case 's':       /* silent */
                                action = let;
                                break;
                        }
                }
                (void) defopen((char *)NULL);
        }

        /*
         * XXX length checking causes some operational/compatibility problem.
         * This has to be revisited in the future as ARC/standards issue.
         */
        if (len > LOGNAME_MAX)
                return (LONGNAME);

        if (len > LOGNAME_MAX_TRAD) {
                if (action == 'w')
                        *warning = *warning | WARN_NAME_TOO_LONG;
                else if (action == 'e')
                        return (LONGNAME);
        }

        if (clower == 0)
                *warning = *warning | WARN_NO_LOWERCHAR;
        if (badc != 0)
                *warning = *warning | WARN_BAD_LOGNAME_CHAR;
        if (bad1char != 0)
                *warning = *warning | WARN_BAD_LOGNAME_FIRST;

        if ((t_pptr = getpwnam(login)) != NULL) {
                if (pptr) *pptr = t_pptr;
                return (NOTUNIQUE);
        }
        return (UNIQUE);
}
______unchanged_portion_omitted_
```

```
**********************************************************
    3238 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
#
# 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 (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
#
# cmd/oamuser/user/Makefile
#

DEFAULTFILES= useradd.dfl

include ../../Makefile.cmd

GREP=           grep

USERADD=        useradd
USERDEL=        userdel
USERMOD=        usermod
ROLEADD=        roleadd
ROLEDEL=        roledel
ROLEMOD=        rolemod

SBINPROG=       $(USERADD)   $(USERDEL)   $(USERMOD)
#
#       Removing sysadm: deleted $(SYSADMPROG) from this target.
#
PROG=           $(SBINPROG)
PRODUCT=        $(PROG)

ADD_OBJ=        useradd.o     uid.o       homedir.o \
                groups.o      call_pass.o      userdefs.o   messages.o \
                val_lgrp.o    funcs.o          val_lprj.o   proj.o

DEL_OBJ=        userdel.o     call_pass.o      rmfiles.o    isbusy.o \
                groups.o      messages.o       funcs.o      proj.o

MOD_OBJ=        usermod.o     uid.o            movedir.o    groups.o \
                rmfiles.o     call_pass.o      isbusy.o     homedir.o \
                userdefs.o    messages.o       val_lgrp.o   funcs.o \
                val_lprj.o    proj.o

OBJECTS=        $(ADD_OBJ)     $(DEL_OBJ)    $(MOD_OBJ)
```

```
SRCS=           $(OBJECTS:.o=.c)

LIBDIR=         ../lib
LIBUSRGRP=      $(LIBDIR)/lib.a
LIBADM=         -ladm
LOCAL=          ../inc
HERE=           .
LINTFLAGS=      -u

ROOTSKEL=       $(ROOTETC)/skel
INSSBINPROG=    $(SBINPROG:%=$(ROOTUSRSBIN)/%)
INSSKELFILE=    $(SKELFILE:%=$(ROOTSKEL)/%)

CPPFLAGS=       -I$(HERE) -I$(LOCAL) $(CPPFLAGS.master)
CERRWARN +=     -_gcc=-Wno-implicit-function-declaration
CERRWARN +=     -_gcc=-Wno-type-limits
CERRWARN +=     -_gcc=-Wno-uninitialized
CERRWARN +=     -_gcc=-Wno-parentheses

$(INSSBINPROG)  := FILEMODE = 0555
$(INSSYSADMPROG):= FILEMODE = 0500
$(INSSKELFILE)  := FILEMODE = 0644

$(USERADD) :=   OBJS = $(ADD_OBJ)
$(USERADD) :=   LIBS = $(LIBUSRGRP)

$(USERDEL) :=   OBJS = $(DEL_OBJ)
$(USERDEL) :=   LIBS = $(LIBUSRGRP)

$(USERMOD) :=   OBJS = $(MOD_OBJ)
$(USERMOD) :=   LIBS = $(LIBUSRGRP)

LDLIBS +=       -lbsm -lnsl -lsecdb -lproject -ltsol

.PARALLEL: $(OBJECTS)

all:            $(PRODUCT)

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

$(USERADD):     $(ADD_OBJ)
$(USERMOD):     $(MOD_OBJ)
$(USERDEL):     $(DEL_OBJ)

install:        all $(ROOTETCDEFAULTFILES) .WAIT \
                        $(ROOTSKEL) $(INSSBINPROG) $(INSSKELFILE)
install:        all .WAIT $(ROOTSKEL) $(INSSBINPROG) $(INSSKELFILE)
                $(RM) $(ROOTUSRSBIN)/$(ROLEADD)
                $(LN) $(ROOTUSRSBIN)/$(USERADD) $(ROOTUSRSBIN)/$(ROLEADD)
                $(RM) $(ROOTUSRSBIN)/$(ROLEDEL)
                $(LN) $(ROOTUSRSBIN)/$(USERDEL) $(ROOTUSRSBIN)/$(ROLEDEL)
                $(RM) $(ROOTUSRSBIN)/$(ROLEMOD)
                $(LN) $(ROOTUSRSBIN)/$(USERMOD) $(ROOTUSRSBIN)/$(ROLEMOD)

clean:
        $(RM) $(OBJECTS)

lint:   lint_SRCS

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

```
**********************************************************
    4796 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/messages.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"        /* SVr4.0 1.6 */

char *errmsgs[] = {
        "WARNING: uid %ld is reserved.\n",
        "WARNING: more than NGROUPS_MAX(%d) groups specified.\n",
        "ERROR: invalid syntax.\n"
            "usage:  useradd [-u uid [-o] | -g group | -G group[[,group]...] |"
            "-d dir | -b base_dir |\n"
            "\t\t-s shell | -c comment | -m [-k skel_dir] | -f inactive |\n"
            "\t\t-e expire | -A authorization [, authorization ...] |\n"
            "\t\t-P profile [, profile ...] | -R role [, role ...] |\n"
            "\t\t-K key=value | -p project [, project ...]] login\n"
            "\tuseradd -D [-g group | -b base_dir | -f inactive | -e expire\n"
            "\t\t-A authorization [, authorization ...] |\n"
            "\t\t-P profile [, profile ...] | -R role [, role ...] |\n"
            "\t\t-K key=value ... -p project] | [-s shell] | [-k skel_dir]\n",
        "ERROR: Invalid syntax.\nusage:  userdel [-r] login\n",
        "ERROR: Invalid syntax.\n"
            "usage:  usermod -u uid [-o] | -g group | -G group[[,group]...] |\n"
            "\t\t-d dir [-m] | -s shell | -c comment |\n"
            "\t\t-l new_logname | -f inactive | -e expire |\n"
            "\t\t-A authorization [, authorization ...] | -K key=value ... |\n"
            "\t\t-P profile [, profile ...] | -R role [, role ...] login\n",
        "ERROR: Unexpected failure.  Defaults unchanged.\n",
        "ERROR: Unable to remove files from home directory.\n",
        "ERROR: Unable to remove home directory.\n",
        "ERROR: Cannot update system files - login cannot be %s.\n",
        "ERROR: uid %ld is already in use.  Choose another.\n",
        "ERROR: %s is already in use.  Choose another.\n",
```

```
        "ERROR: %s does not exist.\n",
        "ERROR: %s is not a valid %s.  Choose another.\n",
        "ERROR: %s is in use.  Cannot %s it.\n",
        "WARNING: %s has no permissions to use %s.\n",
        "ERROR: There is not sufficient space to move %s home directory to %s"
            "\n",
        "ERROR: %s %ld is too big.  Choose another.\n",
        "ERROR: group %s does not exist.  Choose another.\n",
        "ERROR: Unable to %s: %s.\n",
        "ERROR: %s is not a full path name.  Choose another.\n",
        "ERROR: %s is the primary group name.  Choose another.\n",
        "ERROR: Inconsistent password files.  See pwconv(1M).\n",
        "ERROR: %s is not a local user.\n",
        "ERROR: Permission denied.\n",
        "WARNING: Group entry exceeds 2048 char: /etc/group entry truncated.\n",
        "ERROR: invalid syntax.\n"
            "usage:  roleadd [-u uid [-o] | -g group | -G group[[,group]...] |"
            "-d dir |\n"
            "\t\t-s shell | -c comment | -m [-k skel_dir] | -f inactive |\n"
            "\t\t-e expire | -A authorization [, authorization ...] |\n"
            "\t\t-P profile [, profile ...] | -K key=value ] login\n"
            "\troleadd -D [-g group | -b base_dir | -f inactive | -e expire\n"
            "\t\t-A authorization [, authorization ...] |\n"
            "\t\t-P profile [, profile ...]]\n",
        "ERROR: Invalid syntax.\nusage:  roledel [-r] login\n",
        "ERROR: Invalid syntax.\n"
            "usage:  rolemod -u uid [-o] | -g group | -G group[[,group]...] |\n"
            "\t\t-d dir [-m] | -s shell | -c comment |\n"
            "\t\t-l new_logname | -f inactive | -e expire |\n"
            "\t\t-A authorization [, authorization ...] | -K key=value |\n"
            "\t\t-P profile [, profile ...] login\n",
        "ERROR: project %s does not exist.  Choose another.\n",
        "WARNING: more than NPROJECTS_MAX(%d) projects specified.\n",
        "WARNING: Project entry exceeds %d char: /etc/project entry truncated."
            "\n",
        "ERROR: Invalid key.\n",
        "ERROR: Missing value specification.\n",
        "ERROR: Multiple definitions of key ‘‘%s’’.\n",
        "ERROR: Roles must be modified with ‘‘rolemod’’.\n",
        "ERROR: Users must be modified with ‘‘usermod’’.\n",
        "WARNING: gid %ld is reserved.\n",
        "ERROR: Failed to read /etc/group file due to invalid entry or"
            " read error.\n",
        "ERROR: %s is too long.  Choose another.\n",
};
______unchanged_portion_omitted_
```

```
**********************************************************
    4075 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/messages.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#ifndef _MESSAGES_H
#define _MESSAGES_H

#pragma ident   "%Z%%M% %I%     %E% SMI"

extern void errmsg(int, ...);

/* WARNING: uid %d is reserved. */
#define M_RESERVED              0

/* WARNING: more than NGROUPS_MAX(%d) groups specified. */
#define M_MAXGROUPS     1

/* ERROR: invalid syntax.\nusage:  useradd ... */
#define M_AUSAGE                2

/* ERROR: Invalid syntax.\nusage:  userdel [-r] login\n" */
#define M_DUSAGE                3

/* ERROR: Invalid syntax.\nusage:  usermod ... */
#define M_MUSAGE                4


/* ERROR: Unexpected failure.  Defaults unchanged. */
#define M_FAILED        5

/* ERROR: Unable to remove files from home directory. */
#define M_RMFILES       6
```

```
/* ERROR: Unable to remove home directory. */
#define M_RMHOME                7

/* ERROR: Cannot update system files - login cannot be %s. */
#define M_UPDATE                8

/* ERROR: uid %d is already in use.  Choose another. */
#define M_UID_USED      9

/* ERROR: %s is already in use.  Choose another. */
#define M_USED  10

/* ERROR: %s does not exist. */
#define M_EXIST 11

/* ERROR: %s is not a valid %s.  Choose another. */
#define M_INVALID               12

/* ERROR: %s is in use.  Cannot %s it. */
#define M_BUSY  13

/* WARNING: %s has no permissions to use %s. */
#define M_NO_PERM       14

/* ERROR: There is not sufficient space to move %s home directory to %s */
#define M_NOSPACE               15

/* ERROR: %s %d is too big.  Choose another. */
#define M_TOOBIG        16

/* ERROR: group %s does not exist.  Choose another. */
#define M_GRP_NOTUSED   17

/* ERROR: Unable to %s: %s */
#define M_OOPS  18

/* ERROR: %s is not a full path name.  Choose another. */
#define M_RELPATH       19

/* ERROR: %s is the primary group name.  Choose another. */
#define M_SAME_GRP      20

/* ERROR: Inconsistent password files.  See pwconv(1M). */
#define M_HOSED_FILES   21

/* ERROR: %s is not a local user. */
#define M_NONLOCAL      22

/* ERROR: Permission denied. */
#define M_PERM_DENIED   23

/* WARNING: Group entry exceeds 2048 char: /etc/group entry truncated. */
#define M_GROUP_ENTRY_OVF  24

/* ERROR: invalid syntax.\nusage:  roleadd ... */
#define M_ARUSAGE               25

/* ERROR: Invalid syntax.\nusage:  roledel [-r] login\n" */
#define M_DRUSAGE               26

/* ERROR: Invalid syntax.\nusage:  rolemod -u ... */
#define M_MRUSAGE               27

/* ERROR: project %s does not exist.  Choose another. */
#define M_PROJ_NOTUSED 28
```

```
/* WARNING: more than NPROJECTS_MAX(%d) projects specified. */
#define M_MAXPROJECTS   29

/* WARNING: Project entry exceeds 512 char: /etc/project entry truncated. */
#define M_PROJ_ENTRY_OVF  30

/* ERROR: Invalid key. */
#define M_INVALID_KEY   31

/* ERROR: Missing value specification. */
#define M_INVALID_VALUE 32

/* ERROR: Multiple definitions of key ‘‘%s’’. */
#define M_REDEFINED_KEY 33

/* ERROR: Roles must be modified with rolemod */
#define M_ISROLE        34

/* ERROR: Users must be modified with usermod */
#define M_ISUSER        35

/* WARNING: gid %d is reserved. */
#define M_RESERVED_GID          36

/* ERROR: Failed to read /etc/group file due to invalid entry or read error. */
#define M_READ_ERROR    37

/* ERROR: %s is too long.  Choose another. */
#define M_TOO_LONG      38

#endif /* _MESSAGES_H */
```

```
**********************************************************
   17433 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/useradd.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

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


#include        <sys/types.h>
#include        <sys/stat.h>
#include        <sys/param.h>
#include        <stdio.h>
#include        <stdlib.h>
#include        <ctype.h>
#include        <limits.h>
#include        <string.h>
#include        <userdefs.h>
#include        <errno.h>
#include        <project.h>
#include        <unistd.h>
#include        <user_attr.h>
#include        "users.h"
#include        "messages.h"
#include        "userdisp.h"
#include        "funcs.h"

/*
 *  useradd [-u uid [-o] | -g group | -G group [[, group]...] | -d dir [-m]
 *              | -s shell | -c comment | -k skel_dir | -b base_dir] ]
 *              [ -A authorization [, authorization ...]]
 *              [ -P profile [, profile ...]]
 *              [ -K key=value ]
 *              [ -R role [, role ...]] [-p project [, project ...]] login
 *  useradd -D [ -g group ] [ -b base_dir | -f inactive | -e expire |
 *              -s shell | -k skel_dir ]
 *              [ -A authorization [, authorization ...]]
 *              [ -P profile [, profile ...]] [ -K key=value ]
```

```
 *              [ -R role [, role ...]] [-p project [, project ...]] login
 *
 *      This command adds new user logins to the system.  Arguments are:
 *
 *      uid - an integer
 *      group - an existing group's integer ID or char string name
 *      dir - home directory
 *      shell - a program to be used as a shell
 *      comment - any text string
 *      skel_dir - a skeleton directory
 *      base_dir - a directory
 *      login - a string of printable chars except colon(:)
 *      authorization - One or more comma separated authorizations defined
 *                      in auth_attr(4).
 *      profile - One or more comma separated execution profiles defined
 *              in prof_attr(4)
 *      role - One or more comma-separated role names defined in user_attr(4)
 *      project - One or more comma-separated project names or numbers
 *
 */

extern struct userdefs *getusrdef();
extern void dispusrdef();

static void cleanup();

extern uid_t findnextuid(void);
extern int check_perm(), valid_expire();
extern int putusrdef(), valid_uid();
extern int call_passmgmt(), edit_group(), create_home();
extern int edit_project();
extern int **valid_lgroup();
extern projid_t **valid_lproject();
extern void update_def(struct userdefs *);
extern void import_def(struct userdefs *);

static uid_t uid;                       /* new uid */
static char *logname;                   /* login name to add */
static struct userdefs *usrdefs;        /* defaults for useradd */

char *cmdname;

static char homedir[ PATH_MAX + 1 ];    /* home directory */
static char gidstring[32];              /* group id string representation */
static gid_t gid;                       /* gid of new login */
static char uidstring[32];              /* user id string representation */
static char *uidstr = NULL;             /* uid from command line */
static char *base_dir = NULL;           /* base_dir from command line */
static char *group = NULL;              /* group from command line */
static char *grps = NULL;               /* multi groups from command line */
static char *dir = NULL;                /* home dir from command line */
static char *shell = NULL;              /* shell from command line */
static char *comment = NULL;            /* comment from command line */
static char *skel_dir = NULL;           /* skel dir from command line */
static long inact;                      /* inactive days */
static char *inactstr = NULL;           /* inactive from command line */
static char inactstring[10];            /* inactivity string representation */
static char *expirestr = NULL;          /* expiration date from command line */
static char *projects = NULL;           /* project id's from command line */

static char *usertype = NULL;   /* type of user, either role or normal */

typedef enum {
        BASEDIR = 0,
        SKELDIR,
        SHELL
```

```c
} path_opt_t;


static void valid_input(path_opt_t, const char *);

int
main(argc, argv)
int argc;
char *argv[];
{
	int ch, ret, mflag = 0, oflag = 0, Dflag = 0, **gidlist;
	projid_t **projlist;
	char *ptr;			/* loc in a str, may be set by strtol */
	struct group *g_ptr;
	struct project p_ptr;
	char mybuf[PROJECT_BUFSZ];
	struct stat statbuf;		/* status buffer for stat */
	int warning;
	int busy = 0;
	char **nargv;			/* arguments for execvp of passmgmt */
	int argindex;			/* argument index into nargv */

	cmdname = argv[0];

	if (geteuid() != 0) {
		errmsg(M_PERM_DENIED);
		exit(EX_NO_PERM);
	}

	opterr = 0;			/* no print errors from getopt */
	usertype = getusertype(argv[0]);

	change_key(USERATTR_TYPE_KW, usertype);

	while ((ch = getopt(argc, argv,
		    "b:c:Dd:e:f:G:g:k:mop:s:u:A:P:R:K:")) != EOF)
		switch (ch) {
		case 'b':
			base_dir = optarg;
			break;

		case 'c':
			comment = optarg;
			break;

		case 'D':
			Dflag++;
			break;

		case 'd':
			dir = optarg;
			break;

		case 'e':
			expirestr = optarg;
			break;

		case 'f':
			inactstr = optarg;
			break;

		case 'G':
			grps = optarg;
			break;

		case 'g':
```

```c
			group = optarg;
			break;

		case 'k':
			skel_dir = optarg;
			break;

		case 'm':
			mflag++;
			break;

		case 'o':
			oflag++;
			break;

		case 'p':
			projects = optarg;
			break;

		case 's':
			shell = optarg;
			break;

		case 'u':
			uidstr = optarg;
			break;

		case 'A':
			change_key(USERATTR_AUTHS_KW, optarg);
			break;

		case 'P':
			change_key(USERATTR_PROFILES_KW, optarg);
			break;

		case 'R':
			if (is_role(usertype)) {
				errmsg(M_ARUSAGE);
				exit(EX_SYNTAX);
			}
			change_key(USERATTR_ROLES_KW, optarg);
			break;

		case 'K':
			change_key(NULL, optarg);
			break;

		default:
		case '?':
			if (is_role(usertype))
				errmsg(M_ARUSAGE);
			else
				errmsg(M_AUSAGE);
			exit(EX_SYNTAX);
		}

	/* get defaults for adding new users */
	usrdefs = getusrdef(usertype);

	if (Dflag) {
		/* DISPLAY mode */

		/* check syntax */
		if (optind != argc) {
			if (is_role(usertype))
				errmsg(M_ARUSAGE);
			else
				errmsg(M_AUSAGE);
			exit(EX_SYNTAX);
		}

		if (uidstr != NULL || oflag || grps != NULL ||
		    dir != NULL || mflag || comment != NULL) {
			if (is_role(usertype))
				errmsg(M_ARUSAGE);
			else
				errmsg(M_AUSAGE);
			exit(EX_SYNTAX);
		}

		/* Group must be an existing group */
		if (group != NULL) {
			switch (valid_group(group, &g_ptr, &warning)) {
			case INVALID:
				errmsg(M_INVALID, group, "group id");
				exit(EX_BADARG);
				/*NOTREACHED*/
			case TOOBIG:
				errmsg(M_TOOBIG, "gid", group);
				exit(EX_BADARG);
				/*NOTREACHED*/
			case RESERVED:
			case UNIQUE:
				errmsg(M_GRP_NOTUSED, group);
				exit(EX_NAME_NOT_EXIST);
			}
			if (warning)
				warningmsg(warning, group);

			usrdefs->defgroup = g_ptr->gr_gid;
			usrdefs->defgname = g_ptr->gr_name;

		}

		/* project must be an existing project */
		if (projects != NULL) {
			switch (valid_project(projects, &p_ptr, mybuf,
			    sizeof (mybuf), &warning)) {
			case INVALID:
				errmsg(M_INVALID, projects, "project id");
				exit(EX_BADARG);
				/*NOTREACHED*/
			case TOOBIG:
				errmsg(M_TOOBIG, "projid", projects);
				exit(EX_BADARG);
				/*NOTREACHED*/
			case UNIQUE:
				errmsg(M_PROJ_NOTUSED, projects);
				exit(EX_NAME_NOT_EXIST);
			}
			if (warning)
				warningmsg(warning, projects);

			usrdefs->defproj = p_ptr.pj_projid;
			usrdefs->defprojname = p_ptr.pj_name;
		}

		/* base_dir must be an existing directory */
		if (base_dir != NULL) {
			valid_input(BASEDIR, base_dir);
			usrdefs->defparent = base_dir;
		}
```

```
		/* inactivity period is an integer */
		if (inactstr != NULL) {
			/* convert inactstr to integer */
			inact = strtol(inactstr, &ptr, 10);
			if (*ptr || inact < 0) {
				errmsg(M_INVALID, inactstr,
				    "inactivity period");
				exit(EX_BADARG);
			}

			usrdefs->definact = inact;
		}

		/* expiration string is a date, newer than today */
		if (expirestr != NULL) {
			if (*expirestr) {
				if (valid_expire(expirestr, (time_t *)0)
				    == INVALID) {
					errmsg(M_INVALID, expirestr,
					    "expiration date");
					exit(EX_BADARG);
				}
				usrdefs->defexpire = expirestr;
			} else
				/* Unset the expiration date */
				usrdefs->defexpire = "";
		}

		if (shell != NULL) {
			valid_input(SHELL, shell);
			usrdefs->defshell = shell;
		}
		if (skel_dir != NULL) {
			valid_input(SKELDIR, skel_dir);
			usrdefs->defskel = skel_dir;
		}
		update_def(usrdefs);

		/* change defaults for useradd */
		if (putusrdef(usrdefs, usertype) < 0) {
			errmsg(M_UPDATE, "created");
			exit(EX_UPDATE);
		}

		/* Now, display */
		dispusrdef(stdout, (D_ALL & ~D_RID), usertype);
		exit(EX_SUCCESS);

	}

	/* ADD mode */

	/* check syntax */
	if (optind != argc - 1 || (skel_dir != NULL && !mflag)) {
		if (is_role(usertype))
			errmsg(M_ARUSAGE);
		else
			errmsg(M_AUSAGE);
		exit(EX_SYNTAX);
	}

	logname = argv[optind];
	switch (valid_login(logname, (struct passwd **)NULL, &warning)) {
	case INVALID:
		errmsg(M_INVALID, logname, "login name");
```

```
                exit(EX_BADARG);
                /*NOTREACHED*/

        case NOTUNIQUE:
                errmsg(M_USED, logname);
                exit(EX_NAME_EXISTS);
                /*NOTREACHED*/

        case LONGNAME:
                errmsg(M_TOO_LONG, logname);
                exit(EX_BADARG);
                /*NOTREACHED*/
        }

        if (warning)
                warningmsg(warning, logname);
        if (uidstr != NULL) {
                /* convert uidstr to integer */
                errno = 0;
                uid = (uid_t)strtol(uidstr, &ptr, (int)10);
                if (*ptr || errno == ERANGE) {
                        errmsg(M_INVALID, uidstr, "user id");
                        exit(EX_BADARG);
                }

                switch (valid_uid(uid, NULL)) {
                case NOTUNIQUE:
                        if (!oflag) {
                                /* override not specified */
                                errmsg(M_UID_USED, uid);
                                exit(EX_ID_EXISTS);
                        }
                        break;
                case RESERVED:
                        errmsg(M_RESERVED, uid);
                        break;
                case TOOBIG:
                        errmsg(M_TOOBIG, "uid", uid);
                        exit(EX_BADARG);
                        break;
                }

        } else {

                if ((uid = findnextuid()) < 0) {
                        errmsg(M_INVALID, "default id", "user id");
                        exit(EX_ID_EXISTS);
                }
        }

        if (group != NULL) {
                switch (valid_group(group, &g_ptr, &warning)) {
                case INVALID:
                        errmsg(M_INVALID, group, "group id");
                        exit(EX_BADARG);
                        /*NOTREACHED*/
                case TOOBIG:
                        errmsg(M_TOOBIG, "gid", group);
                        exit(EX_BADARG);
                        /*NOTREACHED*/
                case RESERVED:
                case UNIQUE:
                        errmsg(M_GRP_NOTUSED, group);
                        exit(EX_NAME_NOT_EXIST);
                        /*NOTREACHED*/
                }
```

```

                if (warning)
                        warningmsg(warning, group);
                gid = g_ptr->gr_gid;

        } else gid = usrdefs->defgroup;

        if (grps != NULL) {
                if (!*grps)
                        /* ignore -G "" */
                        grps = (char *)0;
                else if (!(gidlist = valid_lgroup(grps, gid)))
                        exit(EX_BADARG);
        }

        if (projects != NULL) {
                if (! *projects)
                        projects = (char *)0;
                else if (! (projlist = valid_lproject(projects)))
                        exit(EX_BADARG);
        }

        /* if base_dir is provided, check its validity; otherwise default */
        if (base_dir != NULL)
                valid_input(BASEDIR, base_dir);
        else
                base_dir = usrdefs->defparent;

        if (dir == NULL) {
                /* set homedir to home directory made from base_dir */
                (void) sprintf(homedir, "%s/%s", base_dir, logname);

        } else if (REL_PATH(dir)) {
                errmsg(M_RELPATH, dir);
                exit(EX_BADARG);

        } else
                (void) strcpy(homedir, dir);

        if (mflag) {
                /* Does home dir. already exist? */
                if (stat(homedir, &statbuf) == 0) {
                        /* directory exists - don't try to create */
                        mflag = 0;

                        if (check_perm(statbuf, uid, gid, S_IXOTH) != 0)
                                errmsg(M_NO_PERM, logname, homedir);
                }
        }
        /*
         * if shell, skel_dir are provided, check their validity.
         * Otherwise default.
         */
        if (shell != NULL)
                valid_input(SHELL, shell);
        else
                shell = usrdefs->defshell;

        if (skel_dir != NULL)
                valid_input(SKELDIR, skel_dir);
        else
                skel_dir = usrdefs->defskel;

        if (inactstr != NULL) {
                /* convert inactstr to integer */
                inact = strtol(inactstr, &ptr, 10);
```

```
		if (*ptr || inact < 0) {
			errmsg(M_INVALID, inactstr, "inactivity period");
			exit(EX_BADARG);
		}
	} else inact = usrdefs->definact;

	/* expiration string is a date, newer than today */
	if (expirestr != NULL) {
		if (*expirestr) {
			if (valid_expire(expirestr, (time_t *)0) == INVALID) {
				errmsg(M_INVALID, expirestr, "expiration date");
				exit(EX_BADARG);
			}
			usrdefs->defexpire = expirestr;
		} else
			/* Unset the expiration date */
			expirestr = (char *)0;

	} else expirestr = usrdefs->defexpire;

	import_def(usrdefs);

	/* must now call passmgmt */

	/* set up arguments to  passmgmt in nargv array */
	nargv = malloc((30 + nkeys * 2) * sizeof (char *));
	argindex = 0;
	nargv[argindex++] = PASSMGMT;
	nargv[argindex++] = "-a";	/* add */

	if (comment != NULL) {
		/* comment */
		nargv[argindex++] = "-c";
		nargv[argindex++] = comment;
	}

	/* flags for home directory */
	nargv[argindex++] = "-h";
	nargv[argindex++] = homedir;

	/* set gid flag */
	nargv[argindex++] = "-g";
	(void) sprintf(gidstring, "%u", gid);
	nargv[argindex++] = gidstring;

	/* shell */
	nargv[argindex++] = "-s";
	nargv[argindex++] = shell;

	/* set inactive */
	nargv[argindex++] = "-f";
	(void) sprintf(inactstring, "%ld", inact);
	nargv[argindex++] = inactstring;

	/* set expiration date */
	if (expirestr != NULL) {
		nargv[argindex++] = "-e";
		nargv[argindex++] = expirestr;
	}

	/* set uid flag */
	nargv[argindex++] = "-u";
	(void) sprintf(uidstring, "%u", uid);
	nargv[argindex++] = uidstring;

	if (oflag) nargv[argindex++] = "-o";
```

```
	if (nkeys > 1)
		addkey_args(nargv, &argindex);

	/* finally - login name */
	nargv[argindex++] = logname;

	/* set the last to null */
	nargv[argindex++] = NULL;

	/* now call passmgmt */
	ret = PEX_FAILED;
	/*
	 * If call_passmgmt fails for any reason other than PEX_BADUID, exit
	 * is invoked with an appropriate error message. If PEX_BADUID is
	 * returned, then if the user specified the ID, exit is invoked
	 * with an appropriate error message. Otherwise we try to pick a
	 * different ID and try again. If we run out of IDs, i.e. no more
	 * users can be created, then -1 is returned and we terminate via exit.
	 * If PEX_BUSY is returned we increment a count, since we will stop
	 * trying if PEX_BUSY reaches 3. For PEX_SUCCESS we immediately
	 * terminate the loop.
	 */
	while (busy < 3 && ret != PEX_SUCCESS) {
		switch (ret = call_passmgmt(nargv)) {
		case PEX_SUCCESS:
			break;
		case PEX_BUSY:
			busy++;
			break;
		case PEX_HOSED_FILES:
			errmsg(M_HOSED_FILES);
			exit(EX_INCONSISTENT);
			break;

		case PEX_SYNTAX:
		case PEX_BADARG:
			/* should NEVER occur that passmgmt usage is wrong */
			if (is_role(usertype))
				errmsg(M_ARUSAGE);
			else
				errmsg(M_AUSAGE);
			exit(EX_SYNTAX);
			break;

		case PEX_BADUID:
			/*
			 * The uid has been taken. If it was specified by a
			 * user, then we must fail. Otherwise, keep trying
			 * to get a good uid until we run out of IDs.
			 */
			if (uidstr != NULL) {
				errmsg(M_UID_USED, uid);
				exit(EX_ID_EXISTS);
			} else {
				if ((uid = findnextuid()) < 0) {
					errmsg(M_INVALID, "default id",
					    "user id");
					exit(EX_ID_EXISTS);
				}
				(void) sprintf(uidstring, "%u", uid);
			}
			break;

		case PEX_BADNAME:
			/* invalid loname */
```

```
                        errmsg(M_USED, logname);
                        exit(EX_NAME_EXISTS);
                        break;

                default:
                        errmsg(M_UPDATE, "created");
                        exit(ret);
                        break;
                }
        }
        if (busy == 3) {
                errmsg(M_UPDATE, "created");
                exit(ret);
        }

        /* add group entry */
        if ((grps != NULL) && edit_group(logname, (char *)0, gidlist, 0)) {
                errmsg(M_UPDATE, "created");
                cleanup(logname);
                exit(EX_UPDATE);
        }

        /* update project database */
        if ((projects != NULL) &&
            edit_project(logname, (char *)NULL, projlist, 0)) {
                errmsg(M_UPDATE, "created");
                cleanup(logname);
                exit(EX_UPDATE);
        }

        /* create home directory */
        if (mflag &&
            (create_home(homedir, skel_dir, uid, gid) != EX_SUCCESS)) {
                (void) edit_group(logname, (char *)0, (int **)0, 1);
                cleanup(logname);
                exit(EX_HOMEDIR);
        }

        return (ret);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
    1242 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/useradd.dfl
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
```

```
#
# 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

# The EXCEED_TRAD indicates the action when the traditional login name
# length limit of eight characters is exceeded.  The value "warning"
# means to issue a warning message and continue.  This is the default.
# The value "error" means to issue an error message and terminate.
# The value "silent" means to continue without issuing any message.
#
EXCEED_TRAD=warning
#EXCEED_TRAD=error
#EXCEED_TRAD=silent
```

```
**********************************************************
   15671 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/oamuser/user/usermod.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

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



#include <sys/types.h>
#include <sys/stat.h>
#include <sys/param.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <limits.h>
#include <string.h>
#include <userdefs.h>
#include <user_attr.h>
#include <nss_dbdefs.h>
#include <errno.h>
#include <project.h>
#include "users.h"
#include "messages.h"
#include "funcs.h"

/*
 *  usermod [-u uid [-o] | -g group | -G group [[,group]...] | -d dir [-m]
 *              | -s shell | -c comment | -l new_logname]
 *              | -f inactive | -e expire ]
 *              [ -A authorization [, authorization ...]]
 *              [ -P profile [, profile ...]]
 *              [ -R role [, role ...]]
 *              [ -K key=value ]
 *              [ -p project [, project]] login
 *
 *      This command adds new user logins to the system.  Arguments are:
```

```
 *
 *      uid - an integer less than MAXUID
 *      group - an existing group's integer ID or char string name
 *      dir - a directory
 *      shell - a program to be used as a shell
 *      comment - any text string
 *      skel_dir - a directory
 *      base_dir - a directory
 *      rid - an integer less than 2**16 (USHORT)
 *      login - a string of printable chars except colon (:)
 *      inactive - number of days a login maybe inactive before it is locked
 *      expire - date when a login is no longer valid
 *      authorization - One or more comma separated authorizations defined
 *                      in auth_attr(4).
 *      profile - One or more comma separated execution profiles defined
 *                in prof_attr(4)
 *      role - One or more comma-separated role names defined in user_attr(4)
 *      key=value - One or more -K options each specifying a valid user_attr(4)
 *              attribute.
 *
 */

extern int **valid_lgroup(), isbusy();
extern int valid_uid(), check_perm(), create_home(), move_dir();
extern int valid_expire(), edit_group(), call_passmgmt();
extern projid_t **valid_lproject();

static uid_t uid;               /* new uid */
static gid_t gid;                       /* gid of new login */
static char *new_logname = NULL;        /* new login name with -l option */
static char *uidstr = NULL;             /* uid from command line */
static char *group = NULL;              /* group from command line */
static char *grps = NULL;               /* multi groups from command line */
static char *dir = NULL;                /* home dir from command line */
static char *shell = NULL;              /* shell from command line */
static char *comment = NULL;            /* comment from command line */
static char *logname = NULL;            /* login name to add */
static char *inactstr = NULL;           /* inactive from command line */
static char *expire = NULL;             /* expiration date from command line */
static char *projects = NULL;           /* project ids from command line */
static char *usertype;

char *cmdname;
static char gidstring[32], uidstring[32];
char inactstring[10];

char *
strcpmalloc(str)
char *str;
{
        if (str == NULL)
                return (NULL);

        return (strdup(str));
}
______unchanged_portion_omitted_

int
main(argc, argv)
int argc;
char **argv;
{
        int ch, ret = EX_SUCCESS, call_pass = 0, oflag = 0;
        int tries, mflag = 0, inact, **gidlist, flag = 0;
        boolean_t fail_if_busy = B_FALSE;
        char *ptr;
	struct passwd *pstruct;		/* password struct for login */
	struct passwd *pw;
	struct group *g_ptr;	/* validated group from -g */
	struct stat statbuf;		/* status buffer for stat */
#ifndef att
	FILE *pwf;		/* fille ptr for opened passwd file */
#endif
	int warning;
	projid_t **projlist;
	char **nargv;			/* arguments for execvp of passmgmt */
	int argindex;			/* argument index into nargv */
	userattr_t *ua;
	char *val;
	int isrole;			/* current account is role */

	cmdname = argv[0];

	if (geteuid() != 0) {
		errmsg(M_PERM_DENIED);
		exit(EX_NO_PERM);
	}

	opterr = 0;			/* no print errors from getopt */
	/* get user type based on the program name */
	usertype = getusertype(argv[0]);

	while ((ch = getopt(argc, argv,
				"c:d:e:f:G:g:l:mop:s:u:A:P:R:K:")) != EOF)
		switch (ch) {
		case 'c':
			comment = optarg;
			flag++;
			break;
		case 'd':
			dir = optarg;
			fail_if_busy = B_TRUE;
			flag++;
			break;
		case 'e':
			expire = optarg;
			flag++;
			break;
		case 'f':
			inactstr = optarg;
			flag++;
			break;
		case 'G':
			grps = optarg;
			flag++;
			break;
		case 'g':
			group = optarg;
			fail_if_busy = B_TRUE;
			flag++;
			break;
		case 'l':
			new_logname = optarg;
			fail_if_busy = B_TRUE;
			flag++;
			break;
		case 'm':
			mflag++;
			flag++;
			fail_if_busy = B_TRUE;
			break;
		case 'o':
```

```
			oflag++;
			flag++;
			fail_if_busy = B_TRUE;
			break;
		case 'p':
			projects = optarg;
			flag++;
			break;
		case 's':
			shell = optarg;
			flag++;
			break;
		case 'u':
			uidstr = optarg;
			flag++;
			fail_if_busy = B_TRUE;
			break;
		case 'A':
			change_key(USERATTR_AUTHS_KW, optarg);
			flag++;
			break;
		case 'P':
			change_key(USERATTR_PROFILES_KW, optarg);
			flag++;
			break;
		case 'R':
			change_key(USERATTR_ROLES_KW, optarg);
			flag++;
			break;
		case 'K':
			change_key(NULL, optarg);
			flag++;
			break;
		default:
		case '?':
			if (is_role(usertype))
				errmsg(M_MRUSAGE);
			else
				errmsg(M_MUSAGE);
			exit(EX_SYNTAX);
		}

	if (optind != argc - 1 || flag == 0) {
		if (is_role(usertype))
			errmsg(M_MRUSAGE);
		else
			errmsg(M_MUSAGE);
		exit(EX_SYNTAX);
	}

	if ((!uidstr && oflag) || (mflag && !dir)) {
		if (is_role(usertype))
			errmsg(M_MRUSAGE);
		else
			errmsg(M_MUSAGE);
		exit(EX_SYNTAX);
	}

	logname = argv[optind];

	/* Determine whether the account is a role or not */
	if ((ua = getusernam(logname)) == NULL ||
	    (val = kva_match(ua->attr, USERATTR_TYPE_KW)) == NULL ||
	    strcmp(val, USERATTR_TYPE_NONADMIN_KW) != 0)
		isrole = 0;
	else
```

```
		isrole = 1;

	/* Verify that rolemod is used for roles and usermod for users */
	if (isrole != is_role(usertype)) {
		if (isrole)
			errmsg(M_ISROLE);
		else
			errmsg(M_ISUSER);
		exit(EX_SYNTAX);
	}

	/* Set the usertype key; defaults to the commandline  */
	usertype = getsetdefval(USERATTR_TYPE_KW, usertype);

	if (is_role(usertype)) {
		/* Roles can't have roles */
		if (getsetdefval(USERATTR_ROLES_KW, NULL) != NULL) {
			errmsg(M_MRUSAGE);
			exit(EX_SYNTAX);
		}
		/* If it was an ordinary user, delete its roles */
		if (!isrole)
			change_key(USERATTR_ROLES_KW, "");
	}

#ifdef att
	pw = getpwnam(logname);
#else
	/*
	 * Do this with fgetpwent to make sure we are only looking on local
	 * system (since passmgmt only works on local system).
	 */
	if ((pwf = fopen("/etc/passwd", "r")) == NULL) {
		errmsg(M_OOPS, "open", "/etc/passwd");
		exit(EX_FAILURE);
	}
	while ((pw = fgetpwent(pwf)) != NULL)
		if (strcmp(pw->pw_name, logname) == 0)
			break;

	fclose(pwf);
#endif

	if (pw == NULL) {
		char		pwdb[NSS_BUFLEN_PASSWD];
		struct passwd	pwd;

		if (getpwnam_r(logname, &pwd, pwdb, sizeof (pwdb)) == NULL) {
			/* This user does not exist. */
			errmsg(M_EXIST, logname);
			exit(EX_NAME_NOT_EXIST);
		} else {
			/* This user exists in non-local name service. */
			errmsg(M_NONLOCAL, logname);
			exit(EX_NOT_LOCAL);
		}
	}

	pstruct = passwd_cpmalloc(pw);

	/*
	 * We can't modify a logged in user if any of the following
	 * are being changed:
	 * uid (-u & -o), group (-g), home dir (-m), loginname (-l).
	 * If none of those are specified it is okay to go ahead
	 * some types of changes only take effect on next login, some
```

```
	 * like authorisations and profiles take effect instantly.
	 * One might think that -K type=role should require that the
	 * user not be logged in, however this would make it very
	 * difficult to make the root account a role using this command.
	 */
	if (isbusy(logname)) {
		if (fail_if_busy) {
			errmsg(M_BUSY, logname, "change");
			exit(EX_BUSY);
		}
		warningmsg(WARN_LOGGED_IN, logname);
	}

	if (new_logname && strcmp(new_logname, logname)) {
		switch (valid_login(new_logname, (struct passwd **)NULL,
			&warning)) {
		case INVALID:
			errmsg(M_INVALID, new_logname, "login name");
			exit(EX_BADARG);
			/*NOTREACHED*/

		case NOTUNIQUE:
			errmsg(M_USED, new_logname);
			exit(EX_NAME_EXISTS);
			/*NOTREACHED*/

		case LONGNAME:
			errmsg(M_TOO_LONG, new_logname);
			exit(EX_BADARG);
			/*NOTREACHED*/

		default:
			call_pass = 1;
			break;
		}
		if (warning)
			warningmsg(warning, logname);
	}

	if (uidstr) {
		/* convert uidstr to integer */
		errno = 0;
		uid = (uid_t)strtol(uidstr, &ptr, (int)10);
		if (*ptr || errno == ERANGE) {
			errmsg(M_INVALID, uidstr, "user id");
			exit(EX_BADARG);
		}

		if (uid != pstruct->pw_uid) {
			switch (valid_uid(uid, NULL)) {
			case NOTUNIQUE:
				if (!oflag) {
					/* override not specified */
					errmsg(M_UID_USED, uid);
					exit(EX_ID_EXISTS);
				}
				break;
			case RESERVED:
				errmsg(M_RESERVED, uid);
				break;
			case TOOBIG:
				errmsg(M_TOOBIG, "uid", uid);
				exit(EX_BADARG);
				break;
			}
```

```c
                        call_pass = 1;

                } else {
                        /* uid's the same, so don't change anything */
                        uidstr = NULL;
                        oflag = 0;
                }

        } else uid = pstruct->pw_uid;

        if (group) {
                switch (valid_group(group, &g_ptr, &warning)) {
                case INVALID:
                        errmsg(M_INVALID, group, "group id");
                        exit(EX_BADARG);
                        /*NOTREACHED*/
                case TOOBIG:
                        errmsg(M_TOOBIG, "gid", group);
                        exit(EX_BADARG);
                        /*NOTREACHED*/
                case UNIQUE:
                        errmsg(M_GRP_NOTUSED, group);
                        exit(EX_NAME_NOT_EXIST);
                        /*NOTREACHED*/
                case RESERVED:
                        gid = (gid_t)strtol(group, &ptr, (int)10);
                        errmsg(M_RESERVED_GID, gid);
                        break;
                }
                if (warning)
                        warningmsg(warning, group);

                if (g_ptr != NULL)
                        gid = g_ptr->gr_gid;
                else
                        gid = pstruct->pw_gid;

                /* call passmgmt if gid is different, else ignore group */
                if (gid != pstruct->pw_gid)
                        call_pass = 1;
                else group = NULL;

        } else gid = pstruct->pw_gid;

        if (grps && *grps) {
                if (!(gidlist = valid_lgroup(grps, gid)))
                        exit(EX_BADARG);
        } else
                gidlist = (int **)0;

        if (projects && *projects) {
                if (! (projlist = valid_lproject(projects)))
                        exit(EX_BADARG);
        } else
                projlist = (projid_t **)0;

        if (dir) {
                if (REL_PATH(dir)) {
                        errmsg(M_RELPATH, dir);
                        exit(EX_BADARG);
                }
                if (strcmp(pstruct->pw_dir, dir) == 0) {
                        /* home directory is the same so ignore dflag & mflag */
                        dir = NULL;
                        mflag = 0;
                } else call_pass = 1;
```

```c
        }

        if (mflag) {
                if (stat(dir, &statbuf) == 0) {
                        /* Home directory exists */
                        if (check_perm(statbuf, pstruct->pw_uid,
                            pstruct->pw_gid, S_IWOTH|S_IXOTH) != 0) {
                                errmsg(M_NO_PERM, logname, dir);
                                exit(EX_NO_PERM);
                        }

                } else ret = create_home(dir, NULL, uid, gid);

                if (ret == EX_SUCCESS)
                        ret = move_dir(pstruct->pw_dir, dir, logname);

                if (ret != EX_SUCCESS)
                        exit(ret);
        }

        if (shell) {
                if (REL_PATH(shell)) {
                        errmsg(M_RELPATH, shell);
                        exit(EX_BADARG);
                }
                if (strcmp(pstruct->pw_shell, shell) == 0) {
                        /* ignore s option if shell is not different */
                        shell = NULL;
                } else {
                        if (stat(shell, &statbuf) < 0 ||
                            (statbuf.st_mode & S_IFMT) != S_IFREG ||
                            (statbuf.st_mode & 0555) != 0555) {

                                errmsg(M_INVALID, shell, "shell");
                                exit(EX_BADARG);
                        }

                        call_pass = 1;
                }
        }

        if (comment)
                /* ignore comment if comment is not changed */
                if (strcmp(pstruct->pw_comment, comment))
                        call_pass = 1;
                else
                        comment = NULL;

        /* inactive string is a positive integer */
        if (inactstr) {
                /* convert inactstr to integer */
                inact = (int)strtol(inactstr, &ptr, 10);
                if (*ptr || inact < 0) {
                        errmsg(M_INVALID, inactstr, "inactivity period");
                        exit(EX_BADARG);
                }
                call_pass = 1;
        }

        /* expiration string is a date, newer than today */
        if (expire) {
                if (*expire &&
                    valid_expire(expire, (time_t *)0) == INVALID) {
                        errmsg(M_INVALID, expire, "expiration date");
                        exit(EX_BADARG);
                }
                call_pass = 1;
        }

        if (nkeys > 0)
                call_pass = 1;

        /* that's it for validations - now do the work */

        if (grps) {
                /* redefine login's supplentary group memberships */
                ret = edit_group(logname, new_logname, gidlist, 1);
                if (ret != EX_SUCCESS) {
                        errmsg(M_UPDATE, "modified");
                        exit(ret);
                }
        }
        if (projects) {
                ret = edit_project(logname, (char *)NULL, projlist, 0);
                if (ret != EX_SUCCESS) {
                        errmsg(M_UPDATE, "modified");
                        exit(ret);
                }
        }


        if (!call_pass) exit(ret);

        /* only get to here if need to call passmgmt */
        /* set up arguments to  passmgmt in nargv array */
        nargv = malloc((30 + nkeys * 2) * sizeof (char *));

        argindex = 0;
        nargv[argindex++] = PASSMGMT;
        nargv[argindex++] = "-m";       /* modify */

        if (comment) {  /* comment */
                nargv[argindex++] = "-c";
                nargv[argindex++] = comment;
        }

        if (dir) {
                /* flags for home directory */
                nargv[argindex++] = "-h";
                nargv[argindex++] = dir;
        }

        if (group) {
                /* set gid flag */
                nargv[argindex++] = "-g";
                (void) sprintf(gidstring, "%u", gid);
                nargv[argindex++] = gidstring;
        }

        if (shell) {    /* shell */
                nargv[argindex++] = "-s";
                nargv[argindex++] = shell;
        }

        if (inactstr) {
                nargv[argindex++] = "-f";
                nargv[argindex++] = inactstr;
        }

        if (expire) {
                nargv[argindex++] = "-e";
                nargv[argindex++] = expire;
```

```
        }

        if (uidstr) {   /* set uid flag */
                nargv[argindex++] = "-u";
                (void) sprintf(uidstring, "%u", uid);
                nargv[argindex++] = uidstring;
        }

        if (oflag) nargv[argindex++] = "-o";

        if (new_logname) {      /* redefine login name */
                nargv[argindex++] = "-l";
                nargv[argindex++] = new_logname;
        }

        if (nkeys > 0)
                addkey_args(nargv, &argindex);

        /* finally - login name */
        nargv[argindex++] = logname;

        /* set the last to null */
        nargv[argindex++] = NULL;

        /* now call passmgmt */
        ret = PEX_FAILED;
        for (tries = 3; ret != PEX_SUCCESS && tries--; ) {
                switch (ret = call_passmgmt(nargv)) {
                case PEX_SUCCESS:
                case PEX_BUSY:
                        break;

                case PEX_HOSED_FILES:
                        errmsg(M_HOSED_FILES);
                        exit(EX_INCONSISTENT);
                        break;

                case PEX_SYNTAX:
                case PEX_BADARG:
                        /* should NEVER occur that passmgmt usage is wrong */
                        if (is_role(usertype))
                                errmsg(M_MRUSAGE);
                        else
                                errmsg(M_MUSAGE);
                        exit(EX_SYNTAX);
                        break;

                case PEX_BADUID:
                        /* uid in use - shouldn't happen print message anyway */
                        errmsg(M_UID_USED, uid);
                        exit(EX_ID_EXISTS);
                        break;

                case PEX_BADNAME:
                        /* invalid loname */
                        errmsg(M_USED, logname);
                        exit(EX_NAME_EXISTS);
                        break;

                default:
                        errmsg(M_UPDATE, "modified");
                        exit(ret);
                        break;
                }
        }
        if (tries == 0) {
                errmsg(M_UPDATE, "modified");
        }

        exit(ret);
        /*NOTREACHED*/
}
```
______***unchanged_portion_omitted_***

****
44951 Mon Aug 12 18:24:52 2013
new/usr/src/cmd/prstat/prstat.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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
 */

#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/LWPID      "
#define PSINFO_HEADER_LWP_LGRP \
"   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/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%% %-.16s/%d"
#define PSINFO_LINE_LGRP \
"%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 %-.12s/%d"
"%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 %-.12s/%d"
#define USER_LINE \
"%6d %-8s %5.5s %5.5s   %3.3s%% %9s %3.3s%%"
#define TASK_LINE \
"%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"
#define PROJECT_LINE \
"%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"
```

```
#define ZONE_LINE \
"%6d %8d %5s %5s   %3.3s%% %9s %3.3s%% %28s"

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

/* global variables */

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

static table_t  pid_tbl = {0, 0, NULL};         /* selected processes */
static table_t  cpu_tbl = {0, 0, NULL};         /* selected processors */
static table_t  set_tbl = {0, 0, NULL};         /* selected processor sets */
static table_t  prj_tbl = {0, 0, NULL};         /* selected projects */
static table_t  tsk_tbl = {0, 0, NULL};         /* selected tasks */
static table_t  lgr_tbl = {0, 0, NULL};         /* selected lgroups */
static zonetbl_t zone_tbl = {0, 0, NULL};       /* selected zones */
static uidtbl_t euid_tbl = {0, 0, NULL};        /* selected effective users */
static uidtbl_t ruid_tbl = {0, 0, NULL};        /* selected real users */

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

static list_t   lwps;                           /* list of lwps/processes */
static list_t   users;                          /* list of users */
static list_t   tasks;                          /* list of tasks */
static list_t   projects;                       /* list of projects */
static list_t   zones;                          /* list of zones */
static list_t   lgroups;                        /* list of lgroups */

static volatile uint_t sigwinch = 0;
static volatile uint_t sigtstp = 0;
static volatile uint_t sigterm = 0;

static long pagesize;

/* default settings */

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

______***unchanged_portion_omitted_***

```
/*
 * A routine to display the contents of the list on the screen
 */
static void
list_print(list_t *list)
{
```

```
        lwp_info_t *lwp;
        id_info_t *id;
        char usr[4], sys[4], trp[4], tfl[4];
        char dfl[4], lck[4], slp[4], lat[4];
        char vcx[4], icx[4], scl[4], sig[4];
        char psize[6], prssize[6], pmem[6], pcpu[6], ptime[12];
        char pstate[7], pnice[4], ppri[4];
        char pname[LOGNAME_MAX+1];
        char projname[PROJNAME_MAX+1];
        char zonename[ZONENAME_MAX+1];
        float cpu, mem;
        double loadavg[3] = {0, 0, 0};
        int i, lwpid;

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

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

        switch (list->l_type) {
        case LT_PROJECTS:
                if (opts.o_outpmode & OPT_LWPS)
                        (void) printf(PROJECT_HEADER_LWP);
                else
                        (void) printf(PROJECT_HEADER_PROC);
                break;
        case LT_TASKS:
                if (opts.o_outpmode & OPT_LWPS)
                        (void) printf(TASK_HEADER_LWP);
                else
                        (void) printf(TASK_HEADER_PROC);
                break;
        case LT_ZONES:
                if (opts.o_outpmode & OPT_LWPS)
                        (void) printf(ZONE_HEADER_LWP);
                else
                        (void) printf(ZONE_HEADER_PROC);
                break;
        case LT_USERS:
                if (opts.o_outpmode & OPT_LWPS)
                        (void) printf(USER_HEADER_LWP);
                else
                        (void) printf(USER_HEADER_PROC);
                break;
        case LT_LWPS:
                if (opts.o_outpmode & OPT_LWPS) {
                        if (opts.o_outpmode & OPT_PSINFO) {
                                if (opts.o_outpmode & OPT_LGRP)
                                        (void) printf(PSINFO_HEADER_LWP_LGRP);
                                else
                                        (void) printf(PSINFO_HEADER_LWP);
                        }
                        if (opts.o_outpmode & OPT_MSACCT)
                                (void) printf(USAGE_HEADER_LWP);
                } else {
                        if (opts.o_outpmode & OPT_PSINFO) {
                                if (opts.o_outpmode & OPT_LGRP)
                                        (void) printf(PSINFO_HEADER_PROC_LGRP);
                                else
                                        (void) printf(PSINFO_HEADER_PROC);
                        }
                        if (opts.o_outpmode & OPT_MSACCT)
                                (void) printf(USAGE_HEADER_PROC);
                }
                break;
        }

        (void) putp(t_uloff);
        (void) putp(t_eol);
        (void) putchar('\n');

        for (i = 0; i < list->l_used; i++) {
                switch (list->l_type) {
                case LT_PROJECTS:
                case LT_TASKS:
                case LT_USERS:
                case LT_ZONES:
                        id = list->l_ptrs[i];
                        /*
                         * CPU usage and memory usage normalization
                         */
                        if (total_cpu >= 100)
                                cpu = (100 * id->id_pctcpu) / total_cpu;
                        else
                                cpu = id->id_pctcpu;
                        if (id->id_sizematch == B_FALSE && total_mem >= 100)
                                mem = (100 * id->id_pctmem) / total_mem;
                        else
                                mem = id->id_pctmem;
                        if (list->l_type == LT_USERS) {
                                pwd_getname(id->id_uid, pname, sizeof (pname),
                                    opts.o_outpmode & OPT_NORESOLVE,
                                    opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
                                    LOGIN_WIDTH);
                        } else if (list->l_type == LT_ZONES) {
                        if (list->l_type == LT_USERS)
                                pwd_getname(id->id_uid, pname, LOGNAME_MAX + 1,
                                    opts.o_outpmode & OPT_NORESOLVE);
                        else if (list->l_type == LT_ZONES)
                                getzonename(id->id_zoneid, zonename,
                                    sizeof (zonename),
                                    opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
                                    ZONE_WIDTH);
                        } else {
                                    ZONENAME_MAX);
                        else
                                getprojname(id->id_projid, projname,
                                    sizeof (projname),
                                    opts.o_outpmode & OPT_NORESOLVE,
                                    opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
                                    PROJECT_WIDTH);
                        }
                                    PROJNAME_MAX,
                                    opts.o_outpmode & OPT_NORESOLVE);
                        Format_size(psize, id->id_size, 6);
                        Format_size(prssize, id->id_rssize, 6);
                        Format_pct(pmem, mem, 4);
                        Format_pct(pcpu, cpu, 4);
                        Format_time(ptime, id->id_time, 10);
                        if (opts.o_outpmode & OPT_TTY)
                                (void) putchar('\r');
                        if (list->l_type == LT_PROJECTS)
```

```
                                (void) printf(PROJECT_LINE, (int)id->id_projid,
                                    id->id_nproc, psize, prssize, pmem, ptime,
                                    pcpu, projname);
                        else if (list->l_type == LT_TASKS)
                                (void) printf(TASK_LINE, (int)id->id_taskid,
                                    id->id_nproc, psize, prssize, pmem, ptime,
                                    pcpu, projname);
                        else if (list->l_type == LT_ZONES)
                                (void) printf(ZONE_LINE, (int)id->id_zoneid,
                                    id->id_nproc, psize, prssize, pmem, ptime,
                                    pcpu, zonename);
                        else
                                (void) printf(USER_LINE, id->id_nproc, pname,
                                    psize, prssize, pmem, ptime, pcpu);
                        (void) putp(t_eol);
                        (void) putchar('\n');
                        break;
                case LT_LWPS:
                        lwp = list->l_ptrs[i];
                        if (opts.o_outpmode & OPT_LWPS)
                                lwpid = lwp->li_info.pr_lwp.pr_lwpid;
                        else
                                lwpid = lwp->li_info.pr_nlwp +
                                    lwp->li_info.pr_nzomb;
                        pwd_getname(lwp->li_info.pr_uid, pname, sizeof (pname),
                            opts.o_outpmode & OPT_NORESOLVE,
                            opts.o_outpmode & (OPT_TERMCAP|OPT_TRUNC),
                            LOGIN_WIDTH);
                        pwd_getname(lwp->li_info.pr_uid, pname, LOGNAME_MAX + 1,
                            opts.o_outpmode & OPT_NORESOLVE);
                        if (opts.o_outpmode & OPT_PSINFO) {
                                Format_size(psize, lwp->li_info.pr_size, 6);
                                Format_size(prssize, lwp->li_info.pr_rssize, 6);
                                Format_state(pstate,
                                    lwp->li_info.pr_lwp.pr_sname,
                                    lwp->li_info.pr_lwp.pr_onpro, 7);
                                if (strcmp(lwp->li_info.pr_lwp.pr_clname,
                                    "RT") == 0 ||
                                    strcmp(lwp->li_info.pr_lwp.pr_clname,
                                    "SYS") == 0 ||
                                    lwp->li_info.pr_lwp.pr_sname == 'Z')
                                        (void) strcpy(pnice, "  -");
                                else
                                        Format_num(pnice,
                                            lwp->li_info.pr_lwp.pr_nice - NZERO,
                                            4);
                                Format_num(ppri, lwp->li_info.pr_lwp.pr_pri, 4);
                                Format_pct(pcpu,
                                    FRC2PCT(lwp->li_info.pr_lwp.pr_pctcpu), 4);
                                if (opts.o_outpmode & OPT_LWPS)
                                        Format_time(ptime,
                                            lwp->li_info.pr_lwp.pr_time.tv_sec,
                                            10);
                                else
                                        Format_time(ptime,
                                            lwp->li_info.pr_time.tv_sec, 10);
                                if (opts.o_outpmode & OPT_TTY)
                                        (void) putchar('\r');
                                stripfname(lwp->li_info.pr_fname);
                                if (opts.o_outpmode & OPT_LGRP) {
                                        (void) printf(PSINFO_LINE_LGRP,
                                            (int)lwp->li_info.pr_pid, pname,
                                            psize, prssize, pstate,
                                            ppri, pnice, ptime, pcpu,
                                            psize, prssize, pstate, ppri, pnice,
                                            ptime, pcpu,
                                            (int)lwp->li_info.pr_lwp.pr_lgrp,
                                            lwp->li_info.pr_fname, lwpid);
                                } else {
                                        (void) printf(PSINFO_LINE,
                                            (int)lwp->li_info.pr_pid, pname,
                                            psize, prssize,
                                            pstate, ppri, pnice,
                                            psize, prssize, pstate, ppri, pnice,
                                            ptime, pcpu,
                                            lwp->li_info.pr_fname, lwpid);
                                }
                                (void) putp(t_eol);
                                (void) putchar('\n');
                        }
                        if (opts.o_outpmode & OPT_MSACCT) {
                                Format_pct(usr, lwp->li_usr, 4);
                                Format_pct(sys, lwp->li_sys, 4);
                                Format_pct(slp, lwp->li_slp, 4);
                                Format_num(vcx, lwp->li_vcx, 4);
                                Format_num(icx, lwp->li_icx, 4);
                                Format_num(scl, lwp->li_scl, 4);
                                Format_num(sig, lwp->li_sig, 4);
                                Format_pct(trp, lwp->li_trp, 4);
                                Format_pct(tfl, lwp->li_tfl, 4);
                                Format_pct(dfl, lwp->li_dfl, 4);
                                Format_pct(lck, lwp->li_lck, 4);
                                Format_pct(lat, lwp->li_lat, 4);
                                if (opts.o_outpmode & OPT_TTY)
                                        (void) putchar('\r');
                                stripfname(lwp->li_info.pr_fname);
                                (void) printf(USAGE_LINE,
                                    (int)lwp->li_info.pr_pid, pname,
                                    usr, sys, trp, tfl, dfl, lck,
                                    slp, lat, vcx, icx, scl, sig,
                                    lwp->li_info.pr_fname, lwpid);
                                (void) putp(t_eol);
                                (void) putchar('\n');
                        }
                        break;
                }
        }

        if (opts.o_outpmode & OPT_TTY)
                (void) putchar('\r');
        if (opts.o_outpmode & OPT_TERMCAP) {
                switch (list->l_type) {
                case LT_PROJECTS:
                case LT_USERS:
                case LT_TASKS:
                case LT_ZONES:
                        while (i++ < opts.o_nbottom) {
                                (void) putp(t_eol);
                                (void) putchar('\n');
                        }
                        break;
                case LT_LWPS:
                        while (i++ < opts.o_ntop) {
                                (void) putp(t_eol);
                                (void) putchar('\n');
                        }
                }
        }

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

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

        (void) printf(TOTAL_LINE, total_procs, total_lwps,
            loadavg[LOADAVG_1MIN], loadavg[LOADAVG_5MIN],
            loadavg[LOADAVG_15MIN]);
        (void) putp(t_eol);
        (void) putchar('\n');
        if (opts.o_outpmode & OPT_TTY)
                (void) putchar('\r');
        (void) putp(t_eol);
        (void) fflush(stdout);
}
```

******unchanged_portion_omitted_**

```
int
main(int argc, char **argv)
{
        DIR *procdir;
        char *p;
        char *sortk = "cpu";    /* default sort key */
        int opt;
        int timeout;
        struct pollfd pollset;
        char key;

        (void) setlocale(LC_ALL, "");
        (void) textdomain(TEXT_DOMAIN);
        Progname(argv[0]);
        lwpid_init();
        fd_init(Setrlimit());

        pagesize = sysconf(_SC_PAGESIZE);

        while ((opt = getopt(argc, argv,
            "vcd:HmarRLtu:U:n:p:C:P:h:s:S:j:k:TJWz:Z")) != (int)EOF) {
            "vcd:HmarRLtu:U:n:p:C:P:h:s:S:j:k:TJz:Z")) != (int)EOF) {
                switch (opt) {
                case 'r':
                        opts.o_outpmode |= OPT_NORESOLVE;
                        break;
                case 'R':
                        opts.o_outpmode |= OPT_REALTIME;
                        break;
                case 'c':
                        opts.o_outpmode &= ~OPT_TERMCAP;
                        opts.o_outpmode &= ~OPT_FULLSCREEN;
                        break;
                case 'd':
                        if (optarg) {
                                if (*optarg == 'u')
                                        opts.o_outpmode |= OPT_UDATE;
                                else if (*optarg == 'd')
                                        opts.o_outpmode |= OPT_DDATE;
                                else
                                        Usage();
                        } else {
                                Usage();
                        }
                        break;
                case 'h':
                        fill_table(&lgr_tbl, optarg, 'h');
                        break;
                case 'H':
                        opts.o_outpmode |= OPT_LGRP;
                        break;
                case 'm':
                case 'v':
                        opts.o_outpmode &= ~OPT_PSINFO;
                        opts.o_outpmode |=  OPT_MSACCT;
                        break;
                case 't':
                        opts.o_outpmode &= ~OPT_PSINFO;
                        opts.o_outpmode |= OPT_USERS;
                        break;
                case 'a':
                        opts.o_outpmode |= OPT_SPLIT | OPT_USERS;
                        break;
                case 'T':
                        opts.o_outpmode |= OPT_SPLIT | OPT_TASKS;
                        break;
                case 'J':
                        opts.o_outpmode |= OPT_SPLIT | OPT_PROJECTS;
                        break;
                case 'n':
                        if ((p = strtok(optarg, ",")) == NULL)
                                Die(gettext("invalid argument for -n\n"));
                        opts.o_ntop = Atoi(p);
                        if (p = strtok(NULL, ","))
                                opts.o_nbottom = Atoi(p);
                        opts.o_outpmode &= ~OPT_FULLSCREEN;
                        break;
                case 's':
                        opts.o_sortorder = -1;
                        sortk = optarg;
                        break;
                case 'S':
                        opts.o_sortorder = 1;
                        sortk = optarg;
                        break;
                case 'u':
                        if ((p = strtok(optarg, ", ")) == NULL)
                                Die(gettext("invalid argument for -u\n"));
                        add_uid(&euid_tbl, p);
                        while (p = strtok(NULL, ", "))
                                add_uid(&euid_tbl, p);
                        break;
                case 'U':
                        if ((p = strtok(optarg, ", ")) == NULL)
                                Die(gettext("invalid argument for -U\n"));
                        add_uid(&ruid_tbl, p);
                        while (p = strtok(NULL, ", "))
                                add_uid(&ruid_tbl, p);
                        break;
                case 'p':
                        fill_table(&pid_tbl, optarg, 'p');
                        break;
                case 'C':
                        fill_set_table(optarg);
                        opts.o_outpmode |= OPT_PSETS;
                        break;
                case 'P':
                        fill_table(&cpu_tbl, optarg, 'P');
                        break;
                case 'k':
                        fill_table(&tsk_tbl, optarg, 'k');
                        break;
                case 'j':
                        fill_prj_table(optarg);
                        break;
                case 'L':
```

```
                        opts.o_outpmode |= OPT_LWPS;
                        break;
                case 'W':
                        opts.o_outpmode |= OPT_TRUNC;
                        break;
                case 'z':
                        if ((p = strtok(optarg, ", ")) == NULL)
                                Die(gettext("invalid argument for -z\n"));
                        add_zone(&zone_tbl, p);
                        while (p = strtok(NULL, ", "))
                                add_zone(&zone_tbl, p);
                        break;
                case 'Z':
                        opts.o_outpmode |= OPT_SPLIT | OPT_ZONES;
                        break;
                default:
                        Usage();
                }
        }

        (void) atexit(Exit);
        if ((opts.o_outpmode & OPT_USERS) &&
            !(opts.o_outpmode & OPT_SPLIT))
                opts.o_nbottom = opts.o_ntop;
        if (opts.o_ntop == 0 || opts.o_nbottom == 0)
                Die(gettext("invalid argument for -n\n"));
        if (!(opts.o_outpmode & OPT_SPLIT) && (opts.o_outpmode & OPT_USERS) &&
            ((opts.o_outpmode & (OPT_PSINFO | OPT_MSACCT))))
                Die(gettext("-t option cannot be used with -v or -m\n"));

        if ((opts.o_outpmode & OPT_SPLIT) && (opts.o_outpmode & OPT_USERS) &&
            !((opts.o_outpmode & (OPT_PSINFO | OPT_MSACCT))))
                Die(gettext("-t option cannot be used with "
                    "-a, -J, -T or -Z\n"));

        if ((opts.o_outpmode & OPT_USERS) &&
            (opts.o_outpmode & (OPT_TASKS | OPT_PROJECTS | OPT_ZONES)))
                Die(gettext("-a option cannot be used with "
                    "-t, -J, -T or -Z\n"));

        if (((opts.o_outpmode & OPT_TASKS) &&
            (opts.o_outpmode & (OPT_PROJECTS|OPT_ZONES))) ||
            ((opts.o_outpmode & OPT_PROJECTS) &&
            (opts.o_outpmode & (OPT_TASKS|OPT_ZONES)))) {
                Die(gettext(
                    "-J, -T and -Z options are mutually exclusive\n"));
        }

        /*
         * There is not enough space to combine microstate information and
         * lgroup information and still fit in 80-column output.
         */
        if ((opts.o_outpmode & OPT_LGRP) && (opts.o_outpmode & OPT_MSACCT)) {
                Die(gettext("-H and -m options are mutually exclusive\n"));
        }

        if (argc > optind)
                opts.o_interval = Atoi(argv[optind++]);
        if (argc > optind)
                opts.o_count = Atoi(argv[optind++]);
        if (opts.o_count == 0)
                Die(gettext("invalid counter value\n"));
        if (argc > optind)
                Usage();
        if (opts.o_outpmode & OPT_REALTIME)
                Priocntl("RT");
        if (isatty(STDOUT_FILENO) == 1 && isatty(STDIN_FILENO))
                opts.o_outpmode |= OPT_TTY;     /* interactive */
        if (!(opts.o_outpmode & OPT_TTY)) {
                opts.o_outpmode &= ~OPT_TERMCAP; /* no termcap for pipes */
                opts.o_outpmode &= ~OPT_FULLSCREEN;
        }
        if (opts.o_outpmode & OPT_TERMCAP)
                ldtermcap();            /* can turn OPT_TERMCAP off */
        if (opts.o_outpmode & OPT_TERMCAP)
                (void) setsize();
        list_alloc(&lwps, opts.o_ntop);
        list_alloc(&users, opts.o_nbottom);
        list_alloc(&tasks, opts.o_nbottom);
        list_alloc(&projects, opts.o_nbottom);
        list_alloc(&zones, opts.o_nbottom);
        list_alloc(&lgroups, opts.o_nbottom);
        list_setkeyfunc(sortk, &opts, &lwps, LT_LWPS);
        list_setkeyfunc(NULL, &opts, &users, LT_USERS);
        list_setkeyfunc(NULL, &opts, &tasks, LT_TASKS);
        list_setkeyfunc(NULL, &opts, &projects, LT_PROJECTS);
        list_setkeyfunc(NULL, &opts, &zones, LT_ZONES);
        list_setkeyfunc(NULL, &opts, &lgroups, LT_LGRPS);
        if (opts.o_outpmode & OPT_TERMCAP)
                curses_on();
        if ((procdir = opendir("/proc")) == NULL)
                Die(gettext("cannot open /proc directory\n"));
        if (opts.o_outpmode & OPT_TTY) {
                (void) printf(gettext("Please wait...\r"));
                if (!(opts.o_outpmode & OPT_TERMCAP))
                        (void) putchar('\n');
                (void) fflush(stdout);
        }
        set_signals();
        pollset.fd = STDIN_FILENO;
        pollset.events = POLLIN;
        timeout = opts.o_interval * MILLISEC;

        /*
         * main program loop
         */
        do {
                if (sigterm == 1)
                        break;
                if (sigtstp == 1) {
                        curses_off();
                        (void) signal(SIGTSTP, SIG_DFL);
                        (void) kill(0, SIGTSTP);
                        /*
                         * prstat stops here until it receives SIGCONT signal.
                         */
                        sigtstp = 0;
                        (void) signal(SIGTSTP, sig_handler);
                        curses_on();
                        print_movecur = FALSE;
                        if (opts.o_outpmode & OPT_FULLSCREEN)
                                sigwinch = 1;
                }
                if (sigwinch == 1) {
                        if (setsize() == 1) {
                                list_free(&lwps);
                                list_free(&users);
                                list_free(&tasks);
                                list_free(&projects);
                                list_free(&zones);
                                list_alloc(&lwps, opts.o_ntop);
                                list_alloc(&users, opts.o_nbottom);
```

```
                                list_alloc(&tasks, opts.o_nbottom);
                                list_alloc(&projects, opts.o_nbottom);
                                list_alloc(&zones, opts.o_nbottom);
                        }
                        sigwinch = 0;
                        (void) signal(SIGWINCH, sig_handler);
                }
                prstat_scandir(procdir);
                list_refresh(&lwps);
                if (print_movecur)
                        (void) putp(movecur);
                print_movecur = TRUE;
                if ((opts.o_outpmode & OPT_PSINFO) ||
                    (opts.o_outpmode & OPT_MSACCT)) {
                        list_sort(&lwps);
                        list_print(&lwps);
                }
                if (opts.o_outpmode & OPT_USERS) {
                        list_getsize(&users);
                        list_sort(&users);
                        list_print(&users);
                        list_clear(&users);
                }
                if (opts.o_outpmode & OPT_TASKS) {
                        list_getsize(&tasks);
                        list_sort(&tasks);
                        list_print(&tasks);
                        list_clear(&tasks);
                }
                if (opts.o_outpmode & OPT_PROJECTS) {
                        list_getsize(&projects);
                        list_sort(&projects);
                        list_print(&projects);
                        list_clear(&projects);
                }
                if (opts.o_outpmode & OPT_ZONES) {
                        list_getsize(&zones);
                        list_sort(&zones);
                        list_print(&zones);
                        list_clear(&zones);
                }
                if (opts.o_count == 1)
                        break;
                /*
                 * If poll() returns -1 and sets errno to EINTR here because
                 * the process received a signal, it is Ok to abort this
                 * timeout and loop around because we check the signals at the
                 * top of the loop.
                 */
                if (opts.o_outpmode & OPT_TTY) {
                        if (poll(&pollset, (nfds_t)1, timeout) > 0) {
                                if (read(STDIN_FILENO, &key, 1) == 1) {
                                        if (tolower(key) == 'q')
                                                break;
                                }
                        }
                } else {
                        (void) sleep(opts.o_interval);
                }
        } while (opts.o_count == (-1) || --opts.o_count);

        if (opts.o_outpmode & OPT_TTY)
                (void) putchar('\r');
        return (0);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    5877 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/prstat/prstat.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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
 */

#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 */
        struct lwp_info *li_next;       /* pointer to next lwp */
        struct lwp_info *li_prev;       /* pointer to previous lwp */
} lwp_info_t;
______unchanged_portion_omitted_
```

```
**********************************************************
    6954 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/prstat/prtable.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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
 */

#include <procfs.h>
#include <unistd.h>
#include <stdlib.h>
#include <pwd.h>
#include <ctype.h>
#include <string.h>
#include <libintl.h>
#include <errno.h>
#include <zone.h>
#include <libzonecfg.h>
#include <wchar.h>

#include "prstat.h"
#include "prutil.h"
#include "prtable.h"

static plwp_t   *plwp_tbl[PLWP_TBL_SZ];

void
lwpid_init()
{
        (void) memset(&plwp_tbl, 0, sizeof (plwp_t *) * PLWP_TBL_SZ);
}
______unchanged_portion_omitted_

void
pwd_getname(uid_t uid, char *name, size_t length, int noresolve,
    int trunc, size_t width)
pwd_getname(uid_t uid, char *name, int length, int noresolve)
{
        struct passwd *pwd;
```

```
        size_t n;

        if (noresolve || (pwd = getpwuid(uid)) == NULL) {
                n = snprintf(NULL, 0, "%u", uid);
                if (trunc && n > width)
                        (void) snprintf(name, length, "%.*u%c",
                            width - 1, uid, '*');
                else
                        (void) snprintf(name, length, "%u", uid);
        } else {
                n = mbstowcs(NULL, pwd->pw_name, 0);
                if (n == (size_t)-1)
                        (void) snprintf(name, length, "%s", "ERROR");
                else if (trunc && n > width)
                        (void) snprintf(name, length, "%.*s%c",
                            width - 1, pwd->pw_name, '*');
                else
                        (void) snprintf(name, length, "%s", pwd->pw_name);
        }
}
______unchanged_portion_omitted_
```

****
    2466 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/prstat/prtable.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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
 */

#ifndef _PRTABLE_H
#define _PRTABLE_H

#ifdef  __cplusplus
extern "C" {
#endif

#include <limits.h>
#include <zone.h>
#include "prstat.h"

#define PLWP_TBL_SZ     4096    /* hash table of plwp_t structures */
#define LWP_ACTIVE      1

typedef struct {
        size_t          t_size;
        size_t          t_nent;
        long            *t_list;
} table_t;
```

______*unchanged_portion_omitted_*

```
extern void pwd_getname(uid_t, char *, size_t, int, int, size_t);
extern void pwd_getname(uid_t, char *, int, int);
extern void add_uid(uidtbl_t *, char *);
extern int has_uid(uidtbl_t *, uid_t);
extern void add_element(table_t *, long);
extern int has_element(table_t *, long);
extern void add_zone(zonetbl_t *, char *);
extern int has_zone(zonetbl_t *, zoneid_t);
extern void convert_zone(zonetbl_t *);
extern int foreach_element(table_t *, void *, void (*)(long, void *));
```

```
extern void lwpid_init();
extern void lwpid_add(lwp_info_t *, pid_t, id_t);
extern lwp_info_t *lwpid_get(pid_t, id_t);
extern int lwpid_pidcheck(pid_t);
extern void lwpid_del(pid_t, id_t);
extern void lwpid_set_active(pid_t, id_t);
extern int lwpid_is_active(pid_t, id_t);

#ifdef  __cplusplus
}
```

______*unchanged_portion_omitted_*

```
**********************************************************
    7718 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/prstat/prutil.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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
 */

#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_

void
Usage()
{
        (void) fprintf(stderr, gettext(
            "Usage:\tprstat [-acHJLmrRtTvWZ] [-u euidlist] [-U uidlist]\n"
            "Usage:\tprstat [-acHJLmrRtTvZ] [-u euidlist] [-U uidlist]\n"
            "\t[-p pidlist] [-P cpulist] [-C psrsetlist] [-h lgrouplist]\n"
            "\t[-j projidlist] [-k taskidlist] [-z zoneidlist]\n"
            "\t[-s key | -S key] [-n nprocs[,nusers]] [-d d|u]\n"
            "\t[interval [counter]]\n"));
        exit(1);
}
______unchanged_portion_omitted_

void
getprojname(projid_t projid, char *str, size_t len, int noresolve,
    int trunc, size_t width)
getprojname(projid_t projid, char *str, int len, int noresolve)
{
        struct project proj;
        size_t n;

        if (noresolve || getprojbyid(projid, &proj, projbuf, PROJECT_BUFSZ) ==
            NULL) {
            NULL)
                (void) snprintf(str, len, "%-6d", (int)projid);
        } else {
                n = mbstowcs(NULL, proj.pj_name, 0);
                if (n == (size_t)-1)
                        (void) snprintf(str, len, "%-28s", "ERROR");
                else if (trunc && n > width)
                        (void) snprintf(str, len, "%.*s%c", width - 1,
                            proj.pj_name, '*');
                else
                        (void) snprintf(str, len, "%-28s", proj.pj_name);
        }
}

void
getzonename(zoneid_t zoneid, char *str, size_t len, int trunc, size_t width)
getzonename(zoneid_t zoneid, char *str, int len)
{
        char zone_name[ZONENAME_MAX];
        size_t n;

        if (getzonenamebyid(zoneid, zone_name, sizeof (zone_name)) < 0) {
        if (getzonenamebyid(zoneid, zone_name, sizeof (zone_name)) < 0)
```

```
                (void) snprintf(str, len, "%-6d", (int)zoneid);
        } else {
                n = mbstowcs(NULL, zone_name, 0);
                if (n == (size_t)-1)
                        (void) snprintf(str, len, "%-28s", "ERROR");
                else if (trunc && n > width)
                        (void) snprintf(str, len, "%.*s%c", width - 1,
                            zone_name, '*');
                else
                        (void) snprintf(str, len, "%-28s", zone_name);
        }
}
```
______***unchanged_portion_omitted_***

****
1878 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/prstat/prutil.h
_2989 Eliminate use of LOGNAME_MAX in ON_
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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
 */

#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 getprojname(projid_t, char *, int, int);
extern void getzonename(projid_t, char *, int);
extern void stripfname(char *);
```

```
#ifdef  __cplusplus
}
```
______**_unchanged_portion_omitted_**

****
61349 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/ps/ps.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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) 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)
#define MAXUGNAME 10    /* max chars in a user/group name or printed u/g id */

/* 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_

#define NFIELDS (sizeof (fname) / sizeof (fname[0]))

static  int     retcode = 1;
static  int     lflg;
static  int     Aflg;
static  int     uflg;
static  int     Uflg;
static  int     Gflg;
static  int     aflg;
static  int     dflg;
static  int     Lflg;
static  int     Pflg;
static  int     Wflg;
static  int     yflg;
static  int     pflg;
static  int     fflg;
static  int     cflg;
static  int     jflg;
static  int     gflg;
static  int     sflg;
static  int     tflg;
static  int     zflg;
static  int     Zflg;
static  int     hflg;
static  int     Hflg;
static  uid_t   tuid = (uid_t)-1;
static  int     errflg;

static  int     ndev;           /* number of devices */
static  int     maxdev;         /* number of devl structures allocated */

#define DNINCR  100
#define DNSIZE  14
static struct devl {            /* device list   */
        char    dname[DNSIZE];  /* device name   */
        dev_t   ddev;           /* device number */
} *devl;
```

______*unchanged_portion_omitted_*

```
static int
stdmain(int argc, char **argv)
{
        char    *p;
        char    *p1;
        char    *parg;
        int     c;
        int     i;
        int     pgerrflg = 0;   /* err flg: non-numeric arg w/p & g options */
        size_t  size, len;
        DIR     *dirp;
        struct dirent *dentp;
        pid_t   maxpid;
        pid_t   id;
        int     ret;
        char    loc_stime_str[32];

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

        (void) memset(&euid_tbl, 0, sizeof (euid_tbl));
        (void) memset(&ruid_tbl, 0, sizeof (ruid_tbl));
        (void) memset(&egid_tbl, 0, sizeof (egid_tbl));
        (void) memset(&rgid_tbl, 0, sizeof (rgid_tbl));

        kbytes_per_page = sysconf(_SC_PAGESIZE) / 1024;

        (void) gettimeofday(&now, NULL);

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

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

        /*
         * TRANSLATION_NOTE
         * Specify the printf format with width and precision for
         * the STIME field.
         */
        len = snprintf(loc_stime_str, sizeof (loc_stime_str),
            dcgettext(NULL, "%8.8s", LC_TIME), "STIME");
        if (len >= sizeof (loc_stime_str))
                len = sizeof (loc_stime_str) - 1;

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

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

```
                        Hflg++;
                        break;
                case 'h':
                        /*
                         * Show processes/threads with given home lgroups
                         */
                        hflg++;
                        p1 = optarg;
                        do {
                                int id;

                                /*
                                 * Get all IDs in the list, verify for
                                 * correctness and place in lgrps array.
                                 */
                                parg = getarg(&p1);
                                /* Convert string to integer */
                                ret = str2id(parg, (pid_t *)&id, 0,
                                    MAX_LGRP_ID);
                                /* Complain if ID didn't parse correctly */
                                if (ret != 0) {
                                        pgerrflg++;
                                        (void) fprintf(stderr,
                                            gettext("ps: %s "), parg);
                                        if (ret == EINVAL)
                                                (void) fprintf(stderr,
                                                    gettext("is an invalid "
                                                    "non-numeric argument"));
                                        else
                                                (void) fprintf(stderr,
                                                    gettext("exceeds valid "
                                                    "range"));
                                        (void) fprintf(stderr,
                                            gettext(" for -h option\n"));
                                        continue;
                                }

                                /* Extend lgrps array if needed */
                                if (nlgrps == lgrps_size) {
                                        /* Double the size of the lgrps array */
                                        if (lgrps_size == 0)
                                                lgrps_size = SIZ;
                                        lgrps_size *= 2;
                                        lgrps = Realloc(lgrps,
                                            lgrps_size * sizeof (int));
                                }
                                /* place the id in the lgrps table */
                                lgrps[nlgrps++] = id;
                        } while (*p1);
                        break;
                case 'l':               /* long listing */
                        lflg++;
                        break;
                case 'f':               /* full listing */
                        fflg++;
                        break;
                case 'j':
                        jflg++;
                        break;
                case 'c':
                        /*
                         * Format output to reflect scheduler changes:
                         * high numbers for high priorities and don't
                         * print nice or p_cpu values.  'c' option only
                         * effective when used with 'l' or 'f' options.
                         */
                        cflg++;
                        break;
                case 'A':               /* list every process */
                case 'e':               /* (obsolete) list every process */
                        Aflg++;
                        tflg = Gflg = Uflg = uflg = pflg = gflg = sflg = 0;
                        zflg = hflg = 0;
                        break;
                case 'a':
                        /*
                         * Same as 'e' except no session group leaders
                         * and no non-terminal processes.
                         */
                        aflg++;
                        break;
                case 'd':       /* same as e except no session leaders */
                        dflg++;
                        break;
                case 'L':       /* show lwps */
                        Lflg++;
                        break;
                case 'P':       /* show bound processor */
                        Pflg++;
                        break;
                case 'W':       /* truncate long names */
                        Wflg++;
                        break;
                case 'y':       /* omit F & ADDR, report RSS & SZ in Kby */
                        yflg++;
                        break;
                case 'n':       /* no longer needed; retain as no-op */
                        (void) fprintf(stderr,
                            gettext("ps: warning: -n option ignored\n"));
                        break;
                case 't':               /* terminals */
#define TSZ     30
                        tflg++;
                        p1 = optarg;
                        do {
                                char nambuf[TSZ+6];     /* for "/dev/" + '\0' */
                                struct stat64 s;
                                parg = getarg(&p1);
                                p = Realloc(NULL, TSZ+1);       /* for '\0' */
                                /* zero the buffer before using it */
                                p[0] = '\0';
                                size = TSZ;
                                if (isdigit(*parg)) {
                                        (void) strcpy(p, "tty");
                                        size -= 3;
                                }
                                (void) strncat(p, parg, size);
                                if (ntty == ttysz) {
                                        if ((ttysz *= 2) == 0)
                                                ttysz = NTTYS;
                                        tty = Realloc(tty,
                                            (ttysz + 1) * sizeof (struct tty));
                                }
                                tty[ntty].tdev = PRNODEV;
                                (void) strcpy(nambuf, "/dev/");
                                (void) strcat(nambuf, p);
                                if (stat64(nambuf, &s) == 0)
                                        tty[ntty].tdev = s.st_rdev;
                                tty[ntty++].tname = p;
                        } while (*p1);
                        break;
                case 'p':               /* proc ids */
```

```
                        pflg++;
                        p1 = optarg;
                        do {
                                pid_t id;

                                parg = getarg(&p1);
                                if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
                                        pgerrflg++;
                                        (void) fprintf(stderr,
                                            gettext("ps: %s "), parg);
                                        if (ret == EINVAL)
                                                (void) fprintf(stderr,
                                                    gettext("is an invalid "
                                                    "non-numeric argument"));
                                        else
                                                (void) fprintf(stderr,
                                                    gettext("exceeds valid "
                                                    "range"));
                                        (void) fprintf(stderr,
                                            gettext(" for -p option\n"));
                                        continue;
                                }

                                if (npid == pidsz) {
                                        if ((pidsz *= 2) == 0)
                                                pidsz = SIZ;
                                        pid = Realloc(pid,
                                            pidsz * sizeof (pid_t));
                                }
                                pid[npid++] = id;
                        } while (*p1);
                        break;
                case 's':               /* session */
                        sflg++;
                        p1 = optarg;
                        do {
                                pid_t id;

                                parg = getarg(&p1);
                                if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
                                        pgerrflg++;
                                        (void) fprintf(stderr,
                                            gettext("ps: %s "), parg);
                                        if (ret == EINVAL)
                                                (void) fprintf(stderr,
                                                    gettext("is an invalid "
                                                    "non-numeric argument"));
                                        else
                                                (void) fprintf(stderr,
                                                    gettext("exceeds valid "
                                                    "range"));
                                        (void) fprintf(stderr,
                                            gettext(" for -s option\n"));
                                        continue;
                                }

                                if (nsessid == sessidsz) {
                                        if ((sessidsz *= 2) == 0)
                                                sessidsz = SIZ;
                                        sessid = Realloc(sessid,
                                            sessidsz * sizeof (pid_t));
                                }
                                sessid[nsessid++] = id;
                        } while (*p1);
                        break;
                case 'g':               /* proc group */
```

```
                        gflg++;
                        p1 = optarg;
                        do {
                                pid_t id;

                                parg = getarg(&p1);
                                if ((ret = str2id(parg, &id, 0, maxpid)) != 0) {
                                        pgerrflg++;
                                        (void) fprintf(stderr,
                                            gettext("ps: %s "), parg);
                                        if (ret == EINVAL)
                                                (void) fprintf(stderr,
                                                    gettext("is an invalid "
                                                    "non-numeric argument"));
                                        else
                                                (void) fprintf(stderr,
                                                    gettext("exceeds valid "
                                                    "range"));
                                        (void) fprintf(stderr,
                                            gettext(" for -g option\n"));
                                        continue;
                                }

                                if (ngrpid == grpidsz) {
                                        if ((grpidsz *= 2) == 0)
                                                grpidsz = SIZ;
                                        grpid = Realloc(grpid,
                                            grpidsz * sizeof (pid_t));
                                }
                                grpid[ngrpid++] = id;
                        } while (*p1);
                        break;
                case 'u':               /* effective user name or number */
                        uflg++;
                        p1 = optarg;
                        do {
                                parg = getarg(&p1);
                                add_ugentry(&euid_tbl, parg);
                        } while (*p1);
                        break;
                case 'U':               /* real user name or number */
                        Uflg++;
                        p1 = optarg;
                        do {
                                parg = getarg(&p1);
                                add_ugentry(&ruid_tbl, parg);
                        } while (*p1);
                        break;
                case 'G':               /* real group name or number */
                        Gflg++;
                        p1 = optarg;
                        do {
                                parg = getarg(&p1);
                                add_ugentry(&rgid_tbl, parg);
                        } while (*p1);
                        break;
                case 'o':               /* output format */
                        p = optarg;
                        while ((p = parse_format(p)) != NULL)
                                ;
                        break;
                case 'z':               /* zone name or number */
                        zflg++;
                        p1 = optarg;
                        do {
                                zoneid_t id;
```

```

                                parg = getarg(&p1);
                                if (zone_get_id(parg, &id) != 0) {
                                        pgerrflg++;
                                        (void) fprintf(stderr,
                                            gettext("ps: unknown zone %s\n"),
                                            parg);
                                        continue;
                                }

                                if (nzoneid == zoneidsz) {
                                        if ((zoneidsz *= 2) == 0)
                                                zoneidsz = SIZ;
                                        zoneid = Realloc(zoneid,
                                            zoneidsz * sizeof (zoneid_t));
                                }
                                zoneid[nzoneid++] = id;
                        } while (*p1);
                        break;
                case 'Z':               /* show zone name */
                        Zflg++;
                        break;
                default:                        /* error on ? */
                        errflg++;
                        break;
                }

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

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

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

                i = 0;
                name = gettty(&info);
                if (*name == '?') {
                        (void) fprintf(stderr,
                            gettext("ps: can't find controlling terminal\n"));
                        exit(1);
                }
                if (ntty == ttysz) {
                        if ((ttysz *= 2) == 0)
                                ttysz = NTTYS;
                        tty = Realloc(tty, (ttysz + 1) * sizeof (struct tty));
                }
                tty[ntty].tdev = info.pr_ttydev;
                tty[ntty++].tname = name;
                tty[ntty].tname = NULL;
                tflg++;
                tuid = getuid();
        }
        if (Aflg) {
                Gflg = Uflg = uflg = pflg = sflg = gflg = aflg = dflg = 0;
                zflg = hflg = 0;
        }
        if (Aflg | aflg | dflg)
                tflg = 0;

        i = 0;          /* prepare to exit on name lookup errors */
        i += uconv(&euid_tbl);
        i += uconv(&ruid_tbl);
        i += gconv(&egid_tbl);
        i += gconv(&rgid_tbl);
        if (i)
                exit(1);

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

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

                        for (f = fields; f != NULL; f = f->next) {
                                if (f != fields)
                                        (void) printf(" ");
                                switch (f->fname) {
                                case F_TTY:
                                        (void) printf("%-*s",
                                            f->width, f->header);
                                        break;
                                case F_FNAME:
                                case F_COMM:
                                case F_ARGS:
                                        /*
                                         * Print these headers full width
                                         * unless they appear at the end.
                                         */
                                        if (f->next != NULL) {
                                                (void) printf("%-*s",
                                                    f->width, f->header);
                                        } else {
                                                (void) printf("%s",
                                                    f->header);
                                        }
                                        break;
                                default:
                                        (void) printf("%*s",
                                            f->width, f->header);
                                        break;
                                }
                        }
                        (void) printf("\n");
                }
        } else {        /* print standard header */
                /*
```

```
                 * All fields before 'PID' are printed with a trailing space
                 * as a separator and that is how we print the headers too.
                 */
                if (lflg) {
                        if (yflg)
                                (void) printf("S ");
                        else
                                (void) printf(" F S ");
                }
                if (Zflg)
                        (void) printf("    ZONE ");
                if (fflg) {
                        (void) printf("     UID ");
                } else if (lflg)
                        (void) printf("   UID ");

                (void) printf("%*s", pidwidth,  "PID");
                if (lflg || fflg)
                        (void) printf(" %*s", pidwidth, "PPID");
                if (jflg)
                        (void) printf(" %*s %*s", pidwidth, "PGID",
                            pidwidth, "SID");
                if (Lflg)
                        (void) printf("   LWP");
                if (Pflg)
                        (void) printf(" PSR");
                if (Lflg && fflg)
                        (void) printf("  NLWP");
                if (cflg)
                        (void) printf("  CLS PRI");
                else if (lflg || fflg) {
                        (void) printf("   C");
                        if (lflg)
                                (void) printf(" PRI NI");
                }
                if (lflg) {
                        if (yflg)
                                (void) printf("   RSS     SZ    WCHAN");
                        else
                                (void) printf("     ADDR     SZ    WCHAN");
                }
                if (fflg)
                        (void) printf(" %s", loc_stime_str);
                if (Hflg)
                        (void) printf(" LGRP");
                if (Lflg)
                        (void) printf(" TTY        LTIME CMD\n");
                else
                        (void) printf(" TTY         TIME CMD\n");
        }


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

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

                        if (i >= 1 && pid[i] == pid[i - 1])
                                continue;
                        (void) sprintf(pname, "%d", (int)pid[i]);
                        if (print_proc(pname) == 0)
                                retcode = 0;
                }
        } else {
                /*
                 * Determine which processes to print info about by searching
                 * the /proc directory and looking at each process.
                 */
                if ((dirp = opendir(procdir)) == NULL) {
                        (void) fprintf(stderr,
                            gettext("ps: cannot open PROC directory %s\n"),
                            procdir);
                        exit(1);
                }

                /* for each active process --- */
                while (dentp = readdir(dirp)) {
                        if (dentp->d_name[0] == '.')    /* skip . and .. */
                                continue;
                        if (print_proc(dentp->d_name) == 0)
                                retcode = 0;
                }

                (void) closedir(dirp);
        }
        return (retcode);
}
______unchanged_portion_omitted_

static void
usage(void)             /* print usage message and quit */
{
        struct def_field *df, *sorted[NFIELDS];
        int pos = 80, i = 0;

        static char usage1[] =
            "ps [ -aAdefHlcjLPWyZ ] [ -o format ] [ -t termlist ]";
            "ps [ -aAdefHlcjLPyZ ] [ -o format ] [ -t termlist ]";
        static char usage2[] =
            "\t[ -u userlist ] [ -U userlist ] [ -G grouplist ]";
        static char usage3[] =
            "\t[ -p proclist ] [ -g pgrplist ] [ -s sidlist ]";
        static char usage4[] =
            "\t[ -z zonelist ] [-h lgrplist]";
        static char usage5[] =
            "  'format' is one or more of:";

        (void) fprintf(stderr,
            gettext("usage: %s\n%s\n%s\n%s\n%s"),
            gettext(usage1), gettext(usage2), gettext(usage3),
            gettext(usage4), gettext(usage5));

        /*
         * Now print out the possible output formats such that they neatly fit
         * into eighty columns.  Note that the fact that we are determining
         * this output programmatically means that a gettext() is impossible --
         * but it would be a mistake to localize the output formats anyway as
         * they are tokens for input, not output themselves.
         */
        for (df = &fname[0]; df < &fname[NFIELDS]; df++)
                sorted[i++] = df;

        (void) qsort(sorted, NFIELDS, sizeof (void *), field_cmp);

        for (i = 0; i < NFIELDS; i++) {
```

```
                if (pos + strlen((df = sorted[i])->fname) + 1 >= 80) {
                        (void) fprintf(stderr, "\n\t");
                        pos = 8;
                }

                (void) fprintf(stderr, "%s%s", pos > 8 ? " " : "", df->fname);
                pos += strlen(df->fname) + 1;
        }

        (void) fprintf(stderr, "\n");

        exit(1);
}
______unchanged_portion_omitted_

/*
 * Print info about the process.
 */
static void
prcom(psinfo_t *psinfo, char *ttyp)
{
        char    *cp;
        long    tm;
        int     bytesleft;
        int     wcnt, length;
        wchar_t wchar;
        struct passwd *pwd;
        int     zombie_lwp;
        char    zonename[ZONENAME_MAX];

        /*
         * If process is zombie, call zombie print routine and return.
         */
        if (psinfo->pr_nlwp == 0) {
                if (fields != NULL)
                        pr_fields(psinfo, ttyp, print_zombie_field);
                else
                        przom(psinfo);
                return;
        }

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

        /*
         * If user specified '-o format', print requested fields and return.
         */
        if (fields != NULL) {
                pr_fields(psinfo, ttyp, print_field);
                return;
        }

        /*
         * All fields before 'PID' are printed with a trailing space as a
         * separator, rather than keeping track of which column is first.  All
         * other fields are printed with a leading space.
         */
        if (lflg) {
                if (!yflg)
                        (void) printf("%2x ", psinfo->pr_flag & 0377); /* F */
                (void) printf("%c ", psinfo->pr_lwp.pr_sname);  /* S */
        }

        if (Zflg) {                                             /* ZONE */
                if (getzonenamebyid(psinfo->pr_zoneid, zonename,
                    sizeof (zonename)) < 0) {
                        if (snprintf(NULL, 0, "%d",
                            ((int)psinfo->pr_zoneid)) > 7)
                                (void) printf(" %6.6d%c ",
                                    ((int)psinfo->pr_zoneid), ’*’);
                        else
                                (void) printf(" %7.7d ",
                                    ((int)psinfo->pr_zoneid));
                        (void) printf(" %7.7d ", ((int)psinfo->pr_zoneid));
                } else {
                        size_t nw;

                        nw = mbstowcs(NULL, zonename, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%8.8s ", "ERROR");
                        else if (nw > 8)
                                (void) wprintf(L"%7.7s%c ", zonename, ’*’);
                        else
                                (void) wprintf(L"%8.8s ", zonename);
                        (void) printf("%8.8s ", zonename);
                }
        }

        if (fflg) {                                             /* UID */
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL) {
                        size_t nw;

                        nw = mbstowcs(NULL, pwd->pw_name, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%8.8s ", "ERROR");
                        else if (nw > 8)
                                (void) wprintf(L"%7.7s%c ", pwd->pw_name, ’*’);
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL)
                        (void) printf("%8.8s ", pwd->pw_name);
                        else
                                (void) wprintf(L"%8.8s ", pwd->pw_name);
                } else {
                        if (snprintf(NULL, 0, "%u",
                            (psinfo->pr_euid)) > 7)
                                (void) printf(" %6.6u%c ", psinfo->pr_euid,
                                    ’*’);
                        else
                                (void) printf(" %7.7u ", psinfo->pr_euid);
                }
        } else if (lflg) {
                if (snprintf(NULL, 0, "%u", (psinfo->pr_euid)) > 6)
                        (void) printf("%5.5u%c ", psinfo->pr_euid, ’*’);
                else
                        (void) printf("%6u ", psinfo->pr_euid);
        }
        (void) printf("%*d", pidwidth, (int)psinfo->pr_pid); /* PID */
        if (lflg || fflg)
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_ppid); /* PPID */
        if (jflg) {
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_pgid);      /* PGID */
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_sid);       /* SID  */
        }
        if (Lflg)
                (void) printf(" %5d", (int)psinfo->pr_lwp.pr_lwpid); /* LWP */
        if (Pflg) {
                if (psinfo->pr_lwp.pr_bindpro == PBIND_NONE)    /* PSR */
                        (void) printf("   -");
                else
                        (void) printf(" %3d", psinfo->pr_lwp.pr_bindpro);
        }
```

```
        if (Lflg && fflg)                                       /* NLWP */
                (void) printf(" %5d", psinfo->pr_nlwp + psinfo->pr_nzomb);
        if (cflg) {
                if (zombie_lwp)                                 /* CLS */
                        (void) printf("     ");
                else
                        (void) printf(" %4s", psinfo->pr_lwp.pr_clname);
                (void) printf(" %3d", psinfo->pr_lwp.pr_pri);   /* PRI */
        } else if (lflg || fflg) {
                (void) printf(" %3d", psinfo->pr_lwp.pr_cpu & 0377); /* C   */
                if (lflg) {                                         /* PRI NI */
                        /*
                         * Print priorities the old way (lower numbers
                         * mean higher priority) and print nice value
                         * for time sharing procs.
                         */
                        (void) printf(" %3d", psinfo->pr_lwp.pr_oldpri);
                        if (psinfo->pr_lwp.pr_oldpri != 0)
                                (void) printf(" %2d", psinfo->pr_lwp.pr_nice);
                        else
                                (void) printf(" %2.2s",
                                    psinfo->pr_lwp.pr_clname);
                }
        }
        if (lflg) {
                if (yflg) {
                        if (psinfo->pr_flag & SSYS)             /* RSS */
                                (void) printf("     0");
                        else if (psinfo->pr_rssize)
                                (void) printf(" %5lu",
                                    (ulong_t)psinfo->pr_rssize);
                        else
                                (void) printf("     ?");
                        if (psinfo->pr_flag & SSYS)             /* SZ */
                                (void) printf("      0");
                        else if (psinfo->pr_size)
                                (void) printf(" %6lu",
                                    (ulong_t)psinfo->pr_size);
                        else
                                (void) printf("      ?");
                } else {
#ifndef _LP64
                        if (psinfo->pr_addr)                    /* ADDR */
                                (void) printf(" %8lx",
                                    (ulong_t)psinfo->pr_addr);
                        else
#endif
                                (void) printf("        ?");
                        if (psinfo->pr_flag & SSYS)             /* SZ */
                                (void) printf("      0");
                        else if (psinfo->pr_size)
                                (void) printf(" %6lu",
                                    (ulong_t)psinfo->pr_size / kbytes_per_page);
                        else
                                (void) printf("      ?");
                }
                if (psinfo->pr_lwp.pr_sname != ’S’)             /* WCHAN */
                        (void) printf("         ");
#ifndef _LP64
                else if (psinfo->pr_lwp.pr_wchan)
                        (void) printf(" %8lx",
                            (ulong_t)psinfo->pr_lwp.pr_wchan);
#endif
                else
                        (void) printf("        ?");
        }
        if (fflg) {                                             /* STIME */
                int width = fname[F_STIME].width;
                if (Lflg)
                        prtime(psinfo->pr_lwp.pr_start, width + 1, 1);
                else
                        prtime(psinfo->pr_start, width + 1, 1);
        }

        if (Hflg) {
                /* Display home lgroup */
                (void) printf(" %4d", (int)psinfo->pr_lwp.pr_lgrp);
        }

        (void) printf(" %-8.14s", ttyp);                        /* TTY */
        if (Lflg) {
                tm = psinfo->pr_lwp.pr_time.tv_sec;
                if (psinfo->pr_lwp.pr_time.tv_nsec > 500000000)
                        tm++;
        } else {
                tm = psinfo->pr_time.tv_sec;
                if (psinfo->pr_time.tv_nsec > 500000000)
                        tm++;
        }
        (void) printf(" %4ld:%.2ld", tm / 60, tm % 60);         /* [L]TIME */

        if (zombie_lwp) {
                (void) printf(" <defunct>\n");
                return;
        }

        if (!fflg) {                                            /* CMD */
                wcnt = namencnt(psinfo->pr_fname, 16, 8);
                (void) printf(" %.*s\n", wcnt, psinfo->pr_fname);
                return;
        }


        /*
         * PRARGSZ == length of cmd arg string.
         */
        psinfo->pr_psargs[PRARGSZ-1] = ’\0’;
        bytesleft = PRARGSZ;
        for (cp = psinfo->pr_psargs; *cp != ’\0’; cp += length) {
                length = mbtowc(&wchar, cp, MB_LEN_MAX);
                if (length == 0)
                        break;
                if (length < 0 || !iswprint(wchar)) {
                        if (length < 0)
                                length = 1;
                        if (bytesleft <= length) {
                                *cp = ’\0’;
                                break;
                        }
                        /* omit the unprintable character */
                        (void) memmove(cp, cp+length, bytesleft-length);
                        length = 0;
                }
                bytesleft -= length;
        }
        wcnt = namencnt(psinfo->pr_psargs, PRARGSZ, lflg ? 35 : PRARGSZ);
        (void) printf(" %.*s\n", wcnt, psinfo->pr_psargs);
}
```

______unchanged_portion_omitted_

```
static void
print_field(psinfo_t *psinfo, struct field *f, const char *ttyp)
```

```
{
        int width = f->width;
        struct passwd *pwd;
        struct group *grp;
        time_t cputime;
        int bytesleft;
        int wcnt;
        wchar_t wchar;
        char *cp;
        int length;
        ulong_t mask;
        char c, *csave;
        int zombie_lwp;

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

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

                        nw = mbstowcs(NULL, pwd->pw_name, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%*s ", width, "ERROR");
                        else if (Wflg && nw > width)
                                (void) wprintf(L"%.*s%c", width - 1,
                                    pwd->pw_name, ’*’);
                if ((pwd = getpwuid(psinfo->pr_uid)) != NULL)
                        (void) printf("%*s", width, pwd->pw_name);
                        else
                                (void) wprintf(L"%*s", width, pwd->pw_name);
                } else {
                        if (Wflg && snprintf(NULL, 0, "%u",
                            (psinfo->pr_uid)) > width)

                                (void) printf("%*u%c", width - 1,
                                    psinfo->pr_uid, ’*’);
                        else
                                (void) printf("%*u", width, psinfo->pr_uid);
                }
                break;
        case F_USER:
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL) {
                        size_t nw;

                        nw = mbstowcs(NULL, pwd->pw_name, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%*s ", width, "ERROR");
                        else if (Wflg && nw > width)
                                (void) wprintf(L"%.*s%c", width - 1,
                                    pwd->pw_name, ’*’);
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL)
                        (void) printf("%*s", width, pwd->pw_name);
                        else
                                (void) wprintf(L"%*s", width, pwd->pw_name);
                } else {
                        if (Wflg && snprintf(NULL, 0, "%u",
                            (psinfo->pr_euid)) > width)

                                (void) printf("%*u%c", width - 1,
                                    psinfo->pr_euid, ’*’);
                        else
                                (void) printf("%*u", width, psinfo->pr_euid);
                }
                break;
        case F_RGROUP:
```

```
		if ((grp = getgrgid(psinfo->pr_gid)) != NULL)
			(void) printf("%*s", width, grp->gr_name);
		else
			(void) printf("%*u", width, psinfo->pr_gid);
		break;
	case F_GROUP:
		if ((grp = getgrgid(psinfo->pr_egid)) != NULL)
			(void) printf("%*s", width, grp->gr_name);
		else
			(void) printf("%*u", width, psinfo->pr_egid);
		break;
	case F_RUID:
		(void) printf("%*u", width, psinfo->pr_uid);
		break;
	case F_UID:
		(void) printf("%*u", width, psinfo->pr_euid);
		break;
	case F_RGID:
		(void) printf("%*u", width, psinfo->pr_gid);
		break;
	case F_GID:
		(void) printf("%*u", width, psinfo->pr_egid);
		break;
	case F_PID:
		(void) printf("%*d", width, (int)psinfo->pr_pid);
		break;
	case F_PPID:
		(void) printf("%*d", width, (int)psinfo->pr_ppid);
		break;
	case F_PGID:
		(void) printf("%*d", width, (int)psinfo->pr_pgid);
		break;
	case F_SID:
		(void) printf("%*d", width, (int)psinfo->pr_sid);
		break;
	case F_PSR:
		if (zombie_lwp || psinfo->pr_lwp.pr_bindpro == PBIND_NONE)
			(void) printf("%*s", width, "-");
		else
			(void) printf("%*d", width, psinfo->pr_lwp.pr_bindpro);
		break;
	case F_LWP:
		(void) printf("%*d", width, (int)psinfo->pr_lwp.pr_lwpid);
		break;
	case F_NLWP:
		(void) printf("%*d", width, psinfo->pr_nlwp + psinfo->pr_nzomb);
		break;
	case F_OPRI:
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else
			(void) printf("%*d", width, psinfo->pr_lwp.pr_oldpri);
		break;
	case F_PRI:
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else
			(void) printf("%*d", width, psinfo->pr_lwp.pr_pri);
		break;
	case F_F:
		mask = 0xffffffffUL;
		if (width < 8)
			mask >>= (8 - width) * 4;
		(void) printf("%*lx", width, psinfo->pr_flag & mask);
		break;
	case F_S:
```

```
		(void) printf("%*c", width, psinfo->pr_lwp.pr_sname);
		break;
	case F_C:
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else
			(void) printf("%*d", width, psinfo->pr_lwp.pr_cpu);
		break;
	case F_PCPU:
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else if (Lflg)
			prtpct(psinfo->pr_lwp.pr_pctcpu, width);
		else
			prtpct(psinfo->pr_pctcpu, width);
		break;
	case F_PMEM:
		prtpct(psinfo->pr_pctmem, width);
		break;
	case F_OSZ:
		(void) printf("%*lu", width,
		    (ulong_t)psinfo->pr_size / kbytes_per_page);
		break;
	case F_VSZ:
		(void) printf("%*lu", width, (ulong_t)psinfo->pr_size);
		break;
	case F_RSS:
		(void) printf("%*lu", width, (ulong_t)psinfo->pr_rssize);
		break;
	case F_NICE:
		/* if pr_oldpri is zero, then this class has no nice */
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else if (psinfo->pr_lwp.pr_oldpri != 0)
			(void) printf("%*d", width, psinfo->pr_lwp.pr_nice);
		else
			(void) printf("%*.*s", width, width,
			    psinfo->pr_lwp.pr_clname);
		break;
	case F_CLASS:
		if (zombie_lwp)
			(void) printf("%*s", width, "-");
		else
			(void) printf("%*.*s", width, width,
			    psinfo->pr_lwp.pr_clname);
		break;
	case F_STIME:
		if (Lflg)
			prtime(psinfo->pr_lwp.pr_start, width, 0);
		else
			prtime(psinfo->pr_start, width, 0);
		break;
	case F_ETIME:
		if (Lflg)
			print_time(delta_secs(&psinfo->pr_lwp.pr_start),
			    width);
		else
			print_time(delta_secs(&psinfo->pr_start), width);
		break;
	case F_TIME:
		if (Lflg) {
			cputime = psinfo->pr_lwp.pr_time.tv_sec;
			if (psinfo->pr_lwp.pr_time.tv_nsec > 500000000)
				cputime++;
		} else {
			cputime = psinfo->pr_time.tv_sec;
```

```
                        if (psinfo->pr_time.tv_nsec > 500000000)
                                cputime++;
                }
                print_time(cputime, width);
                break;
        case F_TTY:
                (void) printf("%-*s", width, ttyp);
                break;
        case F_ADDR:
                if (zombie_lwp)
                        (void) printf("%*s", width, "-");
                else if (Lflg)
                        (void) printf("%*lx", width,
                            (long)psinfo->pr_lwp.pr_addr);
                else
                        (void) printf("%*lx", width, (long)psinfo->pr_addr);
                break;
        case F_WCHAN:
                if (!zombie_lwp && psinfo->pr_lwp.pr_wchan)
                        (void) printf("%*lx", width,
                            (long)psinfo->pr_lwp.pr_wchan);
                else
                        (void) printf("%*.*s", width, width, "-");
                break;
        case F_FNAME:
                /*
                 * Print full width unless this is the last output format.
                 */
                if (zombie_lwp) {
                        if (f->next != NULL)
                                (void) printf("%-*s", width, "<defunct>");
                        else
                                (void) printf("%s", "<defunct>");
                        break;
                }
                wcnt = namencnt(psinfo->pr_fname, 16, width);
                if (f->next != NULL)
                        (void) printf("%-*.*s", width, wcnt, psinfo->pr_fname);
                else
                        (void) printf("%-.*s", wcnt, psinfo->pr_fname);
                break;
        case F_COMM:
                if (zombie_lwp) {
                        if (f->next != NULL)
                                (void) printf("%-*s", width, "<defunct>");
                        else
                                (void) printf("%s", "<defunct>");
                        break;
                }
                csave = strpbrk(psinfo->pr_psargs, " \t\r\v\f\n");
                if (csave) {
                        c = *csave;
                        *csave = '\0';
                }
                /* FALLTHROUGH */
        case F_ARGS:
                /*
                 * PRARGSZ == length of cmd arg string.
                 */
                if (zombie_lwp) {
                        (void) printf("%-*s", width, "<defunct>");
                        break;
                }
                psinfo->pr_psargs[PRARGSZ-1] = '\0';
                bytesleft = PRARGSZ;
                for (cp = psinfo->pr_psargs; *cp != '\0'; cp += length) {
```

```
                        length = mbtowc(&wchar, cp, MB_LEN_MAX);
                        if (length == 0)
                                break;
                        if (length < 0 || !iswprint(wchar)) {
                                if (length < 0)
                                        length = 1;
                                if (bytesleft <= length) {
                                        *cp = '\0';
                                        break;
                                }
                                /* omit the unprintable character */
                                (void) memmove(cp, cp+length, bytesleft-length);
                                length = 0;
                        }
                        bytesleft -= length;
                }
                wcnt = namencnt(psinfo->pr_psargs, PRARGSZ, width);
                /*
                 * Print full width unless this is the last format.
                 */
                if (f->next != NULL)
                        (void) printf("%-*.*s", width, wcnt,
                            psinfo->pr_psargs);
                else
                        (void) printf("%-.*s", wcnt,
                            psinfo->pr_psargs);
                if (f->fname == F_COMM && csave)
                        *csave = c;
                break;
        case F_TASKID:
                (void) printf("%*d", width, (int)psinfo->pr_taskid);
                break;
        case F_PROJID:
                (void) printf("%*d", width, (int)psinfo->pr_projid);
                break;
        case F_PROJECT:
                {
                        struct project cproj;
                        char proj_buf[PROJECT_BUFSZ];

                        if ((getprojbyid(psinfo->pr_projid, &cproj,
                            (void *)&proj_buf, PROJECT_BUFSZ)) == NULL) {
                                if (Wflg && snprintf(NULL, 0, "%d",
                                    ((int)psinfo->pr_projid)) > width)
                                        (void) printf("%.*d%c", width - 1,
                                            ((int)psinfo->pr_projid), '*');
                                else
                            (void *)&proj_buf, PROJECT_BUFSZ)) == NULL)
                                        (void) printf("%*d", width,
                                            (int)psinfo->pr_projid);
                        } else {
                                size_t nw;

                                if (cproj.pj_name != NULL)
                                        nw = mbstowcs(NULL, cproj.pj_name, 0);
                                if (cproj.pj_name == NULL)
                                        (void) printf("%*s ", width, "---");
                                else if (nw == (size_t)-1)
                                        (void) printf("%*s ", width, "ERROR");
                                else if (Wflg && nw > width)
                                        (void) wprintf(L"%.*s%c", width - 1,
                                            cproj.pj_name, '*');
                                else
                                        (void) wprintf(L"%*s", width,
                                            cproj.pj_name);
                                (void) printf("%*s", width,
                                    (cproj.pj_name != NULL) ?
                                    cproj.pj_name : "---");
                        }
                }
                break;
        case F_PSET:
                if (zombie_lwp || psinfo->pr_lwp.pr_bindpset == PS_NONE)
                        (void) printf("%*s", width, "-");
                else
                        (void) printf("%*d", width, psinfo->pr_lwp.pr_bindpset);
                break;
        case F_ZONEID:
                (void) printf("%*d", width, (int)psinfo->pr_zoneid);
                break;
        case F_ZONE:
                {
                        char zonename[ZONENAME_MAX];

                        if (getzonenamebyid(psinfo->pr_zoneid, zonename,
                            sizeof (zonename)) < 0) {
                                if (Wflg && snprintf(NULL, 0, "%d",
                                    ((int)psinfo->pr_zoneid)) > width)
                                        (void) printf("%.*d%c", width - 1,
                                            ((int)psinfo->pr_zoneid), '*');
                                else
                                        (void) printf("%*d", width,
                                            (int)psinfo->pr_zoneid);
                                    ((int)psinfo->pr_zoneid));
                        } else {
                                size_t nw;

                                nw = mbstowcs(NULL, zonename, 0);
                                if (nw == (size_t)-1)
                                        (void) printf("%*s ", width, "ERROR");
                                else if (Wflg && nw > width)
                                        (void) wprintf(L"%.*s%c", width - 1,
                                            zonename, '*');
                                else
                                        (void) wprintf(L"%*s", width, zonename);
                                (void) printf("%*s", width, zonename);
                        }
                }
                break;
        case F_CTID:
                if (psinfo->pr_contract == -1)
                        (void) printf("%*s", width, "-");
                else
                        (void) printf("%*ld", width, (long)psinfo->pr_contract);
                break;
        case F_LGRP:
                /* Display home lgroup */
                (void) printf("%*d", width, (int)psinfo->pr_lwp.pr_lgrp);
                break;

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

*______unchanged_portion_omitted_*

```
static void
przom(psinfo_t *psinfo)
{
        long    tm;
```

```
        struct passwd *pwd;
        char zonename[ZONENAME_MAX];

        /*
         * All fields before 'PID' are printed with a trailing space as a
         * spearator, rather than keeping track of which column is first.  All
         * other fields are printed with a leading space.
         */
        if (lflg) {     /* F S */
                if (!yflg)
                        (void) printf("%2x ", psinfo->pr_flag & 0377); /* F */
                (void) printf("%c ", psinfo->pr_lwp.pr_sname);  /* S */
        }
        if (Zflg) {
                if (getzonenamebyid(psinfo->pr_zoneid, zonename,
                    sizeof (zonename)) < 0) {
                        if (snprintf(NULL, 0, "%d",
                            ((int)psinfo->pr_zoneid)) > 7)
                                (void) printf(" %6.6d%c ",
                                    ((int)psinfo->pr_zoneid), '*');
                        else
                                (void) printf(" %7.7d ",
                                    ((int)psinfo->pr_zoneid));
                        (void) printf(" %7.7d ", ((int)psinfo->pr_zoneid));
                } else {
                        size_t nw;

                        nw = mbstowcs(NULL, zonename, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%8.8s ", "ERROR");
                        else if (nw > 8)
                                (void) wprintf(L"%7.7s%c ", zonename, '*');
                        else
                                (void) wprintf(L"%8.8s ", zonename);
                        (void) printf("%8.8s ", zonename);
                }
        }
        if (Hflg) {
                /* Display home lgroup */
                (void) printf(" %6d", (int)psinfo->pr_lwp.pr_lgrp); /* LGRP */
        }
        if (fflg) {
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL) {
                        size_t nw;

                        nw = mbstowcs(NULL, pwd->pw_name, 0);
                        if (nw == (size_t)-1)
                                (void) printf("%8.8s ", "ERROR");
                        else if (nw > 8)
                                (void) wprintf(L"%7.7s%c ", pwd->pw_name, '*');
                if ((pwd = getpwuid(psinfo->pr_euid)) != NULL)
                        (void) printf("%8.8s ", pwd->pw_name);
                        else
                                (void) wprintf(L"%8.8s ", pwd->pw_name);
                } else {
                        if (snprintf(NULL, 0, "%u",
                            (psinfo->pr_euid)) > 7)
                                (void) printf(" %6.6u%c ", psinfo->pr_euid,
                                    '*');
                        else
                                (void) printf(" %7.7u ", psinfo->pr_euid);
                }
        } else if (lflg) {
                if (snprintf(NULL, 0, "%u", (psinfo->pr_euid)) > 6)
                        (void) printf("%5.5u%c ", psinfo->pr_euid, '*');
                else
        } else if (lflg)
                        (void) printf("%6u ", psinfo->pr_euid);
        }

        (void) printf("%*d", pidwidth, (int)psinfo->pr_pid);    /* PID */
        if (lflg || fflg)
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_ppid);                      /* PPID */

        if (jflg) {
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_pgid);                      /* PGID */
                (void) printf(" %*d", pidwidth,
                    (int)psinfo->pr_sid);                       /* SID  */
        }

        if (Lflg)
                (void) printf(" %5d", 0);                       /* LWP */
        if (Pflg)
                (void) printf("   -");                          /* PSR */
        if (Lflg && fflg)
                (void) printf(" %5d", 0);                       /* NLWP */

        if (cflg) {
                (void) printf(" %4s", "-");     /* zombies have no class */
                (void) printf(" %3d", psinfo->pr_lwp.pr_pri);   /* PRI  */
        } else if (lflg || fflg) {
                (void) printf(" %3d", psinfo->pr_lwp.pr_cpu & 0377); /* C   */
                if (lflg)
                        (void) printf(" %3d %2s",
                            psinfo->pr_lwp.pr_oldpri, "-");     /* PRI NI */
        }
        if (lflg) {
                if (yflg)                               /* RSS SZ WCHAN */
                        (void) printf(" %5d %6d %8s", 0, 0, "-");
                else                                    /* ADDR SZ WCHAN */
                        (void) printf(" %8s %6d %8s", "-", 0, "-");
        }
        if (fflg) {
                int width = fname[F_STIME].width;
                (void) printf(" %*.*s", width, width, "-");     /* STIME */
        }
        (void) printf(" %-8.14s", "?");                         /* TTY */

        tm = psinfo->pr_time.tv_sec;
        if (psinfo->pr_time.tv_nsec > 500000000)
                tm++;
        (void) printf(" %4ld:%.2ld", tm / 60, tm % 60); /* TIME */
        (void) printf(" <defunct>\n");
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    5355 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/pwck/pwck.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 2007 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

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


#pragma ident   "%Z%%M% %I%     %E% SMI"

#include <sys/types.h>
#include <sys/param.h>
#include <sys/signal.h>
#include <sys/sysmacros.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <locale.h>
#include <errno.h>
#include <unistd.h>
#include <limits.h>

#define ERROR1  "Too many/few fields"
#define ERROR2  "Bad character(s) in logname"
#define ERROR2a "First char in logname not alphabetic"
#define ERROR2b "Logname field NULL"
#define ERROR2c "Logname contains no lower-case letters"
#define ERROR3  "Logname too long/short"
#define ERROR4  "Invalid UID"
#define ERROR5  "Invalid GID"
#define ERROR6  "Login directory not found"
#define ERROR6a "Login directory null"
#define ERROR7  "Optional shell file not found"

static int eflag, code = 0;
```

```
static int badc;
static int lc;
static char buf[512];
static void error(char *);

int
main(int argc, char **argv)
{
        int delim[512];
        char logbuf[512];
        FILE *fptr;
        struct stat obuf;
        uid_t uid;
        gid_t gid;
        int i, j, colons;
        char *pw_file;
        struct stat stat_buf;
        char *str, *lastc;

        (void) setlocale(LC_ALL, "");

#if !defined(TEXT_DOMAIN)       /* Should be defined by cc -D */
#define TEXT_DOMAIN "SYS_TEST"
#endif
        (void) textdomain(TEXT_DOMAIN);

        if (argc == 1)
                pw_file = "/etc/passwd";
        else
                pw_file = argv[1];

        if ((fptr = fopen(pw_file, "r")) == NULL) {
                (void) fprintf(stderr, gettext("cannot open %s\n"), pw_file);
                exit(1);
        }

        if (fstat(fileno(fptr), &stat_buf) < 0) {
                (void) fprintf(stderr, gettext("fstat failed for %s\n"),
                    pw_file);
                (void) fclose(fptr);
                exit(1);
        }

        if (stat_buf.st_size == 0) {
                (void) fprintf(stderr, gettext("file %s is empty\n"), pw_file);
                (void) fclose(fptr);
                exit(1);
        }

        while (fgets(buf, sizeof (buf), fptr) != NULL) {

                colons = 0;
                badc = 0;
                lc = 0;
                eflag = 0;

                /* Check that entry is not a nameservice redirection */

                if (buf[0] == '+' || buf[0] == '-')  {
                        /*
                         * Should set flag here to allow special case checking
                         * in the rest of the code,
                         * but for now, we'll just ignore this entry.
                         */
                        continue;
                }
```

```
		/* Check number of fields */

		for (i = 0; buf[i] != NULL; i++)
			if (buf[i] == ':') {
				delim[colons] = i;
				++colons;
			}

		if (colons != 6) {
			error(ERROR1);
			continue;
		}
		delim[6] = i - 1;
		delim[7] = NULL;

		/*
		 * Check the first char is alpha; the rest alphanumeric;
		 * and that the name does not consist solely of uppercase
		 * alpha chars
		 */
		if (buf[0] == ':')
			error(ERROR2b);
		else if (!isalpha(buf[0]))
			error(ERROR2a);

		for (i = 0; buf[i] != ':'; i++) {
			if (!isalnum(buf[i]) &&
			    buf[i] != '_' &&
			    buf[i] != '-' &&
			    buf[i] != '.')
				badc++;
			else if (islower(buf[i]))
				lc++;
		}
		if (lc == 0)
			error(ERROR2c);
		if (badc > 0)
			error(ERROR2);

		/* Check for valid number of characters in logname */

		if (i <= 0 || i > LOGNAME_MAX)
		if (i <= 0 || i > 8)
			error(ERROR3);

		/* Check that UID is numeric and <= MAXUID */

		errno = 0;
		str = &buf[delim[1] + 1];
		uid = strtol(str, &lastc, 10);
		if (lastc != str + (delim[2] - delim[1]) - 1 ||
		    uid > MAXUID || errno == ERANGE)
			error(ERROR4);

		/* Check that GID is numeric and <= MAXUID */

		errno = 0;
		str = &buf[delim[2] + 1];
		gid = strtol(str, &lastc, 10);
		if (lastc != str + (delim[3] - delim[2]) - 1 ||
		    gid > MAXUID || errno == ERANGE)
			error(ERROR5);

		/* Check initial working directory */
```

```
		for (j = 0, i = (delim[4] + 1); i < delim[5]; j++, i++)
			logbuf[j] = buf[i];
		logbuf[j] = '\0';

		if (logbuf[0] == NULL)
			error(ERROR6a);
		else if ((stat(logbuf, &obuf)) == -1)
			error(ERROR6);

		/* Check program to use as shell  */

		if ((buf[(delim[5] + 1)]) != '\n') {

			for (j = 0, i = (delim[5] + 1); i < delim[6]; j++, i++)
				logbuf[j] = buf[i];
			logbuf[j] = '\0';

			if (strcmp(logbuf, "*") == 0)   /* subsystem login */
				continue;

			if ((stat(logbuf, &obuf)) == -1)
				error(ERROR7);

			for (j = 0; j < 512; j++)
				logbuf[j] = NULL;
		}
	}
	(void) fclose(fptr);
	return (code);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
    4517 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
#
# 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 (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
#

include         $(SRC)/cmd/Makefile.cmd

# Note: Why SUBDIRS-common isn't sorted alphabetically
#
# The items under SGS are not independent of each other.
# They must be built in an order that ensures that
# all dependencies of an item have been built before the
# item itself.
#
SUBDIRS-common= libconv         \
                .WAIT           \
                libdl           \
                libelf          \
                liblddbg        \
                .WAIT           \
                libld           \
                libldmake       \
                libldstab       \
                librtld         \
                libcrle         \
                .WAIT           \
                0@0             \
                getloginx       \
                ld              \
                ldd             \
                lddstub         \
                rtld            \
                link_audit      \
                .WAIT           \
                librtld_db      \
                ldprof          \
                pvs             \
                crle            \
                ar              \
                dump            \
```

```
                elfdump         \
                elfedit         \
                elfwrap         \
                error           \
                gprof           \
                lari            \
                lex             \
                lorder          \
                m4              \
                mcs             \
                moe             \
                nm              \
                prof            \
                ranlib          \
                size            \
                symorder        \
                tsort           \
                unifdef         \
                yacc

SUBDIRS-i386=
SUBDIRS-sparc=  rtld.4.x

SUBDIRS=        $(SUBDIRS-common) $(SUBDIRS-$(MACH))

# Messaging support
#
POSUBDIRS=      m4              nm      tsort           yacc
POFILE=         sgs.po
POFILES=        $(POSUBDIRS:%=%/%.po)

MSGSUBDIRS=     ld              ldd             libld           liblddbg \
                libldstab       librtld         rtld            libelf \
                ldprof          libcrle         pvs             elfdump \
                elfedit         crle            moe             lari \
                librtld_db      elfwrap         ar

MSGDIR=         messages


all :=          TARGET= all
install :=      TARGET= install
clean :=        TARGET= clean
clobber :=      TARGET= clobber
delete :=       TARGET= delete
lint :=         TARGET= lint
_msg :=         TARGET= catalog
_msg_gettext := TARGET= catalog
_msg_sgsmsg :=  TARGET= catalog
chkmsg :=       TARGET= chkmsg


.KEEP_STATE:

.PARALLEL:      $(SUBDIRS)

all install:    native-add .WAIT $(SUBDIRS)

include         $(SRC)/cmd/Makefile.targ

# Messaging support
#
_msg: _msg_gettext _msg_sgsmsg

_msg_gettext: $(MSGDOMAIN)/$(POFILE)
```

```
_msg_sgsmsg: $(MSGDIR)

$(MSGDOMAIN)/$(POFILE): \
                $(MSGDOMAIN) $(POFILE)

$(POFILE):      $(POSUBDIRS)
                $(RM) $(POFILE)
                cat $(POFILES) > $(POFILE)

$(MSGDIR):      $(MSGSUBDIRS) FRC
                @ cd $@; pwd; $(MAKE) $(TARGET)

chkmsg:         libconv $(MSGSUBDIRS) FRC

check:          chkmsg

# built from lib/Makefile
install_lib:    FRC
                @ cd lex; pwd; $(MAKE) $@
                @ cd yacc; pwd; $(MAKE) $@

lint:           $(SUBDIRS)

delete \
clean clobber:  native-clobber .WAIT $(SUBDIRS) $(MSGDIR)

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


# Integration of ld and ld.so.1 in some developement cycles requires that both
# of these modules be built using the new ld.  This ‘native’ target allows us
# to build a local ld which will then be used to build the delivered version of
# itself and ld.so.1.  Once this new functionality appears in the standard ld
# this target can be disabled.

native-add:     native-proto FRC
                @ cd tools/$(MACH); pwd; $(MAKE) native
                @ cd libconv/$(MACH); pwd; $(MAKE)
                @ cd libelf/$(MACH); pwd; $(MAKE) native
                @ cd liblddbg/$(MACH); pwd; $(MAKE) native
                @ cd libldstab/$(MACH); pwd; $(MAKE) native
                @ cd libld/$(MACH); pwd; $(MAKE) native
                @ cd ld/$(MACH); pwd; $(MAKE) native

native-clobber:
                @ cd tools; pwd; $(MAKE) $(TARGET)
                $(RM) -r proto/$(MACH)

native-proto:
                -@mkdir -p proto/$(MACH)

FRC:

#
# Cross-reference customization: ignore the directories named by XRPRUNE,
# and tweak the file globs slightly.
#
XRPRUNE=        rtld.4.x packages abi
XRADD=          *.msg mapfile* llib-[a-z]*
XRDEL=          Makefile* kobj_*

#
# Establish a set of directories for xref to search.  As there are duplicates
# of things like headers, and only one file will be added to the xref database,
# we want xref to list the source file.
```

```
#
XRDIRS=         . \
                ../../common/elfcap \
                ../../head \
                ../../uts/common/krtld \
                ../../uts/common/sys \
                ../../uts/sparc/sys \
                ../../uts/sparc/krtld \
                ../../uts/intel/ia32/krtld \
                ../../uts/intel/amd64/krtld

xref:           FRC
                @ $(RM) cscope.*
                xref -p -x cscope
```

```
**********************************************************
     964 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
```

```
#
# 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 (c) 2013 Gary Mills
#
# Copyright (c) 1996 by Sun Microsystems, Inc.
# All rights reserved.

include		$(SRC)/cmd/sgs/Makefile.sub
```

```
**********************************************************
    1261 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/Makefile.com
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
```

```
#
# 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 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#

LIBRARY=        getloginx.a
VERS=           .1

OBJECTS=        getloginx.o

include         $(SRC)/lib/Makefile.lib

MAPFILES=       ../common/mapfile-vers
LDLIBS +=       -lc
BUILD.SO=       $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(LDLIBS)

SRCS=           $(OBJECTS:%.o=../common/%.c)

CLEANFILES +=   $(LINTOUT)
CLOBBERFILES += $(DYNLIB)

ROOTDYNLIB=     $(DYNLIB:%=$(ROOTLIBDIR)/%)
```

****
1123 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/Makefile.targ
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

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

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

all:            $(DYNLIB)

$(DYNLIB):      pics .WAIT

include         $(SRC)/lib/Makefile.targ

delete:
                $(RM) $(DYNLIB)

lint:           lintcheck
```

****
1177 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/amd64/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
#
# 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 (c) 2013 Gary Mills
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#

include		$(SRC)/cmd/sgs/getloginx/Makefile.com

ASFLAGS +=	-D__amd64 $(amd64_ASFLAGS)

.KEEP_STATE:

install:	all $(ROOTLIBDIR64)/$(DYNLIB)

include		$(SRC)/cmd/sgs/getloginx/Makefile.targ
include		../../Makefile.sub.64
```

****
    1195 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/common/getloginx.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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 (c) 2013 Gary Mills
 *
 *      Copyright (c) 1996 by Sun Microsystems, Inc.
 *
 * Truncate a long login name returned by getlogin()
 * for programs that copy the name to a nine-byte buffer.
 *
 * Use:
 *      LD_PRELOAD=getloginx.so.1 program args ...
 *
 */

extern char *getloginx(void);

char *
getlogin(void)
{
        return (getloginx());
}

/* */
```

****
1310 Mon Aug 12 18:24:53 2013
new/usr/src/cmd/sgs/getloginx/common/mapfile-vers
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
#
# 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
#

#
# 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

SYMBOL_VERSION ILLUMOS_0.1 {    # first release of getloginx.so.1
    global:
        getlogin;
    local:
        *;
};
```

```
**********************************************************
    1093 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/sgs/getloginx/i386/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
```

```
#
# 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 (c) 2013 Gary Mills
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

include		$(SRC)/cmd/sgs/getloginx/Makefile.com

.KEEP_STATE:

include		$(SRC)/cmd/sgs/getloginx/Makefile.targ

install:	all $(ROOTDYNLIB)
```

****
1093 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/sgs/getloginx/sparc/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
#
# 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 (c) 2013 Gary Mills
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

include         $(SRC)/cmd/sgs/getloginx/Makefile.com

.KEEP_STATE:

include         $(SRC)/cmd/sgs/getloginx/Makefile.targ

install:        all $(ROOTDYNLIB)
```

****
    1136 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/sgs/getloginx/sparcv9/Makefile
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
#
# 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 (c) 2013 Gary Mills
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

include		$(SRC)/cmd/sgs/getloginx/Makefile.com

.KEEP_STATE:

install:	all $(ROOTLIBDIR64)/$(DYNLIB)

include		$(SRC)/cmd/sgs/getloginx/Makefile.targ
include		../../Makefile.sub.64
```

```
**********************************************************
   18980 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/w/w.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*
 * University Copyright- Copyright (c) 1982, 1986, 1988
 * The Regents of the University of California
 * All Rights Reserved
 *
 * University Acknowledgment- Portions of this document are derived from
 * software developed by the University of California, Berkeley, and its
 * contributors.
 */

/*
 * This is the new w command which takes advantage of
 * the /proc interface to gain access to the information
 * of all the processes currently on the system.
 *
 * This program also implements 'uptime'.
 *
 * Maintenance note:
 *
 * Much of this code is replicated in whodo.c.  If you're
 * fixing bugs here, then you should probably fix 'em there too.
 */

#include <stdio.h>
#include <string.h>
#include <stdarg.h>
#include <stdlib.h>
#include <ctype.h>
#include <fcntl.h>
#include <time.h>
```

```
#include <errno.h>
#include <sys/types.h>
#include <utmpx.h>
#include <sys/stat.h>
#include <dirent.h>
#include <procfs.h>             /* /proc header file */
#include <locale.h>
#include <unistd.h>
#include <sys/loadavg.h>
#include <limits.h>
#include <priv_utils.h>

/*
 * Use the full lengths from utmpx for user and line.
 * utmpx defines wider fields for user and line.  For compatibility of output,
 * we are limiting these to the old maximums in utmp. Define UTMPX_NAMELEN
 * to use the full lengths.
 */
#ifndef UTMPX_NAMELEN
/* XXX - utmp - fix name length */
#define NMAX            (_POSIX_LOGIN_NAME_MAX - 1)
#define LMAX            12
#else   /* UTMPX_NAMELEN */
static struct utmpx dummy;
#define NMAX            (sizeof (dummy.ut_user))
#define LMAX            (sizeof (dummy.ut_line))
#endif /* UTMPX_NAMELEN */

/* Print minimum field widths. */
#define LOGIN_WIDTH     8
#define LINE_WIDTH      12

#define DIV60(t)        ((t+30)/60)     /* x/60 rounded */

#ifdef ERR
#undef ERR
#endif
#define ERR             (-1)

#define HSIZE           256             /* size of process hash table   */
#define PROCDIR         "/proc"
#define INITPROCESS     (pid_t)1        /* init process pid */
#define NONE            'n'             /* no state */
#define RUNNING         'r'             /* runnable process */
#define ZOMBIE          'z'             /* zombie process */
#define VISITED         'v'             /* marked node as visited */
#define PRINTF(a)       if (printf a < 0) { \
                perror((gettext("%s: printf failed"), prog)); \
                exit(1); }

struct uproc {
        pid_t   p_upid;                 /* process id */
        char    p_state;                /* numeric value of process state */
        dev_t   p_ttyd;                 /* controlling tty of process */
        time_t  p_time;                 /* seconds of user & system time */
        time_t  p_ctime;                /* seconds of child user & sys time */
        int     p_igintr;               /* 1 = ignores SIGQUIT and SIGINT */
        char    p_comm[PRARGSZ+1];      /* command */
        char    p_args[PRARGSZ+1];      /* command line arguments */
        struct uproc    *p_child,       /* first child pointer */
                        *p_sibling,     /* sibling pointer */
                        *p_pgrpl,       /* pgrp link */
                        *p_link;        /* hash table chain pointer */
};

/*
```

```
 *      define hash table for struct uproc
 *      Hash function uses process id
 *      and the size of the hash table(HSIZE)
 *      to determine process index into the table.
 */
static struct uproc     pr_htbl[HSIZE];

static struct   uproc   *findhash(pid_t);
static time_t   findidle(char *);
static void     clnarglist(char *);
static void     showtotals(struct uproc *);
static void     calctotals(struct uproc *);
static void     prttime(time_t, char *);
static void     prtat(time_t *time);
static void     checkampm(char *str);

static char     *prog;          /* pointer to invocation name */
static int      header = 1;     /* true if -h flag: don't print heading */
static int      lflag = 1;      /* set if -l flag; 0 for -s flag: short form */
static char     *sel_user;      /* login of particular user selected */
static char     firstchar;      /* first char of name of prog invoked as */
static int      login;          /* true if invoked as login shell */
static time_t   now;            /* current time of day */
static time_t   uptime;         /* time of last reboot & elapsed time since */
static int      nusers;         /* number of users logged in now */
static time_t   idle;           /* number of minutes user is idle */
static time_t   jobtime;        /* total cpu time visible */
static char     doing[520];     /* process attached to terminal */
static time_t   proctime;       /* cpu time of process in doing */
static pid_t    curpid, empty;
static int      add_times;      /* boolean: add the cpu times or not */

#if SIGQUIT > SIGINT
#define ACTSIZE SIGQUIT
#else
#define ACTSIZE SIGINT
#endif

int
main(int argc, char *argv[])
{
        struct utmpx    *ut;
        struct utmpx    *utmpbegin;
        struct utmpx    *utmpend;
        struct utmpx    *utp;
        struct uproc    *up, *parent, *pgrp;
        struct psinfo   info;
        struct sigaction actinfo[ACTSIZE];
        struct pstatus  statinfo;
        size_t          size;
        struct stat     sbuf;
        DIR             *dirp;
        struct  dirent  *dp;
        char            pname[64];
        char            *fname;
        int             procfd;
        char            *cp;
        int             i;
        int             days, hrs, mins;
        int             entries;
        double          loadavg[3];

        /*
         * This program needs the proc_owner privilege
         */
        (void) __init_suid_priv(PU_CLEARLIMITSET, PRIV_PROC_OWNER,
```

```
            (char *)NULL);

        (void) setlocale(LC_ALL, "");
#if !defined(TEXT_DOMAIN)
#define TEXT_DOMAIN "SYS_TEST"
#endif
        (void) textdomain(TEXT_DOMAIN);

        login = (argv[0][0] == '-');
        cp = strrchr(argv[0], '/');
        firstchar = login ? argv[0][1] : (cp == 0) ? argv[0][0] : cp[1];
        prog = argv[0];

        while (argc > 1) {
                if (argv[1][0] == '-') {
                        for (i = 1; argv[1][i]; i++) {
                                switch (argv[1][i]) {

                                case 'h':
                                        header = 0;
                                        break;

                                case 'l':
                                        lflag++;
                                        break;
                                case 's':
                                        lflag = 0;
                                        break;

                                case 'u':
                                case 'w':
                                        firstchar = argv[1][i];
                                        break;

                                default:
                                        (void) fprintf(stderr, gettext(
                                            "%s: bad flag %s\n"),
                                            prog, argv[1]);
                                        exit(1);
                                }
                        }
                } else {
                        if (!isalnum(argv[1][0]) || argc > 2) {
                                (void) fprintf(stderr, gettext(
                                    "usage: %s [ -hlsuw ] [ user ]\n"), prog);
                                exit(1);
                        } else
                                sel_user = argv[1];
                }
                argc--; argv++;
        }

        /*
         * read the UTMP_FILE (contains information about each logged in user)
         */
        if (stat(UTMPX_FILE, &sbuf) == ERR) {
                (void) fprintf(stderr, gettext("%s: stat error of %s: %s\n"),
                    prog, UTMPX_FILE, strerror(errno));
                exit(1);
        }
        entries = sbuf.st_size / sizeof (struct futmpx);
        size = sizeof (struct utmpx) * entries;
        if ((ut = malloc(size)) == NULL) {
                (void) fprintf(stderr, gettext("%s: malloc error of %s: %s\n"),
                    prog, UTMPX_FILE, strerror(errno));
                exit(1);
        }

        (void) utmpxname(UTMPX_FILE);

        utmpbegin = ut;
        utmpend = (struct utmpx *)((char *)utmpbegin + size);

        setutxent();
        while ((ut < utmpend) && ((utp = getutxent()) != NULL))
                (void) memcpy(ut++, utp, sizeof (*ut));
        endutxent();

        (void) time(&now);      /* get current time */

        if (header) {   /* print a header */
                prtat(&now);
                for (ut = utmpbegin; ut < utmpend; ut++) {
                        if (ut->ut_type == USER_PROCESS) {
                                if (!nonuser(*ut))
                                        nusers++;
                        } else if (ut->ut_type == BOOT_TIME) {
                                uptime = now - ut->ut_xtime;
                                uptime += 30;
                                days = uptime / (60*60*24);
                                uptime %= (60*60*24);
                                hrs = uptime / (60*60);
                                uptime %= (60*60);
                                mins = uptime / 60;

                                PRINTF((gettext("  up")));
                                if (days > 0)
                                        PRINTF((gettext(
                                            " %d day(s),"), days));
                                if (hrs > 0 && mins > 0) {
                                        PRINTF((" %2d:%02d,", hrs, mins));
                                } else {
                                        if (hrs > 0)
                                                PRINTF((gettext(
                                                    " %d hr(s),"), hrs));
                                        if (mins > 0)
                                                PRINTF((gettext(
                                                    " %d min(s),"), mins));
                                }
                        }
                }

                ut = utmpbegin; /* rewind utmp data */
                PRINTF((((nusers == 1) ?
                    gettext("  %d user") : gettext("  %d users")), nusers));
                /*
                 * Print 1, 5, and 15 minute load averages.
                 */
                (void) getloadavg(loadavg, 3);
                PRINTF((gettext(",  load average: %.2f, %.2f, %.2f\n"),
                    loadavg[LOADAVG_1MIN], loadavg[LOADAVG_5MIN],
                    loadavg[LOADAVG_15MIN]));

                if (firstchar == 'u')   /* uptime command */
                        exit(0);

                if (lflag) {
                        PRINTF((dcgettext(NULL, "User     tty           "
                            "login@  idle   JCPU   PCPU  what\n", LC_TIME)));
                } else {
                        PRINTF((dcgettext(NULL,
                            "User     tty           idle   what\n", LC_TIME)));
```

```
                }

                if (fflush(stdout) == EOF) {
                        perror((gettext("%s: fflush failed\n"), prog));
                        exit(1);
                }
        }

        /*
         * loop through /proc, reading info about each process
         * and build the parent/child tree
         */
        if (!(dirp = opendir(PROCDIR))) {
                (void) fprintf(stderr, gettext("%s: could not open %s: %s\n"),
                    prog, PROCDIR, strerror(errno));
                exit(1);
        }

        while ((dp = readdir(dirp)) != NULL) {
                if (dp->d_name[0] == '.')
                        continue;
retry:
                (void) sprintf(pname, "%s/%s/", PROCDIR, dp->d_name);
                fname = pname + strlen(pname);
                (void) strcpy(fname, "psinfo");
                if ((procfd = open(pname, O_RDONLY)) < 0)
                        continue;
                if (read(procfd, &info, sizeof (info)) != sizeof (info)) {
                        int err = errno;
                        (void) close(procfd);
                        if (err == EAGAIN)
                                goto retry;
                        if (err != ENOENT)
                                (void) fprintf(stderr, gettext(
                                    "%s: read() failed on %s: %s \n"),
                                    prog, pname, strerror(err));
                        continue;
                }
                (void) close(procfd);

                up = findhash(info.pr_pid);
                up->p_ttyd = info.pr_ttydev;
                up->p_state = (info.pr_nlwp == 0? ZOMBIE : RUNNING);
                up->p_time = 0;
                up->p_ctime = 0;
                up->p_igintr = 0;
                (void) strncpy(up->p_comm, info.pr_fname,
                    sizeof (info.pr_fname));
                up->p_args[0] = 0;

                if (up->p_state != NONE && up->p_state != ZOMBIE) {
                        (void) strcpy(fname, "status");

                        /* now we need the proc_owner privilege */
                        (void) __priv_bracket(PRIV_ON);

                        procfd = open(pname, O_RDONLY);

                        /* drop proc_owner privilege after open */
                        (void) __priv_bracket(PRIV_OFF);

                        if (procfd < 0)
                                continue;

                        if (read(procfd, &statinfo, sizeof (statinfo))
                            != sizeof (statinfo)) {
                                int err = errno;
                                (void) close(procfd);
                                if (err == EAGAIN)
                                        goto retry;
                                if (err != ENOENT)
                                        (void) fprintf(stderr, gettext(
                                            "%s: read() failed on %s: %s \n"),
                                            prog, pname, strerror(err));
                                continue;
                        }
                        (void) close(procfd);

                        up->p_time = statinfo.pr_utime.tv_sec +
                            statinfo.pr_stime.tv_sec;   /* seconds */
                        up->p_ctime = statinfo.pr_cutime.tv_sec +
                            statinfo.pr_cstime.tv_sec;

                        (void) strcpy(fname, "sigact");

                        /* now we need the proc_owner privilege */
                        (void) __priv_bracket(PRIV_ON);

                        procfd = open(pname, O_RDONLY);

                        /* drop proc_owner privilege after open */
                        (void) __priv_bracket(PRIV_OFF);

                        if (procfd < 0)
                                continue;

                        if (read(procfd, actinfo, sizeof (actinfo))
                            != sizeof (actinfo)) {
                                int err = errno;
                                (void) close(procfd);
                                if (err == EAGAIN)
                                        goto retry;
                                if (err != ENOENT)
                                        (void) fprintf(stderr, gettext(
                                            "%s: read() failed on %s: %s \n"),
                                            prog, pname, strerror(err));
                                continue;
                        }
                        (void) close(procfd);

                        up->p_igintr =
                            actinfo[SIGINT-1].sa_handler == SIG_IGN &&
                            actinfo[SIGQUIT-1].sa_handler == SIG_IGN;

                        /*
                         * Process args.
                         */
                        up->p_args[0] = 0;
                        clnarglist(info.pr_psargs);
                        (void) strcat(up->p_args, info.pr_psargs);
                        if (up->p_args[0] == 0 ||
                            up->p_args[0] == ’-’ && up->p_args[1] <= ’ ’ ||
                            up->p_args[0] == ’?’) {
                                (void) strcat(up->p_args, " (");
                                (void) strcat(up->p_args, up->p_comm);
                                (void) strcat(up->p_args, ")");
                        }
                }

                /*
                 * link pgrp together in case parents go away
                 * Pgrp chain is a single linked list originating
```

```
                 * from the pgrp leader to its group member.
                 */
                if (info.pr_pgid != info.pr_pid) {      /* not pgrp leader */
                        pgrp = findhash(info.pr_pgid);
                        up->p_pgrpl = pgrp->p_pgrpl;
                        pgrp->p_pgrpl = up;
                }
                parent = findhash(info.pr_ppid);

                /* if this is the new member, link it in */
                if (parent->p_upid != INITPROCESS) {
                        if (parent->p_child) {
                                up->p_sibling = parent->p_child;
                                up->p_child = 0;
                        }
                        parent->p_child = up;
                }
        }

        /* revert to non-privileged user after opening */
        (void) __priv_relinquish();

        (void) closedir(dirp);
        (void) time(&now);      /* get current time */

        /*
         * loop through utmpx file, printing process info
         * about each logged in user
         */
        for (ut = utmpbegin; ut < utmpend; ut++) {
                if (ut->ut_type != USER_PROCESS)
                        continue;
                if (sel_user && strncmp(ut->ut_name, sel_user, NMAX) != 0)
                        continue;       /* we’re looking for somebody else */

                /* print login name of the user */
                PRINTF(("%-*.*s ", LOGIN_WIDTH, NMAX, ut->ut_name));
                PRINTF(("%-*.*s ", NMAX, NMAX, ut->ut_name));

                /* print tty user is on */
                if (lflag) {
                        PRINTF(("%-*.*s", LINE_WIDTH, LMAX, ut->ut_line));
                        PRINTF(("%-*.*s", LMAX, LMAX, ut->ut_line));
                } else {
                        if (ut->ut_line[0] == ’p’ && ut->ut_line[1] == ’t’ &&
                            ut->ut_line[2] == ’s’ && ut->ut_line[3] == ’/’) {
                                PRINTF(("%-*.3s", LMAX, &ut->ut_line[4]));
                        } else {
                                PRINTF(("%-*.*s", LINE_WIDTH, LMAX,
                                    ut->ut_line));
                                PRINTF(("%-*.*s", LMAX, LMAX, ut->ut_line));
                        }
                }

                /* print when the user logged in */
                if (lflag) {
                        time_t tim = ut->ut_xtime;
                        prtat(&tim);
                }

                /* print idle time */
                idle = findidle(ut->ut_line);
                if (idle >= 36 * 60) {
                        PRINTF((dcgettext(NULL, "%2ddays ", LC_TIME),
                            (idle + 12 * 60) / (24 * 60)));
                } else
```

```
                        prttime(idle, " ");
                showtotals(findhash(ut->ut_pid));
        }
        if (fclose(stdout) == EOF) {
                perror((gettext("%s: fclose failed"), prog));
                exit(1);
        }
        return (0);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
   11096 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/wall/wall.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */



/*
 * Copyright 1988-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2012 Joyent, Inc. All rights reserved.
 *
 * Copyright (c) 2013 Gary Mills
 */

#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <grp.h>
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
#include <ctype.h>
#include <sys/stat.h>
#include <utmpx.h>
#include <sys/utsname.h>
#include <dirent.h>
#include <pwd.h>
#include <fcntl.h>
#include <time.h>
#include <errno.h>
#include <locale.h>
#include <syslog.h>
#include <sys/wait.h>
#include <limits.h>
#include <libzonecfg.h>
#include <zone.h>
#include <sys/contract/process.h>
```

```
#include <libcontract.h>
#include <sys/ctfs.h>

/*
 * Use the full lengths from utmpx for user and line.
 * utmpx defines wider fields for user and line.  For compatibility of output,
 * we are limiting these to the old maximums in utmp. Define UTMPX_NAMELEN
 * to use the full lengths.
 */
#define NMAX    (sizeof (((struct utmpx *)0)->ut_user))
#define LMAX    (sizeof (((struct utmpx *)0)->ut_line))
#ifndef UTMPX_NAMELEN
/* XXX - utmp -fix name length */
#define NMAX    (_POSIX_LOGIN_NAME_MAX - 1)
#define LMAX    12
#else /* UTMPX_NAMELEN */
#define NMAX    (sizeof (((struct utmpx *)0)->ut_user)
#define LMAX    (sizeof (((struct utmpx *)0)->ut_line)
#endif /* UTMPX_NAMELEN */

static char     mesg[3000];
static char     *infile;
static int      gflag;
static struct   group *pgrp;
static char     *grpname;
static char     line[MAXNAMLEN+1] = "???";
static char     systm[MAXNAMLEN+1];
static time_t   tloc;
static struct   utsname utsn;
static char     who[NMAX+1]     = "???";
static char     who[9]  = "???";
static char     time_buf[50];
#define DATE_FMT        "%a %b %e %H:%M:%S"

static void sendmes(struct utmpx *, zoneid_t);
static void sendmes_tozone(zoneid_t, int);
static int chkgrp(char *);
static char *copy_str_till(char *, char *, char, int);

static int init_template(void);
int contract_abandon_id(ctid_t);

int
main(int argc, char *argv[])
{
        FILE    *f;
        char    *ptr, *start;
        struct  passwd *pwd;
        char    *term_name;
        int     c;
        int     aflag = 0;
        int     errflg = 0;
        int zflg = 0;
        int Zflg = 0;

        char *zonename = NULL;
        zoneid_t *zoneidlist = NULL;
        uint_t nzids_saved, nzids = 0;

        (void) setlocale(LC_ALL, "");

        while ((c = getopt(argc, argv, "g:az:Z")) != EOF)
                switch (c) {
                case 'a':
                        aflag++;
                        break;
                case 'g':
                        if (gflag) {
                                (void) fprintf(stderr,
                                    "Only one group allowed\n");
                                return (1);
                        }
                        if ((pgrp = getgrnam(grpname = optarg)) == NULL) {
                                (void) fprintf(stderr, "Unknown group %s\n",
                                    grpname);
                                return (1);
                        }
                        gflag++;
                        break;
                case 'z':
                        zflg++;
                        zonename = optarg;
                        if (getzoneidbyname(zonename) == -1) {
                                (void) fprintf(stderr, "Specified zone %s "
                                    "is invalid", zonename);
                                return (1);
                        }
                        break;
                case 'Z':
                        Zflg++;
                        break;
                case '?':
                        errflg++;
                        break;
                }

        if (errflg) {
                (void) fprintf(stderr,
                    "Usage: wall [-a] [-g group] [-z zone] [-Z] [files...]\n");
                return (1);
        }

        if (zflg && Zflg) {
                (void) fprintf(stderr, "Cannot use -z with -Z\n");
                return (1);
        }

        if (optind < argc)
                infile = argv[optind];

        if (uname(&utsn) == -1) {
                (void) fprintf(stderr, "wall: uname() failed, %s\n",
                    strerror(errno));
                return (2);
        }
        (void) strcpy(systm, utsn.nodename);

        /*
         * Get the name of the terminal wall is running from.
         */

        if ((term_name = ttyname(fileno(stderr))) != NULL) {
                /*
                 * skip the leading "/dev/" in term_name
                 */
                (void) strncpy(line, &term_name[5], sizeof (line) - 1);
        }

        if (who[0] == '?') {
                if (pwd = getpwuid(getuid()))
                        (void) strncpy(&who[0], pwd->pw_name, sizeof (who));
        }
```

```
        f = stdin;
        if (infile) {
                f = fopen(infile, "r");
                if (f == NULL) {
                        (void) fprintf(stderr, "Cannot open %s\n", infile);
                        return (1);
                }
        }

        start = &mesg[0];
        ptr = start;
        while ((ptr - start) < 3000) {
                size_t n;

                if (fgets(ptr, &mesg[sizeof (mesg)] - ptr, f) == NULL)
                        break;
                if ((n = strlen(ptr)) == 0)
                        break;
                ptr += n;
        }
        (void) fclose(f);

        /*
         * If the request is from the rwall daemon then use the caller's
         * name and host.  We determine this if all of the following is true:
         *      1) First 5 characters are "From "
         *      2) Next non-white characters are of the form "name@host:"
         */
        if (strcmp(line, "???") == 0) {
                char rwho[MAXNAMLEN+1];
                char rsystm[MAXNAMLEN+1];
                char *cp;

                if (strncmp(mesg, "From ", 5) == 0) {
                        cp = &mesg[5];
                        cp = copy_str_till(rwho, cp, '@', MAXNAMLEN + 1);
                        if (rwho[0] != '\0') {
                                cp = copy_str_till(rsystm, ++cp, ':',
                                    MAXNAMLEN + 1);
                                if (rsystm[0] != '\0') {
                                        (void) strcpy(systm, rsystm);
                                        (void) strncpy(rwho, who,
                                            sizeof (who));
                                        (void) strncpy(rwho, who, 9);
                                        (void) strcpy(line, "rpc.rwalld");
                                }
                        }
                }
        }
        (void) time(&tloc);
        (void) strftime(time_buf, sizeof (time_buf),
            DATE_FMT, localtime(&tloc));

        if (zflg != 0) {
                if ((zoneidlist =
                    malloc(sizeof (zoneid_t))) == NULL ||
                    (*zoneidlist = getzoneidbyname(zonename)) == -1)
                        return (errno);
                nzids = 1;
        } else if (Zflg != 0) {
                if (zone_list(NULL, &nzids) != 0)
                        return (errno);
again:
                nzids *= 2;
                if ((zoneidlist = malloc(nzids * sizeof (zoneid_t))) == NULL)
```

```
                        exit(errno);
                nzids_saved = nzids;
                if (zone_list(zoneidlist, &nzids) != 0) {
                        (void) free(zoneidlist);
                        return (errno);
                }
                if (nzids > nzids_saved) {
                        free(zoneidlist);
                        goto again;
                }
        }
        if (zflg || Zflg) {
                for (; nzids > 0; --nzids)
                        sendmes_tozone(zoneidlist[nzids-1], aflag);
                free(zoneidlist);
        } else
                sendmes_tozone(getzoneid(), aflag);

        return (0);
}
```

______***unchanged_portion_omitted_***

```
/*
 * Note to future maintainers: with the change of wall to use the
 * getutxent() API, the forked children (created by this function)
 * must call _exit as opposed to exit. This is necessary to avoid
 * unwanted fflushing of getutxent's stdio stream (caused by atexit
 * processing).
 */
static void
sendmes(struct utmpx *p, zoneid_t zid)
{
        int i;
        char *s;
        static char device[LMAX + 6];
        char *bp;
        int ibp;
        FILE *f;
        int fd, tmpl_fd;
        boolean_t zoneenter = B_FALSE;

        if (zid != getzoneid()) {
                zoneenter = B_TRUE;
                tmpl_fd = init_template();
                if (tmpl_fd == -1) {
                        (void) fprintf(stderr, "Could not initialize "
                            "process contract");
                        return;
                }
        }

        while ((i = (int)fork()) == -1) {
                (void) alarm(60);
                (void) wait((int *)0);
                (void) alarm(0);
        }

        if (i)
                return;

        if (zoneenter && zone_enter(zid) == -1) {
                char zonename[ZONENAME_MAX];
                (void) getzonenamebyid(zid, zonename, ZONENAME_MAX);
                (void) fprintf(stderr, "Could not enter zone "
                    "%s\n", zonename);
        }
```

```
        if (zoneenter)
                (void) ct_tmpl_clear(tmpl_fd);

        if (gflag)
                if (!chkgrp(p->ut_user))
                        _exit(0);

        (void) signal(SIGHUP, SIG_IGN);
        (void) alarm(60);
        s = &device[0];
        (void) snprintf(s, sizeof (device), "/dev/%.*s", LMAX, p->ut_line);

        /* check if the device is really a tty */
        if ((fd = open(s, O_WRONLY|O_NOCTTY|O_NONBLOCK)) == -1) {
                (void) fprintf(stderr, "Cannot send to %.*s on %s\n",
                    NMAX, p->ut_user, s);
                perror("open");
                (void) fflush(stderr);
                _exit(1);
        } else {
                if (!isatty(fd)) {
                        (void) fprintf(stderr,
                            "Cannot send to device %.*s %s\n",
                            LMAX, p->ut_line,
                            "because it's not a tty");
                        openlog("wall", 0, LOG_AUTH);
                        syslog(LOG_CRIT, "%.*s in utmpx is not a tty\n",
                            LMAX, p->ut_line);
                        closelog();
                        (void) fflush(stderr);
                        _exit(1);
                }
        }
#ifdef DEBUG
        (void) close(fd);
        f = fopen("wall.debug", "a");
#else
        f = fdopen(fd, "w");
#endif
        if (f == NULL) {
                (void) fprintf(stderr, "Cannot send to %-.*s on %s\n",
                    NMAX, &p->ut_user[0], s);
                perror("open");
                (void) fflush(stderr);
                _exit(1);
        }
        (void) fprintf(f,
            "\07\07\07Broadcast Message from %s (%s) on %s %19.19s",
            who, line, systm, time_buf);
        if (gflag)
                (void) fprintf(f, " to group %s", grpname);
        (void) fprintf(f, "...\n");
#ifdef DEBUG
        (void) fprintf(f, "DEBUG: To %.*s on %s\n", NMAX, p->ut_user, s);
        (void) fprintf(f, "DEBUG: To %.8s on %s\n", p->ut_user, s);
#endif
        i = strlen(mesg);
        for (bp = mesg; --i >= 0; bp++) {
                ibp = (unsigned int)((unsigned char) *bp);
                if (*bp == '\n')
                        (void) putc('\r', f);
                if (isprint(ibp) || *bp == '\r' || *bp == '\013' ||
                    *bp == ' ' || *bp == '\t' || *bp == '\n' || *bp == '\007') {
                        (void) putc(*bp, f);
                } else {
                        if (!isascii(*bp)) {
                                (void) fputs("M-", f);
                                *bp = toascii(*bp);
                        }
                        if (iscntrl(*bp)) {
                                (void) putc('^', f);
                                (void) putc(*bp + 0100, f);
                        }
                        else
                                (void) putc(*bp, f);
                }

                if (*bp == '\n')
                        (void) fflush(f);

                if (ferror(f) || feof(f)) {
                        (void) printf("\n\007Write failed\n");
                        (void) fflush(stdout);
                        _exit(1);
                }
        }
        (void) fclose(f);
        (void) close(fd);
        _exit(0);
}


static int
chkgrp(char *name)
{
        int i;
        char user[NMAX + 1];
        char *p;

        (void) strlcpy(user, name, sizeof (user));
        for (i = 0; pgrp->gr_mem[i] && pgrp->gr_mem[i][0]; i++) {
                if (strcmp(user, pgrp->gr_mem[i]) == 0)
                for (p = name; *p && *p != ' '; p++)
                ;
                *p = 0;
                if (strncmp(name, pgrp->gr_mem[i], 8) == 0)
                        return (1);
        }

        return (0);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
   20484 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/who/who.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */


/*
 * Copyright (c) 2013 Gary Mills
 *
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

/*
 *      This program analyzes information found in /var/adm/utmpx
 *
 *      Additionally information is gathered from /etc/inittab
 *      if requested.
 *
 *
 *      Syntax:
 *
 *              who am i        Displays info on yourself
 *
 *              who -a          Displays information about All
 *                              entries in /var/adm/utmpx
 *
 *              who -b          Displays info on last boot
 *
 *              who -d          Displays info on DEAD PROCESSES
 *
 *              who -H          Displays HEADERS for output
 *
 *              who -l          Displays info on LOGIN entries
 *
 *              who -m          Same as who am i
 *
 *              who -p          Displays info on PROCESSES spawned by init
 *
 *              who -q          Displays short information on
```

```
 *                              current users who LOGGED ON
 *
 *              who -r          Displays info of current run-level
 *
 *              who -s          Displays requested info in SHORT form
 *
 *              who -t          Displays info on TIME changes
 *
 *              who -T          Displays writeability of each user
 *                              (+ writeable, - non-writeable, ? hung)
 *
 *              who -u          Displays LONG info on users
 *                              who have LOGGED ON
 */

#define         DATE_FMT        "%b %e %H:%M"

/*
 *  %b  Abbreviated month name
 *  %e  Day of month
 *  %H  hour (24-hour clock)
 *  %M  minute
 */
#include        <errno.h>
#include        <fcntl.h>
#include        <stdio.h>
#include        <string.h>
#include        <sys/types.h>
#include        <unistd.h>
#include        <stdlib.h>
#include        <sys/stat.h>
#include        <time.h>
#include        <utmpx.h>
#include        <locale.h>
#include        <pwd.h>
#include        <limits.h>

static void process(void);
static void ck_file(char *);
static void dump(void);

static struct   utmpx *utmpp;   /* pointer for getutxent()      */

/*
 * Use the full lengths from utmpx for user and line.
 * utmpx defines wider fields for user and line.  For compatibility of output,
 * we are limiting these to the old maximums in utmp. Define UTMPX_NAMELEN
 * to use the full lengths.
 */
#ifndef UTMPX_NAMELEN
/* XXX - utmp - fix name length */
#define NMAX    (_POSIX_LOGIN_NAME_MAX - 1)
#define LMAX    12
#else /* UTMPX_NAMELEN */
#define NMAX    (sizeof (utmpp->ut_user))
#define LMAX    (sizeof (utmpp->ut_line))
#endif

/* Print minimum field widths. */
#define LOGIN_WIDTH     8
#define LINE_WIDTH      12

static char     comment[BUFSIZ]; /* holds inittab comment       */
static char     errmsg[BUFSIZ]; /* used in snprintf for errors  */
static int      fildes;         /* file descriptor for inittab  */
static int      Hopt = 0;       /* 1 = who -H                   */
static char     *inittab;       /* ptr to inittab contents      */
static char     *iinit;         /* index into inittab           */
static int      justme = 0;     /* 1 = who am i                 */
static struct   tm *lptr;       /* holds user login time        */
static char     *myname;        /* pointer to invoker's name    */
static char     *mytty;         /* holds device user is on      */
static char     nameval[sizeof (utmpp->ut_user) + 1]; /*  invoker's name */
static int      number = 8;     /* number of users per -q line  */
static int      optcnt = 0;     /* keeps count of options       */
static char     outbuf[BUFSIZ]; /* buffer for output            */
static char     *program;       /* holds name of this program   */
#ifdef  XPG4
static int      aopt = 0;       /* 1 = who -a                   */
static int      dopt = 0;       /* 1 = who -d                   */
#endif  /* XPG4 */
static int      qopt = 0;       /* 1 = who -q                   */
static int      sopt = 0;       /* 1 = who -s                   */
static struct   stat stbuf;     /* area for stat buffer         */
static struct   stat *stbufp;   /* ptr to structure             */
static int      terse = 1;      /* 1 = print terse msgs         */
static int      Topt = 0;       /* 1 = who -T                   */
static time_t   timnow;         /* holds current time           */
static int      totlusrs = 0;   /* cntr for users on system     */
static int      uopt = 0;       /* 1 = who -u                   */
static char     user[sizeof (utmpp->ut_user) + 1]; /* holds user name */
static int      validtype[UTMAXTYPE+1]; /* holds valid types    */
static int      wrap;           /* flag to indicate wrap        */
static char     time_buf[128];  /* holds date and time string   */
static char     *end;           /* used in strtol for end pointer */

int
main(int argc, char **argv)
{
        int     goerr = 0;      /* non-zero indicates cmd error */
        int     i;
        int     optsw;          /* switch for while of getopt() */

        (void) setlocale(LC_ALL, "");

#if     !defined(TEXT_DOMAIN)   /* Should be defined by cc -D */
#define TEXT_DOMAIN "SYS_TEST"  /* Use this only if it weren't */
#endif
        (void) textdomain(TEXT_DOMAIN);

        validtype[USER_PROCESS] = 1;
        validtype[EMPTY] = 0;
        stbufp = &stbuf;

        /*
         *      Strip off path name of this command
         */
        for (i = strlen(argv[0]); i >= 0 && argv[0][i] != '/'; --i)
                ;
        for (i = strlen(argv[0]); i >= 0 && argv[0][i] != '/'; --i);
        if (i >= 0)
                argv[0] += i+1;
        program = argv[0];

        /*
         *      Buffer stdout for speed
         */
        setbuf(stdout, outbuf);

        /*
         *      Retrieve options specified on command line
         *      XCU4 - add -m option
```

```
         */
        while ((optsw = getopt(argc, argv, "abdHlmn:pqrstTu")) != EOF) {
                optcnt++;
                switch (optsw) {

                case 'a':
                        optcnt += 7;
                        validtype[BOOT_TIME] = 1;
                        validtype[DEAD_PROCESS] = 1;
                        validtype[LOGIN_PROCESS] = 1;
                        validtype[INIT_PROCESS] = 1;
                        validtype[RUN_LVL] = 1;
                        validtype[OLD_TIME] = 1;
                        validtype[NEW_TIME] = 1;
                        validtype[USER_PROCESS] = 1;
#ifdef  XPG4
                        aopt = 1;
#endif  /* XPG4 */
                        uopt = 1;
                        Topt = 1;
                        if (!sopt) terse = 0;
                        break;

                case 'b':
                        validtype[BOOT_TIME] = 1;
                        if (!uopt) validtype[USER_PROCESS] = 0;
                        break;

                case 'd':
                        validtype[DEAD_PROCESS] = 1;
                        if (!uopt) validtype[USER_PROCESS] = 0;
#ifdef  XPG4
                        dopt = 1;
#endif  /* XPG4 */
                        break;

                case 'H':
                        optcnt--; /* Don't count Header */
                        Hopt = 1;
                        break;

                case 'l':
                        validtype[LOGIN_PROCESS] = 1;
                        if (!uopt) validtype[USER_PROCESS] = 0;
                        terse = 0;
                        break;
                case 'm':               /* New XCU4 option */
                        justme = 1;
                        break;

                case 'n':
                        errno = 0;
                        number = strtol(optarg, &end, 10);
                        if (errno != 0 || *end != '\0') {
                                (void) fprintf(stderr, gettext(
                                    "%s: Invalid numeric argument\n"),
                                    program);
                                exit(1);
                        }
                        if (number < 1) {
                                (void) fprintf(stderr, gettext(
                                    "%s: Number of users per line must "
                                    "be at least 1\n"), program);
                                exit(1);
                        }
                        break;
                case 'p':
                        validtype[INIT_PROCESS] = 1;
                        if (!uopt) validtype[USER_PROCESS] = 0;
                        break;

                case 'q':
                        qopt = 1;
                        break;

                case 'r':
                        validtype[RUN_LVL] = 1;
                        terse = 0;
                        if (!uopt) validtype[USER_PROCESS] = 0;
                        break;

                case 's':
                        sopt = 1;
                        terse = 1;
                        break;

                case 't':
                        validtype[OLD_TIME] = 1;
                        validtype[NEW_TIME] = 1;
                        if (!uopt) validtype[USER_PROCESS] = 0;
                        break;

                case 'T':
                        Topt = 1;
#ifdef  XPG4
                        terse = 1;      /* XPG4 requires -T */
#else   /* XPG4 */
                        terse = 0;
#endif  /* XPG4 */
                        break;

                case 'u':
                        uopt = 1;
                        validtype[USER_PROCESS] = 1;
                        if (!sopt) terse = 0;
                        break;

                case '?':
                        goerr++;
                        break;
                default:
                        break;
                }
        }
#ifdef  XPG4
        /*
         * XCU4 changes - check for illegal sopt, Topt & aopt combination
         */
        if (sopt == 1) {
                terse = 1;
                if (Topt == 1 || aopt == 1)
                goerr++;
        }
#endif  /* XPG4 */

        if (goerr > 0) {
#ifdef  XPG4
                /*
                 * XCU4 - slightly different usage with -s -a & -T
                 */
                (void) fprintf(stderr, gettext("\nUsage:\t%s"), program);
```

```
                (void) fprintf(stderr,
                    gettext(" -s [-bdHlmpqrtu] [utmpx_like_file]\n"));

                (void) fprintf(stderr, gettext(
                    "\t%s [-abdHlmpqrtTu] [utmpx_like_file]\n"), program);
#else   /* XPG4 */
                (void) fprintf(stderr, gettext(
                    "\nUsage:\t%s [-abdHlmpqrstTu] [utmpx_like_file]\n"),
                    program);
#endif  /* XPG4 */
                (void) fprintf(stderr,
                    gettext("\t%s -q [-n x] [utmpx_like_file]\n"), program);
                (void) fprintf(stderr, gettext("\t%s [am i]\n"), program);
                /*
                 * XCU4 changes - be explicit with "am i" options
                 */
                (void) fprintf(stderr, gettext("\t%s [am I]\n"), program);
                (void) fprintf(stderr, gettext(
                    "a\tall (bdlprtu options)\n"));
                (void) fprintf(stderr, gettext("b\tboot time\n"));
                (void) fprintf(stderr, gettext("d\tdead processes\n"));
                (void) fprintf(stderr, gettext("H\tprint header\n"));
                (void) fprintf(stderr, gettext("l\tlogin processes\n"));
                (void) fprintf(stderr, gettext(
                    "n #\tspecify number of users per line for -q\n"));
                (void) fprintf(stderr,
                    gettext("p\tprocesses other than getty or users\n"));
                (void) fprintf(stderr, gettext("q\tquick %s\n"), program);
                (void) fprintf(stderr, gettext("r\trun level\n"));
                (void) fprintf(stderr, gettext(
                "s\tshort form of %s (no time since last output or pid)\n"),
                    program);
                (void) fprintf(stderr, gettext("t\ttime changes\n"));
                (void) fprintf(stderr, gettext(
                    "T\tstatus of tty (+ writable, - not writable, "
                    "? hung)\n"));
                (void) fprintf(stderr, gettext("u\tuseful information\n"));
                (void) fprintf(stderr,
                    gettext("m\tinformation only about current terminal\n"));
                (void) fprintf(stderr, gettext(
                    "am i\tinformation about current terminal "
                    "(same as -m)\n"));
                (void) fprintf(stderr, gettext(
                    "am I\tinformation about current terminal "
                    "(same as -m)\n"));
                exit(1);
        }

        /*
         * XCU4: If -q option ignore all other options
         */
        if (qopt == 1) {
                Hopt = 0;
                sopt = 0;
                Topt = 0;
                uopt = 0;
                justme = 0;
                validtype[ACCOUNTING] = 0;
                validtype[BOOT_TIME] = 0;
                validtype[DEAD_PROCESS] = 0;
                validtype[LOGIN_PROCESS] = 0;
                validtype[INIT_PROCESS] = 0;
                validtype[RUN_LVL] = 0;
                validtype[OLD_TIME] = 0;
                validtype[NEW_TIME] = 0;
                validtype[USER_PROCESS] = 1;
        }

        if (argc == optind + 1) {
                optcnt++;
                ck_file(argv[optind]);
                (void) utmpxname(argv[optind]);
        }

        /*
         *      Test for 'who am i' or 'who am I'
         *      XCU4 - check if justme was already set by -m option
         */
        if (justme == 1 || (argc == 3 && strcmp(argv[1], "am") == 0 &&
            ((argv[2][0] == 'i' || argv[2][0] == 'I') &&
            argv[2][1] == '\0'))) {
                justme = 1;
                myname = nameval;
                (void) cuserid(myname);
                if ((mytty = ttyname(fileno(stdin))) == NULL &&
                    (mytty = ttyname(fileno(stdout))) == NULL &&
                    (mytty = ttyname(fileno(stderr))) == NULL) {
                        (void) fprintf(stderr, gettext(
                        "Must be attached to terminal for 'am I' option\n"));
                        (void) fflush(stderr);
                        exit(1);
                } else
                        mytty += 5; /* bump past "/dev/" */
        }

        if (!terse) {
                if (Hopt)
                        (void) printf(gettext(
        "NAME       LINE         TIME          IDLE    PID  COMMENTS\n"));

                timnow = time(0);

                if ((fildes = open("/etc/inittab",
                    O_NONBLOCK|O_RDONLY)) == -1) {
                        (void) snprintf(errmsg, sizeof (errmsg),
                            gettext("%s: Cannot open /etc/inittab"), program);
                        perror(errmsg);
                        exit(errno);
                }

                if (fstat(fildes, stbufp) == -1) {
                        (void) snprintf(errmsg, sizeof (errmsg),
                            gettext("%s: Cannot stat /etc/inittab"), program);
                        perror(errmsg);
                        exit(errno);
                }

                if ((inittab = malloc(stbufp->st_size + 1)) == NULL) {
                        (void) snprintf(errmsg, sizeof (errmsg),
                            gettext("%s: Cannot allocate %ld bytes"),
                            program, stbufp->st_size);
                        perror(errmsg);
                        exit(errno);
                }

                if (read(fildes, inittab, stbufp->st_size)
                    != stbufp->st_size) {
                        (void) snprintf(errmsg, sizeof (errmsg),
                            gettext("%s: Error reading /etc/inittab"),
                            program);
                        perror(errmsg);
                        exit(errno);
```

```
                }

                inittab[stbufp->st_size] = '\0';
                iinit = inittab;
        } else {
                if (Hopt) {
#ifdef  XPG4
                        if (dopt) {
                                (void) printf(gettext(
                        "NAME       LINE         TIME           COMMENTS\n"));
                        } else {
                                (void) printf(
                                    gettext("NAME       LINE         TIME\n"));
                        }
#else   /* XPG4 */
                        (void) printf(
                            gettext("NAME       LINE         TIME\n"));
#endif  /* XPG4 */
                }
        }
        process();

        /*
         *      'who -q' requires EOL upon exit,
         *      followed by total line
         */
        if (qopt)
                (void) printf(gettext("\n# users=%d\n"), totlusrs);
        return (0);
}

static void
dump()
{
        char    device[sizeof (utmpp->ut_line) + 1];
        time_t hr;
        time_t  idle;
        time_t min;
        char    path[sizeof (utmpp->ut_line) + 6];
        int     pexit;
        int     pterm;
        int     rc;
        char    w;      /* writeability indicator */

        /*
         * Get and check user name
         */
        if (utmpp->ut_user[0] == '\0')
                (void) strcpy(user, "   .");
        else {
                (void) strncpy(user, utmpp->ut_user, sizeof (user));
                user[sizeof (user) - 1] = '\0';
        }
        totlusrs++;

        /*
         * Do print in 'who -q' format
         */
        if (qopt) {
                /*
                 * XCU4 - Use non user macro for correct user count
                 */
                if (((totlusrs - 1) % number) == 0 && totlusrs > 1)
                        (void) printf("\n");
                (void) printf("%-*.*s ", LOGIN_WIDTH, NMAX, user);
                (void) printf("%-*s ", NMAX, user);
                return;
        }


        pexit = (int)’ ’;
        pterm = (int)’ ’;

        /*
         *      Get exit info if applicable
         */
        if (utmpp->ut_type == RUN_LVL || utmpp->ut_type == DEAD_PROCESS) {
                pterm = utmpp->ut_exit.e_termination;
                pexit = utmpp->ut_exit.e_exit;
        }

        /*
         *      Massage ut_xtime field
         */
        lptr = localtime(&utmpp->ut_xtime);
        (void) strftime(time_buf, sizeof (time_buf),
            dcgettext(NULL, DATE_FMT, LC_TIME), lptr);

        /*
         *      Get and massage device
         */
        if (utmpp->ut_line[0] == ’\0’)
                (void) strcpy(device, "     .");
        else {
                (void) strncpy(device, utmpp->ut_line,
                    sizeof (utmpp->ut_line));
                device[sizeof (utmpp->ut_line)] = ’\0’;
        }

        /*
         *      Get writeability if requested
         *      XCU4 - only print + or - for user processes
         */
        if (Topt && (utmpp->ut_type == USER_PROCESS)) {
                w = ’-’;
                (void) strcpy(path, "/dev/");
                (void) strncpy(path + 5, utmpp->ut_line,
                    sizeof (utmpp->ut_line));
                path[5 + sizeof (utmpp->ut_line)] = ’\0’;

                if ((rc = stat(path, stbufp)) == -1) w = ’?’;
                else if ((stbufp->st_mode & S_IWOTH) ||
                    (stbufp->st_mode & S_IWGRP))  /* Check group & other */
                        w = ’+’;

        } else
                w = ’ ’;

        /*
         *      Print the TERSE portion of the output
         */
        (void) printf("%-*.*s %c %-12s %s", LOGIN_WIDTH, NMAX, user,
            w, device, time_buf);
        (void) printf("%-*s %c %-12s %s", NMAX, user, w, device, time_buf);

        if (!terse) {
                /*
                 *      Stat device for idle time
                 *      (Don’t complain if you can’t)
                 */
                rc = -1;
                if (utmpp->ut_type == USER_PROCESS) {
```

```
                        (void) strcpy(path, "/dev/");
                        (void) strncpy(path + 5, utmpp->ut_line,
                            sizeof (utmpp->ut_line));
                        path[5 + sizeof (utmpp->ut_line)] = ’\0’;
                        rc = stat(path, stbufp);
                }
                if (rc != -1) {
                        idle = timnow - stbufp->st_mtime;
                        hr = idle/3600;
                        min = (unsigned)(idle/60)%60;
                        if (hr == 0 && min == 0)
                                (void) printf(gettext("   .  "));
                        else {
                                if (hr < 24)
                                        (void) printf(" %2d:%2.2d", (int)hr,
                                            (int)min);
                                else
                                        (void) printf(gettext("  old "));
                        }
                }

                /*
                 *      Add PID for verbose output
                 */
                if (utmpp->ut_type != BOOT_TIME &&
                    utmpp->ut_type != RUN_LVL &&
                    utmpp->ut_type != ACCOUNTING)
                        (void) printf("  %5ld", utmpp->ut_pid);

                /*
                 *      Handle /etc/inittab comment
                 */
                if (utmpp->ut_type == DEAD_PROCESS) {
                        (void) printf(gettext("  id=%4.4s "),
                            utmpp->ut_id);
                        (void) printf(gettext("term=%-3d "), pterm);
                        (void) printf(gettext("exit=%d  "), pexit);
                } else if (utmpp->ut_type != INIT_PROCESS) {
                        /*
                         *      Search for each entry in inittab
                         *      string. Keep our place from
                         *      search to search to try and
                         *      minimize the work. Wrap once if needed
                         *      for each entry.
                         */
                        wrap = 0;
                        /*
                         *      Look for a line beginning with
                         *      utmpp->ut_id
                         */
                        while ((rc = strncmp(utmpp->ut_id, iinit,
                            strcspn(iinit, ":"))) != 0) {
                                for (; *iinit != ’\n’; iinit++)
                                        ;
                                for (; *iinit != ’\n’; iinit++);
                                iinit++;

                                /*
                                 *      Wrap once if necessary to
                                 *      find entry in inittab
                                 */
                                if (*iinit == ’\0’) {
                                        if (!wrap) {
                                                iinit = inittab;
                                                wrap = 1;
                                        }
                                }
                        }

                        if (*iinit != '\0') {
                                /*
                                 *      We found our entry
                                 */
                                for (iinit++; *iinit != '#' &&
                                    *iinit != '\n'; iinit++)
                                        ;
                                         *iinit != '\n'; iinit++);
                                if (*iinit == '#') {
                                        for (iinit++; *iinit == ' ' ||
                                            *iinit == '\t'; iinit++)
                                                ;
                                                 *iinit == '\t'; iinit++);
                                        for (rc = 0; *iinit != '\n'; iinit++)
                                                comment[rc++] = *iinit;
                                        comment[rc] = '\0';
                                } else
                                        (void) strcpy(comment, " ");

                                (void) printf("  %s", comment);
                        } else
                                iinit = inittab;        /* Reset pointer */
                }
                if (utmpp->ut_type == INIT_PROCESS)
                        (void) printf(gettext("  id=%4.4s"), utmpp->ut_id);
        }
#ifdef  XPG4
        else
                if (dopt && utmpp->ut_type == DEAD_PROCESS) {
                        (void) printf(gettext("\tterm=%-3d "), pterm);
                        (void) printf(gettext("exit=%d  "), pexit);
                }
#endif  /* XPG4 */


        /*
         *      Handle RUN_LVL process - If no alt. file - Only one!
         */
        if (utmpp->ut_type == RUN_LVL) {
                (void) printf("     %c  %5ld  %c", pterm, utmpp->ut_pid,
                    pexit);
                if (optcnt == 1 && !validtype[USER_PROCESS]) {
                        (void) printf("\n");
                        exit(0);
                }
        }

        /*
         *      Handle BOOT_TIME process -  If no alt. file - Only one!
         */
        if (utmpp->ut_type == BOOT_TIME) {
                if (optcnt == 1 && !validtype[USER_PROCESS]) {
                        (void) printf("\n");
                        exit(0);
                }
        }

        /*
         *      Get remote host from utmpx structure
         */
        if (utmpp && utmpp->ut_host[0])
                (void) printf("\t(%.*s)", sizeof (utmpp->ut_host),
                    utmpp->ut_host);
```

```
        /*
         *      Now, put on the trailing EOL
         */
        (void) printf("\n");
}

static void
process()
{
        struct passwd *pwp;
        int i = 0;
        char *ttname;

        /*
         *      Loop over each entry in /var/adm/utmpx
         */

        setutxent();
        while ((utmpp = getutxent()) != NULL) {
#ifdef DEBUG
        (void) printf(
            "ut_user '%s'\nut_id '%s'\nut_line '%s'\nut_type '%d'\n\n",
            utmpp->ut_user, utmpp->ut_id, utmpp->ut_line, utmpp->ut_type);
#endif
                if (utmpp->ut_type <= UTMAXTYPE) {
                        /*
                         *      Handle "am i"
                         */
                        if (justme) {
                                if (strncmp(myname, utmpp->ut_user,
                                    sizeof (utmpp->ut_user)) == 0 &&
                                    strncmp(mytty, utmpp->ut_line,
                                    sizeof (utmpp->ut_line)) == 0 &&
                                    utmpp->ut_type == USER_PROCESS) {
                                        /*
                                         * we have have found ourselves
                                         * in the utmp file and the entry
                                         * is a user process, this is not
                                         * meaningful otherwise
                                         *
                                         */

                                        dump();
                                        exit(0);
                                }
                                continue;
                        }

                        /*
                         *      Print the line if we want it
                         */
                        if (validtype[utmpp->ut_type]) {
#ifdef  XPG4
                                if (utmpp->ut_type == LOGIN_PROCESS) {
                                        if ((utmpp->ut_line[0] == '\0') ||
                                            (strcmp(utmpp->ut_user,
                                            "LOGIN") != 0))
                                        (strcmp(utmpp->ut_user, "LOGIN") != 0))
                                                continue;
                                }
#endif  /* XPG4 */
                                dump();
                        }
                } else {
                        (void) fprintf(stderr,
				    gettext("%s: Error --- entry has ut_type "
				    "of %d\n"), program, utmpp->ut_type);
				(void) fprintf(stderr,
				    gettext(" when maximum is %d\n"), UTMAXTYPE);
			}
		}

		/*
		 * If justme is set at this point than the utmp entry
		 * was not found.
		 */
		if (justme) {
			static struct utmpx utmpt;

			pwp = getpwuid(geteuid());
			if (pwp != NULL)
				while (i < (int)sizeof (utmpt.ut_user) &&
				    *pwp->pw_name != 0)
					utmpt.ut_user[i++] = *pwp->pw_name++;

			ttname = ttyname(1);

			i = 0;
			if (ttname != NULL)
				while (i < (int)sizeof (utmpt.ut_line) &&
				    *ttname != 0)
					utmpt.ut_line[i++] = *ttname++;

			utmpt.ut_id[0] = 0;
			utmpt.ut_pid = getpid();
			utmpt.ut_type = USER_PROCESS;
			(void) time(&utmpt.ut_xtime);
			utmpp = &utmpt;
			dump();
			exit(0);
		}
}
```

______***unchanged_portion_omitted_***

****
21012 Mon Aug 12 18:24:54 2013
new/usr/src/cmd/whodo/whodo.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*
 * University Copyright- Copyright (c) 1982, 1986, 1988
 * The Regents of the University of California
 * All Rights Reserved
 *
 * University Acknowledgment- Portions of this document are derived from
 * software developed by the University of California, Berkeley, and its
 * contributors.
 */

/*
 * This is the new whodo command which takes advantage of
 * the /proc interface to gain access to the information
 * of all the processes currently on the system.
 *
 * Maintenance note:
 *
 * Much of this code is replicated in w.c.  If you're
 * fixing bugs here, then you should probably fix 'em there too.
 */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <fcntl.h>
#include <time.h>
#include <errno.h>
#include <sys/types.h>
#include <utmpx.h>
```

```
#include <sys/utsname.h>
#include <sys/stat.h>
#include <sys/mkdev.h>
#include <dirent.h>
#include <procfs.h>             /* /proc header file */
#include <sys/wait.h>
#include <locale.h>
#include <unistd.h>
#include <limits.h>
#include <priv_utils.h>

/*
 * Use the full lengths from utmpx for user and line.
 * utmpx defines wider fields for user and line.  For compatibility of output,
 * we are limiting these to the old maximums in utmp. Define UTMPX_NAMELEN
 * to use the full lengths.
 */
#define NMAX    (sizeof (((struct utmpx *)0)->ut_user))
#define LMAX    (sizeof (((struct utmpx *)0)->ut_line))
#ifndef UTMPX_NAMELEN
/* XXX - utmp - fix name length */
#define NMAX            (_POSIX_LOGIN_NAME_MAX - 1)
#define LMAX            12
#else /* UTMPX_NAMELEN */
static struct utmpx dummy;
#define NMAX    (sizeof (dummy.ut_user))
#define LMAX    (sizeof (dummy.ut_line))
#endif /* UTMPX_NAMELEN */

/* Print minimum field widths. */
#define LOGIN_WIDTH     8
#define LINE_WIDTH      12

#define DIV60(t)        ((t+30)/60)    /* x/60 rounded */

#ifdef ERR
#undef ERR
#endif
#define ERR             (-1)

#define DEVNAMELEN      14
#define HSIZE           256             /* size of process hash table */
#define PROCDIR         "/proc"
#define INITPROCESS     (pid_t)1        /* init process pid */
#define NONE            'n'             /* no state */
#define RUNNING         'r'             /* runnable process */
#define ZOMBIE          'z'             /* zombie process */
#define VISITED         'v'             /* marked node as visited */

static int      ndevs;                  /* number of configured devices */
static int      maxdev;                 /* slots for configured devices */
#define DNINCR  100
static struct devl {                    /* device list   */
        char    dname[DEVNAMELEN];      /* device name   */
        dev_t   ddev;                   /* device number */
} *devl;
```
______***unchanged_portion_omitted_***

```
/*
 *      define  hash table for struct uproc
 *      Hash function uses process id
 *      and the size of the hash table(HSIZE)
 *      to determine process index into the table.
 */
static struct uproc     pr_htbl[HSIZE];
static struct   uproc   *findhash(pid_t);
static time_t   findidle(char *);
static void     clnarglist(char *);
static void     showproc(struct uproc *);
static void     showtotals(struct uproc *);
static void     calctotals(struct uproc *);
static char     *getty(dev_t);
static void     prttime(time_t, char *);
static void     prtat(time_t *);
static void     checkampm(char *);

static char     *prog;
static int      header = 1;     /* true if -h flag: don't print heading */
static int      lflag = 0;      /* true if -l flag: w command format */
static char     *sel_user;      /* login of particular user selected */
static time_t   now;            /* current time of day */
static time_t   uptime;         /* time of last reboot & elapsed time since */
static int      nusers;         /* number of users logged in now */
static time_t   idle;           /* number of minutes user is idle */
static time_t   jobtime;        /* total cpu time visible */
static char     doing[520];     /* process attached to terminal */
static time_t   proctime;       /* cpu time of process in doing */
static int      empty;
static pid_t    curpid;

#if SIGQUIT > SIGINT
#define ACTSIZE SIGQUIT
#else
#define ACTSIZE SIGINT
#endif

int
main(int argc, char *argv[])
{
        struct utmpx    *ut;
        struct utmpx    *utmpbegin;
        struct utmpx    *utmpend;
        struct utmpx    *utp;
        struct tm               *tm;
        struct uproc    *up, *parent, *pgrp;
        struct psinfo   info;
        struct sigaction actinfo[ACTSIZE];
        struct pstatus  statinfo;
        size_t          size;
        struct stat     sbuf;
        struct utsname  uts;
        DIR             *dirp;
        struct  dirent  *dp;
        char            pname[64];
        char            *fname;
        int             procfd;
        int             i;
        int             days, hrs, mins;
        int             entries;

        /*
         * This program needs the proc_owner privilege
         */
        (void) __init_suid_priv(PU_CLEARLIMITSET, PRIV_PROC_OWNER,
            (char *)NULL);

        (void) setlocale(LC_ALL, "");
#if !defined(TEXT_DOMAIN)
#define TEXT_DOMAIN "SYS_TEST"
#endif
        (void) textdomain(TEXT_DOMAIN);
```

```
        prog = argv[0];

        while (argc > 1) {
                if (argv[1][0] == '-') {
                        for (i = 1; argv[1][i]; i++) {
                                switch (argv[1][i]) {

                                case 'h':
                                        header = 0;
                                        break;

                                case 'l':
                                        lflag++;
                                        break;

                                default:
                                        (void) printf(gettext(
                                            "usage: %s [ -hl ] [ user ]\n"),
                                            prog);
                                        exit(1);
                                }
                        }
                } else {
                        if (!isalnum(argv[1][0]) || argc > 2) {
                                (void) printf(gettext(
                                    "usage: %s [ -hl ] [ user ]\n"), prog);
                                exit(1);
                        } else
                                sel_user = argv[1];
                }
                argc--; argv++;
        }

        /*
         * read the UTMPX_FILE (contains information about
         * each logged in user)
         */
        if (stat(UTMPX_FILE, &sbuf) == ERR) {
                (void) fprintf(stderr, gettext("%s: stat error of %s: %s\n"),
                    prog, UTMPX_FILE, strerror(errno));
                exit(1);
        }
        entries = sbuf.st_size / sizeof (struct futmpx);
        size = sizeof (struct utmpx) * entries;

        if ((ut = malloc(size)) == NULL) {
                (void) fprintf(stderr, gettext("%s: malloc error of %s: %s\n"),
                    prog, UTMPX_FILE, strerror(errno));
                exit(1);
        }

        (void) utmpxname(UTMPX_FILE);

        utmpbegin = ut;
        /* LINTED pointer cast may result in improper alignment */
        utmpend = (struct utmpx *)((char *)utmpbegin + size);

        setutxent();
        while ((ut < utmpend) && ((utp = getutxent()) != NULL))
                (void) memcpy(ut++, utp, sizeof (*ut));
        endutxent();

        (void) time(&now);      /* get current time */

        if (header) {   /* print a header */
```

```
                if (lflag) {    /* w command format header */
                        prtat(&now);
                        for (ut = utmpbegin; ut < utmpend; ut++) {
                                if (ut->ut_type == USER_PROCESS) {
                                        nusers++;
                                } else if (ut->ut_type == BOOT_TIME) {
                                        uptime = now - ut->ut_xtime;
                                        uptime += 30;
                                        days = uptime / (60*60*24);
                                        uptime %= (60*60*24);
                                        hrs = uptime / (60*60);
                                        uptime %= (60*60);
                                        mins = uptime / 60;

                                        (void) printf(dcgettext(NULL,
                                            "  up %d day(s), %d hr(s), "
                                            "%d min(s)", LC_TIME),
                                            days, hrs, mins);
                                }
                        }

                        ut = utmpbegin; /* rewind utmp data */
                        (void) printf(dcgettext(NULL,
                            "  %d user(s)\n", LC_TIME), nusers);
                        (void) printf(dcgettext(NULL, "User     tty           "
                            "login@  idle   JCPU   PCPU  what\n", LC_TIME));
                } else {        /* standard whodo header */
                        char date_buf[100];

                        /*
                         * print current time and date
                         */
                        (void) strftime(date_buf, sizeof (date_buf),
                            dcgettext(NULL, "%C", LC_TIME), localtime(&now));
                        (void) printf("%s\n", date_buf);

                        /*
                         * print system name
                         */
                        (void) uname(&uts);
                        (void) printf("%s\n", uts.nodename);
                }
        }

        /*
         * loop through /proc, reading info about each process
         * and build the parent/child tree
         */
        if (!(dirp = opendir(PROCDIR))) {
                (void) fprintf(stderr, gettext("%s: could not open %s: %s\n"),
                    prog, PROCDIR, strerror(errno));
                exit(1);
        }

        while ((dp = readdir(dirp)) != NULL) {
                if (dp->d_name[0] == '.')
                        continue;
retry:
                (void) snprintf(pname, sizeof (pname),
                    "%s/%s/", PROCDIR, dp->d_name);
                fname = pname + strlen(pname);
                (void) strcpy(fname, "psinfo");
                if ((procfd = open(pname, O_RDONLY)) < 0)
                        continue;
                if (read(procfd, &info, sizeof (info)) != sizeof (info)) {
                        int err = errno;
```

```
                        (void) close(procfd);
                        if (err == EAGAIN)
                                goto retry;
                        if (err != ENOENT)
                                (void) fprintf(stderr, gettext(
                                    "%s: read() failed on %s: %s\n"),
                                    prog, pname, strerror(err));
                        continue;
                }
                (void) close(procfd);

                up = findhash(info.pr_pid);
                up->p_ttyd = info.pr_ttydev;
                up->p_state = (info.pr_nlwp == 0? ZOMBIE : RUNNING);
                up->p_time = 0;
                up->p_ctime = 0;
                up->p_igintr = 0;
                (void) strncpy(up->p_comm, info.pr_fname,
                    sizeof (info.pr_fname));
                up->p_args[0] = 0;

                if (up->p_state != NONE && up->p_state != ZOMBIE) {
                        (void) strcpy(fname, "status");

                        /* now we need the proc_owner privilege */
                        (void) __priv_bracket(PRIV_ON);

                        procfd = open(pname, O_RDONLY);

                        /* drop proc_owner privilege after open */
                        (void) __priv_bracket(PRIV_OFF);

                        if (procfd  < 0)
                                continue;

                        if (read(procfd, &statinfo, sizeof (statinfo))
                            != sizeof (statinfo)) {
                                int err = errno;
                                (void) close(procfd);
                                if (err == EAGAIN)
                                        goto retry;
                                if (err != ENOENT)
                                        (void) fprintf(stderr, gettext(
                                            "%s: read() failed on %s: %s \n"),
                                            prog, pname, strerror(err));
                                continue;
                        }
                        (void) close(procfd);

                        up->p_time = statinfo.pr_utime.tv_sec +
                            statinfo.pr_stime.tv_sec;
                        up->p_ctime = statinfo.pr_cutime.tv_sec +
                            statinfo.pr_cstime.tv_sec;

                        (void) strcpy(fname, "sigact");

                        /* now we need the proc_owner privilege */
                        (void) __priv_bracket(PRIV_ON);

                        procfd = open(pname, O_RDONLY);

                        /* drop proc_owner privilege after open */
                        (void) __priv_bracket(PRIV_OFF);

                        if (procfd  < 0)
                                continue;
                        if (read(procfd, actinfo, sizeof (actinfo))
                            != sizeof (actinfo)) {
                                int err = errno;
                                (void) close(procfd);
                                if (err == EAGAIN)
                                        goto retry;
                                if (err != ENOENT)
                                        (void) fprintf(stderr, gettext(
                                            "%s: read() failed on %s: %s \n"),
                                            prog, pname, strerror(err));
                                continue;
                        }
                        (void) close(procfd);

                        up->p_igintr =
                            actinfo[SIGINT-1].sa_handler == SIG_IGN &&
                            actinfo[SIGQUIT-1].sa_handler == SIG_IGN;

                        up->p_args[0] = 0;

                        /*
                         * Process args if there’s a chance we’ll print it.
                         */
                        if (lflag) { /* w command needs args */
                                clnarglist(info.pr_psargs);
                                (void) strcpy(up->p_args, info.pr_psargs);
                                if (up->p_args[0] == 0 ||
                                    up->p_args[0] == ’-’ &&
                                    up->p_args[1] <= ’ ’ ||
                                    up->p_args[0] == ’?’) {
                                        (void) strcat(up->p_args, " (");
                                        (void) strcat(up->p_args, up->p_comm);
                                        (void) strcat(up->p_args, ")");
                                }
                        }

                }

                /*
                 * link pgrp together in case parents go away
                 * Pgrp chain is a single linked list originating
                 * from the pgrp leader to its group member.
                 */
                if (info.pr_pgid != info.pr_pid) {      /* not pgrp leader */
                        pgrp = findhash(info.pr_pgid);
                        up->p_pgrplink = pgrp->p_pgrplink;
                        pgrp->p_pgrplink = up;
                }
                parent = findhash(info.pr_ppid);

                /* if this is the new member, link it in */
                if (parent->p_upid != INITPROCESS) {
                        if (parent->p_child) {
                                up->p_sibling = parent->p_child;
                                up->p_child = 0;
                        }
                        parent->p_child = up;
                }

        }

        /* revert to non-privileged user */
        (void) __priv_relinquish();

        (void) closedir(dirp);
        (void) time(&now);      /* get current time */
```

```
        /*
         * loop through utmpx file, printing process info
         * about each logged in user
         */
        for (ut = utmpbegin; ut < utmpend; ut++) {
                time_t tim;

                if (ut->ut_type != USER_PROCESS)
                        continue;
                if (sel_user && strncmp(ut->ut_name, sel_user, NMAX) != 0)
                        continue;       /* we’re looking for somebody else */
                if (lflag) {    /* -l flag format (w command) */
                        /* print login name of the user */
                        (void) printf("%-*.*s ", LOGIN_WIDTH, (int)NMAX,
                            ut->ut_name);
                        (void) printf("%-*.*s ", NMAX, NMAX, ut->ut_name);

                        /* print tty user is on */
                        (void) printf("%-*.*s", LINE_WIDTH, (int)LMAX,
                            ut->ut_line);
                        (void) printf("%-*.*s", LMAX, LMAX, ut->ut_line);

                        /* print when the user logged in */
                        tim = ut->ut_xtime;
                        (void) prtat(&tim);

                        /* print idle time */
                        idle = findidle(ut->ut_line);
                        if (idle >= 36 * 60)
                                (void) printf(dcgettext(NULL, "%2ddays ",
                                    LC_TIME), (idle + 12 * 60) / (24 * 60));
                        else
                                prttime(idle, " ");
                        showtotals(findhash((pid_t)ut->ut_pid));
                } else {        /* standard whodo format */
                        tim = ut->ut_xtime;
                        tm = localtime(&tim);
                        (void) printf("\n%-*.*s %-*.*s %2.1d:%2.2d\n",
                            LINE_WIDTH, (int)LMAX, ut->ut_line,
                            LOGIN_WIDTH, (int)NMAX, ut->ut_name, tm->tm_hour,
                            tm->tm_min);
                            LMAX, LMAX, ut->ut_line,
                            NMAX, NMAX, ut->ut_name, tm->tm_hour, tm->tm_min);
                        showproc(findhash((pid_t)ut->ut_pid));
                }
        }

        return (0);
}

/*
 * Used for standard whodo format.
 * This is the recursive routine descending the process
 * tree starting from the given process pointer(up).
 * It used depth-first search strategy and also marked
 * each node as printed as it traversed down the tree.
 */
static void
showproc(struct uproc *up)
{
        struct  uproc   *zp;

        if (up->p_state == VISITED) /* we already been here */
                return;
        /* print the data for this process */
        if (up->p_state == ZOMBIE)
                (void) printf("    %-*.*s %5d %4.1ld:%2.2ld %s\n",
                    LINE_WIDTH, (int)LMAX, "  ?", (int)up->p_upid, 0L, 0L,
                    "<defunct>");
                    LMAX, LMAX, "  ?", (int)up->p_upid, 0L, 0L, "<defunct>");
        else if (up->p_state != NONE) {
                (void) printf("    %-*.*s %5d %4.1ld:%2.2ld %s\n",
                    LINE_WIDTH, (int)LMAX, getty(up->p_ttyd), (int)up->p_upid,
                    LMAX, LMAX, getty(up->p_ttyd), (int)up->p_upid,
                    up->p_time / 60L, up->p_time % 60L,
                    up->p_comm);
        }
        up->p_state = VISITED;

        /* descend for its children */
        if (up->p_child) {
                showproc(up->p_child);
                for (zp = up->p_child->p_sibling; zp; zp = zp->p_sibling) {
                        showproc(zp);
                }
        }

        /* print the pgrp relation */
        if (up->p_pgrplink)
                showproc(up->p_pgrplink);
}
```
______***unchanged_portion_omitted_***

```
**********************************************************
   10785 Mon Aug 12 18:24:54 2013
new/usr/src/head/limits.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
```

```
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */


#ifndef _LIMITS_H
#define _LIMITS_H

#pragma ident   "%Z%%M% %I%     %E% SMI"        /* SVr4.0 1.34  */

#include <sys/feature_tests.h>
#include <sys/isa_defs.h>
#include <iso/limits_iso.h>

/*
 * Include fixed width type limits as proposed by the ISO/JTC1/SC22/WG14 C
 * committee's working draft for the revision of the current ISO C standard,
 * ISO/IEC 9899:1990 Programming language - C.  These are not currently
 * required by any standard but constitute a useful, general purpose set
 * of type definitions and limits which is namespace clean with respect to
 * all standards.
 */
#if defined(__EXTENSIONS__) || !defined(_STRICT_STDC) || \
        defined(__XOPEN_OR_POSIX)
#include <sys/int_limits.h>
#endif

#ifdef  __cplusplus
extern "C" {
#endif

#if defined(__EXTENSIONS__) || !defined(_STRICT_STDC) || \
        defined(__XOPEN_OR_POSIX)
```

```
#define SSIZE_MAX       LONG_MAX        /* max value of an "ssize_t" */

/*
 * ARG_MAX is calculated as follows:
 * NCARGS - space for other stuff on initial stack
 * like aux vectors, saved registers, etc..
 */
#define _ARG_MAX32      1048320 /* max length of args to exec 32-bit program */
#define _ARG_MAX64      2096640 /* max length of args to exec 64-bit program */
#ifdef  _LP64
#define ARG_MAX         _ARG_MAX64      /* max length of arguments to exec */
#else   /* _LP64 */
#define ARG_MAX         _ARG_MAX32      /* max length of arguments to exec */
#endif  /* _LP64 */

#ifndef MAX_CANON
#define MAX_CANON       256     /* max bytes in line for canonical processing */
#endif

#ifndef MAX_INPUT
#define MAX_INPUT       512     /* max size of a char input buffer */
#endif

#define NGROUPS_MAX     16      /* max number of groups for a user */

#ifndef PATH_MAX
#define PATH_MAX        1024    /* max # of characters in a path name */
#endif

#define SYMLINK_MAX     1024    /* max # of characters a symlink can contain */

#define PIPE_BUF        5120    /* max # bytes atomic in write to a pipe */

#ifndef TMP_MAX
#define TMP_MAX         17576   /* 26 * 26 * 26 */
#endif

/*
 * POSIX conformant definitions - An implementation may define
 * other symbols which reflect the actual implementation. Alternate
 * definitions may not be as restrictive as the POSIX definitions.
 */
#define _POSIX_AIO_LISTIO_MAX       2
#define _POSIX_AIO_MAX              1
#define _POSIX_ARG_MAX           4096
#ifdef _XPG6
#define _POSIX_CHILD_MAX           25
#else
#define _POSIX_CHILD_MAX            6   /* POSIX.1-1990 default */
#endif
#define _POSIX_CLOCKRES_MIN     20000000
#define _POSIX_DELAYTIMER_MAX      32
#define _POSIX_LINK_MAX             8
#define _POSIX_MAX_CANON          255
#define _POSIX_MAX_INPUT          255
#define _POSIX_MQ_OPEN_MAX          8
#define _POSIX_MQ_PRIO_MAX         32
#define _POSIX_NAME_MAX            14
#ifdef _XPG6
#define _POSIX_NGROUPS_MAX          8
#define _POSIX_OPEN_MAX            20
#define _POSIX_PATH_MAX           256
#else                                   /* POSIX.1-1990 defaults */
#define _POSIX_NGROUPS_MAX          0
#define _POSIX_OPEN_MAX            16
```

```
#define _POSIX_PATH_MAX           255
#endif
#define _POSIX_PIPE_BUF           512
#define _POSIX_RTSIG_MAX            8
#define _POSIX_SEM_NSEMS_MAX      256
#define _POSIX_SEM_VALUE_MAX    32767
#define _POSIX_SIGQUEUE_MAX        32
#define _POSIX_SSIZE_MAX        32767
#define _POSIX_STREAM_MAX           8
#define _POSIX_TIMER_MAX           32
#ifdef _XPG6
#define _POSIX_TZNAME_MAX           6
#else
#define _POSIX_TZNAME_MAX           3   /* POSIX.1-1990 default */
#endif
/* POSIX.1c conformant */
#define _POSIX_LOGIN_NAME_MAX                   9
#define _POSIX_THREAD_DESTRUCTOR_ITERATIONS     4
#define _POSIX_THREAD_KEYS_MAX                  128
#define _POSIX_THREAD_THREADS_MAX               64
#define _POSIX_TTY_NAME_MAX                     9
/* UNIX 03 conformant */
#define _POSIX_HOST_NAME_MAX                    255
#define _POSIX_RE_DUP_MAX                       255
#define _POSIX_SYMLINK_MAX                      255
#define _POSIX_SYMLOOP_MAX                      8

/*
 * POSIX.2 and XPG4-XSH4 conformant definitions
 */

#define _POSIX2_BC_BASE_MAX               99
#define _POSIX2_BC_DIM_MAX              2048
#define _POSIX2_BC_SCALE_MAX              99
#define _POSIX2_BC_STRING_MAX           1000
#define _POSIX2_COLL_WEIGHTS_MAX           2
#define _POSIX2_EXPR_NEST_MAX             32
#define _POSIX2_LINE_MAX                2048
#define _POSIX2_RE_DUP_MAX               255
/* UNIX 03 conformant */
#define _POSIX2_CHARCLASS_NAME_MAX        14

#define BC_BASE_MAX             _POSIX2_BC_BASE_MAX
#define BC_DIM_MAX              _POSIX2_BC_DIM_MAX
#define BC_SCALE_MAX            _POSIX2_BC_SCALE_MAX
#define BC_STRING_MAX           _POSIX2_BC_STRING_MAX
#define COLL_WEIGHTS_MAX        10
#define EXPR_NEST_MAX           _POSIX2_EXPR_NEST_MAX
#define LINE_MAX                _POSIX2_LINE_MAX
#if !defined(_XPG6)
#define RE_DUP_MAX              _POSIX2_RE_DUP_MAX
#else
#define RE_DUP_MAX              _POSIX_RE_DUP_MAX
#endif /* !defined(_XPG6) */

#endif /* defined(__EXTENSIONS__) || !defined(_STRICT_STDC) ... */

#if defined(__EXTENSIONS__) || \
        (!defined(_STRICT_STDC) && !defined(_POSIX_C_SOURCE)) || \
        defined(_XOPEN_SOURCE)

/*
 * For dual definitions for PASS_MAX and sysconf.c
 */
#define _PASS_MAX_XPG   8       /* old standards PASS_MAX */
#define _PASS_MAX       256     /* modern Solaris PASS_MAX */
```

```
#if defined(_XPG3) && !defined(_XPG6)
#define PASS_MAX        _PASS_MAX_XPG   /* max # of characters in a password */
#else   /* XPG6 or just Solaris */
#define PASS_MAX        _PASS_MAX       /* max # of characters in a password */
#endif  /* defined(_XPG3) && !defined(_XPG6) */

#define CHARCLASS_NAME_MAX      _POSIX2_CHARCLASS_NAME_MAX

#define NL_ARGMAX       9       /* max value of "digit" in calls to the */
                                /* NLS printf() and scanf() */
#define NL_LANGMAX      14      /* max # of bytes in a LANG name */
#define NL_MSGMAX       32767   /* max message number */
#define NL_NMAX         1       /* max # bytes in N-to-1 mapping characters */
#define NL_SETMAX       255     /* max set number */
#define NL_TEXTMAX      2048    /* max set number */
#define NZERO           20      /* default process priority */

#define WORD_BIT        32      /* # of bits in a "word" or "int" */
#if defined(_LP64)
#define LONG_BIT        64      /* # of bits in a "long" */
#else   /* _ILP32 */
#define LONG_BIT        32      /* # of bits in a "long" */
#endif

/* Marked as LEGACY in SUSv2 and removed in UNIX 03 */
#ifndef _XPG6
#define DBL_DIG         15      /* digits of precision of a "double" */
#define DBL_MAX         1.7976931348623157081452E+308   /* max decimal value */
                                                        /* of a double */
#define FLT_DIG         6               /* digits of precision of a "float" */
#define FLT_MAX         3.40282346638528859811704E+38F  /* max decimal value */
                                                        /* of a "float" */
#endif

/* Marked as LEGACY in SUSv1 and removed in SUSv2 */
#ifndef _XPG5
#define DBL_MIN         2.2250738585072013830903E-308   /* min decimal value */
                                                        /* of a double */
#define FLT_MIN         1.17549435082228750796888E-38F  /* min decimal value */
                                                        /* of a float */
#endif

#endif  /* defined(__EXTENSIONS__) || (!defined(_STRICT_STDC) ... */

#define _XOPEN_IOV_MAX  16      /* max # iovec/process with readv()/writev() */
#define _XOPEN_NAME_MAX 255     /* max # bytes in filename excluding null */
#define _XOPEN_PATH_MAX 1024    /* max # bytes in a pathname */

#define IOV_MAX         _XOPEN_IOV_MAX

#if defined(__EXTENSIONS__) || \
        (!defined(_STRICT_STDC) && !defined(__XOPEN_OR_POSIX))

#define FCHR_MAX        1048576         /* max size of a file in bytes */
#define PID_MAX         999999          /* max value for a process ID */

/*
 * POSIX 1003.1a, section 2.9.5, table 2-5 contains [NAME_MAX] and the
 * related text states:
 *
 * A definition of one of the values from Table 2-5 shall be omitted from the
 * <limits.h> on specific implementations where the corresponding value is
 * equal to or greater than the stated minimum, but where the value can vary
 * depending on the file to which it is applied. The actual value supported for
 * a specific pathname shall be provided by the pathconf() (5.7.1) function.
 *
 * This is clear that any machine supporting multiple file system types
 * and/or a network can not include this define, regardless of protection
 * by the _POSIX_SOURCE and _POSIX_C_SOURCE flags.
 *
 * #define      NAME_MAX        14
 */

#define CHILD_MAX       25      /* max # of processes per user id */
#ifndef OPEN_MAX
#define OPEN_MAX        256     /* max # of files a process can have open */
#endif

#define PIPE_MAX        5120    /* max # bytes written to a pipe in a write */

#define STD_BLK         1024    /* # bytes in a physical I/O block */
#define UID_MAX         2147483647      /* max value for a user or group ID */
#define USI_MAX         4294967295      /* max decimal value of an "unsigned" */
#define SYSPID_MAX      1       /* max pid of system processes */

#ifndef SYS_NMLN                /* also defined in sys/utsname.h */
#define SYS_NMLN        257     /* 4.0 size of utsname elements */
#endif

#ifndef CLK_TCK

#if !defined(_CLOCK_T) || __cplusplus >= 199711L
#define _CLOCK_T
typedef long    clock_t;
#endif  /* !_CLOCK_T */

extern long _sysconf(int);      /* System Private interface to sysconf() */
#define CLK_TCK ((clock_t)_sysconf(3))  /* 3 is _SC_CLK_TCK */

#endif /* CLK_TCK */

#ifdef  __USE_LEGACY_LOGNAME__
#define LOGNAME_MAX     8       /* max # of characters in a login name */
#else   /* __USE_LEGACY_LOGNAME__ */
#define LOGNAME_MAX     32      /* max # of characters in a login name */
                                /* Increased for illumos */
#endif  /* __USE_LEGACY_LOGNAME__ */
#define LOGIN_NAME_MAX  (LOGNAME_MAX + 1)       /* max buffer size */
#define LOGNAME_MAX_TRAD        8               /* traditional length */
#define LOGIN_NAME_MAX_TRAD     (LOGNAME_MAX_TRAD + 1)  /* and size */

#define TTYNAME_MAX     128     /* max # of characters in a tty name */

#endif  /* if defined(__EXTENSIONS__) || (!defined(_STRICT_STDC) ... */

#if     defined(__EXTENSIONS__) || (_POSIX_C_SOURCE >= 199506L)
#include <sys/unistd.h>

#if !defined(_SIZE_T) || __cplusplus >= 199711L
#define _SIZE_T
#if defined(_LP64) || defined(_I32LPx)
typedef unsigned long size_t;   /* size of something in bytes */
#else
typedef unsigned int  size_t;   /* (historical version) */
#endif
#endif  /* _SIZE_T */

extern long _sysconf(int);      /* System Private interface to sysconf() */

#define PTHREAD_STACK_MIN       ((size_t)_sysconf(_SC_THREAD_STACK_MIN))
/* Added for UNIX98 conformance */
```

```
#define PTHREAD_DESTRUCTOR_ITERATIONS   _POSIX_THREAD_DESTRUCTOR_ITERATIONS
#define PTHREAD_KEYS_MAX                _POSIX_THREAD_KEYS_MAX
#define PTHREAD_THREADS_MAX             _POSIX_THREAD_THREADS_MAX
#endif  /* defined(__EXTENSIONS__) || (_POSIX_C_SOURCE >= 199506L) */

#ifdef  __cplusplus
}
```

______*unchanged_portion_omitted_*

****
11675 Mon Aug 12 18:24:54 2013
new/usr/src/head/stdlib.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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 (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/* Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved. */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

#ifndef _STDLIB_H
#define _STDLIB_H

#include <iso/stdlib_iso.h>
#include <iso/stdlib_c99.h>

#if defined(__EXTENSIONS__) || defined(_XPG4)
#include <sys/wait.h>
#endif

/*
 * Allow global visibility for symbols defined in
 * C++ "std" namespace in <iso/stdlib_iso.h>.
 */
#if __cplusplus >= 199711L
using std::div_t;
using std::ldiv_t;
using std::size_t;
using std::abort;
using std::abs;
using std::atexit;
using std::atof;
using std::atoi;
using std::atol;
using std::bsearch;
using std::calloc;
using std::div;
using std::exit;
```

```
using std::free;
using std::getenv;
using std::labs;
using std::ldiv;
using std::malloc;
using std::mblen;
using std::mbstowcs;
using std::mbtowc;
using std::qsort;
using std::rand;
using std::realloc;
using std::srand;
using std::strtod;
using std::strtol;
using std::strtoul;
using std::system;
using std::wcstombs;
using std::wctomb;
#endif

#ifdef  __cplusplus
extern "C" {
#endif

#ifndef _UID_T
#define _UID_T
typedef unsigned int    uid_t;          /* UID type             */
#endif  /* !_UID_T */

#if defined(__STDC__)

/* large file compilation environment setup */
#if !defined(_LP64) && _FILE_OFFSET_BITS == 64

#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname        mkstemp         mkstemp64
#pragma redefine_extname        mkstemps        mkstemps64
#else   /* __PRAGMA_REDEFINE_EXTNAME */
#define mkstemp                 mkstemp64
#define mkstemps                mkstemps64
#endif  /* __PRAGMA_REDEFINE_EXTNAME */

#endif  /* _FILE_OFFSET_BITS == 64 */

/* In the LP64 compilation environment, all APIs are already large file */
#if defined(_LP64) && defined(_LARGEFILE64_SOURCE)

#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname        mkstemp64       mkstemp
#pragma redefine_extname        mkstemps64      mkstemps
#else   /* __PRAGMA_REDEFINE_EXTNAME */
#define mkstemp64               mkstemp
#define mkstemps64              mkstemps
#endif  /* __PRAGMA_REDEFINE_EXTNAME */

#endif  /* _LP64 && _LARGEFILE64_SOURCE */

#if defined(__EXTENSIONS__) || \
        (!defined(_STRICT_STDC) && !defined(__XOPEN_OR_POSIX)) || \
        (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_REENTRANT)
extern int rand_r(unsigned int *);
#endif

extern void _exithandle(void);

#if defined(__EXTENSIONS__) || \
	(!defined(_STRICT_STDC) && !defined(_POSIX_C_SOURCE)) || \
	defined(_XPG4)
extern double drand48(void);
extern double erand48(unsigned short *);
extern long jrand48(unsigned short *);
extern void lcong48(unsigned short *);
extern long lrand48(void);
extern long mrand48(void);
extern long nrand48(unsigned short *);
extern unsigned short *seed48(unsigned short *);
extern void srand48(long);
extern int putenv(char *);
extern void setkey(const char *);
#endif /* defined(__EXTENSIONS__) || !defined(_STRICT_STDC) ... */

/*
 * swab() has historically been in <stdlib.h> as delivered from AT&T
 * and continues to be visible in the default compilation environment.
 * As of Issue 4 of the X/Open Portability Guides, swab() was declared
 * in <unistd.h>. As a result, with respect to X/Open namespace the
 * swab() declaration in this header is only visible for the XPG3
 * environment.
 */
#if (defined(__EXTENSIONS__) || \
	(!defined(_STRICT_STDC__) && !defined(_POSIX_C_SOURCE))) && \
	(!defined(_XOPEN_SOURCE) || (defined(_XPG3) && !defined(_XPG4)))
#ifndef _SSIZE_T
#define _SSIZE_T
#if defined(_LP64) || defined(_I32LPx)
typedef long	ssize_t;	/* size of something in bytes or -1 */
#else
typedef int	ssize_t;	/* (historical version) */
#endif
#endif	/* !_SSIZE_T */

extern void swab(const char *, char *, ssize_t);
#endif /* defined(__EXTENSIONS__) || !defined(_STRICT_STDC) ... */

#if defined(__EXTENSIONS__) || \
	!defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
	(defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64)
extern int	mkstemp(char *);
#if !defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int	mkstemps(char *, int);
#endif
#endif /* defined(__EXTENSIONS__) ... */

#if	defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
	    !defined(__PRAGMA_REDEFINE_EXTNAME))
extern int	mkstemp64(char *);
#if !defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int	mkstemps64(char *, int);
#endif
#endif	/* _LARGEFILE64_SOURCE... */

#if defined(__EXTENSIONS__) || \
	(!defined(_STRICT_STDC) && !defined(__XOPEN_OR_POSIX)) || \
	defined(_XPG4_2)
extern long a64l(const char *);
extern char *ecvt(double, int, int *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
extern char *fcvt(double, int, int *_RESTRICT_KYWD, int *_RESTRICT_KYWD);
extern char *gcvt(double, int, char *);
extern int getsubopt(char **, char *const *, char **);
extern int  grantpt(int);
extern char *initstate(unsigned, char *, size_t);
extern char *l64a(long);
```

```
extern char *mktemp(char *);
extern char *ptsname(int);
extern long random(void);
extern char *realpath(const char *_RESTRICT_KYWD, char *_RESTRICT_KYWD);
extern char *setstate(const char *);
extern void srandom(unsigned);
extern int  unlockpt(int);
/* Marked LEGACY in SUSv2 and removed in SUSv3 */
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern int ttyslot(void);
extern void *valloc(size_t);
#endif /* !defined(_XPG6) || defined(__EXTENSIONS__) */
#endif /* defined(__EXTENSIONS__) || ... || defined(_XPG4_2) */

#if defined(__EXTENSIONS__) || \
	(!defined(_STRICT_STDC) && !defined(__XOPEN_OR_POSIX)) || \
	defined(_XPG6)
extern int posix_memalign(void **, size_t, size_t);
extern int posix_openpt(int);
extern int setenv(const char *, const char *, int);
extern int unsetenv(const char *);
#endif

#if defined(__EXTENSIONS__) || \
	(!defined(_STRICT_STDC) && !defined(__XOPEN_OR_POSIX))
extern char *canonicalize_file_name(const char *);
extern int clearenv(void);
extern void closefrom(int);
extern int daemon(int, int);
extern int dup2(int, int);
extern int dup3(int, int, int);
extern int fdwalk(int (*)(void *, int), void *);
extern char *qecvt(long double, int, int *, int *);
extern char *qfcvt(long double, int, int *, int *);
extern char *qgcvt(long double, int, char *);
extern char *getcwd(char *, size_t);
extern const char *getexecname(void);

#ifndef __GETLOGIN_DEFINED	/* Avoid duplicate in unistd.h */
#define __GETLOGIN_DEFINED
#ifndef __USE_LEGACY_LOGNAME__
#ifdef	__PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin getloginx
#else	/* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx(void);
#define getlogin	getloginx
#endif	/* __PRAGMA_REDEFINE_EXTNAME */
#endif	/* __USE_LEGACY_LOGNAME__ */
extern char *getlogin(void);
#endif	/* __GETLOGIN_DEFINED */

extern int getopt(int, char *const *, const char *);
extern char *optarg;
extern int optind, opterr, optopt;
extern char *getpass(const char *);
extern char *getpassphrase(const char *);
extern int getpw(uid_t, char *);
extern int isatty(int);
extern void *memalign(size_t, size_t);
extern char *ttyname(int);
extern char *mkdtemp(char *);
extern const char *getprogname(void);
extern void setprogname(const char *);

#if !defined(_STRICT_STDC) && defined(_LONGLONG_TYPE)
extern char *lltostr(long long, char *);
extern char *ulltostr(unsigned long long, char *);
#endif  /* !defined(_STRICT_STDC) && defined(_LONGLONG_TYPE) */

#endif /* defined(__EXTENSIONS__) || !defined(_STRICT_STDC) ... */

#else /* not __STDC__ */

#if defined(__EXTENSIONS__) || !defined(_XOPEN_OR_POSIX) || \
        (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_REENTRANT)
extern int rand_r();
#endif  /* defined(__EXTENSIONS__) || defined(_REENTRANT) ... */

extern void _exithandle();

#if defined(__EXTENSIONS__) || !defined(_POSIX_C_SOURCE) || defined(_XPG4)
extern double drand48();
extern double erand48();
extern long jrand48();
extern void lcong48();
extern long lrand48();
extern long mrand48();
extern long nrand48();
extern unsigned short *seed48();
extern void srand48();
extern int putenv();
extern void setkey();
#endif /* defined(__EXTENSIONS__) || !defined(_POSIX_C_SOURCE) ... */

#if (defined(__EXTENSIONS__) || !defined(_POSIX_C_SOURCE)) && \
        (!defined(_XOPEN_SOURCE) || (defined(_XPG3) && !defined(_XPG4)))
extern void swab();
#endif

#if defined(__EXTENSIONS__) || \
        !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64)
extern int      mkstemp();
#if !defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int      mkstemps();
#endif
#endif  /* defined(__EXTENSIONS__) ... */

#if     defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
            !defined(__PRAGMA_REDEFINE_EXTNAME))
extern int      mkstemp64();
#if !defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int      mkstemps64();
#endif
#endif  /* _LARGEFILE64_SOURCE... */

#if defined(__EXTENSIONS__) || !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)
extern long a64l();
extern char *ecvt();
extern char *fcvt();
extern char *gcvt();
extern int getsubopt();
extern int grantpt();
extern char *initstate();
extern char *l64a();
extern char *mktemp();
extern char *ptsname();
extern long random();
extern char *realpath();
extern char *setstate();
extern void srandom();
/* Marked LEGACY in SUSv2 and removed in SUSv3 */
```

```
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern int ttyslot();
extern void *valloc();
#endif /* !defined(_XPG6) || defined(__EXTENSIONS__) */
#endif /* defined(__EXTENSIONS__) || ... || defined(_XPG4_2) */

#if defined(__EXTENSIONS__) || !defined(__XOPEN_OR_POSIX) || defined(_XPG6)
extern int posix_memalign();
extern int posix_openpt();
extern int setenv();
extern int unsetenv();
#endif

#if defined(__EXTENSIONS__) || !defined(__XOPEN_OR_POSIX)
extern char *canonicalize_file_name();
extern int clearenv();
extern void closefrom();
extern int daemon();
extern int dup2();
extern int dup3();
extern int fdwalk();
extern char *qecvt();
extern char *qfcvt();
extern char *qgcvt();
extern char *getcwd();
extern char *getexecname();

#ifndef __GETLOGIN_DEFINED      /* Avoid duplicate in unistd.h */
#define __GETLOGIN_DEFINED
#ifndef __USE_LEGACY_LOGNAME__
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin getloginx
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx();
#define getlogin        getloginx
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */
extern char *getlogin();
#endif  /* __GETLOGIN_DEFINED */

extern int getopt();
extern char *optarg;
extern int optind, opterr, optopt;
extern char *getpass();
extern char *getpassphrase();
extern int getpw();
extern int isatty();
extern void *memalign();
extern char *ttyname();
extern char *mkdtemp();
extern char *getprogname();
extern void setprogname();

#if defined(_LONGLONG_TYPE)
extern char *lltostr();
extern char *ulltostr();
#endif  /* defined(_LONGLONG_TYPE) */
#endif  /* defined(__EXTENSIONS__) || !defined(__XOPEN_OR_POSIX) ... */

#endif  /* __STDC__ */

#ifdef  __cplusplus
}
#endif

#endif  /* _STDLIB_H */
```

```
**********************************************************
   40793 Mon Aug 12 18:24:54 2013
new/usr/src/head/unistd.h
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

/* Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved. */

#ifndef _UNISTD_H
#define _UNISTD_H

#include <sys/feature_tests.h>

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

#ifdef  __cplusplus
extern "C" {
#endif

/* Symbolic constants for the "access" routine: */
#define R_OK    4       /* Test for Read permission */
#define W_OK    2       /* Test for Write permission */
#define X_OK    1       /* Test for eXecute permission */
#define F_OK    0       /* Test for existence of File */

#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
#define F_ULOCK 0       /* Unlock a previously locked region */
#define F_LOCK  1       /* Lock a region for exclusive use */
#define F_TLOCK 2       /* Test and lock a region for exclusive use */
#define F_TEST  3       /* Test a region for other processes locks */
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */

/* Symbolic constants for the "lseek" routine: */

#ifndef SEEK_SET
```

```
#define SEEK_SET        0       /* Set file pointer to "offset" */
#endif

#ifndef SEEK_CUR
#define SEEK_CUR        1       /* Set file pointer to current plus "offset" */
#endif

#ifndef SEEK_END
#define SEEK_END        2       /* Set file pointer to EOF plus "offset" */
#endif

#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
#ifndef SEEK_DATA
#define SEEK_DATA       3       /* Set file pointer to next data past offset */
#endif

#ifndef SEEK_HOLE
#define SEEK_HOLE       4       /* Set file pointer to next hole past offset */
#endif
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* Path names: */
#define GF_PATH "/etc/group"    /* Path name of the "group" file */
#define PF_PATH "/etc/passwd"   /* Path name of the "passwd" file */
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

/*
 * compile-time symbolic constants,
 * Support does not mean the feature is enabled.
 * Use pathconf/sysconf to obtain actual configuration value.
 */

/* Values unchanged in UNIX 03 */
#define _POSIX_ASYNC_IO                 1
#define _POSIX_JOB_CONTROL              1
#define _POSIX_SAVED_IDS                1
#define _POSIX_SYNC_IO                  1

/*
 * POSIX.1b compile-time symbolic constants.
 */
#if defined(_XPG6)
#define _POSIX_ASYNCHRONOUS_IO          200112L
#define _POSIX_FSYNC                    200112L
#define _POSIX_MAPPED_FILES             200112L
#define _POSIX_MEMLOCK                  200112L
#define _POSIX_MEMLOCK_RANGE            200112L
#define _POSIX_MEMORY_PROTECTION        200112L
#define _POSIX_MESSAGE_PASSING          200112L
#define _POSIX_PRIORITY_SCHEDULING      200112L
#define _POSIX_REALTIME_SIGNALS         200112L
#define _POSIX_SEMAPHORES               200112L
#define _POSIX_SHARED_MEMORY_OBJECTS    200112L
#define _POSIX_SYNCHRONIZED_IO          200112L
#else
#define _POSIX_ASYNCHRONOUS_IO          1
#define _POSIX_FSYNC                    1
#define _POSIX_MAPPED_FILES             1
#define _POSIX_MEMLOCK                  1
#define _POSIX_MEMLOCK_RANGE            1
#define _POSIX_MEMORY_PROTECTION        1
#define _POSIX_MESSAGE_PASSING          1
#define _POSIX_PRIORITY_SCHEDULING      1
#define _POSIX_REALTIME_SIGNALS         1
#define _POSIX_SEMAPHORES               1
```

```
#define _POSIX_SHARED_MEMORY_OBJECTS    1
#define _POSIX_SYNCHRONIZED_IO          1
#endif

/*
 * POSIX.1c compile-time symbolic constants.
 */
#if defined(_XPG6)
#define _POSIX_THREAD_SAFE_FUNCTIONS            200112L
#define _POSIX_THREADS                          200112L
#define _POSIX_THREAD_ATTR_STACKADDR            200112L
#define _POSIX_THREAD_ATTR_STACKSIZE            200112L
#define _POSIX_THREAD_PROCESS_SHARED            200112L
#define _POSIX_THREAD_PRIORITY_SCHEDULING       200112L
#define _POSIX_TIMERS                           200112L
#else
#define _POSIX_THREAD_SAFE_FUNCTIONS            1
#define _POSIX_THREADS                          1
#define _POSIX_THREAD_ATTR_STACKADDR            1
#define _POSIX_THREAD_ATTR_STACKSIZE            1
#define _POSIX_THREAD_PROCESS_SHARED            1
#define _POSIX_THREAD_PRIORITY_SCHEDULING       1
#define _POSIX_TIMERS                           1
#endif

/* New in UNIX 03 */
#define _POSIX_ADVISORY_INFO                    200112L
#define _POSIX_BARRIERS                         200112L
#define _POSIX_CLOCK_SELECTION                  200112L
#define _POSIX_IPV6                             200112L
#define _POSIX_MONOTONIC_CLOCK                  200112L
#define _POSIX_RAW_SOCKETS                      200112L
#define _POSIX_READER_WRITER_LOCKS              200112L
#define _POSIX_SPAWN                            200112L
#define _POSIX_SPIN_LOCKS                       200112L
#define _POSIX_TIMEOUTS                         200112L

/*
 * Support for the POSIX.1 mutex protocol attribute. For realtime applications
 * which need mutexes to support priority inheritance/ceiling.
 */
#if defined(_XPG6)
#define _POSIX_THREAD_PRIO_INHERIT              200112L
#define _POSIX_THREAD_PRIO_PROTECT              200112L
#else
#define _POSIX_THREAD_PRIO_INHERIT              1
#define _POSIX_THREAD_PRIO_PROTECT              1
#endif

#ifndef _POSIX_VDISABLE
#define _POSIX_VDISABLE         0
#endif

#ifndef NULL
#if defined(_LP64)
#define NULL    0L
#else
#define NULL    0
#endif
#endif

#define STDIN_FILENO    0
#define STDOUT_FILENO   1
#define STDERR_FILENO   2

/*
```

```
 * Large File Summit-related announcement macros.  The system supports both
 * the additional and transitional Large File Summit interfaces.  (The final
 * two macros provide a finer granularity breakdown of _LFS64_LARGEFILE.)
 */
#define _LFS_LARGEFILE          1
#define _LFS64_LARGEFILE        1
#define _LFS64_STDIO            1
#define _LFS64_ASYNCHRONOUS_IO  1

/* large file compilation environment setup */
#if !defined(_LP64) && _FILE_OFFSET_BITS == 64
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname        ftruncate       ftruncate64
#pragma redefine_extname        lseek           lseek64
#pragma redefine_extname        pread           pread64
#pragma redefine_extname        pwrite          pwrite64
#pragma redefine_extname        truncate        truncate64
#pragma redefine_extname        lockf           lockf64
#pragma redefine_extname        tell            tell64
#else   /* __PRAGMA_REDEFINE_EXTNAME */
#define ftruncate                       ftruncate64
#define lseek                           lseek64
#define pread                           pread64
#define pwrite                          pwrite64
#define truncate                        truncate64
#define lockf                           lockf64
#define tell                            tell64
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* !_LP64 && _FILE_OFFSET_BITS == 64 */

/* In the LP64 compilation environment, the APIs are already large file */
#if defined(_LP64) && defined(_LARGEFILE64_SOURCE)
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname        ftruncate64     ftruncate
#pragma redefine_extname        lseek64         lseek
#pragma redefine_extname        pread64         pread
#pragma redefine_extname        pwrite64        pwrite
#pragma redefine_extname        truncate64      truncate
#pragma redefine_extname        lockf64         lockf
#pragma redefine_extname        tell64          tell
#else   /* __PRAGMA_REDEFINE_EXTNAME */
#define ftruncate64                     ftruncate
#define lseek64                         lseek
#define pread64                         pread
#define pwrite64                        pwrite
#define truncate64                      truncate
#define lockf64                         lockf
#define tell64                          tell
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* _LP64 && _LARGEFILE64_SOURCE */

#if defined(__STDC__)

extern int access(const char *, int);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int acct(const char *);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern unsigned alarm(unsigned);
/* Marked as LEGACY in SUSv2 and removed in SUSv3 */
#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern int brk(void *);
#endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
extern int chdir(const char *);
extern int chown(const char *, uid_t, gid_t);
/* Marked as LEGACY in SUSv2 and removed in SUSv3 */
```

```c
#if !defined(_POSIX_C_SOURCE) || (defined(_XOPEN_SOURCE) && \
        !defined(_XPG6)) || defined(__EXTENSIONS__)
extern int chroot(const char *);
#endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE))... */
extern int close(int);
#if defined(_XPG4) || defined(__EXTENSIONS__)
extern size_t confstr(int, char *, size_t);
extern char *crypt(const char *, const char *);
#endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
#if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
        defined(__EXTENSIONS__)
extern char *ctermid(char *);
#endif /* (!defined(_POSIX_C_SOURCE) ... */
#if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
extern char *ctermid_r(char *);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
/* Marked as LEGACY in SUSv2 and removed in SUSv3 */
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern char *cuserid(char *);
#endif
extern int dup(int);
extern int dup2(int, int);
extern int dup3(int, int, int);
#if defined(_XPG4) || defined(__EXTENSIONS__)
extern void encrypt(char *, int);
#endif /* defined(XPG4) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void endusershell(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int execl(const char *, const char *, ...);
extern int execle(const char *, const char *, ...);
extern int execlp(const char *, const char *, ...);
extern int execv(const char *, char *const *);
extern int execve(const char *, char *const *, char *const *);
extern int execvp(const char *, char *const *);
extern void _exit(int)
        __NORETURN;
/*
 * The following fattach prototype is duplicated in <stropts.h>. The
 * duplication is necessitated by XPG4.2 which requires the prototype
 * be defined in <stropts.h>.
 */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int fattach(int, const char *);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int fchdir(int);
extern int fchown(int, uid_t, gid_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int fchroot(int);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
        defined(__EXTENSIONS__)
extern int fdatasync(int);
#endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
/*
 * The following fdetach prototype is duplicated in <stropts.h>. The
 * duplication is necessitated by XPG4.2 which requires the prototype
 * be defined in <stropts.h>.
 */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int fdetach(const char *);
#endif /* !defined(__XOPEN_OR_POSIX)... */
extern pid_t fork(void);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
```

```c
extern pid_t fork1(void);
extern pid_t forkall(void);
#endif /* !defined(__XOPEN_OR_POSIX)... */
extern long fpathconf(int, int);
#if !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2) || \
        defined(__EXTENSIONS__)
extern int fsync(int);
#endif /* !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2)... */
#if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int ftruncate(int, off_t);
#endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
extern char *getcwd(char *, size_t);
#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern int getdtablesize(void);
#endif
extern gid_t getegid(void);
extern uid_t geteuid(void);
extern gid_t getgid(void);
extern int getgroups(int, gid_t *);
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern long gethostid(void);
#endif
#if defined(_XPG4_2)
extern int gethostname(char *, size_t);
#elif  !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int gethostname(char *, int);
#endif

#ifndef __GETLOGIN_DEFINED      /* Avoid duplicate in stdlib.h */
#define __GETLOGIN_DEFINED
#ifndef __USE_LEGACY_LOGNAME__
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin getloginx
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx(void);
#define getlogin        getloginx
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */
extern char *getlogin(void);
#endif  /* __GETLOGIN_DEFINED */

#if defined(_XPG4) || defined(__EXTENSIONS__)
extern int  getopt(int, char *const *, const char *);
extern char *optarg;
extern int  opterr, optind, optopt;
/* Marked as LEGACY in SUSv2 and removed in SUSv3 */
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern char *getpass(const char *);
#endif
#endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
/* Marked as LEGACY in SUSv2 and removed in SUSv3 */
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern int getpagesize(void);
#endif
extern pid_t getpgid(pid_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern pid_t getpid(void);
extern pid_t getppid(void);
extern pid_t getpgrp(void);

#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
char *gettxt(const char *, const char *);
```

```
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t getsid(pid_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern uid_t getuid(void);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern char *getusershell(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern char *getwd(char *);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
/*
 * The following ioctl prototype is duplicated in <stropts.h>. The
 * duplication is necessitated by XPG4.2 which requires the prototype
 * be defined in <stropts.h>.
 */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int ioctl(int, int, ...);
extern int isaexec(const char *, char *const *, char *const *);
extern int issetugid(void);
#endif
extern int isatty(int);
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int lchown(const char *, uid_t, gid_t);
#endif
extern int link(const char *, const char *);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern offset_t llseek(int, offset_t, int);
#endif
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int lockf(int, int, off_t);
#endif
extern off_t lseek(int, off_t, int);
#if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
        defined(__EXTENSIONS__)
extern int nice(int);
#endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int mincore(caddr_t, size_t, char *);
#endif
extern long pathconf(const char *, int);
extern int pause(void);
extern int pipe(int *);
extern int pipe2(int *, int);
#if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern ssize_t pread(int, void *, size_t, off_t);
#endif
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void profil(unsigned short *, size_t, unsigned long, unsigned int);
#endif
/*
 * pthread_atfork() is also declared in <pthread.h> as per SUSv3. The
 * declarations are identical. A change to either one may also require
 * appropriate namespace updates in order to avoid redeclaration
 * warnings in the case where both prototypes are exposed via inclusion
 * of both <pthread.h> and <unistd.h>.
 */
#if !defined(__XOPEN_OR_POSIX) || \
        ((_POSIX_C_SOURCE > 2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern int pthread_atfork(void (*) (void), void (*) (void), void (*) (void));
#endif /* !defined(__XOPEN_OR_POSIX) || ((_POSIX_C_SOURCE > 2) ... */
```

```
#if !defined(_LP64) && \
        (!defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__))
extern int ptrace(int, pid_t, int, int);
#endif
#if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern ssize_t pwrite(int, const void *, size_t, off_t);
#endif
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* per RFC 3542; This is also defined in netdb.h */
extern int rcmd_af(char **, unsigned short, const char *, const char *,
        const char *, int *, int);
#endif
extern ssize_t read(int, void *, size_t);
#if !defined(__XOPEN_OR_POSIX) || \
        defined(_XPG4_2) || defined(__EXTENSIONS__)
extern ssize_t readlink(const char *_RESTRICT_KYWD, char *_RESTRICT_KYWD,
        size_t);
#endif
#if (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3) && !defined(_XPG4))) || \
        defined(__EXTENSIONS__)
#if __cplusplus >= 199711L
namespace std {
#endif
extern int rename(const char *, const char *);
#if __cplusplus >= 199711L
} /* end of namespace std */

using std::rename;
#endif /* __cplusplus >= 199711L */
#endif /* (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int resolvepath(const char *, char *, size_t);
/* per RFC 3542; This is also defined in netdb.h */
extern int rexec_af(char **, unsigned short, const char *, const char *,
        const char *, int *, int);
#endif /* !defined(__XOPEN_OR_POSIX)|| defined(__EXTENSIONS__) */
extern int rmdir(const char *);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* per RFC 3542; This is also defined in netdb.h */
extern int rresvport_af(int *, int);
#endif

#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern void *sbrk(intptr_t);
#endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG6) || defined(__EXTENSIONS__)
extern int setegid(gid_t);
extern int seteuid(uid_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG6) ... */
extern int setgid(gid_t);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int setgroups(int, const gid_t *);
extern int sethostname(char *, int);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int setpgid(pid_t, pid_t);
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t setpgrp(void);
extern int setregid(gid_t, gid_t);
extern int setreuid(uid_t, uid_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern pid_t setsid(void);
extern int setuid(uid_t);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
```

```
extern void setusershell(void);
#endif /* !defined(__XOPEN_OR_POSIX)|| defined(__EXTENSIONS__) */
extern unsigned sleep(unsigned);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int stime(const time_t *);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if defined(_XPG4)
/* __EXTENSIONS__ makes the SVID Third Edition prototype in stdlib.h visible */
extern void swab(const void *_RESTRICT_KYWD, void *_RESTRICT_KYWD, ssize_t);
#endif /* defined(_XPG4) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int symlink(const char *, const char *);
extern void sync(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) */
#if defined(_XPG5) && !defined(_XPG6)
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname sysconf __sysconf_xpg5
#else /* __PRAGMA_REDEFINE_EXTNAME */
#define sysconf __sysconf_xpg5
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif /* defined(_XPG5) && !defined(_XPG6) */
extern long sysconf(int);
extern pid_t tcgetpgrp(int);
extern int tcsetpgrp(int, pid_t);
#if !defined(__XOPEN_OR_POSIX) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern off_t tell(int);
#endif /* !defined(__XOPEN_OR_POSIX)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int truncate(const char *, off_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern char *ttyname(int);
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern useconds_t ualarm(useconds_t, useconds_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern int unlink(const char *);
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int usleep(useconds_t);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t vfork(void) __RETURNS_TWICE;
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void vhangup(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern ssize_t write(int, const void *, size_t);
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void yield(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

#if !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE) || \
        defined(__EXTENSIONS__)
        /* || defined(_XPG7) */
extern int faccessat(int, const char *, int, int);
extern int fchownat(int, const char *, uid_t, gid_t, int);
extern int linkat(int, const char *, int, const char *, int);
extern ssize_t readlinkat(int, const char *_RESTRICT_KYWD,
        char *_RESTRICT_KYWD, size_t);
extern int renameat(int, const char *, int, const char *);
extern int symlinkat(const char *, int, const char *);
extern int unlinkat(int, const char *, int);
#endif  /* !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
```

```
extern int get_nprocs(void);
extern int get_nprocs_conf(void);
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

/* transitional large file interface versions */
#if     defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
            !defined(__PRAGMA_REDEFINE_EXTNAME))
extern int ftruncate64(int, off64_t);
extern off64_t lseek64(int, off64_t, int);
extern ssize_t  pread64(int, void *, size_t, off64_t);
extern ssize_t  pwrite64(int, const void *, size_t, off64_t);
extern off64_t  tell64(int);
extern int      truncate64(const char *, off64_t);
extern int      lockf64(int, int, off64_t);
#endif  /* _LARGEFILE64_SOURCE */

#else  /* __STDC__ */

extern int access();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int acct();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern unsigned alarm();
#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern int brk();
#endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
extern int chdir();
extern int chown();
#if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
        defined(__EXTENSIONS__)
extern int chroot();
#endif /* (!defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
extern int close();
#if defined(_XPG4) || defined(__EXTENSIONS__)
extern size_t confstr();
extern char *crypt();
#endif /* defined(XPG4) || defined(__EXTENSIONS__) */
#if !defined(_POSIX_C_SOURCE) || defined(_XPG3) || defined(__EXTENSIONS__)
extern char *ctermid();
#endif /* (!defined(_POSIX_C_SOURCE) || defined(_XPG3)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
extern char *ctermid_r();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern char *cuserid();
#endif
extern int dup();
extern int dup2();
extern int dup3();
#if defined(_XPG4) || defined(__EXTENSIONS__)
extern void encrypt();
#endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void endusershell();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int execl();
extern int execle();
extern int execlp();
extern int execv();
extern int execve();
extern int execvp();
extern void _exit();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int fattach();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
```

```
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int fchdir();
extern int fchown();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int fchroot();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
        defined(__EXTENSIONS__)
extern int fdatasync();
#endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
#if !defined(__XOPEN_OR_POSIX)
extern int fdetach();
#endif /* !defined(__XOPEN_OR_POSIX) */
extern pid_t fork();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern pid_t fork1();
extern pid_t forkall();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern long fpathconf();
#if !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2) || \
        defined(__EXTENSIONS__)
extern int fsync();
#endif /* !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2)... */
#if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int ftruncate();
#endif /* !defined(__XOPEN_OR_POSIX) (_POSIX_C_SOURCE > 2)... */
extern char *getcwd();
#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern int getdtablesize();
#endif
extern gid_t getegid();
extern uid_t geteuid();
extern gid_t getgid();
extern int getgroups();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern long gethostid();
#endif
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int gethostname();
#endif

#ifndef __GETLOGIN_DEFINED      /* Avoid duplicate in stdlib.h */
#define __GETLOGIN_DEFINED
#ifndef __USE_LEGACY_LOGNAME__
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin       getloginx
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx();
#define getlogin        getloginx
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */
extern char *getlogin();
#endif  /* __GETLOGIN_DEFINED */

#if defined(_XPG4) || defined(__EXTENSIONS__)
extern int  getopt();
extern char *optarg;
extern int  opterr, optind, optopt;
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern char *getpass();
#endif
#endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
```

```
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
#if !defined(_XPG6) || defined(__EXTENSIONS__)
extern int getpagesize();
#endif
extern pid_t getpgid();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern pid_t getpid();
extern pid_t getppid();
extern pid_t getpgrp();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
char *gettxt();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t getsid();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) */
extern uid_t getuid();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern char *getusershell();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern char *getwd();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int ioctl();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int isaexec();
extern int issetugid();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int isatty();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int lchown();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) */
extern int link();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern offset_t llseek();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int lockf();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern off_t lseek();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int mincore();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
        defined(__EXTENSIONS__)
extern int nice();
#endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
extern long pathconf();
extern int pause();
extern int pipe();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG5) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern ssize_t pread();
#endif
#if !defined(_LP64) && \
        (!defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__))
extern void profil();
extern int ptrace();
#endif
#if !defined(__XOPEN_OR_POSIX) || \
        ((_POSIX_C_SOURCE > 2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
```

```
extern int pthread_atfork();
#endif /* !defined(__XOPEN_OR_POSIX) || ((_POSIX_C_SOURCE > 2) ...  */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG5) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern ssize_t pwrite();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG5) */
extern ssize_t read();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* per RFC 3542; This is also defined in netdb.h */
extern int rcmd_af();
#endif
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern ssize_t readlink();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3) && !defined(_XPG4))) || \
        defined(__EXTENSIONS__)
extern int rename();
#endif /* (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int resolvepath();
/* per RFC 3542; This is also defined in netdb.h */
extern int rexec_af();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int rmdir();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
/* per RFC 3542; This is also defined in netdb.h */
extern int rresvport_af();
#endif
#if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
        defined(__EXTENSIONS__)
extern void *sbrk();
#endif /* !defined(__XOPEN_OR_POSIX)|| (defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG6) || defined(__EXTENSIONS__)
extern int setegid();
extern int seteuid();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG6) ... */
extern int setgid();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int setgroups();
extern int sethostname();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern int setpgid();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t setpgrp();
extern int setregid();
extern int setreuid();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern pid_t setsid();
extern int setuid();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void setusershell();
#endif /* !defined(__XOPEN_OR_POSIX)|| defined(__EXTENSIONS__) */
extern unsigned sleep();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int stime();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
#if defined(_XPG4)
/* __EXTENSIONS__ makes the SVID Third Edition prototype in stdlib.h visible */
extern void swab();
#endif /* defined(_XPG4) */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int symlink();
extern void sync();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if defined(_XPG5)
```

```
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname sysconf __sysconf_xpg5
extern long sysconf();
#else /* __PRAGMA_REDEFINE_EXTNAME */
extern long __sysconf_xpg5();
#define sysconf __sysconf_xpg5
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* defined(_XPG5) */
extern pid_t tcgetpgrp();
extern int tcsetpgrp();
#if !defined(__XOPEN_OR_POSIX) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern off_t tell();
#endif /* !defined(__XOPEN_OR_POSIX)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
        (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
        defined(__EXTENSIONS__)
extern int truncate();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern char *ttyname();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern useconds_t ualarm();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
extern int unlink();
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern int usleep();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
extern pid_t vfork();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void vhangup();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
extern ssize_t write();
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern void yield();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

#if !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE) || \
        defined(__EXTENSIONS__)
        /* || defined(_XPG7) */
extern int faccessat();
extern int fchownat();
extern int linkat();
extern ssize_t readlinkat();
extern int renameat();
extern int symlinkat();
extern int unlinkat();
#endif  /* !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE)... */
#if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
extern int get_nprocs();
extern int get_nprocs_conf();
#endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

/* transitional large file interface versions */
#if     defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
            !defined(__PRAGMA_REDEFINE_EXTNAME))
extern int ftruncate64();
extern off64_t lseek64();
extern ssize_t pread64();
extern ssize_t pwrite64();
extern off64_t tell64();
extern int truncate64();
extern int lockf64();
#endif  /* _LARGEFILE64_SOURCE */
```

```
#endif /* __STDC__ */

#if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
#pragma unknown_control_flow(vfork)
#endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */

/*
 * getlogin_r() & ttyname_r() prototypes are defined here.
 */

/*
 * Previous releases of Solaris, starting at 2.3, provided definitions of
 * various functions as specified in POSIX.1c, Draft 6.  For some of these
 * functions, the final POSIX 1003.1c standard had a different number of
 * arguments and return values.
 *
 * The following segment of this header provides support for the standard
 * interfaces while supporting applications written under earlier
 * releases.  The application defines appropriate values of the feature
 * test macros _POSIX_C_SOURCE and _POSIX_PTHREAD_SEMANTICS to indicate
 * whether it was written to expect the Draft 6 or standard versions of
 * these interfaces, before including this header.  This header then
 * provides a mapping from the source version of the interface to an
 * appropriate binary interface.  Such mappings permit an application
 * to be built from libraries and objects which have mixed expectations
 * of the definitions of these functions.
 *
 * For applications using the Draft 6 definitions, the binary symbol is the
 * same as the source symbol, and no explicit mapping is needed.  For the
 * standard interface, the function func() is mapped to the binary symbol
 * _posix_func().  The preferred mechanism for the remapping is a compiler
 * #pragma.  If the compiler does not provide such a #pragma, the header file
 * defines a static function func() which calls the _posix_func() version;
 * this has to be done instead of #define since POSIX specifies that an
 * application can #undef the symbol and still be bound to the correct
 * implementation.  Unfortunately, the statics confuse lint so we fallback to
 * #define in that case.
 *
 * NOTE: Support for the Draft 6 definitions is provided for compatibility
 * only.  New applications/libraries should use the standard definitions.
 */

#if     defined(__EXTENSIONS__) || defined(_REENTRANT) || \
        !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE - 0 >= 199506L) || \
        defined(_POSIX_PTHREAD_SEMANTICS)

#if     defined(__STDC__)

#if     (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_POSIX_PTHREAD_SEMANTICS)

#ifndef __USE_LEGACY_LOGNAME__
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r __posix_getloginx_r
extern int getlogin_r(char *, int);
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern int __posix_getloginx_r(char *, int);
#define getlogin_r      __posix_getloginx_r
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#else   /* __USE_LEGACY_LOGNAME__ */
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r __posix_getlogin_r
#pragma redefine_extname ttyname_r __posix_ttyname_r
extern int getlogin_r(char *, int);
extern int ttyname_r(int, char *, size_t);
#else  /* __PRAGMA_REDEFINE_EXTNAME */
```

```
extern int __posix_getlogin_r(char *, int);
extern int __posix_ttyname_r(int, char *, size_t);

#ifdef __lint

#define getlogin_r      __posix_getlogin_r
#define ttyname_r       __posix_ttyname_r

#else /* !__lint */

static int
getlogin_r(char *__name, int __len)
{
        return (__posix_getlogin_r(__name, __len));
}

#endif /* !__lint */
#endif /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */

#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname ttyname_r __posix_ttyname_r
extern int ttyname_r(int, char *, size_t);
#else  /* __PRAGMA_REDEFINE_EXTNAME */
extern int __posix_ttyname_r(int, char *, size_t);

#ifdef __lint

#define ttyname_r       __posix_ttyname_r

#else /* !__lint */

static int
ttyname_r(int __fildes, char *__buf, size_t __size)
{
        return (__posix_ttyname_r(__fildes, __buf, __size));
}

#endif /* !__lint */
#endif /* __PRAGMA_REDEFINE_EXTNAME */

#else  /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

#ifndef __USE_LEGACY_LOGNAME__
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r getloginx_r
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx_r(char *, int);
#define getlogin_r      getloginx_r
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */
extern char *getlogin_r(char *, int);

extern char *ttyname_r(int, char *, int);

#endif /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

#else  /* __STDC__ */

#if (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_POSIX_PTHREAD_SEMANTICS)

#ifndef __USE_LEGACY_LOGNAME__
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r __posix_getloginx_r
extern int getlogin_r();
```

```
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern int __posix_getloginx_r();
#define getlogin_r      __posix_getloginx_r
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
#else   /* __USE_LEGACY_LOGNAME__ */
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r __posix_getlogin_r
#pragma redefine_extname ttyname_r __posix_ttyname_r
extern int getlogin_r();
extern int ttyname_r();
#else  /* __PRAGMA_REDEFINE_EXTNAME */

extern int __posix_getlogin_r();
extern int __posix_ttyname_r();

#ifdef  __lint

#define getlogin_r      __posix_getlogin_r
#define ttyname_r       __posix_ttyname_r

#else /* !__lint */

static int
getlogin_r(__name, __len)
        char *__name;
        int __len;
{
        return (__posix_getlogin_r(__name, __len));
}
#endif /* !__lint */
#endif /* __PRAGMA_REDEFINE_EXTNAME */
#endif  /* __USE_LEGACY_LOGNAME__ */

#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname ttyname_r __posix_ttyname_r
extern int ttyname_r();
#else  /* __PRAGMA_REDEFINE_EXTNAME */

extern int __posix_ttyname_r();

#ifdef  __lint

#define ttyname_r       __posix_ttyname_r

#else /* !__lint */

static int
ttyname_r(__fildes, __buf, __size)
        int __fildes;
        char *__buf;
        size_t __size;
{
        return (__posix_ttyname_r(__fildes, __buf, __size));
}
#endif /* !__lint */
#endif /* __PRAGMA_REDEFINE_EXTNAME */

#else  /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

#ifndef __USE_LEGACY_LOGNAME__
#ifdef __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname getlogin_r     getloginx_r
#else   /* __PRAGMA_REDEFINE_EXTNAME */
extern char *getloginx_r();
#define getlogin_r      getloginx_r
#endif  /* __PRAGMA_REDEFINE_EXTNAME */
```

```
#endif  /* __USE_LEGACY_LOGNAME__ */
extern char *getlogin_r();

extern char *ttyname_r();

#endif /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

#endif /* __STDC__ */

#endif /* defined(__EXTENSIONS__) || defined(_REENTRANT)... */

#ifdef  __cplusplus
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    4669 Mon Aug 12 18:24:54 2013
new/usr/src/lib/libc/port/gen/getlogin.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
/*
 * 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 2008 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */

#pragma weak _getlogin = getloginx
#pragma weak _getlogin_r = getloginx_r
#pragma ident   "%Z%%M% %I%     %E% SMI"

#pragma weak _getlogin = getlogin
#pragma weak _getlogin_r = getlogin_r

#include "lint.h"
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <limits.h>
#include "utmpx.h"
#include <unistd.h>
#include <errno.h>
#include <thread.h>
#include <synch.h>
#include <mtlib.h>
#include "tsd.h"

/* Revert the renames done in unistd.h */
#ifdef  __PRAGMA_REDEFINE_EXTNAME
#pragma redefine_extname        getlogint       getlogin
#pragma redefine_extname        getlogint_r     getlogin_r
#pragma redefine_extname        __posix_getlogint_r     __posix_getlogin_r
#else   /* __PRAGMA_REDEFINE_EXTNAME */
#ifdef  getlogin
```

```
#undef  getlogin
#endif  /* getlogin */
#ifdef  getlogin_r
#undef  getlogin_r
#endif  /* getlogin_r */
#ifdef  __posix_getlogin_r
#undef  __posix_getlogin_r
#endif  /* __posix_getlogin_r */
#define getlogint       getlogin
#define getlogint_r     getlogin_r
#define __posix_getlogint_r     __posix_getlogin_r
#endif  /* __PRAGMA_REDEFINE_EXTNAME */

/*
 * Use the full length of a login name.
 * The utmpx interface provides for a 32 character login name.
 * XXX - _POSIX_LOGIN_NAME_MAX limits the length of a login name.  The utmpx
 * interface provides for a 32 character login name, but for the sake of
 * compatibility, we are still using the old utmp-imposed limit.
 */
#define NMAX    (sizeof (((struct utmpx *)0)->ut_user))

/*
 * Common function
 * POSIX.1c Draft-6 version of the function getlogin_r.
 * It was implemented by Solaris 2.3.
 */
static char *
getl_r_common(char *answer, size_t namelen, size_t maxlen)
char *
getlogin_r(char *answer, int namelen)
{
        int             uf;
        off64_t         me;
        struct futmpx   ubuf;

        if (namelen < _POSIX_LOGIN_NAME_MAX) {
                errno = ERANGE;
                return (NULL);
        }

        if ((me = (off64_t)ttyslot()) < 0)
                return (NULL);
        if ((uf = open64(UTMPX_FILE, 0)) < 0)
                return (NULL);
        (void) lseek64(uf, me * sizeof (ubuf), SEEK_SET);
        if (read(uf, &ubuf, sizeof (ubuf)) != sizeof (ubuf)) {
                (void) close(uf);
                return (NULL);
        }
        (void) close(uf);
        if (ubuf.ut_user[0] == '\0')
                return (NULL);

        /* Insufficient buffer size */
        if (namelen < strnlen(&ubuf.ut_user[0], maxlen)) {
                errno = ERANGE;
                return (NULL);
        }
        (void) strncpy(&answer[0], &ubuf.ut_user[0], maxlen);
        answer[maxlen] = '\0';
        (void) strncpy(&answer[0], &ubuf.ut_user[0],
            _POSIX_LOGIN_NAME_MAX - 1);
        answer[_POSIX_LOGIN_NAME_MAX - 1] = '\0';
        return (&answer[0]);
}
```

```
/*
 * POSIX.1c Draft-6 version of the function getlogin_r.
 * It was implemented by Solaris 2.3.
 */
char *
getlogint_r(char *answer, int namelen)
{
        return (getl_r_common(answer, (size_t)namelen, LOGNAME_MAX_TRAD));
}

/*
 * POSIX.1c standard version of the function getlogin_r.
 * User gets it via static getlogin_r from the header file.
 */
int
__posix_getlogint_r(char *name, int namelen)
__posix_getlogin_r(char *name, int namelen)
{
        int nerrno = 0;
        int oerrno = errno;

        errno = 0;
        if (getl_r_common(name, (size_t)namelen, LOGNAME_MAX_TRAD) == NULL) {
        if (getlogin_r(name, namelen) == NULL) {
                if (errno == 0)
                        nerrno = EINVAL;
                else
                        nerrno = errno;
        }
        errno = oerrno;
        return (nerrno);
}

char *
getlogint(void)
getlogin(void)
{
        char *answer = tsdalloc(_T_LOGIN, LOGIN_NAME_MAX_TRAD, NULL);
        char *answer = tsdalloc(_T_LOGIN, _POSIX_LOGIN_NAME_MAX, NULL);

        if (answer == NULL)
                return (NULL);
        return (getl_r_common(answer, LOGIN_NAME_MAX_TRAD, LOGNAME_MAX_TRAD));
}

/*
 * POSIX.1c Draft-6 version of the function getlogin_r.
 * It was implemented by Solaris 2.3.
 * For extended login names, selected by redefine_extname in unistd.h.
 */
char *
getloginx_r(char *answer, int namelen)
{
        return (getl_r_common(answer, (size_t)namelen, NMAX));
}

/*
 * POSIX.1c standard version of the function getlogin_r.
 * User gets it via static getlogin_r from the header file.
 * For extended login names, selected by redefine_extname in unistd.h.
 */
int
__posix_getloginx_r(char *name, int namelen)
{
        int nerrno = 0;
```

```
        int oerrno = errno;

        errno = 0;
        if (getl_r_common(name, (size_t)namelen, NMAX) == NULL) {
                if (errno == 0)
                        nerrno = EINVAL;
                else
                        nerrno = errno;
        }
        errno = oerrno;
        return (nerrno);
}

/*
 * For extended login names, selected by redefine_extname in unistd.h.
 */
char *
getloginx(void)
{
        char *answer = tsdalloc(_T_LOGIN, LOGIN_NAME_MAX, NULL);

        if (answer == NULL)
                return (NULL);
        return (getl_r_common(answer, LOGIN_NAME_MAX, NMAX));
        return (getlogin_r(answer, _POSIX_LOGIN_NAME_MAX));
}
```

______***unchanged_portion_omitted_***

```
************************************************************
   12091 Mon Aug 12 18:24:54 2013
new/usr/src/lib/libc/port/gen/sysconf.c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
************************************************************
/*
 * 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) 1988 AT&T */
/*        All Rights Reserved   */

/* sysconf(3C) - returns system configuration information */

#pragma weak _sysconf = sysconf

#include "lint.h"
#include <mtlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/sysconfig.h>
#include <limits.h>
#include <time.h>
#include <errno.h>
#include <nss_dbdefs.h>
#include <thread.h>
#include <xti.h>
#include "libc.h"
#include "xpg6.h"

/* from nss_common.c */
extern size_t _nss_get_bufsizes(int);

long
sysconf(int name)
{
        static int _pagesize = 0;
        static int _hz = 0;
        static pid_t _maxpid = 0;
        static int _stackprot = 0;
        static int _ngroups_max;
```

```
        extern int __xpg4;

        switch (name) {
                default:
                        errno = EINVAL;
                        return (-1L);

                case _SC_ARG_MAX:
                        return ((long)ARG_MAX);

                case _SC_CLK_TCK:
                        if (_hz <= 0)
                                _hz = _sysconfig(_CONFIG_CLK_TCK);
                        return (_hz);

                case _SC_JOB_CONTROL:
                        return ((long)_POSIX_JOB_CONTROL);

                case _SC_SAVED_IDS:
                        return ((long)_POSIX_SAVED_IDS);

                case _SC_CHILD_MAX:
                        return (_sysconfig(_CONFIG_CHILD_MAX));

                case _SC_NGROUPS_MAX:
                        if (_ngroups_max <= 0)
                                _ngroups_max = _sysconfig(_CONFIG_NGROUPS);
                        return (_ngroups_max);

                case _SC_OPEN_MAX:
                        return (_sysconfig(_CONFIG_OPEN_FILES));

                case _SC_VERSION:
                        if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
                                return (200112L);
                        else
                                return (199506L);

                case _SC_PAGESIZE:
                        if (_pagesize <= 0)
                                _pagesize = _sysconfig(_CONFIG_PAGESIZE);
                        return (_pagesize);

                case _SC_XOPEN_VERSION:
                        if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
                                return (600L);
                        else if (__xpg4 == 0)
                                return (_sysconfig(_CONFIG_XOPEN_VER));
                        else
                                return (4L);

                case _SC_XOPEN_XCU_VERSION:
                        if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
                                return (600L);
                        else
                                return (4L);

                /*
                 * old value for pre XPG5 conformant systems to match
                 * getpass() length.
                 * XPG5 special cased with __sysconf_xpg5()
                 * new value for default and modern XPG systems.
                 */
                case _SC_PASS_MAX:
                        if ((__xpg4 == 1) &&
                            (!(__xpg6 & _C99SUSv3_XPG6_sysconf_version)))
```

```
				return ((long)_PASS_MAX_XPG);
			else
				return ((long)_PASS_MAX);

		case _SC_LOGNAME_MAX:
			return ((long)LOGNAME_MAX);

		case _SC_STREAM_MAX:
			return (_sysconfig(_CONFIG_OPEN_FILES));

		case _SC_TZNAME_MAX:
			return (-1L);

		case _SC_NPROCESSORS_CONF:
			return (_sysconfig(_CONFIG_NPROC_CONF));

		case _SC_NPROCESSORS_ONLN:
			return (_sysconfig(_CONFIG_NPROC_ONLN));

		case _SC_NPROCESSORS_MAX:
			return (_sysconfig(_CONFIG_NPROC_MAX));

		case _SC_STACK_PROT:
			if (_stackprot == 0)
				_stackprot = _sysconfig(_CONFIG_STACK_PROT);
			return (_stackprot);

		/* POSIX.4 names */

		/*
		 * Each of the following also have _POSIX_* symbols
		 * defined in <unistd.h>. Values here should align
		 * with values in the header. Up until the SUSv3 standard
		 * we defined these simply as 1. With the introduction
		 * of the new revision, these were changed to 200112L.
		 * The standard allows us to change the value, however,
		 * we have kept both values in case application programs
		 * are relying on the previous value even though an
		 * application doing so is technically wrong.
		 */
		case _SC_ASYNCHRONOUS_IO:
		case _SC_FSYNC:
		case _SC_MAPPED_FILES:
		case _SC_MEMLOCK:
		case _SC_MEMLOCK_RANGE:
		case _SC_MEMORY_PROTECTION:
		case _SC_MESSAGE_PASSING:
		case _SC_PRIORITY_SCHEDULING:
		case _SC_REALTIME_SIGNALS:
		case _SC_SEMAPHORES:
		case _SC_SHARED_MEMORY_OBJECTS:
		case _SC_SYNCHRONIZED_IO:
		case _SC_TIMERS:
			if (__xpg6 & _C99SUSv3_mode_ON)
				return (200112L);
			else
				return (1L);

		case _SC_PRIORITIZED_IO:
#ifdef _POSIX_PRIORITIZED_IO
			return (1L);
#else
			return (-1L);
#endif

		case _SC_AIO_LISTIO_MAX:
```

```
			return (_sysconfig(_CONFIG_AIO_LISTIO_MAX));

		case _SC_AIO_MAX:
			return (_sysconfig(_CONFIG_AIO_MAX));

		case _SC_AIO_PRIO_DELTA_MAX:
			return (_sysconfig(_CONFIG_AIO_PRIO_DELTA_MAX));

		case _SC_DELAYTIMER_MAX:
			return (_sysconfig(_CONFIG_DELAYTIMER_MAX));

		case _SC_MQ_OPEN_MAX:
			return (_sysconfig(_CONFIG_MQ_OPEN_MAX));

		case _SC_MQ_PRIO_MAX:
			return (_sysconfig(_CONFIG_MQ_PRIO_MAX));

		case _SC_RTSIG_MAX:
			return (_sysconfig(_CONFIG_RTSIG_MAX));

		case _SC_SEM_NSEMS_MAX:
			return (_sysconfig(_CONFIG_SEM_NSEMS_MAX));

		case _SC_SEM_VALUE_MAX:
			return (_sysconfig(_CONFIG_SEM_VALUE_MAX));

		case _SC_SIGQUEUE_MAX:
			return (_sysconfig(_CONFIG_SIGQUEUE_MAX));

		case _SC_SIGRT_MAX:
			return (_sysconfig(_CONFIG_SIGRT_MAX));

		case _SC_SIGRT_MIN:
			return (_sysconfig(_CONFIG_SIGRT_MIN));

		case _SC_TIMER_MAX:
			return (_sysconfig(_CONFIG_TIMER_MAX));

		case _SC_PHYS_PAGES:
			return (_sysconfig(_CONFIG_PHYS_PAGES));

		case _SC_AVPHYS_PAGES:
			return (_sysconfig(_CONFIG_AVPHYS_PAGES));

		/* XPG4/POSIX.1-1990/POSIX.2-1992 names */
		case _SC_2_C_BIND:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_CHAR_TERM:
			return ((long)_POSIX2_CHAR_TERM);

		case _SC_2_C_DEV:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_C_VERSION:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (199209L);
		case _SC_2_FORT_DEV:
			return (-1L);

		case _SC_2_FORT_RUN:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_LOCALEDEF:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_SW_DEV:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_UPE:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (1L);

		case _SC_2_VERSION:
			if (__xpg6 & _C99SUSv3_XPG6_sysconf_version)
				return (200112L);
			else
				return (199209L);

		case _SC_BC_BASE_MAX:
			return ((long)BC_BASE_MAX);

		case _SC_BC_DIM_MAX:
			return ((long)BC_DIM_MAX);

		case _SC_BC_SCALE_MAX:
			return ((long)BC_SCALE_MAX);

		case _SC_BC_STRING_MAX:
			return ((long)BC_STRING_MAX);

		case _SC_COLL_WEIGHTS_MAX:
			return ((long)COLL_WEIGHTS_MAX);

		case _SC_EXPR_NEST_MAX:
			return ((long)EXPR_NEST_MAX);

		case _SC_LINE_MAX:
			return ((long)LINE_MAX);

		case _SC_RE_DUP_MAX:
			return ((long)RE_DUP_MAX);

		case _SC_XOPEN_CRYPT:
			return (1L);

		case _SC_XOPEN_ENH_I18N:
			return ((long)_XOPEN_ENH_I18N);

		case _SC_XOPEN_SHM:
			return ((long)_XOPEN_SHM);
```

```
		/* XPG4v2 (SUS) names */
		case _SC_XOPEN_UNIX:
			return (1L);

		case _SC_XOPEN_LEGACY:
			return (1L);

		case _SC_ATEXIT_MAX:
			return (-1L);

		case _SC_IOV_MAX:
			return ((long)IOV_MAX);

		case _SC_T_IOV_MAX:
			return ((long)T_IOV_MAX);

		/* XPG5 (SUSv2) names */
		case _SC_XOPEN_REALTIME:
			return (1L);

		case _SC_XOPEN_REALTIME_THREADS:
#if defined(_POSIX_THREAD_PRIORITY_SCHEDULING) && \
	defined(_POSIX_THREAD_PRIO_INHERIT) && \
	defined(_POSIX_THREAD_PRIO_PROTECT)
			return (1L);
#else
			return (-1L);
#endif

		case _SC_XBS5_ILP32_OFF32:
			return (1L);

		case _SC_XBS5_ILP32_OFFBIG:
			return (1L);

		case _SC_XBS5_LP64_OFF64:
			return (1L);

		case _SC_XBS5_LPBIG_OFFBIG:
			return (1L);

		/* POSIX.1c names */
		case _SC_THREAD_DESTRUCTOR_ITERATIONS:
			return (-1L);

		case _SC_GETGR_R_SIZE_MAX:
			return ((long)_nss_get_bufsizes(_SC_GETGR_R_SIZE_MAX));

		case _SC_GETPW_R_SIZE_MAX:
			return ((long)NSS_BUFLEN_PASSWD);

		case _SC_LOGIN_NAME_MAX:
			return ((long)(LOGIN_NAME_MAX));
			return ((long)(LOGNAME_MAX + 1));

		case _SC_THREAD_KEYS_MAX:
			return (-1L);

		case _SC_THREAD_STACK_MIN:
			return ((long)thr_min_stack());

		case _SC_THREAD_THREADS_MAX:
			return (-1L);

		case _SC_TTY_NAME_MAX:
			return ((long)TTYNAME_MAX);
		case _SC_BARRIERS:
			return ((long)_POSIX_BARRIERS);

		case _SC_CLOCK_SELECTION:
			return ((long)_POSIX_CLOCK_SELECTION);

		case _SC_MONOTONIC_CLOCK:
			return ((long)_POSIX_MONOTONIC_CLOCK);

		case _SC_SPAWN:
			return ((long)_POSIX_SPAWN);

		case _SC_SPIN_LOCKS:
			return ((long)_POSIX_SPIN_LOCKS);

		case _SC_THREADS:
		case _SC_THREAD_ATTR_STACKADDR:
		case _SC_THREAD_ATTR_STACKSIZE:
		case _SC_THREAD_PRIORITY_SCHEDULING:
		case _SC_THREAD_PRIO_INHERIT:
		case _SC_THREAD_PRIO_PROTECT:
		case _SC_THREAD_PROCESS_SHARED:
		case _SC_THREAD_SAFE_FUNCTIONS:
			if (__xpg6 & _C99SUSv3_mode_ON)
				return (200112L);
			else
				return (1L);

		case _SC_TIMEOUTS:
			return ((long)_POSIX_TIMEOUTS);

		/* 1216676 - cache info */
		case _SC_COHER_BLKSZ:
			return (_sysconfig(_CONFIG_COHERENCY));

		case _SC_SPLIT_CACHE:
			return (_sysconfig(_CONFIG_SPLIT_CACHE));

		case _SC_ICACHE_SZ:
			return (_sysconfig(_CONFIG_ICACHESZ));

		case _SC_DCACHE_SZ:
			return (_sysconfig(_CONFIG_DCACHESZ));

		case _SC_ICACHE_LINESZ:
			return (_sysconfig(_CONFIG_ICACHELINESZ));

		case _SC_DCACHE_LINESZ:
			return (_sysconfig(_CONFIG_DCACHELINESZ));

		case _SC_ICACHE_BLKSZ:
			return (_sysconfig(_CONFIG_ICACHEBLKSZ));

		case _SC_DCACHE_BLKSZ:
			return (_sysconfig(_CONFIG_DCACHEBLKSZ));

		case _SC_DCACHE_TBLKSZ:
			return (_sysconfig(_CONFIG_DCACHETBLKSZ));

		case _SC_ICACHE_ASSOC:
			return (_sysconfig(_CONFIG_ICACHE_ASSOC));

		case _SC_DCACHE_ASSOC:
			return (_sysconfig(_CONFIG_DCACHE_ASSOC));
```

```
		case _SC_MAXPID:
			if (_maxpid <= 0)
				_maxpid = _sysconfig(_CONFIG_MAXPID);
			return (_maxpid);

		case _SC_CPUID_MAX:
			return (_sysconfig(_CONFIG_CPUID_MAX));

		case _SC_EPHID_MAX:
			return (_sysconfig(_CONFIG_EPHID_MAX));

		/* UNIX 03 names - XPG6/SUSv3/POSIX.1-2001 */

		case _SC_REGEXP:
			return ((long)_POSIX_REGEXP);

		case _SC_SHELL:
			return ((long)_POSIX_SHELL);

		case _SC_ADVISORY_INFO:
			return ((long)_POSIX_ADVISORY_INFO);

		case _SC_HOST_NAME_MAX:
			return ((long)_POSIX_HOST_NAME_MAX);

		case _SC_READER_WRITER_LOCKS:
			return ((long)_POSIX_READER_WRITER_LOCKS);

		case _SC_IPV6:
			return ((long)_POSIX_IPV6);

		case _SC_RAW_SOCKETS:
			return ((long)_POSIX_RAW_SOCKETS);

		case _SC_XOPEN_STREAMS:
			return ((long)_XOPEN_STREAMS);

		case _SC_SYMLOOP_MAX:
			return (_sysconfig(_CONFIG_SYMLOOP_MAX));

		case _SC_V6_ILP32_OFF32:
			return (1L);

		case _SC_V6_ILP32_OFFBIG:
			return (1L);

		case _SC_V6_LP64_OFF64:
			return (1L);

		case _SC_V6_LPBIG_OFFBIG:
			return (1L);

		/* Unsupported UNIX 03 options */
		case _SC_2_PBS:
		case _SC_2_PBS_ACCOUNTING:
		case _SC_2_PBS_CHECKPOINT:
		case _SC_2_PBS_LOCATE:
		case _SC_2_PBS_MESSAGE:
		case _SC_2_PBS_TRACK:
		case _SC_CPUTIME:
		case _SC_SPORADIC_SERVER:
		case _SC_SS_REPL_MAX:
		case _SC_THREAD_CPUTIME:
		case _SC_THREAD_SPORADIC_SERVER:
		case _SC_TRACE:
		case _SC_TRACE_EVENT_FILTER:
```

```
                case _SC_TRACE_EVENT_NAME_MAX:
                case _SC_TRACE_INHERIT:
                case _SC_TRACE_LOG:
                case _SC_TRACE_NAME_MAX:
                case _SC_TRACE_SYS_MAX:
                case _SC_TRACE_USER_EVENT_MAX:
                case _SC_TYPED_MEMORY_OBJECTS:
                        return (-1L);
        }
}
```
______***unchanged_portion_omitted_***

****
   45866 Mon Aug 12 18:24:55 2013
new/usr/src/lib/libc/port/llib-lc
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
/*
 * 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 2011 Nexenta Systems, Inc. All rights reserved.
 * Copyright 2013 OmniTI Computer Consulting, Inc. All rights reserved.
 * Copyright (c) 2013 Gary Mills
 */

/* LINTLIBRARY */
/* PROTOLIB1 */

#define __EXTENSIONS__

#include <aio.h>
#include <alloca.h>
#include <attr.h>
#include <atomic.h>
#include <ctype.h>
#include <deflt.h>
#include <dirent.h>
#include <dlfcn.h>
#include <door.h>
#include <err.h>
#include <sys/errno.h>
#include <euc.h>
#include <fcntl.h>
#include <float.h>
#include <fmtmsg.h>
#include <fnmatch.h>
#include <ftw.h>
#include <glob.h>
#include <getwidth.h>
#include <grp.h>
#include <iconv.h>
#include <langinfo.h>
#include <libgen.h>
#include <libw.h>
#include <locale.h>
#include <memory.h>
#include <mon.h>
```

```
#include <mqueue.h>
#include <nan.h>
#include <ndbm.h>
#include <limits.h>
#include <nl_types.h>
#include <poll.h>
#include <project.h>
#include <priv.h>
#include <pwd.h>
#include <rctl.h>
#include <regex.h>
#include <rpcsvc/ypclnt.h>
#include <sched.h>
#include <search.h>
#include <semaphore.h>
#include <setjmp.h>
#include <shadow.h>
#include <siginfo.h>
#include <signal.h>
#include <stdarg.h>
#include <ucred.h>
#include <sys/ucred.h>
#include <unistd.h>
#include <ulimit.h>
#include <utime.h>
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stropts.h>
#include <synch.h>
#include <sys/acctctl.h>
#include <sys/acl.h>
#include <sys/asynch.h>
#include <sys/byteorder.h>
#include <sys/cladm.h>
#include <sys/corectl.h>
#include <sys/dl.h>
#include <sys/exacct.h>
#include <sys/fcntl.h>
#include <sys/file.h>
#include <sys/fs/namenode.h>
#include <sys/instance.h>
#include <sys/ipc.h>
#include <sys/lwp.h>
#include <sys/mkdev.h>
#include <sys/mman.h>
#include <sys/mnttab.h>
#include <sys/mount.h>
#include <sys/msg.h>
#include <sys/param.h>
#include <sys/priocntl.h>
#include <sys/procset.h>
#include <sys/processor.h>
#include <sys/pset.h>
#include <sys/rctl_impl.h>
#include <sys/sem.h>
#include <sys/shm.h>
#include <sys/sid.h>
#include <sys/signal.h>
#include <sys/stat.h>
#include <sys/statvfs.h>
#include <sys/strlog.h>
#include <sys/stropts.h>
#include <sys/syscall.h>
#include <sys/sysconfig.h>
```

```
#include <sys/syslog.h>
#include <sys/systeminfo.h>
#include <sys/task.h>
#include <sys/termio.h>
#include <sys/termios.h>
#include <sys/u8_textprep.h>
#include <sys/time.h>
#include <sys/timeb.h>
#include <sys/times.h>
#include <sys/types.h>
#include <sys/uadmin.h>
#include <sys/utsname.h>
#include <sys/vfstab.h>
#include <sys/sendfile.h>
#include <sys/zone.h>
#include <termio.h>
#include <time.h>
#include <tzfile.h>
#include <ucontext.h>
#include <utmpx.h>
#include <values.h>
#include <wait.h>
#include <wchar.h>
#include <wctype.h>
#include <widec.h>
#include <wordexp.h>
#include <thread.h>
#include <pthread.h>
#include <schedctl.h>
#include <zone.h>
#include <port.h>
#include <spawn.h>
#include <inttypes.h>
#include <getopt.h>
#include <stdio_ext.h>
#if defined(__i386)
#include <sys/sysi86.h>
#endif
#if defined(__amd64)
#include <stack_unwind.h>
#endif

/*
 * This really comes from the crt*.s startup modules.
 */
char **environ;

/*
 * This is a GNU/Linux/BSD compatibility interface,
 * not declared in any header file.
 */
const char *__progname;

/*
 * POSIX versions of standard libc routines; these aren’t extracted
 * from the headers above since we cannot #define _POSIX_C_SOURCE.
 */
int __posix_readdir_r(DIR * _RESTRICT_KYWD, struct dirent * _RESTRICT_KYWD,
    struct dirent ** _RESTRICT_KYWD);
int __posix_getgrgid_r(gid_t, struct group *, char *, size_t, struct group **);
int __posix_getgrnam_r(const char *, struct group *, char *, size_t,
    struct group **);
int __posix_getpwuid_r(uid_t, struct passwd *, char *, size_t,
    struct passwd **);
int __posix_getpwnam_r(const char *, struct passwd *, char *, size_t,
    struct passwd **);
```

```
int __posix_sigwait(const sigset_t * _RESTRICT_KYWD, int * _RESTRICT_KYWD);
char *__posix_asctime_r(const struct tm *_RESTRICT_KYWD, char *_RESTRICT_KYWD);
char *__posix_ctime_r(const time_t *, char *);
int __posix_ttyname_r(int, char *, size_t);
int __posix_getlogin_r(char *, int);
int __posix_getloginx_r(char *, int);

/*
 * XPG4 versions of standard libc routines; these aren’t extracted
 * from the headers above since we cannot #define _XPG4_2.
 */
int __xpg4_putmsg(int, const struct strbuf *, const struct strbuf *, int);
int __xpg4_putpmsg(int, const struct strbuf *, const struct strbuf *, int, int);

/*
 * These aren’t extracted from the headers above because:
 * - We cannot #define _STRPTIME_DONTZERO
 * - We cannot #define _XPG5
 */
char *__strptime_dontzero(const char *, const char *, struct tm *);
long __sysconf_xpg5(int);
wchar_t *__wcstok_xpg5(wchar_t *_RESTRICT_KYWD,
        const wchar_t *_RESTRICT_KYWD, wchar_t **_RESTRICT_KYWD);
size_t __wcsftime_xpg5(wchar_t *_RESTRICT_KYWD, size_t,
        const wchar_t *_RESTRICT_KYWD, const struct tm *_RESTRICT_KYWD);
wint_t __fgetwc_xpg5(__FILE *);
wint_t __getwc_xpg5(__FILE *);
wint_t __getwchar_xpg5(void);
wint_t __fputwc_xpg5(wint_t, __FILE *);
wint_t __putwc_xpg5(wint_t, __FILE *);
wint_t __putwchar_xpg5(wint_t);
wchar_t *__fgetws_xpg5(wchar_t *_RESTRICT_KYWD, int, __FILE *_RESTRICT_KYWD);
int __fputws_xpg5(const wchar_t *_RESTRICT_KYWD, __FILE *_RESTRICT_KYWD);
wint_t __ungetwc_xpg5(wint_t, __FILE *);

/*
 * /usr/src/lib/libc/port/gen routines
 */

/* _ctype.c */

/* _loc_data.c */

/* _locale.c */

/* _set_tab.c */
int _set_tab(const char *loc, int cat);

/* _xftw.c */
int _xftw(int ver, const char *path, int (*fn)(), int depth);

/* a64l.c */
long a64l(const char *);

/* abort.c */
void abort(void);

/* abs.c */
int abs(int arg);
long labs(long int arg);

/* assert.c */
void _assert(const char *assertion, const char *filename, int line_num);
void __assert_c99(const char *assertion, const char *filename, int line_num,
    const char *funcname);
```

```
/* atexit.c */
int atexit(void(*func)());
void _exithandle(void);

/* atof.c */
double atof(const char *p);

/* atoi.c */
int atoi(const char *p);

/* atol.c */
long atol(const char *p);

/* basename.c */
char *basename(char *s);

/* bcmp.c */
int bcmp(const void *s1, const void *s2, size_t len);

/* bcopy.c */
void bcopy(const void *s1, void *s2, size_t len);

/* bsearch.c */
void *bsearch(const void *ky, const void *bs, size_t nel,
                size_t width, int (*compar)());

/* bzero.c */
void bzero(void *sp, size_t len);

/* calloc.c */
void *calloc(size_t num, size_t size);

/* catclose.c */
int catclose(nl_catd catd);

/* catgets.c */
char *catgets(nl_catd catd, int set_num, int msg_num, const char *s);

/* catopen.c */
nl_catd catopen(const char *name, int mode);

/* cfgetispeed.c */
speed_t cfgetispeed(const struct termios *termios_p);

/* cfgetospeed.c */
speed_t cfgetospeed(const struct termios *termios_p);

/* cfree.c */
void cfree(void *p, size_t num, size_t size);

/* cfsetispeed.c */
int cfsetispeed(struct termios *termios_p, speed_t speed);

/* cfsetospeed.c */
int cfsetospeed(struct termios *termios_p, speed_t speed);

/* cftime.c */
int cftime(char *buf, char *format, const time_t *t);
int ascftime(char *buf, const char *format, const struct tm *tm);

/* clock.c */
clock_t clock(void);

/* closedir.c */
int closedir(DIR *dirp);
```

```
/* confstr.c */
size_t confstr(int name, char *buf, size_t length);

/* crypt.c */
void setkey(const char *key);
void encrypt(char *block, int fake);
char *crypt(const char *key, const char *salt);

/* csetlen.c */
int csetlen(int cset);
int csetcol(int cset);

/* ctime.c */
char *ctime(const time_t *t);
char *ctime_r(const time_t *, char *buf, int);
char *asctime(const struct tm *t);
char *asctime_r(const struct tm *, char *, int);

/* ctypefcns.c */
int isalpha(int c);
int isupper(int c);
int islower(int c);
int isdigit(int c);
int isxdigit(int c);
int isalnum(int c);
int isspace(int c);
int ispunct(int c);
int isprint(int c);
int isgraph(int c);
int iscntrl(int c);
int isascii(int c);
int _toupper(int c);
int _tolower(int c);
int toascii(int c);

/* daemon.c */
int daemon(int nochdir, int noclose);

/* directio.c */
int directio(int filedes, int advice);

/* dirname.c */
char *dirname(char *s);

/* div.c */
div_t div(int numer, int denom);
ldiv_t ldiv(long int numer, long int denom);

/* drand48.c */
double drand48(void);
double erand48(unsigned short *xsubi);
long krand48(unsigned short *xsubi, unsigned int m);
long lrand48(void);
long mrand48(void);
void srand48(long seedval);
unsigned short *seed48(unsigned short seed16v[3]);
void lcong48(unsigned short param[7]);
long nrand48(unsigned short *xsubi);
long jrand48(unsigned short *xsubi);

/* dup.c */
int dup(int fildes);
int dup2(int fildes, int fildes2);
int dup3(int fildes, int fildes2, int flags);

/* ecvt.c */
```

```
char *ecvt(double value, int ndigit, int *_RESTRICT_KYWD decpt,
        int *_RESTRICT_KYWDsign);
char *fcvt(double value, int ndigit, int *_RESTRICT_KYWD decpt,
        int *_RESTRICT_KYWD sign);

/* err.c */
void _errfp(FILE *, int, const char *, ...);
void _verrfp(FILE *, int, const char *, va_list);
void _errxfp(FILE *, int, const char *, ...);
void _verrxfp(FILE *, int, const char *, va_list);
void _warnfp(FILE *, const char *, ...);
void _vwarnfp(FILE *, const char *, va_list);
void _warnxfp(FILE *, const char *, ...);
void _vwarnxfp(FILE *, const char *, va_list);

/* errlst.c */

/* euclen.c */
int euccol(const unsigned char *s);
int euclen(const unsigned char *s);
int eucscol(const unsigned char *s);

/* execvp.c */
/* VARARGS1 */
int execlp(const char *, const char *, ...);
int execvp(const char *name, char *const *argv);

/* fattach.c */
int fattach(int fildes, const char *path);

/* fdetach.c */
int fdetach(const char *path);

/* ffs.c */
int ffs(int field);

/* fmtmsg.c */
int addseverity(int value, const char *string);
int fmtmsg(long class, const char *label, int severity, const char *text,
                const char *action, const char *tag);

/* ftime.c */
int ftime(struct timeb *tp);

/* ftok.c */
key_t ftok(const char *path, int id);

/* gcvt.c */
char *gcvt(double number, int ndigit, char *buf);

/* getcwd.c */
char *getcwd(char *str, size_t size);

/* getdate.c */
struct tm *getdate(const char *expression);
#ifdef getdate_err
#undef getdate_err
#endif
int getdate_err;

/* getdate_data.c */

/* getdate_gd.c */

/* getdtblsize.c */
int getdtablesize(void);
```

```
/* getenv.c */
char *getenv(const char *name);

/* getexecname.c */
const char *getexecname(void);

/* getgrnam.c */
struct group *getgrnam(const char *name);
struct group *getgrgid(gid_t gid);
struct group *fgetgrent_r(FILE *, struct group *, char *, int);
struct group *getgrent_r(struct group *, char *, int);
struct group *getgrgid_r(gid_t, struct group *, char *, int);
struct group *getgrnam_r(const char *, struct group *, char *, int);

/* gethostid.c */
long gethostid(void);

/* gethz.c */
int gethz(void);

/* getisax.c */
uint_t getisax(uint32_t *, uint_t);

/* getlogin.c */
char *getloginx(void);
char *getloginx_r(char *, int);
#ifdef  getlogin
#undef  getlogin
#endif  /* getlogin */
char *getlogin(void);
#ifdef  getlogin_r
#undef  getlogin_r
#endif  /* getlogin_r */
char *getlogin_r(char *, int);

/* getmntent.c */
int getmntany(FILE *fd, struct mnttab *mgetp, struct mnttab *mrefp);
int getmntent(FILE *fd, struct mnttab *mp);

/* getnetgrent.c */
int setnetgrent(const char *grp);
int endnetgrent(void);
int getnetgrent(char **machinep, char **namep, char **domainp);

/* getopt.c */
int getopt(int argc, char *const *argv, const char *opts);

/* getopt_long.c */
int getopt_clip(int argc, char *const *argv, const char *optstring,
                const struct option *long_options, int *long_index);
int getopt_long(int argc, char *const *argv, const char *optstring,
                const struct option *long_options, int *long_index);
int getopt_long_only(int argc, char *const *argv, const char *optstring,
                const struct option *long_options, int *long_index);

/* getpagesize.c */
int getpagesize(void);

/* getpw.c */
int getpw(uid_t uid, char *buf);

/* getpwnam.c */
struct passwd *getpwnam(const char *name);
struct passwd *getpwuid(uid_t uid);
struct passwd *fgetpwent_r(FILE *, struct passwd *, char *, int);
struct passwd *getpwent_r(struct passwd *, char *, int);
struct passwd *getpwnam_r(const char *, struct passwd *, char *, int);
struct passwd *getpwuid_r(uid_t, struct passwd *, char *, int);

/* getrusage.c */
int getrusage(int who, struct rusage *rusage);

/* gettimeofday.c */
int gettimeofday(struct timeval *_RESTRICT_KYWD tp, void *_RESTRICT_KYWD);

/* getspent.c */
void setspent(void);
void endspent(void);
struct spwd *getspent(void);
struct spwd *getspent_r(struct spwd *, char *, int);
struct spwd *fgetspent(FILE *f);
struct spwd *fgetspent_r(FILE *, struct spwd *, char *, int);
struct spwd *getspnam(const char *name);
struct spwd *getspnam_r(const char *, struct spwd *, char *, int);
int putspent(const struct spwd *p, FILE *f);

/* getspent_r.c */
int str2spwd(const char *, int, void *, char *, int);

/* getsubopt.c */
int getsubopt(char **optionsp, char *const *tokens, char **valuep);

/* gettxt.c */
char *gettxt(const char *msg_id, const char *dflt_str);

/* getusershell.c */
char *getusershell(void);
void endusershell(void);
void setusershell(void);

/* getut.c */
struct utmp *getutent(void);
struct utmp *getutid(const struct utmp *entry);
struct utmp *getutline(const struct utmp *entry);
struct utmp *pututline(const struct utmp *entry);
void setutent(void);
void endutent(void);
int utmpname(const char *newfile);
void updwtmp(const char *file, struct utmp *ut);
void getutmp(const struct utmpx *utx, struct utmp *ut);
void getutmpx(const struct utmp *ut, struct utmpx *utx);
struct utmp *makeut(struct utmp *utmp);

/* getutx.c */
struct utmpx *getutxent(void);
struct utmpx *getutxid(const struct utmpx *entry);
struct utmpx *getutxline(const struct utmpx *entry);
struct utmpx *pututxline(const struct utmpx *entry);
void setutxent(void);
void endutxent(void);
int utmpxname(const char *newfile);
void updwtmpx(const char *filex, struct utmpx *utx);
struct utmpx *makeutx(const struct utmpx *utmp);
struct utmpx *modutx(const struct utmpx *utp);

/* getvfsent.c */
int getvfsspec(FILE *fd, struct vfstab *vp, char *special);
int getvfsfile(FILE *fd, struct vfstab *vp, char *mountp);
int getvfsany(FILE *fd, struct vfstab *vgetp, struct vfstab *vrefp);
int getvfsent(FILE *fd, struct vfstab *vp);
```

```
/* getwd.c */
char *getwd(char *pathname);

/* getwidth.c */
void getwidth(eucwidth_t *eucstruct);

/* hsearch.c */
int hcreate(size_t size);
void hdestroy(void);
ENTRY *hsearch(ENTRY item, ACTION action);

/* iconv.c  */
size_t iconv(iconv_t cd, const char **_RESTRICT_KYWD inbuf,
    size_t *_RESTRICT_KYWD inbytesleft, char **_RESTRICT_KYWD outbuf,
    size_t *_RESTRICT_KYWD outbytesleft);
int iconv_close(iconv_t cd);
iconv_t iconv_open(const char *tocode, const char *fromcode);

/* imaxabs.c */
intmax_t imaxabs(intmax_t j);

/* imaxdiv.c */
imaxdiv_t imaxdiv(intmax_t numer, intmax_t denom);

/* index.c */
char *index(const char *sp, int c);

/* initgroups.c */
int initgroups(const char *uname, gid_t agroup);

/* innetgr.c */
int innetgr(const char *group, const char *machine, const char *name,
    const char *domain);

/* insque.c */
void insque(void *elem, void *pred);
void remque(void *elem);

/* isaexec.c */
int isaexec(const char *, char *const *, char *const *);

/* isastream.c */
int isastream(int fd);

/* isatty.c */
int isatty(int f);

/* killpg.c */
int killpg(pid_t pgrp, int sig);

/* l64a.c */
char *l64a(long lg);

/* lckpwdf.c */
int lckpwdf(void);
int ulckpwdf(void);

/* lfind.c */
void * lfind(const void *ky, const void *bs, size_t *nelp,
                size_t width, int (*compar)());

/* localeconv.c */
struct lconv *localeconv(void);

/* lsearch.c */
void * lsearch(const void *ky, void *bs, size_t *nelp,
                size_t width, int (*compar)());

/* madvise.c */
int madvise(caddr_t addr, size_t len, int advice);

/* malloc.c */
void *malloc(size_t size);
void *realloc(void *old, size_t size);
void free(void *old);

/* mbstowcs.c */
size_t mbstowcs(wchar_t *_RESTRICT_KYWD pwcs, const char *_RESTRICT_KYWD s,
        size_t n);

/* mbtowc.c */
int mbtowc(wchar_t *_RESTRICT_KYWD wchar, const char *_RESTRICT_KYWD s,
        size_t n);
int mblen(const char *s, size_t n);

/* memalign.c */
void *memalign(size_t align, size_t nbytes);

/* memccpy.c */
void *memccpy(void *_RESTRICT_KYWDs, const void *_RESTRICT_KYWD s0, int c,
        size_t n);

/* memchr.c */
void *memchr(const void *sptr, int c1, size_t n);

/* memcmp.c */
int memcmp(const void *s1, const void *s2, size_t n);

/* memcpy.c */
void *memcpy(void *_RESTRICT_KYWD s, const void *_RESTRICT_KYWD s0, size_t n);

/* memmove.c */
void *memmove(void *s, const void *s0, size_t n);

/* memset.c */
void *memset(void *sp1, int c, size_t n);

/* mkdev.c */
dev_t __makedev(const int version, const major_t majdev,
                const minor_t mindev);
major_t __major(const int version, const dev_t devnum);
minor_t __minor(const int version, const dev_t devnum);

/* mkfifo.c */
int mkfifo(const char *path, mode_t mode);

/* mktemp.c */
char *mktemp(char *as);

/* mlock.c */
int mlock(caddr_t addr, size_t len);

/* mlockall.c */
int mlockall(int flags);

/* mon.c */
void monitor(int (*alowpc)(), int (*ahighpc)(), WORD *buffer,
                size_t bufsize, size_t nfunc);

/* msync.c */
int msync(caddr_t addr, size_t len, int flags);
```

```
/* munlock.c */
int munlock(caddr_t addr, size_t len);

/* munlockall.c */
int munlockall(void);

/* ndbm.c */
void dbm_setdefwrite(DBM *db);
int dbm_flush(DBM *db);
int dbm_flushpag(DBM *db);
DBM *dbm_open(const char *file, int flags, mode_t mode);
void dbm_close(DBM *db);
int dbm_close_status(DBM *db);
datum dbm_fetch(DBM *db, datum key);
int dbm_delete(DBM *db, datum key);
int dbm_store(DBM *db, datum key, datum dat, int replace);
datum dbm_firstkey(DBM *db);
datum dbm_nextkey(DBM *db);
datum dbm_do_nextkey(DBM *db, datum inkey);

/* new_list.c */

/* nftw.c */
int nftw(const char *path, int (*fn)(), int depth, int flags);

/* nl_langinfo.c */
char *nl_langinfo(nl_item item);

/* opendir.c */
DIR *opendir(const char *filename);

/* opt_data.c */

/* perror.c */
void perror(const char *s);

/* pipe.c */
int pipe(int *fds);

/* psiginfo.c */
void psiginfo(siginfo_t *sip, char *s);

/* psignal.c */
void psignal(int sig, const char *s);

/* pt.c */
char *ptsname(int fd);
int unlockpt(int fd);
int grantpt(int fd);

/* putenv.c */
int putenv(char *change);
int setenv(const char *envname, const char *envval, int overwrite);
int unsetenv(const char *name);

/* putpwent.c */
int putpwent(const struct passwd *p, FILE *f);

/* qsort.c */
void qsort(void *base, size_t n, size_t size, int (*compar)());

/* raise.c */
int raise(int sig);

/* rand.c */
void srand(unsigned x);
```

```
int rand(void);
int rand_r(unsigned int *);

/* random.c */
void srandom(unsigned x);
char *initstate(unsigned seed, char *arg_state, size_t n);
char *setstate(const char *arg_state);
long random(void);

/* rctlops.c */
int rctl_walk(int (*callback)(const char *, void *), void *walk_data);
hrtime_t rctlblk_get_firing_time(rctlblk_t *rblk);
uint_t rctlblk_get_global_action(rctlblk_t *rblk);
uint_t rctlblk_get_global_flags(rctlblk_t *rblk);
uint_t rctlblk_get_local_action(rctlblk_t *rblk, int *signalp);
uint_t rctlblk_get_local_flags(rctlblk_t *rblk);
id_t rctlblk_get_recipient_pid(rctlblk_t *rblk);
rctl_priv_t rctlblk_get_privilege(rctlblk_t *rblk);
rctl_qty_t rctlblk_get_value(rctlblk_t *rblk);
void rctlblk_set_local_action(rctlblk_t *rblk, uint_t action, int signal);
void rctlblk_set_local_flags(rctlblk_t *rblk, uint_t flags);
void rctlblk_set_privilege(rctlblk_t *rblk, rctl_priv_t priv);
void rctlblk_set_value(rctlblk_t *rblk, rctl_qty_t val);
size_t rctlblk_size(void);

/* readdir.c */
struct dirent *readdir(DIR *dirp);

/* realpath.c */
char *realpath(const char *_RESTRICT_KYWD raw, char *_RESTRICT_KYWD canon);

/* regexpr.c */
char *re_comp(const char *sp);
int re_exec(const char *p1);

/* rindex.c */
char *rindex(const char *sp, int c);

/* rename.c */
int remove(const char *filename);
int rename(const char *old, const char *new);

/* rewinddir.c */
#undef rewinddir
void rewinddir(DIR *dirp);

/* scandir.c */
int alphasort(const struct dirent **, const struct dirent **);
int scandir(const char *dirname, struct dirent *(*namelist[]),
        int (*select)(const struct dirent *),
        int (*dcomp)(const struct dirent **, const struct dirent **));

/* scrwidth.c */
int scrwidth(wchar_t c);

/* seekdir.c */
void seekdir(DIR *dirp, long loc);

/* select.c */
int pselect(int nfds,
        fd_set *_RESTRICT_KYWD readfds,
        fd_set *_RESTRICT_KYWD writefds,
        fd_set *_RESTRICT_KYWD errorfds,
        const struct timespec *_RESTRICT_KYWD timeout,
        const sigset_t *_RESTRICT_KYWD sigmask);
int select(int nfds,
```

```
        fd_set *_RESTRICT_KYWD readfds,
        fd_set *_RESTRICT_KYWD writefds,
        fd_set *_RESTRICT_KYWD errorfds,
        struct timeval *_RESTRICT_KYWD timeout);

/* setlocale.c */
char *setlocale(int cat, const char *loc);

/* setpriority.c */
int getpriority(int which, id_t who);
int setpriority(int which, id_t who, int prio);

/* settimeofday.c */
int settimeofday(struct timeval *tp, void *);

/* sigflag.c */
int sigflag(int sig, int flag, int on);

/* siglist.c */

/* sigsend.c */
int sigsend(idtype_t idtype, id_t id, int sig);

/* sigsetops.c */
int sigfillset(sigset_t *set);
int sigemptyset(sigset_t *set);
int sigaddset(sigset_t *set, int sig);
int sigdelset(sigset_t *set, int sig);
int sigismember(const sigset_t *set, int sig);

/* scalls.c */
unsigned sleep(unsigned sleep_tm);

/* ssignal.c */
int (*ssignal(int sig, int (*fn)())) ();
int gsignal(int sig);

/* str2id.c */

/* str2sig.c */
int str2sig(const char *s, int *sigp);
int sig2str(int i, char *s);

/* strcat.c */
char *strcat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

/* strchr.c */
char *strchr(const char *sp, int c);

/* strcmp.c */
int strcmp(const char *s1, const char *s2);

/* strcpy.c */
char *strcpy(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

/* strcspn.c */
size_t strcspn(const char *string, const char *charset);

/* strdup.c */
char *strdup(const char *s1);

/* strerror.c */
char *strerror(int errnum);
int strerror_r(int errnum, char *strerrbuf, size_t buflen);

/* strftime.c */
```

```
size_t strftime(char *_RESTRICT_KYWD s, size_t maxsize,
                const char *_RESTRICT_KYWD format,
                const struct tm *_RESTRICT_KYWD tm);

/* strlen.c */
size_t strlen(const char *s);

/* strncat.c */
char *strncat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2, size_t n);

/* strncmp.c */
int strncmp(const char *s1, const char *s2, size_t n);

/* strncpy.c */
char *strncpy(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2, size_t n);

/* strpbrk.c */
char *strpbrk(const char *string, const char *brkset);

/* strrchr.c */
char *strrchr(const char *sp, int c);

/* strsep.c */
char *strsep(char **stringp, const char *delim);

/* strspn.c */
size_t strspn(const char *string, const char *charset);

/* strstr.c */
char *strstr(const char *as1, const char *as2);

/* strtod.c */
double strtod(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
float strtof(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
long double strtold(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);

/* strtoimax.c */
intmax_t strtoimax(const char *_RESTRICT_KYWD nptr,
        char **_RESTRICT_KYWD endptr, int base);

/* strtok.c */
char *strtok(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD sepset);
char *strtok_r(char *_RESTRICT_KYWD, const char *_RESTRICT_KYWD,
        char **_RESTRICT_KYWD);

/* strtol.c */
long strtol(const char *_RESTRICT_KYWD str, char **_RESTRICT_KYWD nptr,
        int base);

/* strtoul.c */
unsigned long strtoul(const char *_RESTRICT_KYWD str,
        char **_RESTRICT_KYWD nptr, int base);

/* strtoumax.c */
uintmax_t strtoumax(const char *_RESTRICT_KYWD nptr,
                    char **_RESTRICT_KYWD endptr, int base);

/* strxfrm.c */
size_t strxfrm(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2,
        size_t n);
int strcoll(const char *s1, const char *s2);

/* swab.c */
void swab(const char *_RESTRICT_KYWD from, char *_RESTRICT_KYWD to, ssize_t n);

/* swapctl.c */
```

```
int swapctl(int cmd, void *arg);

/* sysconf.c */
long sysconf(int name);

/* syslog.c */
/* VARARGS2 */
void syslog(int pri, const char *fmt, ...);
void vsyslog(int pri, const char *fmt, va_list ap);
void openlog(const char *ident, int logstat, int logfac);
void closelog(void);
int setlogmask(int pmask);

/* tcdrain.c */
int tcdrain(int fildes);

/* tcflow.c */
int tcflow(int fildes, int action);

/* tcflush.c */
int tcflush(int fildes, int queue_selector);

/* tcgetattr.c */
int tcgetattr(int fildes, struct termios *termios_p);

/* tcgetpgrp.c */
pid_t tcgetpgrp(int fd);

/* tcgetsid.c */
pid_t tcgetsid(int fd);

/* tcsendbreak.c */
int tcsendbreak(int fildes, int duration);

/* tcsetattr.c */
int tcsetattr(int fildes, int optional_actions,
        const struct termios *termios_p);

/* tcsetpgrp.c */
int tcsetpgrp(int fd, pid_t pgrp);

/* tell.c */
long tell(int f);

/* telldir.c */
long telldir(DIR *dirp);

/* tfind.c */
void *tfind(const void *ky, void *const *rtp, int (*compar)());

/* time_comm.c */
struct tm *localtime(const time_t *timep);
struct tm *localtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
struct tm *gmtime(const time_t *clock);
struct tm *gmtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
double difftime(time_t time1, time_t time0);
time_t mktime(struct tm *timeptr);
void _ltzset(time_t tim);
void tzset(void);

/* time_data.c */

/* time_gdata.c */

/* tolower.c */
int tolower(int c);
```

```
/* toupper.c */
int toupper(int c);

/* truncate.c */
int ftruncate(int fildes, off_t len);
int truncate(const char *path, off_t len);

/* tsearch.c */
void *tsearch(const void *ky, void **rtp, int (*compar)());
void *tdelete(const void *ky, void **rtp, int (*compar)());
void twalk(const void *rt, void (*action)());

/* ttyname.c */
char *ttyname(int f);
char *_ttyname_dev(dev_t rdev, char *buffer, size_t buflen);
char *ttyname_r(int, char *, int);

/* ttyslot.c */
int ttyslot(void);

/* ualarm.c */
unsigned ualarm(unsigned usecs, unsigned reload);

/* ulimit.c */
/* VARARGS1 */
long ulimit(int cmd, ...);

/* scalls.c */
int usleep(unsigned n);

/* valloc.c */
void *valloc(size_t size);

/* waitpid.c */
pid_t wait(int *stat_loc);
pid_t waitpid(pid_t pid, int *stat_loc, int options);
pid_t wait3(int *status, int options, struct rusage *rp);
pid_t wait4(pid_t pid, int *status, int options, struct rusage *rusage);

/* wcstombs.c */
size_t wcstombs(char *_RESTRICT_KYWD s, const wchar_t *_RESTRICT_KYWD pwcs,
        size_t n);

/* wctomb.c */
int wctomb(char *s, wchar_t wchar);

/* wdata.c */

/* wisprint.c */
int wisprint(wchar_t c);

/* xgetwidth.c */
void _xgetwidth(void);

/*
 * /usr/src/lib/libc/port/intl routines
 */

/* gettext.c */
char *bindtextdomain(const char *domain, const char *binding);
char *dcgettext(const char *domain, const char *msg_id, const int category);
char *dgettext(const char *domain, const char *msg_id);
char *gettext(const char *msg_id);
char *textdomain(const char *domain);
```

```
/*
 * /usr/src/lib/libc/port/print routines
 */

/* fprintf.c */
/* VARARGS2 */
int fprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format, ...);

/* printf.c */
/* VARARGS1 */
int printf(const char *_RESTRICT_KYWD format, ...);

/* snprintf.c */
/* VARARGS2 */
int snprintf(char *_RESTRICT_KYWD string, size_t n,
                const char *_RESTRICT_KYWD format, ...);

/* sprintf.c */
/* VARARGS2 */
int sprintf(char *_RESTRICT_KYWD string,
                const char *_RESTRICT_KYWD format, ...);

/* vfprintf.c */
/* VARARGS2 */
int vfprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format,
                va_list);

/* vprintf.c */
/* VARARGS1 */
int vprintf(const char *_RESTRICT_KYWD format, va_list);

/* vsnprintf.c */
/* VARARGS2 */
int vsnprintf(char *_RESTRICT_KYWD string, size_t n,
                const char *_RESTRICT_KYWD format, va_list);

/* vsprintf.c */
/* VARARGS2 */
int vsprintf(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD format,
        va_list);

/*
 * /usr/src/lib/libc/port/regex routines
 */

/* glob.c */
extern int glob(const char *restrict pattern, int flags,
    int(*errfunc)(const char *epath, int eerrno), glob_t *restrict pglob);
extern void globfree(glob_t *pglob);

/* regex.c */
char *regex(const char *regexp, const char *stringp, ...);
#ifdef __loc1
#undef __loc1
#endif
char *__loc1;

/* regcmp.c */
char *regcmp(const char *regexp, ...);
#ifdef __i_size
#undef __i_size
#endif
int __i_size;

/*
 * /usr/src/lib/libc/port/stdio routines
 */

/* _filbuf.c */
int _filbuf(FILE *iop);

/* _flsbuf.c */
int _flsbuf(int ch, FILE *iop);

/* _wrtchk.c */
int _wrtchk(FILE *iop);

/* clearerr.c */
void clearerr(FILE *iop);

/* ctermid.c */
char *ctermid(char *s);
char *ctermid_r(char *s);

/* cuserid.c */
char *cuserid(char *s);

/* data.c */

/* doscan.c */
int _doscan(FILE *iop, const char *fmt, va_list va_alist);

/* fdopen.c */
FILE *fdopen(int fd, const char *type);

/* feof.c */
int feof(FILE *iop);

/* ferror.c */
int ferror(FILE *iop);

/* fgetc.c */
int fgetc(FILE *iop);

/* fgets.c */
char *fgets(char *_RESTRICT_KYWD buf, int size, FILE *_RESTRICT_KYWD iop);

/* fileno.c */
int _fileno(FILE *iop);

/* flush.c */
void _cleanup(void);
FILE *_findiop(void);
typedef unsigned char Uchar;
void _setbufend(FILE *iop, Uchar *end);
Uchar *_realbufend(FILE *iop);
void _bufsync(FILE *iop, Uchar *bufend);
int _xflsbuf(FILE *iop);
int fflush(FILE *iop);
int fclose(FILE *iop);

/* fopen.c */
FILE *fopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type);
FILE *freopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type,
                FILE *_RESTRICT_KYWD iop);

/* fpos.c */
int fgetpos(FILE *_RESTRICT_KYWD stream, fpos_t *_RESTRICT_KYWD pos);
int fsetpos(FILE *stream, const fpos_t *pos);

/* fputc.c */
int fputc(int ch, FILE *iop);
```

```
/* fputs.c */
int fputs(const char *_RESTRICT_KYWD ptr, FILE *_RESTRICT_KYWD iop);

/* fread.c */
size_t fread(void *_RESTRICT_KYWD ptr, size_t size, size_t count,
        FILE *_RESTRICT_KYWD iop);

/* fseek.c */
int fseek(FILE *iop, long offset, int ptrname);

/* ftell.c */
long ftell(FILE *iop);

/* fwrite.c */
size_t fwrite(const void *_RESTRICT_KYWD ptr1, size_t size, size_t count,
        FILE *_RESTRICT_KYWD iop);

/* getc.c */
int getc(FILE *iop);

/* getchar.c */
int getchar(void);

/* getpass.c */
char *getpass(const char *prompt);

/* getpass.c */
char *getpassphrase(const char *prompt);

/* gets.c */
char *gets(char *buf);

/* getw.c */
int getw(FILE *stream);

/* popen.c */
FILE *popen(const char *cmd, const char *mode);
int pclose(FILE *ptr);

/* putc.c */
int putc(int ch, FILE *iop);

/* putchar.c */
int putchar(int ch);

/* puts.c */
int puts(const char *ptr);

/* putw.c */
int putw(int w, FILE *stream);

/* rewind.c */
void rewind(FILE *iop);

/* scanf.c */
/* VARARGS1 */
int scanf(const char *_RESTRICT_KYWD fmt, ...);

/* VARARGS2 */
int fscanf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD fmt, ...);

/* VARARGS2 */
int sscanf(const char *_RESTRICT_KYWD str, const char *_RESTRICT_KYWD fmt, ...);

/* setbuf.c */
```

```
void setbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf);

/* setvbuf.c */
int setvbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf, int type,
                size_t size);

/* system.c */
int system(const char *s);

/* tempnam.c */
char *tempnam(const char *dir, const char *pfx);

/* tmpfile.c */
FILE *tmpfile(void);

/* tmpnam.c */
char *tmpnam(char *s);
char *tmpnam_r(char *);

/* ungetc.c */
int ungetc(int c, FILE *iop);

/*
 * /usr/src/lib/libc/port/sys routines
 */

/* exacctsys.c */
size_t getacct(idtype_t idtype, id_t id, void *buf, size_t bufsize);
int putacct(idtype_t idtype, id_t id, void *buf, size_t bufsize, int flags);
int wracct(idtype_t idtype, id_t id, int flags);

/* execl.c */
/* VARARGS1 */
int execl(const char *name, const char *, ...);

/* execle.c */
int execle(const char *, const char *file, ...);

/* execv.c */
int execv(const char *file, char *const *argv);

/* lockf.c */
int lockf(int fildes, int function, off_t size);

/* meminfosys.c */
int meminfo(const uint64_t *inaddr, int addr_count, const uint_t *info_req,
        int info_count, uint64_t *outdata, uint_t *validity);

/* msgsys.c */
int msgget(key_t key, int msgflg);
int msgctl(int msqid, int cmd, struct msqid_ds *buf);
ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);
int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);

/* nfssys.c */
/*
int exportfs(char *dir, struct export *ep);
int nfs_getfh(char *path, fhandle_t *fhp);
int nfssvc(int fd);
*/

/* psetsys.c */
int pset_create(psetid_t *npset);
int pset_destroy(psetid_t pset);
int pset_assign(psetid_t pset, processorid_t cpu, psetid_t *opset);
int pset_assign_forced(psetid_t pset, processorid_t cpu, psetid_t *opset);
```

```
int pset_info(psetid_t pset, int *type, u_int *numcpus, processorid_t *cpulist);
int pset_bind(psetid_t pset, idtype_t idtype, id_t id, psetid_t *opset);
int pset_bind_lwp(psetid_t pset, id_t id, pid_t, psetid_t *opset);


/* rctlsys.c */
int getrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
    int flags);
int setrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
    int flags);
/* (private functions) */
int setprojrctl(const char *name, rctlblk_t *new_rblk, size_t size, int flags);
int rctlctl(const char *, rctlblk_t *, int);
size_t rctllist(char *, size_t);


/* semsys.c */
int semctl(int semid, int semnum, int cmd, ...);
int semget(key_t key, int nsems, int semflg);
int semop(int semid, struct sembuf *sops, size_t nsops);

/* shmsys.c */
void *shmat(int shmid, const void *shmaddr, int shmflg);
int shmctl(int shmid, int cmd, struct shmid_ds *buf);
#if defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4)
int shmdt(const void *);
#else
int shmdt(char *);
#endif /* defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4) */
int shmget(key_t key, size_t size, int shmflg);

/* tasksys.c */
taskid_t settaskid(projid_t project, uint_t flags);
taskid_t gettaskid(void);
projid_t getprojid(void);

/*
 * /usr/src/lib/libc/port/widec routines
 */

/* fgetws.c */
wchar_t *fgetws(wchar_t *_RESTRICT_KYWD ptr, int  size,
        FILE *_RESTRICT_KYWD iop);

/* fputwc.c */
wint_t fputwc(wint_t wc, FILE *iop);
wint_t putwc(wint_t wc, FILE *iop);

/* fputws.c */
int fputws(const wchar_t *_RESTRICT_KYWD ptr, FILE *_RESTRICT_KYWD iop);

/* getwchar.c */
wint_t getwchar(void);

/* getwidth.c */
void getwidth(eucwidth_t *eucstruct);

/* getws.c */
wchar_t *getws(wchar_t *ptr);

/* iswctype.c */
int iswctype(wint_t wc, wctype_t charclass);
int iswalpha(wint_t c);
int iswupper(wint_t c);
int iswlower(wint_t c);
int iswdigit(wint_t c);
int iswxdigit(wint_t c);
int iswalnum(wint_t c);
int iswspace(wint_t c);
int iswpunct(wint_t c);
int iswprint(wint_t c);
int iswgraph(wint_t c);
int iswcntrl(wint_t c);
int isphonogram(wint_t c);
int isideogram(wint_t c);
int isenglish(wint_t c);
int isnumber(wint_t c);
int isspecial(wint_t c);

/* libwcollate.c */

/* putwchar.c */
wint_t putwchar(wint_t c);

/* putws.c */
int putws(const wchar_t *ptr);

/* scrwidth.c */

/* strtows.c */
wchar_t *strtows(wchar_t *s1, char *s2);
char *wstostr(char *s1, wchar_t *s2);

/* trwctype.c */
wint_t towupper(wint_t c);
wint_t towlower(wint_t c);

/* ungetwc.c */
wint_t ungetwc(wint_t wc, FILE *iop);

/* wcollate.c */
size_t wcsxfrm(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
        size_t n);
int wcscoll(const wchar_t *s1, const wchar_t *s2);

/* wcsftime.c */
#if !defined(__amd64)           /* XX64 - fix me */
size_t wcsftime(wchar_t *wcs, size_t maxsize,
        const char *format, const struct tm *timeptr);
#endif  /* __amd64 */

/* wcstring.c */
wint_t fgetwc(FILE *iop);
wint_t getwc(FILE *iop);
int wcwidth(wchar_t wc);
int wcswidth(const wchar_t *pwcs, size_t n);

/* wcswcs.c */
wchar_t *wcswcs(const wchar_t *ws1, const wchar_t *ws2);

/* wcsxfrm.c - empty file! */

/* wcsxfrm.xpg4.c */

/* wisprint.c */
int wisprint(wchar_t c);

/* wscasecmp.c */
int wscasecmp(const wchar_t *s1, const wchar_t *s2);

/* wscat.c */
wchar_t *wcscat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
```

```
wchar_t *wscat(wchar_t *s1, const wchar_t *s2);

/* wschr.c */
wchar_t *wcschr(const wchar_t *sp, wchar_t c);
wchar_t *wschr(const wchar_t *sp, wchar_t c);

/* wscmp.c */
int wcscmp(const wchar_t *s1, const wchar_t *s2);
int wscmp(const wchar_t *s1, const wchar_t *s2);

/* wscol.c */
int wscol(const wchar_t *s1);

/* wscpy.c */
wchar_t *wcscpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
wchar_t *wscpy(wchar_t *s1, const wchar_t *s2);

/* wscspn.c */
size_t wcscspn(const wchar_t *string, const wchar_t *charset);
size_t wscspn(const wchar_t *string, const wchar_t *charset);

/* wsdup.c */
wchar_t *wsdup(const wchar_t *s1);

/* wslen.c */
size_t wcslen(const wchar_t *s);
size_t wslen(const wchar_t *s);

/* wsncasecmp.c */
int wsncasecmp(const wchar_t *s1, const wchar_t *s2, size_t n);

/* wsncat.c */
wchar_t *wcsncat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
        size_t n);
wchar_t *wsncat(wchar_t *s1, const wchar_t *s2, size_t n);

/* wsncmp.c */
int wcsncmp(const wchar_t *s1, const wchar_t *s2, size_t n);
int wsncmp(const wchar_t *s1, const wchar_t *s2, size_t n);

/* wsncpy.c */
wchar_t *wcsncpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
        size_t n);
wchar_t *wsncpy(wchar_t *s1, const wchar_t *s2, size_t n);

/* wspbrk.c */
wchar_t *wcspbrk(const wchar_t *string, const wchar_t *brkset);
wchar_t *wspbrk(const wchar_t *string, const wchar_t *brkset);

/* wsprintf.c */
int wsprintf(wchar_t *wstring, const char *format, ...);

/* wsrchr.c */
wchar_t *wcsrchr(const wchar_t *sp, wchar_t c);
wchar_t *wsrchr(const wchar_t *sp, wchar_t c);

/* wsscanf.c */
int wsscanf(wchar_t *s, const char *format, ...);

/* wssize.c */

/* wsspn.c */
size_t wcsspn(const wchar_t *string, const wchar_t *charset);
size_t wsspn(const wchar_t *string, const wchar_t *charset);

/* wstod.c */
```

```
double wcstod(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
float wcstof(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
long double wcstold(const wchar_t *_RESTRICT_KYWD cp,
        wchar_t **_RESTRICT_KYWD ptr);
double wstod(const wchar_t *cp, wchar_t **ptr);

/* wstok.c */
#if !defined(__amd64)           /* XX64 - fix me */
wchar_t *wcstok(wchar_t *string, const wchar_t *sepset);
wchar_t *wstok(wchar_t *string, const wchar_t *sepset);
#endif  /* __amd64 */

/* wcstol.c */
long wcstol(const wchar_t *_RESTRICT_KYWD str, wchar_t **_RESTRICT_KYWD ptr,
        int base);
long long wcstoll(const wchar_t *_RESTRICT_KYWD str,
        wchar_t **_RESTRICT_KYWD ptr, int base);

/* wcstoul.c */
unsigned long wcstoul(const wchar_t *_RESTRICT_KYWD str,
        wchar_t **_RESTRICT_KYWD ptr, int base);
unsigned long long wcstoull(const wchar_t *_RESTRICT_KYWD str,
        wchar_t **_RESTRICT_KYWD ptr, int base);

/* wcstoimax.c */
intmax_t wcstoimax(const wchar_t *_RESTRICT_KYWD nptr,
        wchar_t **_RESTRICT_KYWD endptr, int base);
uintmax_t wcstoumax(const wchar_t *_RESTRICT_KYWD nptr,
        wchar_t **_RESTRICT_KYWD endptr, int base);

/* wstol.c */
long wstol(const wchar_t *str, wchar_t **ptr, int base);

/* wstoll.c */
long long wstoll(const wchar_t *str, wchar_t **ptr, int base);
long long watoll(const wchar_t *p);

/* wsxfrm.c */
size_t wsxfrm(wchar_t *s1, const wchar_t *s2, size_t n);
int wscoll(const wchar_t *s1, const wchar_t *s2);

/*
 * /usr/src/lib/libc/port/gen/event_port.c
 */
int port_dispatch(int port, int flags, int source, int events, uintptr_t object,
    void *user);

/*
 * /usr/src/lib/libc/$MACH/gen routines
 */

/* alloca.s */

void *__builtin_alloca(size_t);

/*
 * modctl(int arg, ...) and utssys(...) are not available from a header
 * file, but our utilities which make use of it should be able to be
 * lint clean.
 */
int modctl(int arg, ...);
int utssys(void *buf, int arg, int type, void *outbp);


typedef float single;
typedef unsigned extended[3];
```

```
typedef long double quadruple;
typedef unsigned fp_exception_field_type;

typedef char decimal_string[512];

enum fp_class_type {
        fp_zero  = 0,
        fp_subnormal    = 1,
        fp_normal       = 2,
        fp_infinity     = 3,
        fp_quiet        = 4,
        fp_signaling    = 5
};
```
______***unchanged_portion_omitted_***

****
   54824 Mon Aug 12 18:24:55 2013
new/usr/src/lib/libc/port/mapfile-vers
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
****

```
#
# 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 2010 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright (c) 2013 Gary Mills
#

#
# 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.
#

#
# README README README README README README: how to update this file
#   1) each version of Solaris/OpenSolaris gets a version number.
#      (Actually since Solaris is actually a series of OpenSolaris releases
#       we'll just use OpenSolaris for this exercise.)
#       OpenSolaris 2008.11 gets 1.23
```

```
#       OpenSolaris 2009.04 gets 1.24
#       etc.
#   2) each project integration uses a unique version number.
#       PSARC/2008/123 gets 1.24.1
#       PSARC/2008/456 gets 1.24.2
#       etc.
#


# 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_0.5 {    # common C++ ABI exit handlers
    protected:
        __cxa_atexit;
        __cxa_finalize;
} ILLUMOS_0.4;
```
______***unchanged_portion_omitted_***

```
# There should never be more than one SUNWprivate version.
# Don't add any more.  Add new private symbols to SUNWprivate_1.1

SYMBOL_VERSION SUNWprivate_1.1 {
    global:
        ___Argv                 { FLAGS = NODIRECT };
        cfree                   { FLAGS = NODIRECT };
        _cswidth;
        __ctype_mask;
        __environ_lock          { FLAGS = NODIRECT };
        __inf_read;
        __inf_written;
        __i_size;
        _isnanf                 { TYPE = FUNCTION; FILTER = libm.so.2 };
        __iswrune;
        __libc_threaded;
        _lib_version            { FLAGS = NODIRECT };
        _logb                   { TYPE = FUNCTION; FILTER = libm.so.2 };
        _lone                   { FLAGS = NODYNSORT };
        _lten                   { FLAGS = NODYNSORT };
        _lzero                  { FLAGS = NODYNSORT };
        __malloc_lock;
        _memcmp;
        _memcpy                 { FLAGS = NODYNSORT };
        _memmove;
        _memset;
        _modff                  { TYPE = FUNCTION; FILTER = libm.so.2 };
        __nan_read;
        __nan_written;
        __nextwctype;
        __nis_debug_bind;
        __nis_debug_calls;
        __nis_debug_file;
        __nis_debug_rpc;
        __nis_prefsrv;
        __nis_preftype;
        __nis_server;
        _nss_default_finders;
        __progname              { FLAGS = NODIRECT };
        _smbuf;
        _sp;
        __strdupa_str           { FLAGS = NODIRECT };
        __strdupa_len           { FLAGS = NODIRECT };
        _tdb_bootstrap;
        __threaded;
        thr_probe_getfunc_addr;
        __trans_lower;
        __trans_upper;
        _uberdata;
        __xpg6                  { FLAGS = NODIRECT };

$if _ELF32
        _dladdr                 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dladdr1                { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlclose                { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dldump                 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlerror                { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlinfo                 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlmopen                { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlopen                 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _dlsym                  { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _ld_libc                { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
        _sys_errlist;
        _sys_errs;
        _sys_index;
        _sys_nerr               { FLAGS = NODYNSORT };
        _sys_num_err;
$elif sparcv9
        _dladdr         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dladdr1        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlclose        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dldump         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlerror        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlinfo         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlmopen        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlopen         { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _dlsym          { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
        _ld_libc        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
$elif amd64
        _dladdr         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dladdr1        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlamd64getunwind { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlclose        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dldump         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlerror        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlinfo         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlmopen        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlopen         { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
        _dlsym          { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
```

```
        _ld_libc        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
$else
$error unknown platform
$endif

$if _sparc
        __lyday_to_month;
        __mon_lengths;
        __yday_to_month;
$endif
$if i386
        _sse_hw;
$endif

    protected:
        acctctl;
        allocids;
        _assert_c99;
        __assert_c99;
        _assfail;
        attr_count;
        attr_to_data_type;
        attr_to_name;
        attr_to_option;
        attr_to_xattr_view;
        _autofssys;
        _bufsync;
        _cladm;
        __class_quadruple;
        core_get_default_content;
        core_get_default_path;
        core_get_global_content;
        core_get_global_path;
        core_get_options;
        core_get_process_content;
        core_get_process_path;
        core_set_default_content;
        core_set_default_path;
        core_set_global_content;
        core_set_global_path;
        core_set_options;
        core_set_process_content;
        core_set_process_path;
        dbm_close_status;
        dbm_do_nextkey;
        dbm_setdefwrite;
        _D_cplx_div;
        _D_cplx_div_ix;
        _D_cplx_div_rx;
        _D_cplx_mul;
        defclose_r;
        defcntl;
        defcntl_r;
        defopen;
        defopen_r;
        defread;
        defread_r;
        _delete;
        _dgettext;
        _doprnt;
        _doscan;
        _errfp;
        _errxfp;
        exportfs;
        _F_cplx_div;
        _F_cplx_div_ix;
```

```
	_F_cplx_div_rx;
	_F_cplx_mul;
	__fgetwc_xpg5;
	__fgetws_xpg5;
	_findbuf;
	_findiop;
	__fini_daemon_priv;
	_finite;
	_fork1			{ FLAGS = NODYNSORT };
	_forkall		{ FLAGS = NODYNSORT };
	_fpclass;
	_fpgetmask;
	_fpgetround;
	_fpgetsticky;
	_fprintf;
	_fpsetmask;
	_fpsetround;
	_fpsetsticky;
	__fputwc_xpg5;
	__fputws_xpg5;
	_ftw;
	_gcvt;
	_getarg;
	__getcontext;
	_getdents;
	_get_exit_frame_monitor;
	_getfp;
	_getgroupsbymember;
	_getlogin_r;
	getloginx;
	getloginx_r;
	_getsp;
	__gettsp;
	getvmusage;
	__getwchar_xpg5;
	__getwc_xpg5;
	gtty;
	__idmap_flush_kcache;
	__idmap_reg;
	__idmap_unreg;
	__init_daemon_priv;
	__init_suid_priv;
	_insert;
	inst_sync;
	_iswctype;
	klpd_create;
	klpd_getpath;
	klpd_getport;
	klpd_getucred;
	klpd_register;
	klpd_register_id;
	klpd_unregister;
	klpd_unregister_id;
	_lgrp_home_fast		{ FLAGS = NODYNSORT };
	_lgrpsys;
	_lltostr;
	_lock_clear;
	_lock_try;
	_ltzset;
	lwp_self;
	makeut;
	makeutx;
	_mbftowc;
	mcfiller;
	mntopt;
	modctl;
```

```
	modutx;
	msgctl64;
	__multi_innetgr;
	_mutex_destroy		{ FLAGS = NODYNSORT };
	mutex_held;
	_mutex_init		{ FLAGS = NODYNSORT };
	_mutex_unlock		{ FLAGS = NODYNSORT };
	name_to_attr;
	nfs_getfh;
	nfssvc;
	_nfssys;
	__nis_get_environment;
	_nss_db_state_destr;
	nss_default_key2str;
	nss_delete;
	nss_endent;
	nss_getent;
	_nss_initf_group;
	_nss_initf_netgroup;
	_nss_initf_passwd;
	_nss_initf_shadow;
	nss_packed_arg_init;
	nss_packed_context_init;
	nss_packed_getkey;
	nss_packed_set_status;
	nss_search;
	nss_setent;
	_nss_XbyY_fgets;
	__nsw_extended_action_v1;
	__nsw_freeconfig_v1;
	__nsw_getconfig_v1;
	__nthreads;
	__openattrdirat;
	option_to_attr;
	__posix_getloginx_r;
	__priv_bracket;
	__priv_relinquish;
	pset_assign_forced;
	pset_bind_lwp;
	_psignal;
	_pthread_setcleanupinit;
	__putwchar_xpg5;
	__putwc_xpg5;
	rctlctl;
	rctllist;
	_realbufend;
	_resume;
	_resume_ret;
	_rpcsys;
	_sbrk_grow_aligned;
	scrwidth;
	semctl64;
	_semctl64;
	set_setcontext_enforcement;
	_setbufend;
	__set_errno;
	setprojrctl;
	_setregid;
	_setreuid;
	setsigacthandler;
	shmctl64;
	_shmctl64;
	sigflag;
	_signal;
	_sigoff;
	_sigon;
```

```
        _so_accept;
        _so_bind;
        _sockconfig;
        _so_connect;
        _so_getpeername;
        _so_getsockname;
        _so_getsockopt;
        _so_listen;
        _so_recv;
        _so_recvfrom;
        _so_recvmsg;
        _so_send;
        _so_sendmsg;
        _so_sendto;
        _so_setsockopt;
        _so_shutdown;
        _so_socket;
        _so_socketpair;
        str2group;
        str2passwd;
        str2spwd;
        __strptime_dontzero;
        stty;
        syscall;
        _sysconfig;
        __systemcall;
        thr_continue_allmutators;
        _thr_continue_allmutators;
        thr_continue_mutator;
        _thr_continue_mutator;
        thr_getstate;
        _thr_getstate;
        thr_mutators_barrier;
        _thr_mutators_barrier;
        thr_probe_setup;
        _thr_schedctl;
        thr_setmutator;
        _thr_setmutator;
        thr_setstate;
        _thr_setstate;
        thr_sighndlrinfo;
        _thr_sighndlrinfo;
        _thr_slot_offset;
        thr_suspend_allmutators;
        _thr_suspend_allmutators;
        thr_suspend_mutator;
        _thr_suspend_mutator;
        thr_wait_mutator;
        _thr_wait_mutator;
        __tls_get_addr;
        tpool_create;
        tpool_dispatch;
        tpool_destroy;
        tpool_wait;
        tpool_suspend;
        tpool_suspended;
        tpool_resume;
        tpool_member;
        _ttyname_dev;
        _ucred_alloc;
        ucred_getamask;
        _ucred_getamask;
        ucred_getasid;
        _ucred_getasid;
        ucred_getatid;
        _ucred_getatid;
```

```
        ucred_getauid;
        _ucred_getauid;
        _ulltostr;
        _uncached_getgrgid_r;
        _uncached_getgrnam_r;
        _uncached_getpwnam_r;
        _uncached_getpwuid_r;
        __ungetwc_xpg5;
        _unordered;
        utssys;
        _verrfp;
        _verrxfp;
        _vwarnfp;
        _vwarnxfp;
        _warnfp;
        _warnxfp;
        __wcsftime_xpg5;
        __wcstok_xpg5;
        wdbindf;
        wdchkind;
        wddelim;
        _wrtchk;
        _xflsbuf;
        _xgetwidth;
        zone_add_datalink;
        zone_boot;
        zone_check_datalink;
        zone_create;
        zone_destroy;
        zone_enter;
        zone_getattr;
        zone_get_id;
        zone_list;
        zone_list_datalink;
        zonept;
        zone_remove_datalink;
        zone_setattr;
        zone_shutdown;
        zone_version;

$if _ELF32
        __divdi3;
        _file_set;
        _fprintf_c89;
        _fscanf_c89;
        _fwprintf_c89;
        _fwscanf_c89;
        _imaxabs_c89;
        _imaxdiv_c89;
        __moddi3;
        _printf_c89;
        _scanf_c89;
        _snprintf_c89;
        _sprintf_c89;
        _sscanf_c89;
        _strtoimax_c89;
        _strtoumax_c89;
        _swprintf_c89;
        _swscanf_c89;
        __udivdi3;
        __umoddi3;
        _vfprintf_c89;
        _vfscanf_c89;
        _vfwprintf_c89;
        _vfwscanf_c89;
        _vprintf_c89;
        _vscanf_c89;
        _vsnprintf_c89;
        _vsprintf_c89;
        _vsscanf_c89;
        _vswprintf_c89;
        _vswscanf_c89;
        _vwprintf_c89;
        _vwscanf_c89;
        _wcstoimax_c89;
        _wcstoumax_c89;
        _wprintf_c89;
        _wscanf_c89;
$endif

$if _sparc
        _cerror;
        install_utrap;
        _install_utrap;
        nop;
        _Q_cplx_div;
        _Q_cplx_div_ix;
        _Q_cplx_div_rx;
        _Q_cplx_lr_div;
        _Q_cplx_lr_div_ix;
        _Q_cplx_lr_div_rx;
        _Q_cplx_lr_mul;
        _Q_cplx_mul;
        _QgetRD;
        _xregs_clrptr;
$endif

$if sparc32
        __ashldi3;
        __ashrdi3;
        _cerror64;
        __cmpdi2;
        __floatdidf;
        __floatdisf;
        __floatundidf;
        __floatundisf;
        __lshrdi3;
        __muldi3;
        __ucmpdi2;
$endif

$if _x86
        _D_cplx_lr_div;
        _D_cplx_lr_div_ix;
        _D_cplx_lr_div_rx;
        _F_cplx_lr_div;
        _F_cplx_lr_div_ix;
        _F_cplx_lr_div_rx;
        __fltrounds;
        sysi86;
        _sysi86;
        _X_cplx_div;
        _X_cplx_div_ix;
        _X_cplx_div_rx;
        _X_cplx_lr_div;
        _X_cplx_lr_div_ix;
        _X_cplx_lr_div_rx;
        _X_cplx_mul;
        __xgetRD;
        __xtol;
        __xtoll;
        __xtoul;
```

```
        __xtoull;
$endif

$if i386
        __divrem64;
        ___tls_get_addr;
        __udivrem64;
$endif

# The following functions should not be exported from libc,
# but /lib/libm.so.2, some older versions of the Studio
# compiler/debugger components, and some ancient programs
# found in /usr/dist reference them.  When we no longer
# care about these old and broken binary objects, these
# symbols should be deleted.
        _brk                                    { FLAGS = NODYNSORT };
        _cond_broadcast                         { FLAGS = NODYNSORT };
        _cond_init                              { FLAGS = NODYNSORT };
        _cond_signal                            { FLAGS = NODYNSORT };
        _cond_wait                              { FLAGS = NODYNSORT };
        _ecvt                                   { FLAGS = NODYNSORT };
        _fcvt                                   { FLAGS = NODYNSORT };
        _getc_unlocked                          { FLAGS = NODYNSORT };
        _llseek                                 { FLAGS = NODYNSORT };
        _pthread_attr_getdetachstate            { FLAGS = NODYNSORT };
        _pthread_attr_getinheritsched           { FLAGS = NODYNSORT };
        _pthread_attr_getschedparam             { FLAGS = NODYNSORT };
        _pthread_attr_getschedpolicy            { FLAGS = NODYNSORT };
        _pthread_attr_getscope                  { FLAGS = NODYNSORT };
        _pthread_attr_getstackaddr              { FLAGS = NODYNSORT };
        _pthread_attr_getstacksize              { FLAGS = NODYNSORT };
        _pthread_attr_init                      { FLAGS = NODYNSORT };
        _pthread_condattr_getpshared            { FLAGS = NODYNSORT };
        _pthread_condattr_init                  { FLAGS = NODYNSORT };
        _pthread_cond_init                      { FLAGS = NODYNSORT };
        _pthread_create                         { FLAGS = NODYNSORT };
        _pthread_getschedparam                  { FLAGS = NODYNSORT };
        _pthread_join                           { FLAGS = NODYNSORT };
        _pthread_key_create                     { FLAGS = NODYNSORT };
        _pthread_mutexattr_getprioceiling       { FLAGS = NODYNSORT };
        _pthread_mutexattr_getprotocol          { FLAGS = NODYNSORT };
        _pthread_mutexattr_getpshared           { FLAGS = NODYNSORT };
        _pthread_mutexattr_init                 { FLAGS = NODYNSORT };
        _pthread_mutex_getprioceiling           { FLAGS = NODYNSORT };
        _pthread_mutex_init                     { FLAGS = NODYNSORT };
        _pthread_sigmask                        { FLAGS = NODYNSORT };
        _rwlock_init                            { FLAGS = NODYNSORT };
        _rw_rdlock                              { FLAGS = NODYNSORT };
        _rw_unlock                              { FLAGS = NODYNSORT };
        _rw_wrlock                              { FLAGS = NODYNSORT };
        _sbrk_unlocked                          { FLAGS = NODYNSORT };
        _select                                 { FLAGS = NODYNSORT };
        _sema_init                              { FLAGS = NODYNSORT };
        _sema_post                              { FLAGS = NODYNSORT };
        _sema_trywait                           { FLAGS = NODYNSORT };
        _sema_wait                              { FLAGS = NODYNSORT };
        _sysfs                                  { FLAGS = NODYNSORT };
        _thr_create                             { FLAGS = NODYNSORT };
        _thr_exit                               { FLAGS = NODYNSORT };
        _thr_getprio                            { FLAGS = NODYNSORT };
        _thr_getspecific                        { FLAGS = NODYNSORT };
        _thr_join                               { FLAGS = NODYNSORT };
        _thr_keycreate                          { FLAGS = NODYNSORT };
        _thr_kill                               { FLAGS = NODYNSORT };
        _thr_main                               { FLAGS = NODYNSORT };
        _thr_self                               { FLAGS = NODYNSORT };
        _thr_setspecific                        { FLAGS = NODYNSORT };
        _thr_sigsetmask                         { FLAGS = NODYNSORT };
        _thr_stksegment                         { FLAGS = NODYNSORT };
        _ungetc_unlocked                        { FLAGS = NODYNSORT };

    local:
        __imax_lldiv                            { FLAGS = NODYNSORT };
        _ti_thr_self                            { FLAGS = NODYNSORT };
        *;

$if lf64
        _seekdir64              { FLAGS = NODYNSORT };
        _telldir64              { FLAGS = NODYNSORT };
$endif

$if _sparc
        __cerror                { FLAGS = NODYNSORT };
$endif

$if sparc32
        __cerror64              { FLAGS = NODYNSORT };
$endif

$if sparcv9
        __cleanup               { FLAGS = NODYNSORT };
$endif

$if i386
        _syscall6               { FLAGS = NODYNSORT };
        __systemcall6           { FLAGS = NODYNSORT };
$endif

$if amd64
        ___tls_get_addr         { FLAGS = NODYNSORT };
$endif
};
```
______***unchanged_portion_omitted_***

```
**********************************************************
   25709 Mon Aug 12 18:24:55 2013
new/usr/src/man/man1/ps.1
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
’\" te
.\" Copyright 1989 AT&T
.\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved
.\" Copyright (c) 2012, Joyent, Inc. All Rights Reserved
.\" Copyright (c) 2013 Gary Mills
.\" Portions Copyright (c) 1992, X/Open Company Limited All Rights Reserved
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\" are reprinted and reproduced in electronic form in the Sun OS Reference Manu
.\" and Electronics Engineers, Inc and The Open Group. In the event of any discr
.\"  This notice shall appear on any product containing this material.
.\" The contents of this file are subject to the terms of the Common Development
.\"  See the License for the specific language governing permissions and limitat
.\" the fields enclosed by brackets "[]" replaced with your own identifying info
.TH PS 1 "Apr 16, 2013"
.TH PS 1 "Aug 16, 2009"
.SH NAME
ps \- report process status
.SH SYNOPSIS
.LP
.nf
\fBps\fR [\fB-aAcdefjHlLPWyZ\fR] [\fB-g\fR \fIgrplist\fR] [\fB-h\fR \fIlgrplist\
\fBps\fR [\fB-aAcdefjHlLPyZ\fR] [\fB-g\fR \fIgrplist\fR] [\fB-h\fR \fIlgrplist\f
     [\fB-n\fR \fInamelist\fR] [\fB-o\fR \fIformat\fR]... [\fB-p\fR \fIproclist\
     [\fB-s\fR \fIsidlist\fR] [\fB-t\fR \fIterm\fR] [\fB-u\fR \fIuidlist\fR] [\f
     [\fB-G\fR \fIgidlist\fR] [\fB-z\fR \fIzonelist\fR]
.fi

.SH DESCRIPTION
.sp
.LP
The \fBps\fR command prints information about active processes. Without
options, \fBps\fR prints information about processes that have the same
effective user \fBID\fR and the same controlling terminal as the invoker. The
output contains only the process \fBID\fR, terminal identifier, cumulative
execution time, and the command name. Otherwise, the information that is
displayed is controlled by the options.
.sp
.LP
Some options accept lists as arguments. Items in a list can be either separated
by commas or else enclosed in quotes and separated by commas or spaces. Values
for \fIproclist\fR and \fIgrplist\fR must be numeric.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.RS 15n
Lists information about \fBa\fRll processes most frequently requested: all
those except session leaders and processes not associated with a terminal.
.RE

.sp
.ne 2
.na
\fB\fB-A\fR\fR
```

```
.ad
.RS 15n
Lists information for all processes. Identical to \fB-e\fR, below.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 15n
Prints information in a format that reflects scheduler properties as described
in \fBpriocntl\fR(1). The \fB-c\fR option affects the output of the \fB-f\fR
and \fB-l\fR options, as described below.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR\fR
.ad
.RS 15n
Lists information about all processes except session leaders.
.RE

.sp
.ne 2
.na
\fB\fB-e\fR\fR
.ad
.RS 15n
Lists information about \fBe\fRvery process now running.
.sp
When the \fB-e\fRoption is specified, options \fB-z\fR, \fB-t\fR, \fB-u\fR,
\fB-U\fR, \fB-g\fR, \fB-G\fR, \fB-p\fR, \fB-g\fR, \fB-s\fR and \fB-a\fR options
have no effect.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR\fR
.ad
.RS 15n
Generates a \fBf\fRull listing. (See below for significance of columns in a
full listing.)
.RE

.sp
.ne 2
.na
\fB\fB-g\fR \fIgrplist\fR\fR
.ad
.RS 15n
Lists only process data whose group leader’s \fBID\fR number(s) appears in
\fIgrplist\fR. (A group leader is a process whose process \fBID\fR number is
identical to its process group \fBID\fR number.)
.RE

.sp
.ne 2
.na
\fB\fB-G\fR \fIgidlist\fR\fR
.ad
.RS 15n
Lists information for processes whose real group ID numbers are given in
```

```
\fIgidlist\fR. The \fIgidlist\fR must be a single argument in the form of a
blank- or comma-separated list.
.RE

.sp
.ne 2
.na
\fB\fB-h\fR \fIlgrplist\fR\fR
.ad
.RS 15n
Lists only the processes homed to the specified \fIlgrplist\fR. Nothing is
listed for any invalid group specified in \fIlgrplist\fR.
.RE

.sp
.ne 2
.na
\fB\fB-H\fR\fR
.ad
.RS 15n
Prints the home lgroup of the process under an additional column header, LGRP.
.RE

.sp
.ne 2
.na
\fB\fB-j\fR\fR
.ad
.RS 15n
Prints session \fBID\fR and process group \fBID\fR.
.RE

.sp
.ne 2
.na
\fB\fB-l\fR\fR
.ad
.RS 15n
Generates a \fBl\fRong listing. (See below.)
.RE

.sp
.ne 2
.na
\fB\fB-L\fR\fR
.ad
.RS 15n
Prints information about each light weight process (\fIlwp\fR) in each selected
process. (See below.)
.RE

.sp
.ne 2
.na
\fB\fB-n\fR \fInamelist\fR\fR
.ad
.RS 15n
Specifies the name of an alternative system \fInamelist\fR file in place of the
default. This option is accepted for compatibility, but is ignored.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIformat\fR\fR
.ad
```

```
.RS 15n
Prints information according to the format specification given in \fIformat\fR.
This is fully described in \fBDISPLAY FORMATS\fR. Multiple \fB-o\fR options can
be specified; the format specification is interpreted as the
space-character-separated concatenation of all the \fIformat\fR
option-arguments.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR \fIproclist\fR\fR
.ad
.RS 15n
Lists only process data whose process \fBID\fR numbers are given in
\fIproclist\fR.
.RE

.sp
.ne 2
.na
\fB\fB-P\fR\fR
.ad
.RS 15n
Prints the number of the processor to which the process or lwp is bound, if
any, under an additional column header, \fBPSR\fR.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIsidlist\fR\fR
.ad
.RS 15n
Lists information on all session leaders whose \fBID\fRs appear in
\fIsidlist\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR \fIterm\fR\fR
.ad
.RS 15n
Lists only process data associated with \fIterm\fR. Terminal identifiers are
specified as a device file name, and an identifier. For example, \fBterm/a\fR,
or \fBpts/0\fR.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIuidlist\fR\fR
.ad
.RS 15n
Lists only process data whose effective user \fBID\fR number or login name is
given in \fIuidlist\fR. In the listing, the numerical user \fBID\fR is printed
unless you give the \fB-f\fR option, which prints the login name.
.RE

.sp
.ne 2
.na
\fB\fB-U\fR \fIuidlist\fR\fR
.ad
.RS 15n
```

```
Lists information for processes whose real user \fBID\fR numbers or login names
are given in \fIuidlist\fR. The \fIuidlist\fR must be a single argument in the
form of a blank- or comma-separated list.
.RE

.sp
.ne 2
.na
\fB\fB-W\fR\fR
.ad
.RS 15n
Truncate long names even when \fBps\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-y\fR\fR
.ad
.RS 15n
Under a long listing (\fB-l\fR), omits the obsolete \fBF\fR and \fBADDR\fR
columns and includes an \fBRSS\fR column to report the resident set size of the
process. Under the \fB-y\fR option, both \fBRSS\fR and \fBSZ\fR (see below) is
reported in units of kilobytes instead of pages.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fIzonelist\fR\fR
.ad
.RS 15n
Lists only processes in the specified zones. Zones can be specified either by
name or ID. This option is only useful when executed in the global zone.
.RE

.sp
.ne 2
.na
\fB\fB-Z\fR\fR
.ad
.RS 15n
Prints the name of the zone with which the process is associated under an
additional column header, \fBZONE\fR. The \fBZONE\fR column width is limited to
8 characters. Use \fBps\fR \fB-eZ\fR for a quick way to see information about
every process now running along with the associated zone name. Use
.sp
.in +2
.nf
ps -eo zone,uid,pid,ppid,time,comm,...
.fi
.in -2
.sp

to see zone names wider than 8 characters.
.RE

.sp
.LP
Many of the options shown are used to select processes to list. If any are
specified, the default list is ignored and \fBps\fR selects the processes
represented by the inclusive OR of all the selection-criteria options.
.SH DISPLAY FORMATS
```

```
.sp
.LP
Under the \fB-f\fR option, \fBps\fR tries to determine the command name and
arguments given when the process was created by examining the user block.
Failing this, the command name is printed, as it would have appeared without
the \fB-f\fR option, in square brackets.
.sp
.LP
The column headings and the meaning of the columns in a \fBps\fR listing are
given below; the letters \fBf\fR and \fBl\fR indicate the option (f\fBull\fR or
\fBl\fRong, respectively) that causes the corresponding heading to appear;
\fBall\fR means that the heading always appears. \fBNote:\fR These two options
determine only what information is provided for a process; they do not
determine which processes are listed.
.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
lower priority. With the \fB-c\fR option, higher numbers mean higher priority.
.RE

.sp
.ne 2
.na
\fB\fBNI\fR(l)\fR
.ad
.RS 14n
Nice value, used in priority computation. Not printed when the \fB-c\fR option
is used. Only processes in the certain scheduling classes have a nice value.
.RE

.sp
.ne 2
.na
\fB\fBADDR\fR(l)\fR
.ad
.RS 14n
The memory address of the process.
.RE

.sp
.ne 2
.na
\fB\fBSZ\fR(l)\fR
.ad
.RS 14n
The total size of the process in virtual memory, including all mapped files and
devices, in pages. See \fBpagesize\fR(1).
.RE

.sp
.ne 2
.na
\fB\fBWCHAN\fR(l)\fR
.ad
.RS 14n
The address of an event for which the process is sleeping (if blank, the
process is running).
.RE

.sp
.ne 2
.na
\fB\fBSTIME\fR(f)\fR
.ad
.RS 14n
The starting time of the process, given in hours, minutes, and seconds. (A
process begun more than twenty-four hours before the \fBps\fR inquiry is
executed is given in months and days.)
```

```
.RE

.sp
.ne 2
.na
\fB\fBTTY\fR(all)\fR
.ad
.RS 14n
The controlling terminal for the process (the message, \fB?\fR, is printed when
there is no controlling terminal).
.RE

.sp
.ne 2
.na
\fB\fBTIME\fR(all)\fR
.ad
.RS 14n
The cumulative execution time for the process.
.RE

.sp
.ne 2
.na
\fB\fBLTIME\fR(all)\fR
.ad
.RS 14n
The execution time for the lwp being reported.
.RE

.sp
.ne 2
.na
\fB\fBCMD\fR(all)\fR
.ad
.RS 14n
The command name (the full command name and its arguments, up to a limit of 80
characters, are printed under the \fB-f\fR option).
.RE

.sp
.LP
The following two additional columns are printed when the \fB-j\fR option is
specified:
.sp
.ne 2
.na
\fB\fBPGID\fR\fR
.ad
.RS 8n
The process ID of the process group leader.
.RE

.sp
.ne 2
.na
\fB\fBSID\fR\fR
.ad
.RS 8n
The process ID of the session leader.
.RE

.sp
.LP
The following two additional columns are printed when the \fB-L\fR option is
specified:
```

```
.sp
.ne 2
.na
\fB\fBLWP\fR\fR
.ad
.RS 8n
The lwp ID of the lwp being reported.
.RE

.sp
.ne 2
.na
\fB\fBNLWP\fR\fR
.ad
.RS 8n
The number of lwps in the process (if \fB-f\fR is also specified).
.RE

.sp
.LP
Under the \fB-L\fR option, one line is printed for each lwp in the process and
the time-reporting fields \fBSTIME\fR and \fBLTIME\fR show the values for the
lwp, not the process. A traditional single-threaded process contains only one
lwp.
.sp
.LP
A process that has exited and has a parent, but has not yet been waited for by
the parent, is marked \fB<defunct>\fR\&.
.SS "\fB-o\fR format"
.sp
.LP
The \fB-o\fR option allows the output format to be specified under user
control.
.sp
.LP
The format specification must be a list of names presented as a single
argument, blank- or comma-separated. Each variable has a default header. The
default header can be overridden by appending an equals sign and the new text
of the header. The rest of the characters in the argument is used as the header
text. The fields specified are written in the order specified on the command
line, and should be arranged in columns in the output. The field widths are
selected by the system to be at least as wide as the header text (default or
overridden value). If the header text is null, such as \fB-o\fR \fIuser=,\fR
the field width is at least as wide as the default header text.
Long names are not truncated in this mode.
If all header text fields are null, no header line is written.
the field width is at least as wide as the default header text. If all header
text fields are null, no header line is written.
.sp
.LP
The following names are recognized in the POSIX locale:
.sp
.ne 2
.na
\fB\fBuser\fR\fR
.ad
.RS 10n
The effective user \fBID\fR of the process. This is the textual user \fBID\fR,
if it can be obtained and the field width permits, or a decimal representation
otherwise.
.RE

.sp
.ne 2
.na
\fB\fBruser\fR\fR
```

```
.ad
.RS 10n
The real user \fBID\fR of the process. This is the textual user \fBID\fR, if it
can be obtained and the field width permits, or a decimal representation
otherwise.
.RE

.sp
.ne 2
.na
\fB\fBgroup\fR\fR
.ad
.RS 10n
The effective group \fBID\fR of the process. This is the textual group
\fBID,\fR if it can be obtained and the field width permits, or a decimal
representation otherwise.
.RE

.sp
.ne 2
.na
\fB\fBrgroup\fR\fR
.ad
.RS 10n
The real group \fBID\fR of the process. This is the textual group \fBID,\fR if
it can be obtained and the field width permits, or a decimal representation
otherwise.
.RE

.sp
.ne 2
.na
\fB\fBpid\fR\fR
.ad
.RS 10n
The decimal value of the process \fBID\fR.
.RE

.sp
.ne 2
.na
\fB\fBppid\fR\fR
.ad
.RS 10n
The decimal value of the parent process \fBID\fR.
.RE

.sp
.ne 2
.na
\fB\fBpgid\fR\fR
.ad
.RS 10n
The decimal value of the process group \fBID.\fR
.RE

.sp
.ne 2
.na
\fB\fBpcpu\fR\fR
.ad
.RS 10n
The ratio of CPU time used recently to CPU time available in the same period,
expressed as a percentage. The meaning of ‘‘recently’’ in this context is
unspecified. The CPU time available is determined in an unspecified manner.
.RE
```

```
.sp
.ne 2
.na
\fB\fBvsz\fR\fR
.ad
.RS 10n
The total size of the process in virtual memory, in kilobytes.
.RE

.sp
.ne 2
.na
\fB\fBnice\fR\fR
.ad
.RS 10n
The decimal value of the system scheduling priority of the process. See
\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
Processor utilization for scheduling (obsolete).
.RE

.sp
.ne 2
.na
\fB\fBuid\fR\fR
.ad
.RS 11n
The effective user \fBID\fR number of the process as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBruid\fR\fR
.ad
.RS 11n
The real user \fBID\fR number of the process as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBgid\fR\fR
.ad
.RS 11n
The effective group \fBID\fR number of the process as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBrgid\fR\fR
.ad
.RS 11n
The real group \fBID\fR number of the process as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBprojid\fR\fR
.ad
.RS 11n
The project \fBID\fR number of the process as a decimal integer.
.RE
```

```
.sp
.ne 2
.na
\fB\fBproject\fR\fR
.ad
.RS 11n
The project \fBID\fR of the process as a textual value if that value can be
obtained; otherwise, as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBzoneid\fR\fR
.ad
.RS 11n
The zone \fBID\fR number of the process as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBzone\fR\fR
.ad
.RS 11n
The zone \fBID\fR of the process as a textual value if that value can be
obtained; otherwise, as a decimal integer.
.RE

.sp
.ne 2
.na
\fB\fBsid\fR\fR
.ad
.RS 11n
The process ID of the session leader.
.RE

.sp
.ne 2
.na
\fB\fBtaskid\fR\fR
.ad
.RS 11n
The task \fBID\fR of the process.
.RE

.sp
.ne 2
.na
\fB\fBclass\fR\fR
.ad
.RS 11n
The scheduling class of the process.
.RE

.sp
.ne 2
.na
\fB\fBpri\fR\fR
.ad
.RS 11n
The priority of the process. Higher numbers mean higher priority.
.RE

.sp
```

```
.ne 2
.na
\fB\fBopri\fR\fR
.ad
.RS 11n
The obsolete priority of the process. Lower numbers mean higher priority.
.RE

.sp
.ne 2
.na
\fB\fBlwp\fR\fR
.ad
.RS 11n
The decimal value of the lwp \fBID\fR. Requesting this formatting option causes
one line to be printed for each lwp in the process.
.RE

.sp
.ne 2
.na
\fB\fBnlwp\fR\fR
.ad
.RS 11n
The number of lwps in the process.
.RE

.sp
.ne 2
.na
\fB\fBpsr\fR\fR
.ad
.RS 11n
The number of the processor to which the process or lwp is bound.
.RE

.sp
.ne 2
.na
\fB\fBpset\fR\fR
.ad
.RS 11n
The \fBID\fR of the processor set to which the process or lwp is bound.
.RE

.sp
.ne 2
.na
\fB\fBaddr\fR\fR
.ad
.RS 11n
The memory address of the process.
.RE

.sp
.ne 2
.na
\fB\fBosz\fR\fR
.ad
.RS 11n
The total size of the process in virtual memory, in pages.
.RE

.sp
.ne 2
.na
```

```
\fB\fBwchan\fR\fR
.ad
.RS 11n
The address of an event for which the process is sleeping (if \(mi, the process
is running).
.RE

.sp
.ne 2
.na
\fB\fBstime\fR\fR
.ad
.RS 11n
The starting time or date of the process, printed with no blanks.
.RE

.sp
.ne 2
.na
\fB\fBrss\fR\fR
.ad
.RS 11n
The resident set size of the process, in kilobytes. The \fBrss\fR value
reported by \fBps\fR is an estimate provided by \fBproc\fR(4) that might
underestimate the actual resident set size. Users who wish to get more accurate
usage information for capacity planning should use \fBpmap\fR(1) \fB-x\fR
instead.
.RE

.sp
.ne 2
.na
\fB\fBpmem\fR\fR
.ad
.RS 11n
The ratio of the process’s resident set size to the physical memory on the
machine, expressed as a percentage.
.RE

.sp
.ne 2
.na
\fB\fBfname\fR\fR
.ad
.RS 11n
The first 8 bytes of the base name of the process’s executable file.
.RE

.sp
.ne 2
.na
\fB\fBctid\fR\fR
.ad
.RS 11n
The contract ID of the process contract the process is a member of as a decimal
integer.
.RE

.sp
.ne 2
.na
\fB\fBlgrp\fR\fR
.ad
.RS 11n
The home lgroup of the process.
.RE
```

```

.sp
.ne 2
.na
\fB\fBdmodel\fR\fR
.ad
.RS 11n
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 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
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
.sp
.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
.sp
.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
.sp
.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
.sp
.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
```

```
.sp
.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
.sp
.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).
```

```
**********************************************************
   16922 Mon Aug 12 18:24:55 2013
new/usr/src/man/man1m/prstat.1m
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
'\" te
.\" Copyright (c) 2013 Gary Mills
.\" Copyright (c) 2006, 2009 Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\"  See the License for the specific language governing permissions and limitat
.\" the fields enclosed by brackets "[]" replaced with your own identifying info
.TH PRSTAT 1M "Apr 15, 2013"
.TH PRSTAT 1M "Jun 25, 2009"
.SH NAME
prstat \- report active process statistics
.SH SYNOPSIS
.LP
.nf
\fBprstat\fR [\fB-acHJLmRrtTvWZ\fR] [\fB-d\fR u | d] [\fB-C\fR \fIpsrsetlist\fR]
\fBprstat\fR [\fB-acHJLmRrtTv\fR] [\fB-d\fR u | d] [\fB-C\fR \fIpsrsetlist\fR] [
     [\fB-j\fR \fIprojlist\fR] [\fB-k\fR \fItasklist\fR] [\fB-n\fR \fIntop\fR[,\
     [\fB-p\fR \fIpidlist\fR] [\fB-P\fR \fIcpulist\fR] [\fB-s\fR \fIkey\fR | \fB
     [\fB-u\fR \fIeuidlist\fR] [\fB-U\fR \fIuidlist\fR] [\fB-z\fR \fIzoneidlist\
     [\fB-u\fR \fIeuidlist\fR] [\fB-U\fR \fIuidlist\fR] [\fB-z\fR \fIzoneidlist\
     [\fIinterval\fR [\fIcount\fR]]
.fi

.SH DESCRIPTION
.sp
.LP
The \fBprstat\fR utility iteratively examines all active processes on the
system and reports statistics based on the selected output mode and sort order.
\fBprstat\fR provides options to examine only processes matching specified
\fBPID\fRs, \fBUID\fRs, zone \fBID\fRs, \fBCPU\fR \fBID\fRs, and processor set
\fBID\fRs.
.sp
.LP
The \fB-j\fR, \fB-k\fR, \fB-C\fR, \fB-p\fR, \fB-P\fR, \fB-u\fR, \fB-U\fR, and
\fB-z\fR options accept lists as arguments. Items in a list can be either
separated by commas or enclosed in quotes and separated by commas or spaces.
.sp
.LP
If you do not specify an option, \fBprstat\fR examines all processes and
reports statistics sorted by \fBCPU\fR usage.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Report information about processes and users. In this mode \fBprstat\fR
displays separate reports about processes and users at the same time.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.sp .6
```

```
.RS 4n
Print new reports below previous reports instead of overprinting them.
Long names are not truncated in this mode.
.RE

.sp
.ne 2
.na
\fB\fB-C\fR \fIpsrsetlist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or lwps that are bound to processor sets in the given
list. Each processor set is identified by an integer as reported by
\fBpsrset\fR(1M). The load averages displayed are the sum of the load averages
of the specified processor sets (see \fBpset_getloadavg\fR(3C)). Processes with
one or more LWPs bound to processor sets in the given list are reported even
when the \fB-L\fR option is not used.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fBu | d\fR\fR
.ad
.sp .6
.RS 4n
Specify \fBu\fR for a printed representation of the internal representation of
time. See \fBtime\fR(2). Specify \fBd\fR for standard date format. See
\fBdate\fR(1).
.RE

.sp
.ne 2
.na
\fB\fB-h\fR \fIlgrplist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or lwps whose home \fIlgroup\fR is in the given list of
\fIlgroups\fR. No processes or lwps will be listed for invalid \fIlgroups\fR.
.RE

.sp
.ne 2
.na
\fB\fB-H\fR\fR
.ad
.sp .6
.RS 4n
Report information about home \fIlgroup\fR. In this mode, \fBprstat\fR adds an
extra column showing process or lwps home \fIlgroup\fR with the header LGRP.
.RE

.sp
.ne 2
.na
\fB\fB-j\fR \fIprojlist\fR\fR
.ad
.sp .6
.RS 4n
Report only processes or lwps whose project \fBID\fR is in the given list. Each
project \fBID\fR can be specified as either a project name or a numerical
project \fBID\fR. See \fBproject\fR(4).
.RE
```

```
.sp
.ne 2
.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, or projects statistics are
reported if the \fB-a\fR, \fB-t\fR, \fB-T\fR, or \fB-J\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
.sp
.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
```

```
\fBLWPID\fR
.ad
.sp .6
.RS 4n
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
.sp
.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
.sp
.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
.sp
.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
.sp
.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.
```

```
**********************************************************
   12884 Mon Aug 12 18:24:55 2013
new/usr/src/man/man1m/useradd.1m
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
'\" te
.\" Copyright (c) 2013 Gary Mills
.\" Copyright (c) 2008 Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 1989 AT&T
.\" 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 USERADD 1M "Apr 16, 2013"
.TH USERADD 1M "Feb 19, 2008"
.SH NAME
useradd \- administer a new user login on the system
.SH SYNOPSIS
.LP
.nf
\fBuseradd\fR [\fB-A\fR \fIauthorization\fR [,\fIauthorization...\fR]]
     [\fB-b\fR \fIbase_dir\fR] [\fB-c\fR \fIcomment\fR] [\fB-d\fR \fIdir\fR] [\f
     [\fB-f\fR \fIinactive\fR] [\fB-g\fR \fIgroup\fR] [\fB-G\fR \fIgroup\fR [,\f
     [\fB-K\fR \fIkey=value\fR] [\fB-m\fR [\fB-k\fR \fIskel_dir\fR]] [\fB-p\fR \
     [\fB-P\fR \fIprofile\fR [,\fIprofile...\fR]] [\fB-R\fR \fIrole\fR [,\fIrole
     [\fB-s\fR \fIshell\fR] [\fB-u\fR \fIuid\fR [\fB-o\fR]] \fIlogin\fR
.fi

.LP
.nf
\fBuseradd\fR \fB-D\fR [\fB-A\fR \fIauthorization\fR [,\fIauthorization...\fR]]
     [\fB-b\fR \fIbase_dir\fR] [\fB-s\fR \fIshell\fR [\fB-k\fR \fIskel_dir\fR]]
     [\fB-f\fR \fIinactive\fR] [\fB-g\fR \fIgroup\fR] [\fB-K\fR \fIkey=value\fR]
     [\fB-P\fR \fIprofile\fR [,\fIprofile...\fR]] [\fB-R\fR \fIrole\fR [,\fIrole
.fi

.SH DESCRIPTION
.sp
.LP
\fBuseradd\fR adds a new user to the \fB/etc/passwd\fR and \fB/etc/shadow\fR
and \fB/etc/user_attr\fR files. The \fB-A\fR and \fB-P\fR options respectively
assign authorizations and profiles to the user. The \fB-R\fR option assigns
roles to a user. The \fB-p\fR option associates a project with a user. The
\fB-K\fR option adds a \fIkey=value\fR pair to \fB/etc/user_attr\fR for the
user. Multiple \fIkey=value\fR pairs may be added with multiple \fB-K\fR
options.
.sp
.LP
\fBuseradd\fR also creates supplementary group memberships for the user
(\fB-G\fR option) and creates the home directory (\fB-m\fR option) for the user
if requested. The new login remains locked until the \fBpasswd\fR(1) command is
executed.
.sp
.LP
Specifying \fBuseradd\fR \fB-D\fR with the \fB-s\fR, \fB-k\fR,\fB-g\fR,
\fB-b\fR, \fB-f\fR, \fB-e\fR, \fB-A\fR, \fB-P\fR, \fB-p\fR, \fB-R\fR, or
\fB-K\fR option (or any combination of these options) sets the default values
for the respective fields. See the \fB-D\fR option, below. Subsequent
\fBuseradd\fR commands without the \fB-D\fR option use these arguments.
.sp
.LP
The system file entries created with this command have a limit of 2048
characters per line. Specifying long arguments to several options can exceed
this limit.
.sp
.LP
```

```
\fBuseradd\fR requires that usernames be in the format described in
\fBpasswd\fR(4). A warning message is displayed if these restrictions are not
met. See \fBpasswd\fR(4) for the requirements for usernames.
.LP
To change the action of \fBuseradd\fR when the traditional login name
length limit of eight characters is exceeded, edit the file
\fB/etc/default/useradd\fR by removing the \fB#\fR (pound sign) before the
appropriate \fBEXCEED_TRAD=\fR entry, and adding it before the others.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-A\fR \fIauthorization\fR\fR
.ad
.sp .6
.RS 4n
One or more comma separated authorizations defined in \fBauth_attr\fR(4). Only
a user or role who has \fBgrant\fR rights to the authorization can assign it to
an account.
.RE

.sp
.ne 2
.na
\fB\fB-b\fR \fIbase_dir\fR\fR
.ad
.sp .6
.RS 4n
The base directory for new login home directories (see the \fB-d\fR option
below. When a new user account is being created, \fIbase_dir\fR must already
exist unless the \fB-m\fR option or the \fB-d\fR option is also specified.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR \fIcomment\fR\fR
.ad
.sp .6
.RS 4n
Any text string. It is generally a short description of the login, and is
currently used as the field for the user's full name. This information is
stored in the user's \fB/etc/passwd\fR entry.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fIdir\fR\fR
.ad
.sp .6
.RS 4n
The home directory of the new user. It defaults to
\fIbase_dir\fR/\fIaccount_name\fR, where \fIbase_dir\fR is the base directory
for new login home directories and \fIaccount_name\fR is the new login name.
.RE

.sp
.ne 2
.na
\fB\fB-D\fR\fR
.ad
.sp .6
```

```
.RS 4n
Display the default values for \fBgroup\fR, \fBbase_dir\fR, \fBskel_dir\fR,
\fBshell\fR, \fBinactive\fR, \fBexpire\fR, \fBproj\fR, \fBprojname\fR and
\fBkey=value\fR pairs. When used with the \fB-g\fR, \fB-b\fR, \fB-f\fR,
\fB-e\fR, \fB-A\fR, \fB-P\fR, \fB-p\fR, \fB-R\fR, or \fB-K\fR options, the
\fB-D\fR option sets the default values for the specified fields. The default
values are:
.sp
.ne 2
.na
\fBgroup\fR
.ad
.sp .6
.RS 4n
\fBother\fR (\fBGID\fR of 1)
.RE

.sp
.ne 2
.na
\fBbase_dir\fR
.ad
.sp .6
.RS 4n
\fB/home\fR
.RE

.sp
.ne 2
.na
\fBskel_dir\fR
.ad
.sp .6
.RS 4n
\fB/etc/skel\fR
.RE

.sp
.ne 2
.na
\fBshell\fR
.ad
.sp .6
.RS 4n
\fB/bin/sh\fR
.RE

.sp
.ne 2
.na
\fBinactive\fR
.ad
.sp .6
.RS 4n
\fB0\fR
.RE

.sp
.ne 2
.na
\fBexpire\fR
.ad
.sp .6
.RS 4n
null
.RE
```

```
.sp
.ne 2
.na
\fBauths\fR
.ad
.sp .6
.RS 4n
null
.RE

.sp
.ne 2
.na
\fBprofiles\fR
.ad
.sp .6
.RS 4n
null
.RE

.sp
.ne 2
.na
\fBproj\fR
.ad
.sp .6
.RS 4n
\fB3\fR
.RE

.sp
.ne 2
.na
\fBprojname\fR
.ad
.sp .6
.RS 4n
\fBdefault\fR
.RE

.sp
.ne 2
.na
\fBkey=value (pairs defined in \fBuser_attr\fR(4)\fR
.ad
.sp .6
.RS 4n
not present
.RE

.sp
.ne 2
.na
\fBroles\fR
.ad
.sp .6
.RS 4n
null
.RE

.RE

.sp
.ne 2
.na
```

```
\fB\fB-e\fR \fIexpire\fR\fR
.ad
.sp .6
.RS 4n
Specify the expiration date for a login. After this date, no user will be able
to access this login. The expire option argument is a date entered using one of
the date formats included in the template file \fB/etc/datemsk\fR. See
\fBgetdate\fR(3C).
.sp
If the date format that you choose includes spaces, it must be quoted. For
example, you can enter \fB10/6/90\fR or \fBOctober 6, 1990\fR. A null value
(\fB" "\fR) defeats the status of the expired date. This option is useful for
creating temporary logins.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIinactive\fR\fR
.ad
.sp .6
.RS 4n
The maximum number of days allowed between uses of a login ID before that
\fBID\fR is declared invalid. Normal values are positive integers. A value of
\fB0\fR defeats the status.
.RE

.sp
.ne 2
.na
\fB\fB-g\fR \fIgroup\fR\fR
.ad
.sp .6
.RS 4n
An existing group's integer \fBID\fR or character-string name. Without the
\fB-D\fR option, it defines the new user's primary group membership and
defaults to the default group. You can reset this default value by invoking
\fBuseradd\fR \fB-D\fR \fB-g\fR \fIgroup\fR. GIDs 0-99 are reserved for
allocation by the Solaris Operating System.
.RE

.sp
.ne 2
.na
\fB\fB-G\fR \fIgroup\fR\fR
.ad
.sp .6
.RS 4n
An existing group's integer \fBID\fR or character-string name. It defines the
new user's supplementary group membership. Duplicates between \fIgroup\fR with
the \fB-g\fR and \fB-G\fR options are ignored. No more than \fBNGROUPS_MAX\fR
groups can be specified. GIDs 0-99 are reserved for allocation by the Solaris
Operating System.
.RE

.sp
.ne 2
.na
\fB\fB-K\fR \fIkey=value\fR\fR
.ad
.sp .6
.RS 4n
A \fIkey=value\fR pair to add to the user's attributes. Multiple \fB-K\fR
options may be used to add multiple \fIkey=value\fR pairs. The generic \fB-K\fR
option with the appropriate key may be used instead of the specific implied key
options (\fB-A\fR, \fB-P\fR, \fB-R\fR, \fB-p\fR). See \fBuser_attr\fR(4) for a
```

```
list of valid \fIkey=value\fR pairs. The "type" key is not a valid key for this
option. Keys may not be repeated.
.RE

.sp
.ne 2
.na
\fB\fB-k\fR \fIskel_dir\fR\fR
.ad
.sp .6
.RS 4n
A directory that contains skeleton information (such as \fB\&.profile\fR) that
can be copied into a new user's home directory. This directory must already
exist. The system provides the \fB/etc/skel\fR directory that can be used for
this purpose.
.RE

.sp
.ne 2
.na
\fB\fB-m\fR\fR
.ad
.sp .6
.RS 4n
Create the new user's home directory if it does not already exist. If the
directory already exists, it must have read, write, and execute permissions by
\fIgroup\fR, where \fIgroup\fR is the user's primary group.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR\fR
.ad
.sp .6
.RS 4n
This option allows a \fBUID\fR to be duplicated (non-unique).
.RE

.sp
.ne 2
.na
\fB\fB-P\fR \fIprofile\fR\fR
.ad
.sp .6
.RS 4n
One or more comma-separated execution profiles defined in \fBprof_attr\fR(4).
.RE

.sp
.ne 2
.na
\fB\fB-p\fR \fIprojname\fR\fR
.ad
.sp .6
.RS 4n
Name of the project with which the added user is associated. See the
\fIprojname\fR field as defined in \fBproject\fR(4).
.RE

.sp
.ne 2
.na
\fB\fB-R\fR \fIrole\fR\fR
.ad
.sp .6
```

```
.RS 4n
One or more comma-separated execution profiles defined in \fBuser_attr\fR(4).
Roles cannot be assigned to other roles.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIshell\fR\fR
.ad
.sp .6
.RS 4n
Full pathname of the program used as the user’s shell on login. It defaults to
an empty field causing the system to use \fB/bin/sh\fR as the default. The
value of \fIshell\fR must be a valid executable file.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIuid\fR\fR
.ad
.sp .6
.RS 4n
The \fBUID\fR of the new user. This \fBUID\fR must be a non-negative decimal
integer below \fBMAXUID\fR as defined in \fB<sys/param.h>\fR\&. The \fBUID\fR
defaults to the next available (unique) number above the highest number
currently assigned. For example, if \fBUID\fRs 100, 105, and 200 are assigned,
the next default \fBUID\fR number will be 201. \fBUID\fRs \fB0\fR-\fB99\fR are
reserved for allocation by the Solaris Operating System.
.RE

.SH FILES
.sp
.LP
\fB/etc/default/useradd\fR
.sp
.LP
\fB/etc/datemsk\fR
.sp
.LP
\fB/etc/passwd\fR
.sp
.LP
\fB/etc/shadow\fR
.sp
.LP
\fB/etc/group\fR
.sp
.LP
\fB/etc/skel\fR
.sp
.LP
\fB/usr/include/limits.h\fR
.sp
.LP
\fB/etc/user_attr\fR
.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
```

```
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Committed
.TE

.SH SEE ALSO
.sp
.LP
\fBpasswd\fR(1), \fBprofiles\fR(1), \fBroles\fR(1), \fBusers\fR(1B),
\fBgroupadd\fR(1M), \fBgroupdel\fR(1M), \fBgroupmod\fR(1M), \fBgrpck\fR(1M),
\fBlogins\fR(1M), \fBpwck\fR(1M), \fBuserdel\fR(1M), \fBusermod\fR(1M),
\fBgetdate\fR(3C), \fBauth_attr\fR(4), \fBpasswd\fR(4), \fBprof_attr\fR(4),
\fBproject\fR(4), \fBuser_attr\fR(4), \fBattributes\fR(5)
.SH DIAGNOSTICS
.sp
.LP
In case of an error, \fBuseradd\fR prints an error message and exits with a
non-zero status.
.sp
.LP
The following indicates that \fBlogin\fR specified is already in use:
.sp
.in +2
.nf
UX: useradd: ERROR: login is already in use. Choose another.
.fi
.in -2
.sp

.sp
.LP
The following indicates that the \fIuid\fR specified with the \fB-u\fR option
is not unique:
.sp
.in +2
.nf
UX: useradd: ERROR: uid \fIuid\fR is already in use. Choose another.
.fi
.in -2
.sp

.sp
.LP
The following indicates that the \fIgroup\fR specified with the \fB-g\fR option
is already in use:
.sp
.in +2
.nf
UX: useradd: ERROR: group \fIgroup\fR does not exist. Choose another.
.fi
.in -2
.sp

.sp
.LP
The following indicates that the \fIuid\fR specified with the \fB-u\fR option
is in the range of reserved \fBUID\fRs (from \fB0\fR-\fB99\fR):
.sp
.in +2
.nf
UX: useradd: WARNING: uid \fIuid\fR is reserved.
.fi
.in -2
.sp
```

```
.sp
.LP
The following indicates that the \fIuid\fR specified with the \fB-u\fR option
exceeds \fBMAXUID\fR as defined in \fB<sys/param.h>\fR:
.sp
.in +2
.nf
UX: useradd: ERROR: uid \fIuid\fR is too big. Choose another.
.fi
.in -2
.sp

.sp
.LP
The following indicates that the \fB/etc/passwd\fR or \fB/etc/shadow\fR files
do not exist:
.sp
.in +2
.nf
UX: useradd: ERROR: Cannot update system files - login cannot be created.
.fi
.in -2
.sp

.SH NOTES
.sp
.LP
The \fBuseradd\fR utility adds definitions to only the local \fB/etc/group\fR,
\fBetc/passwd\fR, \fB/etc/passwd\fR, \fB/etc/shadow\fR, \fB/etc/project\fR, and
\fB/etc/user_attr\fR files. If a network name service such as \fBNIS\fR or
\fBNIS+\fR is being used to supplement the local \fB/etc/passwd\fR file with
additional entries, \fBuseradd\fR cannot change information supplied by the
network name service. However \fBuseradd\fR will verify the uniqueness of the
user name (or role) and user id and the existence of any group names specified
against the external name service.
```

```
**********************************************************
    7698 Mon Aug 12 18:24:55 2013
new/usr/src/man/man3c/getlogin.3c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
'\" te
.\" Copyright (c) 2013 Gary Mills
.\" Copyright (c) 2004 Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" http://www.opengroup.org/bookstore/.
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\"  This notice shall appear on any product containing this material.
.\" 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 GETLOGIN 3C "May 25, 2013"
.TH GETLOGIN 3C "May 18, 2004"
.SH NAME
getlogin, getlogin_r \- get login name
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBchar *\fR\fBgetlogin\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBchar *\fR\fBgetlogin_r\fR(\fBchar *\fR\fIname\fR, \fBint\fR \fInamelen\fR);
.fi

.SS "Standard conforming"
.LP
.nf
cc [ \fIflag \fR... ] \fIfile\fR... \fB-D_POSIX_PTHREAD_SEMANTICS\fR [ \fIlibrar

\fBint\fR \fBgetlogin_r\fR(\fBchar *\fR\fIname\fR, \fBsize_t\fR \fInamesize\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetlogin()\fR function returns a pointer to the login name as found in
\fB/var/adm/utmpx\fR. It can be used in conjunction with \fBgetpwnam\fR(3C) to
locate the correct password file entry when the same user \fBID\fR is shared by
several login names.
.sp
.LP
The login name plus the terminating null byte can be up to 33 characters
in length.
Newly-compiled programs should use the \fBLOGIN_NAME_MAX\fR symbol,
defined in <\fBlimits.h\fR>, to size the buffer.
Older programs that call \fBgetlogin()\fR  expect only the legacy
9-character length.
These automatically link to a version of the \fBgetlogin()\fR functions that
truncates longer login names.
It's also possible to compile new programs that link to truncating versions
of these functions by defining \fB__USE_LEGACY_LOGNAME__\fR in the
compile environment.
.sp
.LP
Some older programs will correctly handle long login names returned
```

```
by the \fBgetlogin()\fR function.
For this case, the user compatibility library
\fB/usr/lib/getloginx.so.1\fR redirects to a version of the \fBgetlogin()\fR
function that returns the long name.
This library should be added to such an application
at runtime using \fBLD_PRELOAD\fR.
.sp
.LP
If \fBgetlogin()\fR is called within a process that is not attached to a
terminal, it returns a null pointer. The correct procedure for determining the
login name is to call \fBcuserid\fR(3C), or to call \fBgetlogin()\fR and if it
fails to call \fBgetpwuid\fR(3C).
.sp
.LP
The \fBgetlogin_r()\fR function has the same functionality as \fBgetlogin()\fR
except that the caller must supply a buffer \fIname\fR with length
\fInamelen\fR to store the result.  The \fIname\fR buffer must be at least
\fBLOGIN_NAME_MAX\fR bytes in size (defined in <\fBlimits.h\fR>). The
\fB_POSIX_LOGIN_NAME_MAX\fR bytes in size (defined in <\fBlimits.h\fR>). The
POSIX version (see \fBstandards\fR(5)) of \fBgetlogin_r()\fR takes a
\fInamesize\fR parameter of type \fBsize_t\fR.
.SH RETURN VALUES
.sp
.LP
Upon successful completion, \fBgetlogin()\fR returns a pointer to the login
name or a null pointer if the user's login name cannot be found.  Otherwise it
returns a null pointer and sets \fBerrno\fR to indicate the error.
.sp
.LP
The standard-conforming \fBgetlogin_r()\fR returns \fB0\fR if successful, or
the error number upon failure.
.SH ERRORS
.sp
.LP
The \fBgetlogin_r()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBERANGE\fR\fR
.ad
.RS 10n
The size of the buffer is smaller than the result to be returned.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
And entry for the current user was not found in the \fB/var/adm/utmpx\fR file.
.RE

.sp
.LP
The \fBgetlogin()\fR and \fBgetlogin_r()\fR functions may fail if:
.sp
.ne 2
.na
\fB\fBEMFILE\fR\fR
.ad
.RS 10n
There are {\fBOPEN_MAX\fR} file descriptors currently open in the calling
process.
.RE
```

```
.sp
.ne 2
.na
\fB\fBENFILE\fR\fR
.ad
.RS 10n
The maximum allowable number of files is currently open in the system.
.RE

.sp
.ne 2
.na
\fB\fBENXIO\fR\fR
.ad
.RS 10n
The calling process has no controlling terminal.
.RE

.sp
.LP
The \fBgetlogin_r()\fR function may fail if:
.sp
.ne 2
.na
\fB\fBERANGE\fR\fR
.ad
.RS 10n
The size of the buffer is smaller than the result to be returned.
.RE

.SH USAGE
.sp
.LP
The return value of \fBgetlogin()\fR points to thread-specific data whose
content is overwritten on each call by the same thread.
.sp
.LP
Three names associated with the current process can be determined:
\fBgetpwuid(\fR\fBgeteuid()\fR\fB)\fR returns the name associated with the
effective user ID of the process; \fBgetlogin()\fR returns the name associated
with the current login activity; and \fBgetpwuid(\fR\fBgetuid()\fR\fB)\fR
returns the name associated with the real user ID of the process.
.SH FILES
.sp
.ne 2
.na
\fB\fB/var/adm/utmpx\fR\fR
.ad
.RS 18n
user access and administration information
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/getloginx.so.1\fR\fR
.ad
.RS 18n
A compatibility library that returns long login names to older applications.
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/64/getloginx.so.1\fR\fR
.ad
```

```
.RS 18n
A 64-bit compatibility library to return long login names.
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Standard
_
MT-Level        See below.
.TE

.SH SEE ALSO
.sp
.LP
\fBgeteuid\fR(2), \fBgetuid\fR(2), \fBcuserid\fR(3C), \fBgetgrnam\fR(3C),
\fBgetpwnam\fR(3C), \fBgetpwuid\fR(3C), \fButmpx\fR(4), \fBattributes\fR(5),
\fBstandards\fR(5)
.SH NOTES
.sp
.LP
When compiling multithreaded programs, see \fBIntro\fR(3).
.sp
.LP
The \fBgetlogin()\fR function is safe to use in multithreaded applications, but
is discouraged. The \fBgetlogin_r()\fR function should be used instead.
.sp
.LP
Solaris 2.4 and earlier releases provided a \fBgetlogin_r()\fR as specified in
POSIX.1c Draft 6. The final POSIX.1c standard changed the interface as
described above. Support for the Draft 6 interface is provided for
compatibility only and may not be supported in future releases. New
applications and libraries should use the standard-conforming interface.
```

```
**********************************************************
   28316 Mon Aug 12 18:24:55 2013
new/usr/src/man/man3c/sysconf.3c
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
'\" te
.\" Copyright (c) 2013 Gary Mills
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" http://www.opengroup.org/bookstore/.
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\"  This notice shall appear on any product containing this material.
.\" 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 SYSCONF 3C "Apr 16, 2013"
.TH SYSCONF 3C "Mar 26, 2008"
.SH NAME
sysconf \- get configurable system variables
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBlong\fR \fBsysconf\fR(\fBint\fR \fIname\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBsysconf()\fR function provides a method for an application to determine
the current value of a configurable system limit or option (variable).
.sp
.LP
The \fIname\fR argument represents the system variable to be queried. The
following table lists the minimal set of system variables from \fB<limits.h>\fR
and \fB<unistd.h>\fR that can be returned by \fBsysconf()\fR and the symbolic
constants defined in \fB<unistd.h>\fR that are the corresponding values used
for \fIname\fR on the  SPARC and x86 platforms.
.sp
.in +2
.nf
    Name                   Return Value              Meaning
_______________________________________________________________________
_SC_2_C_BIND            _POSIX2_C_BIND             Supports the C lang-
                                                   uage binding option
_SC_2_C_DEV             _POSIX2_C_DEV              Supports the C lang-
                                                   uage development
                                                   utilities option
_SC_2_C_VERSION         _POSIX2_C_VERSION          Integer value
                                                   indicates version
                                                   of ISO POSIX-2
                                                   standard (Commands)
_SC_2_CHAR_TERM         _POSIX2_CHAR_TERM          Supports at least
                                                   one terminal
_SC_2_FORT_DEV          _POSIX2_FORT_DEV           Supports FORTRAN
                                                   Development
                                                   Utilities Option
_SC_2_FORT_RUN          _POSIX2_FORT_RUN           Supports FORTRAN
                                                   Run-time Utilities
                                                   Option
_SC_2_LOCALEDEF         _POSIX2_LOCALEDEF          Supports creation
                                                   of locales by the
```

```
                                                   localedef utility
_SC_2_SW_DEV            _POSIX2_SW_DEV             Supports Software
                                                   Development Utility
                                                   Option
_SC_2_UPE               _POSIX2_UPE                Supports User
                                                   Portability
                                                   Utilities Option
_SC_2_VERSION           _POSIX2_VERSION            Integer value
                                                   indicates version
                                                   of ISO POSIX-2
                                                   standard (C language
                                                   binding)
_SC_AIO_LISTIO_MAX      AIO_LISTIO_MAX             Max number of I/O
                                                   operations in a
                                                   single list I/O call
                                                   supported
_SC_AIO_MAX             AIO_MAX                    Max number of
                                                   outstanding
                                                   asynchronous I/O
                                                   operations supported
_SC_AIO_PRIO_DELTA_MAX  AIO_PRIO_DELTA_MAX         Max amount by which
                                                   process can decrease
                                                   its asynchronous
                                                   I/O priority level
                                                   from its own
                                                   scheduling priority
_SC_ARG_MAX             ARG_MAX                    Max size of argv[]
                                                   plus envp[]
_SC_ASYNCHRONOUS_IO     _POSIX_ASYNCHRONOUS_IO     Supports
                                                   Asynchronous I/O
_SC_ATEXIT_MAX          ATEXIT_MAX                 Max number of
                                                   functions that can
                                                   be registered with
                                                   atexit()
_SC_AVPHYS_PAGES                                   Number of physical
                                                   memory pages not
                                                   currently in use by
                                                   system
_SC_BARRIERS            _POSIX_BARRIERS            Supports Barriers
                                                   option
_SC_BC_BASE_MAX         BC_BASE_MAX                Maximum obase values
                                                   allowed by bc
_SC_BC_DIM_MAX          BC_DIM_MAX                 Max number of
                                                   elements permitted
                                                   in array by bc
_SC_BC_SCALE_MAX        BC_SCALE_MAX               Max scale value
                                                   allowed by bc
_SC_BC_STRING_MAX       BC_STRING_MAX              Max length of string
                                                   constant allowed by
                                                   bc
_SC_CHILD_MAX           CHILD_MAX                  Max processes
                                                   allowed to a UID
_SC_CLK_TCK             CLK_TCK                    Ticks per second
                                                   (clock_t)
_SC_CLOCK_SELECTION     _POSIX_CLOCK_SELECTION     Supports Clock
                                                   Selection option
_SC_COLL_WEIGHTS_MAX    COLL_WEIGHTS_MAX           Max number of
                                                   weights that can be
                                                   assigned to entry of
                                                   the LC_COLLATE order
                                                   keyword in locale
                                                   definition file
_SC_CPUID_MAX                                      Max possible
                                                   processor ID
_SC_DELAYTIMER_MAX      DELAYTIMER_MAX             Max number of timer
                                                   expiration overruns
```

```
_SC_EXPR_NEST_MAX       EXPR_NEST_MAX              Max number of
                                                   parentheses by expr
_SC_FSYNC               _POSIX_FSYNC               Supports File
                                                   Synchronization
_SC_GETGR_R_SIZE_MAX                               Max size of group
                                                   entry buffer
_SC_GETPW_R_SIZE_MAX                               Max size of password
                                                   entry buffer
_SC_HOST_NAME_MAX       _POSIX_HOST_NAME_MAX       Maximum length of a
                                                   host name (excluding
                                                   terminating null)
_SC_IOV_MAX             IOV_MAX                    Max number of iovec
                                                   structures available
                                                   to one process for
                                                   use with readv()
                                                   and writev()
_SC_JOB_CONTROL         _POSIX_JOB_CONTROL         Job control
                                                   supported?
_SC_LINE_MAX            LINE_MAX                   Max length of input
                                                   line
_SC_LOGIN_NAME_MAX      LOGIN_NAME_MAX             Max length of login
_SC_LOGIN_NAME_MAX      LOGNAME_MAX + 1            Max length of login
                                                   name
_SC_LOGNAME_MAX         LOGNAME_MAX
_SC_MAPPED_FILES        _POSIX_MAPPED_FILES        Supports Memory
                                                   Mapped Files
_SC_MAXPID                                         Max pid value
_SC_MEMLOCK             _POSIX_MEMLOCK             Supports Process
                                                   Memory Locking
_SC_MEMLOCK_RANGE       _POSIX_MEMLOCK_RANGE       Supports Range
                                                   Memory Locking
_SC_MEMORY_PROTECTION   _POSIX_MEMORY_PROTECTION   Supports Memory
                                                   Protection
_SC_MESSAGE_PASSING     _POSIX_MESSAGE_PASSING     Supports Message
                                                   Passing
_SC_MONOTONIC_CLOCK     _POSIX_MONOTONIC_CLOCK     Supports Monotonic
                                                   Clock option
_SC_MQ_OPEN_MAX         MQ_OPEN_MAX                Max number of open
                                                   message queues a
                                                   process can hold
_SC_MQ_PRIO_MAX         MQ_PRIO_MAX                Max number of
                                                   message priorities
                                                   supported
_SC_NGROUPS_MAX         NGROUPS_MAX                Max simultaneous
                                                   groups to which
                                                   one can belong
_SC_NPROCESSORS_CONF                               Number of processors
                                                   configured
_SC_NPROCESSORS_MAX                                Max number of
                                                   processors supported
                                                   by platform
_SC_NPROCESSORS_ONLN                               Number of processors
                                                   online
_SC_OPEN_MAX            OPEN_MAX                   Max open files per
                                                   process
_SC_PAGESIZE            PAGESIZE                   System memory page
                                                   size
_SC_PAGE_SIZE           PAGESIZE                   Same as _SC_PAGESIZE
_SC_PASS_MAX            PASS_MAX                   Max number of
                                                   significant bytes
                                                   in a password
_SC_PHYS_PAGES                                     Total number of
                                                   pages of physical
                                                   memory in system
_SC_PRIORITIZED_IO      _POSIX_PRIORITIZED_IO      Supports Prioritized
                                                   I/O
```

```
_SC_PRIORITY_SCHEDULING _POSIX_PRIORITY_SCHEDULING Supports Process
                                                   Scheduling
_SC_RAW_SOCKETS         _POSIX_RAW_SOCKETS         Supports Raw Sockets
                                                   option
_SC_RE_DUP_MAX          RE_DUP_MAX                 Max number of
                                                   repeated occurrences
                                                   of a regular
                                                   expression permitted
                                                   when using interval
                                                   notation \e{m,n\e}
_SC_READER_WRITER_LOCKS _POSIX_READER_WRITER_LOCKS Supports IPV6 option
_SC_REALTIME_SIGNALS    _POSIX_REALTIME_SIGNALS    Supports Realtime
                                                   Signals
_SC_REGEXP              _POSIX_REGEXP              Supports Regular
                                                   Expression Handling
                                                   option
_SC_RTSIG_MAX           RTSIG_MAX                  Max number of
                                                   realtime signals
                                                   reserved for
                                                   application use
_SC_SAVED_IDS           _POSIX_SAVED_IDS           Saved IDs
                                                   (seteuid())
                                                   supported?
_SC_SEM_NSEMS_MAX       SEM_NSEMS_MAX              Max number of POSIX
                                                   semaphores a process
                                                   can have
_SC_SEM_VALUE_MAX       SEM_VALUE_MAX              Max value a POSIX
                                                   semaphore can have
_SC_SEMAPHORES          _POSIX_SEMAPHORES          Supports Semaphores
_SC_SHARED_MEMORY_      _POSIX_SHARED_MEMORY_      Supports Shared
   OBJECTS                 OBJECTS                 Memory Objects
_SC_SHELL               _POSIX_SHELL               Supports POSIX shell
_SC_SIGQUEUE_MAX        SIGQUEUE_MAX               Max number of queued
                                                   signals that a
                                                   process can send and
                                                   have pending at
                                                   receiver(s) at a
                                                   time
_SC_SPAWN               _POSIX_SPAWN               Supports Spawn option
_SC_SPIN_LOCKS          _POSIX_SPIN_LOCKS          Supports Spin Locks
                                                   option
_SC_STACK_PROT                                     Default stack
                                                   protection
_SC_STREAM_MAX          STREAM_MAX                 Number of streams
                                                   one process can
                                                   have open at a time
_SC_SYMLOOP_MAX         _POSIX_SYMLOOP_MAX         Max number of symbolic
                                                   links that can be
                                                   reliably traversed in
                                                   the resolution of a
                                                   pathname in the absence
                                                   of a loop
_SC_SYNCHRONIZED_IO     _POSIX_SYNCHRONIZED_IO     Supports
                                                   Synchronized I/O
_SC_THREAD_ATTR_        _POSIX_THREAD_ATTR_        Supports Thread
   STACKADDR               STACKADDR               Stack Address
                                                   Attribute option
_SC_THREAD_ATTR_        _POSIX_THREAD_ATTR_        Supports Thread
   STACKSIZE               STACKSIZE               Stack Size
                                                   Attribute option
_SC_THREAD_DESTRUCTOR_  PTHREAD_DESTRUCTOR_        Number attempts made
   ITERATIONS              ITERATIONS              to destroy thread-
                                                   specific data on
                                                   thread exit
_SC_THREAD_KEYS_MAX     PTHREAD_KEYS_MAX           Max number of data
                                                   keys per process
```

```
_SC_THREAD_PRIO_        _POSIX_THREAD_PRIO_        Supports Priority
   INHERIT                 INHERIT                 Inheritance option
_SC_THREAD_PRIO_        _POSIX_THREAD_PRIO_        Supports Priority
   PROTECT                 PROTECT                 Protection option
_SC_THREAD_PRIORITY_    _POSIX_THREAD_PRIORITY_    Supports Thread
   SCHEDULING              SCHEDULING              Execution
                                                   Scheduling option
_SC_THREAD_PROCESS_     _POSIX_THREAD_PROCESS_     Supports
   SHARED                  SHARED                  Process-Shared
                                                   Synchronization
                                                   option
_SC_THREAD_SAFE_        _POSIX_THREAD_SAFE_        Supports Thread-Safe
   FUNCTIONS               FUNCTIONS               Functions option
_SC_THREAD_STACK_MIN    PTHREAD_STACK_MIN          Min byte size of
                                                   thread stack storage
_SC_THREAD_THREADS_MAX  PTHREAD_THREADS_MAX        Max number of
                                                   threads per process
_SC_THREADS             _POSIX_THREADS             Supports Threads
                                                   option
_SC_TIMEOUTS            _POSIX_TIMEOUTS            Supports Timeouts
                                                   option
_SC_TIMER_MAX           TIMER_MAX                  Max number of timer
                                                   per process
                                                   supported
_SC_TIMERS              _POSIX_TIMERS              Supports Timers
_SC_TTY_NAME_MAX        TTYNAME_MAX                Max length of tty
                                                   device name
_SC_TZNAME_MAX          TZNAME_MAX                 Max number of bytes
                                                   supported for name
                                                   of a time zone
_SC_V6_ILP32_OFF32      _POSIX_V6_ILP32_OFF32      Supports  X/Open
                                                   ILP32 w/32-bit
                                                   offset build
                                                   environment
_SC_V6_ILP32_OFFBIG     _POSIX_V6_ILP32_OFFBIG     Supports  X/Open
                                                   ILP32 w/64-bit
                                                   offset build
                                                   environment
_SC_V6_LP64_OFF64       _POSIX_V6_LP64_OFF64       Supports  X/Open
                                                   LP64 w/64-bit
                                                   offset build
                                                   environment
_SC_V6_LPBIG_OFFBIG     _POSIX_V6_LPBIG_OFFBIG     Same as
                                                   _SC_V6_LP64_OFF64
_SC_VERSION             _POSIX_VERSION             POSIX.1 version
                                                   supported
_SC_XBS5_ILP32_OFF32    _XBS_ILP32_OFF32           Indicates support
                                                   for X/Open ILP32
                                                   w/32-bit offset
                                                   build environment
_SC_XBS5_ILP32_OFFBIG   _XBS5_ILP32_OFFBIG         Indicates support
                                                   for X/Open ILP32
                                                   w/64-bit offset
                                                   build environment
_SC_XBS5_LP64_OFF64     _XBS5_LP64_OFF64           Indicates support of
                                                   X/Open LP64,
                                                   64-bit offset
                                                   build environment
_SC_XBS5_LPBIG_OFFBIG   _XBS5_LP64_OFF64           Same as
                                                   _SC_XBS5_LP64_OFF64
_SC_XOPEN_CRYPT         _XOPEN_CRYPT               Supports X/Open
                                                   Encryption Feature
                                                   Group
_SC_XOPEN_ENH_I18N      _XOPEN_ENH_I18N            Supports X/Open
                                                   Enhanced
                                                   Internationalization
```

```
                                                   Feature Group
_SC_XOPEN_LEGACY        _XOPEN_LEGACY              Supports X/Open
                                                   Legacy Feature Group
_SC_XOPEN_REALTIME      _XOPEN_REALTIME            Supports X/Open
                                                   POSIX Realtime
                                                   Feature Group
_SC_XOPEN_REALTIME_     _XOPEN_REALTIME_THREADS    Supports X/Open
     THREADS                                       POSIX Reatime
                                                   Threads Feature
                                                   Group
_SC_XOPEN_SHM           _XOPEN_SHM                 Supports X/Open
                                                   Shared Memory
                                                   Feature Group
_SC_XOPEN_STREAMS       _POSIX_XOPEN_STREAMS       Supports XSI Streams
                                                   option group
_SC_XOPEN_UNIX          _XOPEN_UNIX                Supports X/Open CAE
                                                   Specification,
                                                   August 1994, System
                                                   Interfaces and
                                                   Headers, Issue 4,
                                                   Version 2
_SC_XOPEN_VERSION       _XOPEN_VERSION             Integer value
                                                   indicates version of
                                                   X/Open Portability
                                                   Guide to which
                                                   implementation
                                                   conforms
_SC_XOPEN_XCU_VERSION   _XOPEN_XCU_VERSION         Integer value
                                                   indicates version of
                                                   XCU specification to
                                                   which implementation
                                                   conforms
.fi
.in -2
.sp

.sp
.LP
The following options are not supported and return \(mi1:
.sp

.sp
.TS
l l
l l .
\fB_SC_2_PBS\fR \fB_POSIX2_PBS\fR
\fB_SC_2_PBS_ACCOUNTING\fR      \fB_POSIX2_PBS_ACCOUNTING\fR
\fB_SC_2_PBS_CHECKPOINT\fR      \fB_POSIX2_PBS_CHECKPOINT\fR
\fB_SC_2_PBS_LOCATE\fR  \fB_POSIX2_PBS_LOCATE\fR
\fB_SC_2_PBS_MESSAGE\fR \fB_POSIX2_PBS_MESSAGE\fR
\fB_SC_2_PBS_TRACK\fR   \fB_POSIX2_PBS_TRACK\fR
\fB_SC_ADVISORY_INFO\fR \fB_POSIX_ADVISORY_INFO\fR
\fB_SC_CPUTIME\fR       \fB_POSIX_CPUTIME\fR
\fB_SC_SPORADIC_SERVER\fR       \fB_POSIX_SPORADIC_SERVER\fR
\fB_SC_SS_REPL_MAX\fR   \fB_POSIX_SS_REPL_MAX\fR
\fB_SC_THREAD_CPUTIME\fR        \fB_POSIX_THREAD_CPUTIME\fR
\fB_SC_THREAD_SPORADIC_SERVER\fR        \fB_POSIX_THREAD_SPORADIC_SERVER\fR
\fB_SC_TRACE\fR \fB_POSIX_TRACE\fR
\fB_SC_TRACE_EVENT_FILTER\fR    \fB_POSIX_TRACE_EVENT_FILTER\fR
\fB_SC_TRACE_EVENT_NAME_MAX\fR  \fB_POSIX_TRACE_EVENT_NAME_MAX\fR
\fB_SC_TRACE_INHERIT\fR \fB_POSIX_TRACE_INHERIT\fR
\fB_SC_TRACE_LOG\fR     \fB_POSIX_TRACE_LOG\fR
\fB_SC_TRACE_NAME_MAX\fR        \fB_POSIX_TRACE_NAME_MAX\fR
\fB_SC_TRACE_SYS_MAX\fR \fB_POSIX_TRACE_SYS_MAX\fR
\fB_SC_TRACE_USER_EVENT_MAX\fR  \fB_POSIX_TRACE_USER_EVENT_MAX\fR
\fB_SC_TYPED_MEMORY_OBJECTS\fR  \fB_POSIX_TYPED_MEMORY_OBJECTS\fR
```

```
.TE

.SH RETURN VALUES
.sp
.LP
Upon successful completion, \fBsysconf()\fR returns the current variable value
on the system. The value returned will not be more restrictive than the
corresponding value described to the application when it was compiled with the
implementation’s <\fBlimits.h\fR>, <\fBunistd.h\fR> or <\fBtime.h\fR>. With
only a few obvious exceptions such as \fB_SC_AVPHYS_PAGES\fR and
\fB_SC_NPROCESSORS_ONLN\fR, the value will not change during the lifetime of
the calling process.
.sp
.LP
If \fIname\fR is an invalid value, \fBsysconf()\fR returns \fB\(mi1\fR and sets
\fBerrno\fR to indicate the error. If the variable corresponding to \fIname\fR
is associated with functionality that is not supported by the system,
\fBsysconf()\fR returns \fB\(mi1\fR without changing the value of \fIerrno\fR.
.sp
.LP
Calling \fBsysconf()\fR with the following returns \fB\(mi1\fR without setting
\fBerrno\fR, because no maximum limit can be determined. The system supports at
least the minimum values and can support higher values depending upon system
resources.
.sp
.in +2
.nf
Variable                            Minimum supported value
_SC_AIO_MAX                        _POSIX_AIO_MAX
_SC_ATEXIT_MAX                     32
_SC_MQ_OPEN_MAX                    32
_SC_THREAD_THREADS_MAX             _POSIX_THREAD_THREADS_MAX
_SC_THREAD_KEYS_MAX                _POSIX_THREAD_KEYS_MAX
_SC_THREAD_DESTRUCTOR_ITERATIONS   _POSIX_THREAD_DESTRUCTOR_ITERATIONS
.fi
.in -2

.sp
.LP
The following SPARC and x86 platform variables return \fBEINVAL\fR:
.sp
.in +2
.nf
_SC_COHER_BLKSZ        _SC_DCACHE_ASSOC
_SC_DCACHE_BLKSZ       _SC_DCACHE_LINESZ
_SC_DCACHE_SZ          _SC_DCACHE_TBLKSZ
_SC_ICACHE_ASSOC       _SC_ICACHE_BLKSZ
_SC_ICACHE_LINESZ      _SC_ICACHE_SZ
_SC_SPLIT_CACHE
.fi
.in -2

.SH ERRORS
.sp
.LP
The \fBsysconf()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The value of the \fIname\fR argument is invalid.
.RE

.SH ATTRIBUTES
```

```
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Architecture    SPARC and x86
_
Interface Stability     Committed
_
MT-Level        MT-Safe, Async-Signal-Safe
_
Standard        See \fBstandards\fR(5).
.TE

.SH SEE ALSO
.sp
.LP
\fBpooladm\fR(1M), \fBzoneadm\fR(1M), \fBfpathconf\fR(2), \fBseteuid\fR(2),
\fBsetrlimit\fR(2), \fBconfstr\fR(3C), \fBattributes\fR(5), \fBstandards\fR(5)
.SH NOTES
.sp
.LP
A call to \fBsetrlimit()\fR can cause the value of \fBOPEN_MAX\fR to change.
.sp
.LP
Multiplying  \fBsysconf\fR(\fB_SC_PHYS_PAGES\fR) or
\fBsysconf\fR(\fB_SC_AVPHYS_PAGES\fR) by \fBsysconf\fR(\fB_SC_PAGESIZE\fR) to
determine memory amount in bytes can exceed the maximum values representable in
a 32-bit signed or unsigned integer.
.sp
.LP
The value of \fBCLK_TCK\fR can be variable and it should not be assumed that
\fBCLK_TCK\fR is a compile-time constant.
.sp
.LP
If the caller is in a non-global zone and the pools facility is active,
\fBsysconf\fR(\fB_SC_NPROCESSORS_CONF\fR) and
\fBsysconf\fR(\fB_SC_NPROCESSORS_ONLN\fR) return the number of processors in
the processor set of the pool to which the zone is bound.
```

```
**********************************************************
    9883 Mon Aug 12 18:24:55 2013
new/usr/src/man/man4/passwd.4
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
'\" te
.\" Copyright (c) 2013 Gary Mills
.\" Copyright (c) 2004, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 1989 AT&T
.\" 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 PASSWD 4 "Apr 16, 2013"
.TH PASSWD 4 "Jul 28, 2004"
.SH NAME
passwd \- password file
.SH SYNOPSIS
.LP
.nf
\fB/etc/passwd\fR
.fi

.SH DESCRIPTION
.sp
.LP
The file \fB/etc/passwd\fR is a local source of information about users'
accounts. The password file can be used in conjunction with other naming
sources, such as the \fBNIS\fR maps \fBpasswd.byname\fR and \fBpasswd.bygid\fR,
data from the \fBNIS+\fR \fBpasswd\fR table, or password data stored on an LDAP
server. Programs use the \fBgetpwnam\fR(3C) routines to access this
information.
.sp
.LP
Each \fBpasswd\fR entry is a single line of the form:
.sp
.in +2
.nf
\fIusername\fR\fB:\fR\fIpassword\fR\fB:\fR\fIuid\fR\fB:\fR
\fIgid\fR\fB:\fR\fIgcos-field\fR\fB:\fR\fIhome-dir\fR\fB:\fR
\fIlogin-shell\fR
.fi
.in -2
.sp

.sp
.LP
where
.sp
.ne 2
.na
\fB\fIusername\fR\fR
.ad
.RS 15n
is the user's login name.
.sp
The login (\fBlogin\fR) and role (\fBrole\fR) fields accept a string of no more
than 32 bytes consisting of characters from the set of alphabetic
than eight bytes consisting of characters from the set of alphabetic
characters, numeric characters, period (\fB\&.\fR), underscore (\fB_\fR), and
hyphen (\fB-\fR). The first character should be alphabetic and the field should
contain at least one lower case alphabetic character. A warning message is
displayed if these restrictions are not met.
.sp
The \fBlogin\fR and \fBrole\fR fields must contain at least one character and
must not contain a colon (\fB:\fR) or a newline (\fB\en\fR).
```

```
.RE

.sp
.ne 2
.na
\fB\fIpassword\fR\fR
.ad
.RS 15n
is an empty field. The encrypted password for the user is in the corresponding
entry in the \fB/etc/shadow\fR file. \fBpwconv\fR(1M) relies on a special value
of '\fBx\fR' in the password field of \fB/etc/passwd\fR. If this value
of '\fBx\fR' exists in the password field of \fB/etc/passwd\fR, this indicates
that the password for the user is already in \fB/etc/shadow\fR and should not
be modified.
.RE

.sp
.ne 2
.na
\fB\fIuid\fR\fR
.ad
.RS 15n
is the user's unique numerical \fBID\fR for the system.
.RE

.sp
.ne 2
.na
\fB\fIgid\fR\fR
.ad
.RS 15n
is the unique numerical \fBID\fR of the group that the user belongs to.
.RE

.sp
.ne 2
.na
\fB\fIgcos-field\fR\fR
.ad
.RS 15n
is the user's real name, along with information to pass along in a mail-message
heading. (It is called the gcos-field for historical reasons.) An ''\fB&\fR\&''
(ampersand) in this field stands for the login name (in cases where the login
name appears in a user's real name).
.RE

.sp
.ne 2
.na
\fB\fIhome-dir\fR\fR
.ad
.RS 15n
is the pathname to the directory in which the user is initially positioned upon
logging in.
.RE

.sp
.ne 2
.na
\fB\fIlogin-shell\fR\fR
.ad
.RS 15n
is the user's initial shell program. If this field is empty, the default shell
is \fB/usr/bin/sh\fR.
.RE
```

```
.sp
.LP
The maximum value of the \fIuid\fR and \fIgid\fR fields is \fB2147483647\fR. To
maximize interoperability and compatibility, administrators are recommended to
assign users a range of \fBUID\fRs and \fBGID\fRs below \fB60000\fR where
possible. (\fBUID\fRs from \fB0\fR-\fB99\fR inclusive are reserved by the
operating system vendor for use in future applications. Their use by end system
users or vendors of layered products is not supported and may cause security
related issues with future applications.)
.sp
.LP
The password file is an \fBASCII\fR file that resides in the \fB/etc\fR
directory. Because the encrypted passwords on a secure system are always kept
in the \fBshadow\fR file, \fB/etc/passwd\fR has general read permission on all
systems and can be used by routines that map between numerical user \fBID\fRs
and user names.
.sp
.LP
Blank lines are treated as malformed entries in the \fBpasswd\fR file and cause
consumers of the file , such as \fBgetpwnam\fR(3C), to fail.
.sp
.LP
The password file can contain entries beginning with a ‘+’ (plus sign) or ’-’
(minus sign) to selectively incorporate entries from another naming service
source, such as NIS, NIS+, or LDAP.
.sp
.LP
A line beginning with a ’+’ means to incorporate entries from the naming
service source. There are three styles of the ’+’ entries in this file. A
single + means to insert all the entries from the alternate naming service
source at that point, while a +\fIname\fR means to insert the specific entry,
if one exists, from the naming service source. A +@\fInetgroup\fR means to
insert the entries for all members of the network group \fInetgroup\fR from the
alternate naming service. If a +\fIname\fR entry has a non-null \fBpassword\fR,
\fIgcos\fR, \fIhome-dir\fR, or \fIlogin-shell\fR field, the value of that field
overrides what is contained in the alternate naming service. The \fIuid\fR and
\fIgid\fR fields cannot be overridden.
.sp
.LP
A line beginning with a ‘\(mi’ means to disallow entries from the alternate
naming service. There are two styles of ‘-‘ entries in this file. -\fIname\fR
means to disallow any subsequent entries (if any) for \fIname\fR (in this file
or in a naming service), and -@\fInetgroup\fR means to disallow any subsequent
entries for all members of the network group \fInetgroup\fR.
.sp
.LP
This is also supported by specifying ‘‘passwd : compat’’ in
\fBnsswitch.conf\fR(4). The "compat" source might not be supported in future
releases. The preferred sources are \fBfiles\fR followed by the identifier of a
name service, such as \fBnis\fR or \fBldap\fR. This has the effect of
incorporating the entire contents of the naming service’s \fBpasswd\fR database
or password-related information after the \fBpasswd\fR file.
.sp
.LP
Note that in compat mode, for every \fB/etc/passwd\fR entry, there must be a
corresponding entry in the \fB/etc/shadow\fR file.
.sp
.LP
Appropriate precautions must be taken to lock the \fB/etc/passwd\fR file
against simultaneous changes if it is to be edited with a text editor;
\fBvipw\fR(1B) does the necessary locking.
.SH EXAMPLES
.LP
\fBExample 1 \fRSample \fBpasswd\fR File
.sp
.LP
```

```
The following is a sample \fBpasswd\fR file:

.sp
.in +2
.nf
root:x:0:1:Super-User:/:/sbin/sh
fred:6k/7KCFRPNVXg:508:10:& Fredericks:/usr2/fred:/bin/csh
.fi
.in -2
.sp

.sp
.LP
and the sample password entry from \fBnsswitch.conf\fR:

.sp
.in +2
.nf
passwd: files ldap
.fi
.in -2
.sp

.sp
.LP
In this example, there are specific entries for users \fBroot\fR and \fBfred\fR
to assure that they can login even when the system is running single-user. In
addition, anyone whose password information is stored on an LDAP server will be
able to login with their usual password, shell, and home directory.

.sp
.LP
If the password file is:

.sp
.in +2
.nf
root:x:0:1:Super-User:/:/sbin/sh
fred:6k/7KCFRPNVXg:508:10:& Fredericks:/usr2/fred:/bin/csh
+
.fi
.in -2
.sp

.sp
.LP
and the password entry in \fBnsswitch.conf\fR is:

.sp
.in +2
.nf
passwd: compat
.fi
.in -2
.sp

.sp
.LP
then all the entries listed in the \fBNIS\fR \fBpasswd.byuid\fR and
\fBpasswd.byname\fR maps will be effectively incorporated after the entries for
\fBroot\fR and \fBfred\fR. If the password entry in \fBnsswitch.conf\fR is:

.sp
.in +2
.nf
passwd_compat: ldap
```

```
passwd: compat
.fi
.in -2

.sp
.LP
then all password-related entries stored on the LDAP server will be
incorporated after the entries for \fBroot\fR and \fBfred\fR.

.sp
.LP
The following is a sample \fBpasswd\fR file when \fBshadow\fR does not exist:

.sp
.in +2
.nf
root:q.mJzTnu8icf.:0:1:Super-User:/:/sbin/sh
fred:6k/7KCFRPNVXg:508:10:& Fredericks:/usr2/fred:/bin/csh
+john:
+@documentation:no-login:
+::::Guest
.fi
.in -2
.sp

.sp
.LP
The following is a sample \fBpasswd\fR file when \fBshadow\fR does exist:

.sp
.in +2
.nf
root:##root:0:1:Super-User:/:/sbin/sh
fred:##fred:508:10:& Fredericks:/usr2/fred:/bin/csh
+john:
+@documentation:no-login:
+::::Guest
.fi
.in -2
.sp

.sp
.LP
In this example, there are specific entries for users \fBroot\fR and
\fBfred\fR, to assure that they can log in even when the system is running
standalone. The user \fBjohn\fR will have his password entry in the naming
service source incorporated without change, anyone in the netgroup
\fBdocumentation\fR will have their password field disabled, and anyone else
will be able to log in with their usual password, shell, and home directory,
but with a \fIgcos\fR field of \fBGuest\fR

.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/nsswitch.conf\fR\fR
.ad
.RS 22n

.RE

.sp
.ne 2
.na
\fB\fB/etc/passwd\fR\fR
.ad
```

```
.RS 22n

.RE

.sp
.ne 2
.na
\fB\fB/etc/shadow\fR\fR
.ad
.RS 22n

.RE

.SH SEE ALSO
.sp
.LP
\fBchgrp\fR(1), \fBchown\fR(1), \fBfinger\fR(1), \fBgroups\fR(1),
\fBlogin\fR(1), \fBnewgrp\fR(1), \fBnispasswd\fR(1), \fBpasswd\fR(1),
\fBsh\fR(1), \fBsort\fR(1), \fBdomainname\fR(1M), \fBgetent\fR(1M),
\fBin.ftpd\fR(1M), \fBpassmgmt\fR(1M), \fBpwck\fR(1M), \fBpwconv\fR(1M),
\fBsu\fR(1M), \fBuseradd\fR(1M), \fBuserdel\fR(1M), \fBusermod\fR(1M),
\fBa64l\fR(3C), \fBcrypt\fR(3C), \fBgetpw\fR(3C), \fBgetpwnam\fR(3C),
\fBgetspnam\fR(3C), \fBputpwent\fR(3C), \fBgroup\fR(4), \fBhosts.equiv\fR(4),
\fBnsswitch.conf\fR(4), \fBshadow\fR(4), \fBenviron\fR(5),
\fBunistd.h\fR(3HEAD)
.sp
.LP
\fISystem Administration Guide: Basic Administration\fR
```

```
**********************************************************
   89641 Mon Aug 12 18:24:55 2013
new/usr/src/pkg/manifests/SUNWcs.mf
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
#
# 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 2012 Nexenta Systems, Inc. All rights reserved.
# Copyright (c) 2013 Gary Mills
#

<include SUNWcs.man1.inc>
<include SUNWcs.man1m.inc>
<include SUNWcs.man4.inc>
<include SUNWcs.man5.inc>
<include SUNWcs.man7d.inc>
<include SUNWcs.man7fs.inc>
set name=pkg.fmri value=pkg:/SUNWcs@$(PKGVERS)
set name=pkg.description \
    value="core software for a specific instruction-set architecture"
set name=pkg.summary value="Core Solaris"
set name=info.classification value=org.opensolaris.category.2008:System/Core
set name=variant.arch value=$(ARCH)
dir path=dev group=sys
dir path=etc group=sys
dir path=etc/cron.d group=sys
dir path=etc/crypto group=sys
dir path=etc/crypto/certs group=sys
dir path=etc/crypto/crls group=sys
dir path=etc/default group=sys
dir path=etc/dev group=sys
dir path=etc/devices group=sys
dir path=etc/dfs group=sys
dir path=etc/dhcp group=sys
dir path=etc/fs group=sys
dir path=etc/fs/dev group=sys
dir path=etc/fs/hsfs group=sys
dir path=etc/fs/ufs group=sys
dir path=etc/ftpd group=sys
dir path=etc/inet group=sys
dir path=etc/init.d group=sys
dir path=etc/lib group=sys
dir path=etc/logadm.d group=sys
dir path=etc/mail group=mail
```

```
dir path=etc/net group=sys
dir path=etc/net/ticlts group=sys
dir path=etc/net/ticots group=sys
dir path=etc/net/ticotsord group=sys
dir path=etc/opt group=sys
dir path=etc/rc0.d group=sys
dir path=etc/rc1.d group=sys
dir path=etc/rc2.d group=sys
dir path=etc/rc3.d group=sys
dir path=etc/rcS.d group=sys
dir path=etc/rpcsec group=sys
dir path=etc/saf
dir path=etc/saf/zsmon group=sys
dir path=etc/sasl group=sys
dir path=etc/security group=sys
dir path=etc/security/audit group=sys
dir path=etc/security/audit/localhost group=sys
dir path=etc/security/auth_attr.d group=sys
dir path=etc/security/dev group=sys
dir path=etc/security/exec_attr.d group=sys
dir path=etc/security/lib group=sys
dir path=etc/security/prof_attr.d group=sys
dir path=etc/skel group=sys
dir path=etc/svc group=sys
dir path=etc/svc/profile group=sys
dir path=etc/svc/profile/site group=sys
dir path=etc/svc/volatile group=sys
dir path=etc/sysevent group=sys
dir path=etc/sysevent/config group=sys
dir path=etc/tm group=sys
dir path=etc/user_attr.d group=sys
dir path=export group=sys
dir path=home group=root mode=0555
dir path=lib
dir path=lib/crypto
dir path=lib/inet
dir path=lib/svc
dir path=lib/svc/bin
dir path=lib/svc/capture
dir path=lib/svc/manifest group=sys
dir path=lib/svc/manifest/application group=sys
dir path=lib/svc/manifest/application/management group=sys
dir path=lib/svc/manifest/application/security group=sys
dir path=lib/svc/manifest/device group=sys
dir path=lib/svc/manifest/milestone group=sys
dir path=lib/svc/manifest/network group=sys
dir path=lib/svc/manifest/network/dns group=sys
dir path=lib/svc/manifest/network/ipsec group=sys
dir path=lib/svc/manifest/network/ldap group=sys
dir path=lib/svc/manifest/network/routing group=sys
dir path=lib/svc/manifest/network/rpc group=sys
dir path=lib/svc/manifest/network/shares group=sys
dir path=lib/svc/manifest/network/ssl group=sys
dir path=lib/svc/manifest/platform group=sys
$(sparc_ONLY)dir path=lib/svc/manifest/platform/sun4u group=sys
dir path=lib/svc/manifest/site group=sys
dir path=lib/svc/manifest/system group=sys
dir path=lib/svc/manifest/system/device group=sys
dir path=lib/svc/manifest/system/filesystem group=sys
dir path=lib/svc/manifest/system/security group=sys
dir path=lib/svc/manifest/system/svc group=sys
dir path=lib/svc/method
dir path=lib/svc/monitor
dir path=lib/svc/seed
dir path=lib/svc/share
dir path=mnt group=sys
dir path=opt group=sys
dir path=proc group=root mode=0555
dir path=root group=root mode=0700
dir path=sbin group=sys
dir path=system group=root
dir path=system/contract group=root mode=0555
dir path=system/object group=root mode=0555
dir path=tmp group=sys mode=1777
dir path=usr group=sys
dir path=usr/bin
dir path=usr/bin/$(ARCH32)
dir path=usr/bin/$(ARCH64)
dir path=usr/ccs
dir path=usr/ccs/bin
dir path=usr/demo
dir path=usr/games
dir path=usr/has
dir path=usr/has/bin
dir path=usr/has/lib
dir path=usr/has/man
dir path=usr/has/man/man1has
dir path=usr/kernel group=sys
dir path=usr/kernel/drv group=sys
dir path=usr/kernel/drv/$(ARCH64) group=sys
dir path=usr/kernel/exec group=sys
dir path=usr/kernel/exec/$(ARCH64) group=sys
dir path=usr/kernel/fs group=sys
dir path=usr/kernel/fs/$(ARCH64) group=sys
dir path=usr/kernel/pcbe group=sys
dir path=usr/kernel/pcbe/$(ARCH64) group=sys
dir path=usr/kernel/sched group=sys
dir path=usr/kernel/sched/$(ARCH64) group=sys
dir path=usr/kernel/strmod group=sys
dir path=usr/kernel/strmod/$(ARCH64) group=sys
dir path=usr/kernel/sys group=sys
dir path=usr/kernel/sys/$(ARCH64) group=sys
dir path=usr/kvm
dir path=usr/lib
dir path=usr/lib/$(ARCH64)
dir path=usr/lib/audit
dir path=usr/lib/class
dir path=usr/lib/class/FX
dir path=usr/lib/class/IA
dir path=usr/lib/class/RT
dir path=usr/lib/class/SDC
dir path=usr/lib/class/TS
dir path=usr/lib/crypto
dir path=usr/lib/devfsadm group=sys
dir path=usr/lib/devfsadm/linkmod group=sys
dir path=usr/lib/fs group=sys
dir path=usr/lib/fs/autofs group=sys
dir path=usr/lib/fs/autofs/$(ARCH64) group=sys
dir path=usr/lib/fs/cachefs group=sys
dir path=usr/lib/fs/ctfs group=sys
dir path=usr/lib/fs/dev group=sys
dir path=usr/lib/fs/fd group=sys
dir path=usr/lib/fs/hsfs group=sys
dir path=usr/lib/fs/lofs group=sys
dir path=usr/lib/fs/mntfs group=sys
dir path=usr/lib/fs/nfs group=sys
dir path=usr/lib/fs/nfs/$(ARCH64) group=sys
dir path=usr/lib/fs/objfs group=sys
dir path=usr/lib/fs/proc group=sys
dir path=usr/lib/fs/sharefs group=sys
dir path=usr/lib/fs/tmpfs group=sys
dir path=usr/lib/fs/ufs group=sys
```

```
dir path=usr/lib/help
dir path=usr/lib/help/auths
dir path=usr/lib/help/auths/locale
dir path=usr/lib/help/auths/locale/C
dir path=usr/lib/help/profiles
dir path=usr/lib/help/profiles/locale
dir path=usr/lib/help/profiles/locale/C
dir path=usr/lib/iconv
dir path=usr/lib/inet
dir path=usr/lib/inet/$(ARCH32)
dir path=usr/lib/inet/$(ARCH64)
dir path=usr/lib/inet/dhcp
dir path=usr/lib/inet/dhcp/nsu
dir path=usr/lib/inet/dhcp/svc
dir path=usr/lib/locale
dir path=usr/lib/locale/C
dir path=usr/lib/locale/C/LC_COLLATE
dir path=usr/lib/locale/C/LC_CTYPE
dir path=usr/lib/locale/C/LC_MESSAGES
dir path=usr/lib/locale/C/LC_MONETARY
dir path=usr/lib/locale/C/LC_NUMERIC
dir path=usr/lib/locale/C/LC_TIME
dir path=usr/lib/netsvc group=sys
dir path=usr/lib/pci
dir path=usr/lib/rcm
dir path=usr/lib/rcm/modules
dir path=usr/lib/rcm/scripts
dir path=usr/lib/reparse
dir path=usr/lib/saf
dir path=usr/lib/secure
dir path=usr/lib/secure/$(ARCH64)
dir path=usr/lib/security
dir path=usr/lib/sysevent
dir path=usr/lib/sysevent/modules
dir path=usr/net group=sys
dir path=usr/net/nls group=sys
dir path=usr/net/servers group=sys
dir path=usr/old
dir path=usr/platform group=sys
dir path=usr/sadm
dir path=usr/sadm/bin
dir path=usr/sadm/install
dir path=usr/sadm/install/scripts
dir path=usr/sbin
$(i386_ONLY)dir path=usr/sbin/$(ARCH32)
dir path=usr/sbin/$(ARCH64)
dir path=usr/share
dir path=usr/share/doc group=other
dir path=usr/share/doc/ksh
dir path=usr/share/doc/ksh/images
dir path=usr/share/doc/ksh/images/callouts
dir path=usr/share/lib
dir path=usr/share/lib/mailx
dir path=usr/share/lib/pub
dir path=usr/share/lib/tabset
dir path=usr/share/lib/xml group=sys
dir path=usr/share/lib/xml/dtd group=sys
dir path=usr/share/lib/xml/style group=sys
dir path=usr/share/man
dir path=usr/share/man/man1
dir path=usr/share/man/man1m
dir path=usr/share/man/man4
dir path=usr/share/man/man5
dir path=usr/share/man/man7d
dir path=usr/share/man/man7fs
dir path=usr/share/src group=sys
dir path=var group=sys
dir path=var/adm group=sys mode=0775
dir path=var/adm/exacct group=adm owner=adm
dir path=var/adm/log group=adm owner=adm
dir path=var/adm/streams group=sys
dir path=var/audit group=sys
dir path=var/cores group=sys
dir path=var/cron group=sys
dir path=var/games
dir path=var/idmap group=daemon owner=daemon
dir path=var/inet group=sys
dir path=var/ld
dir path=var/ld/$(ARCH64)
dir path=var/log group=sys
dir path=var/logadm
dir path=var/mail group=mail mode=1777
dir path=var/mail/:saved group=mail mode=0775
dir path=var/news
dir path=var/opt group=sys
dir path=var/preserve mode=1777
dir path=var/run group=sys
dir path=var/sadm group=sys
dir path=var/sadm/system group=sys
dir path=var/sadm/system/admin group=sys
dir path=var/saf
dir path=var/saf/zsmon group=sys
dir path=var/spool
dir path=var/spool/cron group=sys
dir path=var/spool/cron/atjobs group=sys
dir path=var/spool/cron/crontabs group=sys
dir path=var/spool/locks group=uucp owner=uucp
dir path=var/svc group=sys
dir path=var/svc/log group=sys
dir path=var/svc/manifest group=sys
dir path=var/svc/manifest/application group=sys
dir path=var/svc/manifest/application/management group=sys
dir path=var/svc/manifest/application/print group=sys
dir path=var/svc/manifest/application/security group=sys
dir path=var/svc/manifest/device group=sys
dir path=var/svc/manifest/milestone group=sys
dir path=var/svc/manifest/network group=sys
dir path=var/svc/manifest/network/dns group=sys
dir path=var/svc/manifest/network/ipsec group=sys
dir path=var/svc/manifest/network/ldap group=sys
dir path=var/svc/manifest/network/nfs group=sys
dir path=var/svc/manifest/network/nis group=sys
dir path=var/svc/manifest/network/routing group=sys
dir path=var/svc/manifest/network/rpc group=sys
dir path=var/svc/manifest/network/security group=sys
dir path=var/svc/manifest/network/shares group=sys
dir path=var/svc/manifest/network/ssl group=sys
dir path=var/svc/manifest/platform group=sys
$(sparc_ONLY)dir path=var/svc/manifest/platform/sun4u group=sys
$(sparc_ONLY)dir path=var/svc/manifest/platform/sun4v group=sys
dir path=var/svc/manifest/site group=sys
dir path=var/svc/manifest/system group=sys
dir path=var/svc/manifest/system/device group=sys
dir path=var/svc/manifest/system/filesystem group=sys
dir path=var/svc/manifest/system/security group=sys
dir path=var/svc/manifest/system/svc group=sys
dir path=var/svc/profile group=sys
dir path=var/tmp group=sys mode=1777
driver name=dump perms="dump 0660 root sys"
driver name=fssnap \
    policy="ctl read_priv_set=sys_config write_priv_set=sys_config" \
    perms="* 0640 root sys" perms="ctl 0666 root sys"
```

```
driver name=kstat perms="* 0666 root sys"
driver name=ksyms perms="* 0666 root sys"
driver name=logindmux
driver name=ptm clone_perms="ptmx 0666 root sys"
driver name=pts perms="* 0644 root sys" perms="0 0620 root tty" \
    perms="1 0620 root tty" perms="2 0620 root tty" perms="3 0620 root tty"
file path=etc/.login group=sys preserve=renamenew
file path=etc/cron.d/.proto group=sys mode=0744
file path=etc/cron.d/at.deny group=sys preserve=true
file path=etc/cron.d/cron.deny group=sys preserve=true
file path=etc/cron.d/queuedefs group=sys
file path=etc/crypto/kmf.conf group=sys preserve=true
file path=etc/crypto/pkcs11.conf group=sys preserve=true
file path=etc/datemsk group=sys mode=0444
file path=etc/default/cron group=sys preserve=true
file path=etc/default/devfsadm group=sys preserve=true
file path=etc/default/fs group=sys preserve=true
file path=etc/default/init group=sys preserve=true
file path=etc/default/keyserv group=sys preserve=true
file path=etc/default/login group=sys preserve=true
file path=etc/default/nss group=sys preserve=true
file path=etc/default/passwd group=sys preserve=true
file path=etc/default/su group=sys preserve=true
file path=etc/default/syslogd group=sys preserve=true
file path=etc/default/tar group=sys preserve=true
file path=etc/default/useradd group=sys preserve=true
file path=etc/default/utmpd group=sys preserve=true
file path=etc/dev/reserved_devnames group=sys preserve=true
file path=etc/device.tab group=root mode=0444 preserve=true
file path=etc/dfs/dfstab group=sys preserve=true
file path=etc/dfs/fstypes group=root preserve=true
file path=etc/dfs/sharetab group=root mode=0444 preserve=true
file path=etc/dgroup.tab group=sys mode=0444 preserve=true
file path=etc/dhcp/inittab group=sys preserve=true
file path=etc/dhcp/inittab6 group=sys preserve=true
file path=etc/dumpdates group=sys mode=0664 preserve=true
file path=etc/format.dat group=sys preserve=true
file path=etc/fs/dev/mount mode=0555
file path=etc/fs/hsfs/mount mode=0555
file path=etc/fs/ufs/mount mode=0555
file path=etc/ftpd/ftpusers group=sys preserve=true
file path=etc/group group=sys preserve=true
file path=etc/inet/hosts group=sys preserve=true
file path=etc/inet/inetd.conf group=sys preserve=true
file path=etc/inet/ipaddrsel.conf group=sys preserve=true
file path=etc/inet/netmasks group=sys preserve=true
file path=etc/inet/networks group=sys preserve=true
file path=etc/inet/protocols group=sys preserve=true
file path=etc/inet/services group=sys preserve=true
file path=etc/inet/wanboot.conf.sample group=sys mode=0444
file path=etc/init.d/PRESERVE group=sys mode=0744 preserve=true
file path=etc/init.d/README group=sys preserve=true
file path=etc/init.d/cachefs.daemon group=sys mode=0744 preserve=true
file path=etc/init.d/ldap.client group=sys mode=0744
file path=etc/init.d/nscd group=sys mode=0744
file path=etc/init.d/sysetup group=sys mode=0744 preserve=true
file path=etc/init.d/ufs_quota group=sys mode=0744 preserve=true
file path=etc/inittab group=sys preserve=true
file path=etc/ioctl.syscon group=sys preserve=true
file path=etc/ksh.kshrc group=sys preserve=renameold
file path=etc/logadm.conf group=sys preserve=true timestamp=19700101T000000Z
file path=etc/logindevperm group=sys preserve=true
file path=etc/magic mode=0444
file path=etc/mail/mailx.rc preserve=true
file path=etc/mailcap preserve=true
file path=etc/mime.types preserve=true
file path=etc/mnttab group=root mode=0444 preserve=true
file path=etc/motd group=sys preserve=true
file path=etc/net/ticlts/hosts group=sys
file path=etc/net/ticlts/services group=sys preserve=true
file path=etc/net/ticots/hosts group=sys
file path=etc/net/ticots/services group=sys preserve=true
file path=etc/net/ticotsord/hosts group=sys
file path=etc/net/ticotsord/services group=sys preserve=true
file path=etc/netconfig group=sys preserve=true
file path=etc/nscd.conf group=sys preserve=true
file path=etc/nsswitch.ad group=sys
file path=etc/nsswitch.conf group=sys preserve=true
file path=etc/nsswitch.dns group=sys
file path=etc/nsswitch.files group=sys
file path=etc/nsswitch.ldap group=sys
file path=etc/pam.conf group=sys preserve=true
file path=etc/passwd group=sys preserve=true
file path=etc/profile group=sys preserve=true
file path=etc/project group=sys preserve=true
file path=etc/rc2.d/README group=sys
file path=etc/rc3.d/README group=sys
file path=etc/rcS.d/README group=sys
file path=etc/remote preserve=true
file path=etc/rpc group=sys preserve=true
file path=etc/saf/_sactab group=sys preserve=true
file path=etc/saf/_sysconfig group=sys preserve=true
file path=etc/saf/zsmon/_pmtab group=sys preserve=true
file path=etc/security/audit_class group=sys preserve=renamenew
file path=etc/security/audit_event group=sys preserve=renamenew
file path=etc/security/audit_warn group=sys mode=0740 preserve=renamenew
file path=etc/security/auth_attr group=sys preserve=true \
    timestamp=19700101T000000Z
file path=etc/security/auth_attr.d/SUNWcs group=sys
file path=etc/security/crypt.conf group=sys preserve=renamenew
file path=etc/security/dev/audio mode=0400
file path=etc/security/dev/fd0 mode=0400
file path=etc/security/dev/sr0 mode=0400
file path=etc/security/dev/st0 mode=0400
file path=etc/security/dev/st1 mode=0400
file path=etc/security/exec_attr group=sys preserve=true \
    timestamp=19700101T000000Z
file path=etc/security/exec_attr.d/SUNWcs group=sys
file path=etc/security/kmfpolicy.xml
file path=etc/security/lib/audio_clean group=sys mode=0555
file path=etc/security/lib/fd_clean group=sys mode=0555
file path=etc/security/lib/sr_clean group=sys mode=0555
file path=etc/security/lib/st_clean group=sys mode=0555
file path=etc/security/policy.conf group=sys preserve=true
file path=etc/security/priv_names group=sys preserve=renameold
file path=etc/security/prof_attr group=sys preserve=true \
    timestamp=19700101T000000Z
file path=etc/security/prof_attr.d/SUNWcs group=sys
file path=etc/shadow group=sys mode=0400 preserve=true
file path=etc/skel/.profile group=other preserve=true
file path=etc/skel/local.cshrc group=sys preserve=true
file path=etc/skel/local.login group=sys preserve=true
file path=etc/skel/local.profile group=sys preserve=true
file path=etc/svc/profile/generic_limited_net.xml group=sys mode=0444
file path=etc/svc/profile/generic_open.xml group=sys mode=0444
file path=etc/svc/profile/inetd_generic.xml group=sys mode=0444
file path=etc/svc/profile/inetd_upgrade.xml group=sys mode=0444
file path=etc/svc/profile/ns_dns.xml group=sys mode=0444
file path=etc/svc/profile/ns_files.xml group=sys mode=0444
file path=etc/svc/profile/ns_ldap.xml group=sys mode=0444
file path=etc/svc/profile/ns_nis.xml group=sys mode=0444
file path=etc/svc/profile/ns_none.xml group=sys mode=0444
```

```
$(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,SPARC-Enterprise.xml \
    group=sys mode=0444
$(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire-15000.xml \
    group=sys mode=0444
$(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire-880.xml \
    group=sys mode=0444
$(sparc_ONLY)file path=etc/svc/profile/platform_SUNW,Sun-Fire.xml group=sys \
    mode=0444
$(sparc_ONLY)file \
    path=etc/svc/profile/platform_SUNW,Ultra-Enterprise-10000.xml group=sys \
    mode=0444
$(sparc_ONLY)file \
    path=etc/svc/profile/platform_SUNW,UltraSPARC-IIi-Netract.xml group=sys \
    mode=0444
file path=etc/svc/profile/platform_none.xml group=sys mode=0444
$(sparc_ONLY)file path=etc/svc/profile/platform_sun4v.xml group=sys mode=0444
file path=etc/sysevent/config/README group=sys mode=0444
file path=etc/sysevent/config/SUNW,EC_dr,ESC_dr_req,sysevent.conf group=sys
file path=etc/syslog.conf group=sys preserve=true
file path=etc/ttydefs group=sys preserve=true
file path=etc/ttysrch group=sys preserve=true
file path=etc/user_attr group=sys preserve=true timestamp=19700101T000000Z
file path=etc/user_attr.d/SUNWcs group=sys
file path=etc/vfstab group=sys preserve=true
file path=lib/inet/in.mpathd mode=0555
file path=lib/inet/ipmgmtd mode=0555
file path=lib/inet/netcfgd mode=0555
file path=lib/inet/nwamd mode=0555
file path=lib/svc/bin/lsvcrun group=sys mode=0555
file path=lib/svc/bin/mfstscan group=sys mode=0555
file path=lib/svc/bin/restore_repository group=sys mode=0555
file path=lib/svc/bin/sqlite group=sys mode=0555
file path=lib/svc/bin/svc.configd group=sys mode=0555
file path=lib/svc/bin/svc.ipfd group=sys mode=0555
file path=lib/svc/bin/svc.startd group=sys mode=0555
file path=lib/svc/manifest/milestone/multi-user-server.xml group=sys mode=0444
file path=lib/svc/manifest/milestone/multi-user.xml group=sys mode=0444
file path=lib/svc/manifest/milestone/name-services.xml group=sys mode=0444
file path=lib/svc/manifest/milestone/network.xml group=sys mode=0444
file path=lib/svc/manifest/milestone/single-user.xml group=sys mode=0444
file path=lib/svc/manifest/milestone/sysconfig.xml group=sys mode=0444
file path=lib/svc/manifest/network/dlmgmt.xml group=sys mode=0444
file path=lib/svc/manifest/network/dns/client.xml group=sys mode=0444
file path=lib/svc/manifest/network/dns/install.xml group=sys mode=0444
file path=lib/svc/manifest/network/forwarding.xml group=sys mode=0444
file path=lib/svc/manifest/network/inetd-upgrade.xml group=sys mode=0444
file path=lib/svc/manifest/network/inetd.xml group=sys mode=0444
file path=lib/svc/manifest/network/ipsec/ike.xml group=sys mode=0444
file path=lib/svc/manifest/network/ipsec/ipsecalgs.xml group=sys mode=0444
file path=lib/svc/manifest/network/ipsec/manual-key.xml group=sys mode=0444
file path=lib/svc/manifest/network/ipsec/policy.xml group=sys mode=0444
file path=lib/svc/manifest/network/ldap/client.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-initial.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-install.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-ipmgmt.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-ipqos.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-iptun.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-location.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-loopback.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-netcfg.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-netmask.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-physical.xml group=sys mode=0444
file path=lib/svc/manifest/network/network-routing-setup.xml group=sys \
    mode=0444
file path=lib/svc/manifest/network/network-service.xml group=sys mode=0444
file path=lib/svc/manifest/network/routing/legacy-routing.xml group=sys \
    mode=0444
file path=lib/svc/manifest/network/rpc/bind.xml group=sys mode=0444
file path=lib/svc/manifest/network/rpc/keyserv.xml group=sys mode=0444
file path=lib/svc/manifest/network/shares/group.xml group=sys mode=0444
file path=lib/svc/manifest/network/shares/reparsed.xml group=sys mode=0444
file path=lib/svc/manifest/network/socket-filter-kssl.xml group=sys mode=0444
file path=lib/svc/manifest/network/ssl/kssl-proxy.xml group=sys mode=0444
file path=lib/svc/manifest/system/auditd.xml group=sys mode=0444
file path=lib/svc/manifest/system/auditset.xml group=sys mode=0444
file path=lib/svc/manifest/system/boot-archive-update.xml group=sys mode=0444
file path=lib/svc/manifest/system/boot-archive.xml group=sys mode=0444
file path=lib/svc/manifest/system/boot-config.xml group=sys mode=0444
file path=lib/svc/manifest/system/consadm.xml group=sys mode=0444
file path=lib/svc/manifest/system/console-login.xml group=sys mode=0444
file path=lib/svc/manifest/system/coreadm.xml group=sys mode=0444
file path=lib/svc/manifest/system/cron.xml group=sys mode=0444
file path=lib/svc/manifest/system/cryptosvc.xml group=sys mode=0444
file path=lib/svc/manifest/system/device/allocate.xml group=sys mode=0444
file path=lib/svc/manifest/system/device/devices-audio.xml group=sys mode=0444
file path=lib/svc/manifest/system/device/devices-local.xml group=sys mode=0444
file path=lib/svc/manifest/system/device/mpxio-upgrade.xml group=sys mode=0444
file path=lib/svc/manifest/system/early-manifest-import.xml group=sys \
    mode=0444
file path=lib/svc/manifest/system/extended-accounting.xml group=sys mode=0444
file path=lib/svc/manifest/system/filesystem/local-fs.xml group=sys mode=0444
file path=lib/svc/manifest/system/filesystem/minimal-fs.xml group=sys \
    mode=0444
file path=lib/svc/manifest/system/filesystem/root-fs.xml group=sys mode=0444
file path=lib/svc/manifest/system/filesystem/usr-fs.xml group=sys mode=0444
$(i386_ONLY)file path=lib/svc/manifest/system/hostid.xml group=sys mode=0444
file path=lib/svc/manifest/system/hotplug.xml group=sys mode=0444
file path=lib/svc/manifest/system/identity.xml group=sys mode=0444
file path=lib/svc/manifest/system/idmap.xml group=sys mode=0444
file path=lib/svc/manifest/system/keymap.xml group=sys mode=0444
file path=lib/svc/manifest/system/logadm-upgrade.xml group=sys mode=0444
file path=lib/svc/manifest/system/manifest-import.xml group=sys mode=0444
file path=lib/svc/manifest/system/name-service-cache.xml group=sys mode=0444
file path=lib/svc/manifest/system/pfexecd.xml group=sys mode=0444
file path=lib/svc/manifest/system/rbac.xml group=sys mode=0444
file path=lib/svc/manifest/system/rmtmpfiles.xml group=sys mode=0444
file path=lib/svc/manifest/system/sac.xml group=sys mode=0444
file path=lib/svc/manifest/system/svc/global.xml group=sys mode=0444
file path=lib/svc/manifest/system/svc/restarter.xml group=sys mode=0444
file path=lib/svc/manifest/system/system-log.xml group=sys mode=0444
file path=lib/svc/manifest/system/utmp.xml group=sys mode=0444
file path=lib/svc/manifest/system/vtdaemon.xml group=sys mode=0444
file path=lib/svc/method/boot-archive mode=0555
file path=lib/svc/method/boot-archive-update mode=0555
file path=lib/svc/method/console-login mode=0555
file path=lib/svc/method/devices-audio mode=0555
file path=lib/svc/method/devices-local mode=0555
file path=lib/svc/method/dns-install mode=0555
file path=lib/svc/method/fs-local mode=0555
file path=lib/svc/method/fs-minimal mode=0555
file path=lib/svc/method/fs-root mode=0555
file path=lib/svc/method/fs-usr mode=0555
file path=lib/svc/method/identity-domain mode=0555
file path=lib/svc/method/identity-node mode=0555
file path=lib/svc/method/inetd-upgrade mode=0555
file path=lib/svc/method/keymap mode=0555
file path=lib/svc/method/ldap-client mode=0555
file path=lib/svc/method/logadm-upgrade mode=0555
file path=lib/svc/method/manifest-import mode=0555
file path=lib/svc/method/mpxio-upgrade mode=0555
file path=lib/svc/method/net-init mode=0555
file path=lib/svc/method/net-install mode=0555
```

```
file path=lib/svc/method/net-ipmgmt mode=0555
file path=lib/svc/method/net-ipqos mode=0555
file path=lib/svc/method/net-iptun mode=0555
file path=lib/svc/method/net-loc mode=0555
file path=lib/svc/method/net-loopback mode=0555
file path=lib/svc/method/net-netmask mode=0555
file path=lib/svc/method/net-nwam mode=0555
file path=lib/svc/method/net-physical mode=0555
file path=lib/svc/method/net-routing-setup mode=0555
file path=lib/svc/method/net-svc mode=0555
file path=lib/svc/method/rmtmpfiles mode=0555
file path=lib/svc/method/rpc-bind mode=0555
file path=lib/svc/method/svc-allocate mode=0555
file path=lib/svc/method/svc-auditd mode=0555
file path=lib/svc/method/svc-auditset mode=0555
file path=lib/svc/method/svc-boot-config mode=0555
file path=lib/svc/method/svc-consadm mode=0555
file path=lib/svc/method/svc-cron mode=0555
file path=lib/svc/method/svc-dlmgmtd mode=0555
file path=lib/svc/method/svc-forwarding mode=0555
$(i386_ONLY)file path=lib/svc/method/svc-hostid mode=0555
file path=lib/svc/method/svc-hotplug mode=0555
file path=lib/svc/method/svc-legacy-routing mode=0555
file path=lib/svc/method/svc-nscd mode=0555
file path=lib/svc/method/svc-rbac mode=0555
file path=lib/svc/method/svc-sockfilter mode=0555
file path=lib/svc/method/svc-utmpd mode=0555
file path=lib/svc/method/system-log mode=0555
file path=lib/svc/method/vtdaemon mode=0555
file path=lib/svc/method/yp mode=0555
# global.db is not needed in non-global zones, and it’s pretty large.
file path=lib/svc/seed/global.db group=sys mode=0444 \
    variant.opensolaris.zone=global
# symmetrically, nonglobal.db is not needed in global zones.
file path=lib/svc/seed/nonglobal.db group=sys mode=0444 \
    variant.opensolaris.zone=nonglobal
file path=lib/svc/share/README mode=0444
file path=lib/svc/share/fs_include.sh mode=0444
file path=lib/svc/share/ipf_include.sh mode=0444
file path=lib/svc/share/mfsthistory mode=0444
file path=lib/svc/share/net_include.sh mode=0444
file path=lib/svc/share/routing_include.sh mode=0444
file path=lib/svc/share/smf_include.sh mode=0444
file path=root/.bashrc group=root preserve=true
file path=root/.profile group=root preserve=true
file path=sbin/autopush mode=0555
$(i386_ONLY)file path=sbin/biosdev mode=0555
file path=sbin/bootadm mode=0555
file path=sbin/cryptoadm mode=0555
file path=sbin/devprop mode=0555
file path=sbin/dhcpagent mode=0555
file path=sbin/dhcpinfo mode=0555
file path=sbin/dlmgmtd mode=0555
file path=sbin/fdisk mode=0555
file path=sbin/fiocompress mode=0555
file path=sbin/hostconfig mode=0555
file path=sbin/ifconfig mode=0555
file path=sbin/ifparse mode=0555
file path=sbin/init group=sys mode=0555
$(i386_ONLY)file path=sbin/installgrub group=sys mode=0555
file path=sbin/ipmpstat mode=0555
file path=sbin/mount mode=0555
file path=sbin/mountall group=sys mode=0555
file path=sbin/netstrategy mode=0555
file path=sbin/rc0 group=sys mode=0744
file path=sbin/rc1 group=sys mode=0744
file path=sbin/rc2 group=sys mode=0744
file path=sbin/rc3 group=sys mode=0744
file path=sbin/rcS group=sys mode=0744
file path=sbin/route mode=0555
file path=sbin/routeadm mode=0555
file path=sbin/soconfig mode=0555
file path=sbin/su.static group=sys mode=0555
file path=sbin/sulogin mode=0555
file path=sbin/swapadd group=sys mode=0744
file path=sbin/sync mode=0555
file path=sbin/tzreload mode=0555
file path=sbin/uadmin group=sys mode=0555
file path=sbin/umount mode=0555
file path=sbin/umountall group=sys mode=0555
file path=sbin/uname mode=0555
file path=sbin/wusbadm mode=0555
file path=sbin/zonename mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/amt mode=0555
file path=usr/bin/$(ARCH32)/decrypt mode=0555
file path=usr/bin/$(ARCH32)/digest mode=0555
file path=usr/bin/$(ARCH32)/ksh93 mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/newtask group=sys mode=4555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/nohup mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/prctl mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/prstat mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/ps mode=0555
file path=usr/bin/$(ARCH32)/savecore mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/setuname mode=0555
$(i386_ONLY)file path=usr/bin/$(ARCH32)/uptime mode=4555
file path=usr/bin/$(ARCH64)/amt mode=0555
file path=usr/bin/$(ARCH64)/crle mode=0555
file path=usr/bin/$(ARCH64)/decrypt mode=0555
file path=usr/bin/$(ARCH64)/digest mode=0555
file path=usr/bin/$(ARCH64)/ksh93 mode=0555
file path=usr/bin/$(ARCH64)/ls mode=0555
file path=usr/bin/$(ARCH64)/moe mode=0555
file path=usr/bin/$(ARCH64)/newtask group=sys mode=4555
file path=usr/bin/$(ARCH64)/nohup mode=0555
file path=usr/bin/$(ARCH64)/prctl mode=0555
file path=usr/bin/$(ARCH64)/prstat mode=0555
file path=usr/bin/$(ARCH64)/ps mode=0555
file path=usr/bin/$(ARCH64)/savecore mode=0555
file path=usr/bin/$(ARCH64)/setuname mode=0555
file path=usr/bin/$(ARCH64)/uptime mode=4555
$(i386_ONLY)file path=usr/bin/addbadsec mode=0555
file path=usr/bin/alias mode=0555
file path=usr/bin/amt mode=0555
file path=usr/bin/arch mode=0555
file path=usr/bin/at group=sys mode=4755
file path=usr/bin/atq group=sys mode=4755
file path=usr/bin/atrm group=sys mode=4755
file path=usr/bin/auths mode=0555
file path=usr/bin/basename mode=0555
file path=usr/bin/busstat mode=0555
file path=usr/bin/captoinfo mode=0555
file path=usr/bin/cat mode=0555
file path=usr/bin/chgrp mode=0555
file path=usr/bin/chmod mode=0555
file path=usr/bin/chown mode=0555
file path=usr/bin/ckdate mode=0555
file path=usr/bin/ckgid mode=0555
file path=usr/bin/ckint mode=0555
file path=usr/bin/ckitem mode=0555
file path=usr/bin/ckkeywd mode=0555
file path=usr/bin/ckpath mode=0555
file path=usr/bin/ckrange mode=0555
```

```
file path=usr/bin/ckstr mode=0555
file path=usr/bin/cktime mode=0555
file path=usr/bin/ckuid mode=0555
file path=usr/bin/ckyorn mode=0555
file path=usr/bin/clear mode=0555
file path=usr/bin/coreadm mode=0555
file path=usr/bin/cp mode=0555
file path=usr/bin/cpio mode=0555
file path=usr/bin/crle mode=0555
file path=usr/bin/crontab mode=4555
file path=usr/bin/crypt mode=0555
file path=usr/bin/csh mode=0555
file path=usr/bin/ctrun mode=0555
file path=usr/bin/ctstat mode=0555
file path=usr/bin/ctwatch mode=0555
file path=usr/bin/date mode=0555
file path=usr/bin/dd mode=0555
file path=usr/bin/devattr mode=0555
file path=usr/bin/devfree mode=0555
file path=usr/bin/devreserv mode=0555
file path=usr/bin/dirname mode=0555
$(i386_ONLY)file path=usr/bin/diskscan mode=0555
file path=usr/bin/domainname mode=0555
file path=usr/bin/du mode=0555
file path=usr/bin/dumpcs mode=0555
file path=usr/bin/dumpkeys mode=0555
file path=usr/bin/echo mode=0555
file path=usr/bin/ed mode=0555
file path=usr/bin/egrep mode=0555
file path=usr/bin/eject mode=0555
file path=usr/bin/env mode=0555
file path=usr/bin/expr mode=0555
file path=usr/bin/false mode=0555
file path=usr/bin/fdetach mode=0555
file path=usr/bin/fdformat mode=4555
file path=usr/bin/fgrep mode=0555
file path=usr/bin/file mode=0555
file path=usr/bin/find mode=0555
file path=usr/bin/fmt mode=0555
file path=usr/bin/fmtmsg mode=0555
file path=usr/bin/fold mode=0555
file path=usr/bin/fsstat mode=0555
file path=usr/bin/geniconvtbl mode=0555
file path=usr/bin/getconf mode=0555
file path=usr/bin/getdev mode=0555
file path=usr/bin/getdgrp mode=0555
file path=usr/bin/getent mode=0555
file path=usr/bin/getfacl mode=0555
file path=usr/bin/getopt mode=0555
file path=usr/bin/gettext mode=0555
file path=usr/bin/getvol mode=0555
file path=usr/bin/grep mode=0555
file path=usr/bin/groups mode=0555
file path=usr/bin/head mode=0555
file path=usr/bin/hostid mode=0555
file path=usr/bin/hostname mode=0555
file path=usr/bin/i286 mode=0555
file path=usr/bin/iconv mode=0555
file path=usr/bin/id mode=0555
file path=usr/bin/infocmp mode=0555
file path=usr/bin/iostat mode=0555
file path=usr/bin/isainfo mode=0555
file path=usr/bin/isalist mode=0555
file path=usr/bin/kbd mode=0555
file path=usr/bin/keylogin mode=0555
file path=usr/bin/keylogout mode=0555
file path=usr/bin/kmfcfg mode=0555
file path=usr/bin/kvmstat mode=0555
file path=usr/bin/line mode=0555
file path=usr/bin/listdgrp mode=0555
file path=usr/bin/listusers mode=0555
file path=usr/bin/loadkeys mode=0555
file path=usr/bin/logger mode=0555
file path=usr/bin/login mode=4555
file path=usr/bin/logins mode=0750
file path=usr/bin/ls mode=0555
file path=usr/bin/m4 mode=0555
file path=usr/bin/mach mode=0555
file path=usr/bin/mail group=mail mode=2511
file path=usr/bin/mailx group=mail mode=2511
file path=usr/bin/makedev mode=0555
file path=usr/bin/mesg mode=0555
file path=usr/bin/mkdir mode=0555
file path=usr/bin/mkpwdict mode=0555
file path=usr/bin/mktemp mode=0555
file path=usr/bin/moe mode=0555
file path=usr/bin/more mode=0555
file path=usr/bin/mpstat mode=0555
file path=usr/bin/mt mode=0555
file path=usr/bin/netstat mode=0555
file path=usr/bin/newgrp group=sys mode=4755
file path=usr/bin/nice mode=0555
file path=usr/bin/optisa mode=0555
file path=usr/bin/pagesize mode=0555
file path=usr/bin/passwd group=sys mode=6555
file path=usr/bin/pathchk mode=0555
file path=usr/bin/pax mode=0555
file path=usr/bin/pfexec mode=0555
file path=usr/bin/pg mode=0555
file path=usr/bin/pgrep mode=0555
file path=usr/bin/pktool mode=0555
file path=usr/bin/pr mode=0555
file path=usr/bin/printf mode=0555
file path=usr/bin/priocntl mode=0555
file path=usr/bin/profiles mode=0555
file path=usr/bin/projects mode=0555
file path=usr/bin/putdev mode=0555
file path=usr/bin/putdgrp mode=0555
file path=usr/bin/pwd mode=0555
file path=usr/bin/renice mode=0555
file path=usr/bin/rm mode=0555
file path=usr/bin/rmdir mode=0555
file path=usr/bin/roles mode=0555
file path=usr/bin/rpcinfo mode=0555
file path=usr/bin/runat mode=0555
file path=usr/bin/script mode=0555
file path=usr/bin/sed mode=0555
file path=usr/bin/setfacl mode=0555
file path=usr/bin/setpgrp group=sys mode=0555
file path=usr/bin/settime mode=0555
file path=usr/bin/shcomp mode=0555
file path=usr/bin/strchg group=root mode=0555
file path=usr/bin/strconf group=root mode=0555
file path=usr/bin/stty mode=0555
file path=usr/bin/su group=sys mode=4555
file path=usr/bin/svcprop mode=0555
file path=usr/bin/svcs mode=0555
file path=usr/bin/tabs mode=0555
file path=usr/bin/tail mode=0555
file path=usr/bin/tic mode=0555
file path=usr/bin/time mode=0555
file path=usr/bin/tip mode=4511 owner=uucp
```

```
file path=usr/bin/tpmadm mode=0555
file path=usr/bin/tput mode=0555
file path=usr/bin/tr mode=0555
file path=usr/bin/true mode=0555
file path=usr/bin/tty mode=0555
file path=usr/bin/tzselect mode=0555
file path=usr/bin/userattr mode=0555
file path=usr/bin/vmstat mode=0555
file path=usr/bin/which mode=0555
file path=usr/bin/who mode=0555
file path=usr/bin/wracct mode=0555
file path=usr/bin/write group=tty mode=2555
file path=usr/bin/xargs mode=0555
file path=usr/bin/xstr mode=0555
file path=usr/has/bin/edit mode=0555
file path=usr/has/bin/sh mode=0555
file path=usr/has/man/man1has/edit.1has
file path=usr/has/man/man1has/ex.1has
file path=usr/has/man/man1has/sh.1has
file path=usr/has/man/man1has/vi.1has
file path=usr/kernel/drv/$(ARCH64)/dump group=sys
file path=usr/kernel/drv/$(ARCH64)/fssnap group=sys
file path=usr/kernel/drv/$(ARCH64)/kstat group=sys
file path=usr/kernel/drv/$(ARCH64)/ksyms group=sys
file path=usr/kernel/drv/$(ARCH64)/logindmux group=sys
file path=usr/kernel/drv/$(ARCH64)/ptm group=sys
file path=usr/kernel/drv/$(ARCH64)/pts group=sys
$(i386_ONLY)file path=usr/kernel/drv/dump group=sys
file path=usr/kernel/drv/dump.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/fssnap group=sys
file path=usr/kernel/drv/fssnap.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/kstat group=sys
file path=usr/kernel/drv/kstat.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/ksyms group=sys
file path=usr/kernel/drv/ksyms.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/logindmux group=sys
file path=usr/kernel/drv/logindmux.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/ptm group=sys
file path=usr/kernel/drv/ptm.conf group=sys
$(i386_ONLY)file path=usr/kernel/drv/pts group=sys
file path=usr/kernel/drv/pts.conf group=sys
file path=usr/kernel/exec/$(ARCH64)/javaexec group=sys mode=0755
file path=usr/kernel/exec/$(ARCH64)/shbinexec group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/exec/javaexec group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/exec/shbinexec group=sys mode=0755
file path=usr/kernel/fs/$(ARCH64)/fdfs group=sys mode=0755
file path=usr/kernel/fs/$(ARCH64)/pcfs group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/fs/fdfs group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/fs/pcfs group=sys mode=0755
file path=usr/kernel/sched/$(ARCH64)/FX group=sys mode=0755
file path=usr/kernel/sched/$(ARCH64)/FX_DPTBL group=sys mode=0755
file path=usr/kernel/sched/$(ARCH64)/IA group=sys mode=0755
file path=usr/kernel/sched/$(ARCH64)/RT group=sys mode=0755
file path=usr/kernel/sched/$(ARCH64)/RT_DPTBL group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sched/FX group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sched/FX_DPTBL group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sched/IA group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sched/RT group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sched/RT_DPTBL group=sys mode=0755
file path=usr/kernel/strmod/$(ARCH64)/cryptmod group=sys mode=0755
file path=usr/kernel/strmod/$(ARCH64)/rlmod group=sys mode=0755
file path=usr/kernel/strmod/$(ARCH64)/telmod group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/strmod/cryptmod group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/strmod/rlmod group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/strmod/telmod group=sys mode=0755
file path=usr/kernel/sys/$(ARCH64)/acctctl group=sys mode=0755
file path=usr/kernel/sys/$(ARCH64)/exacctsys group=sys mode=0755
file path=usr/kernel/sys/$(ARCH64)/sysacct group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sys/acctctl group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sys/exacctsys group=sys mode=0755
$(i386_ONLY)file path=usr/kernel/sys/sysacct group=sys mode=0755
file path=usr/kvm/README group=sys
file path=usr/lib/$(ARCH64)/libshare.so.1
file path=usr/lib/audit/audit_record_attr mode=0444
file path=usr/lib/calprog mode=0555
file path=usr/lib/class/FX/FXdispadmin mode=0555
file path=usr/lib/class/FX/FXpriocntl mode=0555
file path=usr/lib/class/IA/IAdispadmin mode=0555
file path=usr/lib/class/IA/IApriocntl mode=0555
file path=usr/lib/class/RT/RTdispadmin mode=0555
file path=usr/lib/class/RT/RTpriocntl mode=0555
file path=usr/lib/class/SDC/SDCdispadmin mode=0555
file path=usr/lib/class/SDC/SDCpriocntl mode=0555
file path=usr/lib/class/TS/TSdispadmin mode=0555
file path=usr/lib/class/TS/TSpriocntl mode=0555
file path=usr/lib/devfsadm/linkmod/SUNW_audio_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_cfg_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_disk_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_fssnap_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_ieee1394_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_lofi_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_md_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_misc_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_misc_link_$(ARCH).so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_port_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_ramdisk_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_sgen_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_smp_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_tape_link.so group=sys
file path=usr/lib/devfsadm/linkmod/SUNW_usb_link.so group=sys
$(i386_ONLY)file path=usr/lib/devfsadm/linkmod/SUNW_xen_link.so group=sys
file path=usr/lib/diffh mode=0555
file path=usr/lib/expreserve mode=0555
file path=usr/lib/exrecover mode=0555
file path=usr/lib/fs/cachefs/cachefsd mode=0555
file path=usr/lib/fs/cachefs/cachefslog mode=0555
file path=usr/lib/fs/cachefs/cachefspack mode=0555
file path=usr/lib/fs/cachefs/cachefsstat mode=0555
file path=usr/lib/fs/cachefs/cachefswssize mode=0555
file path=usr/lib/fs/cachefs/cfsadmin mode=0555
file path=usr/lib/fs/cachefs/cfsfstype mode=0555
file path=usr/lib/fs/cachefs/cfstagchk mode=0555
file path=usr/lib/fs/cachefs/dfshares mode=0555
file path=usr/lib/fs/cachefs/fsck mode=0555
file path=usr/lib/fs/cachefs/mount mode=0555
file path=usr/lib/fs/cachefs/share mode=0555
file path=usr/lib/fs/cachefs/umount mode=0555
file path=usr/lib/fs/cachefs/unshare mode=0555
file path=usr/lib/fs/ctfs/mount mode=0555
file path=usr/lib/fs/fd/mount mode=0555
file path=usr/lib/fs/hsfs/fstyp.so.1 mode=0555
file path=usr/lib/fs/hsfs/labelit mode=0555
file path=usr/lib/fs/lofs/mount mode=0555
file path=usr/lib/fs/mntfs/mount mode=0555
file path=usr/lib/fs/objfs/mount mode=0555
file path=usr/lib/fs/proc/mount mode=0555
file path=usr/lib/fs/sharefs/mount mode=0555
file path=usr/lib/fs/tmpfs/mount mode=0555
file path=usr/lib/fs/ufs/clri mode=0555
file path=usr/lib/fs/ufs/df mode=0555
file path=usr/lib/fs/ufs/edquota mode=0555
file path=usr/lib/fs/ufs/ff mode=0555
```

```
file path=usr/lib/fs/ufs/fsck mode=0555
file path=usr/lib/fs/ufs/fsckall mode=0555
file path=usr/lib/fs/ufs/fsdb mode=0555
file path=usr/lib/fs/ufs/fsirand mode=0555
file path=usr/lib/fs/ufs/fssnap mode=0555
file path=usr/lib/fs/ufs/fstyp.so.1 mode=0555
file path=usr/lib/fs/ufs/labelit mode=0555
file path=usr/lib/fs/ufs/lockfs mode=0555
file path=usr/lib/fs/ufs/mkfs mode=0555
file path=usr/lib/fs/ufs/ncheck mode=0555
file path=usr/lib/fs/ufs/newfs mode=0555
file path=usr/lib/fs/ufs/quot mode=0555
file path=usr/lib/fs/ufs/quota mode=4555
file path=usr/lib/fs/ufs/quotacheck mode=0555
file path=usr/lib/fs/ufs/quotaoff mode=0555
file path=usr/lib/fs/ufs/repquota mode=0555
file path=usr/lib/fs/ufs/tunefs mode=0555
file path=usr/lib/fs/ufs/ufsdump mode=4555
file path=usr/lib/fs/ufs/ufsrestore mode=4555
file path=usr/lib/fs/ufs/volcopy mode=0555
file path=usr/lib/getoptcvt mode=0555
file path=usr/lib/help/auths/locale/C/AllSolAuthsHeader.html
file path=usr/lib/help/auths/locale/C/AuditHeader.html
file path=usr/lib/help/auths/locale/C/AuthJobsAdmin.html
file path=usr/lib/help/auths/locale/C/AuthJobsUser.html
file path=usr/lib/help/auths/locale/C/AuthProfmgrAssign.html
file path=usr/lib/help/auths/locale/C/AuthProfmgrDelegate.html
file path=usr/lib/help/auths/locale/C/AuthProfmgrExecattrWrite.html
file path=usr/lib/help/auths/locale/C/AuthProfmgrRead.html
file path=usr/lib/help/auths/locale/C/AuthProfmgrWrite.html
file path=usr/lib/help/auths/locale/C/AuthReadNDMP.html
file path=usr/lib/help/auths/locale/C/AuthReadSMB.html
file path=usr/lib/help/auths/locale/C/AuthRoleAssign.html
file path=usr/lib/help/auths/locale/C/AuthRoleDelegate.html
file path=usr/lib/help/auths/locale/C/AuthRoleWrite.html
file path=usr/lib/help/auths/locale/C/BindStates.html
file path=usr/lib/help/auths/locale/C/DevAllocHeader.html
file path=usr/lib/help/auths/locale/C/DevAllocate.html
file path=usr/lib/help/auths/locale/C/DevConfig.html
file path=usr/lib/help/auths/locale/C/DevGrant.html
file path=usr/lib/help/auths/locale/C/DevRevoke.html
file path=usr/lib/help/auths/locale/C/DhcpmgrHeader.html
file path=usr/lib/help/auths/locale/C/DhcpmgrWrite.html
file path=usr/lib/help/auths/locale/C/HotplugHeader.html
file path=usr/lib/help/auths/locale/C/HotplugModify.html
file path=usr/lib/help/auths/locale/C/IdmapRules.html
file path=usr/lib/help/auths/locale/C/JobHeader.html
file path=usr/lib/help/auths/locale/C/JobsGrant.html
file path=usr/lib/help/auths/locale/C/LinkSecurity.html
file path=usr/lib/help/auths/locale/C/LoginEnable.html
file path=usr/lib/help/auths/locale/C/LoginHeader.html
file path=usr/lib/help/auths/locale/C/LoginRemote.html
file path=usr/lib/help/auths/locale/C/NetworkAutoconfRead.html
file path=usr/lib/help/auths/locale/C/NetworkAutoconfSelect.html
file path=usr/lib/help/auths/locale/C/NetworkAutoconfWlan.html
file path=usr/lib/help/auths/locale/C/NetworkAutoconfWrite.html
file path=usr/lib/help/auths/locale/C/NetworkHeader.html
file path=usr/lib/help/auths/locale/C/NetworkILBconf.html
file path=usr/lib/help/auths/locale/C/NetworkILBenable.html
file path=usr/lib/help/auths/locale/C/NetworkInterfaceConfig.html
file path=usr/lib/help/auths/locale/C/NetworkVRRP.html
file path=usr/lib/help/auths/locale/C/PriAdmin.html
file path=usr/lib/help/auths/locale/C/ProfmgrHeader.html
file path=usr/lib/help/auths/locale/C/RoleHeader.html
file path=usr/lib/help/auths/locale/C/SmfAllocate.html
file path=usr/lib/help/auths/locale/C/SmfAutofsStates.html
file path=usr/lib/help/auths/locale/C/SmfCoreadmStates.html
file path=usr/lib/help/auths/locale/C/SmfCronStates.html
file path=usr/lib/help/auths/locale/C/SmfExAcctFlowStates.html
file path=usr/lib/help/auths/locale/C/SmfExAcctNetStates.html
file path=usr/lib/help/auths/locale/C/SmfExAcctProcessStates.html
file path=usr/lib/help/auths/locale/C/SmfExAcctTaskStates.html
file path=usr/lib/help/auths/locale/C/SmfHeader.html
file path=usr/lib/help/auths/locale/C/SmfILBStates.html
file path=usr/lib/help/auths/locale/C/SmfIPsecStates.html
file path=usr/lib/help/auths/locale/C/SmfIdmapStates.html
file path=usr/lib/help/auths/locale/C/SmfInetdStates.html
file path=usr/lib/help/auths/locale/C/SmfLocationStates.html
file path=usr/lib/help/auths/locale/C/SmfMDNSStates.html
file path=usr/lib/help/auths/locale/C/SmfManageAudit.html
file path=usr/lib/help/auths/locale/C/SmfManageHeader.html
file path=usr/lib/help/auths/locale/C/SmfManageHotplug.html
file path=usr/lib/help/auths/locale/C/SmfManageZFSSnap.html
file path=usr/lib/help/auths/locale/C/SmfModifyAppl.html
file path=usr/lib/help/auths/locale/C/SmfModifyDepend.html
file path=usr/lib/help/auths/locale/C/SmfModifyFramework.html
file path=usr/lib/help/auths/locale/C/SmfModifyHeader.html
file path=usr/lib/help/auths/locale/C/SmfModifyMethod.html
file path=usr/lib/help/auths/locale/C/SmfNADDStates.html
file path=usr/lib/help/auths/locale/C/SmfNDMPStates.html
file path=usr/lib/help/auths/locale/C/SmfNWAMStates.html
file path=usr/lib/help/auths/locale/C/SmfNscdStates.html
file path=usr/lib/help/auths/locale/C/SmfPowerStates.html
file path=usr/lib/help/auths/locale/C/SmfReparseStates.html
file path=usr/lib/help/auths/locale/C/SmfRoutingStates.html
file path=usr/lib/help/auths/locale/C/SmfSMBFSStates.html
file path=usr/lib/help/auths/locale/C/SmfSMBStates.html
file path=usr/lib/help/auths/locale/C/SmfSendmailStates.html
file path=usr/lib/help/auths/locale/C/SmfSshStates.html
file path=usr/lib/help/auths/locale/C/SmfSyslogStates.html
file path=usr/lib/help/auths/locale/C/SmfVRRPStates.html
file path=usr/lib/help/auths/locale/C/SmfValueAudit.html
file path=usr/lib/help/auths/locale/C/SmfValueCoreadm.html
file path=usr/lib/help/auths/locale/C/SmfValueExAcctFlow.html
file path=usr/lib/help/auths/locale/C/SmfValueExAcctNet.html
file path=usr/lib/help/auths/locale/C/SmfValueExAcctProcess.html
file path=usr/lib/help/auths/locale/C/SmfValueExAcctTask.html
file path=usr/lib/help/auths/locale/C/SmfValueFirewall.html
file path=usr/lib/help/auths/locale/C/SmfValueHeader.html
file path=usr/lib/help/auths/locale/C/SmfValueIPsec.html
file path=usr/lib/help/auths/locale/C/SmfValueIdmap.html
file path=usr/lib/help/auths/locale/C/SmfValueInetd.html
file path=usr/lib/help/auths/locale/C/SmfValueMDNS.html
file path=usr/lib/help/auths/locale/C/SmfValueNADD.html
file path=usr/lib/help/auths/locale/C/SmfValueNDMP.html
file path=usr/lib/help/auths/locale/C/SmfValueNWAM.html
file path=usr/lib/help/auths/locale/C/SmfValueRouting.html
file path=usr/lib/help/auths/locale/C/SmfValueSMB.html
file path=usr/lib/help/auths/locale/C/SmfValueVscan.html
file path=usr/lib/help/auths/locale/C/SmfValueVt.html
file path=usr/lib/help/auths/locale/C/SmfVscanStates.html
file path=usr/lib/help/auths/locale/C/SmfVtStates.html
file path=usr/lib/help/auths/locale/C/SmfWpaStates.html
file path=usr/lib/help/auths/locale/C/SysCpuPowerMgmt.html
file path=usr/lib/help/auths/locale/C/SysDate.html
file path=usr/lib/help/auths/locale/C/SysHeader.html
file path=usr/lib/help/auths/locale/C/SysMaintenance.html
file path=usr/lib/help/auths/locale/C/SysPowerMgmtBrightness.html
file path=usr/lib/help/auths/locale/C/SysPowerMgmtHeader.html
file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspend.html
file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspendtoDisk.html
file path=usr/lib/help/auths/locale/C/SysPowerMgmtSuspendtoRAM.html
```

```
file path=usr/lib/help/auths/locale/C/SysShutdown.html
file path=usr/lib/help/auths/locale/C/SysSyseventRead.html
file path=usr/lib/help/auths/locale/C/SysSyseventWrite.html
file path=usr/lib/help/auths/locale/C/WifiConfig.html
file path=usr/lib/help/auths/locale/C/WifiWep.html
file path=usr/lib/help/auths/locale/C/ZoneCloneFrom.html
file path=usr/lib/help/auths/locale/C/ZoneHeader.html
file path=usr/lib/help/auths/locale/C/ZoneLogin.html
file path=usr/lib/help/auths/locale/C/ZoneManage.html
file path=usr/lib/help/profiles/locale/C/RtAcctadm.html
file path=usr/lib/help/profiles/locale/C/RtAll.html
file path=usr/lib/help/profiles/locale/C/RtAuditCfg.html
file path=usr/lib/help/profiles/locale/C/RtAuditCtrl.html
file path=usr/lib/help/profiles/locale/C/RtAuditReview.html
file path=usr/lib/help/profiles/locale/C/RtCPUPowerManagement.html
file path=usr/lib/help/profiles/locale/C/RtConsUser.html
file path=usr/lib/help/profiles/locale/C/RtContractObserver.html
file path=usr/lib/help/profiles/locale/C/RtCronMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtCryptoMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtDHCPMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtDatAdmin.html
file path=usr/lib/help/profiles/locale/C/RtDefault.html
file path=usr/lib/help/profiles/locale/C/RtDeviceMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtDeviceSecurity.html
file path=usr/lib/help/profiles/locale/C/RtExAcctFlow.html
file path=usr/lib/help/profiles/locale/C/RtExAcctNet.html
file path=usr/lib/help/profiles/locale/C/RtExAcctProcess.html
file path=usr/lib/help/profiles/locale/C/RtExAcctTask.html
file path=usr/lib/help/profiles/locale/C/RtFTPMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtFileSysMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtFileSysSecurity.html
file path=usr/lib/help/profiles/locale/C/RtHotplugMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtIPFilterMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtIdmapMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtIdmapNameRulesMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtInetdMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtKerberosClntMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtKerberosSrvrMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtLogMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtMailMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtMaintAndRepair.html
file path=usr/lib/help/profiles/locale/C/RtMediaBkup.html
file path=usr/lib/help/profiles/locale/C/RtMediaCtlg.html
file path=usr/lib/help/profiles/locale/C/RtMediaRestore.html
file path=usr/lib/help/profiles/locale/C/RtNDMPMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtNameServiceAdmin.html
file path=usr/lib/help/profiles/locale/C/RtNameServiceSecure.html
file path=usr/lib/help/profiles/locale/C/RtNetAutoconfAdmin.html
file path=usr/lib/help/profiles/locale/C/RtNetAutoconfUser.html
file path=usr/lib/help/profiles/locale/C/RtNetILB.html
file path=usr/lib/help/profiles/locale/C/RtNetIPsec.html
file path=usr/lib/help/profiles/locale/C/RtNetLinkSecure.html
file path=usr/lib/help/profiles/locale/C/RtNetMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtNetObservability.html
file path=usr/lib/help/profiles/locale/C/RtNetSecure.html
file path=usr/lib/help/profiles/locale/C/RtNetVRRP.html
file path=usr/lib/help/profiles/locale/C/RtNetWifiMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtNetWifiSecure.html
file path=usr/lib/help/profiles/locale/C/RtObAccessMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtOperator.html
file path=usr/lib/help/profiles/locale/C/RtPriAdmin.html
file path=usr/lib/help/profiles/locale/C/RtPrntAdmin.html
file path=usr/lib/help/profiles/locale/C/RtProcManagement.html
file path=usr/lib/help/profiles/locale/C/RtReparseMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtReservedProfile.html
file path=usr/lib/help/profiles/locale/C/RtRightsDelegate.html
file path=usr/lib/help/profiles/locale/C/RtSMBFSMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtSMBMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtSoftwareInstall.html
file path=usr/lib/help/profiles/locale/C/RtSysAdmin.html
file path=usr/lib/help/profiles/locale/C/RtSysEvMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmt.html
file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtBrightness.html
file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspend.html
file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspendtoDisk.html
file path=usr/lib/help/profiles/locale/C/RtSysPowerMgmtSuspendtoRAM.html
file path=usr/lib/help/profiles/locale/C/RtUserMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtUserSecurity.html
file path=usr/lib/help/profiles/locale/C/RtVscanMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtZFSFileSysMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtZFSStorageMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtZoneMngmnt.html
file path=usr/lib/help/profiles/locale/C/RtZoneSecurity.html
file path=usr/lib/hotplugd mode=0555
file path=usr/lib/iconv/646da.8859.t mode=0444
file path=usr/lib/iconv/646de.8859.t mode=0444
file path=usr/lib/iconv/646en.8859.t mode=0444
file path=usr/lib/iconv/646es.8859.t mode=0444
file path=usr/lib/iconv/646fr.8859.t mode=0444
file path=usr/lib/iconv/646it.8859.t mode=0444
file path=usr/lib/iconv/646sv.8859.t mode=0444
file path=usr/lib/iconv/8859.646.t mode=0444
file path=usr/lib/iconv/8859.646da.t mode=0444
file path=usr/lib/iconv/8859.646de.t mode=0444
file path=usr/lib/iconv/8859.646en.t mode=0444
file path=usr/lib/iconv/8859.646es.t mode=0444
file path=usr/lib/iconv/8859.646fr.t mode=0444
file path=usr/lib/iconv/8859.646it.t mode=0444
file path=usr/lib/iconv/8859.646sv.t mode=0444
file path=usr/lib/iconv/iconv_data mode=0444
file path=usr/lib/idmapd mode=0555
file path=usr/lib/inet/$(ARCH32)/in.iked mode=0555
file path=usr/lib/inet/$(ARCH64)/in.iked mode=0555
file path=usr/lib/inet/certdb mode=0555
file path=usr/lib/inet/certlocal mode=0555
file path=usr/lib/inet/certrldb mode=0555
file path=usr/lib/inet/inetd mode=0555
file path=usr/lib/intrd mode=0555
file path=usr/lib/isaexec mode=0555
file path=usr/lib/kssladm mode=0555
$(sparc_ONLY)file path=usr/lib/ld.so
file path=usr/lib/libshare.so.1
file path=usr/lib/makekey mode=0555
file path=usr/lib/more.help
file path=usr/lib/newsyslog group=sys mode=0555
file path=usr/lib/passmgmt group=sys mode=0555
file path=usr/lib/pci/pcidr mode=0555
file path=usr/lib/pci/pcidr_plugin.so
file path=usr/lib/pfexecd mode=0555
file path=usr/lib/platexec mode=0555
file path=usr/lib/rcm/modules/SUNW_aggr_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_cluster_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_dump_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_filesys_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_ibpart_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_ip_anon_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_ip_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_mpxio_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_network_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_swap_rcm.so mode=0555
$(sparc_ONLY)file path=usr/lib/rcm/modules/SUNW_ttymux_rcm.so mode=0555
file path=usr/lib/rcm/modules/SUNW_vlan_rcm.so mode=0555
```

```
file path=usr/lib/rcm/modules/SUNW_vnic_rcm.so mode=0555
file path=usr/lib/rcm/rcm_daemon mode=0555
file path=usr/lib/reparse/reparsed group=sys mode=0555
file path=usr/lib/saf/listen group=sys mode=0755
file path=usr/lib/saf/nlps_server group=sys mode=0755
file path=usr/lib/saf/sac group=sys mode=0555
file path=usr/lib/saf/ttymon group=sys mode=0555
file path=usr/lib/sysevent/modules/datalink_mod.so
file path=usr/lib/sysevent/modules/devfsadmd_mod.so
file path=usr/lib/sysevent/modules/sysevent_conf_mod.so
file path=usr/lib/sysevent/modules/sysevent_reg_mod.so
file path=usr/lib/sysevent/syseventconfd mode=0555
file path=usr/lib/sysevent/syseventd mode=0555
file path=usr/lib/utmp_update mode=4555
file path=usr/lib/utmpd mode=0555
file path=usr/lib/vtdaemon mode=0555
file path=usr/lib/vtinfo mode=0555
file path=usr/lib/vtxlock mode=0555
file path=usr/sadm/bin/puttext mode=0555
file path=usr/sadm/install/miniroot.db group=sys mode=0444
file path=usr/sadm/install/scripts/i.ipsecalgs group=sys mode=0555
file path=usr/sadm/install/scripts/i.kcfconf group=sys mode=0555
file path=usr/sadm/install/scripts/i.kmfconf group=sys mode=0555
file path=usr/sadm/install/scripts/i.manifest group=sys mode=0555
file path=usr/sadm/install/scripts/i.pkcs11conf group=sys mode=0555
file path=usr/sadm/install/scripts/i.rbac group=sys mode=0555
file path=usr/sadm/install/scripts/r.ipsecalgs group=sys mode=0555
file path=usr/sadm/install/scripts/r.kcfconf group=sys mode=0555
file path=usr/sadm/install/scripts/r.kmfconf group=sys mode=0555
file path=usr/sadm/install/scripts/r.manifest group=sys mode=0555
file path=usr/sadm/install/scripts/r.pkcs11conf group=sys mode=0555
file path=usr/sadm/install/scripts/r.rbac group=sys mode=0555
file path=usr/sadm/ugdates mode=0444
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/add_drv group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/modinfo group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/modload group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/modunload group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/pbind group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/prtconf group=sys mode=2555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/psrset group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/rem_drv group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/swap group=sys mode=2555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/sysdef group=sys mode=2555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/update_drv group=sys mode=0555
$(i386_ONLY)file path=usr/sbin/$(ARCH32)/whodo mode=4555
file path=usr/sbin/$(ARCH64)/add_drv group=sys mode=0555
file path=usr/sbin/$(ARCH64)/modinfo group=sys mode=0555
file path=usr/sbin/$(ARCH64)/modload group=sys mode=0555
file path=usr/sbin/$(ARCH64)/modunload group=sys mode=0555
file path=usr/sbin/$(ARCH64)/pbind group=sys mode=0555
file path=usr/sbin/$(ARCH64)/prtconf group=sys mode=2555
file path=usr/sbin/$(ARCH64)/psrset group=sys mode=0555
file path=usr/sbin/$(ARCH64)/rem_drv group=sys mode=0555
file path=usr/sbin/$(ARCH64)/swap group=sys mode=2555
file path=usr/sbin/$(ARCH64)/sysdef group=sys mode=2555
file path=usr/sbin/$(ARCH64)/update_drv group=sys mode=0555
file path=usr/sbin/$(ARCH64)/whodo mode=4555
file path=usr/sbin/6to4relay mode=0555
file path=usr/sbin/acctadm mode=0555
file path=usr/sbin/allocate mode=4555
file path=usr/sbin/arp mode=0555
file path=usr/sbin/audit mode=0555
file path=usr/sbin/auditconfig mode=0555
file path=usr/sbin/auditd mode=0555
file path=usr/sbin/auditrecord mode=0555
file path=usr/sbin/auditreduce mode=0555
file path=usr/sbin/auditstat mode=0555
file path=usr/sbin/cfgadm mode=0555
file path=usr/sbin/chroot mode=0555
file path=usr/sbin/clear_locks mode=0555
file path=usr/sbin/clinfo mode=0555
file path=usr/sbin/clri mode=0555
file path=usr/sbin/consadm group=sys mode=0555
file path=usr/sbin/cron group=sys mode=0555
file path=usr/sbin/devfsadm group=sys mode=0755
file path=usr/sbin/devinfo mode=0555
file path=usr/sbin/df mode=0555
file path=usr/sbin/dfmounts mode=0555
file path=usr/sbin/dispadmin mode=0555
file path=usr/sbin/dminfo mode=0555
file path=usr/sbin/dumpadm mode=0555
file path=usr/sbin/eeprom group=sys mode=2555
file path=usr/sbin/ff mode=0555
file path=usr/sbin/fmthard group=sys mode=0555
file path=usr/sbin/format mode=0555
file path=usr/sbin/fsck mode=0555
file path=usr/sbin/fstyp group=sys mode=0555
file path=usr/sbin/fuser mode=0555
file path=usr/sbin/getdevpolicy group=sys mode=0555
file path=usr/sbin/getmajor group=sys mode=0755
file path=usr/sbin/groupadd group=sys mode=0555
file path=usr/sbin/groupdel group=sys mode=0555
file path=usr/sbin/groupmod group=sys mode=0555
file path=usr/sbin/grpck mode=0555
file path=usr/sbin/halt mode=0755
file path=usr/sbin/hotplug mode=0555
file path=usr/sbin/idmap mode=0555
file path=usr/sbin/if_mpadm mode=0555
file path=usr/sbin/ikeadm mode=0555
file path=usr/sbin/ikecert mode=0555
file path=usr/sbin/inetadm mode=0555
file path=usr/sbin/inetconv mode=0555
file path=usr/sbin/install mode=0555
file path=usr/sbin/installboot group=sys mode=0555
file path=usr/sbin/ipaddrsel mode=0555
file path=usr/sbin/ipsecalgs mode=0555
file path=usr/sbin/ipsecconf mode=0555
file path=usr/sbin/ipseckey mode=0555
file path=usr/sbin/keyserv group=sys mode=0555
file path=usr/sbin/killall mode=0555
file path=usr/sbin/ksslcfg mode=0555
file path=usr/sbin/link mode=0555
file path=usr/sbin/locator mode=0555
file path=usr/sbin/lofiadm mode=0555
file path=usr/sbin/logadm mode=0555
file path=usr/sbin/makedbm mode=0555
file path=usr/sbin/mkdevalloc mode=0555
file path=usr/sbin/mkfile mode=0555
file path=usr/sbin/mknod mode=0555
file path=usr/sbin/mountall group=sys mode=0555
file path=usr/sbin/msgid mode=0555
file path=usr/sbin/mvdir mode=0555
file path=usr/sbin/ndd mode=0555
file path=usr/sbin/nlsadmin group=adm mode=0755
file path=usr/sbin/nscd mode=0555
file path=usr/sbin/nwamadm mode=0555
file path=usr/sbin/nwamcfg mode=0555
file path=usr/sbin/pmadm group=sys mode=0555
file path=usr/sbin/praudit mode=0555
$(i386_ONLY)file path=usr/sbin/prtdiag group=sys mode=2755
file path=usr/sbin/prtvtoc group=sys mode=0555
file path=usr/sbin/psradm group=sys mode=0555
```

```
file path=usr/sbin/psrinfo group=sys mode=0555
file path=usr/sbin/pwck mode=0555
file path=usr/sbin/pwconv group=sys mode=0555
file path=usr/sbin/raidctl mode=0555
file path=usr/sbin/ramdiskadm mode=0555
file path=usr/sbin/rctladm mode=0555
file path=usr/sbin/root_archive group=sys mode=0555
file path=usr/sbin/rpcbind mode=0555
$(i386_ONLY)file path=usr/sbin/rtc mode=0555
file path=usr/sbin/sacadm group=sys mode=4755
file path=usr/sbin/setmnt mode=0555
file path=usr/sbin/shareall mode=0555
file path=usr/sbin/sharectl mode=0555
file path=usr/sbin/sharemgr mode=0555
file path=usr/sbin/shutdown group=sys mode=0755
file path=usr/sbin/smbios mode=0555
file path=usr/sbin/stmsboot mode=0555
file path=usr/sbin/strace group=sys mode=0555
file path=usr/sbin/strclean group=sys mode=0555
file path=usr/sbin/strerr group=sys mode=0555
file path=usr/sbin/sttydefs group=sys mode=0755
file path=usr/sbin/svcadm mode=0555
file path=usr/sbin/svccfg mode=0555
file path=usr/sbin/syncinit mode=0555
file path=usr/sbin/syncloop mode=0555
file path=usr/sbin/syncstat mode=0555
file path=usr/sbin/syseventadm group=sys mode=0555
file path=usr/sbin/syslogd group=sys mode=0555
file path=usr/sbin/tar mode=0555
file path=usr/sbin/traceroute mode=4555
file path=usr/sbin/trapstat mode=0555
file path=usr/sbin/ttyadm group=sys mode=0755
$(i386_ONLY)file path=usr/sbin/ucodeadm mode=0555
file path=usr/sbin/umountall group=sys mode=0555
file path=usr/sbin/unlink mode=0555
file path=usr/sbin/unshareall mode=0555
file path=usr/sbin/useradd group=sys mode=0555
file path=usr/sbin/userdel group=sys mode=0555
file path=usr/sbin/usermod group=sys mode=0555
$(sparc_ONLY)file path=usr/sbin/virtinfo mode=0555
file path=usr/sbin/volcopy mode=0555
file path=usr/sbin/wall group=tty mode=2555
file path=usr/sbin/zdump mode=0555
file path=usr/sbin/zic mode=0555
file path=usr/share/doc/ksh/COMPATIBILITY
file path=usr/share/doc/ksh/DESIGN
file path=usr/share/doc/ksh/OBSOLETE
file path=usr/share/doc/ksh/README
file path=usr/share/doc/ksh/RELEASE
file path=usr/share/doc/ksh/TYPES
file path=usr/share/doc/ksh/images/callouts/1.png
file path=usr/share/doc/ksh/images/callouts/10.png
file path=usr/share/doc/ksh/images/callouts/2.png
file path=usr/share/doc/ksh/images/callouts/3.png
file path=usr/share/doc/ksh/images/callouts/4.png
file path=usr/share/doc/ksh/images/callouts/5.png
file path=usr/share/doc/ksh/images/callouts/6.png
file path=usr/share/doc/ksh/images/callouts/7.png
file path=usr/share/doc/ksh/images/callouts/8.png
file path=usr/share/doc/ksh/images/callouts/9.png
file path=usr/share/doc/ksh/images/tag_bourne.png
file path=usr/share/doc/ksh/images/tag_i18n.png
file path=usr/share/doc/ksh/images/tag_ksh.png
file path=usr/share/doc/ksh/images/tag_ksh88.png
file path=usr/share/doc/ksh/images/tag_ksh93.png
file path=usr/share/doc/ksh/images/tag_l10n.png
file path=usr/share/doc/ksh/images/tag_perf.png
file path=usr/share/doc/ksh/shell_styleguide.docbook
file path=usr/share/doc/ksh/shell_styleguide.html
file path=usr/share/lib/mailx/mailx.help
file path=usr/share/lib/mailx/mailx.help.~
file path=usr/share/lib/tabset/3101
file path=usr/share/lib/tabset/beehive
file path=usr/share/lib/tabset/hds
file path=usr/share/lib/tabset/hds3
file path=usr/share/lib/tabset/std
file path=usr/share/lib/tabset/stdcrt
file path=usr/share/lib/tabset/teleray
file path=usr/share/lib/tabset/vt100
file path=usr/share/lib/tabset/wyse-adds
file path=usr/share/lib/tabset/xerox1720
file path=usr/share/lib/termcap
file path=usr/share/lib/unittab
file path=usr/share/lib/xml/dtd/adt_record.dtd.1
file path=usr/share/lib/xml/dtd/kmfpolicy.dtd
file path=usr/share/lib/xml/dtd/service_bundle.dtd.1 group=sys
file path=usr/share/lib/xml/style/adt_record.xsl.1
file path=var/adm/aculog mode=0600 owner=uucp preserve=true
file path=var/adm/spellhist mode=0666 preserve=true
file path=var/adm/utmpx preserve=true
file path=var/adm/wtmpx group=adm owner=adm preserve=true
file path=var/log/authlog group=sys mode=0600 preserve=true
file path=var/log/syslog group=sys preserve=true
file path=var/sadm/system/admin/default_java group=sys mode=0444
file path=var/saf/zsmon/log group=sys preserve=true
file path=var/spool/cron/crontabs/adm group=sys mode=0600 preserve=true
file path=var/spool/cron/crontabs/root group=sys mode=0600 preserve=true
hardlink path=etc/rc2.d/S20sysetup target=../../etc/init.d/sysetup
hardlink path=etc/rc2.d/S73cachefs.daemon \
    target=../../etc/init.d/cachefs.daemon
hardlink path=etc/rc2.d/S89PRESERVE target=../../etc/init.d/PRESERVE
$(sparc_ONLY)hardlink path=etc/svc/profile/platform_SUNW,Sun-Fire-V890.xml \
    target=./platform_SUNW,Sun-Fire-880.xml
$(sparc_ONLY)hardlink \
    path=etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-40.xml \
    target=./platform_SUNW,UltraSPARC-IIi-Netract.xml
$(sparc_ONLY)hardlink \
    path=etc/svc/profile/platform_SUNW,UltraSPARC-IIe-NetraCT-60.xml \
    target=./platform_SUNW,UltraSPARC-IIi-Netract.xml
hardlink path=sbin/rc5 target=../sbin/rc0
hardlink path=sbin/rc6 target=../sbin/rc0
hardlink path=usr/bin/$(ARCH32)/encrypt target=decrypt
hardlink path=usr/bin/$(ARCH32)/ksh target=ksh93
hardlink path=usr/bin/$(ARCH32)/mac target=digest
hardlink path=usr/bin/$(ARCH32)/rksh target=ksh93
hardlink path=usr/bin/$(ARCH32)/rksh93 target=ksh93
$(i386_ONLY)hardlink path=usr/bin/$(ARCH32)/w target=uptime
hardlink path=usr/bin/$(ARCH64)/encrypt target=decrypt
hardlink path=usr/bin/$(ARCH64)/ksh target=ksh93
hardlink path=usr/bin/$(ARCH64)/mac target=digest
hardlink path=usr/bin/$(ARCH64)/rksh target=ksh93
hardlink path=usr/bin/$(ARCH64)/rksh93 target=ksh93
hardlink path=usr/bin/$(ARCH64)/w target=uptime
hardlink path=usr/bin/bg target=../../usr/bin/alias
hardlink path=usr/bin/cd target=../../usr/bin/alias
hardlink path=usr/bin/cksum target=../../usr/bin/alias
hardlink path=usr/bin/cmp target=../../usr/bin/alias
hardlink path=usr/bin/comm target=../../usr/bin/alias
hardlink path=usr/bin/command target=../../usr/bin/alias
hardlink path=usr/bin/cut target=../../usr/bin/alias
hardlink path=usr/bin/decrypt target=../../usr/lib/isaexec
hardlink path=usr/bin/digest target=../../usr/lib/isaexec
```

```
hardlink path=usr/bin/dispgid target=../../usr/bin/ckgid
hardlink path=usr/bin/dispuid target=../../usr/bin/ckuid
hardlink path=usr/bin/edit target=../has/bin/edit
hardlink path=usr/bin/encrypt target=../../usr/lib/isaexec
hardlink path=usr/bin/fc target=../../usr/bin/alias
hardlink path=usr/bin/fg target=../../usr/bin/alias
hardlink path=usr/bin/getopts target=../../usr/bin/alias
hardlink path=usr/bin/hash target=../../usr/bin/alias
hardlink path=usr/bin/i386 target=../../usr/bin/i286
hardlink path=usr/bin/i486 target=../../usr/bin/i286
hardlink path=usr/bin/i860 target=../../usr/bin/i286
hardlink path=usr/bin/i86pc target=../../usr/bin/i286
hardlink path=usr/bin/iAPX286 target=../../usr/bin/i286
hardlink path=usr/bin/jobs target=../../usr/bin/alias
hardlink path=usr/bin/join target=../../usr/bin/alias
hardlink path=usr/bin/kill target=../../usr/bin/alias
hardlink path=usr/bin/ksh target=../../usr/lib/isaexec
hardlink path=usr/bin/ksh93 target=../../usr/lib/isaexec
hardlink path=usr/bin/ln target=../../usr/bin/cp
hardlink path=usr/bin/logname target=../../usr/bin/alias
hardlink path=usr/bin/m68k target=../../usr/bin/i286
hardlink path=usr/bin/mac target=../../usr/lib/isaexec
hardlink path=usr/bin/mc68000 target=../../usr/bin/i286
hardlink path=usr/bin/mc68010 target=../../usr/bin/i286
hardlink path=usr/bin/mc68020 target=../../usr/bin/i286
hardlink path=usr/bin/mc68030 target=../../usr/bin/i286
hardlink path=usr/bin/mc68040 target=../../usr/bin/i286
hardlink path=usr/bin/mv target=../../usr/bin/cp
hardlink path=usr/bin/newtask target=../../usr/lib/isaexec
hardlink path=usr/bin/nohup target=../../usr/lib/isaexec
hardlink path=usr/bin/page target=../../usr/bin/more
hardlink path=usr/bin/paste target=../../usr/bin/alias
hardlink path=usr/bin/pdp11 target=../../usr/bin/i286
hardlink path=usr/bin/pfbash target=../../usr/bin/pfexec
hardlink path=usr/bin/pfcsh target=../../usr/bin/pfexec
hardlink path=usr/bin/pfksh target=../../usr/bin/pfexec
hardlink path=usr/bin/pfksh93 target=../../usr/bin/pfexec
hardlink path=usr/bin/pfrksh target=../../usr/bin/pfexec
hardlink path=usr/bin/pfrksh93 target=../../usr/bin/pfexec
hardlink path=usr/bin/pfsh target=../../usr/bin/pfexec
hardlink path=usr/bin/pftcsh target=../../usr/bin/pfexec
hardlink path=usr/bin/pfzsh target=../../usr/bin/pfexec
hardlink path=usr/bin/pkill target=../../usr/bin/pgrep
hardlink path=usr/bin/prctl target=../../usr/lib/isaexec
hardlink path=usr/bin/print target=../../usr/bin/alias
hardlink path=usr/bin/prstat target=../../usr/lib/isaexec
hardlink path=usr/bin/ps target=../../usr/lib/isaexec
hardlink path=usr/bin/read target=../../usr/bin/alias
hardlink path=usr/bin/red target=../../usr/bin/ed
hardlink path=usr/bin/rev target=../../usr/bin/alias
hardlink path=usr/bin/rksh target=../../usr/lib/isaexec
hardlink path=usr/bin/rksh93 target=../../usr/lib/isaexec
hardlink path=usr/bin/savecore target=../../usr/lib/isaexec
hardlink path=usr/bin/setuname target=../../usr/lib/isaexec
hardlink path=usr/bin/sleep target=../../usr/bin/alias
hardlink path=usr/bin/sparc target=../../usr/bin/i286
hardlink path=usr/bin/sum target=../../usr/bin/alias
hardlink path=usr/bin/sun target=../../usr/bin/i286
hardlink path=usr/bin/sun2 target=../../usr/bin/i286
hardlink path=usr/bin/sun3 target=../../usr/bin/i286
hardlink path=usr/bin/sun3x target=../../usr/bin/i286
hardlink path=usr/bin/sun4 target=../../usr/bin/i286
hardlink path=usr/bin/sun4c target=../../usr/bin/i286
hardlink path=usr/bin/sun4d target=../../usr/bin/i286
hardlink path=usr/bin/sun4e target=../../usr/bin/i286
hardlink path=usr/bin/sun4m target=../../usr/bin/i286
hardlink path=usr/bin/tee target=../../usr/bin/alias
hardlink path=usr/bin/test target=../../usr/bin/alias
hardlink path=usr/bin/touch target=../../usr/bin/settime
hardlink path=usr/bin/type target=../../usr/bin/alias
hardlink path=usr/bin/u370 target=../../usr/bin/i286
hardlink path=usr/bin/u3b target=../../usr/bin/i286
hardlink path=usr/bin/u3b15 target=../../usr/bin/i286
hardlink path=usr/bin/u3b2 target=../../usr/bin/i286
hardlink path=usr/bin/u3b5 target=../../usr/bin/i286
hardlink path=usr/bin/ulimit target=../../usr/bin/alias
hardlink path=usr/bin/umask target=../../usr/bin/alias
hardlink path=usr/bin/unalias target=../../usr/bin/alias
hardlink path=usr/bin/uniq target=../../usr/bin/alias
hardlink path=usr/bin/uptime target=../../usr/lib/isaexec
hardlink path=usr/bin/vax target=../../usr/bin/i286
hardlink path=usr/bin/vedit target=../has/bin/edit
hardlink path=usr/bin/w target=../../usr/lib/isaexec
hardlink path=usr/bin/wait target=../../usr/bin/alias
hardlink path=usr/bin/wc target=../../usr/bin/alias
hardlink path=usr/has/bin/ex target=edit
hardlink path=usr/has/bin/pfsh target=../../bin/pfexec
hardlink path=usr/has/bin/vedit target=edit
hardlink path=usr/has/bin/vi target=edit
hardlink path=usr/has/bin/view target=edit
hardlink path=usr/lib/fs/hsfs/fstyp target=../../../sbin/fstyp
hardlink path=usr/lib/fs/ufs/dcopy target=../../../../usr/lib/fs/ufs/clri
hardlink path=usr/lib/fs/ufs/fstyp target=../../../sbin/fstyp
hardlink path=usr/lib/fs/ufs/quotaon \
    target=../../../../usr/lib/fs/ufs/quotaoff
hardlink path=usr/lib/inet/in.iked target=../../../usr/lib/isaexec
hardlink path=usr/sadm/bin/dispgid target=../../../usr/bin/ckgid
hardlink path=usr/sadm/bin/dispuid target=../../../usr/bin/ckuid
hardlink path=usr/sadm/bin/errange target=../../../usr/bin/ckrange
hardlink path=usr/sadm/bin/errdate target=../../../usr/bin/ckdate
hardlink path=usr/sadm/bin/errgid target=../../../usr/bin/ckgid
hardlink path=usr/sadm/bin/errint target=../../../usr/bin/ckint
hardlink path=usr/sadm/bin/erritem target=../../../usr/bin/ckitem
hardlink path=usr/sadm/bin/errpath target=../../../usr/bin/ckpath
hardlink path=usr/sadm/bin/errstr target=../../../usr/bin/ckstr
hardlink path=usr/sadm/bin/errtime target=../../../usr/bin/cktime
hardlink path=usr/sadm/bin/erruid target=../../../usr/bin/ckuid
hardlink path=usr/sadm/bin/erryorn target=../../../usr/bin/ckyorn
hardlink path=usr/sadm/bin/helpdate target=../../../usr/bin/ckdate
hardlink path=usr/sadm/bin/helpgid target=../../../usr/bin/ckgid
hardlink path=usr/sadm/bin/helpint target=../../../usr/bin/ckint
hardlink path=usr/sadm/bin/helpitem target=../../../usr/bin/ckitem
hardlink path=usr/sadm/bin/helppath target=../../../usr/bin/ckpath
hardlink path=usr/sadm/bin/helprange target=../../../usr/bin/ckrange
hardlink path=usr/sadm/bin/helpstr target=../../../usr/bin/ckstr
hardlink path=usr/sadm/bin/helptime target=../../../usr/bin/cktime
hardlink path=usr/sadm/bin/helpuid target=../../../usr/bin/ckuid
hardlink path=usr/sadm/bin/helpyorn target=../../../usr/bin/ckyorn
hardlink path=usr/sadm/bin/valdate target=../../../usr/bin/ckdate
hardlink path=usr/sadm/bin/valgid target=../../../usr/bin/ckgid
hardlink path=usr/sadm/bin/valint target=../../../usr/bin/ckint
hardlink path=usr/sadm/bin/valpath target=../../../usr/bin/ckpath
hardlink path=usr/sadm/bin/valrange target=../../../usr/bin/ckrange
hardlink path=usr/sadm/bin/valstr target=../../../usr/bin/ckstr
hardlink path=usr/sadm/bin/valtime target=../../../usr/bin/cktime
hardlink path=usr/sadm/bin/valuid target=../../../usr/bin/ckuid
hardlink path=usr/sadm/bin/valyorn target=../../../usr/bin/ckyorn
hardlink path=usr/sbin/add_drv target=../../usr/lib/isaexec
hardlink path=usr/sbin/audlinks target=./devfsadm
hardlink path=usr/sbin/consadmd target=../../usr/sbin/consadm
hardlink path=usr/sbin/deallocate target=../../usr/sbin/allocate
hardlink path=usr/sbin/devlinks target=./devfsadm
```

```
hardlink path=usr/sbin/dfshares target=../../usr/sbin/dfmounts
hardlink path=usr/sbin/disks target=./devfsadm
hardlink path=usr/sbin/drvconfig target=./devfsadm
hardlink path=usr/sbin/list_devices target=../../usr/sbin/allocate
hardlink path=usr/sbin/mkdevmaps target=../../usr/sbin/mkdevalloc
hardlink path=usr/sbin/modinfo target=../../usr/lib/isaexec
hardlink path=usr/sbin/modload target=../../usr/lib/isaexec
hardlink path=usr/sbin/modunload target=../../usr/lib/isaexec
hardlink path=usr/sbin/pbind target=../../usr/lib/isaexec
hardlink path=usr/sbin/ports target=./devfsadm
hardlink path=usr/sbin/poweroff target=./halt
hardlink path=usr/sbin/prtconf target=../../usr/lib/isaexec
$(sparc_ONLY)hardlink path=usr/sbin/prtdiag target=../../usr/lib/platexec
hardlink path=usr/sbin/psrset target=../../usr/lib/isaexec
hardlink path=usr/sbin/reboot target=./halt
hardlink path=usr/sbin/rem_drv target=../../usr/lib/isaexec
hardlink path=usr/sbin/roleadd target=../../usr/sbin/useradd
hardlink path=usr/sbin/roledel target=../../usr/sbin/userdel
hardlink path=usr/sbin/rolemod target=../../usr/sbin/usermod
hardlink path=usr/sbin/share target=../../usr/sbin/sharemgr
hardlink path=usr/sbin/swap target=../../usr/lib/isaexec
hardlink path=usr/sbin/sysdef target=../../usr/lib/isaexec
hardlink path=usr/sbin/tapes target=./devfsadm
hardlink path=usr/sbin/unshare target=../../usr/sbin/sharemgr
hardlink path=usr/sbin/update_drv target=../../usr/lib/isaexec
hardlink path=usr/sbin/whodo target=../../usr/lib/isaexec
legacy pkg=SUNWcsr \
    desc="core software for a specific instruction-set architecture" \
    name="Core Solaris, (Root)"
legacy pkg=SUNWcsu \
    desc="core software for a specific instruction-set architecture" \
    name="Core Solaris, (Usr)"
legacy pkg=SUNWftpr desc="FTP Server Configuration Files" \
    name="FTP Server, (Root)"
license cr_Sun license=cr_Sun
license lic_CDDL license=lic_CDDL
license usr/src/cmd/cmd-inet/sbin/ifparse/THIRDPARTYLICENSE \
    license=usr/src/cmd/cmd-inet/sbin/ifparse/THIRDPARTYLICENSE
license usr/src/cmd/cmd-inet/usr.lib/in.mpathd/THIRDPARTYLICENSE \
    license=usr/src/cmd/cmd-inet/usr.lib/in.mpathd/THIRDPARTYLICENSE
license usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.arp \
    license=usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.arp
license usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.route \
    license=usr/src/cmd/cmd-inet/usr.sbin/THIRDPARTYLICENSE.route
license usr/src/cmd/cmd-inet/usr.sbin/ifconfig/THIRDPARTYLICENSE \
    license=usr/src/cmd/cmd-inet/usr.sbin/ifconfig/THIRDPARTYLICENSE
license usr/src/cmd/cmd-inet/usr.sbin/in.ftpd/LICENSE \
    license=usr/src/cmd/cmd-inet/usr.sbin/in.ftpd/LICENSE
license usr/src/cmd/cmd-inet/usr.sbin/traceroute/THIRDPARTYLICENSE \
    license=usr/src/cmd/cmd-inet/usr.sbin/traceroute/THIRDPARTYLICENSE
license usr/src/cmd/cron/THIRDPARTYLICENSE \
    license=usr/src/cmd/cron/THIRDPARTYLICENSE
license usr/src/cmd/csh/THIRDPARTYLICENSE \
    license=usr/src/cmd/csh/THIRDPARTYLICENSE
license usr/src/cmd/eeprom/THIRDPARTYLICENSE \
    license=usr/src/cmd/eeprom/THIRDPARTYLICENSE
license usr/src/cmd/fs.d/ufs/THIRDPARTYLICENSE \
    license=usr/src/cmd/fs.d/ufs/THIRDPARTYLICENSE
license usr/src/cmd/mt/THIRDPARTYLICENSE \
    license=usr/src/cmd/mt/THIRDPARTYLICENSE
license usr/src/cmd/script/THIRDPARTYLICENSE \
    license=usr/src/cmd/script/THIRDPARTYLICENSE
license usr/src/cmd/sed/THIRDPARTYLICENSE \
    license=usr/src/cmd/sed/THIRDPARTYLICENSE
license usr/src/cmd/stat/vmstat/THIRDPARTYLICENSE \
    license=usr/src/cmd/stat/vmstat/THIRDPARTYLICENSE
```

```
license usr/src/cmd/tail/THIRDPARTYLICENSE \
    license=usr/src/cmd/tail/THIRDPARTYLICENSE
license usr/src/cmd/tip/THIRDPARTYLICENSE \
    license=usr/src/cmd/tip/THIRDPARTYLICENSE
license usr/src/cmd/tr/THIRDPARTYLICENSE \
    license=usr/src/cmd/tr/THIRDPARTYLICENSE
license usr/src/cmd/vi/THIRDPARTYLICENSE \
    license=usr/src/cmd/vi/THIRDPARTYLICENSE
license usr/src/cmd/which/THIRDPARTYLICENSE \
    license=usr/src/cmd/which/THIRDPARTYLICENSE
license usr/src/cmd/xstr/THIRDPARTYLICENSE \
    license=usr/src/cmd/xstr/THIRDPARTYLICENSE
license usr/src/common/bzip2/LICENSE license=usr/src/common/bzip2/LICENSE
link path=bin target=./usr/bin
link path=etc/TIMEZONE target=./default/init
link path=etc/autopush target=../sbin/autopush
link path=etc/cfgadm target=../usr/sbin/cfgadm
link path=etc/clri target=../usr/sbin/clri
link path=etc/cron target=../usr/sbin/cron
link path=etc/dcopy target=../usr/sbin/dcopy
link path=etc/ff target=../usr/sbin/ff
link path=etc/fmthard target=../usr/sbin/fmthard
link path=etc/format target=../usr/sbin/format
link path=etc/fsck target=../usr/sbin/fsck
link path=etc/fsdb target=../usr/sbin/fsdb
link path=etc/fstyp target=../usr/sbin/fstyp
link path=etc/getty target=../usr/lib/saf/ttymon
link path=etc/grpck target=../usr/sbin/grpck
link path=etc/halt target=../usr/sbin/halt
link path=etc/hosts target=./inet/hosts
link path=etc/inet/ipnodes target=./hosts
link path=etc/inetd.conf target=./inet/inetd.conf
link path=etc/init target=../sbin/init
link path=etc/install target=../usr/sbin/install
link path=etc/killall target=../usr/sbin/killall
link path=etc/labelit target=../usr/sbin/labelit
link path=etc/lib/ld.so.1 target=../../lib/ld.so.1
link path=etc/lib/libdl.so.1 target=../../lib/libdl.so.1
link path=etc/lib/nss_files.so.1 target=../../lib/nss_files.so.1
link path=etc/log target=../var/adm/log
link path=etc/mkfs target=../usr/sbin/mkfs
link path=etc/mknod target=../usr/sbin/mknod
link path=etc/mount target=../sbin/mount
link path=etc/mountall target=../sbin/mountall
link path=etc/ncheck target=../usr/sbin/ncheck
link path=etc/netmasks target=./inet/netmasks
link path=etc/networks target=./inet/networks
link path=etc/protocols target=./inet/protocols
link path=etc/prtconf target=../usr/sbin/prtconf
link path=etc/prtvtoc target=../usr/sbin/prtvtoc
link path=etc/rc0 target=../sbin/rc0
link path=etc/rc1 target=../sbin/rc1
link path=etc/rc2 target=../sbin/rc2
link path=etc/rc3 target=../sbin/rc3
link path=etc/rc5 target=../sbin/rc5
link path=etc/rc6 target=../sbin/rc6
link path=etc/rcS target=../sbin/rcS
link path=etc/reboot target=../usr/sbin/halt
link path=etc/security/audit/localhost/files target=../../../../var/audit
link path=etc/services target=./inet/services
link path=etc/setmnt target=../usr/sbin/setmnt
link path=etc/shutdown target=../usr/sbin/shutdown
link path=etc/sulogin target=../sbin/sulogin
link path=etc/swap target=../usr/sbin/swap
link path=etc/swapadd target=../sbin/swapadd
link path=etc/sysdef target=../usr/sbin/sysdef
```

```
link path=etc/tar target=../usr/sbin/tar
link path=etc/telinit target=../sbin/init
link path=etc/uadmin target=../sbin/uadmin
link path=etc/umount target=../sbin/umount
link path=etc/umountall target=../sbin/umountall
link path=etc/utmpx target=../var/adm/utmpx
link path=etc/volcopy target=../usr/sbin/volcopy
link path=etc/wall target=../usr/sbin/wall
link path=etc/whodo target=../usr/sbin/whodo
link path=etc/wtmpx target=../var/adm/wtmpx
link path=sbin/in.mpathd target=../lib/inet/in.mpathd
link path=sbin/jsh target=../usr/bin/ksh93
link path=sbin/pfsh target=../usr/bin/pfexec
link path=sbin/sh target=../usr/bin/$(ARCH32)/ksh93
link path=sbin/su target=../usr/bin/su
link path=usr/adm target=../var/adm
link path=usr/bin/cachefspack target=../lib/fs/cachefs/cachefspack
link path=usr/bin/cachefsstat target=../lib/fs/cachefs/cachefsstat
link path=usr/bin/df target=../sbin/df
link path=usr/bin/jsh target=ksh93
link path=usr/bin/pwconv target=../sbin/pwconv
link path=usr/bin/rmail target=./mail
link path=usr/bin/sh target=$(ARCH32)/ksh93
link path=usr/bin/strclean target=../sbin/strclean
link path=usr/bin/strerr target=../sbin/strerr
link path=usr/bin/sync target=../../sbin/sync
link path=usr/bin/tar target=../sbin/tar
link path=usr/bin/uname target=../../sbin/uname
link path=usr/ccs/bin/m4 target=../../bin/m4
link path=usr/has/bin/jsh target=sh
link path=usr/has/lib/rsh target=../bin/sh
link path=usr/lib/$(ARCH64)/ld.so.1 target=../../../lib/$(ARCH64)/ld.so.1
link path=usr/lib/cron target=../../etc/cron.d
link path=usr/lib/devfsadm/devfsadmd target=../../sbin/devfsadm
link path=usr/lib/embedded_su target=../bin/su
link path=usr/lib/fs/dev/mount target=../../../../etc/fs/dev/mount
link path=usr/lib/fs/hsfs/mount target=../../../../etc/fs/hsfs/mount
link path=usr/lib/fs/ufs/mount target=../../../../etc/fs/ufs/mount
link path=usr/lib/inet/in.mpathd target=../../../lib/inet/in.mpathd
link path=usr/lib/ld.so.1 target=../../lib/ld.so.1
link path=usr/lib/locale/POSIX target=./C
link path=usr/lib/rsh target=../bin/ksh93
link path=usr/lib/secure/32 target=.
link path=usr/lib/secure/64 target=$(ARCH64)
link path=usr/lib/wusbd target=../../sbin/wusbadm
link path=usr/mail target=../var/mail
link path=usr/net/nls/listen target=../../lib/saf/listen
link path=usr/net/nls/nlps_server target=../../lib/saf/nlps_server
link path=usr/news target=../var/news
link path=usr/preserve target=../var/preserve
link path=usr/pub target=./share/lib/pub
link path=usr/sbin/autopush target=../../sbin/autopush
link path=usr/sbin/bootadm target=../../sbin/bootadm
link path=usr/sbin/cachefslog target=../lib/fs/cachefs/cachefslog
link path=usr/sbin/cachefswssize target=../lib/fs/cachefs/cachefswssize
link path=usr/sbin/cfsadmin target=../lib/fs/cachefs/cfsadmin
link path=usr/sbin/cryptoadm target=../../sbin/cryptoadm
link path=usr/sbin/dcopy target=./clri
link path=usr/sbin/devnm target=./df
link path=usr/sbin/dladm target=../../sbin/dladm
link path=usr/sbin/dlstat target=../../sbin/dlstat
link path=usr/sbin/edquota target=../lib/fs/ufs/edquota
link path=usr/sbin/fdisk target=../../sbin/fdisk
link path=usr/sbin/fiocompress target=../../sbin/fiocompress
link path=usr/sbin/flowadm target=../../sbin/flowadm
link path=usr/sbin/flowstat target=../../sbin/flowstat
link path=usr/sbin/fsdb target=./clri
link path=usr/sbin/fsirand target=../lib/fs/ufs/fsirand
link path=usr/sbin/fssnap target=./clri
link path=usr/sbin/hostconfig target=../../sbin/hostconfig
link path=usr/sbin/ifconfig target=../../sbin/ifconfig
link path=usr/sbin/inetd target=../lib/inet/inetd
link path=usr/sbin/init target=../../sbin/init
$(i386_ONLY)link path=usr/sbin/installgrub target=../../sbin/installgrub
link path=usr/sbin/ipadm target=../../sbin/ipadm
link path=usr/sbin/ipmpstat target=../../sbin/ipmpstat
link path=usr/sbin/labelit target=./clri
link path=usr/sbin/lockfs target=../lib/fs/ufs/lockfs
link path=usr/sbin/mkfs target=./clri
link path=usr/sbin/mount target=../../sbin/mount
link path=usr/sbin/ncheck target=./ff
link path=usr/sbin/newfs target=../lib/fs/ufs/newfs
link path=usr/sbin/quot target=../lib/fs/ufs/quot
link path=usr/sbin/quota target=../lib/fs/ufs/quota
link path=usr/sbin/quotacheck target=../lib/fs/ufs/quotacheck
link path=usr/sbin/quotaoff target=../lib/fs/ufs/quotaoff
link path=usr/sbin/quotaon target=../lib/fs/ufs/quotaon
link path=usr/sbin/repquota target=../lib/fs/ufs/repquota
link path=usr/sbin/route target=../../sbin/route
link path=usr/sbin/routeadm target=../../sbin/routeadm
link path=usr/sbin/sync target=../../sbin/sync
link path=usr/sbin/tunefs target=../lib/fs/ufs/tunefs
link path=usr/sbin/tzreload target=../../sbin/tzreload
link path=usr/sbin/uadmin target=../../sbin/uadmin
link path=usr/sbin/ufsdump target=../lib/fs/ufs/ufsdump
link path=usr/sbin/ufsrestore target=../lib/fs/ufs/ufsrestore
link path=usr/sbin/umount target=../../sbin/umount
link path=usr/sbin/wusbadm target=../../sbin/wusbadm
link path=usr/spool target=../var/spool
link path=usr/src target=./share/src
link path=usr/tmp target=../var/tmp
link path=var/ld/32 target=.
link path=var/ld/64 target=$(ARCH64)
#
# The bootadm binary needs the etc/release file.
#
depend fmri=release/name type=require
#
# intrd and others use the illumos-defaulted perl interpreter
#
depend fmri=runtime/perl-510 type=require
#
# The loadkeys binary needs the keytables.
#
depend fmri=system/data/keyboard/keytables type=require
#
# Depend on terminfo data.
#
depend fmri=system/data/terminfo type=require
#
# Depend on zoneinfo data.
#
depend fmri=system/data/zoneinfo type=require
```

```
**********************************************************
   60132 Mon Aug 12 18:24:56 2013
new/usr/src/pkg/manifests/system-library.mf
2989 Eliminate use of LOGNAME_MAX in ON
1166 useradd have warning with name more 8 chars
**********************************************************
#
# 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) 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2012 OmniTI Computer Consulting, Inc.  All rights reserved.
# Copyright (c) 2013 Gary Mills
#

<include system-library.man3.inc>
<include system-library.man3bsm.inc>
<include system-library.man3c.inc>
<include system-library.man3c_db.inc>
<include system-library.man3cfgadm.inc>
<include system-library.man3commputil.inc>
<include system-library.man3contract.inc>
<include system-library.man3curses.inc>
<include system-library.man3devid.inc>
<include system-library.man3devinfo.inc>
<include system-library.man3dlpi.inc>
<include system-library.man3elf.inc>
<include system-library.man3exacct.inc>
<include system-library.man3ext.inc>
<include system-library.man3fstyp.inc>
<include system-library.man3gen.inc>
<include system-library.man3kstat.inc>
<include system-library.man3kvm.inc>
<include system-library.man3ldap.inc>
<include system-library.man3lgrp.inc>
<include system-library.man3lib.inc>
<include system-library.man3mail.inc>
<include system-library.man3malloc.inc>
<include system-library.man3mp.inc>
<include system-library.man3nsl.inc>
<include system-library.man3nvpair.inc>
<include system-library.man3pam.inc>
<include system-library.man3scf.inc>
<include system-library.man3sec.inc>
<include system-library.man3secdb.inc>
<include system-library.man3sip.inc>
<include system-library.man3socket.inc>
```

```
<include system-library.man3tsol.inc>
<include system-library.man3uuid.inc>
<include system-library.man3volmgt.inc>
<include system-library.man3xcurses.inc>
<include system-library.man3xnet.inc>
<include system-library.man4.inc>
<include system-library.man5.inc>
<include system-library.man7p.inc>
set name=pkg.fmri value=pkg:/system/library@$(PKGVERS)
set name=pkg.description \
    value="core shared libraries for a specific instruction-set architecture"
set name=pkg.summary value="Core Solaris, (Shared Libs)"
set name=info.classification value=org.opensolaris.category.2008:System/Core
set name=variant.arch value=$(ARCH)
$(i386_ONLY)dir path=etc group=sys
$(i386_ONLY)dir path=etc/flash group=sys
$(i386_ONLY)dir path=etc/flash/postcreation group=sys mode=0700
$(i386_ONLY)dir path=etc/flash/precreation group=sys mode=0700
$(i386_ONLY)dir path=etc/flash/preexit group=sys mode=0700
dir path=lib
dir path=lib/$(ARCH64)
dir path=lib/crypto
dir path=lib/crypto/$(ARCH64)
dir path=lib/mpxio
dir path=lib/secure
dir path=lib/secure/$(ARCH64)
dir path=usr group=sys
dir path=usr/bin
dir path=usr/ccs
dir path=usr/ccs/lib
dir path=usr/ccs/lib/$(ARCH64)
dir path=usr/lib
dir path=usr/lib/$(ARCH64)
dir path=usr/lib/cfgadm
dir path=usr/lib/cfgadm/$(ARCH64)
dir path=usr/lib/iconv/$(ARCH64)
$(i386_ONLY)dir path=usr/lib/libc
dir path=usr/lib/lwp
dir path=usr/lib/lwp/$(ARCH64)
dir path=usr/lib/python2.6
dir path=usr/lib/python2.6/vendor-packages
dir path=usr/lib/python2.6/vendor-packages/solaris
dir path=usr/lib/raidcfg
dir path=usr/lib/raidcfg/$(ARCH64)
dir path=usr/lib/scsi
dir path=usr/lib/scsi/$(ARCH64)
dir path=usr/lib/scsi/plugins
dir path=usr/lib/scsi/plugins/scsi
dir path=usr/lib/scsi/plugins/scsi/engines
dir path=usr/lib/scsi/plugins/scsi/engines/$(ARCH64)
dir path=usr/lib/scsi/plugins/ses
dir path=usr/lib/scsi/plugins/ses/framework
dir path=usr/lib/scsi/plugins/ses/framework/$(ARCH64)
dir path=usr/lib/scsi/plugins/ses/vendor
$(sparc_ONLY)dir path=usr/lib/scsi/plugins/ses/vendor/$(ARCH64)
dir path=usr/lib/scsi/plugins/smp
dir path=usr/lib/scsi/plugins/smp/engine
dir path=usr/lib/scsi/plugins/smp/engine/$(ARCH64)
dir path=usr/lib/scsi/plugins/smp/framework
dir path=usr/lib/scsi/plugins/smp/framework/$(ARCH64)
dir path=usr/lib/security
dir path=usr/lib/security/$(ARCH64)
dir path=usr/share/man
dir path=usr/share/man/man3
dir path=usr/share/man/man3bsm
dir path=usr/share/man/man3c
dir path=usr/share/man/man3c_db
dir path=usr/share/man/man3cfgadm
dir path=usr/share/man/man3commputil
dir path=usr/share/man/man3contract
dir path=usr/share/man/man3curses
dir path=usr/share/man/man3devid
dir path=usr/share/man/man3devinfo
dir path=usr/share/man/man3dlpi
dir path=usr/share/man/man3elf
dir path=usr/share/man/man3exacct
dir path=usr/share/man/man3ext
dir path=usr/share/man/man3fstyp
dir path=usr/share/man/man3gen
dir path=usr/share/man/man3kstat
dir path=usr/share/man/man3kvm
dir path=usr/share/man/man3ldap
dir path=usr/share/man/man3lgrp
dir path=usr/share/man/man3lib
dir path=usr/share/man/man3mail
dir path=usr/share/man/man3malloc
dir path=usr/share/man/man3mp
dir path=usr/share/man/man3nsl
dir path=usr/share/man/man3nvpair
dir path=usr/share/man/man3pam
dir path=usr/share/man/man3pool
dir path=usr/share/man/man3scf
dir path=usr/share/man/man3sec
dir path=usr/share/man/man3secdb
dir path=usr/share/man/man3sip
dir path=usr/share/man/man3socket
dir path=usr/share/man/man3tsol
dir path=usr/share/man/man3uuid
dir path=usr/share/man/man3volmgt
dir path=usr/share/man/man3xcurses
dir path=usr/share/man/man3xnet
dir path=usr/share/man/man5
dir path=usr/share/man/man7p
dir path=usr/xpg4
dir path=usr/xpg4/lib
dir path=usr/xpg4/lib/$(ARCH64)
$(i386_ONLY)file path=etc/flash/precreation/caplib group=sys mode=0500
file path=lib/$(ARCH64)/c_synonyms.so.1
file path=lib/$(ARCH64)/ld.so.1
file path=lib/$(ARCH64)/libadm.so.1
file path=lib/$(ARCH64)/libaio.so.1
file path=lib/$(ARCH64)/libavl.so.1
file path=lib/$(ARCH64)/libbsm.so.1
file path=lib/$(ARCH64)/libc.so.1
file path=lib/$(ARCH64)/libc_db.so.1
file path=lib/$(ARCH64)/libcmdutils.so.1
file path=lib/$(ARCH64)/libcontract.so.1
file path=lib/$(ARCH64)/libcryptoutil.so.1
file path=lib/$(ARCH64)/libctf.so.1
file path=lib/$(ARCH64)/libcurses.so.1
file path=lib/$(ARCH64)/libdevice.so.1
file path=lib/$(ARCH64)/libdevid.so.1
file path=lib/$(ARCH64)/libdevinfo.so.1
file path=lib/$(ARCH64)/libdhcputil.so.1
file path=lib/$(ARCH64)/libdl.so.1
file path=lib/$(ARCH64)/libdladm.so.1
file path=lib/$(ARCH64)/libdlpi.so.1
file path=lib/$(ARCH64)/libdoor.so.1
file path=lib/$(ARCH64)/libefi.so.1
file path=lib/$(ARCH64)/libelf.so.1
$(i386_ONLY)file path=lib/$(ARCH64)/libfdisk.so.1
file path=lib/$(ARCH64)/libgen.so.1
```

```
file path=lib/$(ARCH64)/libinetutil.so.1
file path=lib/$(ARCH64)/libintl.so.1
file path=lib/$(ARCH64)/libkmf.so.1
file path=lib/$(ARCH64)/libkmfberder.so.1
file path=lib/$(ARCH64)/libkstat.so.1
file path=lib/$(ARCH64)/libld.so.4
file path=lib/$(ARCH64)/liblddbg.so.4
file path=lib/$(ARCH64)/libmd.so.1
file path=lib/$(ARCH64)/libmd5.so.1
file path=lib/$(ARCH64)/libmp.so.2
file path=lib/$(ARCH64)/libnsl.so.1
file path=lib/$(ARCH64)/libnvpair.so.1
file path=lib/$(ARCH64)/libpam.so.1
file path=lib/$(ARCH64)/libproc.so.1
file path=lib/$(ARCH64)/libpthread.so.1
file path=lib/$(ARCH64)/librcm.so.1
file path=lib/$(ARCH64)/libresolv.so.2
file path=lib/$(ARCH64)/librestart.so.1
file path=lib/$(ARCH64)/librpcsvc.so.1
file path=lib/$(ARCH64)/librt.so.1
file path=lib/$(ARCH64)/librtld.so.1
file path=lib/$(ARCH64)/librtld_db.so.1
file path=lib/$(ARCH64)/libscf.so.1
file path=lib/$(ARCH64)/libsec.so.1
file path=lib/$(ARCH64)/libsecdb.so.1
file path=lib/$(ARCH64)/libsendfile.so.1
file path=lib/$(ARCH64)/libsocket.so.1
file path=lib/$(ARCH64)/libsysevent.so.1
file path=lib/$(ARCH64)/libtermcap.so.1
file path=lib/$(ARCH64)/libthread.so.1
file path=lib/$(ARCH64)/libtsnet.so.1
file path=lib/$(ARCH64)/libtsol.so.2
file path=lib/$(ARCH64)/libumem.so.1
file path=lib/$(ARCH64)/libuuid.so.1
file path=lib/$(ARCH64)/libuutil.so.1
file path=lib/$(ARCH64)/libw.so.1
file path=lib/$(ARCH64)/libxnet.so.1
file path=lib/$(ARCH64)/nss_compat.so.1
file path=lib/$(ARCH64)/nss_dns.so.1
file path=lib/$(ARCH64)/nss_files.so.1
file path=lib/$(ARCH64)/nss_nis.so.1
file path=lib/$(ARCH64)/nss_user.so.1
file path=lib/c_synonyms.so.1
file path=lib/crypto/$(ARCH64)/kmf_mapper_cn.so.1
file path=lib/crypto/$(ARCH64)/kmf_nss.so.1
file path=lib/crypto/$(ARCH64)/kmf_openssl.so.1
file path=lib/crypto/$(ARCH64)/kmf_pkcs11.so.1
file path=lib/crypto/kmf_mapper_cn.so.1
file path=lib/crypto/kmf_nss.so.1
file path=lib/crypto/kmf_openssl.so.1
file path=lib/crypto/kmf_pkcs11.so.1
file path=lib/ld.so.1
file path=lib/libadm.so.1
file path=lib/libaio.so.1
file path=lib/libavl.so.1
file path=lib/libbsm.so.1
file path=lib/libc.so.1 reboot-needed=true
file path=lib/libc_db.so.1
file path=lib/libcmdutils.so.1
file path=lib/libcontract.so.1
file path=lib/libcryptoutil.so.1
file path=lib/libctf.so.1
file path=lib/libcurses.so.1
file path=lib/libdevice.so.1
file path=lib/libdevid.so.1
file path=lib/libdevinfo.so.1
file path=lib/libdhcpagent.so.1
file path=lib/libdhcputil.so.1
file path=lib/libdl.so.1
file path=lib/libdladm.so.1
file path=lib/libdlpi.so.1
file path=lib/libdoor.so.1
file path=lib/libefi.so.1
file path=lib/libelf.so.1
file path=lib/libelfsign.so.1
$(i386_ONLY)file path=lib/libfdisk.so.1
file path=lib/libgen.so.1
file path=lib/libinetutil.so.1
file path=lib/libintl.so.1
file path=lib/libipadm.so.1
file path=lib/libipmp.so.1
file path=lib/libkcfd.so.1
file path=lib/libkmf.so.1
file path=lib/libkmfberder.so.1
file path=lib/libkstat.so.1
file path=lib/libld.so.4
file path=lib/liblddbg.so.4
file path=lib/libmd.so.1
file path=lib/libmd5.so.1
file path=lib/libmp.so.1
file path=lib/libmp.so.2
file path=lib/libnsl.so.1
file path=lib/libnvpair.so.1
file path=lib/libnwam.so.1
file path=lib/libpam.so.1
file path=lib/libproc.so.1
file path=lib/libpthread.so.1
file path=lib/librcm.so.1
file path=lib/libresolv.so.1
file path=lib/libresolv.so.2
file path=lib/librestart.so.1
file path=lib/librpcsvc.so.1
file path=lib/librt.so.1
file path=lib/librtld.so.1
file path=lib/librtld_db.so.1
file path=lib/libscf.so.1
file path=lib/libsec.so.1
file path=lib/libsecdb.so.1
file path=lib/libsendfile.so.1
file path=lib/libsocket.so.1
file path=lib/libsysevent.so.1
file path=lib/libtermcap.so.1
file path=lib/libthread.so.1
file path=lib/libtsnet.so.1
file path=lib/libtsol.so.2
file path=lib/libumem.so.1
file path=lib/libuuid.so.1
file path=lib/libuutil.so.1
file path=lib/libw.so.1
file path=lib/libxnet.so.1
file path=lib/mpxio/stmsboot_util mode=0555
file path=lib/nss_compat.so.1
file path=lib/nss_dns.so.1
file path=lib/nss_files.so.1
file path=lib/nss_nis.so.1
file path=lib/nss_user.so.1
file path=usr/lib/$(ARCH64)/0@0.so.1
file path=usr/lib/$(ARCH64)/getloginx.so.1
file path=usr/lib/$(ARCH64)/libadutils.so.1
file path=usr/lib/$(ARCH64)/libast.so.1
file path=usr/lib/$(ARCH64)/libbsdmalloc.so.1
file path=usr/lib/$(ARCH64)/libcfgadm.so.1
```

```
file path=usr/lib/$(ARCH64)/libcmd.so.1
file path=usr/lib/$(ARCH64)/libcommputil.so.1
file path=usr/lib/$(ARCH64)/libcrle.so.1
file path=usr/lib/$(ARCH64)/libcrypt.so.1
file path=usr/lib/$(ARCH64)/libdisasm.so.1
file path=usr/lib/$(ARCH64)/libdll.so.1
file path=usr/lib/$(ARCH64)/libexacct.so.1
file path=usr/lib/$(ARCH64)/libform.so.1
file path=usr/lib/$(ARCH64)/libfstyp.so.1
file path=usr/lib/$(ARCH64)/libhotplug.so.1
file path=usr/lib/$(ARCH64)/libidmap.so.1
file path=usr/lib/$(ARCH64)/libike.so.1
file path=usr/lib/$(ARCH64)/libipmi.so.1
file path=usr/lib/$(ARCH64)/libipp.so.1
file path=usr/lib/$(ARCH64)/libipsecutil.so.1
file path=usr/lib/$(ARCH64)/libkvm.so.1
file path=usr/lib/$(ARCH64)/libl.so.1
file path=usr/lib/$(ARCH64)/libldap.so.5
file path=usr/lib/$(ARCH64)/liblgrp.so.1
file path=usr/lib/$(ARCH64)/liblm.so.1
file path=usr/lib/$(ARCH64)/libmail.so.1
file path=usr/lib/$(ARCH64)/libmalloc.so.1
file path=usr/lib/$(ARCH64)/libmapmalloc.so.1
file path=usr/lib/$(ARCH64)/libmenu.so.1
file path=usr/lib/$(ARCH64)/libmtmalloc.so.1
file path=usr/lib/$(ARCH64)/libnls.so.1
file path=usr/lib/$(ARCH64)/libpanel.so.1
file path=usr/lib/$(ARCH64)/libpcidb.so.1
file path=usr/lib/$(ARCH64)/libpkcs11.so.1
file path=usr/lib/$(ARCH64)/libproject.so.1
file path=usr/lib/$(ARCH64)/libraidcfg.so.1
file path=usr/lib/$(ARCH64)/libreparse.so.1
$(i386_ONLY)file path=usr/lib/$(ARCH64)/libsaveargs.so.1
file path=usr/lib/$(ARCH64)/libsched.so.1
file path=usr/lib/$(ARCH64)/libsctp.so.1
file path=usr/lib/$(ARCH64)/libshell.so.1
file path=usr/lib/$(ARCH64)/libsip.so.1
file path=usr/lib/$(ARCH64)/libsldap.so.1
file path=usr/lib/$(ARCH64)/libsmbios.so.1
file path=usr/lib/$(ARCH64)/libsoftcrypto.so.1
file path=usr/lib/$(ARCH64)/libsum.so.1
$(sparc_ONLY)file path=usr/lib/$(ARCH64)/libv12n.so.1
file path=usr/lib/$(ARCH64)/libvolmgt.so.1
file path=usr/lib/$(ARCH64)/liby.so.1
file path=usr/lib/$(ARCH64)/libzoneinfo.so.1
file path=usr/lib/$(ARCH64)/nss_ad.so.1
file path=usr/lib/$(ARCH64)/nss_ldap.so.1
file path=usr/lib/$(ARCH64)/passwdutil.so.1
file path=usr/lib/$(ARCH64)/straddr.so.2
file path=usr/lib/$(ARCH64)/watchmalloc.so.1
file path=usr/lib/0@0.so.1
file path=usr/lib/cfgadm/$(ARCH64)/ib.so.1
file path=usr/lib/cfgadm/$(ARCH64)/pci.so.1
$(i386_ONLY)file path=usr/lib/cfgadm/$(ARCH64)/sata.so.1
file path=usr/lib/cfgadm/$(ARCH64)/scsi.so.1
file path=usr/lib/cfgadm/$(ARCH64)/shp.so.1
file path=usr/lib/cfgadm/$(ARCH64)/usb.so.1
file path=usr/lib/cfgadm/ib.so.1
file path=usr/lib/cfgadm/pci.so.1
$(i386_ONLY)file path=usr/lib/cfgadm/sata.so.1
file path=usr/lib/cfgadm/scsi.so.1
file path=usr/lib/cfgadm/shp.so.1
file path=usr/lib/cfgadm/usb.so.1
file path=usr/lib/extendedFILE.so.1
file path=usr/lib/getloginx.so.1
file path=usr/lib/lib.b mode=0444
file path=usr/lib/libadutils.so.1
file path=usr/lib/libast.so.1
file path=usr/lib/libbsdmalloc.so.1
$(i386_ONLY)file path=usr/lib/libc/libc_hwcap1.so.1 reboot-needed=true
$(i386_ONLY)file path=usr/lib/libc/libc_hwcap2.so.1 reboot-needed=true
$(i386_ONLY)file path=usr/lib/libc/libc_hwcap3.so.1 reboot-needed=true
file path=usr/lib/libcfgadm.so.1
file path=usr/lib/libcmd.so.1
file path=usr/lib/libcommputil.so.1
file path=usr/lib/libcrle.so.1
file path=usr/lib/libcrypt.so.1
file path=usr/lib/libdisasm.so.1
file path=usr/lib/libdll.so.1
file path=usr/lib/libexacct.so.1
file path=usr/lib/libform.so.1
file path=usr/lib/libfstyp.so.1
file path=usr/lib/libhotplug.so.1
file path=usr/lib/libidmap.so.1
file path=usr/lib/libike.so.1
file path=usr/lib/libinetsvc.so.1
file path=usr/lib/libipmi.so.1
file path=usr/lib/libipp.so.1
file path=usr/lib/libipsecutil.so.1
file path=usr/lib/libkvm.so.1
file path=usr/lib/libl.so.1
file path=usr/lib/libldap.so.5
file path=usr/lib/liblgrp.so.1
file path=usr/lib/liblm.so.1
file path=usr/lib/libmail.so.1
file path=usr/lib/libmalloc.so.1
file path=usr/lib/libmapmalloc.so.1
file path=usr/lib/libmenu.so.1
file path=usr/lib/libmtmalloc.so.1
file path=usr/lib/libnls.so.1
file path=usr/lib/libpanel.so.1
file path=usr/lib/libpcidb.so.1
file path=usr/lib/libpkcs11.so.1
file path=usr/lib/libproject.so.1
file path=usr/lib/libraidcfg.so.1
file path=usr/lib/libreparse.so.1
file path=usr/lib/libsched.so.1
file path=usr/lib/libsctp.so.1
file path=usr/lib/libshell.so.1
file path=usr/lib/libsip.so.1
file path=usr/lib/libsldap.so.1
file path=usr/lib/libsmbios.so.1
file path=usr/lib/libsoftcrypto.so.1
file path=usr/lib/libsum.so.1
file path=usr/lib/libsys.so.1
$(sparc_ONLY)file path=usr/lib/libv12n.so.1
file path=usr/lib/libvolmgt.so.1
file path=usr/lib/libwrap.so.1.0
file path=usr/lib/liby.so.1
file path=usr/lib/libzoneinfo.so.1
file path=usr/lib/nss_ad.so.1
file path=usr/lib/nss_ldap.so.1
file path=usr/lib/passwdutil.so.1
file path=usr/lib/python2.6/vendor-packages/solaris/__init__.py
file path=usr/lib/python2.6/vendor-packages/solaris/__init__.pyc
file path=usr/lib/python2.6/vendor-packages/solaris/misc.so
file path=usr/lib/raidcfg/$(ARCH64)/mpt.so.1
file path=usr/lib/raidcfg/mpt.so.1
file path=usr/lib/scsi/$(ARCH64)/libscsi.so.1
file path=usr/lib/scsi/$(ARCH64)/libses.so.1
file path=usr/lib/scsi/$(ARCH64)/libsmp.so.1
file path=usr/lib/scsi/libscsi.so.1
```

```
file path=usr/lib/scsi/libses.so.1
file path=usr/lib/scsi/libsmp.so.1
file path=usr/lib/scsi/plugins/scsi/engines/$(ARCH64)/uscsi.so
file path=usr/lib/scsi/plugins/scsi/engines/uscsi.so
file path=usr/lib/scsi/plugins/ses/framework/$(ARCH64)/libses.so
file path=usr/lib/scsi/plugins/ses/framework/$(ARCH64)/ses2.so
file path=usr/lib/scsi/plugins/ses/framework/libses.so
file path=usr/lib/scsi/plugins/ses/framework/ses2.so
file path=usr/lib/scsi/plugins/smp/engine/$(ARCH64)/usmp.so
file path=usr/lib/scsi/plugins/smp/engine/usmp.so
file path=usr/lib/scsi/plugins/smp/framework/$(ARCH64)/sas2.so
file path=usr/lib/scsi/plugins/smp/framework/sas2.so
file path=usr/lib/security/$(ARCH64)/crypt_bsdbf.so.1
file path=usr/lib/security/$(ARCH64)/crypt_bsdmd5.so.1
file path=usr/lib/security/$(ARCH64)/crypt_sha256.so.1
file path=usr/lib/security/$(ARCH64)/crypt_sha512.so.1
file path=usr/lib/security/$(ARCH64)/crypt_sunmd5.so.1
file path=usr/lib/security/$(ARCH64)/pam_allow.so.1
file path=usr/lib/security/$(ARCH64)/pam_authtok_check.so.1
file path=usr/lib/security/$(ARCH64)/pam_authtok_get.so.1
file path=usr/lib/security/$(ARCH64)/pam_authtok_store.so.1
file path=usr/lib/security/$(ARCH64)/pam_deny.so.1
file path=usr/lib/security/$(ARCH64)/pam_dhkeys.so.1
file path=usr/lib/security/$(ARCH64)/pam_dial_auth.so.1
file path=usr/lib/security/$(ARCH64)/pam_ldap.so.1
file path=usr/lib/security/$(ARCH64)/pam_list.so.1
file path=usr/lib/security/$(ARCH64)/pam_passwd_auth.so.1
file path=usr/lib/security/$(ARCH64)/pam_rhosts_auth.so.1
file path=usr/lib/security/$(ARCH64)/pam_roles.so.1
file path=usr/lib/security/$(ARCH64)/pam_sample.so.1
file path=usr/lib/security/$(ARCH64)/pam_tsol_account.so.1
file path=usr/lib/security/$(ARCH64)/pam_unix_account.so.1
file path=usr/lib/security/$(ARCH64)/pam_unix_auth.so.1
file path=usr/lib/security/$(ARCH64)/pam_unix_cred.so.1
file path=usr/lib/security/$(ARCH64)/pam_unix_session.so.1
file path=usr/lib/security/$(ARCH64)/pkcs11_kernel.so.1
file path=usr/lib/security/$(ARCH64)/pkcs11_softtoken.so.1
file path=usr/lib/security/$(ARCH64)/pkcs11_tpm.so.1
file path=usr/lib/security/audit_binfile.so.1
file path=usr/lib/security/audit_remote.so.1
file path=usr/lib/security/audit_syslog.so.1
file path=usr/lib/security/crypt_bsdbf.so.1
file path=usr/lib/security/crypt_bsdmd5.so.1
file path=usr/lib/security/crypt_sha256.so.1
file path=usr/lib/security/crypt_sha512.so.1
file path=usr/lib/security/crypt_sunmd5.so.1
file path=usr/lib/security/pam_allow.so.1
file path=usr/lib/security/pam_authtok_check.so.1
file path=usr/lib/security/pam_authtok_get.so.1
file path=usr/lib/security/pam_authtok_store.so.1
file path=usr/lib/security/pam_deny.so.1
file path=usr/lib/security/pam_dhkeys.so.1
file path=usr/lib/security/pam_dial_auth.so.1
file path=usr/lib/security/pam_ldap.so.1
file path=usr/lib/security/pam_list.so.1
file path=usr/lib/security/pam_passwd_auth.so.1
file path=usr/lib/security/pam_rhosts_auth.so.1
file path=usr/lib/security/pam_roles.so.1
file path=usr/lib/security/pam_sample.so.1
file path=usr/lib/security/pam_tsol_account.so.1
file path=usr/lib/security/pam_unix_account.so.1
file path=usr/lib/security/pam_unix_auth.so.1
file path=usr/lib/security/pam_unix_cred.so.1
file path=usr/lib/security/pam_unix_session.so.1
file path=usr/lib/security/pkcs11_kernel.so.1
file path=usr/lib/security/pkcs11_softtoken.so.1
file path=usr/lib/security/pkcs11_tpm.so.1
file path=usr/lib/straddr.so.2
file path=usr/lib/watchmalloc.so.1
# XXX: Obsoleted by open i18n?
file path=usr/xpg4/lib/$(ARCH64)/libcurses.so.1
file path=usr/xpg4/lib/$(ARCH64)/libcurses.so.2
file path=usr/xpg4/lib/libcurses.so.1
file path=usr/xpg4/lib/libcurses.so.2
legacy pkg=SUNWcsl \
    desc="core shared libraries for a specific instruction-set architecture" \
    name="Core Solaris, (Shared Libs)"
legacy pkg=SUNWcslr \
    desc="core software for a specific instruction-set architecture" \
    name="Core Solaris Libraries (Root)"
license cr_Sun license=cr_Sun
license lic_CDDL license=lic_CDDL
license lic_OSBL license=lic_OSBL
license lic_OSBL_preamble license=lic_OSBL_preamble
# libwrap is part of tcp wrappers along with tcpd
license usr/src/cmd/tcpd/THIRDPARTYLICENSE \
    license=usr/src/cmd/tcpd/THIRDPARTYLICENSE
license usr/src/common/crypto/THIRDPARTYLICENSE.cryptogams \
    license=usr/src/common/crypto/THIRDPARTYLICENSE.cryptogams
license usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.gladman \
    license=usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.gladman
license usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.openssl \
    license=usr/src/common/crypto/aes/amd64/THIRDPARTYLICENSE.openssl
license usr/src/common/crypto/ecc/THIRDPARTYLICENSE \
    license=usr/src/common/crypto/ecc/THIRDPARTYLICENSE
license usr/src/common/crypto/md5/amd64/THIRDPARTYLICENSE \
    license=usr/src/common/crypto/md5/amd64/THIRDPARTYLICENSE
license usr/src/common/mpi/THIRDPARTYLICENSE \
    license=usr/src/common/mpi/THIRDPARTYLICENSE
license usr/src/lib/libast/THIRDPARTYLICENSE \
    license=usr/src/lib/libast/THIRDPARTYLICENSE
license usr/src/lib/libbsdmalloc/THIRDPARTYLICENSE \
    license=usr/src/lib/libbsdmalloc/THIRDPARTYLICENSE
license usr/src/lib/libc/THIRDPARTYLICENSE \
    license=usr/src/lib/libc/THIRDPARTYLICENSE
license usr/src/lib/libcmd/THIRDPARTYLICENSE \
    license=usr/src/lib/libcmd/THIRDPARTYLICENSE
license usr/src/lib/libdll/THIRDPARTYLICENSE \
    license=usr/src/lib/libdll/THIRDPARTYLICENSE
license usr/src/lib/libinetutil/common/THIRDPARTYLICENSE \
    license=usr/src/lib/libinetutil/common/THIRDPARTYLICENSE
license usr/src/lib/libkmf/THIRDPARTYLICENSE \
    license=usr/src/lib/libkmf/THIRDPARTYLICENSE
license usr/src/lib/libldap5/THIRDPARTYLICENSE \
    license=usr/src/lib/libldap5/THIRDPARTYLICENSE
license usr/src/lib/libmp/common/THIRDPARTYLICENSE \
    license=usr/src/lib/libmp/common/THIRDPARTYLICENSE
license usr/src/lib/libresolv/THIRDPARTYLICENSE \
    license=usr/src/lib/libresolv/THIRDPARTYLICENSE
license usr/src/lib/libresolv2/THIRDPARTYLICENSE \
    license=usr/src/lib/libresolv2/THIRDPARTYLICENSE
license usr/src/lib/libshell/THIRDPARTYLICENSE \
    license=usr/src/lib/libshell/THIRDPARTYLICENSE
license usr/src/lib/libsum/THIRDPARTYLICENSE \
    license=usr/src/lib/libsum/THIRDPARTYLICENSE
license usr/src/lib/pam_modules/authtok_check/THIRDPARTYLICENSE \
    license=usr/src/lib/pam_modules/authtok_check/THIRDPARTYLICENSE
license usr/src/lib/passwdutil/THIRDPARTYLICENSE \
    license=usr/src/lib/passwdutil/THIRDPARTYLICENSE
license usr/src/lib/pkcs11/pkcs11_tpm/THIRDPARTYLICENSE \
    license=usr/src/lib/pkcs11/pkcs11_tpm/THIRDPARTYLICENSE
license usr/src/uts/common/sys/THIRDPARTYLICENSE.unicode \
```

```
    license=usr/src/uts/common/sys/THIRDPARTYLICENSE.unicode
link path=lib/$(ARCH64)/libadm.so target=libadm.so.1
link path=lib/$(ARCH64)/libaio.so target=libaio.so.1
link path=lib/$(ARCH64)/libbsm.so target=libbsm.so.1
link path=lib/$(ARCH64)/libc.so reboot-needed=true target=libc.so.1
link path=lib/$(ARCH64)/libc_db.so target=libc_db.so.1
link path=lib/$(ARCH64)/libcontract.so target=libcontract.so.1
link path=lib/$(ARCH64)/libcryptoutil.so target=libcryptoutil.so.1
link path=lib/$(ARCH64)/libctf.so target=libctf.so.1
link path=lib/$(ARCH64)/libcurses.so target=libcurses.so.1
link path=lib/$(ARCH64)/libdevice.so target=libdevice.so.1
link path=lib/$(ARCH64)/libdevid.so target=libdevid.so.1
link path=lib/$(ARCH64)/libdevinfo.so target=libdevinfo.so.1
link path=lib/$(ARCH64)/libdl.so target=libdl.so.1
link path=lib/$(ARCH64)/libdladm.so target=libdladm.so.1
link path=lib/$(ARCH64)/libdlpi.so target=libdlpi.so.1
link path=lib/$(ARCH64)/libdoor.so target=libdoor.so.1
link path=lib/$(ARCH64)/libefi.so target=libefi.so.1
link path=lib/$(ARCH64)/libelf.so target=libelf.so.1
$(i386_ONLY)link path=lib/$(ARCH64)/libfdisk.so target=libfdisk.so.1
link path=lib/$(ARCH64)/libgen.so target=libgen.so.1
link path=lib/$(ARCH64)/libintl.so target=libintl.so.1
link path=lib/$(ARCH64)/libkmf.so target=libkmf.so.1
link path=lib/$(ARCH64)/libkmfberder.so target=libkmfberder.so.1
link path=lib/$(ARCH64)/libkstat.so target=libkstat.so.1
link path=lib/$(ARCH64)/libmd.so target=libmd.so.1
link path=lib/$(ARCH64)/libmd5.so target=libmd5.so.1
link path=lib/$(ARCH64)/libmp.so target=libmp.so.2
link path=lib/$(ARCH64)/libnsl.so target=libnsl.so.1
link path=lib/$(ARCH64)/libnvpair.so target=libnvpair.so.1
link path=lib/$(ARCH64)/libpam.so target=libpam.so.1
link path=lib/$(ARCH64)/libposix4.so target=libposix4.so.1
link path=lib/$(ARCH64)/libposix4.so.1 target=librt.so.1
link path=lib/$(ARCH64)/libproc.so target=libproc.so.1
link path=lib/$(ARCH64)/libpthread.so target=libpthread.so.1
link path=lib/$(ARCH64)/librcm.so target=librcm.so.1
link path=lib/$(ARCH64)/libresolv.so target=libresolv.so.2
link path=lib/$(ARCH64)/librestart.so target=librestart.so.1
link path=lib/$(ARCH64)/librpcsvc.so target=librpcsvc.so.1
link path=lib/$(ARCH64)/librt.so target=librt.so.1
link path=lib/$(ARCH64)/librtld_db.so target=librtld_db.so.1
link path=lib/$(ARCH64)/libscf.so target=libscf.so.1
link path=lib/$(ARCH64)/libsec.so target=libsec.so.1
link path=lib/$(ARCH64)/libsecdb.so target=libsecdb.so.1
link path=lib/$(ARCH64)/libsendfile.so target=libsendfile.so.1
link path=lib/$(ARCH64)/libsocket.so target=libsocket.so.1
link path=lib/$(ARCH64)/libsysevent.so target=libsysevent.so.1
link path=lib/$(ARCH64)/libtermcap.so target=libtermcap.so.1
link path=lib/$(ARCH64)/libtermlib.so target=libtermlib.so.1
link path=lib/$(ARCH64)/libtermlib.so.1 target=libcurses.so.1
link path=lib/$(ARCH64)/libthread.so target=libthread.so.1
link path=lib/$(ARCH64)/libthread_db.so target=libc_db.so.1
link path=lib/$(ARCH64)/libthread_db.so.1 target=libc_db.so.1
link path=lib/$(ARCH64)/libtsnet.so target=libtsnet.so.1
link path=lib/$(ARCH64)/libtsol.so target=libtsol.so.2
link path=lib/$(ARCH64)/libumem.so target=libumem.so.1
link path=lib/$(ARCH64)/libuuid.so target=libuuid.so.1
link path=lib/$(ARCH64)/libuutil.so target=libuutil.so.1
link path=lib/$(ARCH64)/libw.so target=libw.so.1
link path=lib/$(ARCH64)/libxnet.so target=libxnet.so.1
link path=lib/32 target=.
link path=lib/64 target=$(ARCH64)
link path=lib/crypto/32 target=.
link path=lib/crypto/64 target=$(ARCH64)
link path=lib/libadm.so target=libadm.so.1
link path=lib/libaio.so target=libaio.so.1
link path=lib/libbsm.so target=libbsm.so.1
link path=lib/libc.so target=libc.so.1
link path=lib/libc_db.so target=libc_db.so.1
link path=lib/libcontract.so target=libcontract.so.1
link path=lib/libcryptoutil.so target=./libcryptoutil.so.1
link path=lib/libctf.so target=libctf.so.1
link path=lib/libcurses.so target=libcurses.so.1
link path=lib/libdevice.so target=libdevice.so.1
link path=lib/libdevid.so target=libdevid.so.1
link path=lib/libdevinfo.so target=libdevinfo.so.1
link path=lib/libdl.so target=libdl.so.1
link path=lib/libdladm.so target=libdladm.so.1
link path=lib/libdlpi.so target=libdlpi.so.1
link path=lib/libdoor.so target=libdoor.so.1
link path=lib/libefi.so target=libefi.so.1
link path=lib/libelf.so target=libelf.so.1
link path=lib/libelfsign.so target=libelfsign.so.1
$(i386_ONLY)link path=lib/libfdisk.so target=libfdisk.so.1
link path=lib/libgen.so target=libgen.so.1
link path=lib/libintl.so target=libintl.so.1
link path=lib/libipmp.so target=./libipmp.so.1
link path=lib/libkmf.so target=libkmf.so.1
link path=lib/libkmfberder.so target=libkmfberder.so.1
link path=lib/libkstat.so target=libkstat.so.1
link path=lib/libmd.so target=libmd.so.1
link path=lib/libmd5.so target=libmd5.so.1
link path=lib/libmp.so target=libmp.so.2
link path=lib/libnsl.so target=libnsl.so.1
link path=lib/libnvpair.so target=libnvpair.so.1
link path=lib/libnwam.so target=libnwam.so.1
link path=lib/libpam.so target=libpam.so.1
link path=lib/libposix4.so target=libposix4.so.1
link path=lib/libposix4.so.1 target=librt.so.1
link path=lib/libproc.so target=libproc.so.1
link path=lib/libpthread.so target=libpthread.so.1
link path=lib/librcm.so target=./librcm.so.1
link path=lib/libresolv.so target=libresolv.so.2
link path=lib/librpcsvc.so target=librpcsvc.so.1
link path=lib/librt.so target=librt.so.1
link path=lib/librtld_db.so target=librtld_db.so.1
link path=lib/libscf.so target=libscf.so.1
link path=lib/libsec.so target=libsec.so.1
link path=lib/libsecdb.so target=libsecdb.so.1
link path=lib/libsendfile.so target=libsendfile.so.1
link path=lib/libsocket.so target=libsocket.so.1
link path=lib/libsysevent.so target=./libsysevent.so.1
link path=lib/libtermcap.so target=libtermcap.so.1
link path=lib/libtermlib.so target=libtermlib.so.1
link path=lib/libtermlib.so.1 target=libcurses.so.1
link path=lib/libthread.so target=libthread.so.1
link path=lib/libthread_db.so target=libc_db.so.1
link path=lib/libthread_db.so.1 target=libc_db.so.1
link path=lib/libtsol.so target=libtsol.so.2
link path=lib/libumem.so target=libumem.so.1
link path=lib/libuuid.so target=libuuid.so.1
link path=lib/libw.so target=libw.so.1
link path=lib/libxnet.so target=libxnet.so.1
link path=lib/secure/32 target=.
link path=lib/secure/64 target=$(ARCH64)
link path=usr/ccs/lib/$(ARCH64)/libcurses.so \
    target=../../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/ccs/lib/$(ARCH64)/libform.so \
    target=../../../lib/$(ARCH64)/libform.so.1
link path=usr/ccs/lib/$(ARCH64)/libgen.so \
    target=../../../../lib/$(ARCH64)/libgen.so.1
link path=usr/ccs/lib/$(ARCH64)/libl.so \
```

```
    target=../../../lib/$(ARCH64)/libl.so.1
link path=usr/ccs/lib/$(ARCH64)/libmalloc.so \
    target=../../../lib/$(ARCH64)/libmalloc.so.1
link path=usr/ccs/lib/$(ARCH64)/libmenu.so \
    target=../../../lib/$(ARCH64)/libmenu.so.1
link path=usr/ccs/lib/$(ARCH64)/libpanel.so \
    target=../../../lib/$(ARCH64)/libpanel.so.1
link path=usr/ccs/lib/$(ARCH64)/libtermcap.so \
    target=../../../../lib/$(ARCH64)/libtermcap.so.1
link path=usr/ccs/lib/$(ARCH64)/libtermlib.so \
    target=../../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/ccs/lib/$(ARCH64)/liby.so \
    target=../../../lib/$(ARCH64)/liby.so.1
link path=usr/ccs/lib/libcurses.so target=../../../lib/libcurses.so.1
link path=usr/ccs/lib/libform.so target=../../lib/libform.so.1
link path=usr/ccs/lib/libgen.so target=../../../lib/libgen.so.1
link path=usr/ccs/lib/libl.so target=../../lib/libl.so.1
link path=usr/ccs/lib/libmalloc.so target=../../lib/libmalloc.so.1
link path=usr/ccs/lib/libmenu.so target=../../lib/libmenu.so.1
link path=usr/ccs/lib/libpanel.so target=../../lib/libpanel.so.1
link path=usr/ccs/lib/libtermcap.so target=../../../lib/libtermcap.so.1
link path=usr/ccs/lib/libtermlib.so target=../../../lib/libcurses.so.1
link path=usr/ccs/lib/liby.so target=../../lib/liby.so.1
link path=usr/lib/$(ARCH64)/libadm.so \
    target=../../../lib/$(ARCH64)/libadm.so.1
link path=usr/lib/$(ARCH64)/libadm.so.1 \
    target=../../../lib/$(ARCH64)/libadm.so.1
link path=usr/lib/$(ARCH64)/libadutils.so target=./libadutils.so.1
link path=usr/lib/$(ARCH64)/libaio.so \
    target=../../../lib/$(ARCH64)/libaio.so.1
link path=usr/lib/$(ARCH64)/libaio.so.1 \
    target=../../../lib/$(ARCH64)/libaio.so.1
link path=usr/lib/$(ARCH64)/libavl.so.1 \
    target=../../../lib/$(ARCH64)/libavl.so.1
link path=usr/lib/$(ARCH64)/libbsdmalloc.so target=libbsdmalloc.so.1
link path=usr/lib/$(ARCH64)/libbsm.so \
    target=../../../lib/$(ARCH64)/libbsm.so.1
link path=usr/lib/$(ARCH64)/libbsm.so.1 \
    target=../../../lib/$(ARCH64)/libbsm.so.1
link path=usr/lib/$(ARCH64)/libc.so target=../../../lib/$(ARCH64)/libc.so.1
link path=usr/lib/$(ARCH64)/libc.so.1 target=../../../lib/$(ARCH64)/libc.so.1
link path=usr/lib/$(ARCH64)/libc_db.so \
    target=../../../lib/$(ARCH64)/libc_db.so.1
link path=usr/lib/$(ARCH64)/libc_db.so.1 \
    target=../../../lib/$(ARCH64)/libc_db.so.1
link path=usr/lib/$(ARCH64)/libcfgadm.so target=libcfgadm.so.1
link path=usr/lib/$(ARCH64)/libcmd.so target=libcmd.so.1
link path=usr/lib/$(ARCH64)/libcmdutils.so.1 \
    target=../../../lib/$(ARCH64)/libcmdutils.so.1
link path=usr/lib/$(ARCH64)/libcommputil.so target=libcommputil.so.1
link path=usr/lib/$(ARCH64)/libcontract.so \
    target=../../../lib/$(ARCH64)/libcontract.so.1
link path=usr/lib/$(ARCH64)/libcontract.so.1 \
    target=../../../lib/$(ARCH64)/libcontract.so.1
link path=usr/lib/$(ARCH64)/libcrypt.so target=./libcrypt.so.1
link path=usr/lib/$(ARCH64)/libcrypt_d.so target=./libcrypt.so
link path=usr/lib/$(ARCH64)/libcrypt_d.so.1 target=./libcrypt.so.1
link path=usr/lib/$(ARCH64)/libcrypt_i.so target=./libcrypt.so
link path=usr/lib/$(ARCH64)/libcrypt_i.so.1 target=./libcrypt.so.1
link path=usr/lib/$(ARCH64)/libctf.so \
    target=../../../lib/$(ARCH64)/libctf.so.1
link path=usr/lib/$(ARCH64)/libctf.so.1 \
    target=../../../lib/$(ARCH64)/libctf.so.1
link path=usr/lib/$(ARCH64)/libcurses.so \
    target=../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/lib/$(ARCH64)/libcurses.so.1 \
    target=../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/lib/$(ARCH64)/libdevice.so \
    target=../../../lib/$(ARCH64)/libdevice.so.1
link path=usr/lib/$(ARCH64)/libdevice.so.1 \
    target=../../../lib/$(ARCH64)/libdevice.so.1
link path=usr/lib/$(ARCH64)/libdevid.so \
    target=../../../lib/$(ARCH64)/libdevid.so.1
link path=usr/lib/$(ARCH64)/libdevid.so.1 \
    target=../../../lib/$(ARCH64)/libdevid.so.1
link path=usr/lib/$(ARCH64)/libdevinfo.so \
    target=../../../lib/$(ARCH64)/libdevinfo.so.1
link path=usr/lib/$(ARCH64)/libdevinfo.so.1 \
    target=../../../lib/$(ARCH64)/libdevinfo.so.1
link path=usr/lib/$(ARCH64)/libdhcputil.so.1 \
    target=../../../lib/$(ARCH64)/libdhcputil.so.1
link path=usr/lib/$(ARCH64)/libdisasm.so target=libdisasm.so.1
link path=usr/lib/$(ARCH64)/libdl.so target=../../../lib/$(ARCH64)/libdl.so.1
link path=usr/lib/$(ARCH64)/libdl.so.1 \
    target=../../../lib/$(ARCH64)/libdl.so.1
link path=usr/lib/$(ARCH64)/libdlpi.so \
    target=../../../lib/$(ARCH64)/libdlpi.so.1
link path=usr/lib/$(ARCH64)/libdlpi.so.1 \
    target=../../../lib/$(ARCH64)/libdlpi.so.1
link path=usr/lib/$(ARCH64)/libdoor.so \
    target=../../../lib/$(ARCH64)/libdoor.so.1
link path=usr/lib/$(ARCH64)/libdoor.so.1 \
    target=../../../lib/$(ARCH64)/libdoor.so.1
link path=usr/lib/$(ARCH64)/libefi.so \
    target=../../../lib/$(ARCH64)/libefi.so.1
link path=usr/lib/$(ARCH64)/libefi.so.1 \
    target=../../../lib/$(ARCH64)/libefi.so.1
link path=usr/lib/$(ARCH64)/libelf.so \
    target=../../../lib/$(ARCH64)/libelf.so.1
link path=usr/lib/$(ARCH64)/libelf.so.1 \
    target=../../../lib/$(ARCH64)/libelf.so.1
link path=usr/lib/$(ARCH64)/libexacct.so target=libexacct.so.1
$(i386_ONLY)link path=usr/lib/$(ARCH64)/libfdisk.so \
    target=../../../lib/$(ARCH64)/libfdisk.so.1
$(i386_ONLY)link path=usr/lib/$(ARCH64)/libfdisk.so.1 \
    target=../../../lib/$(ARCH64)/libfdisk.so.1
link path=usr/lib/$(ARCH64)/libform.so target=libform.so.1
link path=usr/lib/$(ARCH64)/libfstyp.so target=./libfstyp.so.1
link path=usr/lib/$(ARCH64)/libgen.so \
    target=../../../lib/$(ARCH64)/libgen.so.1
link path=usr/lib/$(ARCH64)/libgen.so.1 \
    target=../../../lib/$(ARCH64)/libgen.so.1
link path=usr/lib/$(ARCH64)/libhotplug.so target=libhotplug.so.1
link path=usr/lib/$(ARCH64)/libidmap.so target=./libidmap.so.1
link path=usr/lib/$(ARCH64)/libinetutil.so.1 \
    target=../../../lib/$(ARCH64)/libinetutil.so.1
link path=usr/lib/$(ARCH64)/libintl.so \
    target=../../../lib/$(ARCH64)/libintl.so.1
link path=usr/lib/$(ARCH64)/libintl.so.1 \
    target=../../../lib/$(ARCH64)/libintl.so.1
link path=usr/lib/$(ARCH64)/libipmi.so target=./libipmi.so.1
link path=usr/lib/$(ARCH64)/libipp.so target=libipp.so.1
link path=usr/lib/$(ARCH64)/libkstat.so \
    target=../../../lib/$(ARCH64)/libkstat.so.1
link path=usr/lib/$(ARCH64)/libkstat.so.1 \
    target=../../../lib/$(ARCH64)/libkstat.so.1
link path=usr/lib/$(ARCH64)/libkvm.so target=libkvm.so.1
link path=usr/lib/$(ARCH64)/libl.so target=libl.so.1
link path=usr/lib/$(ARCH64)/libldap.so target=libldap.so.5
link path=usr/lib/$(ARCH64)/liblddbg.so.4 \
    target=../../../lib/$(ARCH64)/liblddbg.so.4
link path=usr/lib/$(ARCH64)/liblgrp.so target=liblgrp.so.1
```

```
link path=usr/lib/$(ARCH64)/liblm.so target=liblm.so.1
link path=usr/lib/$(ARCH64)/libmail.so target=libmail.so.1
link path=usr/lib/$(ARCH64)/libmalloc.so target=libmalloc.so.1
link path=usr/lib/$(ARCH64)/libmapmalloc.so target=libmapmalloc.so.1
link path=usr/lib/$(ARCH64)/libmd.so target=../../../lib/$(ARCH64)/libmd.so.1
link path=usr/lib/$(ARCH64)/libmd.so.1 \
    target=../../../lib/$(ARCH64)/libmd.so.1
link path=usr/lib/$(ARCH64)/libmd5.so \
    target=../../../lib/$(ARCH64)/libmd5.so.1
link path=usr/lib/$(ARCH64)/libmd5.so.1 \
    target=../../../lib/$(ARCH64)/libmd5.so.1
link path=usr/lib/$(ARCH64)/libmenu.so target=libmenu.so.1
link path=usr/lib/$(ARCH64)/libmp.so target=../../../lib/$(ARCH64)/libmp.so.2
link path=usr/lib/$(ARCH64)/libmp.so.2 \
    target=../../../lib/$(ARCH64)/libmp.so.2
link path=usr/lib/$(ARCH64)/libmtmalloc.so target=libmtmalloc.so.1
link path=usr/lib/$(ARCH64)/libnls.so target=libnls.so.1
link path=usr/lib/$(ARCH64)/libnsl.so \
    target=../../../lib/$(ARCH64)/libnsl.so.1
link path=usr/lib/$(ARCH64)/libnsl.so.1 \
    target=../../../lib/$(ARCH64)/libnsl.so.1
link path=usr/lib/$(ARCH64)/libnvpair.so \
    target=../../../lib/$(ARCH64)/libnvpair.so.1
link path=usr/lib/$(ARCH64)/libnvpair.so.1 \
    target=../../../lib/$(ARCH64)/libnvpair.so.1
link path=usr/lib/$(ARCH64)/libpam.so \
    target=../../../lib/$(ARCH64)/libpam.so.1
link path=usr/lib/$(ARCH64)/libpam.so.1 \
    target=../../../lib/$(ARCH64)/libpam.so.1
link path=usr/lib/$(ARCH64)/libpanel.so target=libpanel.so.1
link path=usr/lib/$(ARCH64)/libpkcs11.so target=libpkcs11.so.1
link path=usr/lib/$(ARCH64)/libposix4.so \
    target=../../../lib/$(ARCH64)/librt.so.1
link path=usr/lib/$(ARCH64)/libposix4.so.1 \
    target=../../../lib/$(ARCH64)/librt.so.1
link path=usr/lib/$(ARCH64)/libproc.so \
    target=../../../lib/$(ARCH64)/libproc.so.1
link path=usr/lib/$(ARCH64)/libproc.so.1 \
    target=../../../lib/$(ARCH64)/libproc.so.1
link path=usr/lib/$(ARCH64)/libproject.so target=libproject.so.1
link path=usr/lib/$(ARCH64)/libpthread.so \
    target=../../../lib/$(ARCH64)/libpthread.so.1
link path=usr/lib/$(ARCH64)/libpthread.so.1 \
    target=../../../lib/$(ARCH64)/libpthread.so.1
link path=usr/lib/$(ARCH64)/librcm.so \
    target=../../../lib/$(ARCH64)/librcm.so.1
link path=usr/lib/$(ARCH64)/librcm.so.1 \
    target=../../../lib/$(ARCH64)/librcm.so.1
link path=usr/lib/$(ARCH64)/libreparse.so target=libreparse.so.1
link path=usr/lib/$(ARCH64)/libresolv.so \
    target=../../../lib/$(ARCH64)/libresolv.so.2
link path=usr/lib/$(ARCH64)/libresolv.so.2 \
    target=../../../lib/$(ARCH64)/libresolv.so.2
$(i386_ONLY)link path=usr/lib/$(ARCH64)/librestart.so \
    target=../../../lib/$(ARCH64)/librestart.so.1
link path=usr/lib/$(ARCH64)/librestart.so.1 \
    target=../../../lib/$(ARCH64)/librestart.so.1
link path=usr/lib/$(ARCH64)/librpcsvc.so \
    target=../../../lib/$(ARCH64)/librpcsvc.so.1
link path=usr/lib/$(ARCH64)/librpcsvc.so.1 \
    target=../../../lib/$(ARCH64)/librpcsvc.so.1
link path=usr/lib/$(ARCH64)/librt.so target=../../../lib/$(ARCH64)/librt.so.1
link path=usr/lib/$(ARCH64)/librt.so.1 \
    target=../../../lib/$(ARCH64)/librt.so.1
link path=usr/lib/$(ARCH64)/librtld.so.1 \
    target=../../../lib/$(ARCH64)/librtld.so.1
```

```
link path=usr/lib/$(ARCH64)/librtld_db.so \
    target=../../../lib/$(ARCH64)/librtld_db.so.1
link path=usr/lib/$(ARCH64)/librtld_db.so.1 \
    target=../../../lib/$(ARCH64)/librtld_db.so.1
link path=usr/lib/$(ARCH64)/libscf.so \
    target=../../../lib/$(ARCH64)/libscf.so.1
link path=usr/lib/$(ARCH64)/libscf.so.1 \
    target=../../../lib/$(ARCH64)/libscf.so.1
link path=usr/lib/$(ARCH64)/libsched.so target=libsched.so.1
link path=usr/lib/$(ARCH64)/libsctp.so target=./libsctp.so.1
link path=usr/lib/$(ARCH64)/libsec.so \
    target=../../../lib/$(ARCH64)/libsec.so.1
link path=usr/lib/$(ARCH64)/libsec.so.1 \
    target=../../../lib/$(ARCH64)/libsec.so.1
link path=usr/lib/$(ARCH64)/libsecdb.so \
    target=../../../lib/$(ARCH64)/libsecdb.so.1
link path=usr/lib/$(ARCH64)/libsecdb.so.1 \
    target=../../../lib/$(ARCH64)/libsecdb.so.1
link path=usr/lib/$(ARCH64)/libsendfile.so \
    target=../../../lib/$(ARCH64)/libsendfile.so.1
link path=usr/lib/$(ARCH64)/libsendfile.so.1 \
    target=../../../lib/$(ARCH64)/libsendfile.so.1
link path=usr/lib/$(ARCH64)/libsip.so target=./libsip.so.1
link path=usr/lib/$(ARCH64)/libsldap.so target=libsldap.so.1
link path=usr/lib/$(ARCH64)/libsmbios.so target=libsmbios.so.1
link path=usr/lib/$(ARCH64)/libsocket.so \
    target=../../../lib/$(ARCH64)/libsocket.so.1
link path=usr/lib/$(ARCH64)/libsocket.so.1 \
    target=../../../lib/$(ARCH64)/libsocket.so.1
link path=usr/lib/$(ARCH64)/libsoftcrypto.so target=./libsoftcrypto.so.1
link path=usr/lib/$(ARCH64)/libsysevent.so \
    target=../../../lib/$(ARCH64)/libsysevent.so.1
link path=usr/lib/$(ARCH64)/libsysevent.so.1 \
    target=../../../lib/$(ARCH64)/libsysevent.so.1
link path=usr/lib/$(ARCH64)/libtermcap.so \
    target=../../../lib/$(ARCH64)/libtermcap.so.1
link path=usr/lib/$(ARCH64)/libtermcap.so.1 \
    target=../../../lib/$(ARCH64)/libtermcap.so.1
link path=usr/lib/$(ARCH64)/libtermlib.so \
    target=../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/lib/$(ARCH64)/libtermlib.so.1 \
    target=../../../lib/$(ARCH64)/libcurses.so.1
link path=usr/lib/$(ARCH64)/libthread.so \
    target=../../../lib/$(ARCH64)/libthread.so.1
link path=usr/lib/$(ARCH64)/libthread.so.1 \
    target=../../../lib/$(ARCH64)/libthread.so.1
link path=usr/lib/$(ARCH64)/libthread_db.so \
    target=../../../lib/$(ARCH64)/libc_db.so.1
link path=usr/lib/$(ARCH64)/libthread_db.so.1 \
    target=../../../lib/$(ARCH64)/libc_db.so.1
link path=usr/lib/$(ARCH64)/libtsnet.so \
    target=../../../lib/$(ARCH64)/libtsnet.so.1
link path=usr/lib/$(ARCH64)/libtsnet.so.1 \
    target=../../../lib/$(ARCH64)/libtsnet.so.1
link path=usr/lib/$(ARCH64)/libtsol.so \
    target=../../../lib/$(ARCH64)/libtsol.so.2
link path=usr/lib/$(ARCH64)/libtsol.so.2 \
    target=../../../lib/$(ARCH64)/libtsol.so.2
link path=usr/lib/$(ARCH64)/libumem.so \
    target=../../../lib/$(ARCH64)/libumem.so.1
link path=usr/lib/$(ARCH64)/libumem.so.1 \
    target=../../../lib/$(ARCH64)/libumem.so.1
link path=usr/lib/$(ARCH64)/libuuid.so \
    target=../../../lib/$(ARCH64)/libuuid.so.1
link path=usr/lib/$(ARCH64)/libuuid.so.1 \
    target=../../../lib/$(ARCH64)/libuuid.so.1
```

```
$(i386_ONLY)link path=usr/lib/$(ARCH64)/libuutil.so \
    target=../../../lib/$(ARCH64)/libuutil.so.1
link path=usr/lib/$(ARCH64)/libuutil.so.1 \
    target=../../../lib/$(ARCH64)/libuutil.so.1
$(sparc_ONLY)link path=usr/lib/$(ARCH64)/libv12n.so target=./libv12n.so.1
link path=usr/lib/$(ARCH64)/libvolmgt.so target=libvolmgt.so.1
link path=usr/lib/$(ARCH64)/libw.so target=../../../lib/$(ARCH64)/libw.so.1
link path=usr/lib/$(ARCH64)/libw.so.1 target=../../../lib/$(ARCH64)/libw.so.1
link path=usr/lib/$(ARCH64)/libxnet.so \
    target=../../../lib/$(ARCH64)/libxnet.so.1
link path=usr/lib/$(ARCH64)/libxnet.so.1 \
    target=../../../lib/$(ARCH64)/libxnet.so.1
link path=usr/lib/$(ARCH64)/liby.so target=liby.so.1
link path=usr/lib/$(ARCH64)/libzoneinfo.so target=./libzoneinfo.so.1
link path=usr/lib/$(ARCH64)/nss_compat.so.1 \
    target=../../../lib/$(ARCH64)/nss_compat.so.1
link path=usr/lib/$(ARCH64)/nss_dns.so.1 \
    target=../../../lib/$(ARCH64)/nss_dns.so.1
link path=usr/lib/$(ARCH64)/nss_files.so.1 \
    target=../../../lib/$(ARCH64)/nss_files.so.1
link path=usr/lib/$(ARCH64)/nss_nis.so.1 \
    target=../../../lib/$(ARCH64)/nss_nis.so.1
link path=usr/lib/$(ARCH64)/nss_user.so.1 \
    target=../../../lib/$(ARCH64)/nss_user.so.1
link path=usr/lib/$(ARCH64)/straddr.so target=straddr.so.2
link path=usr/lib/32 target=.
link path=usr/lib/64 target=$(ARCH64)
link path=usr/lib/cfgadm/$(ARCH64)/ib.so target=./ib.so.1
link path=usr/lib/cfgadm/$(ARCH64)/pci.so target=./pci.so.1
$(i386_ONLY)link path=usr/lib/cfgadm/$(ARCH64)/sata.so target=./sata.so.1
link path=usr/lib/cfgadm/$(ARCH64)/scsi.so target=./scsi.so.1
link path=usr/lib/cfgadm/$(ARCH64)/shp.so target=./shp.so.1
link path=usr/lib/cfgadm/$(ARCH64)/usb.so target=./usb.so.1
link path=usr/lib/cfgadm/ib.so target=./ib.so.1
link path=usr/lib/cfgadm/pci.so target=./pci.so.1
$(i386_ONLY)link path=usr/lib/cfgadm/sata.so target=./sata.so.1
link path=usr/lib/cfgadm/scsi.so target=./scsi.so.1
link path=usr/lib/cfgadm/shp.so target=./shp.so.1
link path=usr/lib/cfgadm/usb.so target=./usb.so.1
link path=usr/lib/libadm.so target=../../lib/libadm.so.1
link path=usr/lib/libadm.so.1 target=../../lib/libadm.so.1
link path=usr/lib/libadutils.so target=./libadutils.so.1
link path=usr/lib/libaio.so target=../../lib/libaio.so.1
link path=usr/lib/libaio.so.1 target=../../lib/libaio.so.1
link path=usr/lib/libavl.so.1 target=../../lib/libavl.so.1
link path=usr/lib/libbsdmalloc.so target=./libbsdmalloc.so.1
link path=usr/lib/libbsm.so target=../../lib/libbsm.so.1
link path=usr/lib/libbsm.so.1 target=../../lib/libbsm.so.1
link path=usr/lib/libc.so target=../../lib/libc.so.1
link path=usr/lib/libc.so.1 target=../../lib/libc.so.1
link path=usr/lib/libc_db.so target=../../lib/libc_db.so.1
link path=usr/lib/libc_db.so.1 target=../../lib/libc_db.so.1
link path=usr/lib/libcfgadm.so target=./libcfgadm.so.1
link path=usr/lib/libcmd.so target=libcmd.so.1
link path=usr/lib/libcmdutils.so.1 target=../../lib/libcmdutils.so.1
link path=usr/lib/libcommputil.so target=./libcommputil.so.1
link path=usr/lib/libcontract.so target=../../lib/libcontract.so.1
link path=usr/lib/libcontract.so.1 target=../../lib/libcontract.so.1
link path=usr/lib/libcrypt.so target=./libcrypt.so.1
link path=usr/lib/libcrypt_d.so target=./libcrypt.so
link path=usr/lib/libcrypt_d.so.1 target=./libcrypt.so.1
link path=usr/lib/libcrypt_i.so target=./libcrypt.so
link path=usr/lib/libcrypt_i.so.1 target=./libcrypt.so.1
link path=usr/lib/libctf.so target=../../lib/libctf.so.1
link path=usr/lib/libctf.so.1 target=../../lib/libctf.so.1
link path=usr/lib/libcurses.so target=../../lib/libcurses.so.1
link path=usr/lib/libcurses.so.1 target=../../lib/libcurses.so.1
link path=usr/lib/libdevice.so target=../../lib/libdevice.so.1
link path=usr/lib/libdevice.so.1 target=../../lib/libdevice.so.1
link path=usr/lib/libdevid.so target=../../lib/libdevid.so.1
link path=usr/lib/libdevid.so.1 target=../../lib/libdevid.so.1
link path=usr/lib/libdevinfo.so target=../../lib/libdevinfo.so.1
link path=usr/lib/libdevinfo.so.1 target=../../lib/libdevinfo.so.1
link path=usr/lib/libdhcpagent.so.1 target=../../lib/libdhcpagent.so.1
link path=usr/lib/libdhcputil.so.1 target=../../lib/libdhcputil.so.1
link path=usr/lib/libdisasm.so target=./libdisasm.so.1
link path=usr/lib/libdl.so target=../../lib/libdl.so.1
link path=usr/lib/libdl.so.1 target=../../lib/libdl.so.1
link path=usr/lib/libdlpi.so target=../../lib/libdlpi.so.1
link path=usr/lib/libdlpi.so.1 target=../../lib/libdlpi.so.1
link path=usr/lib/libdoor.so target=../../lib/libdoor.so.1
link path=usr/lib/libdoor.so.1 target=../../lib/libdoor.so.1
link path=usr/lib/libefi.so target=../../lib/libefi.so.1
link path=usr/lib/libefi.so.1 target=../../lib/libefi.so.1
link path=usr/lib/libelf.so target=../../lib/libelf.so.1
link path=usr/lib/libelf.so.1 target=../../lib/libelf.so.1
link path=usr/lib/libexacct.so target=./libexacct.so.1
$(i386_ONLY)link path=usr/lib/libfdisk.so target=../../lib/libfdisk.so.1
$(i386_ONLY)link path=usr/lib/libfdisk.so.1 target=../../lib/libfdisk.so.1
link path=usr/lib/libform.so target=./libform.so.1
link path=usr/lib/libfstyp.so target=./libfstyp.so.1
link path=usr/lib/libgen.so target=../../lib/libgen.so.1
link path=usr/lib/libgen.so.1 target=../../lib/libgen.so.1
link path=usr/lib/libhotplug.so target=./libhotplug.so.1
link path=usr/lib/libidmap.so target=./libidmap.so.1
link path=usr/lib/libinetutil.so.1 target=../../lib/libinetutil.so.1
link path=usr/lib/libintl.so target=../../lib/libintl.so.1
link path=usr/lib/libintl.so.1 target=../../lib/libintl.so.1
link path=usr/lib/libipmi.so target=./libipmi.so.1
link path=usr/lib/libipp.so target=./libipp.so.1
link path=usr/lib/libkstat.so target=../../lib/libkstat.so.1
link path=usr/lib/libkstat.so.1 target=../../lib/libkstat.so.1
link path=usr/lib/libkvm.so target=./libkvm.so.1
link path=usr/lib/libl.so target=./libl.so.1
link path=usr/lib/libldap.so target=libldap.so.5
link path=usr/lib/liblddbg.so.4 target=../../lib/liblddbg.so.4
link path=usr/lib/liblgrp.so target=./liblgrp.so.1
link path=usr/lib/liblm.so target=./liblm.so.1
link path=usr/lib/libmail.so target=./libmail.so.1
link path=usr/lib/libmalloc.so target=./libmalloc.so.1
link path=usr/lib/libmapmalloc.so target=./libmapmalloc.so.1
link path=usr/lib/libmd.so target=../../lib/libmd.so.1
link path=usr/lib/libmd.so.1 target=../../lib/libmd.so.1
link path=usr/lib/libmd5.so target=../../lib/libmd5.so.1
link path=usr/lib/libmd5.so.1 target=../../lib/libmd5.so.1
link path=usr/lib/libmenu.so target=./libmenu.so.1
link path=usr/lib/libmp.so target=../../lib/libmp.so.2
link path=usr/lib/libmp.so.1 target=../../lib/libmp.so.1
link path=usr/lib/libmp.so.2 target=../../lib/libmp.so.2
link path=usr/lib/libmtmalloc.so target=./libmtmalloc.so.1
link path=usr/lib/libnls.so target=./libnls.so.1
link path=usr/lib/libnsl.so target=../../lib/libnsl.so.1
link path=usr/lib/libnsl.so.1 target=../../lib/libnsl.so.1
link path=usr/lib/libnvpair.so target=../../lib/libnvpair.so.1
link path=usr/lib/libnvpair.so.1 target=../../lib/libnvpair.so.1
link path=usr/lib/libpam.so target=../../lib/libpam.so.1
link path=usr/lib/libpam.so.1 target=../../lib/libpam.so.1
link path=usr/lib/libpanel.so target=./libpanel.so.1
link path=usr/lib/libpkcs11.so target=./libpkcs11.so.1
link path=usr/lib/libposix4.so target=../../lib/librt.so.1
link path=usr/lib/libposix4.so.1 target=../../lib/librt.so.1
link path=usr/lib/libproc.so target=../../lib/libproc.so.1
```

```
link path=usr/lib/libproc.so.1 target=../../lib/libproc.so.1
link path=usr/lib/libproject.so target=./libproject.so.1
link path=usr/lib/libpthread.so target=../../lib/libpthread.so.1
link path=usr/lib/libpthread.so.1 target=../../lib/libpthread.so.1
link path=usr/lib/librcm.so target=../../lib/librcm.so.1
link path=usr/lib/librcm.so.1 target=../../lib/librcm.so.1
link path=usr/lib/libreparse.so target=./libreparse.so.1
link path=usr/lib/libresolv.so target=../../lib/libresolv.so.2
link path=usr/lib/libresolv.so.1 target=../../lib/libresolv.so.1
link path=usr/lib/libresolv.so.2 target=../../lib/libresolv.so.2
link path=usr/lib/librestart.so.1 target=../../lib/librestart.so.1
link path=usr/lib/librpcsvc.so target=../../lib/librpcsvc.so.1
link path=usr/lib/librpcsvc.so.1 target=../../lib/librpcsvc.so.1
link path=usr/lib/librt.so target=../../lib/librt.so.1
link path=usr/lib/librt.so.1 target=../../lib/librt.so.1
link path=usr/lib/librtld.so.1 target=../../lib/librtld.so.1
link path=usr/lib/librtld_db.so target=../../lib/librtld_db.so.1
link path=usr/lib/librtld_db.so.1 target=../../lib/librtld_db.so.1
link path=usr/lib/libscf.so target=../../lib/libscf.so.1
link path=usr/lib/libscf.so.1 target=../../lib/libscf.so.1
link path=usr/lib/libsched.so target=./libsched.so.1
link path=usr/lib/libsctp.so target=./libsctp.so.1
link path=usr/lib/libsec.so target=../../lib/libsec.so.1
link path=usr/lib/libsec.so.1 target=../../lib/libsec.so.1
link path=usr/lib/libsecdb.so target=../../lib/libsecdb.so.1
link path=usr/lib/libsecdb.so.1 target=../../lib/libsecdb.so.1
link path=usr/lib/libsendfile.so target=../../lib/libsendfile.so.1
link path=usr/lib/libsendfile.so.1 target=../../lib/libsendfile.so.1
link path=usr/lib/libsip.so target=./libsip.so.1
link path=usr/lib/libsldap.so target=libsldap.so.1
link path=usr/lib/libsmbios.so target=libsmbios.so.1
link path=usr/lib/libsocket.so target=../../lib/libsocket.so.1
link path=usr/lib/libsocket.so.1 target=../../lib/libsocket.so.1
link path=usr/lib/libsoftcrypto.so target=./libsoftcrypto.so.1
link path=usr/lib/libsys.so target=./libsys.so.1
link path=usr/lib/libsysevent.so target=../../lib/libsysevent.so.1
link path=usr/lib/libsysevent.so.1 target=../../lib/libsysevent.so.1
link path=usr/lib/libtermcap.so target=../../lib/libtermcap.so.1
link path=usr/lib/libtermcap.so.1 target=../../lib/libtermcap.so.1
link path=usr/lib/libtermlib.so target=../../lib/libcurses.so.1
link path=usr/lib/libtermlib.so.1 target=../../lib/libcurses.so.1
link path=usr/lib/libthread.so target=../../lib/libthread.so.1
link path=usr/lib/libthread.so.1 target=../../lib/libthread.so.1
link path=usr/lib/libthread_db.so target=../../lib/libc_db.so.1
link path=usr/lib/libthread_db.so.1 target=../../lib/libc_db.so.1
link path=usr/lib/libtsnet.so target=../../lib/libtsnet.so.1
link path=usr/lib/libtsnet.so.1 target=../../lib/libtsnet.so.1
link path=usr/lib/libtsol.so target=../../lib/libtsol.so.2
link path=usr/lib/libtsol.so.2 target=../../lib/libtsol.so.2
link path=usr/lib/libumem.so target=../../lib/libumem.so.1
link path=usr/lib/libumem.so.1 target=../../lib/libumem.so.1
link path=usr/lib/libuuid.so target=../../lib/libuuid.so.1
link path=usr/lib/libuuid.so.1 target=../../lib/libuuid.so.1
link path=usr/lib/libuutil.so.1 target=../../lib/libuutil.so.1
$(sparc_ONLY)link path=usr/lib/libv12n.so target=./libv12n.so.1
link path=usr/lib/libvolmgt.so target=./libvolmgt.so.1
link path=usr/lib/libw.so target=../../lib/libw.so.1
link path=usr/lib/libw.so.1 target=../../lib/libw.so.1
link path=usr/lib/libwrap.so target=libwrap.so.1.0
link path=usr/lib/libwrap.so.1 target=libwrap.so.1.0
link path=usr/lib/libxnet.so target=../../lib/libxnet.so.1
link path=usr/lib/libxnet.so.1 target=../../lib/libxnet.so.1
link path=usr/lib/liby.so target=./liby.so.1
link path=usr/lib/libzoneinfo.so target=./libzoneinfo.so.1
link path=usr/lib/lwp/$(ARCH64)/libthread.so.1 \
    target=../../$(ARCH64)/libthread.so.1
```

```
link path=usr/lib/lwp/$(ARCH64)/libthread_db.so.1 \
    target=../../$(ARCH64)/libthread_db.so.1
link path=usr/lib/lwp/32 target=.
link path=usr/lib/lwp/64 target=$(ARCH64)
link path=usr/lib/lwp/libthread.so.1 target=../libthread.so.1
link path=usr/lib/lwp/libthread_db.so.1 target=../libthread_db.so.1
link path=usr/lib/nss_compat.so.1 target=../../lib/nss_compat.so.1
link path=usr/lib/nss_dns.so.1 target=../../lib/nss_dns.so.1
link path=usr/lib/nss_files.so.1 target=../../lib/nss_files.so.1
link path=usr/lib/nss_nis.so.1 target=../../lib/nss_nis.so.1
link path=usr/lib/nss_user.so.1 target=../../lib/nss_user.so.1
link path=usr/lib/scsi/$(ARCH64)/libscsi.so target=./libscsi.so.1
link path=usr/lib/scsi/$(ARCH64)/libses.so target=./libses.so.1
link path=usr/lib/scsi/$(ARCH64)/libsmp.so target=./libsmp.so.1
link path=usr/lib/scsi/libscsi.so target=./libscsi.so.1
link path=usr/lib/scsi/libses.so target=./libses.so.1
link path=usr/lib/scsi/libsmp.so target=./libsmp.so.1
link path=usr/lib/security/$(ARCH64)/crypt_bsdbf.so target=./crypt_bsdbf.so.1
link path=usr/lib/security/$(ARCH64)/crypt_bsdmd5.so \
    target=./crypt_bsdmd5.so.1
link path=usr/lib/security/$(ARCH64)/crypt_sha256.so \
    target=./crypt_sha256.so.1
link path=usr/lib/security/$(ARCH64)/crypt_sha512.so \
    target=./crypt_sha512.so.1
link path=usr/lib/security/$(ARCH64)/crypt_sunmd5.so \
    target=./crypt_sunmd5.so.1
link path=usr/lib/security/$(ARCH64)/pam_allow.so target=./pam_allow.so.1
link path=usr/lib/security/$(ARCH64)/pam_authtok_check.so \
    target=./pam_authtok_check.so.1
link path=usr/lib/security/$(ARCH64)/pam_authtok_get.so \
    target=./pam_authtok_get.so.1
link path=usr/lib/security/$(ARCH64)/pam_authtok_store.so \
    target=./pam_authtok_store.so.1
link path=usr/lib/security/$(ARCH64)/pam_deny.so target=./pam_deny.so.1
link path=usr/lib/security/$(ARCH64)/pam_dhkeys.so target=./pam_dhkeys.so.1
link path=usr/lib/security/$(ARCH64)/pam_dial_auth.so \
    target=./pam_dial_auth.so.1
link path=usr/lib/security/$(ARCH64)/pam_ldap.so target=./pam_ldap.so.1
link path=usr/lib/security/$(ARCH64)/pam_list.so target=./pam_list.so.1
link path=usr/lib/security/$(ARCH64)/pam_passwd_auth.so \
    target=./pam_passwd_auth.so.1
link path=usr/lib/security/$(ARCH64)/pam_rhosts_auth.so \
    target=./pam_rhosts_auth.so.1
link path=usr/lib/security/$(ARCH64)/pam_roles.so target=./pam_roles.so.1
link path=usr/lib/security/$(ARCH64)/pam_sample.so target=./pam_sample.so.1
link path=usr/lib/security/$(ARCH64)/pam_tsol_account.so \
    target=./pam_tsol_account.so.1
link path=usr/lib/security/$(ARCH64)/pam_unix_account.so \
    target=./pam_unix_account.so.1
link path=usr/lib/security/$(ARCH64)/pam_unix_auth.so \
    target=./pam_unix_auth.so.1
link path=usr/lib/security/$(ARCH64)/pam_unix_cred.so \
    target=./pam_unix_cred.so.1
link path=usr/lib/security/$(ARCH64)/pam_unix_session.so \
    target=./pam_unix_session.so.1
link path=usr/lib/security/$(ARCH64)/pkcs11_kernel.so \
    target=./pkcs11_kernel.so.1
link path=usr/lib/security/$(ARCH64)/pkcs11_softtoken.so \
    target=./pkcs11_softtoken.so.1
link path=usr/lib/security/$(ARCH64)/pkcs11_tpm.so target=./pkcs11_tpm.so.1
link path=usr/lib/security/64 target=$(ARCH64)
link path=usr/lib/security/audit_binfile.so target=./audit_binfile.so.1
link path=usr/lib/security/audit_remote.so target=./audit_remote.so.1
link path=usr/lib/security/audit_syslog.so target=./audit_syslog.so.1
link path=usr/lib/security/crypt_bsdbf.so target=./crypt_bsdbf.so.1
link path=usr/lib/security/crypt_bsdmd5.so target=./crypt_bsdmd5.so.1
```

```
link path=usr/lib/security/crypt_sha256.so target=./crypt_sha256.so.1
link path=usr/lib/security/crypt_sha512.so target=./crypt_sha512.so.1
link path=usr/lib/security/crypt_sunmd5.so target=./crypt_sunmd5.so.1
link path=usr/lib/security/pam_allow.so target=./pam_allow.so.1
link path=usr/lib/security/pam_authtok_check.so \
    target=./pam_authtok_check.so.1
link path=usr/lib/security/pam_authtok_get.so target=./pam_authtok_get.so.1
link path=usr/lib/security/pam_authtok_store.so \
    target=./pam_authtok_store.so.1
link path=usr/lib/security/pam_deny.so target=./pam_deny.so.1
link path=usr/lib/security/pam_dhkeys.so target=./pam_dhkeys.so.1
link path=usr/lib/security/pam_dial_auth.so target=./pam_dial_auth.so.1
link path=usr/lib/security/pam_ldap.so target=./pam_ldap.so.1
link path=usr/lib/security/pam_list.so target=./pam_list.so.1
link path=usr/lib/security/pam_passwd_auth.so target=./pam_passwd_auth.so.1
link path=usr/lib/security/pam_rhosts_auth.so target=./pam_rhosts_auth.so.1
link path=usr/lib/security/pam_roles.so target=./pam_roles.so.1
link path=usr/lib/security/pam_sample.so target=./pam_sample.so.1
link path=usr/lib/security/pam_tsol_account.so target=./pam_tsol_account.so.1
link path=usr/lib/security/pam_unix_account.so target=./pam_unix_account.so.1
link path=usr/lib/security/pam_unix_auth.so target=./pam_unix_auth.so.1
link path=usr/lib/security/pam_unix_cred.so target=./pam_unix_cred.so.1
link path=usr/lib/security/pam_unix_session.so target=./pam_unix_session.so.1
link path=usr/lib/security/pkcs11_kernel.so target=./pkcs11_kernel.so.1
link path=usr/lib/security/pkcs11_softtoken.so target=./pkcs11_softtoken.so.1
link path=usr/lib/security/pkcs11_tpm.so target=./pkcs11_tpm.so.1
link path=usr/lib/straddr.so target=./straddr.so.2
link path=usr/xpg4/lib/$(ARCH64)/libcurses.so target=libcurses.so.2
link path=usr/xpg4/lib/64 target=$(ARCH64)
link path=usr/xpg4/lib/libcurses.so target=./libcurses.so.2
#
# libses.so needs to dlopen(3C) plugins from usr/lib/scsi/plugins/ses/vendor/,
# a dependency which cannot be automatically derived
#
depend fmri=system/library/storage/scsi-plugins type=require
```