```
**********************************************************
   26812 Mon Apr  6 11:47:10 2015
new/usr/src/head/unistd.h
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

  22 /*
  23  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
  24  * Copyright (c) 2013 Gary Mills
  25  *
  26  * Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
  27  */

  29 /*      Copyright (c) 1988 AT&T */
  30 /*        All Rights Reserved   */

  32 /* Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved. */

  34 #ifndef _UNISTD_H
  35 #define _UNISTD_H

  37 #include <sys/feature_tests.h>

  39 #include <sys/types.h>
  40 #include <sys/unistd.h>

  42 #ifdef  __cplusplus
  43 extern "C" {
  44 #endif

  46 /* Symbolic constants for the "access" routine: */
  47 #define R_OK    4       /* Test for Read permission */
  48 #define W_OK    2       /* Test for Write permission */
  49 #define X_OK    1       /* Test for eXecute permission */
  50 #define F_OK    0       /* Test for existence of File */

  52 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
  53 #define F_ULOCK 0       /* Unlock a previously locked region */
  54 #define F_LOCK  1       /* Lock a region for exclusive use */
  55 #define F_TLOCK 2       /* Test and lock a region for exclusive use */
  56 #define F_TEST  3       /* Test a region for other processes locks */
  57 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */

  59 /* Symbolic constants for the "lseek" routine: */

  61 #ifndef SEEK_SET
```



```
  62 #define SEEK_SET        0       /* Set file pointer to "offset" */
  63 #endif

  65 #ifndef SEEK_CUR
  66 #define SEEK_CUR        1       /* Set file pointer to current plus "offset" */
  67 #endif

  69 #ifndef SEEK_END
  70 #define SEEK_END        2       /* Set file pointer to EOF plus "offset" */
  71 #endif

  73 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
  74 #ifndef SEEK_DATA
  75 #define SEEK_DATA       3       /* Set file pointer to next data past offset */
  76 #endif

  78 #ifndef SEEK_HOLE
  79 #define SEEK_HOLE       4       /* Set file pointer to next hole past offset */
  80 #endif
  81 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

  83 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
  84 /* Path names: */
  85 #define GF_PATH "/etc/group"    /* Path name of the "group" file */
  86 #define PF_PATH "/etc/passwd"   /* Path name of the "passwd" file */
  87 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

  89 /*
  90  * compile-time symbolic constants,
  91  * Support does not mean the feature is enabled.
  92  * Use pathconf/sysconf to obtain actual configuration value.
  93  */

  95 /* Values unchanged in UNIX 03 */
  96 #define _POSIX_ASYNC_IO                 1
  97 #define _POSIX_JOB_CONTROL              1
  98 #define _POSIX_SAVED_IDS                1
  99 #define _POSIX_SYNC_IO                  1

 101 /*
 102  * POSIX.1b compile-time symbolic constants.
 103  */
 104 #if defined(_XPG6)
 105 #define _POSIX_ASYNCHRONOUS_IO          200112L
 106 #define _POSIX_FSYNC                    200112L
 107 #define _POSIX_MAPPED_FILES             200112L
 108 #define _POSIX_MEMLOCK                  200112L
 109 #define _POSIX_MEMLOCK_RANGE            200112L
 110 #define _POSIX_MEMORY_PROTECTION        200112L
 111 #define _POSIX_MESSAGE_PASSING          200112L
 112 #define _POSIX_PRIORITY_SCHEDULING      200112L
 113 #define _POSIX_REALTIME_SIGNALS         200112L
 114 #define _POSIX_SEMAPHORES               200112L
 115 #define _POSIX_SHARED_MEMORY_OBJECTS    200112L
 116 #define _POSIX_SYNCHRONIZED_IO          200112L
 117 #else
 118 #define _POSIX_ASYNCHRONOUS_IO          1
 119 #define _POSIX_FSYNC                    1
 120 #define _POSIX_MAPPED_FILES             1
 121 #define _POSIX_MEMLOCK                  1
 122 #define _POSIX_MEMLOCK_RANGE            1
 123 #define _POSIX_MEMORY_PROTECTION        1
 124 #define _POSIX_MESSAGE_PASSING          1
 125 #define _POSIX_PRIORITY_SCHEDULING      1
 126 #define _POSIX_REALTIME_SIGNALS         1
 127 #define _POSIX_SEMAPHORES               1
```

```
128 #define _POSIX_SHARED_MEMORY_OBJECTS    1
129 #define _POSIX_SYNCHRONIZED_IO          1
130 #endif

132 /*
133  * POSIX.1c compile-time symbolic constants.
134  */
135 #if defined(_XPG6)
136 #define _POSIX_THREAD_SAFE_FUNCTIONS            200112L
137 #define _POSIX_THREADS                          200112L
138 #define _POSIX_THREAD_ATTR_STACKADDR            200112L
139 #define _POSIX_THREAD_ATTR_STACKSIZE            200112L
140 #define _POSIX_THREAD_PROCESS_SHARED            200112L
141 #define _POSIX_THREAD_PRIORITY_SCHEDULING       200112L
142 #define _POSIX_TIMERS                           200112L
143 #else
144 #define _POSIX_THREAD_SAFE_FUNCTIONS            1
145 #define _POSIX_THREADS                          1
146 #define _POSIX_THREAD_ATTR_STACKADDR            1
147 #define _POSIX_THREAD_ATTR_STACKSIZE            1
148 #define _POSIX_THREAD_PROCESS_SHARED            1
149 #define _POSIX_THREAD_PRIORITY_SCHEDULING       1
150 #define _POSIX_TIMERS                           1
151 #endif

153 /* New in UNIX 03 */
154 #define _POSIX_ADVISORY_INFO                    200112L
155 #define _POSIX_BARRIERS                         200112L
156 #define _POSIX_CLOCK_SELECTION                  200112L
157 #define _POSIX_IPV6                             200112L
158 #define _POSIX_MONOTONIC_CLOCK                  200112L
159 #define _POSIX_RAW_SOCKETS                      200112L
160 #define _POSIX_READER_WRITER_LOCKS              200112L
161 #define _POSIX_SPAWN                            200112L
162 #define _POSIX_SPIN_LOCKS                       200112L
163 #define _POSIX_TIMEOUTS                         200112L

165 /*
166  * Support for the POSIX.1 mutex protocol attribute. For realtime applications
167  * which need mutexes to support priority inheritance/ceiling.
168  */
169 #if defined(_XPG6)
170 #define _POSIX_THREAD_PRIO_INHERIT              200112L
171 #define _POSIX_THREAD_PRIO_PROTECT              200112L
172 #else
173 #define _POSIX_THREAD_PRIO_INHERIT              1
174 #define _POSIX_THREAD_PRIO_PROTECT              1
175 #endif

177 #ifndef _POSIX_VDISABLE
178 #define _POSIX_VDISABLE         0
179 #endif

181 #ifndef NULL
182 #if defined(_LP64)
183 #define NULL    0L
184 #else
185 #define NULL    0
186 #endif
187 #endif

189 #define STDIN_FILENO    0
190 #define STDOUT_FILENO   1
191 #define STDERR_FILENO   2

193 /*
```



```
194  * Large File Summit-related announcement macros.  The system supports both
195  * the additional and transitional Large File Summit interfaces.  (The final
196  * two macros provide a finer granularity breakdown of _LFS64_LARGEFILE.)
197  */
198 #define _LFS_LARGEFILE          1
199 #define _LFS64_LARGEFILE        1
200 #define _LFS64_STDIO            1
201 #define _LFS64_ASYNCHRONOUS_IO  1

203 /* large file compilation environment setup */
204 #if !defined(_LP64) && _FILE_OFFSET_BITS == 64
205 #ifdef  __PRAGMA_REDEFINE_EXTNAME
206 #pragma redefine_extname        ftruncate       ftruncate64
207 #pragma redefine_extname        lseek           lseek64
208 #pragma redefine_extname        pread           pread64
209 #pragma redefine_extname        pwrite          pwrite64
210 #pragma redefine_extname        truncate        truncate64
211 #pragma redefine_extname        lockf           lockf64
212 #pragma redefine_extname        tell            tell64
213 #else   /* __PRAGMA_REDEFINE_EXTNAME */
214 #define ftruncate                       ftruncate64
215 #define lseek                           lseek64
216 #define pread                           pread64
217 #define pwrite                          pwrite64
218 #define truncate                        truncate64
219 #define lockf                           lockf64
220 #define tell                            tell64
221 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
222 #endif  /* !_LP64 && _FILE_OFFSET_BITS == 64 */

224 /* In the LP64 compilation environment, the APIs are already large file */
225 #if defined(_LP64) && defined(_LARGEFILE64_SOURCE)
226 #ifdef  __PRAGMA_REDEFINE_EXTNAME
227 #pragma redefine_extname        ftruncate64     ftruncate
228 #pragma redefine_extname        lseek64         lseek
229 #pragma redefine_extname        pread64         pread
230 #pragma redefine_extname        pwrite64        pwrite
231 #pragma redefine_extname        truncate64      truncate
232 #pragma redefine_extname        lockf64         lockf
233 #pragma redefine_extname        tell64          tell
234 #else   /* __PRAGMA_REDEFINE_EXTNAME */
235 #define ftruncate64                     ftruncate
236 #define lseek64                         lseek
237 #define pread64                         pread
238 #define pwrite64                        pwrite
239 #define truncate64                      truncate
240 #define lockf64                         lockf
241 #define tell64                          tell
242 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
243 #endif  /* _LP64 && _LARGEFILE64_SOURCE */

245 extern int access(const char *, int);
246 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
247 extern int acct(const char *);
248 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
249 extern unsigned alarm(unsigned);
250 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
251 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
252         defined(__EXTENSIONS__)
253 extern int brk(void *);
254 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
255 extern int chdir(const char *);
256 extern int chown(const char *, uid_t, gid_t);
257 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
258 #if !defined(_POSIX_C_SOURCE) || (defined(_XOPEN_SOURCE) && \
259         !defined(_XPG6)) || defined(__EXTENSIONS__)
```

```
260 extern int chroot(const char *);
261 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE))... */
262 extern int close(int);
263 #if defined(_XPG4) || defined(__EXTENSIONS__)
264 extern size_t confstr(int, char *, size_t);
265 extern char *crypt(const char *, const char *);
266 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
267 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
268         defined(__EXTENSIONS__)
269 extern char *ctermid(char *);
270 #endif /* (!defined(_POSIX_C_SOURCE) ... */
271 #if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
272 extern char *ctermid_r(char *);
273 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
274 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
275 #if !defined(_XPG6) || defined(__EXTENSIONS__)
276 extern char *cuserid(char *);
277 #endif
278 extern int dup(int);
279 extern int dup2(int, int);
280 extern int dup3(int, int, int);
281 #if defined(_XPG4) || defined(__EXTENSIONS__)
282 extern void encrypt(char *, int);
283 #endif /* defined(XPG4) || defined(__EXTENSIONS__) */
284 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
285 extern void endusershell(void);
286 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
287 extern int execl(const char *, const char *, ...);
288 extern int execle(const char *, const char *, ...);
289 extern int execlp(const char *, const char *, ...);
290 extern int execv(const char *, char *const *);
291 extern int execve(const char *, char *const *, char *const *);
292 extern int execvp(const char *, char *const *);
293 #if defined(_XPG7) || !defined(_STRICT_SYMBOLS)
294 extern int fexecve(int, char *const[], char *const[]);
295 #endif
296 extern void _exit(int)
297         __NORETURN;
298 /*
299  * The following fattach prototype is duplicated in <stropts.h>. The
300  * duplication is necessitated by XPG4.2 which requires the prototype
301  * be defined in <stropts.h>.
302  */
303 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
304 extern int fattach(int, const char *);
305 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
306 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
307 extern int fchdir(int);
308 extern int fchown(int, uid_t, gid_t);
309 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
310 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
311 extern int fchroot(int);
312 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
313 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
314         defined(__EXTENSIONS__)
315 extern int fdatasync(int);
316 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
317 /*
318  * The following fdetach prototype is duplicated in <stropts.h>. The
319  * duplication is necessitated by XPG4.2 which requires the prototype
320  * be defined in <stropts.h>.
321  */
322 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
323 extern int fdetach(const char *);
324 #endif /* !defined(__XOPEN_OR_POSIX)... */
325 extern pid_t fork(void);
```



```
326 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
327 extern pid_t fork1(void);
328 extern pid_t forkall(void);
329 #endif /* !defined(__XOPEN_OR_POSIX)... */
330 extern long fpathconf(int, int);
331 #if !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2) || \
332         defined(__EXTENSIONS__)
333 extern int fsync(int);
334 #endif /* !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2)... */
335 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || defined(_XPG4_2) || \
336         (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
337         defined(__EXTENSIONS__)
338 extern int ftruncate(int, off_t);
339 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
340 extern char *getcwd(char *, size_t);
341 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
342         defined(__EXTENSIONS__)
343 extern int getdtablesize(void);
344 #endif
345 extern gid_t getegid(void);
346 extern uid_t geteuid(void);
347 extern gid_t getgid(void);
348 extern int getgroups(int, gid_t *);
349 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
350 extern long gethostid(void);
351 #endif
352 #if defined(_XPG4_2)
353 extern int gethostname(char *, size_t);
354 #elif  !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
355 extern int gethostname(char *, int);
356 #endif

358 #ifndef __GETLOGIN_DEFINED      /* Avoid duplicate in stdlib.h */
359 #define __GETLOGIN_DEFINED
360 #ifndef __USE_LEGACY_LOGNAME__
361 #ifdef  __PRAGMA_REDEFINE_EXTNAME
362 #pragma redefine_extname getlogin getloginx
363 #else   /* __PRAGMA_REDEFINE_EXTNAME */
364 extern char *getloginx(void);
365 #define getlogin        getloginx
366 #endif  /* __PRAGMA_REDEFINE_EXTNAME */
367 #endif  /* __USE_LEGACY_LOGNAME__ */
368 extern char *getlogin(void);
369 #endif  /* __GETLOGIN_DEFINED */

371 #if defined(_XPG4) || defined(__EXTENSIONS__)
372 extern int  getopt(int, char *const *, const char *);
373 extern char *optarg;
374 extern int  opterr, optind, optopt;
375 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
376 #if !defined(_XPG6) || defined(__EXTENSIONS__)
377 extern char *getpass(const char *);
378 #endif
379 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
380 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
381 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
382 #if !defined(_XPG6) || defined(__EXTENSIONS__)
383 extern int getpagesize(void);
384 #endif
385 extern pid_t getpgid(pid_t);
386 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
387 extern pid_t getpid(void);
388 extern pid_t getppid(void);
389 extern pid_t getpgrp(void);

391 #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 */
_____unchanged_portion_omitted_
```

```
**********************************************************
   22596 Mon Apr  6 11:47:13 2015
new/usr/src/lib/libc/amd64/Makefile
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc. All rights reserved.
#
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBCBASE=       .
LIBCDIR=        $(SRC)/lib/libc
LIBRARY=        libc.a
LIB_PIC=        libc_pic.a
VERS=           .1
CPP=            /usr/lib/cpp
TARGET_ARCH=    amd64

# objects are grouped by source directory

# local objects
STRETS=

CRTOBJS=                        \
        cerror.o

DYNOBJS=

FPOBJS=                         \
        fpgetmask.o             \
        fpgetround.o            \
        fpsetmask.o             \
        fpsetround.o            \
        fpstart.o               \
        ieee.o

I386FPOBJS=                     \
        _D_cplx_div.o           \
        _D_cplx_div_ix.o        \
        _D_cplx_div_rx.o        \
```



```
        _D_cplx_lr_div.o        \
        _D_cplx_lr_div_ix.o     \
        _D_cplx_lr_div_rx.o     \
        _D_cplx_mul.o           \
        _F_cplx_div.o           \
        _F_cplx_div_ix.o        \
        _F_cplx_div_rx.o        \
        _F_cplx_lr_div.o        \
        _F_cplx_lr_div_ix.o     \
        _F_cplx_lr_div_rx.o     \
        _F_cplx_mul.o           \
        _X_cplx_div.o           \
        _X_cplx_div_ix.o        \
        _X_cplx_div_rx.o        \
        _X_cplx_lr_div.o        \
        _X_cplx_lr_div_ix.o     \
        _X_cplx_lr_div_rx.o     \
        _X_cplx_mul.o

FPASMOBJS=                      \
        __xgetRD.o              \
        _xtoll.o                \
        _xtoull.o               \
        _base_il.o              \
        fpcw.o                  \
        fpgetsticky.o           \
        fpsetsticky.o

ATOMICOBJS=                     \
        atomic.o

XATTROBJS=                      \
        xattr_common.o
COMOBJS=                        \
        bcmp.o                  \
        bcopy.o                 \
        bsearch.o               \
        bzero.o                 \
        qsort.o                 \
        strtol.o                \
        strtoul.o               \
        strtoll.o               \
        strtoull.o

GENOBJS=                        \
        _getsp.o                \
        abs.o                   \
        alloca.o                \
        attrat.o                \
        byteorder.o             \
        cuexit.o                \
        ecvt.o                  \
        errlst.o                \
        amd64_data.o            \
        ldivide.o               \
        lock.o                  \
        makectxt.o              \
        memccpy.o               \
        memchr.o                \
        memcmp.o                \
        memcpy.o                \
        memset.o                \
        new_list.o              \
        proc64_id.o             \
        proc64_support.o        \
        setjmp.o                \
```

```
127         siginfolst.o            \
128         siglongjmp.o            \
129         strcmp.o                \
130         strcpy.o                \
131         strlen.o                \
132         strncmp.o               \
133         strncpy.o               \
134         strnlen.o               \
135         sync_instruction_memory.o

137 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
138 # This macro should ALWAYS be empty; native APIs are already 'large file'.
139 COMSYSOBJS64=

141 SYSOBJS64=

143 COMSYSOBJS=                     \
144         __clock_timer.o         \
145         __getloadavg.o          \
146         __rusagesys.o           \
147         __signotify.o           \
148         __sigrt.o               \
149         __time.o                \
150         _lgrp_home_fast.o       \
151         _lgrpsys.o              \
152         _nfssys.o               \
153         _portfs.o               \
154         _pset.o                 \
155         _rpcsys.o               \
156         _sigaction.o            \
157         _so_accept.o            \
158         _so_bind.o              \
159         _so_connect.o           \
160         _so_getpeername.o       \
161         _so_getsockname.o       \
162         _so_getsockopt.o        \
163         _so_listen.o            \
164         _so_recv.o              \
165         _so_recvfrom.o          \
166         _so_recvmsg.o           \
167         _so_send.o              \
168         _so_sendmsg.o           \
169         _so_sendto.o            \
170         _so_setsockopt.o        \
171         _so_shutdown.o          \
172         _so_socket.o            \
173         _so_socketpair.o        \
174         _sockconfig.o           \
175         acct.o                  \
176         acl.o                   \
177         adjtime.o               \
178         alarm.o                 \
179         brk.o                   \
180         chdir.o                 \
181         chroot.o                \
182         cladm.o                 \
183         close.o                 \
184         execve.o                \
185         exit.o                  \
186         facl.o                  \
187         fchdir.o                \
188         fchroot.o               \
189         fdsync.o                \
190         fpathconf.o             \
191         fstatfs.o               \
192         fstatvfs.o              \
```



```
193         getcpuid.o              \
194         getdents.o              \
195         getegid.o               \
196         geteuid.o               \
197         getgid.o                \
198         getgroups.o             \
199         gethrtime.o             \
200         getitimer.o             \
201         getmsg.o                \
202         getpid.o                \
203         getpmsg.o               \
204         getppid.o               \
205         getrlimit.o             \
206         getuid.o                \
207         gtty.o                  \
208         install_utrap.o         \
209         ioctl.o                 \
210         kaio.o                  \
211         kill.o                  \
212         llseek.o                \
213         lseek.o                 \
214         mmapobjsys.o            \
215         memcntl.o               \
216         mincore.o               \
217         mmap.o                  \
218         modctl.o                \
219         mount.o                 \
220         mprotect.o              \
221         munmap.o                \
222         nice.o                  \
223         ntp_adjtime.o           \
224         ntp_gettime.o           \
225         p_online.o              \
226         pathconf.o              \
227         pause.o                 \
228         pcsample.o              \
229         pipe2.o                 \
230         pollsys.o               \
231         pread.o                 \
232         preadv.o                \
233         priocntlset.o           \
234         processor_bind.o        \
235         processor_info.o        \
236         profil.o                \
237         putmsg.o                \
238         putpmsg.o               \
239         pwrite.o                \
240         pwritev.o               \
241         read.o                  \
242         readv.o                 \
243         resolvepath.o           \
244         seteguid.o              \
245         setgid.o                \
246         setgroups.o             \
247         setitimer.o             \
248         setreid.o               \
249         setrlimit.o             \
250         setuid.o                \
251         sigaltstk.o             \
252         sigprocmsk.o            \
253         sigsendset.o            \
254         sigsuspend.o            \
255         statfs.o                \
256         statvfs.o               \
257         stty.o                  \
258         sync.o                  \
```

```
259         sysconfig.o             \
260         sysfs.o                 \
261         sysinfo.o               \
262         syslwp.o                \
263         times.o                 \
264         ulimit.o                \
265         umask.o                 \
266         umount2.o               \
267         utssys.o                \
268         uucopy.o                \
269         vhangup.o               \
270         waitid.o                \
271         write.o                 \
272         writev.o                \
273         yield.o

275 SYSOBJS=                       \
276         __clock_gettime.o       \
277         __getcontext.o          \
278         __uadmin.o              \
279         _lwp_mutex_unlock.o     \
280         _stack_grow.o           \
281         door.o                  \
282         forkx.o                 \
283         forkallx.o              \
284         getcontext.o            \
285         gettimeofday.o          \
286         lwp_private.o           \
287         nuname.o                \
288         syscall.o               \
289         sysi86.o                \
290         tls_get_addr.o          \
291         uadmin.o                \
292         umount.o                \
293         uname.o                 \
294         vforkx.o

296 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
297 # This macro should ALWAYS be empty; native APIs are already 'large file'.
298 PORTGEN64=

300 # objects from source under $(LIBCDIR)/port
301 PORTFP=                         \
302         __flt_decim.o           \
303         __flt_rounds.o          \
304         __tbl_10_b.o            \
305         __tbl_10_h.o            \
306         __tbl_10_s.o            \
307         __tbl_2_b.o             \
308         __tbl_2_h.o             \
309         __tbl_2_s.o             \
310         __tbl_fdq.o             \
311         __tbl_tens.o            \
312         __x_power.o             \
313         _base_sup.o             \
314         aconvert.o              \
315         decimal_bin.o           \
316         double_decim.o          \
317         econvert.o              \
318         fconvert.o              \
319         file_decim.o            \
320         finite.o                \
321         fp_data.o               \
322         func_decim.o            \
323         gconvert.o              \
324         hex_bin.o               \
```



```
325         ieee_globals.o          \
326         pack_float.o            \
327         sigfpe.o                \
328         string_decim.o

330 PORTGEN=                       \
331         _env_data.o             \
332         _xftw.o                 \
333         a64l.o                  \
334         abort.o                 \
335         addsev.o                \
336         ascii_strcasecmp.o      \
337         ascii_strncasecmp.o     \
338         assert.o                \
339         atof.o                  \
340         atoi.o                  \
341         atol.o                  \
342         atoll.o                 \
343         attropen.o              \
344         atexit.o                \
345         atfork.o                \
346         basename.o              \
347         calloc.o                \
348         catgets.o               \
349         catopen.o               \
350         cfgetispeed.o           \
351         cfgetospeed.o           \
352         cfree.o                 \
353         cfsetispeed.o           \
354         cfsetospeed.o           \
355         cftime.o                \
356         clock.o                 \
357         closedir.o              \
358         closefrom.o             \
359         confstr.o               \
360         crypt.o                 \
361         csetlen.o               \
362         ctime.o                 \
363         ctime_r.o               \
364         daemon.o                \
365         deflt.o                 \
366         directio.o              \
367         dirname.o               \
368         div.o                   \
369         drand48.o               \
370         dup.o                   \
371         env_data.o              \
372         err.o                   \
373         errno.o                 \
374         euclen.o                \
375         event_port.o            \
376         execvp.o                \
377         fattach.o               \
378         fdetach.o               \
379         fdopendir.o             \
380         ffs.o                   \
381         fls.o                   \
382         fmtmsg.o                \
383         ftime.o                 \
384         ftok.o                  \
385         ftw.o                   \
386         gcvt.o                  \
387         getauxv.o               \
388         getcwd.o                \
389         getdate_err.o           \
390         getdtblsize.o           \
```

```
391         getenv.o                \
392         getexecname.o           \
393         getgrnam.o              \
394         getgrnam_r.o            \
395         gethostid.o             \
396         gethostname.o           \
397         gethz.o                 \
398         getisax.o               \
399         getloadavg.o            \
400         getlogin.o              \
401         getmntent.o             \
402         getnetgrent.o           \
403         get_nprocs.o            \
404         getopt.o                \
405         getopt_long.o           \
406         getpagesize.o           \
407         getpw.o                 \
408         getpwnam.o              \
409         getpwnam_r.o            \
410         getrusage.o             \
411         getspent.o              \
412         getspent_r.o            \
413         getsubopt.o             \
414         gettxt.o                \
415         getusershell.o          \
416         getut.o                 \
417         getutx.o                \
418         getvfsent.o             \
419         getwd.o                 \
420         getwidth.o              \
421         getxby_door.o           \
422         gtxt.o                  \
423         hsearch.o               \
424         iconv.o                 \
425         imaxabs.o               \
426         imaxdiv.o               \
427         index.o                 \
428         initgroups.o            \
429         insque.o                \
430         isaexec.o               \
431         isastream.o             \
432         isatty.o                \
433         killpg.o                \
434         klpdlib.o               \
435         l64a.o                  \
436         lckpwdf.o               \
437         lconstants.o            \
438         lexp10.o                \
439         lfind.o                 \
440         lfmt.o                  \
441         lfmt_log.o              \
442         lldiv.o                 \
443         llog10.o                \
444         lltostr.o               \
445         lmath.o                 \
446         localtime.o             \
447         lsearch.o               \
448         madvise.o               \
449         malloc.o                \
450         memalign.o              \
451         memmem.o                \
452         mkdev.o                 \
453         mkdtemp.o               \
454         mkfifo.o                \
455         mkstemp.o               \
456         mktemp.o                \
```



```
457         mlock.o                 \
458         mlockall.o              \
459         mon.o                   \
460         msync.o                 \
461         munlock.o               \
462         munlockall.o            \
463         ndbm.o                  \
464         nftw.o                  \
465         nlspath_checks.o        \
466         nsparse.o               \
467         nss_common.o            \
468         nss_dbdefs.o            \
469         nss_deffinder.o         \
470         opendir.o               \
471         opt_data.o              \
472         perror.o                \
473         pfmt.o                  \
474         pfmt_data.o             \
475         pfmt_print.o            \
476         pipe.o                  \
477         plock.o                 \
478         poll.o                  \
479         posix_fadvise.o         \
480         posix_fallocate.o       \
481         posix_madvise.o         \
482         posix_memalign.o        \
483         priocntl.o              \
484         privlib.o               \
485         priv_str_xlate.o        \
486         psiginfo.o              \
487         psignal.o               \
488         pt.o                    \
489         putpwent.o              \
490         putspent.o              \
491         raise.o                 \
492         rand.o                  \
493         random.o                \
494         rctlops.o               \
495         readdir.o               \
496         readdir_r.o             \
497         realpath.o              \
498         reboot.o                \
499         regexpr.o               \
500         remove.o                \
501         rewinddir.o             \
502         rindex.o                \
503         scandir.o               \
504         seekdir.o               \
505         select.o                \
506         setlabel.o              \
507         setpriority.o           \
508         settimeofday.o          \
509         sh_locks.o              \
510         sigflag.o               \
511         siglist.o               \
512         sigsend.o               \
513         sigsetops.o             \
514         ssignal.o               \
515         stack.o                 \
516         stpcpy.o                \
517         stpncpy.o               \
518         str2sig.o               \
519         strcase_charmap.o       \
520         strcat.o                \
521         strchr.o                \
522         strchrnul.o             \
```

```
523         strcspn.o               \
524         strdup.o                \
525         strerror.o              \
526         strlcat.o               \
527         strlcpy.o               \
528         strncat.o               \
529         strndup.o               \
530         strpbrk.o               \
531         strrchr.o               \
532         strsep.o                \
533         strsignal.o             \
534         strspn.o                \
535         strstr.o                \
536         strtod.o                \
537         strtoimax.o             \
538         strtok.o                \
539         strtok_r.o              \
540         strtoumax.o             \
541         swab.o                  \
542         swapctl.o               \
543         sysconf.o               \
544         syslog.o                \
545         tcdrain.o               \
546         tcflow.o                \
547         tcflush.o               \
548         tcgetattr.o             \
549         tcgetpgrp.o             \
550         tcgetsid.o              \
551         tcsendbreak.o           \
552         tcsetattr.o             \
553         tcsetpgrp.o             \
554         tell.o                  \
555         telldir.o               \
556         tfind.o                 \
557         time_data.o             \
558         time_gdata.o            \
559         tls_data.o              \
560         truncate.o              \
561         tsdalloc.o              \
562         tsearch.o               \
563         ttyname.o               \
564         ttyslot.o               \
565         ualarm.o                \
566         ucred.o                 \
567         valloc.o                \
568         vlfmt.o                 \
569         vpfmt.o                 \
570         waitpid.o               \
571         walkstack.o             \
572         wdata.o                 \
573         xgetwidth.o             \
574         xpg4.o                  \
575         xpg6.o

577 PORTPRINT_W=                   \
578         doprnt_w.o

580 PORTPRINT=                     \
581         asprintf.o              \
582         doprnt.o                \
583         fprintf.o               \
584         printf.o                \
585         snprintf.o              \
586         sprintf.o               \
587         vfprintf.o              \
588         vprintf.o               \
```



```
589         vsnprintf.o             \
590         vsprintf.o              \
591         vwprintf.o              \
592         wprintf.o

594 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
595 # This macro should ALWAYS be empty; native APIs are already 'large file'.
596 PORTSTDIO64=

598 PORTSTDIO_W=                    \
599         doscan_w.o

601 PORTSTDIO=                      \
602         __extensions.o          \
603         _endopen.o              \
604         _filbuf.o               \
605         _findbuf.o              \
606         _flsbuf.o               \
607         _wrtchk.o               \
608         clearerr.o              \
609         ctermid.o               \
610         ctermid_r.o             \
611         cuserid.o               \
612         data.o                  \
613         doscan.o                \
614         fdopen.o                \
615         feof.o                  \
616         ferror.o                \
617         fgetc.o                 \
618         fgets.o                 \
619         fileno.o                \
620         flockf.o                \
621         flush.o                 \
622         fopen.o                 \
623         fpos.o                  \
624         fputc.o                 \
625         fputs.o                 \
626         fread.o                 \
627         fseek.o                 \
628         fseeko.o                \
629         ftell.o                 \
630         ftello.o                \
631         fwrite.o                \
632         getc.o                  \
633         getchar.o               \
634         getline.o               \
635         getpass.o               \
636         gets.o                  \
637         getw.o                  \
638         mse.o                   \
639         popen.o                 \
640         putc.o                  \
641         putchar.o               \
642         puts.o                  \
643         putw.o                  \
644         rewind.o                \
645         scanf.o                 \
646         setbuf.o                \
647         setbuffer.o             \
648         setvbuf.o               \
649         system.o                \
650         tempnam.o               \
651         tmpfile.o               \
652         tmpnam_r.o              \
653         ungetc.o                \
654         vscanf.o                \
```

```
655         vwscanf.o               \
656         wscanf.o

658 PORTI18N=                       \
659         getwchar.o              \
660         putwchar.o              \
661         putws.o                 \
662         strtows.o               \
663         wcsnlen.o               \
664         wcsstr.o                \
665         wcstoimax.o             \
666         wcstol.o                \
667         wcstoul.o               \
668         wcswcs.o                \
669         wmemchr.o               \
670         wmemcmp.o               \
671         wmemcpy.o               \
672         wmemmove.o              \
673         wmemset.o               \
674         wscat.o                 \
675         wschr.o                 \
676         wscmp.o                 \
677         wscpy.o                 \
678         wscspn.o                \
679         wsdup.o                 \
680         wslen.o                 \
681         wsncat.o                \
682         wsncmp.o                \
683         wsncpy.o                \
684         wspbrk.o                \
685         wsprintf.o              \
686         wsrchr.o                \
687         wsscanf.o               \
688         wsspn.o                 \
689         wstod.o                 \
690         wstok.o                 \
691         wstol.o                 \
692         wstoll.o                \
693         wsxfrm.o                \
694         gettext.o               \
695         gettext_gnu.o           \
696         gettext_real.o          \
697         gettext_util.o          \
698         plural_parser.o         \
699         wdresolve.o             \
700         _ctype.o                \
701         isascii.o               \
702         toascii.o

704 PORTI18N_COND=                  \
705         wcstol_longlong.o       \
706         wcstoul_longlong.o

708 PORTLOCALE=                    \
709         big5.o                  \
710         btowc.o                 \
711         collate.o               \
712         collcmp.o               \
713         euc.o                   \
714         fnmatch.o               \
715         fgetwc.o                \
716         fgetws.o                \
717         fix_grouping.o          \
718         fputwc.o                \
719         fputws.o                \
720         fwide.o                 \
```



```
721         gb18030.o               \
722         gb2312.o                \
723         gbk.o                   \
724         getdate.o               \
725         isdigit.o               \
726         iswctype.o              \
727         ldpart.o                \
728         lmessages.o             \
729         lnumeric.o              \
730         lmonetary.o             \
731         localeconv.o            \
732         localeimpl.o            \
733         mbftowc.o               \
734         mblen.o                 \
735         mbrlen.o                \
736         mbrtowc.o               \
737         mbsinit.o               \
738         mbsnrtowcs.o            \
739         mbsrtowcs.o             \
740         mbstowcs.o              \
741         mbtowc.o                \
742         mskanji.o               \
743         nextwctype.o            \
744         nl_langinfo.o           \
745         none.o                  \
746         regcomp.o               \
747         regfree.o               \
748         regerror.o              \
749         regexec.o               \
750         rune.o                  \
751         runetype.o              \
752         setlocale.o             \
753         setrunelocale.o         \
754         strcasecmp.o            \
755         strcasestr.o            \
756         strcoll.o               \
757         strfmon.o               \
758         strftime.o              \
759         strncasecmp.o           \
760         strptime.o              \
761         strxfrm.o               \
762         table.o                 \
763         timelocal.o             \
764         tolower.o               \
765         towlower.o              \
766         ungetwc.o               \
767         utf8.o                  \
768         wcrtomb.o               \
769         wcscasecmp.o            \
770         wcscoll.o               \
771         wcsftime.o              \
772         wcsnrtombs.o            \
773         wcsrtombs.o             \
774         wcswidth.o              \
775         wcstombs.o              \
776         wcsxfrm.o               \
777         wctob.o                 \
778         wctomb.o                \
779         wctrans.o               \
780         wctype.o                \
781         wcwidth.o               \
782         wscol.o

784 AIOOBJS=                       \
785         aio.o                   \
786         aio_alloc.o             \
```

```
787         posix_aio.o

789 RTOBJS=                         \
790         clock_timer.o           \
791         mqueue.o                \
792         pos4obj.o               \
793         sched.o                 \
794         sem.o                   \
795         shm.o                   \
796         sigev_thread.o

798 TPOOLOBJS=                       \
799         thread_pool.o

801 THREADSOBJS=                     \
802         alloc.o                 \
803         assfail.o               \
804         cancel.o                \
805         door_calls.o            \
806         tmem.o                  \
807         pthr_attr.o             \
808         pthr_barrier.o          \
809         pthr_cond.o             \
810         pthr_mutex.o            \
811         pthr_rwlock.o           \
812         pthread.o               \
813         rwlock.o                \
814         scalls.o                \
815         sema.o                  \
816         sigaction.o             \
817         spawn.o                 \
818         synch.o                 \
819         tdb_agent.o             \
820         thr.o                   \
821         thread_interface.o      \
822         tls.o                   \
823         tsd.o

825 THREADSMACHOBJS=                \
826         machdep.o

828 THREADSASMOBJS=                 \
829         asm_subr.o

831 UNICODEOBJS=                    \
832         u8_textprep.o           \
833         uconv.o

835 UNWINDMACHOBJS=                 \
836         call_frame_inst.o       \
837         eh_frame.o              \
838         thrp_unwind.o           \
839         unwind.o

841 pics/unwind.o:= COPTFLAG64 =

843 UNWINDASMOBJS=                  \
844         unwind_frame.o

846 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
847 # This macro should ALWAYS be empty; native APIs are already 'large file'.
848 PORTSYS64=

850 PORTSYS=                        \
851         _autofssys.o            \
852         access.o                \
```



```
853         acctctl.o               \
854         bsd_signal.o            \
855         chmod.o                 \
856         chown.o                 \
857         corectl.o               \
858         exacctsys.o             \
859         execl.o                 \
860         execle.o                \
861         execv.o                 \
862         fcntl.o                 \
863         fexecve.o               \
864         getpagesizes.o          \
865         getpeerucred.o          \
866         inst_sync.o             \
867         issetugid.o             \
868         label.o                 \
869         link.o                  \
870         lockf.o                 \
871         lwp.o                   \
872         lwp_cond.o              \
873         lwp_rwlock.o            \
874         lwp_sigmask.o           \
875         meminfosys.o            \
876         mkdir.o                 \
877         mknod.o                 \
878         msgsys.o                \
879         nfssys.o                \
880         open.o                  \
881         pgrpsys.o               \
882         posix_sigwait.o         \
883         ppriv.o                 \
884         psetsys.o               \
885         rctlsys.o               \
886         readlink.o              \
887         rename.o                \
888         sbrk.o                  \
889         semsys.o                \
890         set_errno.o             \
891         sharefs.o               \
892         shmsys.o                \
893         sidsys.o                \
894         siginterrupt.o          \
895         signal.o                \
896         sigpending.o            \
897         sigstack.o              \
898         stat.o                  \
899         symlink.o               \
900         tasksys.o               \
901         time.o                  \
902         time_util.o             \
903         ucontext.o              \
904         unlink.o                \
905         ustat.o                 \
906         utimesys.o              \
907         zone.o

909 PORTREGEX=                      \
910         glob.o                  \
911         regcmp.o                \
912         regex.o                 \
913         wordexp.o

915 VALUES=                        \
916         values-Xa.o

918 MOSTOBJS=                       \
```

```
 919         $(STRETS)               \
 920         $(CRTOBJS)              \
 921         $(DYNOBJS)              \
 922         $(FPOBJS)               \
 923         $(I386FPOBJS)           \
 924         $(FPASMOBJS)            \
 925         $(ATOMICOBJS)           \
 926         $(XATTROBJS)            \
 927         $(COMOBJS)              \
 928         $(GENOBJS)              \
 929         $(PORTFP)               \
 930         $(PORTGEN)              \
 931         $(PORTGEN64)            \
 932         $(PORTI18N)             \
 933         $(PORTI18N_COND)        \
 934         $(PORTLOCALE)           \
 935         $(PORTPRINT)            \
 936         $(PORTPRINT_W)          \
 937         $(PORTREGEX)            \
 938         $(PORTSTDIO)            \
 939         $(PORTSTDIO64)          \
 940         $(PORTSTDIO_W)          \
 941         $(PORTSYS)              \
 942         $(PORTSYS64)            \
 943         $(AIOOBJS)              \
 944         $(RTOBJS)               \
 945         $(TPOOLOBJS)            \
 946         $(THREADSOBJS)          \
 947         $(THREADSMACHOBJS)      \
 948         $(THREADSASMOBJS)       \
 949         $(UNICODEOBJS)          \
 950         $(UNWINDMACHOBJS)       \
 951         $(UNWINDASMOBJS)        \
 952         $(COMSYSOBJS)           \
 953         $(SYSOBJS)              \
 954         $(COMSYSOBJS64)         \
 955         $(SYSOBJS64)            \
 956         $(VALUES)

 958 TRACEOBJS=                     \
 959         plockstat.o

 961 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
 962 # modules whose source is provided in the $(SRC)/lib/common directory.
 963 # This must be done because otherwise the Sun C compiler would insert
 964 # its own versions of these modules and those versions contain code
 965 # to call out to C++ initialization functions.  Such C++ initialization
 966 # functions can call back into libc before thread initialization is
 967 # complete and this leads to segmentation violations and other problems.
 968 # Since libc contains no C++ code, linking with the minimal crti.o and
 969 # crtn.o modules is safe and avoids the problems described above.
 970 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
 971 CRTSRCS= ../../common/amd64

 973 # include common library definitions
 974 include ../../Makefile.lib
 975 include ../../Makefile.lib.64

 977 CFLAGS64 += $(CTF_FLAGS)

 979 # This is necessary to avoid problems with calling _ex_unwind().
 980 # We probably don't want any inlining anyway.
 981 CFLAGS64 += -xinline=

 983 CERRWARN += -_gcc=-Wno-parentheses
 984 CERRWARN += -_gcc=-Wno-switch
```



```
 985 CERRWARN += -_gcc=-Wno-uninitialized
 986 CERRWARN += -_gcc=-Wno-unused-value
 987 CERRWARN += -_gcc=-Wno-unused-label
 988 CERRWARN += -_gcc=-Wno-unused-variable
 989 CERRWARN += -_gcc=-Wno-type-limits
 990 CERRWARN += -_gcc=-Wno-char-subscripts
 991 CERRWARN += -_gcc=-Wno-clobbered
 992 CERRWARN += -_gcc=-Wno-unused-function
 993 CERRWARN += -_gcc=-Wno-address

 995 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
 996 # enables ASSERT() checking in the threads portion of the library.
 997 # This is automatically enabled for DEBUG builds, not for non-debug builds.
 998 THREAD_DEBUG =
 999 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1001 # Make string literals read-only to save memory
1002 CFLAGS64 += $(XSTRCONST)

1004 ALTPICS= $(TRACEOBJS:%=pics/%)

1006 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1008 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1010 CPPFLAGS=       -D_REENTRANT -D$(MACH64) -D__$(MACH64) $(THREAD_DEBUG) \
1011                 -I. -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1012 ASFLAGS=       $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) \
1013                 $(amd64_AS_XARCH)

1015 # As a favor to the dtrace syscall provider, libc still calls the
1016 # old syscall traps that have been obsoleted by the *at() interfaces.
1017 # Delete this to compile libc using only the new *at() system call traps
1018 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1020 # proc64_id.o is built with defines in $(SRC)/uts/intel/sys/x86_archext.h
1021 pics/proc64_id.o        := CFLAGS64 += -I$(SRC)/uts/intel

1023 # Inform the run-time linker about libc specialized initialization
1024 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1025 DYNFLAGS +=     $(RTLDINFO)

1027 # Force libc's internal references to be resolved immediately upon loading
1028 # in order to avoid critical region problems.  Since almost all libc symbols
1029 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1030 DYNFLAGS +=     -znow

1032 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1034 # Override this top level flag so the compiler builds in its native
1035 # C99 mode.  This has been enabled to support the complex arithmetic
1036 # added to libc.
1037 C99MODE=        $(C99_ENABLE)

1039 # libc method of building an archive
1040 # The "$(GREP) -v ' L '" part is necessary only until
1041 # lorder is fixed to ignore thread-local variables.
1042 BUILD.AR= $(RM) $@ ; \
1043         $(AR) q $@ `$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1045 # extra files for the clean target
1046 CLEANFILES=                     \
1047         $(LIBCDIR)/port/gen/errlst.c    \
1048         $(LIBCDIR)/port/gen/new_list.c  \
1049         assym.h                 \
1050         genassym                \
```

```
1051         crt/_rtld.s             \
1052         pics/crti.o             \
1053         pics/crtn.o             \
1054         $(ALTPICS)

1056 CLOBBERFILES += $(LIB_PIC)

1058 # list of C source for lint
1059 SRCS=                                                   \
1060         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1061         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1062         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1063         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1064         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1065         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1066         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1067         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1068         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1069         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1070         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1071         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1072         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1073         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1074         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1075         $(THREADSMACHOBJS:%.o=threads/%.c)              \
1076         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1077         $(UNWINDMACHOBJS:%.o=unwind/%.c)                \
1078         $(FPOBJS:%.o=fp/%.c)                            \
1079         $(I386FPOBJS:%.o=$(LIBCDIR)/i386/fp/%.c)                \
1080         $(LIBCBASE)/gen/ecvt.c                          \
1081         $(LIBCBASE)/gen/makectxt.c                      \
1082         $(LIBCBASE)/gen/siginfolst.c                    \
1083         $(LIBCBASE)/gen/siglongjmp.c                    \
1084         $(LIBCBASE)/gen/sync_instruction_memory.c       \
1085         $(LIBCBASE)/sys/uadmin.c

1087 # conditional assignments
1088 # $(DYNLIB) $(LIB_PIC) := DYNOBJS = _rtbootld.o
1089 $(DYNLIB) := CRTI = crti.o
1090 $(DYNLIB) := CRTN = crtn.o

1092 # Files which need the threads .il inline template
1093 TIL=                            \
1094         aio.o                   \
1095         alloc.o                 \
1096         assfail.o               \
1097         atexit.o                \
1098         atfork.o                \
1099         cancel.o                \
1100         door_calls.o            \
1101         err.o                   \
1102         errno.o                 \
1103         lwp.o                   \
1104         ma.o                    \
1105         machdep.o               \
1106         posix_aio.o             \
1107         pthr_attr.o             \
1108         pthr_barrier.o          \
1109         pthr_cond.o             \
1110         pthr_mutex.o            \
1111         pthr_rwlock.o           \
1112         pthread.o               \
1113         rand.o                  \
1114         rwlock.o                \
1115         scalls.o                \
1116         sched.o                 \
```



```
1117         sema.o                  \
1118         sigaction.o             \
1119         sigev_thread.o          \
1120         spawn.o                 \
1121         stack.o                 \
1122         synch.o                 \
1123         tdb_agent.o             \
1124         thr.o                   \
1125         thread_interface.o      \
1126         thread_pool.o           \
1127         thrp_unwind.o           \
1128         tls.o                   \
1129         tmem.o                  \
1130         tsd.o

1132 $(TIL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/threads/amd64.il

1134 # pics/mul64.o := CFLAGS64 += crt/mul64.il

1136 # large-file-aware components that should be built large

1138 #$(COMSYSOBJS64:%=pics/%) := \
1139 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1141 #$(SYSOBJS64:%=pics/%) := \
1142 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1144 #$(PORTGEN64:%=pics/%) := \
1145 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1147 #$(PORTSTDIO64:%=pics/%) := \
1148 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1150 #$(PORTSYS64:%=pics/%) := \
1151 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1153 $(PORTSTDIO_W:%=pics/%) := \
1154         CPPFLAGS += -D_WIDE

1156 $(PORTPRINT_W:%=pics/%) := \
1157         CPPFLAGS += -D_WIDE

1159 $(PORTPRINT_C89:%=pics/%) := \
1160         CPPFLAGS += -D_C89_INTMAX32

1162 $(PORTSTDIO_C89:%=pics/%) := \
1163         CPPFLAGS += -D_C89_INTMAX32

1165 $(PORTI18N_COND:%=pics/%) := \
1166         CPPFLAGS += -D_WCS_LONGLONG

1168 .KEEP_STATE:

1170 all: $(LIBS) $(LIB_PIC)

1172 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1173 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1174 lint    :=      LINTFLAGS64 += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1176 lint:
1177         @echo $(LINT.c) ... $(LDLIBS)
1178         @$(LINT.c) $(SRCS) $(LDLIBS)

1180 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1181 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1182 $(LINTLIB):= LINTFLAGS64=-nvx -m64
```

```
# object files that depend on inline template
$(TIL:%=pics/%): $(LIBCBASE)/threads/amd64.il
# pics/mul64.o: crt/mul64.il

# include common libc targets
include ../Makefile.targ

# We need to strip out all CTF data from the static library
$(LIB_PIC) := DIR = pics
$(LIB_PIC): pics $$(PICS)
        $(BUILD.AR)
        $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
        $(AR) -ts $@ > /dev/null
        $(POST_PROCESS_A)

ASSYMDEP_OBJS=                  \
        _lwp_mutex_unlock.o     \
        _stack_grow.o           \
        asm_subr.o              \
        getcontext.o            \
        setjmp.o                \
        tls_get_addr.o          \
        vforkx.o

$(ASSYMDEP_OBJS:%=pics/%): assym.h

# assym.h build rules

GENASSYM_C = genassym.c

genassym: $(GENASSYM_C)
        $(NATIVECC) -Iinc -I$(LIBCDIR)/inc $(CPPFLAGS.native)   \
                -o $@ $(GENASSYM_C)

OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

assym.h: $(OFFSETS) genassym
        $(OFFSETS_CREATE) <$(OFFSETS) >$@
        ./genassym >>$@

# derived C source and related explicit dependencies
$(LIBCDIR)/port/gen/errlst.c + \
$(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
        cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   24113 Mon Apr  6 11:47:16 2015
new/usr/src/lib/libc/i386/Makefile.com
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBCDIR=        $(SRC)/lib/libc
LIB_PIC=        libc_pic.a
VERS=           .1
CPP=            /usr/lib/cpp
TARGET_ARCH=    i386

VALUES=         values-Xa.o

# objects are grouped by source directory

# local objects
STRETS=

CRTOBJS=                        \
        cerror.o                \
        cerror64.o

DYNOBJS=                        \
        _rtbootld.o

FPOBJS=                         \
        _D_cplx_div.o           \
        _D_cplx_div_ix.o        \
        _D_cplx_div_rx.o        \
        _D_cplx_lr_div.o        \
        _D_cplx_lr_div_ix.o     \
        _D_cplx_lr_div_rx.o     \
        _D_cplx_mul.o           \
        _F_cplx_div.o           \
        _F_cplx_div_ix.o        \
```



```
        _F_cplx_div_rx.o        \
        _F_cplx_lr_div.o        \
        _F_cplx_lr_div_ix.o     \
        _F_cplx_lr_div_rx.o     \
        _F_cplx_mul.o           \
        _X_cplx_div.o           \
        _X_cplx_div_ix.o        \
        _X_cplx_div_rx.o        \
        _X_cplx_lr_div.o        \
        _X_cplx_lr_div_ix.o     \
        _X_cplx_lr_div_rx.o     \
        _X_cplx_mul.o           \
        fpgetmask.o             \
        fpgetround.o            \
        fpgetsticky.o           \
        fpsetmask.o             \
        fpsetround.o            \
        fpsetsticky.o           \
        fpstart.o               \
        ieee.o

FPASMOBJS=                      \
        __xgetRD.o              \
        _base_il.o              \
        _xtoll.o                \
        _xtoull.o               \
        fpcw.o

ATOMICOBJS=                     \
        atomic.o

XATTROBJS=                      \
        xattr_common.o

COMOBJS=                        \
        bcmp.o                  \
        bcopy.o                 \
        bsearch.o               \
        bzero.o                 \
        qsort.o                 \
        strtol.o                \
        strtoul.o               \
        strtoll.o               \
        strtoull.o

DTRACEOBJS=                     \
        dtrace_data.o

GENOBJS=                        \
        _div64.o                \
        _divdi3.o               \
        _getsp.o                \
        _mul64.o                \
        abs.o                   \
        alloca.o                \
        byteorder.o             \
        byteorder64.o           \
        cuexit.o                \
        ecvt.o                  \
        errlst.o                \
        i386_data.o             \
        ladd.o                  \
        ldivide.o               \
        lmul.o                  \
        lock.o                  \
        lshiftl.o               \
```

```
127         lsign.o                 \
128         lsub.o                  \
129         makectxt.o              \
130         memccpy.o               \
131         memchr.o                \
132         memcmp.o                \
133         memcpy.o                \
134         memset.o                \
135         new_list.o              \
136         setjmp.o                \
137         siginfolst.o            \
138         siglongjmp.o            \
139         strcat.o                \
140         strchr.o                \
141         strcmp.o                \
142         strcpy.o                \
143         strlen.o                \
144         strncat.o               \
145         strncmp.o               \
146         strncpy.o               \
147         strnlen.o               \
148         strrchr.o               \
149         sync_instruction_memory.o

151 # sysobjs that contain large-file interfaces
152 COMSYSOBJS64=                   \
153         fstatvfs64.o            \
154         getdents64.o            \
155         getrlimit64.o           \
156         lseek64.o               \
157         mmap64.o                \
158         pread64.o               \
159         preadv64.o              \
160         pwrite64.o              \
161         pwritev64.o             \
162         setrlimit64.o           \
163         statvfs64.o

165 SYSOBJS64=

167 COMSYSOBJS=                    \
168         __clock_timer.o         \
169         __getloadavg.o          \
170         __rusagesys.o           \
171         __signotify.o           \
172         __sigrt.o               \
173         __time.o                \
174         _lgrp_home_fast.o       \
175         _lgrpsys.o              \
176         _nfssys.o               \
177         _portfs.o               \
178         _pset.o                 \
179         _rpcsys.o               \
180         _sigaction.o            \
181         _so_accept.o            \
182         _so_bind.o              \
183         _so_connect.o           \
184         _so_getpeername.o       \
185         _so_getsockname.o       \
186         _so_getsockopt.o        \
187         _so_listen.o            \
188         _so_recv.o              \
189         _so_recvfrom.o          \
190         _so_recvmsg.o           \
191         _so_send.o              \
192         _so_sendmsg.o           \
```



```
193         _so_sendto.o            \
194         _so_setsockopt.o        \
195         _so_shutdown.o          \
196         _so_socket.o            \
197         _so_socketpair.o        \
198         _sockconfig.o           \
199         acct.o                  \
200         acl.o                   \
201         adjtime.o               \
202         alarm.o                 \
203         brk.o                   \
204         chdir.o                 \
205         chroot.o                \
206         cladm.o                 \
207         close.o                 \
208         execve.o                \
209         exit.o                  \
210         facl.o                  \
211         fchdir.o                \
212         fchroot.o               \
213         fdsync.o                \
214         fpathconf.o             \
215         fstatfs.o               \
216         fstatvfs.o              \
217         getcpuid.o              \
218         getdents.o              \
219         getegid.o               \
220         geteuid.o               \
221         getgid.o                \
222         getgroups.o             \
223         gethrtime.o             \
224         getitimer.o             \
225         getmsg.o                \
226         getpid.o                \
227         getpmsg.o               \
228         getppid.o               \
229         getrlimit.o             \
230         getuid.o                \
231         gtty.o                  \
232         install_utrap.o         \
233         ioctl.o                 \
234         kaio.o                  \
235         kill.o                  \
236         llseek.o                \
237         lseek.o                 \
238         mmapobjsys.o            \
239         memcntl.o               \
240         mincore.o               \
241         mmap.o                  \
242         modctl.o                \
243         mount.o                 \
244         mprotect.o              \
245         munmap.o                \
246         nice.o                  \
247         ntp_adjtime.o           \
248         ntp_gettime.o           \
249         p_online.o              \
250         pathconf.o              \
251         pause.o                 \
252         pcsample.o              \
253         pipe2.o                 \
254         pollsys.o               \
255         pread.o                 \
256         preadv.o                \
257         priocntlset.o           \
258         processor_bind.o        \
```

```
259         processor_info.o        \
260         profil.o                \
261         putmsg.o                \
262         putpmsg.o               \
263         pwrite.o                \
264         pwritev.o               \
265         read.o                  \
266         readv.o                 \
267         resolvepath.o           \
268         seteguid.o              \
269         setgid.o                \
270         setgroups.o             \
271         setitimer.o             \
272         setreid.o               \
273         setrlimit.o             \
274         setuid.o                \
275         sigaltstk.o             \
276         sigprocmsk.o            \
277         sigsendset.o            \
278         sigsuspend.o            \
279         statfs.o                \
280         statvfs.o               \
281         stty.o                  \
282         sync.o                  \
283         sysconfig.o             \
284         sysfs.o                 \
285         sysinfo.o               \
286         syslwp.o                \
287         times.o                 \
288         ulimit.o                \
289         umask.o                 \
290         umount2.o               \
291         utssys.o                \
292         uucopy.o                \
293         vhangup.o               \
294         waitid.o                \
295         write.o                 \
296         writev.o                \
297         yield.o

299 SYSOBJS=                       \
300         __clock_gettime.o       \
301         __getcontext.o          \
302         __uadmin.o              \
303         _lwp_mutex_unlock.o     \
304         _stack_grow.o           \
305         door.o                  \
306         forkx.o                 \
307         forkallx.o              \
308         getcontext.o            \
309         gettimeofday.o          \
310         lwp_private.o           \
311         nuname.o                \
312         ptrace.o                \
313         syscall.o               \
314         sysi86.o                \
315         tls_get_addr.o          \
316         uadmin.o                \
317         umount.o                \
318         uname.o                 \
319         vforkx.o                \
320         xstat.o

322 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
323 PORTGEN64=                      \
324         _xftw64.o               \
```



```
325         attropen64.o            \
326         ftw64.o                 \
327         mkstemp64.o             \
328         nftw64.o                \
329         tell64.o                \
330         truncate64.o

332 # objects from source under $(LIBCDIR)/port
333 PORTFP=                         \
334         __flt_decim.o           \
335         __flt_rounds.o          \
336         __tbl_10_b.o            \
337         __tbl_10_h.o            \
338         __tbl_10_s.o            \
339         __tbl_2_b.o             \
340         __tbl_2_h.o             \
341         __tbl_2_s.o             \
342         __tbl_fdq.o             \
343         __tbl_tens.o            \
344         __x_power.o             \
345         _base_sup.o             \
346         aconvert.o              \
347         decimal_bin.o           \
348         double_decim.o          \
349         econvert.o              \
350         fconvert.o              \
351         file_decim.o            \
352         finite.o                \
353         fp_data.o               \
354         func_decim.o            \
355         gconvert.o              \
356         hex_bin.o               \
357         ieee_globals.o          \
358         pack_float.o            \
359         sigfpe.o                \
360         string_decim.o

362 PORTGEN=                        \
363         _env_data.o             \
364         _xftw.o                 \
365         a64l.o                  \
366         abort.o                 \
367         addsev.o                \
368         ascii_strcasecmp.o      \
369         ascii_strncasecmp.o     \
370         assert.o                \
371         atof.o                  \
372         atoi.o                  \
373         atol.o                  \
374         atoll.o                 \
375         attrat.o                \
376         attropen.o              \
377         atexit.o                \
378         atfork.o                \
379         basename.o              \
380         calloc.o                \
381         catgets.o               \
382         catopen.o               \
383         cfgetispeed.o           \
384         cfgetospeed.o           \
385         cfree.o                 \
386         cfsetispeed.o           \
387         cfsetospeed.o           \
388         cftime.o                \
389         clock.o                 \
390         closedir.o              \
```

```
391         closefrom.o             \
392         confstr.o               \
393         crypt.o                 \
394         csetlen.o               \
395         ctime.o                 \
396         ctime_r.o               \
397         daemon.o                \
398         deflt.o                 \
399         directio.o              \
400         dirname.o               \
401         div.o                   \
402         drand48.o               \
403         dup.o                   \
404         env_data.o              \
405         err.o                   \
406         errno.o                 \
407         euclen.o                \
408         event_port.o            \
409         execvp.o                \
410         fattach.o               \
411         fdetach.o               \
412         fdopendir.o             \
413         ffs.o                   \
414         fls.o                   \
415         fmtmsg.o                \
416         ftime.o                 \
417         ftok.o                  \
418         ftw.o                   \
419         gcvt.o                  \
420         getauxv.o               \
421         getcwd.o                \
422         getdate_err.o           \
423         getdtblsize.o           \
424         getenv.o                \
425         getexecname.o           \
426         getgrnam.o              \
427         getgrnam_r.o            \
428         gethostid.o             \
429         gethostname.o           \
430         gethz.o                 \
431         getisax.o               \
432         getloadavg.o            \
433         getlogin.o              \
434         getmntent.o             \
435         getnetgrent.o           \
436         get_nprocs.o            \
437         getopt.o                \
438         getopt_long.o           \
439         getpagesize.o           \
440         getpw.o                 \
441         getpwnam.o              \
442         getpwnam_r.o            \
443         getrusage.o             \
444         getspent.o              \
445         getspent_r.o            \
446         getsubopt.o             \
447         gettxt.o                \
448         getusershell.o          \
449         getut.o                 \
450         getutx.o                \
451         getvfsent.o             \
452         getwd.o                 \
453         getwidth.o              \
454         getxby_door.o           \
455         gtxt.o                  \
456         hsearch.o               \
```



```
457         iconv.o                 \
458         imaxabs.o               \
459         imaxdiv.o               \
460         index.o                 \
461         initgroups.o            \
462         insque.o                \
463         isaexec.o               \
464         isastream.o             \
465         isatty.o                \
466         killpg.o                \
467         klpdlib.o               \
468         l64a.o                  \
469         lckpwdf.o               \
470         lconstants.o            \
471         lexp10.o                \
472         lfind.o                 \
473         lfmt.o                  \
474         lfmt_log.o              \
475         llabs.o                 \
476         lldiv.o                 \
477         llog10.o                \
478         lltostr.o               \
479         localtime.o             \
480         lsearch.o               \
481         madvise.o               \
482         malloc.o                \
483         memalign.o              \
484         memmem.o                \
485         mkdev.o                 \
486         mkdtemp.o               \
487         mkfifo.o                \
488         mkstemp.o               \
489         mktemp.o                \
490         mlock.o                 \
491         mlockall.o              \
492         mon.o                   \
493         msync.o                 \
494         munlock.o               \
495         munlockall.o            \
496         ndbm.o                  \
497         nftw.o                  \
498         nlspath_checks.o        \
499         nsparse.o               \
500         nss_common.o            \
501         nss_dbdefs.o            \
502         nss_deffinder.o         \
503         opendir.o               \
504         opt_data.o              \
505         perror.o                \
506         pfmt.o                  \
507         pfmt_data.o             \
508         pfmt_print.o            \
509         pipe.o                  \
510         plock.o                 \
511         poll.o                  \
512         posix_fadvise.o         \
513         posix_fallocate.o       \
514         posix_madvise.o         \
515         posix_memalign.o        \
516         priocntl.o              \
517         privlib.o               \
518         priv_str_xlate.o        \
519         psiginfo.o              \
520         psignal.o               \
521         pt.o                    \
522         putpwent.o              \
```

```
523         putspent.o              \
524         raise.o                 \
525         rand.o                  \
526         random.o                \
527         rctlops.o               \
528         readdir.o               \
529         readdir_r.o             \
530         realpath.o              \
531         reboot.o                \
532         regexpr.o               \
533         remove.o                \
534         rewinddir.o             \
535         rindex.o                \
536         scandir.o               \
537         seekdir.o               \
538         select.o                \
539         select_large_fdset.o    \
540         setlabel.o              \
541         setpriority.o           \
542         settimeofday.o          \
543         sh_locks.o              \
544         sigflag.o               \
545         siglist.o               \
546         sigsend.o               \
547         sigsetops.o             \
548         ssignal.o               \
549         stack.o                 \
550         stpcpy.o                \
551         stpncpy.o               \
552         str2sig.o               \
553         strcase_charmap.o       \
554         strchrnul.o             \
555         strcspn.o               \
556         strdup.o                \
557         strerror.o              \
558         strlcat.o               \
559         strlcpy.o               \
560         strndup.o               \
561         strpbrk.o               \
562         strsep.o                \
563         strsignal.o             \
564         strspn.o                \
565         strstr.o                \
566         strtod.o                \
567         strtoimax.o             \
568         strtok.o                \
569         strtok_r.o              \
570         strtoumax.o             \
571         swab.o                  \
572         swapctl.o               \
573         sysconf.o               \
574         syslog.o                \
575         tcdrain.o               \
576         tcflow.o                \
577         tcflush.o               \
578         tcgetattr.o             \
579         tcgetpgrp.o             \
580         tcgetsid.o              \
581         tcsendbreak.o           \
582         tcsetattr.o             \
583         tcsetpgrp.o             \
584         tell.o                  \
585         telldir.o               \
586         tfind.o                 \
587         time_data.o             \
588         time_gdata.o            \
```



```
589         tls_data.o              \
590         truncate.o              \
591         tsdalloc.o              \
592         tsearch.o               \
593         ttyname.o               \
594         ttyslot.o               \
595         ualarm.o                \
596         ucred.o                 \
597         valloc.o                \
598         vlfmt.o                 \
599         vpfmt.o                 \
600         waitpid.o               \
601         walkstack.o             \
602         wdata.o                 \
603         xgetwidth.o             \
604         xpg4.o                  \
605         xpg6.o

607 PORTPRINT_W=                    \
608         doprnt_w.o

610 PORTPRINT=                     \
611         asprintf.o              \
612         doprnt.o                \
613         fprintf.o               \
614         printf.o                \
615         snprintf.o              \
616         sprintf.o               \
617         vfprintf.o              \
618         vprintf.o               \
619         vsnprintf.o             \
620         vsprintf.o              \
621         vwprintf.o              \
622         wprintf.o

624 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
625 PORTPRINT_C89=                  \
626         vfprintf_c89.o          \
627         vprintf_c89.o           \
628         vsnprintf_c89.o         \
629         vsprintf_c89.o          \
630         vwprintf_c89.o

632 PORTSTDIO_C89=                 \
633         vscanf_c89.o            \
634         vwscanf_c89.o           \

636 # portable stdio objects that contain large file interfaces.
637 # Note: fopen64 is a special case, as we build it small.
638 PORTSTDIO64=                    \
639         fopen64.o               \
640         fpos64.o

642 PORTSTDIO_W=                    \
643         doscan_w.o

645 PORTSTDIO=                      \
646         __extensions.o          \
647         _endopen.o              \
648         _filbuf.o               \
649         _findbuf.o              \
650         _flsbuf.o               \
651         _wrtchk.o               \
652         clearerr.o              \
653         ctermid.o               \
654         ctermid_r.o             \
```

```
655         cuserid.o               \
656         data.o                  \
657         doscan.o                \
658         fdopen.o                \
659         feof.o                  \
660         ferror.o                \
661         fgetc.o                 \
662         fgets.o                 \
663         fileno.o                \
664         flockf.o                \
665         flush.o                 \
666         fopen.o                 \
667         fpos.o                  \
668         fputc.o                 \
669         fputs.o                 \
670         fread.o                 \
671         fseek.o                 \
672         fseeko.o                \
673         ftell.o                 \
674         ftello.o                \
675         fwrite.o                \
676         getc.o                  \
677         getchar.o               \
678         getline.o               \
679         getpass.o               \
680         gets.o                  \
681         getw.o                  \
682         mse.o                   \
683         popen.o                 \
684         putc.o                  \
685         putchar.o               \
686         puts.o                  \
687         putw.o                  \
688         rewind.o                \
689         scanf.o                 \
690         setbuf.o                \
691         setbuffer.o             \
692         setvbuf.o               \
693         system.o                \
694         tempnam.o               \
695         tmpfile.o               \
696         tmpnam_r.o              \
697         ungetc.o                \
698         vscanf.o                \
699         vwscanf.o               \
700         wscanf.o

702 PORTI18N=                       \
703         getwchar.o              \
704         putwchar.o              \
705         putws.o                 \
706         strtows.o               \
707         wcsnlen.o               \
708         wcsstr.o                \
709         wcstoimax.o             \
710         wcstol.o                \
711         wcstoul.o               \
712         wcswcs.o                \
713         wmemchr.o               \
714         wmemcmp.o               \
715         wmemcpy.o               \
716         wmemmove.o              \
717         wmemset.o               \
718         wscat.o                 \
719         wschr.o                 \
720         wscmp.o                 \
```



```
721         wscpy.o                 \
722         wscspn.o                \
723         wsdup.o                 \
724         wslen.o                 \
725         wsncat.o                \
726         wsncmp.o                \
727         wsncpy.o                \
728         wspbrk.o                \
729         wsprintf.o              \
730         wsrchr.o                \
731         wsscanf.o               \
732         wsspn.o                 \
733         wstod.o                 \
734         wstok.o                 \
735         wstol.o                 \
736         wstoll.o                \
737         wsxfrm.o                \
738         gettext.o               \
739         gettext_gnu.o           \
740         gettext_real.o          \
741         gettext_util.o          \
742         plural_parser.o         \
743         wdresolve.o             \
744         _ctype.o                \
745         isascii.o               \
746         toascii.o

748 PORTI18N_COND=                  \
749         wcstol_longlong.o       \
750         wcstoul_longlong.o

752 PORTLOCALE=                    \
753         big5.o                  \
754         btowc.o                 \
755         collate.o               \
756         collcmp.o               \
757         euc.o                   \
758         fnmatch.o               \
759         fgetwc.o                \
760         fgetws.o                \
761         fix_grouping.o          \
762         fputwc.o                \
763         fputws.o                \
764         fwide.o                 \
765         gb18030.o               \
766         gb2312.o                \
767         gbk.o                   \
768         getdate.o               \
769         isdigit.o               \
770         iswctype.o              \
771         ldpart.o                \
772         lmessages.o             \
773         lnumeric.o              \
774         lmonetary.o             \
775         localeconv.o            \
776         localeimpl.o            \
777         mbftowc.o               \
778         mblen.o                 \
779         mbrlen.o                \
780         mbrtowc.o               \
781         mbsinit.o               \
782         mbsnrtowcs.o            \
783         mbsrtowcs.o             \
784         mbstowcs.o              \
785         mbtowc.o                \
786         mskanji.o               \
```

```
787         nextwctype.o            \
788         nl_langinfo.o           \
789         none.o                  \
790         regcomp.o               \
791         regfree.o               \
792         regerror.o              \
793         regexec.o               \
794         rune.o                  \
795         runetype.o              \
796         setlocale.o             \
797         setrunelocale.o         \
798         strcasecmp.o            \
799         strcasestr.o            \
800         strcoll.o               \
801         strfmon.o               \
802         strftime.o              \
803         strncasecmp.o           \
804         strptime.o              \
805         strxfrm.o               \
806         table.o                 \
807         timelocal.o             \
808         tolower.o               \
809         towlower.o              \
810         ungetwc.o               \
811         utf8.o                  \
812         wcrtomb.o               \
813         wcscasecmp.o            \
814         wcscoll.o               \
815         wcsftime.o              \
816         wcsnrtombs.o            \
817         wcsrtombs.o             \
818         wcswidth.o              \
819         wcstombs.o              \
820         wcsxfrm.o               \
821         wctob.o                 \
822         wctomb.o                \
823         wctrans.o               \
824         wctype.o                \
825         wcwidth.o               \
826         wscol.o

828 AIOOBJS=                       \
829         aio.o                   \
830         aio_alloc.o             \
831         posix_aio.o

833 RTOBJS=                        \
834         clock_timer.o           \
835         mqueue.o                \
836         pos4obj.o               \
837         sched.o                 \
838         sem.o                   \
839         shm.o                   \
840         sigev_thread.o

842 TPOOLOBJS=                     \
843         thread_pool.o

845 THREADSOBJS=                   \
846         alloc.o                 \
847         assfail.o               \
848         cancel.o                \
849         door_calls.o            \
850         tmem.o                  \
851         pthr_attr.o             \
852         pthr_barrier.o          \
```



```
853         pthr_cond.o             \
854         pthr_mutex.o            \
855         pthr_rwlock.o           \
856         pthread.o               \
857         rwlock.o                \
858         scalls.o                \
859         sema.o                  \
860         sigaction.o             \
861         spawn.o                 \
862         synch.o                 \
863         tdb_agent.o             \
864         thr.o                   \
865         thread_interface.o      \
866         tls.o                   \
867         tsd.o

869 THREADSMACHOBJS=               \
870         machdep.o

872 THREADSASMOBJS=                \
873         asm_subr.o

875 UNICODEOBJS=                   \
876         u8_textprep.o           \
877         uconv.o

879 UNWINDMACHOBJS=                \
880         unwind.o

882 UNWINDASMOBJS=                 \
883         unwind_frame.o

885 # objects that implement the transitional large file API
886 PORTSYS64=                      \
887         lockf64.o               \
888         stat64.o

890 PORTSYS=                        \
891         _autofssys.o            \
892         access.o                \
893         acctctl.o               \
894         bsd_signal.o            \
895         chmod.o                 \
896         chown.o                 \
897         corectl.o               \
898         exacctsys.o             \
899         execl.o                 \
900         execle.o                \
901         execv.o                 \
902         fcntl.o                 \
903         fexecve.o               \
904         getpagesizes.o          \
905         getpeerucred.o          \
906         inst_sync.o             \
907         issetugid.o             \
908         label.o                 \
909         link.o                  \
910         lockf.o                 \
911         lwp.o                   \
912         lwp_cond.o              \
913         lwp_rwlock.o            \
914         lwp_sigmask.o           \
915         meminfosys.o            \
916         mkdir.o                 \
917         mknod.o                 \
918         msgsys.o                \
```

```
 919         nfssys.o                \
 920         open.o                  \
 921         pgrpsys.o               \
 922         posix_sigwait.o         \
 923         ppriv.o                 \
 924         psetsys.o               \
 925         rctlsys.o               \
 926         readlink.o              \
 927         rename.o                \
 928         sbrk.o                  \
 929         semsys.o                \
 930         set_errno.o             \
 931         sharefs.o               \
 932         shmsys.o                \
 933         sidsys.o                \
 934         siginterrupt.o          \
 935         signal.o                \
 936         sigpending.o            \
 937         sigstack.o              \
 938         stat.o                  \
 939         symlink.o               \
 940         tasksys.o               \
 941         time.o                  \
 942         time_util.o             \
 943         ucontext.o              \
 944         unlink.o                \
 945         ustat.o                 \
 946         utimesys.o              \
 947         zone.o

 949 PORTREGEX=                      \
 950         glob.o                  \
 951         regcmp.o                \
 952         regex.o                 \
 953         wordexp.o

 955 MOSTOBJS=                       \
 956         $(STRETS)               \
 957         $(CRTOBJS)              \
 958         $(DYNOBJS)              \
 959         $(FPOBJS)               \
 960         $(FPASMOBJS)            \
 961         $(ATOMICOBJS)           \
 962         $(XATTROBJS)            \
 963         $(COMOBJS)              \
 964         $(DTRACEOBJS)           \
 965         $(GENOBJS)              \
 966         $(PORTFP)               \
 967         $(PORTGEN)              \
 968         $(PORTGEN64)            \
 969         $(PORTI18N)             \
 970         $(PORTI18N_COND)        \
 971         $(PORTLOCALE)           \
 972         $(PORTPRINT)            \
 973         $(PORTPRINT_C89)        \
 974         $(PORTPRINT_W)          \
 975         $(PORTREGEX)            \
 976         $(PORTSTDIO)            \
 977         $(PORTSTDIO64)          \
 978         $(PORTSTDIO_C89)        \
 979         $(PORTSTDIO_W)          \
 980         $(PORTSYS)              \
 981         $(PORTSYS64)            \
 982         $(AIOOBJS)              \
 983         $(RTOBJS)               \
 984         $(TPOOLOBJS)            \
```



```
 985         $(THREADSOBJS)          \
 986         $(THREADSMACHOBJS)      \
 987         $(THREADSASMOBJS)       \
 988         $(UNICODEOBJS)          \
 989         $(UNWINDMACHOBJS)       \
 990         $(UNWINDASMOBJS)        \
 991         $(COMSYSOBJS)           \
 992         $(SYSOBJS)              \
 993         $(COMSYSOBJS64)         \
 994         $(SYSOBJS64)            \
 995         $(VALUES)

 997 TRACEOBJS=                      \
 998         plockstat.o

1000 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
1001 # modules whose source is provided in the $(SRC)/lib/common directory.
1002 # This must be done because otherwise the Sun C compiler would insert
1003 # its own versions of these modules and those versions contain code
1004 # to call out to C++ initialization functions.  Such C++ initialization
1005 # functions can call back into libc before thread initialization is
1006 # complete and this leads to segmentation violations and other problems.
1007 # Since libc contains no C++ code, linking with the minimal crti.o and
1008 # crtn.o modules is safe and avoids the problems described above.
1009 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
1010 CRTSRCS= ../../common/i386

1012 LDPASS_OFF=     $(POUND_SIGN)

1014 # include common library definitions
1015 include ../../Makefile.lib

1017 # we need to override the default SONAME here because we might
1018 # be building a variant object (still libc.so.1, but different filename)
1019 SONAME = libc.so.1

1021 CFLAGS += $(CCVERBOSE) $(CTF_FLAGS)

1023 # This is necessary to avoid problems with calling _ex_unwind().
1024 # We probably don't want any inlining anyway.
1025 XINLINE = -xinline=
1026 CFLAGS += $(XINLINE)

1028 CERRWARN += -_gcc=-Wno-parentheses
1029 CERRWARN += -_gcc=-Wno-switch
1030 CERRWARN += -_gcc=-Wno-uninitialized
1031 CERRWARN += -_gcc=-Wno-unused-value
1032 CERRWARN += -_gcc=-Wno-unused-label
1033 CERRWARN += -_gcc=-Wno-unused-variable
1034 CERRWARN += -_gcc=-Wno-type-limits
1035 CERRWARN += -_gcc=-Wno-char-subscripts
1036 CERRWARN += -_gcc=-Wno-clobbered
1037 CERRWARN += -_gcc=-Wno-unused-function
1038 CERRWARN += -_gcc=-Wno-address

1040 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1041 # enables ASSERT() checking in the threads portion of the library.
1042 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1043 THREAD_DEBUG =
1044 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1046 # Make string literals read-only to save memory.
1047 CFLAGS += $(XSTRCONST)

1049 ALTPICS= $(TRACEOBJS:%=pics/%)
```

```
1051 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) \
1052                 $(EXTPICS) $(LDLIBS)

1054 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1056 #
1057 # EXTN_CPPFLAGS and EXTN_CFLAGS set in enclosing Makefile
1058 #
1059 CFLAGS +=       $(EXTN_CFLAGS)
1060 CPPFLAGS=       -D_REENTRANT -Di386 $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1061                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1062 ASFLAGS=        $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) $(i386_AS_XARCH)

1064 # As a favor to the dtrace syscall provider, libc still calls the
1065 # old syscall traps that have been obsoleted by the *at() interfaces.
1066 # Delete this to compile libc using only the new *at() system call traps
1067 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1069 # Inform the run-time linker about libc specialized initialization
1070 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1071 DYNFLAGS +=     $(RTLDINFO)

1073 # Force libc's internal references to be resolved immediately upon loading
1074 # in order to avoid critical region problems.  Since almost all libc symbols
1075 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1076 DYNFLAGS +=     -znow

1078 DYNFLAGS +=     -e __rtboot
1079 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1081 # Inform the kernel about the initial DTrace area (in case
1082 # libc is being used as the interpreter / runtime linker).
1083 DTRACE_DATA =   -zdtrace=dtrace_data
1084 DYNFLAGS +=     $(DTRACE_DATA)

1086 # DTrace needs an executable data segment.
1087 MAPFILE.NED=

1089 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1091 # Override this top level flag so the compiler builds in its native
1092 # C99 mode.  This has been enabled to support the complex arithmetic
1093 # added to libc.
1094 C99MODE=        $(C99_ENABLE)

1096 # libc method of building an archive
1097 # The "$(GREP) -v ' L '" part is necessary only until
1098 # lorder is fixed to ignore thread-local variables.
1099 BUILD.AR= $(RM) $@ ; \
1100         $(AR) q $@ `$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1102 # extra files for the clean target
1103 CLEANFILES=                     \
1104         $(LIBCDIR)/port/gen/errlst.c    \
1105         $(LIBCDIR)/port/gen/new_list.c  \
1106         assym.h                 \
1107         genassym                \
1108         crt/_rtld.s             \
1109         crt/_rtbootld.s         \
1110         pics/_rtbootld.o        \
1111         pics/crti.o             \
1112         pics/crtn.o             \
1113         $(ALTPICS)

1115 CLOBBERFILES += $(LIB_PIC)
```



```
1117 # list of C source for lint
1118 SRCS=                                                   \
1119         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1120         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1121         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1122         $(DTRACEOBJS:%.o=$(SRC)/common/dtrace/%.c)      \
1123         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1124         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1125         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1126         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1127         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1128         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1129         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1130         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1131         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1132         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1133         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1134         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1135         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1136         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1137         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1138         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1139         $(LIBCBASE)/gen/ecvt.c                          \
1140         $(LIBCBASE)/gen/makectxt.c                      \
1141         $(LIBCBASE)/gen/siginfolst.c                    \
1142         $(LIBCBASE)/gen/siglongjmp.c                    \
1143         $(LIBCBASE)/gen/strcmp.c                        \
1144         $(LIBCBASE)/gen/sync_instruction_memory.c       \
1145         $(LIBCBASE)/sys/ptrace.c                        \
1146         $(LIBCBASE)/sys/uadmin.c

1148 # conditional assignments
1149 $(DYNLIB) := CRTI = crti.o
1150 $(DYNLIB) := CRTN = crtn.o

1152 # Files which need the threads .il inline template
1153 TIL=                            \
1154         aio.o                   \
1155         alloc.o                 \
1156         assfail.o               \
1157         atexit.o                \
1158         atfork.o                \
1159         cancel.o                \
1160         door_calls.o            \
1161         err.o                   \
1162         errno.o                 \
1163         lwp.o                   \
1164         ma.o                    \
1165         machdep.o               \
1166         posix_aio.o             \
1167         pthr_attr.o             \
1168         pthr_barrier.o          \
1169         pthr_cond.o             \
1170         pthr_mutex.o            \
1171         pthr_rwlock.o           \
1172         pthread.o               \
1173         rand.o                  \
1174         rwlock.o                \
1175         scalls.o                \
1176         sched.o                 \
1177         sema.o                  \
1178         sigaction.o             \
1179         sigev_thread.o          \
1180         spawn.o                 \
1181         stack.o                 \
1182         synch.o                 \
```

```
1183         tdb_agent.o             \
1184         thr.o                   \
1185         thread_interface.o      \
1186         thread_pool.o           \
1187         tls.o                   \
1188         tsd.o                   \
1189         tmem.o                  \
1190         unwind.o

1192 THREADS_INLINES = $(LIBCBASE)/threads/i386.il
1193 $(TIL:%=pics/%) := CFLAGS += $(THREADS_INLINES)

1195 # pics/mul64.o := CFLAGS += $(LIBCBASE)/crt/mul64.il

1197 # large-file-aware components that should be built large

1199 $(COMSYSOBJS64:%=pics/%) := \
1200         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1202 $(SYSOBJS64:%=pics/%) := \
1203         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1205 $(PORTGEN64:%=pics/%) := \
1206         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1208 $(PORTSTDIO64:%=pics/%) := \
1209         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1211 $(PORTSYS64:%=pics/%) := \
1212         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1214 $(PORTSTDIO_W:%=pics/%) := \
1215         CPPFLAGS += -D_WIDE

1217 $(PORTPRINT_W:%=pics/%) := \
1218         CPPFLAGS += -D_WIDE

1220 $(PORTPRINT_C89:%=pics/%) := \
1221         CPPFLAGS += -D_C89_INTMAX32

1223 $(PORTSTDIO_C89:%=pics/%) := \
1224         CPPFLAGS += -D_C89_INTMAX32

1226 $(PORTI18N_COND:%=pics/%) := \
1227         CPPFLAGS += -D_WCS_LONGLONG

1229 .KEEP_STATE:

1231 all: $(LIBS) $(LIB_PIC)

1233 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1234 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1235 lint    :=      LINTFLAGS += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1237 lint:
1238         @echo $(LINT.c) ...
1239         @$(LINT.c) $(SRCS) $(LDLIBS)

1241 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1242 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1243 $(LINTLIB):= LINTFLAGS=-nvx

1245 # object files that depend on inline template
1246 $(TIL:%=pics/%): $(LIBCBASE)/threads/i386.il
1247 # pics/mul64.o: $(LIBCBASE)/crt/mul64.il
```



```
1249 # include common libc targets
1250 include $(LIBCDIR)/Makefile.targ

1252 # We need to strip out all CTF and DOF data from the static library
1253 $(LIB_PIC) := DIR = pics
1254 $(LIB_PIC): pics $$(PICS)
1255         $(BUILD.AR)
1256         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1257         $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1258         $(AR) -ts $@ > /dev/null
1259         $(POST_PROCESS_A)

1261 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.c
1262         $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S $(C_PICFLAGS) \
1263             $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1264         $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
1265         $(RM) $(LIBCBASE)/crt/_rtld.s

1267 # partially built from C source
1268 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1269         $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
1270         $(CTFCONVERT_O)

1272 ASSYMDEP_OBJS=                  \
1273         _lwp_mutex_unlock.o     \
1274         _stack_grow.o           \
1275         getcontext.o            \
1276         setjmp.o                \
1277         tls_get_addr.o          \
1278         vforkx.o

1280 $(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

1282 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1284 # assym.h build rules

1286 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1288 genassym: $(GENASSYM_C)
1289         $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc  \
1290                 -D__EXTENSIONS__ $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1292 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1294 assym.h: $(OFFSETS) genassym
1295         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1296         ./genassym >>$@

1298 # derived C source and related explicit dependencies
1299 $(LIBCDIR)/port/gen/errlst.c + \
1300 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1301         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1303 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1305 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   46017 Mon Apr  6 11:47:16 2015
new/usr/src/lib/libc/port/llib-lc
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
/*
 * 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
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 */

/* 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
#include <xlocale.h>

/*
 * 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);
```

```
392 /* ecvt.c */
393 char *ecvt(double value, int ndigit, int *_RESTRICT_KYWD decpt,
394         int *_RESTRICT_KYWDsign);
395 char *fcvt(double value, int ndigit, int *_RESTRICT_KYWD decpt,
396         int *_RESTRICT_KYWD sign);

398 /* err.c */
399 void _errfp(FILE *, int, const char *, ...);
400 void _verrfp(FILE *, int, const char *, va_list);
401 void _errxfp(FILE *, int, const char *, ...);
402 void _verrxfp(FILE *, int, const char *, va_list);
403 void _warnfp(FILE *, const char *, ...);
404 void _vwarnfp(FILE *, const char *, va_list);
405 void _warnxfp(FILE *, const char *, ...);
406 void _vwarnxfp(FILE *, const char *, va_list);

408 /* errlst.c */

410 /* euclen.c */
411 int euccol(const unsigned char *s);
412 int euclen(const unsigned char *s);
413 int eucscol(const unsigned char *s);

415 /* execvp.c */
416 /* VARARGS1 */
417 int execlp(const char *, const char *, ...);
418 int execvp(const char *name, char *const *argv);

420 /* fattach.c */
421 int fattach(int fildes, const char *path);

423 /* fdetach.c */
424 int fdetach(const char *path);

426 /* fexecve.c */
427 int fexecve(int, char *const argv[], char *const envp[]);

429 /* ffs.c */
430 int ffs(int field);

432 /* fmtmsg.c */
433 int addseverity(int value, const char *string);
434 int fmtmsg(long class, const char *label, int severity, const char *text,
435                 const char *action, const char *tag);

437 /* ftime.c */
438 int ftime(struct timeb *tp);

440 /* ftok.c */
441 key_t ftok(const char *path, int id);

443 /* gcvt.c */
444 char *gcvt(double number, int ndigit, char *buf);

446 /* getcwd.c */
447 char *getcwd(char *str, size_t size);

449 /* getdate.c */
450 struct tm *getdate(const char *expression);
451 #ifdef getdate_err
452 #undef getdate_err
453 #endif
454 int getdate_err;

456 /* getdate_data.c */
```



```
458 /* getdate_gd.c */

460 /* getdtblsize.c */
461 int getdtablesize(void);

463 /* getenv.c */
464 char *getenv(const char *name);

466 /* getexecname.c */
467 const char *getexecname(void);

469 /* getgrnam.c */
470 struct group *getgrnam(const char *name);
471 struct group *getgrgid(gid_t gid);
472 struct group *fgetgrent_r(FILE *, struct group *, char *, int);
473 struct group *getgrent_r(struct group *, char *, int);
474 struct group *getgrgid_r(gid_t, struct group *, char *, int);
475 struct group *getgrnam_r(const char *, struct group *, char *, int);

477 /* gethostid.c */
478 long gethostid(void);

480 /* gethz.c */
481 int gethz(void);

483 /* getisax.c */
484 uint_t getisax(uint32_t *, uint_t);

486 /* getlogin.c */
487 char *getloginx(void);
488 char *getloginx_r(char *, int);
489 #ifdef  getlogin
490 #undef  getlogin
491 #endif  /* getlogin */
492 char *getlogin(void);
493 #ifdef  getlogin_r
494 #undef  getlogin_r
495 #endif  /* getlogin_r */
496 char *getlogin_r(char *, int);

498 /* getmntent.c */
499 int getmntany(FILE *fd, struct mnttab *mgetp, struct mnttab *mrefp);
500 int getmntent(FILE *fd, struct mnttab *mp);

502 /* getnetgrent.c */
503 int setnetgrent(const char *grp);
504 int endnetgrent(void);
505 int getnetgrent(char **machinep, char **namep, char **domainp);

507 /* getopt.c */
508 int getopt(int argc, char *const *argv, const char *opts);

510 /* getopt_long.c */
511 int getopt_clip(int argc, char *const *argv, const char *optstring,
512                 const struct option *long_options, int *long_index);
513 int getopt_long(int argc, char *const *argv, const char *optstring,
514                 const struct option *long_options, int *long_index);
515 int getopt_long_only(int argc, char *const *argv, const char *optstring,
516                 const struct option *long_options, int *long_index);

518 /* getpagesize.c */
519 int getpagesize(void);

521 /* getpw.c */
522 int getpw(uid_t uid, char *buf);
```

```
 524 /* getpwnam.c */
 525 struct passwd *getpwnam(const char *name);
 526 struct passwd *getpwuid(uid_t uid);
 527 struct passwd *fgetpwent_r(FILE *, struct passwd *, char *, int);
 528 struct passwd *getpwent_r(struct passwd *, char *, int);
 529 struct passwd *getpwnam_r(const char *, struct passwd *, char *, int);
 530 struct passwd *getpwuid_r(uid_t, struct passwd *, char *, int);

 532 /* getrusage.c */
 533 int getrusage(int who, struct rusage *rusage);

 535 /* gettimeofday.c */
 536 int gettimeofday(struct timeval *_RESTRICT_KYWD tp, void *_RESTRICT_KYWD);

 538 /* getspent.c */
 539 void setspent(void);
 540 void endspent(void);
 541 struct spwd *getspent(void);
 542 struct spwd *getspent_r(struct spwd *, char *, int);
 543 struct spwd *fgetspent(FILE *f);
 544 struct spwd *fgetspent_r(FILE *, struct spwd *, char *, int);
 545 struct spwd *getspnam(const char *name);
 546 struct spwd *getspnam_r(const char *, struct spwd *, char *, int);
 547 int putspent(const struct spwd *p, FILE *f);

 549 /* getspent_r.c */
 550 int str2spwd(const char *, int, void *, char *, int);

 552 /* getsubopt.c */
 553 int getsubopt(char **optionsp, char *const *tokens, char **valuep);

 555 /* gettxt.c */
 556 char *gettxt(const char *msg_id, const char *dflt_str);

 558 /* getusershell.c */
 559 char *getusershell(void);
 560 void endusershell(void);
 561 void setusershell(void);

 563 /* getut.c */
 564 struct utmp *getutent(void);
 565 struct utmp *getutid(const struct utmp *entry);
 566 struct utmp *getutline(const struct utmp *entry);
 567 struct utmp *pututline(const struct utmp *entry);
 568 void setutent(void);
 569 void endutent(void);
 570 int utmpname(const char *newfile);
 571 void updwtmp(const char *file, struct utmp *ut);
 572 void getutmp(const struct utmpx *utx, struct utmp *ut);
 573 void getutmpx(const struct utmp *ut, struct utmpx *utx);
 574 struct utmp *makeut(struct utmp *utmp);

 576 /* getutx.c */
 577 struct utmpx *getutxent(void);
 578 struct utmpx *getutxid(const struct utmpx *entry);
 579 struct utmpx *getutxline(const struct utmpx *entry);
 580 struct utmpx *pututxline(const struct utmpx *entry);
 581 void setutxent(void);
 582 void endutxent(void);
 583 int utmpxname(const char *newfile);
 584 void updwtmpx(const char *filex, struct utmpx *utx);
 585 struct utmpx *makeutx(const struct utmpx *utmp);
 586 struct utmpx *modutx(const struct utmpx *utp);

 588 /* getvfsent.c */
 589 int getvfsspec(FILE *fd, struct vfstab *vp, char *special);
```



```
 590 int getvfsfile(FILE *fd, struct vfstab *vp, char *mountp);
 591 int getvfsany(FILE *fd, struct vfstab *vgetp, struct vfstab *vrefp);
 592 int getvfsent(FILE *fd, struct vfstab *vp);

 594 /* getwd.c */
 595 char *getwd(char *pathname);

 597 /* getwidth.c */
 598 void getwidth(eucwidth_t *eucstruct);

 600 /* hsearch.c */
 601 int hcreate(size_t size);
 602 void hdestroy(void);
 603 ENTRY *hsearch(ENTRY item, ACTION action);

 605 /* iconv.c  */
 606 size_t iconv(iconv_t cd, const char **_RESTRICT_KYWD inbuf,
 607     size_t *_RESTRICT_KYWD inbytesleft, char **_RESTRICT_KYWD outbuf,
 608     size_t *_RESTRICT_KYWD outbytesleft);
 609 int iconv_close(iconv_t cd);
 610 iconv_t iconv_open(const char *tocode, const char *fromcode);

 612 /* imaxabs.c */
 613 intmax_t imaxabs(intmax_t j);

 615 /* imaxdiv.c */
 616 imaxdiv_t imaxdiv(intmax_t numer, intmax_t denom);

 618 /* index.c */
 619 char *index(const char *sp, int c);

 621 /* initgroups.c */
 622 int initgroups(const char *uname, gid_t agroup);

 624 /* innetgr.c */
 625 int innetgr(const char *group, const char *machine, const char *name,
 626     const char *domain);

 628 /* insque.c */
 629 void insque(void *elem, void *pred);
 630 void remque(void *elem);

 632 /* isaexec.c */
 633 int isaexec(const char *, char *const *, char *const *);

 635 /* isastream.c */
 636 int isastream(int fd);

 638 /* isatty.c */
 639 int isatty(int f);

 641 /* killpg.c */
 642 int killpg(pid_t pgrp, int sig);

 644 /* l64a.c */
 645 char *l64a(long lg);

 647 /* lckpwdf.c */
 648 int lckpwdf(void);
 649 int ulckpwdf(void);

 651 /* lfind.c */
 652 void * lfind(const void *ky, const void *bs, size_t *nelp,
 653                 size_t width, int (*compar)());

 655 /* localeconv.c */
```

```
656 struct lconv *localeconv(void);

658 /* lsearch.c */
659 void * lsearch(const void *ky, void *bs, size_t *nelp,
660                 size_t width, int (*compar)());

662 /* madvise.c */
663 int madvise(caddr_t addr, size_t len, int advice);

665 /* malloc.c */
666 void *malloc(size_t size);
667 void *realloc(void *old, size_t size);
668 void free(void *old);

670 /* mbstowcs.c */
671 size_t mbstowcs(wchar_t *_RESTRICT_KYWD pwcs, const char *_RESTRICT_KYWD s,
672         size_t n);

674 /* mbtowc.c */
675 int mbtowc(wchar_t *_RESTRICT_KYWD wchar, const char *_RESTRICT_KYWD s,
676         size_t n);
677 int mblen(const char *s, size_t n);

679 /* memalign.c */
680 void *memalign(size_t align, size_t nbytes);

682 /* memccpy.c */
683 void *memccpy(void *_RESTRICT_KYWDs, const void *_RESTRICT_KYWD s0, int c,
684         size_t n);

686 /* memchr.c */
687 void *memchr(const void *sptr, int c1, size_t n);

689 /* memcmp.c */
690 int memcmp(const void *s1, const void *s2, size_t n);

692 /* memcpy.c */
693 void *memcpy(void *_RESTRICT_KYWD s, const void *_RESTRICT_KYWD s0, size_t n);

695 /* memmove.c */
696 void *memmove(void *s, const void *s0, size_t n);

698 /* memset.c */
699 void *memset(void *sp1, int c, size_t n);

701 /* mkdev.c */
702 dev_t __makedev(const int version, const major_t majdev,
703                 const minor_t mindev);
704 major_t __major(const int version, const dev_t devnum);
705 minor_t __minor(const int version, const dev_t devnum);

707 /* mkfifo.c */
708 int mkfifo(const char *path, mode_t mode);

710 /* mktemp.c */
711 char *mktemp(char *as);

713 /* mlock.c */
714 int mlock(caddr_t addr, size_t len);

716 /* mlockall.c */
717 int mlockall(int flags);

719 /* mon.c */
720 void monitor(int (*alowpc)(), int (*ahighpc)(), WORD *buffer,
721                 size_t bufsize, size_t nfunc);
```



```
723 /* msync.c */
724 int msync(caddr_t addr, size_t len, int flags);

726 /* munlock.c */
727 int munlock(caddr_t addr, size_t len);

729 /* munlockall.c */
730 int munlockall(void);

732 /* ndbm.c */
733 void dbm_setdefwrite(DBM *db);
734 int dbm_flush(DBM *db);
735 int dbm_flushpag(DBM *db);
736 DBM *dbm_open(const char *file, int flags, mode_t mode);
737 void dbm_close(DBM *db);
738 int dbm_close_status(DBM *db);
739 datum dbm_fetch(DBM *db, datum key);
740 int dbm_delete(DBM *db, datum key);
741 int dbm_store(DBM *db, datum key, datum dat, int replace);
742 datum dbm_firstkey(DBM *db);
743 datum dbm_nextkey(DBM *db);
744 datum dbm_do_nextkey(DBM *db, datum inkey);

746 /* new_list.c */

748 /* nftw.c */
749 int nftw(const char *path, int (*fn)(), int depth, int flags);

751 /* nl_langinfo.c */
752 char *nl_langinfo(nl_item item);

754 /* opendir.c */
755 DIR *opendir(const char *filename);

757 /* opt_data.c */

759 /* perror.c */
760 void perror(const char *s);

762 /* pipe.c */
763 int pipe(int *fds);

765 /* psiginfo.c */
766 void psiginfo(siginfo_t *sip, char *s);

768 /* psignal.c */
769 void psignal(int sig, const char *s);

771 /* pt.c */
772 char *ptsname(int fd);
773 int unlockpt(int fd);
774 int grantpt(int fd);

776 /* putenv.c */
777 int putenv(char *change);
778 int setenv(const char *envname, const char *envval, int overwrite);
779 int unsetenv(const char *name);

781 /* putpwent.c */
782 int putpwent(const struct passwd *p, FILE *f);

784 /* qsort.c */
785 void qsort(void *base, size_t n, size_t size, int (*compar)());

787 /* raise.c */
```

```
788 int raise(int sig);

790 /* rand.c */
791 void srand(unsigned x);
792 int rand(void);
793 int rand_r(unsigned int *);

795 /* random.c */
796 void srandom(unsigned x);
797 char *initstate(unsigned seed, char *arg_state, size_t n);
798 char *setstate(const char *arg_state);
799 long random(void);

801 /* rctlops.c */
802 int rctl_walk(int (*callback)(const char *, void *), void *walk_data);
803 hrtime_t rctlblk_get_firing_time(rctlblk_t *rblk);
804 uint_t rctlblk_get_global_action(rctlblk_t *rblk);
805 uint_t rctlblk_get_global_flags(rctlblk_t *rblk);
806 uint_t rctlblk_get_local_action(rctlblk_t *rblk, int *signalp);
807 uint_t rctlblk_get_local_flags(rctlblk_t *rblk);
808 id_t rctlblk_get_recipient_pid(rctlblk_t *rblk);
809 rctl_priv_t rctlblk_get_privilege(rctlblk_t *rblk);
810 rctl_qty_t rctlblk_get_value(rctlblk_t *rblk);
811 void rctlblk_set_local_action(rctlblk_t *rblk, uint_t action, int signal);
812 void rctlblk_set_local_flags(rctlblk_t *rblk, uint_t flags);
813 void rctlblk_set_privilege(rctlblk_t *rblk, rctl_priv_t priv);
814 void rctlblk_set_value(rctlblk_t *rblk, rctl_qty_t val);
815 size_t rctlblk_size(void);

817 /* readdir.c */
818 struct dirent *readdir(DIR *dirp);

820 /* realpath.c */
821 char *realpath(const char *_RESTRICT_KYWD raw, char *_RESTRICT_KYWD canon);

823 /* regexpr.c */
824 char *re_comp(const char *sp);
825 int re_exec(const char *p1);

827 /* rindex.c */
828 char *rindex(const char *sp, int c);

830 /* rename.c */
831 int remove(const char *filename);
832 int rename(const char *old, const char *new);

834 /* rewinddir.c */
835 #undef rewinddir
836 void rewinddir(DIR *dirp);

838 /* scandir.c */
839 int alphasort(const struct dirent **, const struct dirent **);
840 int scandir(const char *dirname, struct dirent *(*namelist[]),
841         int (*select)(const struct dirent *),
842         int (*dcomp)(const struct dirent **, const struct dirent **));

844 /* scrwidth.c */
845 int scrwidth(wchar_t c);

847 /* seekdir.c */
848 void seekdir(DIR *dirp, long loc);

850 /* select.c */
851 int pselect(int nfds,
852         fd_set *_RESTRICT_KYWD readfds,
853         fd_set *_RESTRICT_KYWD writefds,
```



```
854         fd_set *_RESTRICT_KYWD errorfds,
855         const struct timespec *_RESTRICT_KYWD timeout,
856         const sigset_t *_RESTRICT_KYWD sigmask);
857 int select(int nfds,
858         fd_set *_RESTRICT_KYWD readfds,
859         fd_set *_RESTRICT_KYWD writefds,
860         fd_set *_RESTRICT_KYWD errorfds,
861         struct timeval *_RESTRICT_KYWD timeout);

863 /* setlocale.c */
864 char *setlocale(int cat, const char *loc);

866 /* setpriority.c */
867 int getpriority(int which, id_t who);
868 int setpriority(int which, id_t who, int prio);

870 /* settimeofday.c */
871 int settimeofday(struct timeval *tp, void *);

873 /* sigflag.c */
874 int sigflag(int sig, int flag, int on);

876 /* siglist.c */

878 /* sigsend.c */
879 int sigsend(idtype_t idtype, id_t id, int sig);

881 /* sigsetops.c */
882 int sigfillset(sigset_t *set);
883 int sigemptyset(sigset_t *set);
884 int sigaddset(sigset_t *set, int sig);
885 int sigdelset(sigset_t *set, int sig);
886 int sigismember(const sigset_t *set, int sig);

888 /* scalls.c */
889 unsigned sleep(unsigned sleep_tm);

891 /* ssignal.c */
892 int (*ssignal(int sig, int (*fn)())) ();
893 int gsignal(int sig);

895 /* str2id.c */

897 /* str2sig.c */
898 int str2sig(const char *s, int *sigp);
899 int sig2str(int i, char *s);

901 /* strcat.c */
902 char *strcat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

904 /* strchr.c */
905 char *strchr(const char *sp, int c);

907 /* strcmp.c */
908 int strcmp(const char *s1, const char *s2);

910 /* strcpy.c */
911 char *strcpy(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

913 /* strcspn.c */
914 size_t strcspn(const char *string, const char *charset);

916 /* strdup.c */
917 char *strdup(const char *s1);

919 /* strerror.c */
```

```
 920 char *strerror(int errnum);
 921 int strerror_r(int errnum, char *strerrbuf, size_t buflen);

 923 /* strftime.c */
 924 size_t strftime(char *_RESTRICT_KYWD s, size_t maxsize,
 925                 const char *_RESTRICT_KYWD format,
 926                 const struct tm *_RESTRICT_KYWD tm);

 928 /* strlen.c */
 929 size_t strlen(const char *s);

 931 /* strncat.c */
 932 char *strncat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2, size_t n);

 934 /* strncmp.c */
 935 int strncmp(const char *s1, const char *s2, size_t n);

 937 /* strncpy.c */
 938 char *strncpy(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2, size_t n);

 940 /* strpbrk.c */
 941 char *strpbrk(const char *string, const char *brkset);

 943 /* strrchr.c */
 944 char *strrchr(const char *sp, int c);

 946 /* strsep.c */
 947 char *strsep(char **stringp, const char *delim);

 949 /* strspn.c */
 950 size_t strspn(const char *string, const char *charset);

 952 /* strstr.c */
 953 char *strstr(const char *as1, const char *as2);

 955 /* strtod.c */
 956 double strtod(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
 957 float strtof(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
 958 long double strtold(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);

 960 /* strtoimax.c */
 961 intmax_t strtoimax(const char *_RESTRICT_KYWD nptr,
 962         char **_RESTRICT_KYWD endptr, int base);

 964 /* strtok.c */
 965 char *strtok(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD sepset);
 966 char *strtok_r(char *_RESTRICT_KYWD, const char *_RESTRICT_KYWD,
 967         char **_RESTRICT_KYWD);

 969 /* strtol.c */
 970 long strtol(const char *_RESTRICT_KYWD str, char **_RESTRICT_KYWD nptr,
 971         int base);

 973 /* strtoul.c */
 974 unsigned long strtoul(const char *_RESTRICT_KYWD str,
 975         char **_RESTRICT_KYWD nptr, int base);

 977 /* strtoumax.c */
 978 uintmax_t strtoumax(const char *_RESTRICT_KYWD nptr,
 979                     char **_RESTRICT_KYWD endptr, int base);

 981 /* strxfrm.c */
 982 size_t strxfrm(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2,
 983         size_t n);
 984 int strcoll(const char *s1, const char *s2);
```



```
 986 /* swab.c */
 987 void swab(const char *_RESTRICT_KYWD from, char *_RESTRICT_KYWD to, ssize_t n);

 989 /* swapctl.c */
 990 int swapctl(int cmd, void *arg);

 992 /* sysconf.c */
 993 long sysconf(int name);

 995 /* syslog.c */
 996 /* VARARGS2 */
 997 void syslog(int pri, const char *fmt, ...);
 998 void vsyslog(int pri, const char *fmt, va_list ap);
 999 void openlog(const char *ident, int logstat, int logfac);
1000 void closelog(void);
1001 int setlogmask(int pmask);

1003 /* tcdrain.c */
1004 int tcdrain(int fildes);

1006 /* tcflow.c */
1007 int tcflow(int fildes, int action);

1009 /* tcflush.c */
1010 int tcflush(int fildes, int queue_selector);

1012 /* tcgetattr.c */
1013 int tcgetattr(int fildes, struct termios *termios_p);

1015 /* tcgetpgrp.c */
1016 pid_t tcgetpgrp(int fd);

1018 /* tcgetsid.c */
1019 pid_t tcgetsid(int fd);

1021 /* tcsendbreak.c */
1022 int tcsendbreak(int fildes, int duration);

1024 /* tcsetattr.c */
1025 int tcsetattr(int fildes, int optional_actions,
1026         const struct termios *termios_p);

1028 /* tcsetpgrp.c */
1029 int tcsetpgrp(int fd, pid_t pgrp);

1031 /* tell.c */
1032 long tell(int f);

1034 /* telldir.c */
1035 long telldir(DIR *dirp);

1037 /* tfind.c */
1038 void *tfind(const void *ky, void *const *rtp, int (*compar)());

1040 /* time_comm.c */
1041 struct tm *localtime(const time_t *timep);
1042 struct tm *localtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
1043 struct tm *gmtime(const time_t *clock);
1044 struct tm *gmtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
1045 double difftime(time_t time1, time_t time0);
1046 time_t mktime(struct tm *timeptr);
1047 void _ltzset(time_t tim);
1048 void tzset(void);

1050 /* time_data.c */
```

```
1052 /* time_gdata.c */

1054 /* tolower.c */
1055 int tolower(int c);

1057 /* toupper.c */
1058 int toupper(int c);

1060 /* truncate.c */
1061 int ftruncate(int fildes, off_t len);
1062 int truncate(const char *path, off_t len);

1064 /* tsearch.c */
1065 void *tsearch(const void *ky, void **rtp, int (*compar)());
1066 void *tdelete(const void *ky, void **rtp, int (*compar)());
1067 void twalk(const void *rt, void (*action)());

1069 /* ttyname.c */
1070 char *ttyname(int f);
1071 char *_ttyname_dev(dev_t rdev, char *buffer, size_t buflen);
1072 char *ttyname_r(int, char *, int);

1074 /* ttyslot.c */
1075 int ttyslot(void);

1077 /* ualarm.c */
1078 unsigned ualarm(unsigned usecs, unsigned reload);

1080 /* ulimit.c */
1081 /* VARARGS1 */
1082 long ulimit(int cmd, ...);

1084 /* scalls.c */
1085 int usleep(unsigned n);

1087 /* valloc.c */
1088 void *valloc(size_t size);

1090 /* waitpid.c */
1091 pid_t wait(int *stat_loc);
1092 pid_t waitpid(pid_t pid, int *stat_loc, int options);
1093 pid_t wait3(int *status, int options, struct rusage *rp);
1094 pid_t wait4(pid_t pid, int *status, int options, struct rusage *rusage);

1096 /* wcstombs.c */
1097 size_t wcstombs(char *_RESTRICT_KYWD s, const wchar_t *_RESTRICT_KYWD pwcs,
1098         size_t n);

1100 /* wctomb.c */
1101 int wctomb(char *s, wchar_t wchar);

1103 /* wdata.c */

1105 /* wisprint.c */
1106 int wisprint(wchar_t c);

1108 /* xgetwidth.c */
1109 void _xgetwidth(void);

1111 /*
1112  * /usr/src/lib/libc/port/intl routines
1113  */

1115 /* gettext.c */
1116 char *bindtextdomain(const char *domain, const char *binding);
1117 char *dcgettext(const char *domain, const char *msg_id, const int category);
```



```
1118 char *dgettext(const char *domain, const char *msg_id);
1119 char *gettext(const char *msg_id);
1120 char *textdomain(const char *domain);

1122 /*
1123  * /usr/src/lib/libc/port/print routines
1124  */

1126 /* fprintf.c */
1127 /* VARARGS2 */
1128 int fprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format, ...);

1130 /* printf.c */
1131 /* VARARGS1 */
1132 int printf(const char *_RESTRICT_KYWD format, ...);

1134 /* snprintf.c */
1135 /* VARARGS2 */
1136 int snprintf(char *_RESTRICT_KYWD string, size_t n,
1137                 const char *_RESTRICT_KYWD format, ...);

1139 /* sprintf.c */
1140 /* VARARGS2 */
1141 int sprintf(char *_RESTRICT_KYWD string,
1142                 const char *_RESTRICT_KYWD format, ...);

1144 /* vfprintf.c */
1145 /* VARARGS2 */
1146 int vfprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format,
1147                 va_list);

1149 /* vprintf.c */
1150 /* VARARGS1 */
1151 int vprintf(const char *_RESTRICT_KYWD format, va_list);

1153 /* vsnprintf.c */
1154 /* VARARGS2 */
1155 int vsnprintf(char *_RESTRICT_KYWD string, size_t n,
1156                 const char *_RESTRICT_KYWD format, va_list);

1158 /* vsprintf.c */
1159 /* VARARGS2 */
1160 int vsprintf(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD format,
1161         va_list);

1163 /*
1164  * /usr/src/lib/libc/port/regex routines
1165  */

1167 /* glob.c */
1168 extern int glob(const char *restrict pattern, int flags,
1169     int(*errfunc)(const char *epath, int eerrno), glob_t *restrict pglob);
1170 extern void globfree(glob_t *pglob);

1172 /* regex.c */
1173 char *regex(const char *regexp, const char *stringp, ...);
1174 #ifdef __loc1
1175 #undef __loc1
1176 #endif
1177 char *__loc1;

1179 /* regcmp.c */
1180 char *regcmp(const char *regexp, ...);
1181 #ifdef __i_size
1182 #undef __i_size
1183 #endif
```

```
1184 int __i_size;

1186 /*
1187  * /usr/src/lib/libc/port/stdio routines
1188  */

1190 /* _filbuf.c */
1191 int _filbuf(FILE *iop);

1193 /* _flsbuf.c */
1194 int _flsbuf(int ch, FILE *iop);

1196 /* _wrtchk.c */
1197 int _wrtchk(FILE *iop);

1199 /* clearerr.c */
1200 void clearerr(FILE *iop);

1202 /* ctermid.c */
1203 char *ctermid(char *s);
1204 char *ctermid_r(char *s);

1206 /* cuserid.c */
1207 char *cuserid(char *s);

1209 /* data.c */

1211 /* doscan.c */
1212 int _doscan(FILE *iop, const char *fmt, va_list va_alist);

1214 /* fdopen.c */
1215 FILE *fdopen(int fd, const char *type);

1217 /* feof.c */
1218 int feof(FILE *iop);

1220 /* ferror.c */
1221 int ferror(FILE *iop);

1223 /* fgetc.c */
1224 int fgetc(FILE *iop);

1226 /* fgets.c */
1227 char *fgets(char *_RESTRICT_KYWD buf, int size, FILE *_RESTRICT_KYWD iop);

1229 /* fileno.c */
1230 int _fileno(FILE *iop);

1232 /* flush.c */
1233 void _cleanup(void);
1234 FILE *_findiop(void);
1235 typedef unsigned char Uchar;
1236 void _setbufend(FILE *iop, Uchar *end);
1237 Uchar *_realbufend(FILE *iop);
1238 void _bufsync(FILE *iop, Uchar *bufend);
1239 int _xflsbuf(FILE *iop);
1240 int fflush(FILE *iop);
1241 int fclose(FILE *iop);

1243 /* fopen.c */
1244 FILE *fopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type);
1245 FILE *freopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type,
1246                 FILE *_RESTRICT_KYWD iop);

1248 /* fpos.c */
1249 int fgetpos(FILE *_RESTRICT_KYWD stream, fpos_t *_RESTRICT_KYWD pos);
```



```
1250 int fsetpos(FILE *stream, const fpos_t *pos);

1252 /* fputc.c */
1253 int fputc(int ch, FILE *iop);

1255 /* fputs.c */
1256 int fputs(const char *_RESTRICT_KYWD ptr, FILE *_RESTRICT_KYWD iop);

1258 /* fread.c */
1259 size_t fread(void *_RESTRICT_KYWD ptr, size_t size, size_t count,
1260         FILE *_RESTRICT_KYWD iop);

1262 /* fseek.c */
1263 int fseek(FILE *iop, long offset, int ptrname);

1265 /* ftell.c */
1266 long ftell(FILE *iop);

1268 /* fwrite.c */
1269 size_t fwrite(const void *_RESTRICT_KYWD ptr1, size_t size, size_t count,
1270         FILE *_RESTRICT_KYWD iop);

1272 /* getc.c */
1273 int getc(FILE *iop);

1275 /* getchar.c */
1276 int getchar(void);

1278 /* getpass.c */
1279 char *getpass(const char *prompt);

1281 /* getpass.c */
1282 char *getpassphrase(const char *prompt);

1284 /* gets.c */
1285 char *gets(char *buf);

1287 /* getw.c */
1288 int getw(FILE *stream);

1290 /* popen.c */
1291 FILE *popen(const char *cmd, const char *mode);
1292 int pclose(FILE *ptr);

1294 /* putc.c */
1295 int putc(int ch, FILE *iop);

1297 /* putchar.c */
1298 int putchar(int ch);

1300 /* puts.c */
1301 int puts(const char *ptr);

1303 /* putw.c */
1304 int putw(int w, FILE *stream);

1306 /* rewind.c */
1307 void rewind(FILE *iop);

1309 /* scanf.c */
1310 /* VARARGS1 */
1311 int scanf(const char *_RESTRICT_KYWD fmt, ...);

1313 /* VARARGS2 */
1314 int fscanf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD fmt, ...);
```

```
1316 /* VARARGS2 */
1317 int sscanf(const char *_RESTRICT_KYWD str, const char *_RESTRICT_KYWD fmt, ...);

1319 /* setbuf.c */
1320 void setbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf);

1322 /* setvbuf.c */
1323 int setvbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf, int type,
1324                 size_t size);

1326 /* system.c */
1327 int system(const char *s);

1329 /* tempnam.c */
1330 char *tempnam(const char *dir, const char *pfx);

1332 /* tmpfile.c */
1333 FILE *tmpfile(void);

1335 /* tmpnam.c */
1336 char *tmpnam(char *s);
1337 char *tmpnam_r(char *);

1339 /* ungetc.c */
1340 int ungetc(int c, FILE *iop);

1342 /*
1343  * /usr/src/lib/libc/port/sys routines
1344  */

1346 /* exacctsys.c */
1347 size_t getacct(idtype_t idtype, id_t id, void *buf, size_t bufsize);
1348 int putacct(idtype_t idtype, id_t id, void *buf, size_t bufsize, int flags);
1349 int wracct(idtype_t idtype, id_t id, int flags);

1351 /* execl.c */
1352 /* VARARGS1 */
1353 int execl(const char *name, const char *, ...);

1355 /* execle.c */
1356 int execle(const char *, const char *file, ...);

1358 /* execv.c */
1359 int execv(const char *file, char *const *argv);

1361 /* lockf.c */
1362 int lockf(int fildes, int function, off_t size);

1364 /* meminfosys.c */
1365 int meminfo(const uint64_t *inaddr, int addr_count, const uint_t *info_req,
1366         int info_count, uint64_t *outdata, uint_t *validity);

1368 /* msgsys.c */
1369 int msgget(key_t key, int msgflg);
1370 int msgctl(int msqid, int cmd, struct msqid_ds *buf);
1371 ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);
1372 int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);

1374 /* nfssys.c */
1375 /*
1376 int exportfs(char *dir, struct export *ep);
1377 int nfs_getfh(char *path, fhandle_t *fhp);
1378 int nfssvc(int fd);
1379 */

1381 /* psetsys.c */
```



```
1382 int pset_create(psetid_t *npset);
1383 int pset_destroy(psetid_t pset);
1384 int pset_assign(psetid_t pset, processorid_t cpu, psetid_t *opset);
1385 int pset_assign_forced(psetid_t pset, processorid_t cpu, psetid_t *opset);
1386 int pset_info(psetid_t pset, int *type, u_int *numcpus, processorid_t *cpulist);
1387 int pset_bind(psetid_t pset, idtype_t idtype, id_t id, psetid_t *opset);
1388 int pset_bind_lwp(psetid_t pset, id_t id, pid_t, psetid_t *opset);
1389

1391 /* rctlsys.c */
1392 int getrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
1393     int flags);
1394 int setrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
1395     int flags);
1396 /* (private functions) */
1397 int setprojrctl(const char *name, rctlblk_t *new_rblk, size_t size, int flags);
1398 int rctlctl(const char *, rctlblk_t *, int);
1399 size_t rctllist(char *, size_t);


1402 /* semsys.c */
1403 int semctl(int semid, int semnum, int cmd, ...);
1404 int semget(key_t key, int nsems, int semflg);
1405 int semop(int semid, struct sembuf *sops, size_t nsops);

1407 /* shmsys.c */
1408 void *shmat(int shmid, const void *shmaddr, int shmflg);
1409 int shmctl(int shmid, int cmd, struct shmid_ds *buf);
1410 #if defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4)
1411 int shmdt(const void *);
1412 #else
1413 int shmdt(char *);
1414 #endif /* defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4) */
1415 int shmget(key_t key, size_t size, int shmflg);

1417 /* tasksys.c */
1418 taskid_t settaskid(projid_t project, uint_t flags);
1419 taskid_t gettaskid(void);
1420 projid_t getprojid(void);

1422 /*
1423  * /usr/src/lib/libc/port/widec routines
1424  */

1426 /* fgetws.c */
1427 wchar_t *fgetws(wchar_t *_RESTRICT_KYWD ptr, int  size,
1428         FILE *_RESTRICT_KYWD iop);

1430 /* fputwc.c */
1431 wint_t fputwc(wint_t wc, FILE *iop);
1432 wint_t putwc(wint_t wc, FILE *iop);

1434 /* fputws.c */
1435 int fputws(const wchar_t *_RESTRICT_KYWD ptr, FILE *_RESTRICT_KYWD iop);

1437 /* getwchar.c */
1438 wint_t getwchar(void);

1440 /* getwidth.c */
1441 void getwidth(eucwidth_t *eucstruct);

1443 /* getws.c */
1444 wchar_t *getws(wchar_t *ptr);

1446 /* iswctype.c */
1447 int iswctype(wint_t wc, wctype_t charclass);
```

```
1448 int iswalpha(wint_t c);
1449 int iswupper(wint_t c);
1450 int iswlower(wint_t c);
1451 int iswdigit(wint_t c);
1452 int iswxdigit(wint_t c);
1453 int iswalnum(wint_t c);
1454 int iswspace(wint_t c);
1455 int iswpunct(wint_t c);
1456 int iswprint(wint_t c);
1457 int iswgraph(wint_t c);
1458 int iswcntrl(wint_t c);
1459 int isphonogram(wint_t c);
1460 int isideogram(wint_t c);
1461 int isenglish(wint_t c);
1462 int isnumber(wint_t c);
1463 int isspecial(wint_t c);

1465 /* libwcollate.c */

1467 /* putwchar.c */
1468 wint_t putwchar(wint_t c);

1470 /* putws.c */
1471 int putws(const wchar_t *ptr);

1473 /* scrwidth.c */

1475 /* strtows.c */
1476 wchar_t *strtows(wchar_t *s1, char *s2);
1477 char *wstostr(char *s1, wchar_t *s2);

1479 /* trwctype.c */
1480 wint_t towupper(wint_t c);
1481 wint_t towlower(wint_t c);

1483 /* ungetwc.c */
1484 wint_t ungetwc(wint_t wc, FILE *iop);

1486 /* wcollate.c */
1487 size_t wcsxfrm(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1488         size_t n);
1489 int wcscoll(const wchar_t *s1, const wchar_t *s2);

1491 /* wcsftime.c */
1492 #if !defined(__amd64)           /* XX64 - fix me */
1493 size_t wcsftime(wchar_t *wcs, size_t maxsize,
1494         const char *format, const struct tm *timeptr);
1495 #endif  /* __amd64 */

1497 /* wcstring.c */
1498 wint_t fgetwc(FILE *iop);
1499 wint_t getwc(FILE *iop);
1500 int wcwidth(wchar_t wc);
1501 int wcswidth(const wchar_t *pwcs, size_t n);

1503 /* wcswcs.c */
1504 wchar_t *wcswcs(const wchar_t *ws1, const wchar_t *ws2);

1506 /* wcsxfrm.c - empty file! */

1508 /* wcsxfrm.xpg4.c */

1510 /* wisprint.c */
1511 int wisprint(wchar_t c);

1513 /* wscasecmp.c */
```



```
1514 int wscasecmp(const wchar_t *s1, const wchar_t *s2);

1516 /* wscat.c */
1517 wchar_t *wcscat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
1518 wchar_t *wscat(wchar_t *s1, const wchar_t *s2);

1520 /* wschr.c */
1521 wchar_t *wcschr(const wchar_t *sp, wchar_t c);
1522 wchar_t *wschr(const wchar_t *sp, wchar_t c);

1524 /* wscmp.c */
1525 int wcscmp(const wchar_t *s1, const wchar_t *s2);
1526 int wscmp(const wchar_t *s1, const wchar_t *s2);

1528 /* wscol.c */
1529 int wscol(const wchar_t *s1);

1531 /* wscpy.c */
1532 wchar_t *wcscpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
1533 wchar_t *wscpy(wchar_t *s1, const wchar_t *s2);

1535 /* wscspn.c */
1536 size_t wcscspn(const wchar_t *string, const wchar_t *charset);
1537 size_t wscspn(const wchar_t *string, const wchar_t *charset);

1539 /* wsdup.c */
1540 wchar_t *wsdup(const wchar_t *s1);

1542 /* wslen.c */
1543 size_t wcslen(const wchar_t *s);
1544 size_t wslen(const wchar_t *s);

1546 /* wsncasecmp.c */
1547 int wsncasecmp(const wchar_t *s1, const wchar_t *s2, size_t n);

1549 /* wsncat.c */
1550 wchar_t *wcsncat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1551         size_t n);
1552 wchar_t *wsncat(wchar_t *s1, const wchar_t *s2, size_t n);

1554 /* wsncmp.c */
1555 int wcsncmp(const wchar_t *s1, const wchar_t *s2, size_t n);
1556 int wsncmp(const wchar_t *s1, const wchar_t *s2, size_t n);

1558 /* wsncpy.c */
1559 wchar_t *wcsncpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1560         size_t n);
1561 wchar_t *wsncpy(wchar_t *s1, const wchar_t *s2, size_t n);

1563 /* wspbrk.c */
1564 wchar_t *wcspbrk(const wchar_t *string, const wchar_t *brkset);
1565 wchar_t *wspbrk(const wchar_t *string, const wchar_t *brkset);

1567 /* wsprintf.c */
1568 int wsprintf(wchar_t *wstring, const char *format, ...);

1570 /* wsrchr.c */
1571 wchar_t *wcsrchr(const wchar_t *sp, wchar_t c);
1572 wchar_t *wsrchr(const wchar_t *sp, wchar_t c);

1574 /* wsscanf.c */
1575 int wsscanf(wchar_t *s, const char *format, ...);

1577 /* wssize.c */

1579 /* wsspn.c */
```

```
1580 size_t wcsspn(const wchar_t *string, const wchar_t *charset);
1581 size_t wsspn(const wchar_t *string, const wchar_t *charset);

1583 /* wstod.c */
1584 double wcstod(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
1585 float wcstof(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
1586 long double wcstold(const wchar_t *_RESTRICT_KYWD cp,
1587         wchar_t **_RESTRICT_KYWD ptr);
1588 double wstod(const wchar_t *cp, wchar_t **ptr);

1590 /* wstok.c */
1591 #if !defined(__amd64)           /* XX64 - fix me */
1592 wchar_t *wcstok(wchar_t *string, const wchar_t *sepset);
1593 wchar_t *wstok(wchar_t *string, const wchar_t *sepset);
1594 #endif  /* __amd64 */

1596 /* wcstol.c */
1597 long wcstol(const wchar_t *_RESTRICT_KYWD str, wchar_t **_RESTRICT_KYWD ptr,
1598         int base);
1599 long long wcstoll(const wchar_t *_RESTRICT_KYWD str,
1600         wchar_t **_RESTRICT_KYWD ptr, int base);

1602 /* wcstoul.c */
1603 unsigned long wcstoul(const wchar_t *_RESTRICT_KYWD str,
1604         wchar_t **_RESTRICT_KYWD ptr, int base);
1605 unsigned long long wcstoull(const wchar_t *_RESTRICT_KYWD str,
1606         wchar_t **_RESTRICT_KYWD ptr, int base);

1608 /* wcstoimax.c */
1609 intmax_t wcstoimax(const wchar_t *_RESTRICT_KYWD nptr,
1610         wchar_t **_RESTRICT_KYWD endptr, int base);
1611 uintmax_t wcstoumax(const wchar_t *_RESTRICT_KYWD nptr,
1612         wchar_t **_RESTRICT_KYWD endptr, int base);

1614 /* wstol.c */
1615 long wstol(const wchar_t *str, wchar_t **ptr, int base);

1617 /* wstoll.c */
1618 long long wstoll(const wchar_t *str, wchar_t **ptr, int base);
1619 long long watoll(const wchar_t *p);

1621 /* wsxfrm.c */
1622 size_t wsxfrm(wchar_t *s1, const wchar_t *s2, size_t n);
1623 int wscoll(const wchar_t *s1, const wchar_t *s2);

1625 /*
1626  * /usr/src/lib/libc/port/gen/event_port.c
1627  */
1628 int port_dispatch(int port, int flags, int source, int events, uintptr_t object,
1629     void *user);

1631 /*
1632  * /usr/src/lib/libc/$MACH/gen routines
1633  */

1635 /* alloca.s */

1637 void *__builtin_alloca(size_t);

1639 /*
1640  * modctl(int arg, ...) and utssys(...) are not available from a header
1641  * file, but our utilities which make use of it should be able to be
1642  * lint clean.
1643  */
1644 int modctl(int arg, ...);
1645 int utssys(void *buf, int arg, int type, void *outbp);
```



```
1648 typedef float single;
1649 typedef unsigned extended[3];
1650 typedef long double quadruple;
1651 typedef unsigned fp_exception_field_type;

1653 typedef char decimal_string[512];

1655 enum fp_class_type {
1656         fp_zero  = 0,
1657         fp_subnormal    = 1,
1658         fp_normal       = 2,
1659         fp_infinity     = 3,
1660         fp_quiet        = 4,
1661         fp_signaling    = 5
1662 };
_____unchanged_portion_omitted_
```

```
**********************************************************
   56699 Mon Apr  6 11:47:18 2015
new/usr/src/lib/libc/port/mapfile-vers
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
#
# 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) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright (c) 2013 Gary Mills
# Copyright 2015 Garrett D'Amore <garrett@damore.org>
# Copyright 2014 Garrett D'Amore <garrett@damore.org>

#
# MAPFILE HEADER START
#
# WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
# Object versioning must comply with the rules detailed in
#
#       usr/src/lib/README.mapfiles
#
# You should not be making modifications here until you've read the most current
# copy of that file. If you need help, contact a gatekeeper for guidance.
#
# MAPFILE HEADER END
#

$mapfile_version 2

#
# All function names added to this or any other libc mapfile
# must be placed under the 'protected:' designation.
# The 'global:' designation is used *only* for data
# items and for the members of the malloc() family.
#

#
# 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_0.12 {
    protected:
        fexecve;
} ILLUMOS_0.11;

SYMBOL_VERSION ILLUMOS_0.11 {   # Illumos additions
    protected:
        iswxdigit_l;
        isxdigit_l;
} ILLUMOS_0.10;
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    1340 Mon Apr  6 11:47:20 2015
new/usr/src/lib/libc/port/sys/fexecve.c
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

/*
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 */

#include "lint.h"
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <assert.h>

#define FDFILE  "/dev/fd/%u"

/*
 * fexecve.c - implements the fexecve function.
 *
 * We implement in terms of the execve() system call, but use the path
 * /dev/fd/<fd>.  This depends on special handling in the execve system
 * call; note that /dev/fd files are not normally directly executable.
 * This does not actually use the fd filesystem mounted on /dev/fd.
 */
int
fexecve(int fd, char *const argv[], char *const envp[])
{
        char path[32];  /* 8 for "/dev/fd/", 10 for %u, + \0 == 19 */

        if (fcntl(fd, F_GETFL) < 0) {
                /* This will have the effect of returning EBADF */
                return (-1);
        }
        assert(snprintf(NULL, 0, FDFILE, fd) < (sizeof (path) - 1));
        (void) snprintf(path, sizeof (path), FDFILE, fd);
        return (execve(path, argv, envp));
}
```

```
**********************************************************
   25899 Mon Apr  6 11:47:22 2015
new/usr/src/lib/libc/sparc/Makefile.com
5798 fexecve() needed per POSIX 2008
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBCDIR=        $(SRC)/lib/libc
LIB_PIC=        libc_pic.a
VERS=           .1
CPP=            /usr/lib/cpp
TARGET_ARCH=    sparc

# objects are grouped by source directory

# Symbol capabilities objects.
EXTPICS=                        \
        $(LIBCDIR)/capabilities/sun4u/sparc/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-opl/sparc/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-us3-hwcap1/sparc/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-us3-hwcap2/sparc/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4v-hwcap1/sparc/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4v-hwcap2/sparc/pics/symcap.o

# local objects
STRETS=                         \
        stret1.o                \
        stret2.o                \
        stret4.o

CRTOBJS=                        \
        _ftou.o                 \
        cerror.o                \
        cerror64.o              \
        hwmuldiv.o

DYNOBJS=                        \
```



```
        _rtbootld.o

FPOBJS=                         \
        _D_cplx_div.o           \
        _D_cplx_div_ix.o        \
        _D_cplx_div_rx.o        \
        _D_cplx_mul.o           \
        _F_cplx_div.o           \
        _F_cplx_div_ix.o        \
        _F_cplx_div_rx.o        \
        _F_cplx_mul.o           \
        _Q_add.o                \
        _Q_cmp.o                \
        _Q_cmpe.o               \
        _Q_cplx_div.o           \
        _Q_cplx_div_ix.o        \
        _Q_cplx_div_rx.o        \
        _Q_cplx_lr_div.o        \
        _Q_cplx_lr_div_ix.o     \
        _Q_cplx_lr_div_rx.o     \
        _Q_cplx_lr_mul.o        \
        _Q_cplx_mul.o           \
        _Q_div.o                \
        _Q_dtoq.o               \
        _Q_fcc.o                \
        _Q_itoq.o               \
        _Q_lltoq.o              \
        _Q_mul.o                \
        _Q_neg.o                \
        _Q_qtod.o               \
        _Q_qtoi.o               \
        _Q_qtos.o               \
        _Q_qtou.o               \
        _Q_scl.o                \
        _Q_set_except.o         \
        _Q_sqrt.o               \
        _Q_stoq.o               \
        _Q_sub.o                \
        _Q_ulltoq.o             \
        _Q_utoq.o               \
        __quad_mag.o

FPASMOBJS=                      \
        _Q_get_rp_rd.o          \
        fpgetmask.o             \
        fpgetrnd.o              \
        fpgetsticky.o           \
        fpsetmask.o             \
        fpsetrnd.o              \
        fpsetsticky.o

$(__GNUC)FPASMOBJS +=           \
        __quad.o

ATOMICOBJS=                     \
        atomic.o

XATTROBJS=                      \
        xattr_common.o

COMOBJS=                        \
        bcmp.o                  \
        bcopy.o                 \
        bzero.o                 \
        bsearch.o               \
        memccpy.o               \
```

```
127         qsort.o                 \
128         strtol.o                \
129         strtoul.o               \
130         strtoll.o               \
131         strtoull.o

133 DTRACEOBJS=                     \
134         dtrace_data.o

136 GENOBJS=                        \
137         _getsp.o                \
138         _xregs_clrptr.o         \
139         abs.o                   \
140         alloca.o                \
141         ascii_strcasecmp.o      \
142         byteorder.o             \
143         cuexit.o                \
144         ecvt.o                  \
145         errlst.o                \
146         getctxt.o               \
147         ladd.o                  \
148         lmul.o                  \
149         lock.o                  \
150         lshiftl.o               \
151         lsign.o                 \
152         lsub.o                  \
153         makectxt.o              \
154         memchr.o                \
155         memcmp.o                \
156         new_list.o              \
157         setjmp.o                \
158         siginfolst.o            \
159         siglongjmp.o            \
160         smt_pause.o             \
161         sparc_data.o            \
162         strchr.o                \
163         strcmp.o                \
164         strlcpy.o               \
165         strncmp.o               \
166         strncpy.o               \
167         strnlen.o               \
168         swapctxt.o              \
169         sync_instruction_memory.o

171 # sysobjs that contain large-file interfaces
172 COMSYSOBJS64=                   \
173         fstatvfs64.o            \
174         getdents64.o            \
175         getrlimit64.o           \
176         lseek64.o               \
177         mmap64.o                \
178         pread64.o               \
179         preadv64.o              \
180         pwrite64.o              \
181         pwritev64.o             \
182         setrlimit64.o           \
183         statvfs64.o

185 SYSOBJS64=

187 COMSYSOBJS=                     \
188         __clock_timer.o         \
189         __getloadavg.o          \
190         __rusagesys.o           \
191         __signotify.o           \
192         __sigrt.o               \
```



```
193         __time.o                \
194         _lgrp_home_fast.o       \
195         _lgrpsys.o              \
196         _nfssys.o               \
197         _portfs.o               \
198         _pset.o                 \
199         _rpcsys.o               \
200         _sigaction.o            \
201         _so_accept.o            \
202         _so_bind.o              \
203         _so_connect.o           \
204         _so_getpeername.o       \
205         _so_getsockname.o       \
206         _so_getsockopt.o        \
207         _so_listen.o            \
208         _so_recv.o              \
209         _so_recvfrom.o          \
210         _so_recvmsg.o           \
211         _so_send.o              \
212         _so_sendmsg.o           \
213         _so_sendto.o            \
214         _so_setsockopt.o        \
215         _so_shutdown.o          \
216         _so_socket.o            \
217         _so_socketpair.o        \
218         _sockconfig.o           \
219         acct.o                  \
220         acl.o                   \
221         adjtime.o               \
222         alarm.o                 \
223         brk.o                   \
224         chdir.o                 \
225         chroot.o                \
226         cladm.o                 \
227         close.o                 \
228         execve.o                \
229         exit.o                  \
230         facl.o                  \
231         fchdir.o                \
232         fchroot.o               \
233         fdsync.o                \
234         fpathconf.o             \
235         fstatfs.o               \
236         fstatvfs.o              \
237         getcpuid.o              \
238         getdents.o              \
239         getegid.o               \
240         geteuid.o               \
241         getgid.o                \
242         getgroups.o             \
243         gethrtime.o             \
244         getitimer.o             \
245         getmsg.o                \
246         getpid.o                \
247         getpmsg.o               \
248         getppid.o               \
249         getrlimit.o             \
250         getuid.o                \
251         gtty.o                  \
252         install_utrap.o         \
253         ioctl.o                 \
254         kaio.o                  \
255         kill.o                  \
256         llseek.o                \
257         lseek.o                 \
258         memcntl.o               \
```

```
259         mincore.o               \
260         mmap.o                  \
261         mmapobjsys.o            \
262         modctl.o                \
263         mount.o                 \
264         mprotect.o              \
265         munmap.o                \
266         nice.o                  \
267         ntp_adjtime.o           \
268         ntp_gettime.o           \
269         p_online.o              \
270         pathconf.o              \
271         pause.o                 \
272         pcsample.o              \
273         pipe2.o                 \
274         pollsys.o               \
275         pread.o                 \
276         preadv.o                \
277         priocntlset.o           \
278         processor_bind.o        \
279         processor_info.o        \
280         profil.o                \
281         putmsg.o                \
282         putpmsg.o               \
283         pwrite.o                \
284         pwritev.o               \
285         read.o                  \
286         readv.o                 \
287         resolvepath.o           \
288         seteguid.o              \
289         setgid.o                \
290         setgroups.o             \
291         setitimer.o             \
292         setreid.o               \
293         setrlimit.o             \
294         setuid.o                \
295         sigaltstk.o             \
296         sigprocmsk.o            \
297         sigsendset.o            \
298         sigsuspend.o            \
299         statfs.o                \
300         statvfs.o               \
301         stty.o                  \
302         sync.o                  \
303         sysconfig.o             \
304         sysfs.o                 \
305         sysinfo.o               \
306         syslwp.o                \
307         times.o                 \
308         ulimit.o                \
309         umask.o                 \
310         umount2.o               \
311         utssys.o                \
312         uucopy.o                \
313         vhangup.o               \
314         waitid.o                \
315         write.o                 \
316         writev.o                \
317         yield.o

319 SYSOBJS=                       \
320         __clock_gettime.o       \
321         __getcontext.o          \
322         _lwp_mutex_unlock.o     \
323         _stack_grow.o           \
324         __uadmin.o              \
```



```
325         door.o                  \
326         forkx.o                 \
327         forkallx.o              \
328         gettimeofday.o          \
329         ptrace.o                \
330         syscall.o               \
331         tls_get_addr.o          \
332         uadmin.o                \
333         umount.o                \
334         uname.o                 \
335         vforkx.o

337 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
338 PORTGEN64=                      \
339         _xftw64.o               \
340         attropen64.o            \
341         ftw64.o                 \
342         mkstemp64.o             \
343         nftw64.o                \
344         tell64.o                \
345         truncate64.o

347 # objects from source under $(LIBCDIR)/port
348 PORTFP=                         \
349         __flt_decim.o           \
350         __flt_rounds.o          \
351         __tbl_10_b.o            \
352         __tbl_10_h.o            \
353         __tbl_10_s.o            \
354         __tbl_2_b.o             \
355         __tbl_2_h.o             \
356         __tbl_2_s.o             \
357         __tbl_fdq.o             \
358         __tbl_tens.o            \
359         __x_power.o             \
360         _base_sup.o             \
361         aconvert.o              \
362         decimal_bin.o           \
363         double_decim.o          \
364         econvert.o              \
365         fconvert.o              \
366         file_decim.o            \
367         finite.o                \
368         fp_data.o               \
369         func_decim.o            \
370         gconvert.o              \
371         hex_bin.o               \
372         ieee_globals.o          \
373         pack_float.o            \
374         sigfpe.o                \
375         string_decim.o          \
376         ashldi3.o               \
377         ashrdi3.o               \
378         cmpdi2.o                \
379         divdi3.o                \
380         floatdidf.o             \
381         floatdisf.o             \
382         floatundidf.o           \
383         floatundisf.o           \
384         lshrdi3.o               \
385         moddi3.o                \
386         muldi3.o                \
387         qdivrem.o               \
388         ucmpdi2.o               \
389         udivdi3.o               \
390         umoddi3.o
```

```
392 PORTGEN=                        \
393         _env_data.o             \
394         _ftoll.o                \
395         _ftoull.o               \
396         _xftw.o                 \
397         a64l.o                  \
398         abort.o                 \
399         addsev.o                \
400         ascii_strncasecmp.o     \
401         assert.o                \
402         atof.o                  \
403         atoi.o                  \
404         atol.o                  \
405         atoll.o                 \
406         attrat.o                \
407         attropen.o              \
408         atexit.o                \
409         atfork.o                \
410         basename.o              \
411         calloc.o                \
412         catgets.o               \
413         catopen.o               \
414         cfgetispeed.o           \
415         cfgetospeed.o           \
416         cfree.o                 \
417         cfsetispeed.o           \
418         cfsetospeed.o           \
419         cftime.o                \
420         clock.o                 \
421         closedir.o              \
422         closefrom.o             \
423         confstr.o               \
424         crypt.o                 \
425         csetlen.o               \
426         ctime.o                 \
427         ctime_r.o               \
428         daemon.o                \
429         deflt.o                 \
430         directio.o              \
431         dirname.o               \
432         div.o                   \
433         drand48.o               \
434         dup.o                   \
435         env_data.o              \
436         err.o                   \
437         errno.o                 \
438         euclen.o                \
439         event_port.o            \
440         execvp.o                \
441         fattach.o               \
442         fdetach.o               \
443         fdopendir.o             \
444         ffs.o                   \
445         fls.o                   \
446         fmtmsg.o                \
447         ftime.o                 \
448         ftok.o                  \
449         ftw.o                   \
450         gcvt.o                  \
451         getauxv.o               \
452         getcwd.o                \
453         getdate_err.o           \
454         getdtblsize.o           \
455         getenv.o                \
456         getexecname.o           \
```



```
457         getgrnam.o              \
458         getgrnam_r.o            \
459         gethostid.o             \
460         gethostname.o           \
461         gethz.o                 \
462         getisax.o               \
463         getloadavg.o            \
464         getlogin.o              \
465         getmntent.o             \
466         getnetgrent.o           \
467         get_nprocs.o            \
468         getopt.o                \
469         getopt_long.o           \
470         getpagesize.o           \
471         getpw.o                 \
472         getpwnam.o              \
473         getpwnam_r.o            \
474         getrusage.o             \
475         getspent.o              \
476         getspent_r.o            \
477         getsubopt.o             \
478         gettxt.o                \
479         getusershell.o          \
480         getut.o                 \
481         getutx.o                \
482         getvfsent.o             \
483         getwd.o                 \
484         getwidth.o              \
485         getxby_door.o           \
486         gtxt.o                  \
487         hsearch.o               \
488         iconv.o                 \
489         imaxabs.o               \
490         imaxdiv.o               \
491         index.o                 \
492         initgroups.o            \
493         insque.o                \
494         isaexec.o               \
495         isastream.o             \
496         isatty.o                \
497         killpg.o                \
498         klpdlib.o               \
499         l64a.o                  \
500         lckpwdf.o               \
501         lconstants.o            \
502         ldivide.o               \
503         lexp10.o                \
504         lfind.o                 \
505         lfmt.o                  \
506         lfmt_log.o              \
507         llabs.o                 \
508         lldiv.o                 \
509         llog10.o                \
510         lltostr.o               \
511         localtime.o             \
512         lsearch.o               \
513         madvise.o               \
514         malloc.o                \
515         memalign.o              \
516         memmem.o                \
517         mkdev.o                 \
518         mkdtemp.o               \
519         mkfifo.o                \
520         mkstemp.o               \
521         mktemp.o                \
522         mlock.o                 \
```

```
523         mlockall.o              \
524         mon.o                   \
525         msync.o                 \
526         munlock.o               \
527         munlockall.o            \
528         ndbm.o                  \
529         nftw.o                  \
530         nlspath_checks.o        \
531         nsparse.o               \
532         nss_common.o            \
533         nss_dbdefs.o            \
534         nss_deffinder.o         \
535         opendir.o               \
536         opt_data.o              \
537         perror.o                \
538         pfmt.o                  \
539         pfmt_data.o             \
540         pfmt_print.o            \
541         pipe.o                  \
542         plock.o                 \
543         poll.o                  \
544         posix_fadvise.o         \
545         posix_fallocate.o       \
546         posix_madvise.o         \
547         posix_memalign.o        \
548         priocntl.o              \
549         privlib.o               \
550         priv_str_xlate.o        \
551         psiginfo.o              \
552         psignal.o               \
553         pt.o                    \
554         putpwent.o              \
555         putspent.o              \
556         raise.o                 \
557         rand.o                  \
558         random.o                \
559         rctlops.o               \
560         readdir.o               \
561         readdir_r.o             \
562         realpath.o              \
563         reboot.o                \
564         regexpr.o               \
565         remove.o                \
566         rewinddir.o             \
567         rindex.o                \
568         scandir.o               \
569         seekdir.o               \
570         select.o                \
571         select_large_fdset.o    \
572         setlabel.o              \
573         setpriority.o           \
574         settimeofday.o          \
575         sh_locks.o              \
576         sigflag.o               \
577         siglist.o               \
578         sigsend.o               \
579         sigsetops.o             \
580         ssignal.o               \
581         stack.o                 \
582         stpcpy.o                \
583         stpncpy.o               \
584         str2sig.o               \
585         strcase_charmap.o       \
586         strcat.o                \
587         strchrnul.o             \
588         strcspn.o               \
```



```
589         strdup.o                \
590         strerror.o              \
591         strlcat.o               \
592         strncat.o               \
593         strndup.o               \
594         strpbrk.o               \
595         strrchr.o               \
596         strsep.o                \
597         strsignal.o             \
598         strspn.o                \
599         strstr.o                \
600         strtod.o                \
601         strtoimax.o             \
602         strtok.o                \
603         strtok_r.o              \
604         strtoumax.o             \
605         swab.o                  \
606         swapctl.o               \
607         sysconf.o               \
608         syslog.o                \
609         tcdrain.o               \
610         tcflow.o                \
611         tcflush.o               \
612         tcgetattr.o             \
613         tcgetpgrp.o             \
614         tcgetsid.o              \
615         tcsendbreak.o           \
616         tcsetattr.o             \
617         tcsetpgrp.o             \
618         tell.o                  \
619         telldir.o               \
620         tfind.o                 \
621         time_data.o             \
622         time_gdata.o            \
623         tls_data.o              \
624         truncate.o              \
625         tsdalloc.o              \
626         tsearch.o               \
627         ttyname.o               \
628         ttyslot.o               \
629         ualarm.o                \
630         ucred.o                 \
631         valloc.o                \
632         vlfmt.o                 \
633         vpfmt.o                 \
634         waitpid.o               \
635         walkstack.o             \
636         wdata.o                 \
637         xgetwidth.o             \
638         xpg4.o                  \
639         xpg6.o

641 PORTPRINT_W=                    \
642         doprnt_w.o

644 PORTPRINT=                     \
645         asprintf.o              \
646         doprnt.o                \
647         fprintf.o               \
648         printf.o                \
649         snprintf.o              \
650         sprintf.o               \
651         vfprintf.o              \
652         vprintf.o               \
653         vsnprintf.o             \
654         vsprintf.o              \
```

```
655         vwprintf.o              \
656         wprintf.o

658 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
659 PORTPRINT_C89=                  \
660         vfprintf_c89.o          \
661         vprintf_c89.o           \
662         vsnprintf_c89.o         \
663         vsprintf_c89.o          \
664         vwprintf_c89.o

666 PORTSTDIO_C89=                  \
667         vscanf_c89.o            \
668         vwscanf_c89.o

670 # portable stdio objects that contain large file interfaces.
671 # Note: fopen64 is a special case, as we build it small.
672 PORTSTDIO64=                    \
673         fopen64.o               \
674         fpos64.o

676 PORTSTDIO_W=                    \
677         doscan_w.o

679 PORTSTDIO=                      \
680         __extensions.o          \
681         _endopen.o              \
682         _filbuf.o               \
683         _findbuf.o              \
684         _flsbuf.o               \
685         _wrtchk.o               \
686         clearerr.o              \
687         ctermid.o               \
688         ctermid_r.o             \
689         cuserid.o               \
690         data.o                  \
691         doscan.o                \
692         fdopen.o                \
693         feof.o                  \
694         ferror.o                \
695         fgetc.o                 \
696         fgets.o                 \
697         fileno.o                \
698         flockf.o                \
699         flush.o                 \
700         fopen.o                 \
701         fpos.o                  \
702         fputc.o                 \
703         fputs.o                 \
704         fread.o                 \
705         fseek.o                 \
706         fseeko.o                \
707         ftell.o                 \
708         ftello.o                \
709         fwrite.o                \
710         getc.o                  \
711         getchar.o               \
712         getline.o               \
713         getpass.o               \
714         gets.o                  \
715         getw.o                  \
716         popen.o                 \
717         putc.o                  \
718         putchar.o               \
719         puts.o                  \
720         putw.o                  \
```



```
721         rewind.o                \
722         scanf.o                 \
723         setbuf.o                \
724         setbuffer.o             \
725         setvbuf.o               \
726         system.o                \
727         tempnam.o               \
728         tmpfile.o               \
729         tmpnam_r.o              \
730         ungetc.o                \
731         mse.o                   \
732         vscanf.o                \
733         vwscanf.o               \
734         wscanf.o

736 PORTI18N=                       \
737         getwchar.o              \
738         putwchar.o              \
739         putws.o                 \
740         strtows.o               \
741         wcsnlen.o               \
742         wcstoimax.o             \
743         wcstol.o                \
744         wcstoul.o               \
745         wcswcs.o                \
746         wscat.o                 \
747         wschr.o                 \
748         wscmp.o                 \
749         wscpy.o                 \
750         wscspn.o                \
751         wsdup.o                 \
752         wslen.o                 \
753         wsncat.o                \
754         wsncmp.o                \
755         wsncpy.o                \
756         wspbrk.o                \
757         wsprintf.o              \
758         wsrchr.o                \
759         wsscanf.o               \
760         wsspn.o                 \
761         wstod.o                 \
762         wstok.o                 \
763         wstol.o                 \
764         wstoll.o                \
765         wsxfrm.o                \
766         wmemchr.o               \
767         wmemcmp.o               \
768         wmemcpy.o               \
769         wmemmove.o              \
770         wmemset.o               \
771         wcsstr.o                \
772         gettext.o               \
773         gettext_real.o          \
774         gettext_util.o          \
775         gettext_gnu.o           \
776         plural_parser.o         \
777         wdresolve.o             \
778         _ctype.o                \
779         isascii.o               \
780         toascii.o

782 PORTI18N_COND=                  \
783         wcstol_longlong.o       \
784         wcstoul_longlong.o

786 PORTLOCALE=                    \
```

```
787         big5.o                  \
788         btowc.o                 \
789         collate.o               \
790         collcmp.o               \
791         euc.o                   \
792         fnmatch.o               \
793         fgetwc.o                \
794         fgetws.o                \
795         fix_grouping.o          \
796         fputwc.o                \
797         fputws.o                \
798         fwide.o                 \
799         gb18030.o               \
800         gb2312.o                \
801         gbk.o                   \
802         getdate.o               \
803         isdigit.o               \
804         iswctype.o              \
805         ldpart.o                \
806         lmessages.o             \
807         lnumeric.o              \
808         lmonetary.o             \
809         localeimpl.o            \
810         localeconv.o            \
811         mbftowc.o               \
812         mblen.o                 \
813         mbrlen.o                \
814         mbrtowc.o               \
815         mbsinit.o               \
816         mbsnrtowcs.o            \
817         mbsrtowcs.o             \
818         mbstowcs.o              \
819         mbtowc.o                \
820         mskanji.o               \
821         nextwctype.o            \
822         nl_langinfo.o           \
823         none.o                  \
824         regcomp.o               \
825         regfree.o               \
826         regerror.o              \
827         regexec.o               \
828         rune.o                  \
829         runetype.o              \
830         setlocale.o             \
831         setrunelocale.o         \
832         strcasecmp.o            \
833         strcasestr.o            \
834         strcoll.o               \
835         strfmon.o               \
836         strftime.o              \
837         strncasecmp.o           \
838         strptime.o              \
839         strxfrm.o               \
840         table.o                 \
841         timelocal.o             \
842         tolower.o               \
843         towlower.o              \
844         ungetwc.o               \
845         utf8.o                  \
846         wcrtomb.o               \
847         wcscasecmp.o            \
848         wcscoll.o               \
849         wcsftime.o              \
850         wcsnrtombs.o            \
851         wcsrtombs.o             \
852         wcstombs.o              \
```



```
853         wcswidth.o              \
854         wcsxfrm.o               \
855         wctob.o                 \
856         wctomb.o                \
857         wctrans.o               \
858         wctype.o                \
859         wcwidth.o               \
860         wscol.o

862 AIOOBJS=                        \
863         aio.o                   \
864         aio_alloc.o             \
865         posix_aio.o

867 RTOBJS=                         \
868         clock_timer.o           \
869         mqueue.o                \
870         pos4obj.o               \
871         sched.o                 \
872         sem.o                   \
873         shm.o                   \
874         sigev_thread.o

876 TPOOLOBJS=                      \
877         thread_pool.o

879 THREADSOBJS=                    \
880         alloc.o                 \
881         assfail.o               \
882         cancel.o                \
883         door_calls.o            \
884         tmem.o                  \
885         pthr_attr.o             \
886         pthr_barrier.o          \
887         pthr_cond.o             \
888         pthr_mutex.o            \
889         pthr_rwlock.o           \
890         pthread.o               \
891         rwlock.o                \
892         scalls.o                \
893         sema.o                  \
894         sigaction.o             \
895         spawn.o                 \
896         synch.o                 \
897         tdb_agent.o             \
898         thr.o                   \
899         thread_interface.o      \
900         tls.o                   \
901         tsd.o

903 THREADSMACHOBJS=                \
904         machdep.o

906 THREADSASMOBJS=                 \
907         asm_subr.o

909 UNICODEOBJS=                    \
910         u8_textprep.o           \
911         uconv.o

913 UNWINDMACHOBJS=                 \
914         unwind.o

916 UNWINDASMOBJS=                  \
917         unwind_frame.o
```

```
# objects that implement the transitional large file API
PORTSYS64=                      \
        lockf64.o               \
        stat64.o

PORTSYS=                        \
        _autofssys.o            \
        access.o                \
        acctctl.o               \
        bsd_signal.o            \
        chmod.o                 \
        chown.o                 \
        corectl.o               \
        exacctsys.o             \
        execl.o                 \
        execle.o                \
        execv.o                 \
        fcntl.o                 \
        fexecve.o               \
        getpagesizes.o          \
        getpeerucred.o          \
        inst_sync.o             \
        issetugid.o             \
        label.o                 \
        link.o                  \
        lockf.o                 \
        lwp.o                   \
        lwp_cond.o              \
        lwp_rwlock.o            \
        lwp_sigmask.o           \
        meminfosys.o            \
        mkdir.o                 \
        mknod.o                 \
        msgsys.o                \
        nfssys.o                \
        open.o                  \
        pgrpsys.o               \
        posix_sigwait.o         \
        ppriv.o                 \
        psetsys.o               \
        rctlsys.o               \
        readlink.o              \
        rename.o                \
        sbrk.o                  \
        semsys.o                \
        set_errno.o             \
        sharefs.o               \
        shmsys.o                \
        sidsys.o                \
        siginterrupt.o          \
        signal.o                \
        sigpending.o            \
        sigstack.o              \
        stat.o                  \
        symlink.o               \
        tasksys.o               \
        time.o                  \
        time_util.o             \
        ucontext.o              \
        unlink.o                \
        ustat.o                 \
        utimesys.o              \
        zone.o

PORTREGEX=                      \
        glob.o                  \
```



```
        regcmp.o                \
        regex.o                 \
        wordexp.o

VALUES= values-Xa.o

MOSTOBJS=                       \
        $(STRETS)               \
        $(CRTOBJS)              \
        $(DYNOBJS)              \
        $(FPOBJS)               \
        $(FPASMOBJS)            \
        $(ATOMICOBJS)           \
        $(XATTROBJS)            \
        $(COMOBJS)              \
        $(DTRACEOBJS)           \
        $(GENOBJS)              \
        $(PRFOBJS)              \
        $(PORTFP)               \
        $(PORTGEN)              \
        $(PORTGEN64)            \
        $(PORTI18N)             \
        $(PORTI18N_COND)        \
        $(PORTLOCALE)           \
        $(PORTPRINT)            \
        $(PORTPRINT_C89)        \
        $(PORTPRINT_W)          \
        $(PORTREGEX)            \
        $(PORTSTDIO)            \
        $(PORTSTDIO64)          \
        $(PORTSTDIO_C89)        \
        $(PORTSTDIO_W)          \
        $(PORTSYS)              \
        $(PORTSYS64)            \
        $(AIOOBJS)              \
        $(RTOBJS)               \
        $(TPOOLOBJS)            \
        $(THREADSOBJS)          \
        $(THREADSMACHOBJS)      \
        $(THREADSASMOBJS)       \
        $(UNICODEOBJS)          \
        $(UNWINDMACHOBJS)       \
        $(UNWINDASMOBJS)        \
        $(COMSYSOBJS)           \
        $(SYSOBJS)              \
        $(COMSYSOBJS64)         \
        $(SYSOBJS64)            \
        $(VALUES)

TRACEOBJS=                      \
        plockstat.o

# NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
# modules whose source is provided in the $(SRC)/lib/common directory.
# This must be done because otherwise the Sun C compiler would insert
# its own versions of these modules and those versions contain code
# to call out to C++ initialization functions.  Such C++ initialization
# functions can call back into libc before thread initialization is
# complete and this leads to segmentation violations and other problems.
# Since libc contains no C++ code, linking with the minimal crti.o and
# crtn.o modules is safe and avoids the problems described above.
OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
CRTSRCS= ../../common/sparc

# include common library definitions
include $(SRC)/lib/Makefile.lib
```

```
1052 # we need to override the default SONAME here because we might
1053 # be building a variant object (still libc.so.1, but different filename)
1054 SONAME = libc.so.1

1056 CFLAGS += $(CCVERBOSE)

1058 # This is necessary to avoid problems with calling _ex_unwind().
1059 # We probably don't want any inlining anyway.
1060 CFLAGS += -xinline=

1062 CERRWARN += -_gcc=-Wno-parentheses
1063 CERRWARN += -_gcc=-Wno-switch
1064 CERRWARN += -_gcc=-Wno-uninitialized
1065 CERRWARN += -_gcc=-Wno-unused-value
1066 CERRWARN += -_gcc=-Wno-unused-label
1067 CERRWARN += -_gcc=-Wno-unused-variable
1068 CERRWARN += -_gcc=-Wno-type-limits
1069 CERRWARN += -_gcc=-Wno-char-subscripts
1070 CERRWARN += -_gcc=-Wno-clobbered
1071 CERRWARN += -_gcc=-Wno-unused-function
1072 CERRWARN += -_gcc=-Wno-address

1074 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1075 # enables ASSERT() checking in the threads portion of the library.
1076 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1077 THREAD_DEBUG =
1078 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1080 # Make string literals read-only to save memory.
1081 CFLAGS += $(XSTRCONST)

1083 ALTPICS= $(TRACEOBJS:%=pics/%)

1085 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1087 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1089 CFLAGS +=       $(EXTN_CFLAGS)
1090 CPPFLAGS=       -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1091                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1092 ASFLAGS=        $(EXTN_ASFLAGS) -K pic -P -D__STDC__ -D_ASM $(CPPFLAGS) $(sparc_

1094 # As a favor to the dtrace syscall provider, libc still calls the
1095 # old syscall traps that have been obsoleted by the *at() interfaces.
1096 # Delete this to compile libc using only the new *at() system call traps
1097 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1099 # Inform the run-time linker about libc specialized initialization
1100 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1101 DYNFLAGS +=     $(RTLDINFO)

1103 # Force libc's internal references to be resolved immediately upon loading
1104 # in order to avoid critical region problems.  Since almost all libc symbols
1105 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1106 DYNFLAGS +=     -znow

1108 DYNFLAGS +=     -e __rtboot
1109 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1111 # Inform the kernel about the initial DTrace area (in case
1112 # libc is being used as the interpreter / runtime linker).
1113 DTRACE_DATA =   -zdtrace=dtrace_data
1114 DYNFLAGS +=     $(DTRACE_DATA)

1116 # DTrace needs an executable data segment.
```



```
1117 MAPFILE.NED=

1119 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1121 # Override this top level flag so the compiler builds in its native
1122 # C99 mode.  This has been enabled to support the complex arithmetic
1123 # added to libc.
1124 C99MODE=        $(C99_ENABLE)

1126 # libc method of building an archive
1127 # The "$(GREP) -v ' L '" part is necessary only until
1128 # lorder is fixed to ignore thread-local variables.
1129 BUILD.AR= $(RM) $@ ; \
1130         $(AR) q $@ '$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1132 # extra files for the clean target
1133 CLEANFILES=                     \
1134         $(LIBCDIR)/port/gen/errlst.c    \
1135         $(LIBCDIR)/port/gen/new_list.c  \
1136         assym.h                 \
1137         genassym                \
1138         $(LIBCBASE)/crt/_rtld.s         \
1139         $(LIBCBASE)/crt/_rtbootld.s             \
1140         pics/_rtbootld.o        \
1141         pics/crti.o             \
1142         pics/crtn.o             \
1143         $(ALTPICS)

1145 CLOBBERFILES += $(LIB_PIC)

1147 # list of C source for lint
1148 SRCS=                                                   \
1149         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1150         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1151         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1152         $(DTRACEOBJS:%.o=$(SRC)/common/dtrace/%.c)      \
1153         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1154         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1155         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1156         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1157         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1158         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1159         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1160         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1161         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1162         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1163         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1164         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1165         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1166         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1167         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1168         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1169         $(LIBCBASE)/crt/_ftou.c                         \
1170         $(LIBCBASE)/gen/_xregs_clrptr.c                 \
1171         $(LIBCBASE)/gen/byteorder.c                     \
1172         $(LIBCBASE)/gen/ecvt.c                          \
1173         $(LIBCBASE)/gen/getctxt.c                       \
1174         $(LIBCBASE)/gen/lmul.c                          \
1175         $(LIBCBASE)/gen/makectxt.c                      \
1176         $(LIBCBASE)/gen/siginfolst.c                    \
1177         $(LIBCBASE)/gen/siglongjmp.c                    \
1178         $(LIBCBASE)/gen/swapctxt.c                      \
1179         $(LIBCBASE)/sys/ptrace.c                        \
1180         $(LIBCBASE)/sys/uadmin.c

1182 # conditional assignments
```

```
1183 $(DYNLIB) := CRTI = crti.o
1184 $(DYNLIB) := CRTN = crtn.o

1186 # Files which need the threads .il inline template
1187 TIL=                            \
1188         aio.o                   \
1189         alloc.o                 \
1190         assfail.o               \
1191         atexit.o                \
1192         atfork.o                \
1193         cancel.o                \
1194         door_calls.o            \
1195         err.o                   \
1196         errno.o                 \
1197         getctxt.o               \
1198         lwp.o                   \
1199         ma.o                    \
1200         machdep.o               \
1201         posix_aio.o             \
1202         pthr_attr.o             \
1203         pthr_barrier.o          \
1204         pthr_cond.o             \
1205         pthr_mutex.o            \
1206         pthr_rwlock.o           \
1207         pthread.o               \
1208         rand.o                  \
1209         rwlock.o                \
1210         scalls.o                \
1211         sched.o                 \
1212         sema.o                  \
1213         sigaction.o             \
1214         sigev_thread.o          \
1215         spawn.o                 \
1216         stack.o                 \
1217         swapctxt.o              \
1218         synch.o                 \
1219         tdb_agent.o             \
1220         thr.o                   \
1221         thread_interface.o      \
1222         thread_pool.o           \
1223         tls.o                   \
1224         tsd.o                   \
1225         unwind.o

1227 $(TIL:%=pics/%) := CFLAGS += $(LIBCBASE)/threads/sparc.il

1229 # special kludge for inlines with 'cas':
1230 pics/rwlock.o pics/synch.o pics/lwp.o pics/door_calls.o := \
1231         sparc_CFLAGS += -_gcc=-Wa,-xarch=v8plus

1233 # Files in port/fp subdirectory that need base.il inline template
1234 IL=                             \
1235         __flt_decim.o           \
1236         decimal_bin.o

1238 $(IL:%=pics/%) := CFLAGS += $(LIBCBASE)/fp/base.il

1240 # Files in fp subdirectory which need __quad.il inline template
1241 QIL=                            \
1242         _Q_add.o                \
1243         _Q_cmp.o                \
1244         _Q_cmpe.o               \
1245         _Q_div.o                \
1246         _Q_dtoq.o               \
1247         _Q_fcc.o                \
1248         _Q_mul.o                \
```



```
1249         _Q_qtod.o               \
1250         _Q_qtoi.o               \
1251         _Q_qtos.o               \
1252         _Q_qtou.o               \
1253         _Q_sqrt.o               \
1254         _Q_stoq.o               \
1255         _Q_sub.o

1257 $(QIL:%=pics/%) := CFLAGS += $(LIBCDIR)/$(MACH)/fp/__quad.il
1258 pics/_Q%.o := sparc_COPTFLAG = -xO4 -dalign
1259 pics/__quad%.o := sparc_COPTFLAG = -xO4 -dalign

1261 # large-file-aware components that should be built large

1263 $(COMSYSOBJS64:%=pics/%) := \
1264         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1266 $(SYSOBJS64:%=pics/%) := \
1267         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1269 $(PORTGEN64:%=pics/%) := \
1270         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1272 $(PORTSTDIO64:%=pics/%) := \
1273         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1275 $(PORTSYS64:%=pics/%) := \
1276         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1278 $(PORTSTDIO_W:%=pics/%) := \
1279         CPPFLAGS += -D_WIDE

1281 $(PORTPRINT_W:%=pics/%) := \
1282         CPPFLAGS += -D_WIDE

1284 # printf/scanf functions to support c89-sized intmax_t variables
1285 $(PORTPRINT_C89:%=pics/%) := \
1286         CPPFLAGS += -D_C89_INTMAX32

1288 $(PORTSTDIO_C89:%=pics/%) := \
1289         CPPFLAGS += -D_C89_INTMAX32

1291 $(PORTI18N_COND:%=pics/%) := \
1292         CPPFLAGS += -D_WCS_LONGLONG

1294 # Files which need extra optimization
1295 pics/getenv.o := sparc_COPTFLAG = -xO4

1297 .KEEP_STATE:

1299 all: $(LIBS) $(LIB_PIC)

1301 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1302 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1303 lint    :=      LINTFLAGS += -mn

1305 lint:
1306         @echo $(LINT.c) ... $(LDLIBS)
1307         @$(LINT.c) $(SRCS) $(LDLIBS)

1309 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1310 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1311 $(LINTLIB):= LINTFLAGS=-nvx

1313 # object files that depend on inline template
1314 $(TIL:%=pics/%): $(LIBCBASE)/threads/sparc.il
```

```
$(IL:%=pics/%): $(LIBCBASE)/fp/base.il
$(QIL:%=pics/%): $(LIBCDIR)/$(MACH)/fp/__quad.il

# include common libc targets
include $(LIBCDIR)/Makefile.targ

# We need to strip out all CTF and DOF data from the static library
$(LIB_PIC) := DIR = pics
$(LIB_PIC): pics $$(PICS)
        $(BUILD.AR)
        $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
        $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
        $(AR) -ts $@ > /dev/null
        $(POST_PROCESS_A)

# special cases
$(STRETS:%=pics/%): $(LIBCBASE)/crt/stret.s
        $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=%) $(LIBCBASE)/crt/stret.s -o $@
        $(POST_PROCESS_O)

$(LIBCBASE)/crt/_rtbootld.s:    $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.
        $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S -K pic \
            $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
        $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
        $(RM) $(LIBCBASE)/crt/_rtld.s

# partially built from C source
pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
        $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
        $(CTFCONVERT_O)

ASSYMDEP_OBJS=                  \
        _lwp_mutex_unlock.o     \
        _stack_grow.o           \
        asm_subr.o              \
        setjmp.o                \
        smt_pause.o             \
        tls_get_addr.o          \
        unwind_frame.o          \
        vforkx.o

$(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

$(ASSYMDEP_OBJS:%=pics/%): assym.h

# assym.h build rules

assym.h := CFLAGS += -g

GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

genassym: $(GENASSYM_C)
        $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
                $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

assym.h: $(OFFSETS) genassym
        $(OFFSETS_CREATE) <$(OFFSETS) >$@
        ./genassym >>$@

# derived C source and related explicit dependencies
$(LIBCDIR)/port/gen/errlst.c + \
$(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
        cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist
```



```
pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   24531 Mon Apr  6 11:47:25 2015
new/usr/src/lib/libc/sparcv9/Makefile.com
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBCDIR=        $(SRC)/lib/libc
LIB_PIC=        libc_pic.a
VERS=           .1
CPP=            /usr/lib/cpp
TARGET_ARCH=    sparc

# objects are grouped by source directory

# Symbol capabilities objects.
EXTPICS=                        \
        $(LIBCDIR)/capabilities/sun4u/sparcv9/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-opl/sparcv9/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-us3-hwcap1/sparcv9/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4u-us3-hwcap2/sparcv9/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4v-hwcap1/sparcv9/pics/symcap.o \
        $(LIBCDIR)/capabilities/sun4v-hwcap2/sparcv9/pics/symcap.o

# local objects
STRETS=

CRTOBJS=                        \
        __align_cpy_2.o         \
        __align_cpy_4.o         \
        __align_cpy_8.o         \
        _ftou.o                 \
        cerror.o

DYNOBJS=

FPOBJS=                         \
```



```
        _D_cplx_div.o           \
        _D_cplx_div_ix.o        \
        _D_cplx_div_rx.o        \
        _D_cplx_mul.o           \
        _F_cplx_div.o           \
        _F_cplx_div_ix.o        \
        _F_cplx_div_rx.o        \
        _F_cplx_mul.o           \
        _Q_add.o                \
        _Q_cmp.o                \
        _Q_cmpe.o               \
        _Q_cplx_div.o           \
        _Q_cplx_div_ix.o        \
        _Q_cplx_div_rx.o        \
        _Q_cplx_lr_div.o        \
        _Q_cplx_lr_div_ix.o     \
        _Q_cplx_lr_div_rx.o     \
        _Q_cplx_lr_mul.o        \
        _Q_cplx_mul.o           \
        _Q_div.o                \
        _Q_dtoq.o               \
        _Q_fcc.o                \
        _Q_itoq.o               \
        _Q_mul.o                \
        _Q_neg.o                \
        _Q_qtod.o               \
        _Q_qtoi.o               \
        _Q_qtos.o               \
        _Q_qtou.o               \
        _Q_scl.o                \
        _Q_sqrt.o               \
        _Q_stoq.o               \
        _Q_sub.o                \
        _Q_utoq.o

FPOBJS64=                       \
        _Qp_qtox.o              \
        _Qp_qtoux.o             \
        _Qp_xtoq.o              \
        _Qp_uxtoq.o             \
        __dtoul.o               \
        __ftoul.o

FPASMOBJS=                      \
        _Q_get_rp_rd.o          \
        __quad_mag64.o          \
        fpgetmask.o             \
        fpgetrnd.o              \
        fpgetsticky.o           \
        fpsetmask.o             \
        fpsetrnd.o              \
        fpsetsticky.o

$(__GNUC)FPASMOBJS +=           \
        __quad.o

ATOMICOBJS=                     \
        atomic.o

XATTROBJS=                      \
        xattr_common.o

COMOBJS=                        \
        bcmp.o                  \
        bcopy.o                 \
        bsearch.o               \
```

```
127         bzero.o                 \
128         memccpy.o               \
129         qsort.o                 \
130         strtol.o                \
131         strtoul.o               \
132         strtoll.o               \
133         strtoull.o

135 GENOBJS=                        \
136         _getsp.o                \
137         _xregs_clrptr.o         \
138         abs.o                   \
139         alloca.o                \
140         ascii_strcasecmp.o      \
141         byteorder.o             \
142         cuexit.o                \
143         ecvt.o                  \
144         getctxt.o               \
145         lock.o                  \
146         makectxt.o              \
147         memchr.o                \
148         memcmp.o                \
149         new_list.o              \
150         setjmp.o                \
151         siginfolst.o            \
152         siglongjmp.o            \
153         smt_pause.o             \
154         sparc_data.o            \
155         strchr.o                \
156         strcmp.o                \
157         strlcpy.o               \
158         strncmp.o               \
159         strncpy.o               \
160         strnlen.o               \
161         swapctxt.o              \
162         sync_instruction_memory.o

164 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
165 # This macro should ALWAYS be empty; native APIs are already 'large file'.
166 COMSYSOBJS64=

168 SYSOBJS64=

170 COMSYSOBJS=                     \
171         __clock_timer.o         \
172         __getloadavg.o          \
173         __rusagesys.o           \
174         __signotify.o           \
175         __sigrt.o               \
176         __time.o                \
177         _lgrp_home_fast.o       \
178         _lgrpsys.o              \
179         _nfssys.o               \
180         _portfs.o               \
181         _pset.o                 \
182         _rpcsys.o               \
183         _sigaction.o            \
184         _so_accept.o            \
185         _so_bind.o              \
186         _so_connect.o           \
187         _so_getpeername.o       \
188         _so_getsockname.o       \
189         _so_getsockopt.o        \
190         _so_listen.o            \
191         _so_recv.o              \
192         _so_recvfrom.o          \
```



```
193         _so_recvmsg.o           \
194         _so_send.o              \
195         _so_sendmsg.o           \
196         _so_sendto.o            \
197         _so_setsockopt.o        \
198         _so_shutdown.o          \
199         _so_socket.o            \
200         _so_socketpair.o        \
201         _sockconfig.o           \
202         acct.o                  \
203         acl.o                   \
204         adjtime.o               \
205         alarm.o                 \
206         brk.o                   \
207         chdir.o                 \
208         chroot.o                \
209         cladm.o                 \
210         close.o                 \
211         execve.o                \
212         exit.o                  \
213         facl.o                  \
214         fchdir.o                \
215         fchroot.o               \
216         fdsync.o                \
217         fpathconf.o             \
218         fstatfs.o               \
219         fstatvfs.o              \
220         getcpuid.o              \
221         getdents.o              \
222         getegid.o               \
223         geteuid.o               \
224         getgid.o                \
225         getgroups.o             \
226         gethrtime.o             \
227         getitimer.o             \
228         getmsg.o                \
229         getpid.o                \
230         getpmsg.o               \
231         getppid.o               \
232         getrlimit.o             \
233         getuid.o                \
234         gtty.o                  \
235         install_utrap.o         \
236         ioctl.o                 \
237         kaio.o                  \
238         kill.o                  \
239         llseek.o                \
240         lseek.o                 \
241         memcntl.o               \
242         mincore.o               \
243         mmap.o                  \
244         mmapobjsys.o            \
245         modctl.o                \
246         mount.o                 \
247         mprotect.o              \
248         munmap.o                \
249         nice.o                  \
250         ntp_adjtime.o           \
251         ntp_gettime.o           \
252         p_online.o              \
253         pathconf.o              \
254         pause.o                 \
255         pcsample.o              \
256         pipe2.o                 \
257         pollsys.o               \
258         pread.o                 \
```

```
259         preadv.o                \
260         priocntlset.o           \
261         processor_bind.o        \
262         processor_info.o        \
263         profil.o                \
264         putmsg.o                \
265         putpmsg.o               \
266         pwrite.o                \
267         pwritev.o               \
268         read.o                  \
269         readv.o                 \
270         resolvepath.o           \
271         seteguid.o              \
272         setgid.o                \
273         setgroups.o             \
274         setitimer.o             \
275         setreid.o               \
276         setrlimit.o             \
277         setuid.o                \
278         sigaltstk.o             \
279         sigprocmsk.o            \
280         sigsendset.o            \
281         sigsuspend.o            \
282         statfs.o                \
283         statvfs.o               \
284         stty.o                  \
285         sync.o                  \
286         sysconfig.o             \
287         sysfs.o                 \
288         sysinfo.o               \
289         syslwp.o                \
290         times.o                 \
291         ulimit.o                \
292         umask.o                 \
293         umount2.o               \
294         utssys.o                \
295         uucopy.o                \
296         vhangup.o               \
297         waitid.o                \
298         write.o                 \
299         writev.o                \
300         yield.o

302 SYSOBJS=                       \
303         __clock_gettime.o       \
304         __getcontext.o          \
305         __uadmin.o              \
306         _lwp_mutex_unlock.o     \
307         _stack_grow.o           \
308         door.o                  \
309         forkx.o                 \
310         forkallx.o              \
311         gettimeofday.o          \
312         sparc_utrap_install.o   \
313         syscall.o               \
314         tls_get_addr.o          \
315         uadmin.o                \
316         umount.o                \
317         uname.o                 \
318         vforkx.o

320 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
321 # This macro should ALWAYS be empty; native APIs are already 'large file'.
322 PORTGEN64=

324 # objects from source under $(LIBCDIR)/port
```



```
325 PORTFP=                         \
326         __flt_decim.o           \
327         __flt_rounds.o          \
328         __tbl_10_b.o            \
329         __tbl_10_h.o            \
330         __tbl_10_s.o            \
331         __tbl_2_b.o             \
332         __tbl_2_h.o             \
333         __tbl_2_s.o             \
334         __tbl_fdq.o             \
335         __tbl_tens.o            \
336         __x_power.o             \
337         _base_sup.o             \
338         aconvert.o              \
339         decimal_bin.o           \
340         double_decim.o          \
341         econvert.o              \
342         fconvert.o              \
343         file_decim.o            \
344         finite.o                \
345         fp_data.o               \
346         func_decim.o            \
347         gconvert.o              \
348         hex_bin.o               \
349         ieee_globals.o          \
350         pack_float.o            \
351         sigfpe.o                \
352         string_decim.o

354 PORTGEN=                        \
355         _env_data.o             \
356         _xftw.o                 \
357         a64l.o                  \
358         abort.o                 \
359         addsev.o                \
360         ascii_strncasecmp.o     \
361         assert.o                \
362         attrat.o                \
363         atof.o                  \
364         atoi.o                  \
365         atol.o                  \
366         atoll.o                 \
367         attropen.o              \
368         atexit.o                \
369         atfork.o                \
370         basename.o              \
371         calloc.o                \
372         catgets.o               \
373         catopen.o               \
374         cfgetispeed.o           \
375         cfgetospeed.o           \
376         cfree.o                 \
377         cfsetispeed.o           \
378         cfsetospeed.o           \
379         cftime.o                \
380         clock.o                 \
381         closedir.o              \
382         closefrom.o             \
383         confstr.o               \
384         crypt.o                 \
385         csetlen.o               \
386         ctime.o                 \
387         ctime_r.o               \
388         daemon.o                \
389         deflt.o                 \
390         directio.o              \
```

```
391         dirname.o               \
392         div.o                   \
393         drand48.o               \
394         dup.o                   \
395         env_data.o              \
396         err.o                   \
397         errno.o                 \
398         euclen.o                \
399         event_port.o            \
400         execvp.o                \
401         fattach.o               \
402         fdetach.o               \
403         fdopendir.o             \
404         ffs.o                   \
405         fls.o                   \
406         fmtmsg.o                \
407         ftime.o                 \
408         ftok.o                  \
409         ftw.o                   \
410         gcvt.o                  \
411         getauxv.o               \
412         getcwd.o                \
413         getdate_err.o           \
414         getdtblsize.o           \
415         getenv.o                \
416         getexecname.o           \
417         getgrnam.o              \
418         getgrnam_r.o            \
419         gethostid.o             \
420         gethostname.o           \
421         gethz.o                 \
422         getisax.o               \
423         getloadavg.o            \
424         getlogin.o              \
425         getmntent.o             \
426         getnetgrent.o           \
427         get_nprocs.o            \
428         getopt.o                \
429         getopt_long.o           \
430         getpagesize.o           \
431         getpw.o                 \
432         getpwnam.o              \
433         getpwnam_r.o            \
434         getrusage.o             \
435         getspent.o              \
436         getspent_r.o            \
437         getsubopt.o             \
438         gettxt.o                \
439         getusershell.o          \
440         getut.o                 \
441         getutx.o                \
442         getvfsent.o             \
443         getwd.o                 \
444         getwidth.o              \
445         getxby_door.o           \
446         gtxt.o                  \
447         hsearch.o               \
448         iconv.o                 \
449         imaxabs.o               \
450         imaxdiv.o               \
451         index.o                 \
452         initgroups.o            \
453         insque.o                \
454         isaexec.o               \
455         isastream.o             \
456         isatty.o                \
```



```
457         killpg.o                \
458         klpdlib.o               \
459         l64a.o                  \
460         lckpwdf.o               \
461         lconstants.o            \
462         ldivide.o               \
463         lexp10.o                \
464         lfind.o                 \
465         lfmt.o                  \
466         lfmt_log.o              \
467         lldiv.o                 \
468         llog10.o                \
469         lltostr.o               \
470         lmath.o                 \
471         localtime.o             \
472         lsearch.o               \
473         madvise.o               \
474         malloc.o                \
475         memalign.o              \
476         memmem.o                \
477         mkdev.o                 \
478         mkdtemp.o               \
479         mkfifo.o                \
480         mkstemp.o               \
481         mktemp.o                \
482         mlock.o                 \
483         mlockall.o              \
484         mon.o                   \
485         msync.o                 \
486         munlock.o               \
487         munlockall.o            \
488         ndbm.o                  \
489         nftw.o                  \
490         nlspath_checks.o        \
491         nsparse.o               \
492         nss_common.o            \
493         nss_dbdefs.o            \
494         nss_deffinder.o         \
495         opendir.o               \
496         opt_data.o              \
497         perror.o                \
498         pfmt.o                  \
499         pfmt_data.o             \
500         pfmt_print.o            \
501         pipe.o                  \
502         plock.o                 \
503         poll.o                  \
504         posix_fadvise.o         \
505         posix_fallocate.o       \
506         posix_madvise.o         \
507         posix_memalign.o        \
508         priocntl.o              \
509         privlib.o               \
510         priv_str_xlate.o        \
511         psiginfo.o              \
512         psignal.o               \
513         pt.o                    \
514         putpwent.o              \
515         putspent.o              \
516         raise.o                 \
517         rand.o                  \
518         random.o                \
519         rctlops.o               \
520         readdir.o               \
521         readdir_r.o             \
522         realpath.o              \
```

```
523         reboot.o                \
524         regexpr.o               \
525         remove.o                \
526         rewinddir.o             \
527         rindex.o                \
528         scandir.o               \
529         seekdir.o               \
530         select.o                \
531         setlabel.o              \
532         setpriority.o           \
533         settimeofday.o          \
534         sh_locks.o              \
535         sigflag.o               \
536         siglist.o               \
537         sigsend.o               \
538         sigsetops.o             \
539         ssignal.o               \
540         stack.o                 \
541         stpcpy.o                \
542         stpncpy.o               \
543         str2sig.o               \
544         strcase_charmap.o       \
545         strcat.o                \
546         strchrnul.o             \
547         strcspn.o               \
548         strdup.o                \
549         strerror.o              \
550         strlcat.o               \
551         strncat.o               \
552         strndup.o               \
553         strpbrk.o               \
554         strrchr.o               \
555         strsep.o                \
556         strsignal.o             \
557         strspn.o                \
558         strstr.o                \
559         strtod.o                \
560         strtoimax.o             \
561         strtok.o                \
562         strtok_r.o              \
563         strtoumax.o             \
564         swab.o                  \
565         swapctl.o               \
566         sysconf.o               \
567         syslog.o                \
568         tcdrain.o               \
569         tcflow.o                \
570         tcflush.o               \
571         tcgetattr.o             \
572         tcgetpgrp.o             \
573         tcgetsid.o              \
574         tcsendbreak.o           \
575         tcsetattr.o             \
576         tcsetpgrp.o             \
577         tell.o                  \
578         telldir.o               \
579         tfind.o                 \
580         time_data.o             \
581         time_gdata.o            \
582         tls_data.o              \
583         truncate.o              \
584         tsdalloc.o              \
585         tsearch.o               \
586         ttyname.o               \
587         ttyslot.o               \
588         ualarm.o                \
```



```
589         ucred.o                 \
590         valloc.o                \
591         vlfmt.o                 \
592         vpfmt.o                 \
593         waitpid.o               \
594         walkstack.o             \
595         wdata.o                 \
596         xgetwidth.o             \
597         xpg4.o                  \
598         xpg6.o

600 PORTPRINT_W=                     \
601         doprnt_w.o

603 PORTPRINT=                       \
604         asprintf.o              \
605         doprnt.o                \
606         fprintf.o               \
607         printf.o                \
608         snprintf.o              \
609         sprintf.o               \
610         vfprintf.o              \
611         vprintf.o               \
612         vsnprintf.o             \
613         vsprintf.o              \
614         vwprintf.o              \
615         wprintf.o

617 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
618 # This macro should ALWAYS be empty; native APIs are already 'large file'.
619 PORTSTDIO64=

621 PORTSTDIO_W=                     \
622         doscan_w.o

624 PORTSTDIO=                       \
625         __extensions.o          \
626         _endopen.o              \
627         _filbuf.o               \
628         _findbuf.o              \
629         _flsbuf.o               \
630         _wrtchk.o               \
631         clearerr.o              \
632         ctermid.o               \
633         ctermid_r.o             \
634         cuserid.o               \
635         data.o                  \
636         doscan.o                \
637         fdopen.o                \
638         feof.o                  \
639         ferror.o                \
640         fgetc.o                 \
641         fgets.o                 \
642         fileno.o                \
643         flockf.o                \
644         flush.o                 \
645         fopen.o                 \
646         fpos.o                  \
647         fputc.o                 \
648         fputs.o                 \
649         fread.o                 \
650         fseek.o                 \
651         fseeko.o                \
652         ftell.o                 \
653         ftello.o                \
654         fwrite.o                \
```

```
655         getc.o                  \
656         getchar.o               \
657         getline.o               \
658         getpass.o               \
659         gets.o                  \
660         getw.o                  \
661         popen.o                 \
662         putc.o                  \
663         putchar.o               \
664         puts.o                  \
665         putw.o                  \
666         rewind.o                \
667         scanf.o                 \
668         setbuf.o                \
669         setbuffer.o             \
670         setvbuf.o               \
671         system.o                \
672         tempnam.o               \
673         tmpfile.o               \
674         tmpnam_r.o              \
675         ungetc.o                \
676         mse.o                   \
677         vscanf.o                \
678         vwscanf.o               \
679         wscanf.o

681 PORTI18N=                       \
682         getwchar.o              \
683         putwchar.o              \
684         putws.o                 \
685         strtows.o               \
686         wcsnlen.o               \
687         wcstoimax.o             \
688         wcstol.o                \
689         wcstoul.o               \
690         wcswcs.o                \
691         wscat.o                 \
692         wschr.o                 \
693         wscmp.o                 \
694         wscpy.o                 \
695         wscspn.o                \
696         wsdup.o                 \
697         wslen.o                 \
698         wsncat.o                \
699         wsncmp.o                \
700         wsncpy.o                \
701         wspbrk.o                \
702         wsprintf.o              \
703         wsrchr.o                \
704         wsscanf.o               \
705         wsspn.o                 \
706         wstod.o                 \
707         wstok.o                 \
708         wstol.o                 \
709         wstoll.o                \
710         wsxfrm.o                \
711         wmemchr.o               \
712         wmemcmp.o               \
713         wmemcpy.o               \
714         wmemmove.o              \
715         wmemset.o               \
716         wcsstr.o                \
717         gettext.o               \
718         gettext_real.o          \
719         gettext_util.o          \
720         gettext_gnu.o           \
```



```
721         plural_parser.o         \
722         wdresolve.o             \
723         _ctype.o                \
724         isascii.o               \
725         toascii.o

727 PORTI18N_COND=                  \
728         wcstol_longlong.o       \
729         wcstoul_longlong.o

731 PORTLOCALE=                    \
732         big5.o                  \
733         btowc.o                 \
734         collate.o               \
735         collcmp.o               \
736         euc.o                   \
737         fnmatch.o               \
738         fgetwc.o                \
739         fgetws.o                \
740         fix_grouping.o          \
741         fputwc.o                \
742         fputws.o                \
743         fwide.o                 \
744         gb18030.o               \
745         gb2312.o                \
746         gbk.o                   \
747         getdate.o               \
748         isdigit.o               \
749         iswctype.o              \
750         ldpart.o                \
751         lmessages.o             \
752         lnumeric.o              \
753         lmonetary.o             \
754         localeconv.o            \
755         localeimpl.o            \
756         mbftowc.o               \
757         mblen.o                 \
758         mbrlen.o                \
759         mbrtowc.o               \
760         mbsinit.o               \
761         mbsnrtowcs.o            \
762         mbsrtowcs.o             \
763         mbstowcs.o              \
764         mbtowc.o                \
765         mskanji.o               \
766         nextwctype.o            \
767         nl_langinfo.o           \
768         none.o                  \
769         regcomp.o               \
770         regfree.o               \
771         regerror.o              \
772         regexec.o               \
773         rune.o                  \
774         runetype.o              \
775         setlocale.o             \
776         setrunelocale.o         \
777         strcasecmp.o            \
778         strcasestr.o            \
779         strcoll.o               \
780         strfmon.o               \
781         strftime.o              \
782         strncasecmp.o           \
783         strptime.o              \
784         strxfrm.o               \
785         table.o                 \
786         timelocal.o             \
```

```
787         tolower.o               \
788         towlower.o              \
789         ungetwc.o               \
790         utf8.o                  \
791         wcrtomb.o               \
792         wcscasecmp.o            \
793         wcscoll.o               \
794         wcsftime.o              \
795         wcsnrtombs.o            \
796         wcsrtombs.o             \
797         wcstombs.o              \
798         wcswidth.o              \
799         wcsxfrm.o               \
800         wctob.o                 \
801         wctomb.o                \
802         wctrans.o               \
803         wctype.o                \
804         wcwidth.o               \
805         wscol.o

807 AIOOBJS=                        \
808         aio.o                   \
809         aio_alloc.o             \
810         posix_aio.o

812 RTOBJS=                         \
813         clock_timer.o           \
814         mqueue.o                \
815         pos4obj.o               \
816         sched.o                 \
817         sem.o                   \
818         shm.o                   \
819         sigev_thread.o

821 TPOOLOBJS=                      \
822         thread_pool.o

824 THREADSOBJS=                    \
825         alloc.o                 \
826         assfail.o               \
827         cancel.o                \
828         door_calls.o            \
829         tmem.o                  \
830         pthr_attr.o             \
831         pthr_barrier.o          \
832         pthr_cond.o             \
833         pthr_mutex.o            \
834         pthr_rwlock.o           \
835         pthread.o               \
836         rwlock.o                \
837         scalls.o                \
838         sema.o                  \
839         sigaction.o             \
840         spawn.o                 \
841         synch.o                 \
842         tdb_agent.o             \
843         thr.o                   \
844         thread_interface.o      \
845         tls.o                   \
846         tsd.o

848 THREADSMACHOBJS=                \
849         machdep.o

851 THREADSASMOBJS=                 \
852         asm_subr.o
```



```
854 UNICODEOBJS=                    \
855         u8_textprep.o           \
856         uconv.o

858 UNWINDMACHOBJS=                 \
859         unwind.o

861 UNWINDASMOBJS=                  \
862         unwind_frame.o

864 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
865 # This macro should ALWAYS be empty; native APIs are already 'large file'.
866 PORTSYS64=

868 PORTSYS=                        \
869         _autofssys.o            \
870         access.o                \
871         acctctl.o               \
872         bsd_signal.o            \
873         chmod.o                 \
874         chown.o                 \
875         corectl.o               \
876         exacctsys.o             \
877         execl.o                 \
878         execle.o                \
879         execv.o                 \
880         fcntl.o                 \
881         fexecve.o               \
882         getpagesizes.o          \
883         getpeerucred.o          \
884         inst_sync.o             \
885         issetugid.o             \
886         label.o                 \
887         link.o                  \
888         lockf.o                 \
889         lwp.o                   \
890         lwp_cond.o              \
891         lwp_rwlock.o            \
892         lwp_sigmask.o           \
893         meminfosys.o            \
894         mkdir.o                 \
895         mknod.o                 \
896         msgsys.o                \
897         nfssys.o                \
898         open.o                  \
899         pgrpsys.o               \
900         posix_sigwait.o         \
901         ppriv.o                 \
902         psetsys.o               \
903         rctlsys.o               \
904         readlink.o              \
905         rename.o                \
906         sbrk.o                  \
907         semsys.o                \
908         set_errno.o             \
909         sharefs.o               \
910         shmsys.o                \
911         sidsys.o                \
912         siginterrupt.o          \
913         signal.o                \
914         sigpending.o            \
915         sigstack.o              \
916         stat.o                  \
917         symlink.o               \
918         tasksys.o               \
```

```
 919         time.o                  \
 920         time_util.o             \
 921         ucontext.o              \
 922         unlink.o                \
 923         ustat.o                 \
 924         utimesys.o              \
 925         zone.o

 927 PORTREGEX=                       \
 928         glob.o                  \
 929         regcmp.o                \
 930         regex.o                 \
 931         wordexp.o

 933 VALUES= values-Xa.o

 935 MOSTOBJS=                        \
 936         $(STRETS)               \
 937         $(CRTOBJS)              \
 938         $(DYNOBJS)              \
 939         $(FPOBJS)               \
 940         $(FPOBJS64)             \
 941         $(FPASMOBJS)            \
 942         $(ATOMICOBJS)           \
 943         $(XATTROBJS)            \
 944         $(COMOBJS)              \
 945         $(GENOBJS)              \
 946         $(PRFOBJS)              \
 947         $(PORTFP)               \
 948         $(PORTGEN)              \
 949         $(PORTGEN64)            \
 950         $(PORTI18N)             \
 951         $(PORTI18N_COND)        \
 952         $(PORTLOCALE)           \
 953         $(PORTPRINT)            \
 954         $(PORTPRINT_W)          \
 955         $(PORTREGEX)            \
 956         $(PORTSTDIO)            \
 957         $(PORTSTDIO64)          \
 958         $(PORTSTDIO_W)          \
 959         $(PORTSYS)              \
 960         $(PORTSYS64)            \
 961         $(AIOOBJS)              \
 962         $(RTOBJS)               \
 963         $(TPOOLOBJS)            \
 964         $(THREADSOBJS)          \
 965         $(THREADSMACHOBJS)      \
 966         $(THREADSASMOBJS)       \
 967         $(UNICODEOBJS)          \
 968         $(UNWINDMACHOBJS)       \
 969         $(UNWINDASMOBJS)        \
 970         $(COMSYSOBJS)           \
 971         $(SYSOBJS)              \
 972         $(COMSYSOBJS64)         \
 973         $(SYSOBJS64)            \
 974         $(VALUES)

 976 TRACEOBJS=                      \
 977         plockstat.o

 979 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
 980 # modules whose source is provided in the $(SRC)/lib/common directory.
 981 # This must be done because otherwise the Sun C compiler would insert
 982 # its own versions of these modules and those versions contain code
 983 # to call out to C++ initialization functions.  Such C++ initialization
 984 # functions can call back into libc before thread initialization is
```



```
 985 # complete and this leads to segmentation violations and other problems.
 986 # Since libc contains no C++ code, linking with the minimal crti.o and
 987 # crtn.o modules is safe and avoids the problems described above.
 988 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
 989 CRTSRCS= ../../common/sparcv9

 991 # include common library definitions
 992 include $(SRC)/lib/Makefile.lib
 993 include $(SRC)/lib/Makefile.lib.64

 995 # we need to override the default SONAME here because we might
 996 # be building a variant object (still libc.so.1, but different filename)
 997 SONAME = libc.so.1

 999 CFLAGS64 += $(CCVERBOSE)

1001 # This is necessary to avoid problems with calling _ex_unwind().
1002 # We probably don't want any inlining anyway.
1003 CFLAGS64 += -xinline=

1005 CERRWARN += -_gcc=-Wno-parentheses
1006 CERRWARN += -_gcc=-Wno-switch
1007 CERRWARN += -_gcc=-Wno-uninitialized
1008 CERRWARN += -_gcc=-Wno-unused-value
1009 CERRWARN += -_gcc=-Wno-unused-label
1010 CERRWARN += -_gcc=-Wno-unused-variable
1011 CERRWARN += -_gcc=-Wno-type-limits
1012 CERRWARN += -_gcc=-Wno-char-subscripts
1013 CERRWARN += -_gcc=-Wno-clobbered
1014 CERRWARN += -_gcc=-Wno-unused-function
1015 CERRWARN += -_gcc=-Wno-address

1017 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1018 # enables ASSERT() checking in the threads portion of the library.
1019 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1020 THREAD_DEBUG =
1021 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1023 # Make string literals read-only to save memory.
1024 CFLAGS64 += $(XSTRCONST)

1026 ALTPICS= $(TRACEOBJS:%=pics/%)

1028 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1030 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1032 sparcv9_C_PICFLAGS= -K PIC
1033 CFLAGS64 +=     $(EXTN_CFLAGS)
1034 CPPFLAGS=       -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1035                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1036 ASFLAGS=        $(EXTN_ASFLAGS) -K PIC -P -D__STDC__ -D_ASM -D__sparcv9 $(CPPFLA
1037                 $(sparcv9_AS_XARCH)

1039 # As a favor to the dtrace syscall provider, libc still calls the
1040 # old syscall traps that have been obsoleted by the *at() interfaces.
1041 # Delete this to compile libc using only the new *at() system call traps
1042 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1044 # Inform the run-time linker about libc specialized initialization
1045 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1046 DYNFLAGS +=     $(RTLDINFO)

1048 # Force libc's internal references to be resolved immediately upon loading
1049 # in order to avoid critical region problems.  Since almost all libc symbols
1050 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
```

```
1051 DYNFLAGS +=     -znow

1053 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1055 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1057 # Override this top level flag so the compiler builds in its native
1058 # C99 mode.  This has been enabled to support the complex arithmetic
1059 # added to libc.
1060 C99MODE=        $(C99_ENABLE)

1062 # libc method of building an archive
1063 # The "$(GREP) -v ' L '" part is necessary only until
1064 # lorder is fixed to ignore thread-local variables.
1065 BUILD.AR= $(RM) $@ ; \
1066         $(AR) q $@ `$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1068 # extra files for the clean target
1069 CLEANFILES=                     \
1070         $(LIBCDIR)/port/gen/errlst.c    \
1071         $(LIBCDIR)/port/gen/new_list.c  \
1072         assym.h                 \
1073         genassym                \
1074         pics/crti.o             \
1075         pics/crtn.o             \
1076         $(ALTPICS)

1078 CLOBBERFILES += $(LIB_PIC)

1080 # list of C source for lint
1081 SRCS=                                                   \
1082         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1083         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1084         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1085         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1086         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1087         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1088         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1089         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1090         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1091         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1092         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1093         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1094         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1095         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1096         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1097         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1098         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1099         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1100         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1101         $(FPOBJS64:%.o=$(LIBCBASE)/fp/%.c)              \
1102         $(LIBCBASE)/crt/_ftou.c                         \
1103         $(LIBCBASE)/gen/_xregs_clrptr.c                 \
1104         $(LIBCBASE)/gen/byteorder.c                     \
1105         $(LIBCBASE)/gen/ecvt.c                          \
1106         $(LIBCBASE)/gen/getctxt.c                       \
1107         $(LIBCBASE)/gen/makectxt.c                      \
1108         $(LIBCBASE)/gen/siginfolst.c                    \
1109         $(LIBCBASE)/gen/siglongjmp.c                    \
1110         $(LIBCBASE)/gen/swapctxt.c

1112 # conditional assignments
1113 $(DYNLIB) := CRTI = crti.o
1114 $(DYNLIB) := CRTN = crtn.o

1116 # Files which need the threads .il inline template
```



```
1117 TIL=                            \
1118         aio.o                   \
1119         alloc.o                 \
1120         assfail.o               \
1121         atexit.o                \
1122         atfork.o                \
1123         cancel.o                \
1124         door_calls.o            \
1125         err.o                   \
1126         errno.o                 \
1127         getctxt.o               \
1128         lwp.o                   \
1129         ma.o                    \
1130         machdep.o               \
1131         posix_aio.o             \
1132         pthr_attr.o             \
1133         pthr_barrier.o          \
1134         pthr_cond.o             \
1135         pthr_mutex.o            \
1136         pthr_rwlock.o           \
1137         pthread.o               \
1138         rand.o                  \
1139         rwlock.o                \
1140         scalls.o                \
1141         sched.o                 \
1142         sema.o                  \
1143         sigaction.o             \
1144         sigev_thread.o          \
1145         spawn.o                 \
1146         stack.o                 \
1147         swapctxt.o              \
1148         synch.o                 \
1149         tdb_agent.o             \
1150         thr.o                   \
1151         thread_interface.o      \
1152         thread_pool.o           \
1153         tls.o                   \
1154         tsd.o                   \
1155         unwind.o

1157 $(TIL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/threads/sparcv9.il

1159 # Files in fp, port/fp subdirectories that need base.il inline template
1160 IL=                             \
1161         __flt_decim.o           \
1162         decimal_bin.o

1164 $(IL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/fp/base.il

1166 # Files in fp subdirectory which need __quad.il inline template
1167 QIL=                            \
1168         _Q_add.o                \
1169         _Q_cmp.o                \
1170         _Q_cmpe.o               \
1171         _Q_div.o                \
1172         _Q_dtoq.o               \
1173         _Q_fcc.o                \
1174         _Q_mul.o                \
1175         _Q_qtod.o               \
1176         _Q_qtoi.o               \
1177         _Q_qtos.o               \
1178         _Q_qtou.o               \
1179         _Q_sqrt.o               \
1180         _Q_stoq.o               \
1181         _Q_sub.o                \
1182         _Qp_qtox.o              \
```

```
1183         _Qp_qtoux.o

1185 $(QIL:%=pics/%) := CFLAGS64 += $(LIBCDIR)/$(MACH)/fp/__quad.il
1186 pics/_Qp%.o := CFLAGS64 += -I$(LIBCDIR)/$(MACH)/fp
1187 pics/_Q%.o := sparcv9_COPTFLAG = -xO4 -xchip=ultra

1189 # Files in crt subdirectory which need muldiv64.il inline template
1190 #CIL=   mul64.o divrem64.o
1191 #$(CIL:%=pics/%) := CFLAGS += $(LIBCBASE)/crt/mul64.il

1193 # large-file-aware components that should be built large

1195 #$(COMSYSOBJS64:%=pics/%) := \
1196 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1198 #$(SYSOBJS64:%=pics/%) := \
1199 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1201 #$(PORTGEN64:%=pics/%) := \
1202 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1204 #$(PORTSTDIO64:%=pics/%) := \
1205 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1207 #$(PORTSYS64:%=pics/%) := \
1208 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1210 $(PORTSTDIO_W:%=pics/%) := \
1211         CPPFLAGS += -D_WIDE

1213 $(PORTPRINT_W:%=pics/%) := \
1214         CPPFLAGS += -D_WIDE

1216 $(PORTI18N_COND:%=pics/%) := \
1217         CPPFLAGS += -D_WCS_LONGLONG

1219 # Files which need extra optimization
1220 pics/getenv.o := sparcv9_COPTFLAG = -xO4

1222 .KEEP_STATE:

1224 all: $(LIBS) $(LIB_PIC)

1226 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1227 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1228 lint    :=      LINTFLAGS64 += -mn

1230 lint:
1231         @echo $(LINT.c) ... $(LDLIBS)
1232         @$(LINT.c) $(SRCS) $(LDLIBS)

1234 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1235 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1236 $(LINTLIB):= LINTFLAGS64=-nvx -m64

1238 # object files that depend on inline template
1239 $(TIL:%=pics/%): $(LIBCBASE)/threads/sparcv9.il
1240 $(IL:%=pics/%): $(LIBCBASE)/fp/base.il
1241 $(QIL:%=pics/%): $(LIBCDIR)/$(MACH)/fp/__quad.il
1242 #$(CIL:%=pics/%): $(LIBCBASE)/crt/muldiv64.il

1244 # include common libc targets
1245 include $(LIBCDIR)/Makefile.targ

1247 # We need to strip out all CTF and DOF data from the static library
1248 $(LIB_PIC) := DIR = pics
```



```
1249 $(LIB_PIC): pics $$(PICS)
1250         $(BUILD.AR)
1251         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1252         $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1253         $(AR) -ts $@ > /dev/null
1254         $(POST_PROCESS_A)

1256 # special cases
1257 #$(STRETS:%=pics/%): crt/stret.s
1258 #       $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=%) crt/stret.s -o $@
1259 #       $(POST_PROCESS_O)

1261 #crt/_rtbootld.s:       crt/_rtboot.s crt/_rtld.c
1262 #       $(CC) $(CPPFLAGS) -O -S -K pic crt/_rtld.c -o crt/_rtld.s
1263 #       $(CAT) crt/_rtboot.s crt/_rtld.s > $@
1264 #       $(RM) crt/_rtld.s

1266 ASSYMDEP_OBJS=                  \
1267         _lwp_mutex_unlock.o     \
1268         _stack_grow.o           \
1269         asm_subr.o              \
1270         setjmp.o                \
1271         smt_pause.o             \
1272         tls_get_addr.o          \
1273         unwind_frame.o          \
1274         vforkx.o

1276 $(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

1278 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1280 # assym.h build rules

1282 assym.h := CFLAGS64 += -g

1284 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1286 genassym: $(GENASSYM_C)
1287         $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1288                 $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1290 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1292 assym.h: $(OFFSETS) genassym
1293         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1294         ./genassym >>$@

1296 # derived C source and related explicit dependencies
1297 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1298         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1300 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   18272 Mon Apr  6 11:47:28 2015
new/usr/src/lib/libxcurses/h/mks.h
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License, Version 1.0 only
   6  * (the "License").  You may not use this file except in compliance
   7  * with the License.
   8  *
   9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  10  * or http://www.opensolaris.org/os/licensing.
  11  * See the License for the specific language governing permissions
  12  * and limitations under the License.
  13  *
  14  * When distributing Covered Code, include this CDDL HEADER in each
  15  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  16  * If applicable, add the following below this CDDL HEADER, with the
  17  * fields enclosed by brackets "[]" replaced with your own identifying
  18  * information: Portions Copyright [yyyy] [name of copyright owner]
  19  *
  20  * CDDL HEADER END
  21  */
  22 /*
  23  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
  24  *
  25  * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
  26  * Use is subject to license terms.
  27  */

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

  29 /*
  30  * MKS header file.  Defines that make programming easier for us.
  31  * Includes MKS-specific things and posix routines.
  32  *
  33  * Copyright 1985, 1993 by Mortice Kern Systems Inc.  All rights reserved.
  34  *
  35  * $Header: /rd/h/rcs/mks.h 1.233 1995/09/28 19:45:19 mark Exp $
  36  */

  38 #ifndef __M_MKS_H__
  39 #define __M_MKS_H__

  41 /*
  42  * This should be a feature test macro defined in the Makefile or
  43  * cc command line.
  44  */
  45 #ifndef MKS
  46 #define MKS     1
  47 #endif

  49 /*
  50  * Write function declarations as follows:
  51  *      extern char     *function ANSI((char *cp, int flags, NODE *np));
  52  * Expansion of this happens only when __STDC__ is set.
  53  */
  54 #ifdef  __STDC__
  55 #define ANSI(x) x
  56 #define _VOID   void            /* Used in VOID *malloc() */
  57 #else
  58 #define const
  59 #define signed
```



```
  60 #define volatile
  61 #define ANSI(x) ()
  62 #define _VOID   char            /* Used in _VOID *malloc() */
  63 #endif

  65 #ifndef STATIC
  66 #  define       STATIC  static          /* Used for function definition */
  67 #endif  /*STATIC*/

  69 #ifndef STATREF
  70 #  ifdef        __STDC__
  71 #    define     STATREF static
  72 #  else
  73 #    define     STATREF         /* Used in local function forward declaration */
  74 #  endif
  75 #endif  /*STATREF*/

  77 #define LEXTERN extern          /* Library external reference */
  78 #define LDEFN                   /* Define Loadable library entry */

  80 typedef void    (*_sigfun_t)(int);

  82 #include <mkslocal.h>
  83 #include <stdio.h>
  84 #include <unistd.h>
  85 #include <limits.h>
  86 #include <sys/stat.h>   /* required for m_samefile() prototype. */
  87 #include <m_wchar.h>
  88 #include <m_i18n.h>
  89 #include <m_invari.h>

  91 #if M_TFGETC || M_STTY_CC
  92 #include <termios.h>
  93 #endif

  95 #ifndef M_LIBDIR
  96  # error "You must define M_LIBDIR in mkslocal.h"
  97 #endif

  99 #ifndef M_ETCDIR
 100  # error "You must define M_ETCDIR in mkslocal.h"
 101 #endif

 103 #ifndef M_SPOOLDIR
 104  # error "You must define M_SPOOLDIR in mkslocal.h"
 105 #endif

 107 #ifndef M_MANPATH
 108  # error "You must define M_MANPATH in mkslocal.h"
 109 #endif

 111 #if defined(I18N) && !defined(M_NLSDIR)
 112  # error "You must define M_NLSDIR in mkslocal.h"
 113 #endif

 115 #if (defined(M_I18N_MKS_FULL) || defined(M_I18N_MKS_XPG)) && !defined(I18N)
 116  # error I18N must be defined
 117 #endif

 119 /* P_tmpdir - used by tmpnam.c and tempnam.c.
 120  * Could be in <stdio.h>. But in case it is not ..
 121  */
 122 #ifndef P_tmpdir
 123 #  ifndef M_TMPDIR
 124     # error  M_TMPDIR must be defined in mkslocal.h
 125 #  endif
```

```
126 #  define P_tmpdir        M_TMPDIR
127 #endif /* P_tmpdir */

129 /* L_cuserid - used by cuserid.c
130  * Could be in <stdio.h>. But in case it is not ..
131  */
132 #ifndef L_cuserid
133 #  ifndef M_L_CUSERID
134       # error  M_L_CUSERID must be defined in mkslocal.h
135 #  endif
136 #  define L_cuserid        M_L_CUSERID
137 #endif /* L_cuserid */

139 #ifdef  M_AUDIT
140 LEXTERN char    *m_audmode (int, int);
141 #if !defined(M_AUDITW1) || !defined(M_AUDITW2)
142  # error "With M_AUDIT set, you must define M_AUDITW1 and M_AUDITW2"
143 #endif
144 #endif  /*M_AUDIT*/

146 #ifndef M_CS_PATH
147  # error "You must define M_CS_PATH in mkslocal.h"
148 #endif

150 #ifndef M_CS_SHELL
151  # error "You must define M_CS_SHELL in mkslocal.h"
152 #endif

154 #ifndef M_SH_USER_FDS
155 /*
156  * default number of user file descriptors to be used in the shell
157  * Must be >= 10, should be <= OPEN_MAX/2.
158  */
159 #define M_SH_USER_FDS   10
160 #endif /*M_SH_USER_FDS*/

162 #ifndef M_SH_MAX_FUNCTION_EVAL_DEPTH
163 #define M_SH_MAX_FUNCTION_EVAL_DEPTH    100
164 #endif

166 #ifndef M_MANPAGER
167 #define M_MANPAGER      "more -A -s"
168 #endif

170 /* set up alert and verticalTab characters - This assumes an ANSI-C compiler */
171 #undef  M_ALERT
172 #undef  M_VTAB
173 #define M_ALERT '\a'
174 #define M_VTAB  '\v'

176 #ifndef M_ESCAPE
177 #  define       M_ESCAPE '\033'         /* default to ASCII code for <ESC> */
178 #endif /*M_ESCAPE*/

180 #ifndef SETVBUF
181 /* if SETVBUF not previously defined, then use default ANSI-C definition */
182 #  define       SETVBUF setvbuf
183 #endif

185 #ifdef M_NULL
186 /* if M_NULL defined in <mkslocal.h> then want to redefine NULL */
187 #undef        NULL
188 #define       NULL    (M_NULL)
189 #endif /*M_NULL*/

191 /*
```



```
192  * Useful additions to sys/stat.h.
193  */
194 #ifndef S_IRALL
195 #define S_IRALL         (S_IRUSR|S_IRGRP|S_IROTH)
196 #endif
197 #ifndef S_IWALL
198 #define S_IWALL         (S_IWUSR|S_IWGRP|S_IWOTH)
199 #endif
200 #ifndef S_IXALL
201 #define S_IXALL         (S_IXUSR|S_IXGRP|S_IXOTH)
202 #endif

204 #ifndef M_DEFMODE       /* Default directory creation mode */
205 #define M_DEFMODE       ((mode_t)S_IRALL|S_IWALL)/* Default file creation mode*/
206 #endif
207 #ifndef M_DIRMODE
208 #define M_DIRMODE       ((mode_t)S_IRALL|S_IWALL|S_IXALL)
209 #endif

211 #ifndef M_FLDSEP
212 #define M_FLDSEP        ':'     /* UNIX field separator for passwd, PATH */
213 #endif

215 #ifndef M_TTYNAME
216 #define M_TTYNAME "/dev/tty"
217 #endif

219 #ifndef M_NULLNAME
220 #define M_NULLNAME "/dev/null"
221 #endif

223 #ifndef M_FSDELIM
224 #define M_FSDELIM(c)    ((c)=='/')
225 #endif

227 #ifndef M_DRDELIM
228 #define M_DRDELIM(c)    (0)
229 #endif

231 #ifndef M_DIRSTAT
232 #define M_DIRSTAT(name, dp, sb) stat((name), (sb))
233 #endif

235 #ifndef M_HIDDEN
236 #define M_HIDDEN(dirp, dp)      ((dp)->d_name[0] == '.')
237 #endif

239 #ifndef M_FSMOUNT               /* for use by setmnt routine */
240 #define M_FSMOUNT M_ETCDIR(mtab)
241 #endif

243 #ifndef M_FSALL                 /* for use by setmnt routine */
244 #define M_FSALL M_ETCDIR(fstab)
245 #endif

247 #ifndef M_NLSCHARMAP            /* Default charmap file for localedef */
248 #define M_NLSCHARMAP    M_NLSDIR(charmap/ISO_8859-1)
249 #endif

251 #ifndef M_POSIXPATH     /* used when I18N undefined, default posix path */
252 #define M_POSIXPATH     M_NLSDIR(locale/POSIX)
253 #endif

255 #ifndef M_ISEOV
256 #define M_ISEOV(error)  0
257 #endif
```

```
#ifndef M_IS_NATIVE_LOCALE
#define M_IS_NATIVE_LOCALE(s)   (strcmp(s, "POSIX")==0 || strcmp(s, "C")==0)
#endif

#ifndef M_FSCLOSE
#define M_FSCLOSE(fp)
#endif

#ifndef ROOTUID         /* default superuser uid = 0 */
#define ROOTUID 0
#endif

#ifndef ROOTGID         /* default superuser gid = 0 */
#define ROOTGID 0
#endif

#ifndef M_GROUP_PASSWD
#define M_GROUP_PASSWD(grp)  (grp->gr_passwd)
#endif


#ifndef M_NUMSIZE
/*
 * define the expected max length of a printed number. (used in awk)
 * This should be the longest expected size for any type of number
 * ie. float, long etc.
 * This number is used to calculate the approximate
 * number of bytes needed to hold the number.
 */
#define M_NUMSIZE       30
#endif /* M_NUMSIZE */

/*
 * VARARG[12345]: declare variadic functions.
 * Expands to either a standard C prototype or a K&R declaration.
 * For example:
 *
 * #include <stdarg.h>
 * int
 * fprintf VARARG2(FILE*, fp, char*, fmt)
 * {
 *      va_list ap;
 *
 *      va_start(ap, fmt);
 *      cp = va_arg(ap, char*);
 *      va_end(ap);
 * }
 */
#ifndef VARARG1
#ifdef  __STDC__
#define VARARG1(type, name) (type name, ...)
#define VARARG2(t1, n1, t2, n2) (t1 n1, t2 n2, ...)
#define VARARG3(t1, n1, t2, n2, t3, n3) (t1 n1, t2 n2, t3 n3, ...)
#define VARARG4(t1, n1, t2, n2, t3, n3, t4, n4) \
                (t1 n1, t2 n2, t3 n3, t4 n4, ...)
#define VARARG5(t1, n1, t2, n2, t3, n3, t4, n4, t5, n5) \
                (t1 n1, t2 n2, t3 n3, t4 n4, t5 n5, ...)
#else
#define VARARG1(type, name) (name, va_alist) type name; int va_alist
#define VARARG2(t1, n1, t2, n2) (n1, n2, va_alist) t1 n1; t2 n2; int va_alist
#define VARARG3(t1, n1, t2, n2, t3, n3) (n1, n2, n3, va_alist) \
                t1 n1; t2 n2; t3 n3; int va_alist
#define VARARG4(t1, n1, t2, n2, t3, n3, t4, n4) (n1, n2, n3, n4, va_alist) \
                t1 n1; t2 n2; t3 n3; t4 n4; int va_alist
#define VARARG5(t1, n1, t2, n2, t3, n3, t4, n4, t5, n5) \
```



```
                (n1, n2, n3, n4, n5, va_alist) \
                t1 n1; t2 n2; t3 n3; t4 n4; t5 n5; int va_alist
#endif
#endif


/*
 * MKS-specific library entry points.
 */
extern  char    *_cmdname;
LEXTERN char    *basename (char *);
LEXTERN void    crc16 (ushort *, ushort);
LEXTERN void    crccitt (ushort *, ushort);
LEXTERN int     eprintf (const char *, ...);
LEXTERN void    eputs (const char *);
LEXTERN pid_t   fexecve (const char *, char *const *, char *const *);
LEXTERN pid_t   fexecvp (const char *, char *const *);
LEXTERN pid_t   fexecvep (const char *, char *const *, char *const *);
LEXTERN int     execvep (const char *, char *const *, char *const *);
LEXTERN int     isabsname (const char *);
LEXTERN const char *m_cescape (wint_t);
LEXTERN int     m_escapec (char **);
LEXTERN const char *m_toprint (wint_t);
#if M_STTY_CC
LEXTERN int     m_stty_cc (cc_t* cp, char *str);
#endif
LEXTERN char    *m_cmdname (char *);
LEXTERN char    *m_strmode (mode_t);
LEXTERN char    *m_readmode (const char *);
LEXTERN char    *m_readnum (long *, char *, char **, int);
LEXTERN char    *m_readunum (unsigned long *, char *, char **, int);
LEXTERN mode_t  m_getmode (mode_t);
LEXTERN int     m_wallow (int, const char *);
LEXTERN char    *m_pathcat (const char *, const char *);
LEXTERN void    m_sigcleanup (void (*__handler)(int __signo) );
LEXTERN void    m_defaction (int __signo);
LEXTERN char    *m_strdup (const char *s);
LEXTERN int     m_stricmp (const char *, const char *);
LEXTERN char    *m_self (int, char *, char *);
LEXTERN int     m_grouplist (char *user, gid_t *gidlist[]);
LEXTERN int     m_setgroups (int gidsetsize, gid_t grouplist[]);
LEXTERN uint    m_binsrch (uint n, int (*cmp)(uint i));
LEXTERN char    *m_dirname (const char*);
LEXTERN char    *m_confstr (int);

LEXTERN void    m_crcposix (ulong *, const uchar *, size_t);
LEXTERN int     m_setprio (int, unsigned int, int);
LEXTERN int     m_getprio (int, unsigned int);
LEXTERN int     m_incrnice (int, unsigned int, int);
LEXTERN char    *m_devname (dev_t);
LEXTERN char    *m_mountdir (const char *);
LEXTERN int     m_absname(char *, char *, char *, size_t);
LEXTERN int     m_samefile(char *, struct stat *, char *, struct stat *);

/* __m_system() : alternate interface into system() */
LEXTERN int     __m_system (const char *, const char *, const char *);


/* conversion routines - between single byte and UNICODE (wide) strings.
 * These return a pointer to malloc'd memory.
 * It is the caller's responsiblity to free() it, if necessary
 * These are for use primarily on NT
 */
extern char *m_unicodetosb(const wchar_t*);
extern wchar_t *m_sbtounicode(const char*);
```

```
388 /*
389  * things that could go into an "m_stdio.h"
390  */

392 /* m_unlink() : alternate unlink() for use with vendor-provided
393  * libraries that do not have a satisfactory unlink() */
394 #ifndef M_UNLINK
395 #define m_unlink(s)     unlink(s)
396 #endif

398 /* __m_popen() : alternate interface into popen() */
399 LEXTERN FILE    *__m_popen (const char *, const char *,
400                             const char *, const char *);
401 LEXTERN FILE    *__m_popenvp (const char *mode, const char *shell,
402                               char const * const *args);

404 #if M_TFGETC
405 LEXTERN int     m_tfgetc (FILE *fp, struct termios *tp);
406 #else
407 #define         m_tfgetc(fp,tp) fgetc(fp)
408 #endif

410 /* m_fsopen() - special routine for curses */
411 LEXTERN FILE    *m_fsopen (char *, size_t, const char *, FILE *);

413 #ifndef M_FFLUSH_NOT_POSIX_1
414 # define        m_fflush fflush
415 #else
416   LEXTERN       int     m_fflush (FILE *);
417 #endif

419 /* m_fgets return values */
420 enum {
421         M_FGETS_OK,     /* Normal return */
422         M_FGETS_EOF,    /*
423                          * Regular EOF (same as NULL from fgets).
424                          * Buffer is *untouched*.
425                          */
426         M_FGETS_SHORT,  /*
427                          * Short input (buf[strlen(buf)-1] != '\n')
428                          * This is a trailing line, without a newline at the
429                          * end of the file.  The buffer is valid, ending in
430                          * a \0, with no newline.  The case of terminal input
431                          * ending with an EOF in the middle of the line will
432                          * restart -- typing two EOF's will result in this
433                          * case.
434                          */
435         M_FGETS_LONG,   /*
436                          * Line too long: newline not found within len bytes
437                          * (buf[len-1] != '\n').
438                          * At this point, while((c=getc(fp)) != '\n') ...
439                          * is a valid method to get the rest of the line.
440                          */
441         M_FGETS_BINARY, /*
442                          * Input contained an invalid character (e.g. \0)
443                          * Buffer contents *undefined*.
444                          */
445         M_FGETS_ERROR   /*
446                          * A system call returned an error, errno is set.
447                          * Buffer contents *undefined*.
448                          */
449 };
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    8211 Mon Apr  6 11:47:31 2015
new/usr/src/man/man2/Makefile
5798 fexecve() needed per POSIX 2008
**********************************************************
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2013, OmniTI Computer Consulting, Inc
  15 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  16 # Copyright (c) 2015, Joyent, Inc.  All rights reserved.
  17 #

  19 include         $(SRC)/Makefile.master

  21 MANSECT=        2

  23 MANFILES=       Intro.2                 \
  24                 __sparc_utrap_install.2 \
  25                 _lwp_cond_signal.2      \
  26                 _lwp_cond_wait.2        \
  27                 _lwp_info.2             \
  28                 _lwp_kill.2             \
  29                 _lwp_mutex_lock.2       \
  30                 _lwp_self.2             \
  31                 _lwp_sema_wait.2        \
  32                 _lwp_suspend.2          \
  33                 access.2                \
  34                 acct.2                  \
  35                 acl.2                   \
  36                 adjtime.2               \
  37                 alarm.2                 \
  38                 audit.2                 \
  39                 auditon.2               \
  40                 brk.2                   \
  41                 chdir.2                 \
  42                 chmod.2                 \
  43                 chown.2                 \
  44                 chroot.2                \
  45                 close.2                 \
  46                 creat.2                 \
  47                 dup.2                   \
  48                 exec.2                  \
  49                 exit.2                  \
  50                 fcntl.2                 \
  51                 fork.2                  \
  52                 fpathconf.2             \
  53                 getacct.2               \
  54                 getaudit.2              \
  55                 getauid.2               \
  56                 getcontext.2            \
  57                 getdents.2              \
  58                 getgroups.2             \
  59                 getisax.2               \
  60                 getitimer.2             \
  61                 getlabel.2              \
```



```
  62                 getmsg.2                \
  63                 getpflags.2             \
  64                 getpid.2                \
  65                 getppriv.2              \
  66                 getrlimit.2             \
  67                 getsid.2                \
  68                 getuid.2                \
  69                 getustack.2             \
  70                 ioctl.2                 \
  71                 issetugid.2             \
  72                 kill.2                  \
  73                 link.2                  \
  74                 llseek.2                \
  75                 lseek.2                 \
  76                 memcntl.2               \
  77                 meminfo.2               \
  78                 mincore.2               \
  79                 mkdir.2                 \
  80                 mknod.2                 \
  81                 mmap.2                  \
  82                 mmapobj.2               \
  83                 mount.2                 \
  84                 mprotect.2              \
  85                 msgctl.2                \
  86                 msgget.2                \
  87                 msgids.2                \
  88                 msgrcv.2                \
  89                 msgsnap.2               \
  90                 msgsnd.2                \
  91                 munmap.2                \
  92                 nice.2                  \
  93                 ntp_adjtime.2           \
  94                 ntp_gettime.2           \
  95                 open.2                  \
  96                 p_online.2              \
  97                 pause.2                 \
  98                 pcsample.2              \
  99                 pipe.2                  \
 100                 poll.2                  \
 101                 priocntl.2              \
 102                 priocntlset.2           \
 103                 processor_bind.2        \
 104                 processor_info.2        \
 105                 profil.2                \
 106                 pset_bind.2             \
 107                 pset_create.2           \
 108                 pset_info.2             \
 109                 pset_list.2             \
 110                 pset_setattr.2          \
 111                 putmsg.2                \
 112                 read.2                  \
 113                 readlink.2              \
 114                 rename.2                \
 115                 resolvepath.2           \
 116                 rmdir.2                 \
 117                 semctl.2                \
 118                 semget.2                \
 119                 semids.2                \
 120                 semop.2                 \
 121                 setpgid.2               \
 122                 setpgrp.2               \
 123                 setrctl.2               \
 124                 setregid.2              \
 125                 setreuid.2              \
 126                 setsid.2                \
 127                 settaskid.2             \
```

```
128                 setuid.2                \
129                 shmctl.2                \
130                 shmget.2                \
131                 shmids.2                \
132                 shmop.2                 \
133                 sigaction.2             \
134                 sigaltstack.2           \
135                 sigpending.2            \
136                 sigprocmask.2           \
137                 sigsend.2               \
138                 sigsuspend.2            \
139                 sigwait.2               \
140                 stat.2                  \
141                 statvfs.2               \
142                 stime.2                 \
143                 swapctl.2               \
144                 symlink.2               \
145                 sync.2                  \
146                 sysfs.2                 \
147                 sysinfo.2               \
148                 time.2                  \
149                 times.2                 \
150                 uadmin.2                \
151                 ulimit.2                \
152                 umask.2                 \
153                 umount.2                \
154                 uname.2                 \
155                 unlink.2                \
156                 ustat.2                 \
157                 utime.2                 \
158                 utimes.2                \
159                 uucopy.2                \
160                 vfork.2                 \
161                 vhangup.2               \
162                 waitid.2                \
163                 write.2                 \
164                 yield.2                 \

166 MANLINKS=       _Exit.2                         \
167                 _exit.2                         \
168                 _lwp_cond_broadcast.2           \
169                 _lwp_cond_reltimedwait.2        \
170                 _lwp_cond_timedwait.2           \
171                 _lwp_continue.2                 \
172                 _lwp_mutex_trylock.2            \
173                 _lwp_mutex_unlock.2             \
174                 _lwp_sema_init.2                \
175                 _lwp_sema_post.2                \
176                 _lwp_sema_trywait.2             \
177                 execl.2                         \
178                 execle.2                        \
179                 execlp.2                        \
180                 execv.2                         \
181                 execve.2                        \
182                 execvp.2                        \
183                 faccessat.2                     \
184                 facl.2                          \
185                 fchdir.2                        \
186                 fchmod.2                        \
187                 fchmodat.2                      \
188                 fchown.2                        \
189                 fchownat.2                      \
190                 fchroot.2                       \
191                 fexecve.2                       \
192                 fgetlabel.2                     \
193                 fork1.2                         \
```



```
194                 forkall.2                       \
195                 forkallx.2                      \
196                 forkx.2                         \
197                 fstat.2                         \
198                 fstatat.2                       \
199                 fstatvfs.2                      \
200                 futimesat.2                     \
201                 futimens.2                      \
202                 getaudit_addr.2                 \
203                 getegid.2                       \
204                 geteuid.2                       \
205                 getgid.2                        \
206                 getpgid.2                       \
207                 getpgrp.2                       \
208                 getpmsg.2                       \
209                 getppid.2                       \
210                 getprojid.2                     \
211                 getrctl.2                       \
212                 gettaskid.2                     \
213                 intro.2                         \
214                 lchown.2                        \
215                 linkat.2                        \
216                 lstat.2                         \
217                 mkdirat.2                       \
218                 mknodat.2                       \
219                 openat.2                        \
220                 pathconf.2                      \
221                 pipe2.2                         \
222                 ppoll.2                         \
223                 pread.2                         \
224                 preadv.2                        \
225                 pset_assign.2                   \
226                 pset_destroy.2                  \
227                 pset_getattr.2                  \
228                 putacct.2                       \
229                 putpmsg.2                       \
230                 pwrite.2                        \
231                 pwritev.2                       \
232                 readlinkat.2                    \
233                 readv.2                         \
234                 renameat.2                      \
235                 sbrk.2                          \
236                 semtimedop.2                    \
237                 setaudit.2                      \
238                 setaudit_addr.2                 \
239                 setauid.2                       \
240                 setcontext.2                    \
241                 setegid.2                       \
242                 seteuid.2                       \
243                 setgid.2                        \
244                 setgroups.2                     \
245                 setitimer.2                     \
246                 setpflags.2                     \
247                 setppriv.2                      \
248                 setrlimit.2                     \
249                 setustack.2                     \
250                 shmat.2                         \
251                 shmdt.2                         \
252                 sigsendset.2                    \
253                 symlinkat.2                     \
254                 umount2.2                       \
255                 unlinkat.2                      \
256                 utimensat.2                     \
257                 vforkx.2                        \
258                 wracct.2                        \
259                 writev.2
```

```
261 intro.2                         := LINKSRC = Intro.2

263 _lwp_cond_broadcast.2           := LINKSRC = _lwp_cond_signal.2

265 _lwp_cond_reltimedwait.2        := LINKSRC = _lwp_cond_wait.2
266 _lwp_cond_timedwait.2           := LINKSRC = _lwp_cond_wait.2

268 _lwp_mutex_trylock.2            := LINKSRC = _lwp_mutex_lock.2
269 _lwp_mutex_unlock.2             := LINKSRC = _lwp_mutex_lock.2

271 _lwp_sema_init.2               := LINKSRC = _lwp_sema_wait.2
272 _lwp_sema_post.2               := LINKSRC = _lwp_sema_wait.2
273 _lwp_sema_trywait.2            := LINKSRC = _lwp_sema_wait.2

275 _lwp_continue.2                := LINKSRC = _lwp_suspend.2

277 faccessat.2                    := LINKSRC = access.2

279 facl.2                         := LINKSRC = acl.2

281 sbrk.2                         := LINKSRC = brk.2

283 fchdir.2                       := LINKSRC = chdir.2

285 fchmod.2                       := LINKSRC = chmod.2
286 fchmodat.2                     := LINKSRC = chmod.2

288 fchown.2                       := LINKSRC = chown.2
289 fchownat.2                     := LINKSRC = chown.2
290 lchown.2                       := LINKSRC = chown.2

292 fchroot.2                      := LINKSRC = chroot.2

294 execl.2                        := LINKSRC = exec.2
295 execle.2                       := LINKSRC = exec.2
296 execlp.2                       := LINKSRC = exec.2
297 execv.2                        := LINKSRC = exec.2
298 execve.2                       := LINKSRC = exec.2
299 execvp.2                       := LINKSRC = exec.2
300 fexecve.2                      := LINKSRC = exec.2

302 _Exit.2                        := LINKSRC = exit.2
303 _exit.2                        := LINKSRC = exit.2

305 fork1.2                        := LINKSRC = fork.2
306 forkall.2                      := LINKSRC = fork.2
307 forkallx.2                     := LINKSRC = fork.2
308 forkx.2                        := LINKSRC = fork.2

310 pathconf.2                     := LINKSRC = fpathconf.2

312 putacct.2                      := LINKSRC = getacct.2
313 wracct.2                       := LINKSRC = getacct.2

315 getaudit_addr.2               := LINKSRC = getaudit.2
316 setaudit.2                     := LINKSRC = getaudit.2
317 setaudit_addr.2               := LINKSRC = getaudit.2

319 setauid.2                      := LINKSRC = getauid.2

321 setcontext.2                   := LINKSRC = getcontext.2

323 setgroups.2                    := LINKSRC = getgroups.2

325 setitimer.2                    := LINKSRC = getitimer.2
```



```
327 fgetlabel.2                    := LINKSRC = getlabel.2

329 getpmsg.2                      := LINKSRC = getmsg.2

331 setpflags.2                    := LINKSRC = getpflags.2

333 getpgid.2                      := LINKSRC = getpid.2
334 getpgrp.2                      := LINKSRC = getpid.2
335 getppid.2                      := LINKSRC = getpid.2

337 setppriv.2                     := LINKSRC = getppriv.2

339 setrlimit.2                    := LINKSRC = getrlimit.2

341 getegid.2                      := LINKSRC = getuid.2
342 geteuid.2                      := LINKSRC = getuid.2
343 getgid.2                       := LINKSRC = getuid.2

345 setustack.2                    := LINKSRC = getustack.2

347 linkat.2                       := LINKSRC = link.2

349 mkdirat.2                      := LINKSRC = mkdir.2

351 mknodat.2                      := LINKSRC = mknod.2

353 openat.2                       := LINKSRC = open.2

355 pipe2.2                        := LINKSRC = pipe.2

357 ppoll.2                        := LINKSRC = poll.2

359 pset_assign.2                  := LINKSRC = pset_create.2
360 pset_destroy.2                 := LINKSRC = pset_create.2

362 pset_getattr.2                 := LINKSRC = pset_setattr.2

364 putpmsg.2                      := LINKSRC = putmsg.2

366 pread.2                        := LINKSRC = read.2
367 readv.2                        := LINKSRC = read.2
368 preadv.2                       := LINKSRC = read.2

370 readlinkat.2                   := LINKSRC = readlink.2

372 renameat.2                     := LINKSRC = rename.2

374 semtimedop.2                   := LINKSRC = semop.2

376 getrctl.2                      := LINKSRC = setrctl.2

378 getprojid.2                    := LINKSRC = settaskid.2
379 gettaskid.2                    := LINKSRC = settaskid.2

381 setegid.2                      := LINKSRC = setuid.2
382 seteuid.2                      := LINKSRC = setuid.2
383 setgid.2                       := LINKSRC = setuid.2

385 shmat.2                        := LINKSRC = shmop.2
386 shmdt.2                        := LINKSRC = shmop.2

388 sigsendset.2                   := LINKSRC = sigsend.2

390 symlinkat.2                    := LINKSRC = symlink.2
```

```
 392 fstat.2                         := LINKSRC = stat.2
 393 fstatat.2                       := LINKSRC = stat.2
 394 lstat.2                         := LINKSRC = stat.2

 396 fstatvfs.2                      := LINKSRC = statvfs.2

 398 umount2.2                       := LINKSRC = umount.2

 400 unlinkat.2                      := LINKSRC = unlink.2

 402 futimens.2                      := LINKSRC = utimes.2
 403 futimesat.2                     := LINKSRC = utimes.2
 404 utimensat.2                     := LINKSRC = utimes.2

 406 vforkx.2                        := LINKSRC = vfork.2

 408 pwrite.2                        := LINKSRC = write.2
 409 writev.2                        := LINKSRC = write.2
 410 pwritev.2                       := LINKSRC = write.2

 412 .KEEP_STATE:

 414 include         $(SRC)/man/Makefile.man

 416 install:        $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
**********************************************************
   20495 Mon Apr  6 11:47:33 2015
new/usr/src/man/man2/exec.2
5798 fexecve() needed per POSIX 2008
**********************************************************
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
'\" te
.\" Copyright (c) 2008, 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
.Dd Sep 19, 2014
.Dt EXEC 2
.Os
.Sh NAME
.Nm exec ,
.Nm execl ,
.Nm execle ,
.Nm execlp ,
.Nm execv ,
.Nm execve ,
.Nm execvp ,
.Nm fexecve
.Nd execute a file
.Sh SYNOPSIS
.In unistd.h
.
.Ft int
.Fo execl
.Fa "const char *path"
.Fa "const char *arg0"
.Fa "... /* const char *argn, (char *)0 */"
.Fc
.
.Ft int
.Fo execv
.Fa "const char *path"
.Fa "char *const argv[]"
.Fc
.
.Ft int
.Fo execle
.Fa "const char *path"
.Fa "const char *arg0"
.Fa "... /* const char *argn, (char *)0, char *const envp[] */"
.Fc
.
.Ft int
.Fo execve
.Fa "const char *path"
.Fa "char *const argv[]"
.Fa "char *const envp[]"
.Fc
.
.Ft int
.Fo execlp
.Fa "const char *file"
.Fa "const char *arg0"
.Fa "... /* const char *argn, (char *)0 */"
.Fc
```



```
.
.Ft int
.Fo execvp
.Fa "const char *file"
.Fa "char *const argv[]"
.Fc
.
.Ft int
.Fo fexecve
.Fa "int fd"
.Fa "char *const argv[]"
.Fa "char *const envp[]"
.Fc
.Sh DESCRIPTION
Each of the functions in the
.Nm exec
family replaces the current process image with a new process image.
The new image is constructed from a regular, executable file called the
.Em new process image file .
This file is either an
.TH EXEC 2 "Jun 16, 2008"
.SH NAME
exec, execl, execle, execlp, execv, execve, execvp \- execute a file
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBint\fR \fBexecl\fR(\fBconst char *\fR\fIpath\fR, \fBconst char *\fR\fIarg0\fR
     /* const char *\fR\fIargn\fR, \fB(char *)0 */);\fR
.fi

.LP
.nf
\fBint\fR \fBexecv\fR(\fBconst char *\fR\fIpath\fR, \fBchar *const\fR \fIargv[]\
.fi

.LP
.nf
\fBint\fR \fBexecle\fR(\fBconst char *\fR\fIpath\fR, \fBconst char *\fR\fIarg0\f
     /* const char *\fR\fIargn\fR, \fB(char *)0\fR,\fBchar *const\fR \fIenvp\fR[
.fi

.LP
.nf
\fBint\fR \fBexecve\fR(\fBconst char *\fR\fIpath\fR, \fBchar *const\fR \fIargv[]
     \fBchar *const\fR \fIenvp[]\fR);
.fi

.LP
.nf
\fBint\fR \fBexeclp\fR(\fBconst char *\fR\fIfile\fR, \fBconst char *\fR\fIarg0\f
     /* const char *\fR\fIargn\fR, \fB(char *)0 */);\fR
.fi

.LP
.nf
\fBint\fR \fBexecvp\fR(\fBconst char *\fR\fIfile\fR, \fBchar *const\fR \fIargv[]
.fi

.SH DESCRIPTION
.sp
.LP
Each of the functions in the \fBexec\fR family replaces the current process
image with a new process image. The new image is constructed from a regular,
executable file called the \fBnew process image file\fR. This file is either an
```

```
.Lp
.sp
.LP
```
An interpreter file begins with a line of the form
```
.Lp
.Dl #! pathname Op Ar arg
.Lp
where
.Ar pathname
is the path of the interpreter, and
.Ar arg
is an optional argument.
When an interpreter file is executed, the system invokes the
specified interpreter.
The pathname specified in the interpreter file is passed as
.Fa arg0
to the interpreter.
If
.Ar arg
was specified in the interpreter file, it is passed as
.Fa arg1
to the interpreter.
The remaining arguments to
the interpreter are
.Fa arg0
through
.Fa argn
of the originally exec'd file.
The interpreter named by
.Ar pathname
must not be an interpreter file.
.Lp
.sp
.in +2
.nf
#! pathname [\fIarg\fR]
.fi
.in -2

.sp
.LP
where \fIpathname\fR is the path of the interpreter, and \fIarg\fR is an
optional argument. When an interpreter file is executed, the system invokes the
specified interpreter. The pathname specified in the interpreter file is passed
as \fIarg0\fR to the interpreter. If \fIarg\fR was specified in the interpreter
file, it is passed as \fIarg1\fR to the interpreter. The remaining arguments to
the interpreter are \fIarg0\fR through \fIargn\fR of the originally exec'd
file. The interpreter named by \fIpathname\fR must not be an interpreter file.
.sp
.LP
```
When a C-language program is executed as a result of this call, it is entered as a C-language function call as follows:
```
.Lp
.Dl Ft int Fn main "int argc" "char *argv[]"
.Lp
where
.Fa argc
is the argument count and
.Fa argv
is an array of character pointers to the arguments themselves.
In addition, the following variable:
.Lp
.Dl Vt "extern char **" Ns Va environ ;
```



```
.Lp
.sp
.in +2
.nf
int main (int argc, char *argv[]);
.fi
.in -2

.sp
.LP
where \fIargc\fR is the argument count and \fIargv\fR is an array of character
pointers to the arguments themselves. In addition, the following variable:
.sp
.in +2
.nf
extern char **environ;
.fi
.in -2

.sp
.LP
```
is initialized as a pointer to an array of character pointers to the
```
environment strings.
The
.Fa argv
and
.Va environ
arrays are each terminated by a null pointer.
The null pointer terminating the
.Fa argv
array is not counted in
.Fa argc .
.Lp
The value of
.Fa argc
is non-negative, and if greater than 0,
.Fa argv Ns [0]
points to a string containing the name of the file.
If
.Fa argc
is 0,
.Fa argv Ns [0]
is a null pointer, in which case there are no arguments.
Applications should verify that
.Fa argc
is greater than 0 or that
.Fa argv Ns [0]
is not a null pointer before dereferencing
.Fa argv Ns [0] .
.Lp
The arguments specified by a program with one of the
.Nm exec
functions are passed on to the new process image in the
Fn main
arguments.
.Lp
The
.Fa path
argument points to a path name that identifies the new process image file.
.Lp
The
.Fa file
argument is used to construct a pathname that identifies the new
process image file.
If the
.Fa file
```

```
argument contains a slash character, it is used as the pathname for this file.
Otherwise, the path prefix for this file is obtained by a search of the
directories passed in the
.Ev PATH
environment variable.
See
.Xr environ 5 .
The environment is supplied typically by the shell.
If the process image file is not a valid executable object file,
.Fn execlp
and
.Fn execvp
use the contents of that file as standard input to the shell.
In this case, the shell becomes the new process image.
The standard to which the caller conforms determines which shell is used.
See
.Xr standards 5 .
.Lp
The
.Fn fexecve
function is equivalent to
.Fn execve ,
except that instead of using a named file, the file referenced by
the file descriptor
.Fa fd
is used.  Note that this file descriptor must reference a regular
file and must have been opened for execute
.Po with
Dv O_EXEC ,
defined in
.In fcntl.h .
.Pc
The image is loaded from offset zero of the file, regardless of
the offset of
.Fa fd .
.Lp
The arguments represented by
.Fa arg0 Ns  "..."
are pointers to null-terminated character strings.
These strings constitute the argument list available to the new process image.
The list is terminated by a null pointer.
The
.Fa arg0
argument should point to a filename that is associated with the
process being started by one of the
.Nm exec
functions.
.Lp
The
.Fa argv
argument is an array of character pointers to null-terminated strings.
The last member of this array must be a null pointer.
These strings constitute the argument list available to the new process image.
The value in
.Fa argv Ns [0]
should point to a filename that is associated with the process
being started by one of the
.Nm exec
functions.
.Lp
The
.Fa envp
argument is an array of character pointers to null-terminated strings.
These strings constitute the environment for the new process image.
The
.Fa envp
```



```
array is terminated by a null pointer.
For
.Fn execl , execv , execvp ,
and
.Fn execlp ,
the C-language run-time
environment strings. The \fIargv\fR and \fIenviron\fR arrays are each
terminated by a null pointer. The null pointer terminating the \fIargv\fR array
is not counted in \fIargc\fR.
.sp
.LP
The value of \fIargc\fR is non-negative, and if greater than 0, \fIargv\fR[0]
points to a string containing the name of the file. If \fIargc\fR is 0,
\fIargv\fR[0] is a null pointer, in which case there are no arguments.
Applications should verify that \fIargc\fR is greater than 0 or that
\fIargv\fR[0] is not a null pointer before dereferencing \fIargv\fR[0].
.sp
.LP
The arguments specified by a program with one of the \fBexec\fR functions are
passed on to the new process image in the \fBmain()\fR arguments.
.sp
.LP
The \fIpath\fR argument points to a path name that identifies the new process
image file.
.sp
.LP
The \fIfile\fR argument is used to construct a pathname that identifies the new
process image file. If the \fIfile\fR argument contains a slash character, it
is used as the pathname for this file. Otherwise, the path prefix for this file
is obtained by a search of the directories passed in the \fBPATH\fR environment
variable (see \fBenviron\fR(5)). The environment is supplied typically by the
shell. If the process image file is not a valid executable object file,
\fBexeclp()\fR and \fBexecvp()\fR use the contents of that file as standard
input to the shell. In this case, the shell becomes the new process image. The
standard to which the caller conforms determines which shell is used. See
\fBstandards\fR(5).
.sp
.LP
The arguments represented by \fIarg0\fR\&.\|.\|. are pointers to
null-terminated character strings. These strings constitute the argument list
available to the new process image. The list is terminated by a null pointer.
The \fIarg0\fR argument should point to a filename that is associated with the
process being started by one of the \fBexec\fR functions.
.sp
.LP
The \fIargv\fR argument is an array of character pointers to null-terminated
strings. The last member of this array must be a null pointer. These strings
constitute the argument list available to the new process image. The value in
\fIargv\fR[0] should point to a filename that is associated with the process
being started by one of the \fBexec\fR functions.
.sp
.LP
The \fIenvp\fR argument is an array of character pointers to null-terminated
strings. These strings constitute the environment for the new process image.
The \fIenvp\fR array is terminated by a null pointer. For \fBexecl()\fR,
\fBexecv()\fR, \fBexecvp()\fR, and \fBexeclp()\fR, the C-language run-time
start-off routine places a pointer to the environment of the calling process in
the global object
.Va environ ,
and it is used to pass the
the global object \fBextern char **environ\fR, and it is used to pass the
environment of the calling process to the new process image.
.Lp
.sp
.LP
The number of bytes available for the new process's combined argument and
```

```
252 environment lists is
253 .Dv ARG_MAX .
254 It is implementation-dependent whether null
161 environment lists is \fBARG_MAX\fR. It is implementation-dependent whether null
255 terminators, pointers, and/or any alignment bytes are included in this total.
256 .Lp
163 .sp
164 .LP
257 File descriptors open in the calling process image remain open in the new
258 process image, except for those whose close-on-exec flag
259 .Dv FD_CLOEXEC
260 is set; see
261 .Xr fcntl 2 .
262 For those file descriptors that remain open, all
166 process image, except for those whose close-on-exec flag \fBFD_CLOEXEC\fR is
167 set; see \fBfcntl\fR(2). For those file descriptors that remain open, all
263 attributes of the open file description, including file locks, remain
264 unchanged.
265 .Lp
266 The preferred hardware address translation size
267 .Pq see Xr memcntl 2
268 for the
170 .sp
171 .LP
172 The preferred hardware address translation size (see \fBmemcntl\fR(2)) for the
269 stack and heap of the new process image are set to the default system page
270 size.
271 .Lp
175 .sp
176 .LP
272 Directory streams open in the calling process image are closed in the new
273 process image.
274 .Lp
179 .sp
180 .LP
275 The state of conversion descriptors and message catalogue descriptors in the
276 new process image is undefined. For the new process, the equivalent of:
277 .Lp
278 .Dl Fn setlocale LC_ALL \("C" ;
279 .Lp
183 .sp
184 .in +2
185 .nf
186 setlocale(LC_ALL, "C")
187 .fi
188 .in -2

190 .sp
191 .LP
280 is executed at startup.
281 .Lp
282 Signals set to the default action
283 .Pq Dv BSIG_DFL
284 in the calling process image
285 are set to the default action in the new process image
286 .Pq see Xr signal 3C .
287 Signals set to be ignored
288 .Pq Dv SIG_IGN
289 by the calling process image are set to be ignored by the new process image.
290 Signals set to be caught by the calling
291 process image are set to the default action in the new process image
292 .Pq see Xr signal.h 3HEAD .
293 After a successful call to any of the
294 .Nm exec
295 functions, alternate signal stacks are not preserved and the
296 .Dv SA_ONSTACK
```



```
193 .sp
194 .LP
195 Signals set to the default action (\fBSIG_DFL\fR) in the calling process image
196 are set to the default action in the new process image (see \fBsignal\fR(3C)).
197 Signals set to be ignored (\fBSIG_IGN\fR) by the calling process image are set
198 to be ignored by the new process image. Signals set to be caught by the calling
199 process image are set to the default action in the new process image (see
200 \fBsignal.h\fR(3HEAD)). After a successful call to any of the \fBexec\fR
201 functions, alternate signal stacks are not preserved and the \fBSA_ONSTACK\fR
297 flag is cleared for all signals.
298 .Lp
299 After a successful call to any of the
300 .Nm exec
301 functions, any functions
302 previously registered by
303 .Xr atexit 3C
304 are no longer registered.
305 .Lp
203 .sp
204 .LP
205 After a successful call to any of the \fBexec\fR functions, any functions
206 previously registered by \fBatexit\fR(3C) are no longer registered.
207 .sp
208 .LP
306 The saved resource limits in the new process image are set to be a copy of the
307 process's corresponding hard and soft resource limits.
308 .Lp
309 If the
310 .Dv ST_NOSUID
311 bit is set for the file system containing the new
312 process image file, then the effective user ID and effective group
313 ID are unchanged in the new process image.
314 If the set-user-ID mode bit of the new process image file is set
315 .Pq see Xr chmod 2 ,
316 the effective
317 user ID of the new process image is set to the owner ID of the new
318 process image file.
319 Similarly, if the set-group-ID mode bit of the new
320 process image file is set, the effective group ID of the new process
321 image is set to the group ID of the new process image file.
322 The real user ID and real group ID of the new process image remain the same as
323 those of the calling process image.
324 The effective user ID and effective group
325 ID of the new process image are saved
326 .Po
327 as the saved set-user-ID and the saved set-group-ID for use by
328 .Xr setuid 2 Pc .
329 .Lp
211 .sp
212 .LP
213 If the \fBST_NOSUID\fR bit is set for the file system containing the new
214 process image file, then the effective user \fBID\fR and effective group
215 \fBID\fR are unchanged in the new process image. If the set-user-\fBID\fR mode
216 bit of the new process image file is set (see \fBchmod\fR(2)), the effective
217 user \fBID\fR of the new process image is set to the owner \fBID\fR of the new
218 process image file. Similarly, if the set-group-\fBID\fR mode bit of the new
219 process image file is set, the effective group \fBID\fR of the new process
220 image is set to the group \fBID\fR of the new process image file. The real user
221 \fBID\fR and real group \fBID\fR of the new process image remain the same as
222 those of the calling process image. The effective user ID and effective group
223 ID of the new process image are saved (as the saved set-user-ID and the saved
224 set-group-ID for use by \fBsetuid\fR(2).
225 .sp
226 .LP
330 The privilege sets are changed according to the following rules:
331 .Bl -enum
```

```
.It
.RS +4
.TP
1.
The inheritable set, I, is intersected with the limit set, L.  This
mechanism enforces the limit set for processes.
.It
.RE
.RS +4
.TP
2.
The effective set, E, and the permitted set, P, are made equal to the new
inheritable set.
.El
.Lp
.RE
.sp
.LP
The system attempts to set the privilege-aware state to non-PA both before
performing any modifications to the process IDs and privilege sets as well as
after completing the transition to new UIDs and privilege sets, following the
rules outlined in
.Xr privileges 5 .
.Lp
If the
.Brq Dv PRIV_PROC_OWNER
privilege is asserted in the effective set, the
rules outlined in \fBprivileges\fR(5).
.sp
.LP
If the {\fBPRIV_PROC_OWNER\fR} privilege is asserted in the effective set, the
set-user-ID and set-group-ID bits will be honored when the process is being
controlled by
.Xr ptrace 3C .
Additional restrictions can apply when the
traced process has an effective UID of 0.
See
.Xr privileges 5 .
.Lp
controlled by \fBptrace\fR(3C). Additional restriction can apply when the
traced process has an effective UID of 0. See \fBprivileges\fR(5).
.sp
.LP
Any shared memory segments attached to the calling process image will not be
attached to the new process image
.Pq see Xr shmop 2 .
Any mappings
established through
.Xr mmap 2
are not preserved across an
.Nm exec .
Memory
attached to the new process image (see \fBshmop\fR(2)). Any mappings
established through \fBmmap()\fR are not preserved across an \fBexec\fR. Memory
mappings created in the process are unmapped before the address space is
rebuilt for the new process image.
See
.Xr mmap 2 .
.Lp
Memory locks established by the calling process via calls to
.Xr mlockall 3C
or
.Xr mlock 3C
are removed.
If locked pages in the address space of the
rebuilt for the new process image. See \fBmmap\fR(2).
```



```
.sp
.LP
Memory locks established by the calling process via calls to \fBmlockall\fR(3C)
or \fBmlock\fR(3C) are removed. If locked pages in the address space of the
calling process are also mapped into the address spaces the locks established
by the other processes will be unaffected by the call by this process to the
.Nm exec
function.
If the
.Nm exec
function fails, the effect on memory locks is unspecified.
.Lp
If
.Dv _XOPEN_REALTIME
is defined and has a value other than \(mi1, any named
\fBexec\fR function. If the \fBexec\fR function fails, the effect on memory
locks is unspecified.
.sp
.LP
If \fB_XOPEN_REALTIME\fR is defined and has a value other than \(mi1, any named
semaphores open in the calling process are closed as if by appropriate calls to
.Xr sem_close 3C .
.Lp
Profiling is disabled for the new process; see
.Xr profil 2 .
.Lp
Timers created by the calling process with
.Xr timer_create 3C
are deleted
\fBsem_close\fR(3C)
.sp
.LP
Profiling is disabled for the new process; see \fBprofil\fR(2).
.sp
.LP
Timers created by the calling process with \fBtimer_create\fR(3C) are deleted
before replacing the current process image with the new process image.
.Lp
For the
.Dv SCHED_FIFO
and
.Dv SCHED_RR
scheduling policies, the policy and
priority settings are not changed by a call to an
.Nm exec
function.
.Lp
.sp
.LP
For the \fBSCHED_FIFO\fR and \fBSCHED_RR\fR scheduling policies, the policy and
priority settings are not changed by a call to an \fBexec\fR function.
.sp
.LP
All open message queue descriptors in the calling process are closed, as
described in
.Xr mq_close 3C .
.Lp
described in \fBmq_close\fR(3C).
.sp
.LP
Any outstanding asynchronous I/O operations may be cancelled. Those
asynchronous I/O operations that are not canceled will complete as if the
.Nm exec
function had not yet occurred, but any associated signal
notifications are suppressed.
It is unspecified whether the
```

```
.Nm exec
function itself blocks awaiting such I/O completion.
In no event, however, will the new process image created by the
.Nm exec
function be affected by the presence of
outstanding asynchronous I/O operations at the time the
.Nm exec
function is called.
.Lp
All active contract templates are cleared
.Pq see Xr contract 4 .
.Lp
\fBexec\fR function had not yet occurred, but any associated signal
notifications are suppressed. It is unspecified whether the \fBexec\fR function
itself blocks awaiting such I/O completion. In no event, however, will the new
process image created by the \fBexec\fR function be affected by the presence of
outstanding asynchronous I/O operations at the time the \fBexec\fR function is
called.
.sp
.LP
All active contract templates are cleared (see \fBcontract\fR(4)).
.sp
.LP
The new process also inherits the following attributes from the calling
process:
.Bl -bullet -offset indent
.It
.RS +4
.TP
.ie t \(bu
.el o
controlling terminal
.It
.RE
.RS +4
.TP
.ie t \(bu
.el o
current working directory
.It
file-locks
.Po see
.Xr fcntl 2
and
.Xr lockf 3C
.Pc
.It
file mode creation mask
.Pq see Xr umask 2
.It
file size limit
.Pq see Xr ulimit 2
.It
.RE
.RS +4
.TP
.ie t \(bu
.el o
file-locks (see \fBfcntl\fR(2) and \fBlockf\fR(3C))
.RE
.RS +4
.TP
.ie t \(bu
.el o
file mode creation mask (see \fBumask\fR(2))
.RE
```



```
.RS +4
.TP
.ie t \(bu
.el o
file size limit (see \fBulimit\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
limit privilege set
.It
nice value
.Pq see Xr nice 2
.It
parent process ID
.It
pending signals
.Pq see Xr sigpending 2
.It
privilege debugging flag
.Po see Xr privileges 5
and
.Xr getpflags 2
.Pc
.It
process ID
.It
process contract
.Po see Xr contract 4
and
.Xr process 4
.Pc
.It
process group ID
.It
process signal mask
.Pq see Xr sigprocmask 2
.It
processor bindings
.Pq see Xr processor_bind 2
.It
processor set bindings
.Pq see Xr pset_bind 2
.It
project ID
.It
real group ID
.It
real user ID
.It
resource limits
.Pq see Xr getrlimit 2
.It
.RE
.RS +4
.TP
.ie t \(bu
.el o
nice value (see \fBnice\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
parent process \fBID\fR
```

```
.RE
.RS +4
.TP
.ie t \(bu
.el o
pending signals (see \fBsigpending\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
privilege debugging flag (see \fBprivileges\fR(5) and \fBgetpflags\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
process \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
process contract (see \fBcontract\fR(4) and \fBprocess\fR(4))
.RE
.RS +4
.TP
.ie t \(bu
.el o
process group \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
process signal mask (see \fBsigprocmask\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
processor bindings (see \fBprocessor_bind\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
processor set bindings (see \fBpset_bind\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
project \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
real group \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
real user \fBID\fR
```



```
.RE
.RS +4
.TP
.ie t \(bu
.el o
resource limits (see \fBgetrlimit\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
root directory
.It
scheduler class and priority
.Pq see Xr priocntl 2
.It
.Sy semadj
values
.Pq see Xr semop 2
.It
session membership
.Po see
.Xr exit 2
and
.Xr signal 3C
.Pc
.It
supplementary group IDs
.It
task ID
.It
time left until an alarm clock signal
.Pq see Xr alarm 2
.It
.Sy tms_utime , tms_stime , tms_cutime ,
and
.Sy tms_cstime
.Pq see Xr times 2
.It
trace flag
.Po see Xr ptrace 3C
request 0
.Pc
.El
.Lp
A call to any
.Nm exec
function from a process with more than one thread
.RE
.RS +4
.TP
.ie t \(bu
.el o
scheduler class and priority (see \fBpriocntl\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBsemadj\fR values (see \fBsemop\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
session membership (see \fBexit\fR(2) and \fBsignal\fR(3C))
```

```
.RE
.RS +4
.TP
.ie t \(bu
.el o
supplementary group \fBIDs\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
task \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
time left until an alarm clock signal (see \fBalarm\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBtms_utime\fR, \fBtms_stime\fR, \fBtms_cutime\fR, and \fBtms_cstime\fR (see
\fBtimes\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
trace flag (see \fBptrace\fR(3C) request 0)
.RE
.sp
.LP
A call to any \fBexec\fR function from a process with more than one thread
results in all threads being terminated and the new executable image being
loaded and executed.
No destructor functions will be called.
.Lp
Upon successful completion, each of the functions in the
.Nm exec
family marks for update the
.Va st_atime
field of the file.
If an
.Nm exec
function failed but was able to locate the
.Em process image file ,
whether
the
.Va st_atime
field is marked for update is unspecified.
Should the function succeed, the process image file is considered to have
been opened with
.Xr open 2 .
The corresponding
.Xr close 2
is considered to occur at a
loaded and executed. No destructor functions will be called.
.sp
.LP
Upon successful completion, each of the functions in the \fBexec\fR family
marks for update the \fBst_atime\fR field of the file.  If an \fBexec\fR
function failed but was able to locate the \fBprocess image file\fR, whether
the \fBst_atime\fR field is marked for update is unspecified. Should the
function succeed, the process image file is considered to have been opened with
```



```
\fBopen\fR(2). The corresponding \fBclose\fR(2) is considered to occur at a
time after this open, but before process termination or successful completion
of a subsequent call to one of the
.Nm exec
functions.
The
.Fa argv
and
.Fa envp
arrays of pointers and the strings to which those arrays point
will not be modified by a call to one of the
.Nm exec
functions, except as a consequence of replacing the process image.
.Lp
of a subsequent call to one of the \fBexec\fR functions. The \fIargv\fR[\|] and
\fIenvp\fR[\|] arrays of pointers and the strings to which those arrays point
will not be modified by a call to one of the \fBexec\fR functions, except as a
consequence of replacing the process image.
.sp
.LP
The saved resource limits in the new process image are set to be a copy of the
process's corresponding hard and soft limits.
.
.Sh RETURN VALUES
.
If a function in the
.Nm exec
family returns to the calling process image, an
error has occurred; the return value is \fB\(mi1\fR and
.Va errno
is set to indicate the error.
.
.Sh ERRORS
.
The
.Nm exec
functions will fail if:
.Bl -tag -width Er
.It Bq Er E2BIG
.SH RETURN VALUES
.sp
.LP
If a function in the \fBexec\fR family returns to the calling process image, an
error has occurred; the return value is \fB\(mi1\fR and \fBerrno\fR is set to
indicate the error.
.SH ERRORS
.sp
.LP
The \fBexec\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBE2BIG\fR\fR
.ad
.RS 16n
The number of bytes in the new process's argument list is greater than the
system-imposed limit of
.Brq Dv ARG_MAX
bytes.
The argument list limit is sum
system-imposed limit of {\fBARG_MAX\fR} bytes. The argument list limit is sum
of the size of the argument list plus the size of the environment's exported
shell variables.
.
.It Bq Er EACCES
.RE
```

```
.sp
.ne 2
.na
\fB\fBEACCES\fR\fR
.ad
.RS 16n
Search permission is denied for a directory listed in the new process file's
path prefix.
.sp
The new process file is not an ordinary file.
.sp
The new process file mode denies execute permission.
.sp
The
.Brq Dv FILE_DAC_SEARCH
privilege overrides the restriction on directory searches.
The {\fBFILE_DAC_SEARCH\fR} privilege overrides the restriction on directory
searches.
.sp
The
.Brq Dv FILE_DAC_EXECUTE
privilege overrides the lack of execute
The {\fBFILE_DAC_EXECUTE\fR} privilege overrides the lack of execute
permission.
.
.It Bq Er EAGAIN
.RE

.sp
.ne 2
.na
\fB\fBEAGAIN\fR\fR
.ad
.RS 16n
Total amount of system memory available when reading using raw I/O is
temporarily insufficient.
.
.It Bq Er EFAULT
.RE

.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 16n
An argument points to an illegal address.
.
.It Bq Er EINVAL
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 16n
The new process image file has the appropriate permission and has a recognized
executable binary format, but the system does not support execution of a file
with this format.
.
.It Bq Er EINTR
.RE

.sp
```



```
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 16n
A signal was caught during the execution of one of the functions in the
.Nm exec
family.
.El
.
.Lp
The
.Nm exec
functions except
.Fn fexecve
will fail if:
.
.Bl -tag -width Er
.It Bq Er ELOOP
Too many symbolic links were encountered in translating
.Fa path
or
.Fa file .
.
.It Bq Er ENAMETOOLONG
The length of the
.Fa file
or
.Fa path
argument exceeds
.Brq PATH_MAX ,
or the length of a
.Fa file
or
.Fa path
component exceeds
.Brq Dv NAME_MAX
while
.Brq Dv _POSIX_NO_TRUNC
is in effect.
.
.It Bq Er ENOENT
\fIexec\fR family.
.RE

.sp
.ne 2
.na
\fB\fBELOOP\fR\fR
.ad
.RS 16n
Too many symbolic links were encountered in translating \fIpath\fR or
\fIfile\fR.
.RE

.sp
.ne 2
.na
\fB\fBENAMETOOLONG\fR\fR
.ad
.RS 16n
The length of the \fIfile\fR or \fIpath\fR argument exceeds {\fBPATH_MAX\fR},
or the length of a \fIfile\fR or \fIpath\fR component exceeds {\fBNAME_MAX\fR}
while {\fB_POSIX_NO_TRUNC\fR} is in effect.
.RE
```

```
.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 16n
```
One or more components of the new process path name of the file do not exist or is a null pathname.
```
.
.It Bq Er ENOLINK
The
.Fa path
argument points to a remote machine and the link to that machine
.RE

.sp
.ne 2
.na
\fB\fBENOLINK\fR\fR
.ad
.RS 16n
The \fIpath\fR argument points to a remote machine and the link to that machine
```
is no longer active.
```
.
.It Bq Er ENOTDIR
.RE

.sp
.ne 2
.na
\fB\fBENOTDIR\fR\fR
.ad
.RS 16n
```
A component of the new process path of the file prefix is not a directory.
```
.El
.Lp
The
.Nm exec
functions, except for
.Fn execlp
and
.Fn execvp ,
will fail if:
.Bl -tag -width Er
.It Bq Er ENOEXEC
.RE

.sp
.LP
The \fBexec\fR functions, except for \fBexeclp()\fR and \fBexecvp()\fR, will fail if:
.sp
.ne 2
.na
\fB\fBENOEXEC\fR\fR
.ad
.RS 11n
```
The new process image file has the appropriate access permission but is not in the proper format.
```
.El
.Lp
The
.Fn fexecve
function will fail if:
.Bl -tag -width Er
.It Bq Er EBADF
```



```
The
.Fa fd
argument is not a valid file descriptor opened for execute.
.El
.RE

.sp
.Lp
The
.Nm exec
functions except for
.Fn fexecve
may fail if:
.
.Bl -tag -width Er
.It Bq Er ENAMETOOLONG
.LP
The \fBexec\fR functions may fail if:
.sp
.ne 2
.na
\fB\fBENAMETOOLONG\fR\fR
.ad
.RS 16n
```
Pathname resolution of a symbolic link produced an intermediate result whose length exceeds
.Brq Dv PATH_MAX .
.
.It Bq Er ENOMEM
length exceeds {\fBPATH_MAX\fR}.
.RE

.sp
.ne 2
.na
\fB\fBENOMEM\fR\fR
.ad
.RS 16n
```
The new process image requires more memory than is allowed by the hardware or system-imposed by memory management constraints.
See
.Xr brk 2 .
.
.It Bq Er ETXTBSY
system-imposed by memory management constraints. See \fBbrk\fR(2).
.RE

.sp
.ne 2
.na
\fB\fBETXTBSY\fR\fR
.ad
.RS 16n
```
The new process image file is a pure procedure (shared text) file that is currently open for writing by some process.
```
.El
.
.Sh USAGE
.
.RE

.SH USAGE
.sp
.LP
```
As the state of conversion descriptors and message catalogue descriptors in the new process image is undefined, portable applications should not rely on their
```
```

```
726 use and should close them prior to calling one of the
727 .Nm exec
728 functions.
729 .Lp
685 use and should close them prior to calling one of the \fBexec\fR functions.
686 .sp
687 .LP
730 Applications that require other than the default POSIX locale should call
731 .Xr setlocale 3C
732 with the appropriate parameters to establish the locale of the new process.
733 .Lp
734 The
735 .Va environ
736 array should not be accessed directly by the application.
737 Instead,
738 .Xr getenv 3C
739 should be used.
740 .
741 .Sh SEE ALSO
742 .Xr ps 1 ,
743 .Xr sh 1 ,
744 .Xr alarm 2 ,
745 .Xr brk 2 ,
746 .Xr chmod 2 ,
747 .Xr exit 2 ,
748 .Xr fcntl 2 ,
749 .Xr fork 2 ,
750 .Xr getpflags 2 ,
751 .Xr getrlimit 2 ,
752 .Xr memcntl 2 ,
753 .Xr mmap 2 ,
754 .Xr nice 2 ,
755 .Xr open 2 ,
756 .Xr priocntl 2 ,
757 .Xr profil 2 ,
758 .Xr semop 2 ,
759 .Xr shmop 2 ,
760 .Xr sigpending 2 ,
761 .Xr sigprocmask 2 ,
762 .Xr times 2 ,
763 .Xr umask 2 ,
764 .Xr lockf 3C ,
765 .Xr ptrace 3C ,
766 .Xr setlocale 3C ,
767 .Xr signal 3C ,
768 .Xr system 3C ,
769 .Xr timer_create 3C ,
770 .Xr fcntl.h 3HEAD ,
771 .Xr signal.h 3HEAD ,
772 .Xr unistd.h 3HEAD ,
773 .Xr a.out 4 ,
774 .Xr contract 4 ,
775 .Xr process 4 ,
776 .Xr environ 5 ,
777 .Xr privileges 5 ,
778 .Xr standards 5
779 .
780 .Sh INTERFACE STABILITY
781 .
782 .Sy Standard .
783 .
784 .Sh MT-LEVEL
785 .
786 The
787 .Fn execle , execve , fexecve
788 functions are
```



```
789 .Sy Async-Signal-Safe .
790 .
791 .Sh STANDARDS
792 .
793 These functions are available in the following compilation environments.
794 See
795 .Xr standards 5 .
796 .Ss Fn execl , execle , execlp , execv , execve , execvp
797 .Bl -bullet -compact
798 .It
799 .St -p1003.1-90
800 .It
801 .St -xpg3
802 .It
803 .St -xpg4
804 .It
805 .St -xpg4.2
806 .It
807 .St -susv2
808 .It
809 .St -susv3
810 .It
811 .St -p1003.1-2008
812 .El
813 .Ss Fn fexecve
814 .Bl -bullet -compact
815 .It
816 .St -p1003.1-2008
817 .El
818 .
819 .Sh WARNINGS
820 If a program is
821 .Em setuid
822 to a user ID other than the superuser, and
823 the program is executed when the real user ID is super-user, then the
689 \fBsetlocale\fR(3C) with the appropriate parameters to establish the locale of
690 thenew process.
691 .sp
692 .LP
693 The \fIenviron\fR array should not be accessed directly by the application.
694 .SH ATTRIBUTES
695 .sp
696 .LP
697 See \fBattributes\fR(5) for descriptions of the following attributes:
698 .sp

700 .sp
701 .TS
702 box;
703 c | c
704 l | l .
705 ATTRIBUTE TYPE  ATTRIBUTE VALUE
706 _
707 Interface Stability     Committed
708 _
709 MT-Level        See below.
710 _
711 Standard        See \fBstandards\fR(5).
712 .TE

714 .sp
715 .LP
716 The \fBexecle()\fR and \fBexecve()\fR fucntions are Async-Signal-Safe.
717 .SH SEE ALSO
718 .sp
719 .LP
```

```
\fBksh\fR(1), \fBps\fR(1), \fBsh\fR(1), \fBalarm\fR(2), \fBbrk\fR(2),
\fBchmod\fR(2), \fBexit\fR(2), \fBfcntl\fR(2), \fBfork\fR(2),
\fBgetpflags\fR(2), \fBgetrlimit\fR(2), \fBmemcntl\fR(2), \fBmmap\fR(2),
\fBnice\fR(2), \fBpriocntl\fR(2), \fBprofil\fR(2), \fBsemop\fR(2),
\fBshmop\fR(2), \fBsigpending\fR(2), \fBsigprocmask\fR(2), \fBtimes\fR(2),
\fBumask\fR(2), \fBlockf\fR(3C), \fBptrace\fR(3C), \fBsetlocale\fR(3C),
\fBsignal\fR(3C), \fBsystem\fR(3C), \fBtimer_create\fR(3C), \fBa.out\fR(4),
\fBcontract\fR(4), \fBprocess\fR(4), \fBattributes\fR(5), \fBenviron\fR(5),
\fBprivileges\fR(5), \fBstandards\fR(5)
.SH WARNINGS
.sp
.LP
If a program is \fBsetuid\fR to a user \fBID\fR other than the superuser, and
the program is executed when the real user \fBID\fR is super-user, then the
program has some of the powers of a super-user as well.
```

```
**********************************************************
   11493 Mon Apr  6 11:47:36 2015
new/usr/src/pkg/manifests/system-kernel.man2.inc
5798 fexecve() needed per POSIX 2008
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet
# at http://www.illumos.org/license/CDDL.
#

#
# Copyright 2011, Richard Lowe
# Copyright 2012 Nexenta Systems, Inc. All rights reserved.
# Copyright 2013, OmniTI Computer Consulting, Inc.
# Copyright 2015 Garrett D'Amore <garrett@damore.org>
#

file path=usr/share/man/man2/Intro.2
file path=usr/share/man/man2/__sparc_utrap_install.2
file path=usr/share/man/man2/_lwp_cond_signal.2
file path=usr/share/man/man2/_lwp_cond_wait.2
file path=usr/share/man/man2/_lwp_info.2
file path=usr/share/man/man2/_lwp_kill.2
file path=usr/share/man/man2/_lwp_mutex_lock.2
file path=usr/share/man/man2/_lwp_self.2
file path=usr/share/man/man2/_lwp_sema_wait.2
file path=usr/share/man/man2/_lwp_suspend.2
file path=usr/share/man/man2/access.2
file path=usr/share/man/man2/acct.2
file path=usr/share/man/man2/acl.2
file path=usr/share/man/man2/adjtime.2
file path=usr/share/man/man2/alarm.2
file path=usr/share/man/man2/audit.2
file path=usr/share/man/man2/auditon.2
file path=usr/share/man/man2/brk.2
file path=usr/share/man/man2/chdir.2
file path=usr/share/man/man2/chmod.2
file path=usr/share/man/man2/chown.2
file path=usr/share/man/man2/chroot.2
file path=usr/share/man/man2/close.2
file path=usr/share/man/man2/creat.2
file path=usr/share/man/man2/dup.2
file path=usr/share/man/man2/exec.2
file path=usr/share/man/man2/exit.2
file path=usr/share/man/man2/fcntl.2
file path=usr/share/man/man2/fork.2
file path=usr/share/man/man2/fpathconf.2
file path=usr/share/man/man2/getacct.2
file path=usr/share/man/man2/getaudit.2
file path=usr/share/man/man2/getauid.2
file path=usr/share/man/man2/getcontext.2
file path=usr/share/man/man2/getdents.2
file path=usr/share/man/man2/getgroups.2
file path=usr/share/man/man2/getisax.2
file path=usr/share/man/man2/getitimer.2
file path=usr/share/man/man2/getlabel.2
file path=usr/share/man/man2/getmsg.2
file path=usr/share/man/man2/getpflags.2
file path=usr/share/man/man2/getpid.2
file path=usr/share/man/man2/getppriv.2
```



```
file path=usr/share/man/man2/getrlimit.2
file path=usr/share/man/man2/getsid.2
file path=usr/share/man/man2/getuid.2
file path=usr/share/man/man2/getustack.2
file path=usr/share/man/man2/ioctl.2
file path=usr/share/man/man2/issetugid.2
file path=usr/share/man/man2/kill.2
file path=usr/share/man/man2/link.2
file path=usr/share/man/man2/llseek.2
file path=usr/share/man/man2/lseek.2
file path=usr/share/man/man2/memcntl.2
file path=usr/share/man/man2/meminfo.2
file path=usr/share/man/man2/mincore.2
file path=usr/share/man/man2/mkdir.2
file path=usr/share/man/man2/mknod.2
file path=usr/share/man/man2/mmap.2
file path=usr/share/man/man2/mmapobj.2
file path=usr/share/man/man2/mount.2
file path=usr/share/man/man2/mprotect.2
file path=usr/share/man/man2/msgctl.2
file path=usr/share/man/man2/msgget.2
file path=usr/share/man/man2/msgids.2
file path=usr/share/man/man2/msgrcv.2
file path=usr/share/man/man2/msgsnap.2
file path=usr/share/man/man2/msgsnd.2
file path=usr/share/man/man2/munmap.2
file path=usr/share/man/man2/nice.2
file path=usr/share/man/man2/ntp_adjtime.2
file path=usr/share/man/man2/ntp_gettime.2
file path=usr/share/man/man2/open.2
file path=usr/share/man/man2/p_online.2
file path=usr/share/man/man2/pause.2
file path=usr/share/man/man2/pcsample.2
file path=usr/share/man/man2/pipe.2
file path=usr/share/man/man2/poll.2
file path=usr/share/man/man2/priocntl.2
file path=usr/share/man/man2/priocntlset.2
file path=usr/share/man/man2/processor_bind.2
file path=usr/share/man/man2/processor_info.2
file path=usr/share/man/man2/profil.2
file path=usr/share/man/man2/pset_bind.2
file path=usr/share/man/man2/pset_create.2
file path=usr/share/man/man2/pset_info.2
file path=usr/share/man/man2/pset_list.2
file path=usr/share/man/man2/pset_setattr.2
file path=usr/share/man/man2/putmsg.2
file path=usr/share/man/man2/read.2
file path=usr/share/man/man2/readlink.2
file path=usr/share/man/man2/rename.2
file path=usr/share/man/man2/resolvepath.2
file path=usr/share/man/man2/rmdir.2
file path=usr/share/man/man2/semctl.2
file path=usr/share/man/man2/semget.2
file path=usr/share/man/man2/semids.2
file path=usr/share/man/man2/semop.2
file path=usr/share/man/man2/setpgid.2
file path=usr/share/man/man2/setpgrp.2
file path=usr/share/man/man2/setrctl.2
file path=usr/share/man/man2/setregid.2
file path=usr/share/man/man2/setreuid.2
file path=usr/share/man/man2/setsid.2
file path=usr/share/man/man2/settaskid.2
file path=usr/share/man/man2/setuid.2
file path=usr/share/man/man2/shmctl.2
file path=usr/share/man/man2/shmget.2
file path=usr/share/man/man2/shmids.2
```

```
128 file path=usr/share/man/man2/shmop.2
129 file path=usr/share/man/man2/sigaction.2
130 file path=usr/share/man/man2/sigaltstack.2
131 file path=usr/share/man/man2/sigpending.2
132 file path=usr/share/man/man2/sigprocmask.2
133 file path=usr/share/man/man2/sigsend.2
134 file path=usr/share/man/man2/sigsuspend.2
135 file path=usr/share/man/man2/sigwait.2
136 file path=usr/share/man/man2/stat.2
137 file path=usr/share/man/man2/statvfs.2
138 file path=usr/share/man/man2/stime.2
139 file path=usr/share/man/man2/swapctl.2
140 file path=usr/share/man/man2/symlink.2
141 file path=usr/share/man/man2/sync.2
142 file path=usr/share/man/man2/sysfs.2
143 file path=usr/share/man/man2/sysinfo.2
144 file path=usr/share/man/man2/time.2
145 file path=usr/share/man/man2/times.2
146 file path=usr/share/man/man2/uadmin.2
147 file path=usr/share/man/man2/ulimit.2
148 file path=usr/share/man/man2/umask.2
149 file path=usr/share/man/man2/umount.2
150 file path=usr/share/man/man2/uname.2
151 file path=usr/share/man/man2/unlink.2
152 file path=usr/share/man/man2/ustat.2
153 file path=usr/share/man/man2/utime.2
154 file path=usr/share/man/man2/utimes.2
155 file path=usr/share/man/man2/uucopy.2
156 file path=usr/share/man/man2/vfork.2
157 file path=usr/share/man/man2/vhangup.2
158 file path=usr/share/man/man2/waitid.2
159 file path=usr/share/man/man2/write.2
160 file path=usr/share/man/man2/yield.2
161 link path=usr/share/man/man2/_Exit.2 target=exit.2
162 link path=usr/share/man/man2/_exit.2 target=exit.2
163 link path=usr/share/man/man2/_lwp_cond_broadcast.2 target=_lwp_cond_signal.2
164 link path=usr/share/man/man2/_lwp_cond_reltimedwait.2 target=_lwp_cond_wait.2
165 link path=usr/share/man/man2/_lwp_cond_timedwait.2 target=_lwp_cond_wait.2
166 link path=usr/share/man/man2/_lwp_continue.2 target=_lwp_suspend.2
167 link path=usr/share/man/man2/_lwp_mutex_trylock.2 target=_lwp_mutex_lock.2
168 link path=usr/share/man/man2/_lwp_mutex_unlock.2 target=_lwp_mutex_lock.2
169 link path=usr/share/man/man2/_lwp_sema_init.2 target=_lwp_sema_wait.2
170 link path=usr/share/man/man2/_lwp_sema_post.2 target=_lwp_sema_wait.2
171 link path=usr/share/man/man2/_lwp_sema_trywait.2 target=_lwp_sema_wait.2
172 link path=usr/share/man/man2/execl.2 target=exec.2
173 link path=usr/share/man/man2/execle.2 target=exec.2
174 link path=usr/share/man/man2/execlp.2 target=exec.2
175 link path=usr/share/man/man2/execv.2 target=exec.2
176 link path=usr/share/man/man2/execve.2 target=exec.2
177 link path=usr/share/man/man2/execvp.2 target=exec.2
178 link path=usr/share/man/man2/faccessat.2 target=access.2
179 link path=usr/share/man/man2/facl.2 target=acl.2
180 link path=usr/share/man/man2/fchdir.2 target=chdir.2
181 link path=usr/share/man/man2/fchmod.2 target=chmod.2
182 link path=usr/share/man/man2/fchmodat.2 target=chmod.2
183 link path=usr/share/man/man2/fchown.2 target=chown.2
184 link path=usr/share/man/man2/fchownat.2 target=chown.2
185 link path=usr/share/man/man2/fchroot.2 target=chroot.2
186 link path=usr/share/man/man2/fexecve.2 target=exec.2
187 link path=usr/share/man/man2/fgetlabel.2 target=getlabel.2
188 link path=usr/share/man/man2/fork1.2 target=fork.2
189 link path=usr/share/man/man2/forkall.2 target=fork.2
190 link path=usr/share/man/man2/forkallx.2 target=fork.2
191 link path=usr/share/man/man2/forkx.2 target=fork.2
192 link path=usr/share/man/man2/fstat.2 target=stat.2
193 link path=usr/share/man/man2/fstatat.2 target=stat.2
```



```
194 link path=usr/share/man/man2/fstatvfs.2 target=statvfs.2
195 link path=usr/share/man/man2/futimens.2 target=utimes.2
196 link path=usr/share/man/man2/futimesat.2 target=utimes.2
197 link path=usr/share/man/man2/getaudit_addr.2 target=getaudit.2
198 link path=usr/share/man/man2/getegid.2 target=getuid.2
199 link path=usr/share/man/man2/geteuid.2 target=getuid.2
200 link path=usr/share/man/man2/getgid.2 target=getuid.2
201 link path=usr/share/man/man2/getpgid.2 target=getpid.2
202 link path=usr/share/man/man2/getpgrp.2 target=getpid.2
203 link path=usr/share/man/man2/getpmsg.2 target=getmsg.2
204 link path=usr/share/man/man2/getppid.2 target=getpid.2
205 link path=usr/share/man/man2/getprojid.2 target=settaskid.2
206 link path=usr/share/man/man2/getrctl.2 target=setrctl.2
207 link path=usr/share/man/man2/gettaskid.2 target=settaskid.2
208 link path=usr/share/man/man2/intro.2 target=Intro.2
209 link path=usr/share/man/man2/linkat.2 target=link.2
210 link path=usr/share/man/man2/lchown.2 target=chown.2
211 link path=usr/share/man/man2/lstat.2 target=stat.2
212 link path=usr/share/man/man2/mkdirat.2 target=mkdir.2
213 link path=usr/share/man/man2/mknodat.2 target=mknod.2
214 link path=usr/share/man/man2/openat.2 target=open.2
215 link path=usr/share/man/man2/pathconf.2 target=fpathconf.2
216 link path=usr/share/man/man2/pipe2.2 target=pipe.2
217 link path=usr/share/man/man2/ppoll.2 target=poll.2
218 link path=usr/share/man/man2/pread.2 target=read.2
219 link path=usr/share/man/man2/preadv.2 target=read.2
220 link path=usr/share/man/man2/pset_assign.2 target=pset_create.2
221 link path=usr/share/man/man2/pset_destroy.2 target=pset_create.2
222 link path=usr/share/man/man2/pset_getattr.2 target=pset_setattr.2
223 link path=usr/share/man/man2/putacct.2 target=getacct.2
224 link path=usr/share/man/man2/putpmsg.2 target=putmsg.2
225 link path=usr/share/man/man2/pwrite.2 target=write.2
226 link path=usr/share/man/man2/pwritev.2 target=write.2
227 link path=usr/share/man/man2/readlinkat.2 target=readlink.2
228 link path=usr/share/man/man2/readv.2 target=read.2
229 link path=usr/share/man/man2/renameat.2 target=rename.2
230 link path=usr/share/man/man2/sbrk.2 target=brk.2
231 link path=usr/share/man/man2/semtimedop.2 target=semop.2
232 link path=usr/share/man/man2/setaudit.2 target=getaudit.2
233 link path=usr/share/man/man2/setaudit_addr.2 target=getaudit.2
234 link path=usr/share/man/man2/setauid.2 target=getauid.2
235 link path=usr/share/man/man2/setcontext.2 target=getcontext.2
236 link path=usr/share/man/man2/setegid.2 target=setuid.2
237 link path=usr/share/man/man2/seteuid.2 target=setuid.2
238 link path=usr/share/man/man2/setgid.2 target=setuid.2
239 link path=usr/share/man/man2/setgroups.2 target=getgroups.2
240 link path=usr/share/man/man2/setitimer.2 target=getitimer.2
241 link path=usr/share/man/man2/setpflags.2 target=getpflags.2
242 link path=usr/share/man/man2/setppriv.2 target=getppriv.2
243 link path=usr/share/man/man2/setrlimit.2 target=getrlimit.2
244 link path=usr/share/man/man2/setustack.2 target=getustack.2
245 link path=usr/share/man/man2/shmat.2 target=shmop.2
246 link path=usr/share/man/man2/shmdt.2 target=shmop.2
247 link path=usr/share/man/man2/sigsendset.2 target=sigsend.2
248 link path=usr/share/man/man2/symlinkat.2 target=symlink.2
249 link path=usr/share/man/man2/umount2.2 target=umount.2
250 link path=usr/share/man/man2/unlinkat.2 target=unlink.2
251 link path=usr/share/man/man2/utimensat.2 target=utimes.2
252 link path=usr/share/man/man2/vforkx.2 target=vfork.2
253 link path=usr/share/man/man2/wracct.2 target=getacct.2
254 link path=usr/share/man/man2/writev.2 target=write.2
```

```
**********************************************************
    5251 Mon Apr  6 11:47:36 2015
new/usr/src/pkg/manifests/system-test-libctest.mf
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

  12 #
  13 # Copyright (c) 2012 by Delphix. All rights reserved.
  14 # Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.
  15 # Copyright 2015 Garrett D'Amore <garrett@damore.org>
  16 #

  18 set name=pkg.fmri value=pkg:/system/test/libctest@$(PKGVERS)
  19 set name=pkg.description value="C library Unit Tests"
  20 set name=pkg.summary value="C Library Unit Test Suite"
  21 set name=info.classification \
  22     value=org.opensolaris.category.2008:Development/System
  23 set name=variant.arch value=$(ARCH)
  24 dir path=opt/libc-tests
  25 dir path=opt/libc-tests/bin
  26 dir path=opt/libc-tests/cfg
  27 dir path=opt/libc-tests/cfg/symbols
  28 dir path=opt/libc-tests/runfiles
  29 dir path=opt/libc-tests/tests
  30 dir path=opt/libc-tests/tests
  31 dir path=opt/libc-tests/tests/symbols
  32 file path=opt/libc-tests/README mode=0444
  33 file path=opt/libc-tests/bin/libctest mode=0555
  34 file path=opt/libc-tests/cfg/README mode=0444
  35 file path=opt/libc-tests/cfg/compilation.cfg mode=0444
  36 file path=opt/libc-tests/cfg/symbols/README mode=0444
  37 file path=opt/libc-tests/cfg/symbols/ctype_h.cfg mode=0444
  38 file path=opt/libc-tests/cfg/symbols/dirent_h.cfg mode=0444
  39 file path=opt/libc-tests/cfg/symbols/fcntl_h.cfg mode=0444
  40 file path=opt/libc-tests/cfg/symbols/locale_h.cfg mode=0444
  41 file path=opt/libc-tests/cfg/symbols/math_h.cfg mode=0444
  42 file path=opt/libc-tests/cfg/symbols/netdb_h.cfg mode=0444
  43 file path=opt/libc-tests/cfg/symbols/pthread_h.cfg mode=0444
  44 file path=opt/libc-tests/cfg/symbols/signal_h.cfg mode=0444
  45 file path=opt/libc-tests/cfg/symbols/stdio_h.cfg mode=0444
  46 file path=opt/libc-tests/cfg/symbols/stdlib_h.cfg mode=0444
  47 file path=opt/libc-tests/cfg/symbols/strings_h.cfg mode=0444
  48 file path=opt/libc-tests/cfg/symbols/sys_stat_h.cfg mode=0444
  49 file path=opt/libc-tests/cfg/symbols/sys_time_h.cfg mode=0444
  50 file path=opt/libc-tests/cfg/symbols/sys_timeb_h.cfg mode=0444
  51 file path=opt/libc-tests/cfg/symbols/ucontext_h.cfg mode=0444
  52 file path=opt/libc-tests/cfg/symbols/unistd_h.cfg mode=0444
  53 file path=opt/libc-tests/cfg/symbols/wchar_h.cfg mode=0444
  54 file path=opt/libc-tests/cfg/symbols/wctype_h.cfg mode=0444
  55 file path=opt/libc-tests/runfiles/default.run mode=0444
  56 file path=opt/libc-tests/tests/fexecve_test mode=0555
  57 file path=opt/libc-tests/tests/fexecve_test.$(ARCH) mode=0555
  58 file path=opt/libc-tests/tests/fexecve_test.$(ARCH64) mode=0555
  59 file path=opt/libc-tests/tests/fpround_test mode=0555
  60 file path=opt/libc-tests/tests/fpround_test.$(ARCH) mode=0555
  61 file path=opt/libc-tests/tests/fpround_test.$(ARCH64) mode=0555
```



```
  62 file path=opt/libc-tests/tests/newlocale_test mode=0555
  63 file path=opt/libc-tests/tests/newlocale_test.$(ARCH) mode=0555
  64 file path=opt/libc-tests/tests/newlocale_test.$(ARCH64) mode=0555
  65 file path=opt/libc-tests/tests/nl_langinfo_test mode=0555
  66 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH) mode=0555
  67 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH64) mode=0555
  68 file path=opt/libc-tests/tests/symbols/setup mode=0555
  69 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH) mode=0555
  70 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH64) mode=0555
  71 file path=opt/libc-tests/tests/wcsrtombs_test mode=0555
  72 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH) mode=0555
  73 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH64) mode=0555
  74 file path=opt/libc-tests/tests/wctype_test mode=0555
  75 file path=opt/libc-tests/tests/wctype_test.$(ARCH) mode=0555
  76 file path=opt/libc-tests/tests/wctype_test.$(ARCH64) mode=0555
  77 hardlink path=opt/libc-tests/tests/symbols/ctype_h target=setup
  78 hardlink path=opt/libc-tests/tests/symbols/dirent_h target=setup
  79 hardlink path=opt/libc-tests/tests/symbols/fcntl_h target=setup
  80 hardlink path=opt/libc-tests/tests/symbols/locale_h target=setup
  81 hardlink path=opt/libc-tests/tests/symbols/math_h target=setup
  82 hardlink path=opt/libc-tests/tests/symbols/netdb_h target=setup
  83 hardlink path=opt/libc-tests/tests/symbols/pthread_h target=setup
  84 hardlink path=opt/libc-tests/tests/symbols/signal_h target=setup
  85 hardlink path=opt/libc-tests/tests/symbols/stdio_h target=setup
  86 hardlink path=opt/libc-tests/tests/symbols/stdlib_h target=setup
  87 hardlink path=opt/libc-tests/tests/symbols/strings_h target=setup
  88 hardlink path=opt/libc-tests/tests/symbols/sys_stat_h target=setup
  89 hardlink path=opt/libc-tests/tests/symbols/sys_time_h target=setup
  90 hardlink path=opt/libc-tests/tests/symbols/sys_timeb_h target=setup
  91 hardlink path=opt/libc-tests/tests/symbols/ucontext_h target=setup
  92 hardlink path=opt/libc-tests/tests/symbols/unistd_h target=setup
  93 hardlink path=opt/libc-tests/tests/symbols/wchar_h target=setup
  94 hardlink path=opt/libc-tests/tests/symbols/wctype_h target=setup
  95 license lic_CDDL license=lic_CDDL
  96 depend fmri=locale/de type=require
  97 depend fmri=locale/en type=require
  98 depend fmri=locale/en-extra type=require
  99 depend fmri=locale/ja type=require
 100 depend fmri=locale/ru type=require
 101 depend fmri=system/test/testrunner type=require
```

```
**********************************************************
    3679 Mon Apr  6 11:47:36 2015
new/usr/src/test/libc-tests/cfg/symbols/unistd_h.cfg
5798 fexecve() needed per POSIX 2008
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2015 Garrett D'Amore <garrett@damore.org>
#

#
# Definitions found in unistd.h
#

#
# Types.
#
type | pid_t | unistd.h | POSIX+ SUS+

#
# Values.
#
# Note that the standard requires the user declare environ.
# value | environ | char ** | unistd.h | POSIX+ SUS+
value | _CS_PATH                                | int | unistd.h | SUS+

value | _CS_POSIX_V6_ILP32_OFF32_CFLAGS         | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_ILP32_OFF32_LDFLAGS        | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_ILP32_OFF32_LIBS           | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_ILP32_OFFBIG_CFLAGS        | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS       | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_ILP32_OFFBIG_LIBS          | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LP64_OFF64_CFLAGS          | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LP64_OFF64_LDFLAGS         | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LP64_OFF64_LIBS            | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS        | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS       | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_LPBIG_OFFBIG_LIBS          | int | unistd.h | SUSv3+
value | _CS_POSIX_V6_WIDTH_RESTRICTED_ENVS      | int | unistd.h | SUSv3+

#
# Functions
#
func |  access                                                  |\
        int                                                     |\
        const char *; int                                       |\
        unistd.h | POSIX+ SUS+

func |  chown                                                   |\
        int                                                     |\
        const char *; uid_t; gid_t                              |\
        unistd.h | POSIX+ SUS+

func |  execl                                                   |\
        int                                                     |\
        const char *; const char *                              |\
```



```
        unistd.h | POSIX+ SUS+

func |  execle                                                  |\
        int                                                     |\
        const char *; const char *; char *; char *const []      |\
        unistd.h | POSIX+ SUS+

func |  execlp                                                  |\
        int                                                     |\
        const char *; const char *; char *                      |\
        unistd.h | POSIX+ SUS+

func |  execv                                                   |\
        int                                                     |\
        const char *; char *const []                            |\
        unistd.h | POSIX+ SUS+

func |  execve                                                  |\
        int                                                     |\
        const char *; char *const []; char *const []            |\
        unistd.h | POSIX+ SUS+

func |  execvp                                                  |\
        int                                                     |\
        const char *; char *const []                            |\
        unistd.h | POSIX+ SUS+

func |  fchown                                                  |\
        int                                                     |\
        int; uid_t; gid_t                                       |\
        unistd.h | -POSIX+ SUS+

func |  fexecve                                                 |\
        int                                                     |\
        int; char *const []; char *const []                     |\
        unistd.h | -ALL SUSv4+

func |  getlogin                                                |\
        char *                                                  |\
        void                                                    |\
        unistd.h | POSIX+ SUS+

func |  getlogin_r                                              |\
        int                                                     |\
        char *; size_t                                          |\
        unistd.h | -POSIX+ -SUS+ +POSIX-1995+ SUSv2+

func |  lchown                                                  |\
        int                                                     |\
        const char *; uid_t; gid_t                              |\
        unistd.h | -POSIX+ SUS+

func |  link                                                    |\
        int                                                     |\
        const char *; const char *                              |\
        unistd.h | POSIX+ SUS+

# XPG3 may have put this here incorrectly (Open Group says no..., but...)
# Probably this is actually our error, and we should kill it, but we can
# do that when kill off XPG3 support altogether.
func |  rename                                                  |\
        int                                                     |\
        const char *; const char *                              |\
        unistd.h | -POSIX+ +XPG3 -XPG4+

func |  symlink                                                 |\
```

```
128         int                                                     |\
129         const char *; const char *                              |\
130         unistd.h | -XPG3+ -POSIX+ SUS+

132 func |  ttyname                                                  |\
133         char *                                                  |\
134         int                                                     |\
135         unistd.h | POSIX+ SUS+

137 func |  ttyname_r                                                |\
138         int                                                     |\
139         int; char *; size_t                                     |\
140         unistd.h | -POSIX+ -SUS+ +POSIX-1995+ SUSv2+

142 func |  unlink                                                  |\
143         int                                                     |\
144         const char *                                            |\
145         unistd.h | POSIX+ XPG3+
```

****
1147 Mon Apr 6 11:47:37 2015
new/usr/src/test/libc-tests/runfiles/default.run
5798 fexecve() needed per POSIX 2008
****

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

#
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright 2015 Garrett D'Amore <garrett@damore.org>
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
#

[DEFAULT]
pre =
verbose = False
quiet = False
timeout = 60
post =
outputdir = /var/tmp/test_results

[/opt/libc-tests/tests/fexecve_test]

[/opt/libc-tests/tests/fpround_test]

[/opt/libc-tests/tests/newlocale_test]

[/opt/libc-tests/tests/nl_langinfo_test]

[/opt/libc-tests/tests/wcsrtombs_test]

[/opt/libc-tests/tests/wctype_test]

[/opt/libc-tests/tests/symbols]
pre = setup
tests = [
        'ctype_h',
        'dirent_h',
        'fcntl_h',
        'locale_h',
        'math_h',
        'netdb_h',
        'pthread_h',
        'signal_h',
        'stdio_h',
        'stdlib_h',
        'strings_h',
        'sys_stat_h',
        'sys_time_h',
        'sys_timeb_h',
        'ucontext_h',
        'unistd_h',
        'wchar_h',
        'wctype_h'
         ]
```

```
**********************************************************
     654 Mon Apr  6 11:47:38 2015
new/usr/src/test/libc-tests/tests/Makefile
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

  12 #
  13 # Copyright (c) 2012 by Delphix. All rights reserved.
  14 # Copyright 2015 Garrett D'Amore <garrett@damore.org>
  15 #

  17 SUBDIRS = fexecve fpround newlocale nl_langinfo symbols wcsrtombs wctype
  17 SUBDIRS = fpround newlocale nl_langinfo symbols wcsrtombs wctype

  19 include $(SRC)/Makefile.master
  20 include $(SRC)/test/Makefile.com
```

```
**********************************************************
     563 Mon Apr  6 11:47:38 2015
new/usr/src/test/libc-tests/tests/fexecve/Makefile
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

  12 #
  13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
  14 #

  16 include $(SRC)/Makefile.master

  18 PROG = fexecve_test
  19 ARCHPROG = fexecve_test

  21 include ../Makefile.com
```

```
**********************************************************
    4441 Mon Apr  6 11:47:39 2015
new/usr/src/test/libc-tests/tests/fexecve/fexecve_test.c
5798 fexecve() needed per POSIX 2008
**********************************************************
```

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

/*
 * Copyright 2015 Garrett D'Amore <garrett@damore.org>
 */

/*
 * This program tests that fexecve works properly.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <err.h>
#include <errno.h>
#include <unistd.h>
#include <note.h>
#include <sys/utsname.h>
#include <sys/wait.h>
#include "test_common.h"

int extra_debug = 0;

struct utsname un;

void
forkit(char *msg, const char *expect, void (*postfn)(void))
{
        int fd;
        FILE *f;
        pid_t pid;
        char *ptr = NULL;
        size_t cap = 0;
        int wstat;
        int rv;
        test_t t;
        char fname[32];

        (void) strcpy(fname, "/tmp/testXXXXXX");
        t = test_start(msg);

        fd = mkstemp(fname);
        if (fd < 0) {
                test_failed(t, "mkstemp failed: %s", strerror(errno));
                return;
        }

        /* don't leave it in the filesystem */
        (void) unlink(fname);

        pid = vfork();
```



```
        switch (pid) {
        case -1:
                test_failed(t, "vfork failed: %s", strerror(errno));
                return;
        case 0:
                if (dup2(fd, 1) < 0) {
                        test_failed(t, "dup2 failed: %s", strerror(errno));
                        exit(9);
                }
                postfn();
                exit(0);
        default:
                break;
        }

        /* parent */
        f = fdopen(fd, "r");
        if (f == NULL) {
                (void) close(fd);
                test_failed(t, "fdopen failed: %s", strerror(errno));
                (void) wait(NULL);
                return;
        }
        if (waitpid(pid, &wstat, NULL) < 0) {
                test_failed(t, "wait failed: %s", strerror(errno));
                (void) fclose(f);
                return;
        }
        if (!WIFEXITED(wstat) || WEXITSTATUS(wstat) != 0) {
                test_failed(t, "child failed: %#x", wstat);
                (void) fclose(f);
                return;
        }
        (void) lseek(fd, 0, SEEK_SET);
        if ((rv = getline(&ptr, &cap, f)) < 1) {
                test_failed(t, "child gave no data: %d", rv);
                (void) fclose(f);
                return;
        }
        (void) fclose(f);

        if (strncmp(ptr, expect, strlen(expect)) != 0) {
                test_failed(t, "%s != %s", ptr, expect);
                return;
        }

        (void) free(ptr);
        test_passed(t);
}

void
case_badf(void)
{
        int fd = -1;
        int rv;
        char *args[] = { "uname", NULL };
        char *env[] = { NULL };

        rv = fexecve(fd, args, env);
        if (rv != -1) {
                (void) printf("rv is not -1\n");
                (void) exit(0);
        }
        if (errno != EBADF) {
                (void) printf("err %d(%s) != EBADF\n", errno, strerror(errno));
                (void) exit(0);
```

```
128         }
129         (void) printf("GOOD\n");
130         (void) exit(0);
131 }

133 void
134 case_notexec(void)
135 {
136         int fd;
137         int rv;
138         char *args[] = { "uname", NULL };
139         char *env[] = { NULL };

141         fd = open("/usr/bin/uname", O_RDONLY);

143         rv = fexecve(fd, args, env);
144         if (rv != -1) {
145                 (void) printf("rv is not -1\n");
146                 (void) exit(0);
147         }
148         (void) printf("FAILURE\n");
149         (void) exit(0);
150 }

152 void
153 case_uname(void)
154 {
155         int fd;
156         char *args[] = { "uname", NULL };
157         char *env[] = { NULL };

159         fd = open("/usr/bin/uname", O_EXEC);
160         if (fd < 0) {
161                 (void) printf("failed to open /usr/bin/uname: %s",
162                     strerror(errno));
163                 (void) exit(0);
164         }

166         (void) fexecve(fd, args, env);
167         (void) printf("EXEC FAILED: %s\n", strerror(errno));
168         (void) exit(0);
169 }

171 void
172 case_uname_r(void)
173 {
174         int fd;
175         char *args[] = { "uname", "-r", NULL };
176         char *env[] = { NULL };

178         fd = open("/usr/bin/uname", O_EXEC);
179         if (fd < 0) {
180                 (void) printf("failed to open /usr/bin/uname: %s",
181                     strerror(errno));
182                 (void) exit(0);
183         }

185         (void) fexecve(fd, args, env);
186         (void) printf("EXEC FAILED: %s\n", strerror(errno));
187         (void) exit(0);
188 }

190 void
191 test_fexecve_badf(void)
192 {
193         forkit("fexecve (bad FD)", "GOOD\n", case_badf);
```



```
194 }

196 void
197 test_fexecve_notexec(void)
198 {
199         forkit("fexecve (not O_EXEC)", un.sysname, case_notexec);
200 }

202 void
203 test_fexecve_uname(void)
204 {
205         forkit("fexecve (uname)", un.sysname, case_uname);
206 }

208 void
209 test_fexecve_uname_r(void)
210 {
211         forkit("fexecve (uname)", un.release, case_uname_r);
212 }

214 int
215 main(int argc, char **argv)
216 {
217         int optc;

219         (void) uname(&un);

221         while ((optc = getopt(argc, argv, "dfD")) != EOF) {
222                 switch (optc) {
223                 case 'd':
224                         test_set_debug();
225                         break;
226                 case 'f':
227                         test_set_force();
228                         break;
229                 case 'D':
230                         test_set_debug();
231                         extra_debug++;
232                         break;
233                 default:
234                         (void) fprintf(stderr, "Usage: %s [-dfD]\n", argv[0]);
235                         exit(1);
236                 }
237         }

239         test_fexecve_badf();
240         test_fexecve_notexec();
241         test_fexecve_uname();
242         test_fexecve_uname_r();

244         exit(0);
245 }
```

```
**********************************************************
   52795 Mon Apr  6 11:47:39 2015
new/usr/src/uts/common/os/exec.c
5798 fexecve() needed per POSIX 2008
**********************************************************
```

```
/*
 * 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 2015 Garrett D'Amore <garrett@damore.org>
 * Copyright (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
 */

/*      Copyright (c) 1988 AT&T */
/*        All Rights Reserved   */
/*
 * Copyright 2014, Joyent, Inc.  All rights reserved.
 */

#include <sys/types.h>
#include <sys/param.h>
#include <sys/sysmacros.h>
#include <sys/systm.h>
#include <sys/signal.h>
#include <sys/cred_impl.h>
#include <sys/policy.h>
#include <sys/user.h>
#include <sys/errno.h>
#include <sys/file.h>
#include <sys/vfs.h>
#include <sys/vnode.h>
#include <sys/mman.h>
#include <sys/acct.h>
#include <sys/cpuvar.h>
#include <sys/proc.h>
#include <sys/cmn_err.h>
#include <sys/debug.h>
#include <sys/pathname.h>
#include <sys/vm.h>
#include <sys/lgrp.h>
#include <sys/vtrace.h>
#include <sys/exec.h>
#include <sys/exechdr.h>
#include <sys/kmem.h>
#include <sys/prsystm.h>
#include <sys/modctl.h>
#include <sys/vmparam.h>
#include <sys/door.h>
```



```
#include <sys/schedctl.h>
#include <sys/utrap.h>
#include <sys/systeminfo.h>
#include <sys/stack.h>
#include <sys/rctl.h>
#include <sys/dtrace.h>
#include <sys/lwpchan_impl.h>
#include <sys/pool.h>
#include <sys/sdt.h>
#include <sys/brand.h>
#include <sys/klpd.h>

#include <c2/audit.h>

#include <vm/hat.h>
#include <vm/anon.h>
#include <vm/as.h>
#include <vm/seg.h>
#include <vm/seg_vn.h>

#define PRIV_RESET              0x01    /* needs to reset privs */
#define PRIV_SETID              0x02    /* needs to change uids */
#define PRIV_SETUGID            0x04    /* is setuid/setgid/forced privs */
#define PRIV_INCREASE           0x08    /* child runs with more privs */
#define MAC_FLAGS               0x10    /* need to adjust MAC flags */
#define PRIV_FORCED             0x20    /* has forced privileges */

static int execsetid(struct vnode *, struct vattr *, uid_t *, uid_t *,
    priv_set_t *, cred_t *, const char *);
static int hold_execsw(struct execsw *);

uint_t auxv_hwcap = 0;  /* auxv AT_SUN_HWCAP value; determined on the fly */
uint_t auxv_hwcap_2 = 0;        /* AT_SUN_HWCAP2 */
#if defined(_SYSCALL32_IMPL)
uint_t auxv_hwcap32 = 0;        /* 32-bit version of auxv_hwcap */
uint_t auxv_hwcap32_2 = 0;      /* 32-bit version of auxv_hwcap2 */
#endif

#define PSUIDFLAGS              (SNOCD|SUGID)

#define DEVFD                   "/dev/fd/"

/*
 * exece() - system call wrapper around exec_common()
 */
int
exece(const char *fname, const char **argp, const char **envp)
{
        int error;

        error = exec_common(fname, argp, envp, EBA_NONE);
        return (error ? (set_errno(error)) : 0);
}

int
exec_common(const char *fname, const char **argp, const char **envp,
    int brand_action)
{
        vnode_t *vp = NULL, *dir = NULL, *tmpvp = NULL;
        proc_t *p = ttoproc(curthread);
        klwp_t *lwp = ttolwp(curthread);
        struct user *up = PTOU(p);
        long execsz;            /* temporary count of exec size */
        int i;
        int error;
        char exec_file[MAXCOMLEN+1];
```

```
128         struct pathname pn;
129         struct pathname resolvepn;
130         struct uarg args;
131         struct execa ua;
132         k_sigset_t savedmask;
133         lwpdir_t *lwpdir = NULL;
134         tidhash_t *tidhash;
135         lwpdir_t *old_lwpdir = NULL;
136         uint_t old_lwpdir_sz;
137         tidhash_t *old_tidhash;
138         uint_t old_tidhash_sz;
139         ret_tidhash_t *ret_tidhash;
140         lwpent_t *lep;
141         boolean_t brandme = B_FALSE;

143         /*
144          * exec() is not supported for the /proc agent lwp.
145          */
146         if (curthread == p->p_agenttp)
147                 return (ENOTSUP);

149         if (brand_action != EBA_NONE) {
150                 /*
151                  * Brand actions are not supported for processes that are not
152                  * running in a branded zone.
153                  */
154                 if (!ZONE_IS_BRANDED(p->p_zone))
155                         return (ENOTSUP);

157                 if (brand_action == EBA_NATIVE) {
158                         /* Only branded processes can be unbranded */
159                         if (!PROC_IS_BRANDED(p))
160                                 return (ENOTSUP);
161                 } else {
162                         /* Only unbranded processes can be branded */
163                         if (PROC_IS_BRANDED(p))
164                                 return (ENOTSUP);
165                         brandme = B_TRUE;
166                 }
167         } else {
168                 /*
169                  * If this is a native zone, or if the process is already
170                  * branded, then we don't need to do anything.  If this is
171                  * a native process in a branded zone, we need to brand the
172                  * process as it exec()s the new binary.
173                  */
174                 if (ZONE_IS_BRANDED(p->p_zone) && !PROC_IS_BRANDED(p))
175                         brandme = B_TRUE;
176         }

178         /*
179          * Inform /proc that an exec() has started.
180          * Hold signals that are ignored by default so that we will
181          * not be interrupted by a signal that will be ignored after
182          * successful completion of gexec().
183          */
184         mutex_enter(&p->p_lock);
185         prexecstart();
186         schedctl_finish_sigblock(curthread);
187         savedmask = curthread->t_hold;
188         sigorset(&curthread->t_hold, &ignoredefault);
189         mutex_exit(&p->p_lock);

191         /*
192          * Look up path name and remember last component for later.
193          * To help coreadm expand its %d token, we attempt to save
```



```
194          * the directory containing the executable in p_execdir. The
195          * first call to lookuppn() may fail and return EINVAL because
196          * dirvpp is non-NULL. In that case, we make a second call to
197          * lookuppn() with dirvpp set to NULL; p_execdir will be NULL,
198          * but coreadm is allowed to expand %d to the empty string and
199          * there are other cases in which that failure may occur.
200          */
201         if ((error = pn_get((char *)fname, UIO_USERSPACE, &pn)) != 0)
202                 goto out;
203         pn_alloc(&resolvepn);

205         if (strncmp(pn.pn_path, DEVFD, strlen(DEVFD)) == 0) {
206                 /* looks like a /dev/fd node */
207                 char *p = pn.pn_path + strlen(DEVFD);
208                 int fd = stoi(&p);
209                 if ((fd < 0) || (*p != 0) || (p == pn.pn_path)) {
210                         error = EBADF;
211                         goto out;
212                 }
213                 if ((error = fgetstartvp(fd, NULL, &vp)) != 0) {
214                         goto out;       /* error will be EBADF */
215                 }
216                 (void) pn_set(&resolvepn, pn.pn_path);

218         } else if ((error =
219             lookuppn(&pn, &resolvepn, FOLLOW, &dir, &vp)) != 0) {
201         if ((error = lookuppn(&pn, &resolvepn, FOLLOW, &dir, &vp)) != 0) {
220                 pn_free(&resolvepn);
221                 pn_free(&pn);
222                 if (error != EINVAL)
223                         goto out;

225                 dir = NULL;
226                 if ((error = pn_get((char *)fname, UIO_USERSPACE, &pn)) != 0)
227                         goto out;
228                 pn_alloc(&resolvepn);
229                 if ((error = lookuppn(&pn, &resolvepn, FOLLOW, NULLVPP,
230                     &vp)) != 0) {
231                         pn_free(&resolvepn);
232                         pn_free(&pn);
233                         goto out;
234                 }
235         }
236         if (vp == NULL) {
237                 if (dir != NULL)
238                         VN_RELE(dir);
239                 error = ENOENT;
240                 pn_free(&resolvepn);
241                 pn_free(&pn);
242                 goto out;
243         }

245         if ((error = secpolicy_basic_exec(CRED(), vp)) != 0) {
246                 if (dir != NULL)
247                         VN_RELE(dir);
248                 pn_free(&resolvepn);
249                 pn_free(&pn);
250                 VN_RELE(vp);
251                 goto out;
252         }

254         /*
255          * We do not allow executing files in attribute directories.
256          * We test this by determining whether the resolved path
257          * contains a "/" when we're in an attribute directory;
258          * only if the pathname does not contain a "/" the resolved path
```

```
259          * points to a file in the current working (attribute) directory.
260          */
261         if ((p->p_user.u_cdir->v_flag & V_XATTRDIR) != 0 &&
262             strchr(resolvepn.pn_path, '/') == NULL) {
263                 if (dir != NULL)
264                         VN_RELE(dir);
265                 error = EACCES;
266                 pn_free(&resolvepn);
267                 pn_free(&pn);
268                 VN_RELE(vp);
269                 goto out;
270         }

272         bzero(exec_file, MAXCOMLEN+1);
273         (void) strncpy(exec_file, pn.pn_path, MAXCOMLEN);
274         bzero(&args, sizeof (args));
275         args.pathname = resolvepn.pn_path;
276         /* don't free resolvepn until we are done with args */
277         pn_free(&pn);

279         /*
280          * If we're running in a profile shell, then call pfexecd.
281          */
282         if ((CR_FLAGS(p->p_cred) & PRIV_PFEXEC) != 0) {
283                 error = pfexec_call(p->p_cred, &resolvepn, &args.pfcred,
284                     &args.scrubenv);

286                 /* Returning errno in case we're not allowed to execute. */
287                 if (error > 0) {
288                         if (dir != NULL)
289                                 VN_RELE(dir);
290                         pn_free(&resolvepn);
291                         VN_RELE(vp);
292                         goto out;
293                 }

295                 /* Don't change the credentials when using old ptrace. */
296                 if (args.pfcred != NULL &&
297                     (p->p_proc_flag & P_PR_PTRACE) != 0) {
298                         crfree(args.pfcred);
299                         args.pfcred = NULL;
300                         args.scrubenv = B_FALSE;
301                 }
302         }

304         /*
305          * Specific exec handlers, or policies determined via
306          * /etc/system may override the historical default.
307          */
308         args.stk_prot = PROT_ZFOD;
309         args.dat_prot = PROT_ZFOD;

311         CPU_STATS_ADD_K(sys, sysexec, 1);
312         DTRACE_PROC1(exec, char *, args.pathname);

314         ua.fname = fname;
315         ua.argp = argp;
316         ua.envp = envp;

318         /* If necessary, brand this process before we start the exec. */
319         if (brandme)
320                 brand_setbrand(p);

322         if ((error = gexec(&vp, &ua, &args, NULL, 0, &execsz,
323             exec_file, p->p_cred, brand_action)) != 0) {
324                 if (brandme)
```



```
325                         brand_clearbrand(p, B_FALSE);
326                 VN_RELE(vp);
327                 if (dir != NULL)
328                         VN_RELE(dir);
329                 pn_free(&resolvepn);
330                 goto fail;
331         }

333         /*
334          * Free floating point registers (sun4u only)
335          */
336         ASSERT(lwp != NULL);
337         lwp_freeregs(lwp, 1);

339         /*
340          * Free thread and process context ops.
341          */
342         if (curthread->t_ctx)
343                 freectx(curthread, 1);
344         if (p->p_pctx)
345                 freepctx(p, 1);

347         /*
348          * Remember file name for accounting; clear any cached DTrace predicate.
349          */
350         up->u_acflag &= ~AFORK;
351         bcopy(exec_file, up->u_comm, MAXCOMLEN+1);
352         curthread->t_predcache = NULL;

354         /*
355          * Clear contract template state
356          */
357         lwp_ctmpl_clear(lwp);

359         /*
360          * Save the directory in which we found the executable for expanding
361          * the %d token used in core file patterns.
362          */
363         mutex_enter(&p->p_lock);
364         tmpvp = p->p_execdir;
365         p->p_execdir = dir;
366         if (p->p_execdir != NULL)
367                 VN_HOLD(p->p_execdir);
368         mutex_exit(&p->p_lock);

370         if (tmpvp != NULL)
371                 VN_RELE(tmpvp);

373         /*
374          * Reset stack state to the user stack, clear set of signals
375          * caught on the signal stack, and reset list of signals that
376          * restart system calls; the new program's environment should
377          * not be affected by detritus from the old program.  Any
378          * pending held signals remain held, so don't clear t_hold.
379          */
380         mutex_enter(&p->p_lock);
381         lwp->lwp_oldcontext = 0;
382         lwp->lwp_ustack = 0;
383         lwp->lwp_old_stk_ctl = 0;
384         sigemptyset(&up->u_signodefer);
385         sigemptyset(&up->u_sigonstack);
386         sigemptyset(&up->u_sigresethand);
387         lwp->lwp_sigaltstack.ss_sp = 0;
388         lwp->lwp_sigaltstack.ss_size = 0;
389         lwp->lwp_sigaltstack.ss_flags = SS_DISABLE;
```

```
391         /*
392          * Make saved resource limit == current resource limit.
393          */
394         for (i = 0; i < RLIM_NLIMITS; i++) {
395                 /*CONSTCOND*/
396                 if (RLIM_SAVED(i)) {
397                         (void) rctl_rlimit_get(rctlproc_legacy[i], p,
398                             &up->u_saved_rlimit[i]);
399                 }
400         }

402         /*
403          * If the action was to catch the signal, then the action
404          * must be reset to SIG_DFL.
405          */
406         sigdefault(p);
407         p->p_flag &= ~(SNOWAIT|SJCTL);
408         p->p_flag |= (SEXECED|SMSACCT|SMSFORK);
409         up->u_signal[SIGCLD - 1] = SIG_DFL;

411         /*
412          * Delete the dot4 sigqueues/signotifies.
413          */
414         sigqfree(p);

416         mutex_exit(&p->p_lock);

418         mutex_enter(&p->p_pflock);
419         p->p_prof.pr_base = NULL;
420         p->p_prof.pr_size = 0;
421         p->p_prof.pr_off = 0;
422         p->p_prof.pr_scale = 0;
423         p->p_prof.pr_samples = 0;
424         mutex_exit(&p->p_pflock);

426         ASSERT(curthread->t_schedctl == NULL);

428 #if defined(__sparc)
429         if (p->p_utraps != NULL)
430                 utrap_free(p);
431 #endif  /* __sparc */

433         /*
434          * Close all close-on-exec files.
435          */
436         close_exec(P_FINFO(p));
437         TRACE_2(TR_FAC_PROC, TR_PROC_EXEC, "proc_exec:p %p up %p", p, up);

439         /* Unbrand ourself if necessary. */
440         if (PROC_IS_BRANDED(p) && (brand_action == EBA_NATIVE))
441                 brand_clearbrand(p, B_FALSE);

443         setregs(&args);

445         /* Mark this as an executable vnode */
446         mutex_enter(&vp->v_lock);
447         vp->v_flag |= VVMEXEC;
448         mutex_exit(&vp->v_lock);

450         VN_RELE(vp);
451         if (dir != NULL)
452                 VN_RELE(dir);
453         pn_free(&resolvepn);

455         /*
456          * Allocate a new lwp directory and lwpid hash table if necessary.
```



```
457          */
458         if (curthread->t_tid != 1 || p->p_lwpdir_sz != 2) {
459                 lwpdir = kmem_zalloc(2 * sizeof (lwpdir_t), KM_SLEEP);
460                 lwpdir->ld_next = lwpdir + 1;
461                 tidhash = kmem_zalloc(2 * sizeof (tidhash_t), KM_SLEEP);
462                 if (p->p_lwpdir != NULL)
463                         lep = p->p_lwpdir[curthread->t_dslot].ld_entry;
464                 else
465                         lep = kmem_zalloc(sizeof (*lep), KM_SLEEP);
466         }

468         if (PROC_IS_BRANDED(p))
469                 BROP(p)->b_exec();

471         mutex_enter(&p->p_lock);
472         prbarrier(p);

474         /*
475          * Reset lwp id to the default value of 1.
476          * This is a single-threaded process now
477          * and lwp #1 is lwp_wait()able by default.
478          * The t_unpark flag should not be inherited.
479          */
480         ASSERT(p->p_lwpcnt == 1 && p->p_zombcnt == 0);
481         curthread->t_tid = 1;
482         kpreempt_disable();
483         ASSERT(curthread->t_lpl != NULL);
484         p->p_t1_lgrpid = curthread->t_lpl->lpl_lgrpid;
485         kpreempt_enable();
486         if (p->p_tr_lgrpid != LGRP_NONE && p->p_tr_lgrpid != p->p_t1_lgrpid) {
487                 lgrp_update_trthr_migrations(1);
488         }
489         curthread->t_unpark = 0;
490         curthread->t_proc_flag |= TP_TWAIT;
491         curthread->t_proc_flag &= ~TP_DAEMON;   /* daemons shouldn't exec */
492         p->p_lwpdaemon = 0;                     /* but oh well ... */
493         p->p_lwpid = 1;

495         /*
496          * Install the newly-allocated lwp directory and lwpid hash table
497          * and insert the current thread into the new hash table.
498          */
499         if (lwpdir != NULL) {
500                 old_lwpdir = p->p_lwpdir;
501                 old_lwpdir_sz = p->p_lwpdir_sz;
502                 old_tidhash = p->p_tidhash;
503                 old_tidhash_sz = p->p_tidhash_sz;
504                 p->p_lwpdir = p->p_lwpfree = lwpdir;
505                 p->p_lwpdir_sz = 2;
506                 lep->le_thread = curthread;
507                 lep->le_lwpid = curthread->t_tid;
508                 lep->le_start = curthread->t_start;
509                 lwp_hash_in(p, lep, tidhash, 2, 0);
510                 p->p_tidhash = tidhash;
511                 p->p_tidhash_sz = 2;
512         }
513         ret_tidhash = p->p_ret_tidhash;
514         p->p_ret_tidhash = NULL;

516         /*
517          * Restore the saved signal mask and
518          * inform /proc that the exec() has finished.
519          */
520         curthread->t_hold = savedmask;
521         prexecend();
522         mutex_exit(&p->p_lock);
```

```
523         if (old_lwpdir) {
524                 kmem_free(old_lwpdir, old_lwpdir_sz * sizeof (lwpdir_t));
525                 kmem_free(old_tidhash, old_tidhash_sz * sizeof (tidhash_t));
526         }
527         while (ret_tidhash != NULL) {
528                 ret_tidhash_t *next = ret_tidhash->rth_next;
529                 kmem_free(ret_tidhash->rth_tidhash,
530                     ret_tidhash->rth_tidhash_sz * sizeof (tidhash_t));
531                 kmem_free(ret_tidhash, sizeof (*ret_tidhash));
532                 ret_tidhash = next;
533         }

535         ASSERT(error == 0);
536         DTRACE_PROC(exec__success);
537         return (0);

539 fail:
540         DTRACE_PROC1(exec__failure, int, error);
541 out:            /* error return */
542         mutex_enter(&p->p_lock);
543         curthread->t_hold = savedmask;
544         prexecend();
545         mutex_exit(&p->p_lock);
546         ASSERT(error != 0);
547         return (error);
548 }
```

*unchanged_portion_omitted_*