

```

*****
26574 Sat Sep 28 12:17:11 2013
new/usr/src/grub/grub-0.97/grub/asmstub.c
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
unchanged_portion_omitted

1102 /* Initialize a serial device. In the grub shell, PORT is unused. */
1103 int
1104 serial_hw_init (unsigned short port, unsigned int speed,
1105                int word_len, int parity, int stop_bit_len)
1106 {
1107     struct termios termios;
1108     speed_t termios_speed;
1109     int i;
1110
1111     /* Check if the file name is specified. */
1112     if (! serial_device)
1113         return 0;
1114
1115     /* If a serial device is already opened, close it first. */
1116     if (serial_fd >= 0)
1117         close (serial_fd);
1118
1119     /* Open the device file. */
1120     serial_fd = open (serial_device,
1121                    O_RDWR | O_NOCTTY
1122 #if defined(O_SYNC)
1123                    /* O_SYNC is used in Linux (and some others?). */
1124                    | O_SYNC
1125 #elif defined(O_FSYNC)
1126                    /* O_FSYNC is used in FreeBSD. */
1127                    | O_FSYNC
1128 #endif
1129                    );
1130     if (serial_fd < 0)
1131         return 0;
1132
1133     /* Get the termios parameters. */
1134     if (tcgetattr (serial_fd, &termios))
1135         goto fail;
1136
1137     /* Raw mode. */
1138 #if defined(__sun)
1139     termios.c_iflag &= ~(IGNBRK|BRKINT|PARMRK|ISTRIP|INLCR|IGNCR|ICRNL|IXON);
1140     termios.c_oflag &= ~OPOST;
1141     termios.c_lflag &= ~(ECHO|ECHONL|ICANON|ISIG|IEXTEN);
1142     termios.c_cflag &= ~(CSIZE|PARENB);
1143     termios.c_cflag |= CS8;
1144 #else
1145     cfmakeraw (&termios);
1146 #endif
1147
1148     /* Set the speed. */
1149     termios_speed = get_termios_speed (speed);
1150     if (termios_speed == B0)
1151         goto fail;
1152
1153     cfsetispeed (&termios, termios_speed);
1154     cfsetospeed (&termios, termios_speed);
1155
1156     /* Set the word length. */
1157     termios.c_cflag &= ~CSIZE;
1158     switch (word_len)

```

```

1151     {
1152     case UART_5BITS_WORD:
1153         termios.c_cflag |= CS5;
1154         break;
1155     case UART_6BITS_WORD:
1156         termios.c_cflag |= CS6;
1157         break;
1158     case UART_7BITS_WORD:
1159         termios.c_cflag |= CS7;
1160         break;
1161     case UART_8BITS_WORD:
1162         termios.c_cflag |= CS8;
1163         break;
1164     default:
1165         goto fail;
1166     }
1167
1168     /* Set the parity. */
1169     switch (parity)
1170     {
1171     case UART_NO_PARITY:
1172         termios.c_cflag &= ~PARENB;
1173         break;
1174     case UART_ODD_PARITY:
1175         termios.c_cflag |= PARENB;
1176         termios.c_cflag |= PARODD;
1177         break;
1178     case UART_EVEN_PARITY:
1179         termios.c_cflag |= PARENB;
1180         termios.c_cflag &= ~PARODD;
1181         break;
1182     default:
1183         goto fail;
1184     }
1185
1186     /* Set the length of stop bit. */
1187     switch (stop_bit_len)
1188     {
1189     case UART_1_STOP_BIT:
1190         termios.c_cflag &= ~CSTOPB;
1191         break;
1192     case UART_2_STOP_BITS:
1193         termios.c_cflag |= CSTOPB;
1194         break;
1195     default:
1196         goto fail;
1197     }
1198
1199     /* Set the parameters. */
1200     if (tcsetattr (serial_fd, TCSANOW, &termios))
1201         goto fail;
1202
1203 #ifdef SIMULATE_SLOWNESS_OF_SERIAL
1204     serial_speed = speed;
1205 #endif /* SIMULATE_SLOWNESS_OF_SERIAL */
1206
1207     /* Get rid of the flag TERM_NEED_INIT from the serial terminal. */
1208     for (i = 0; term_table[i].name; i++)
1209     {
1210         if (strcmp (term_table[i].name, "serial") == 0)
1211         {
1212             term_table[i].flags &= ~(TERM_NEED_INIT);
1213             break;
1214         }
1215     }
1216

```

`new/usr/src/grub/grub-0.97/grub/asmstub.c`

3

```
1217 return 1;
```

```
1219 fail:
1220 close (serial_fd);
1221 serial_fd = -1;
1222 return 0;
1223 }
```

unchanged\_portion\_omitted

```

*****
22425 Sat Sep 28 12:17:12 2013
new/usr/src/lib/libc/amd64/Makefile
1060 terminos missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #

29 LIBCBASE= .
30 LIBCDIR= $(SRC)/lib/libc
31 LIBRARY= libc.a
32 LIB_PIC= libc_pic.a
33 VERS= .1
34 CPP= /usr/lib/cpp
35 TARGET_ARCH= amd64

37 # objects are grouped by source directory

39 # local objects
40 STRETS=

42 CRTOBSJ= \
43 cerror.o

45 DYNOBJS=

47 FPOBJS= \
48 _base_il.o \
49 fpgetmask.o \
50 fpgetround.o \
51 fpsetmask.o \
52 fpsetround.o \
53 fpstart.o

55 I386FPOBJS= \
56 _D_cplx_div.o \
57 _D_cplx_div_ix.o \
58 _D_cplx_div_rx.o \
59 _D_cplx_lr_div.o

```

```

60 _D_cplx_lr_div_ix.o \
61 _D_cplx_lr_div_rx.o \
62 _D_cplx_mul.o \
63 _F_cplx_div.o \
64 _F_cplx_div_ix.o \
65 _F_cplx_div_rx.o \
66 _F_cplx_lr_div.o \
67 _F_cplx_lr_div_ix.o \
68 _F_cplx_lr_div_rx.o \
69 _F_cplx_mul.o \
70 _X_cplx_div.o \
71 _X_cplx_div_ix.o \
72 _X_cplx_div_rx.o \
73 _X_cplx_lr_div.o \
74 _X_cplx_lr_div_ix.o \
75 _X_cplx_lr_div_rx.o \
76 _X_cplx_mul.o

78 FPASMOBJS= \
79 __xgetRD.o \
80 _xtoll.o \
81 _xtoull.o \
82 fpcw.o \
83 fpgetsticky.o \
84 fpsetsticky.o

86 ATOMICOBJS= \
87 atomic.o

89 XATTROBJS= \
90 xattr_common.o

91 COMOBJS= \
92 bcmp.o \
93 bcopy.o \
94 bsearch.o \
95 bzero.o \
96 qsort.o \
97 strtol.o \
98 strtoul.o \
99 strtoll.o \
100 strtoull.o

102 GENOBJS= \
103 _getsp.o \
104 abs.o \
105 alloca.o \
106 attrat.o \
107 byteorder.o \
108 cuexit.o \
109 ecvt.o \
110 errlst.o \
111 amd64_data.o \
112 ldivide.o \
113 lock.o \
114 makeectxt.o \
115 memccpy.o \
116 memchr.o \
117 memcmp.o \
118 memcpy.o \
119 memset.o \
120 new_list.o \
121 proc64_id.o \
122 proc64_support.o \
123 setjmp.o \
124 siginfolst.o \
125 siglongjmp.o

```

new/usr/src/lib/libc/amd64/Makefile

3

```
126      strcmp.o      \  
127      strcpy.o      \  
128      strlen.o      \  
129      strncmp.o     \  
130      strncpy.o     \  
131      strnlen.o     \  
132      sync_instruction_memory.o
```

```
134 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds  
135 # This macro should ALWAYS be empty; native APIs are already 'large file'.  
136 COMSYSOBS64=
```

```
138 SYSOBS64=
```

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

new/usr/src/lib/libc/amd64/Makefile

4

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

## new/usr/src/lib/libc/amd64/Makefile

```

258     times.o           \
259     ulimit.o          \
260     umask.o           \
261     umount2.o         \
262     utssys.o          \
263     uucopy.o          \
264     vhangup.o         \
265     waitid.o          \
266     write.o           \
267     writev.o          \
268     yield.o           \

270 SYSOBJS=
271     __clock_gettime.o \
272     __getcontext.o   \
273     __uadmin.o        \
274     _lwp_mutex_unlock.o \
275     _stack_grow.o    \
276     door.o           \
277     forkx.o           \
278     forkallx.o        \
279     getcontext.o      \
280     gettimeofday.o   \
281     lwp_private.o     \
282     nuname.o          \
283     syscall.o         \
284     sysi86.o          \
285     tls_get_addr.o   \
286     uadmin.o          \
287     umount.o          \
288     uname.o           \
289     vforkx.o          \

```

```

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

```

```

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

```

5

## new/usr/src/lib/libc/amd64/Makefile

```

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

```

6

```

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

```

```

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

```

```

522      strerror.o      \
523      strlcat.o       \
524      strlcpy.o       \
525      strncat.o       \
526      strndup.o       \
527      strpbrk.o       \
528      strrchr.o       \
529      strsep.o        \
530      strsignal.o     \
531      strspn.o        \
532      strstr.o        \
533      strtod.o        \
534      strtointmax.o   \
535      strtok.o        \
536      strtok_r.o     \
537      strtoumax.o    \
538      swab.o          \
539      swapctl.o       \
540      sysconf.o       \
541      syslog.o        \
542      tcdrain.o       \
543      tcflow.o        \
544      tcflush.o       \
545      tcgetattr.o     \
546      tcgetpgrp.o     \
547      tcgetsid.o      \
548      tcsendbreak.o   \
549      tcsetattr.o     \
550      tcsetpgrp.o     \
551      tell.o          \
552      telldir.o       \
553      tfind.o         \
554      time_data.o     \
555      time_gdata.o    \
556      tls_data.o      \
557      truncate.o      \
558      tsdalloc.o      \
559      tsearch.o       \
560      ttyname.o       \
561      ttyslot.o       \
562      ualarm.o        \
563      ucred.o         \
564      valloc.o        \
565      vlfmt.o         \
566      vpfmt.o         \
567      waitpid.o       \
568      walkstack.o     \
569      wdata.o         \
570      xgetwidth.o     \
571      xpg4.o          \
572      xpg6.o          \
574 PORTPRINT_W=      \
575      doprnt_w.o     \
577 PORTPRINT=        \
578      asprintf.o      \
579      doprnt.o        \
580      fprintf.o       \
581      printf.o        \
582      snprintf.o      \
583      sprintf.o       \
584      vfprintf.o      \
585      vprintf.o       \
586      vsnprintf.o    \
587      vsprintf.o     \

```

```

588      vwprintf.o      \
589      wprintf.o       \
591 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
592 # This macro should ALWAYS be empty; native APIs are already 'large file'.
593 PORTSTDIO64=
595 PORTSTDIO_W=       \
596      doscan_w.o     \
598 PORTSTDIO=        \
599      _extensions.o  \
600      _endopen.o     \
601      _filbuf.o      \
602      _findbuf.o     \
603      _flsbuf.o      \
604      _wrtchk.o      \
605      clearerr.o     \
606      ctermid.o      \
607      ctermid_r.o    \
608      cuserid.o      \
609      data.o         \
610      doscan.o       \
611      fdopen.o       \
612      feof.o         \
613      ferrror.o      \
614      fgetc.o        \
615      fgets.o        \
616      fileno.o       \
617      flockf.o       \
618      flush.o        \
619      fopen.o        \
620      fpos.o         \
621      fputc.o        \
622      fputs.o        \
623      fread.o        \
624      fseek.o        \
625      fseeko.o       \
626      ftell.o        \
627      ftello.o       \
628      fwrite.o       \
629      getc.o         \
630      getchar.o      \
631      getline.o      \
632      getpass.o      \
633      gets.o         \
634      getw.o         \
635      mse.o          \
636      popen.o        \
637      putc.o         \
638      putchar.o     \
639      puts.o         \
640      putw.o         \
641      rewind.o       \
642      scanf.o        \
643      setbuf.o       \
644      setbuffer.o    \
645      setvbuf.o      \
646      system.o       \
647      tempnam.o      \
648      tmpfile.o      \
649      tmpnam_r.o     \
650      ungetc.o       \
651      vscanf.o       \
652      vwscanf.o      \
653      wscanf.o       \

```

```

655 PORTI18N= \
656     getwchar.o \
657     putwchar.o \
658     putws.o \
659     strcasecmp.o \
660     strcasestr.o \
661     strncasecmp.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     wscasecmp.o \
675     wscat.o \
676     wschr.o \
677     wscmp.o \
678     wscpy.o \
679     wscspn.o \
680     wsdup.o \
681     wslen.o \
682     wsncasecmp.o \
683     wscat.o \
684     wscmp.o \
685     wscpy.o \
686     wspbrk.o \
687     vsprintf.o \
688     wsrchr.o \
689     wsscanf.o \
690     wssp.o \
691     wstod.o \
692     wstok.o \
693     wstol.o \
694     wstoll.o \
695     wsxfrm.o \
696     gettext.o \
697     gettext_gnu.o \
698     gettext_real.o \
699     gettext_util.o \
700     isdigit.o \
701     plural_parser.o \
702     wdresolve.o \
703     _ctype.o \
704     isascii.o \
705     toascii.o \
707 PORTI18N_COND= \
708     wcstol_longlong.o \
709     wcstoul_longlong.o \
711 PORTLOCALE= \
712     big5.o \
713     btowc.o \
714     collate.o \
715     collcmp.o \
716     euc.o \
717     fnmatch.o \
718     fgetwc.o \
719     fgetws.o \

```

```

720     fix_grouping.o \
721     fputwc.o \
722     fputws.o \
723     fwide.o \
724     gb18030.o \
725     gb2312.o \
726     gbk.o \
727     getdate.o \
728     iswctype.o \
729     ldpart.o \
730     lmessages.o \
731     lnumeric.o \
732     lmonetary.o \
733     localeconv.o \
734     mbftowc.o \
735     mblen.o \
736     mbrlen.o \
737     mbrtowc.o \
738     mbsinit.o \
739     mbsnrto wcs.o \
740     mbsrtowcs.o \
741     mbstowcs.o \
742     mbtowc.o \
743     mskanji.o \
744     nextwctype.o \
745     nl_langinfo.o \
746     none.o \
747     regcomp.o \
748     regfree.o \
749     regerror.o \
750     regexec.o \
751     rune.o \
752     runetype.o \
753     setlocale.o \
754     setrunelocale.o \
755     strcoll.o \
756     strfmon.o \
757     strftime.o \
758     strtptime.o \
759     strxfrm.o \
760     table.o \
761     timelocal.o \
762     tolower.o \
763     towlower.o \
764     ungetwc.o \
765     utf8.o \
766     wcrto mb.o \
767     wcscoll.o \
768     wcsftime.o \
769     wcsnrto mbs.o \
770     wcsrtombs.o \
771     wcswidth.o \
772     wcstombs.o \
773     wcsxfrm.o \
774     wctob.o \
775     wctomb.o \
776     wctrans.o \
777     wctype.o \
778     wcwidth.o \
779     wscol.o \
781 AIOOBS= \
782     aio.o \
783     aio_alloc.o \
784     posix_aio.o \

```



```

786 RTOBJS= \
787     clock_timer.o \
788     mqueue.o \
789     posixobj.o \
790     sched.o \
791     sem.o \
792     shm.o \
793     sigev_thread.o

795 TPOOLBJS= \
796     thread_pool.o

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

821 THREADSMACHOBJS= \
822     machdep.o

824 THREADSASMOBJS= \
825     asm_subr.o

827 UNICODOBJS= \
828     u8_textprep.o \
829     uconv.o

831 UNWINDMACHOBJS= \
832     call_frame_inst.o \
833     eh_frame.o \
834     thrp_unwind.o \
835     unwind.o

837 pics/unwind.o:= COPTFLAG64 =

839 UNWINDASMOBJS= \
840     unwind_frame.o

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

846 PORTSYS= \
847     _autofssys.o \
848     access.o \
849     acctctl.o \
850     bsd_signal.o \
851     chmod.o

```

```

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

904 PORTREGEX= \
905     glob.o \
906     regcmp.o \
907     regex.o \
908     wordexp.o

910 VALUES= \
911     values-Xa.o

913 MOSTOBJS= \
914     $(STRETS) \
915     $(CROBJS) \
916     $(DYNBJS) \
917     $(FPOBJS)

```

```

918 $(I386FPOBJS) \
919 $(FPASMOBJS) \
920 $(ATOMICOBJS) \
921 $(XATTROBJS) \
922 $(COMOBJS) \
923 $(GENOBJS) \
924 $(PORTFP) \
925 $(PORTGEN) \
926 $(PORTGEN64) \
927 $(PORTI18N) \
928 $(PORTI18N_COND) \
929 $(PORTLOCALE) \
930 $(PORTPRINT) \
931 $(PORTPRINT_W) \
932 $(PORTREGEX) \
933 $(PORTSTDIO) \
934 $(PORTSTDIO64) \
935 $(PORTSTDIO_W) \
936 $(PORTSYS) \
937 $(PORTSYS64) \
938 $(AIOOBJS) \
939 $(RTOBJS) \
940 $(TPOOLBJS) \
941 $(THREADSOBJS) \
942 $(THREADSMACHOBJS) \
943 $(THREADSASMOBJS) \
944 $(UNICODEOBJS) \
945 $(UNWINDMACHOBJS) \
946 $(UNWINDASMOBJS) \
947 $(COMSYSOBJS) \
948 $(SYSOBJS) \
949 $(COMSYSOBJS64) \
950 $(SYSOBJS64) \
951 $(VALUES)

953 TRACEOBJS= \
954 plockstat.o

956 # NOTE: libc.so.1 must be linked with the minimal crt1.o and crtn.o
957 # modules whose source is provided in the $(SRC)/lib/common directory.
958 # This must be done because otherwise the Sun C compiler would insert
959 # its own versions of these modules and those versions contain code
960 # to call out to C++ initialization functions. Such C++ initialization
961 # functions can call back into libc before thread initialization is
962 # complete and this leads to segmentation violations and other problems.
963 # Since libc contains no C++ code, linking with the minimal crt1.o and
964 # crtn.o modules is safe and avoids the problems described above.
965 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
966 CRTSRCS= ../../common/amd64

968 # include common library definitions
969 include ../../Makefile.lib
970 include ../../Makefile.lib.64

972 CFLAGS64 += $(CTF_FLAGS)

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

978 CERRWARN += _gcc=-Wno-parentheses
979 CERRWARN += _gcc=-Wno-switch
980 CERRWARN += _gcc=-Wno-uninitialized
981 CERRWARN += _gcc=-Wno-unused-value
982 CERRWARN += _gcc=-Wno-unused-label
983 CERRWARN += _gcc=-Wno-unused-variable

```

```

984 CERRWARN += _gcc=-Wno-type-limits
985 CERRWARN += _gcc=-Wno-char-subscripts
986 CERRWARN += _gcc=-Wno-clobbered
987 CERRWARN += _gcc=-Wno-unused-function
988 CERRWARN += _gcc=-Wno-address

990 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
991 # enables ASSERT() checking in the threads portion of the library.
992 # This is automatically enabled for DEBUG builds, not for non-debug builds.
993 THREAD_DEBUG =
994 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

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

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

1001 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1003 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1005 CPPFLAGS= -D_REENTRANT -D$(MACH64) -D_$(MACH64) $(THREAD_DEBUG) \
1006 -I. -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1007 ASFLAGS= $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) \
1008 $(amd64_AS_XARCH)

1010 # As a favor to the dtrace syscall provider, libc still calls the
1011 # old syscall traps that have been obsoleted by the *at() interfaces.
1012 # Delete this to compile libc using only the new *at() system call traps
1013 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1015 # proc64_id.o is built with defines in $(SRC)/uts/intel/sys/x86_archext.h
1016 pics/proc64_id.o := CFLAGS64 += -I$(SRC)/uts/intel

1018 # Inform the run-time linker about libc specialized initialization
1019 RTLDINFO = -z rtldinfo=tls_rtldinfo
1020 DYNFLAGS += $(RTLDINFO)

1022 # Force libc's internal references to be resolved immediately upon loading
1023 # in order to avoid critical region problems. Since almost all libc symbols
1024 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1025 DYNFLAGS += -znow

1027 BUILD.s= $(AS) $(ASFLAGS) $< -o $@

1029 # Override this top level flag so the compiler builds in its native
1030 # C99 mode. This has been enabled to support the complex arithmetic
1031 # added to libc.
1032 C99MODE= $(C99_ENABLE)

1034 # libc method of building an archive
1035 # The "$(GREP) -v ' L '" part is necessary only until
1036 # lorder is fixed to ignore thread-local variables.
1037 BUILD.AR= $(RM) $@ ; \
1038 $(AR) q $@ $(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR)

1040 # extra files for the clean target
1041 CLEANFILES= \
1042 $(LIBCDIR)/port/gen/errlst.c \
1043 $(LIBCDIR)/port/gen/new_list.c \
1044 assym.h \
1045 genassym \
1046 crt/_rtld.s \
1047 pics/crt1.o \
1048 pics/crtn.o \
1049 $(ALTPICS)

```

```

1051 CLOBBERFILES += $(LIB_PIC)

1053 # list of C source for lint
1054 SRCS=
1055     $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c) \
1056     $(XATROBJS:%.o=$(SRC)/common/xattr/%.c) \
1057     $(COMOBJS:%.o=$(SRC)/common/util/%.c) \
1058     $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c) \
1059     $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c) \
1060     $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c) \
1061     $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c) \
1062     $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c) \
1063     $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c) \
1064     $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c) \
1065     $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c) \
1066     $(AIOBJS:%.o=$(LIBCDIR)/port/aio/%.c) \
1067     $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c) \
1068     $(TPOOLBJS:%.o=$(LIBCDIR)/port/tpool/%.c) \
1069     $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c) \
1070     $(THREADSMACHOBJS:%.o=threads/%.c) \
1071     $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c) \
1072     $(UNWINDMACHOBJS:%.o=unwind/%.c) \
1073     $(FPOBJS:%.o=fp/%.c) \
1074     $(I386FPOBJS:%.o=$(LIBCDIR)/i386/fp/%.c) \
1075     $(LIBCBASE)/gen/ecvt.c \
1076     $(LIBCBASE)/gen/makeextxt.c \
1077     $(LIBCBASE)/gen/signfolst.c \
1078     $(LIBCBASE)/gen/siglongjmp.c \
1079     $(LIBCBASE)/gen/sync_instruction_memory.c \
1080     $(LIBCBASE)/sys/uadmin.c

1082 # conditional assignments
1083 # $(DYNLIB) $(LIB_PIC) := DYNOBJS = _rtbootld.o
1084 $(DYNLIB) := CRTI = crti.o
1085 $(DYNLIB) := CRTN = crtn.o

1087 # Files which need the threads .il inline template
1088 TIL=
1089     aio.o \
1090     alloc.o \
1091     assfail.o \
1092     atexit.o \
1093     atfork.o \
1094     cancel.o \
1095     door_calls.o \
1096     err.o \
1097     errno.o \
1098     lwp.o \
1099     ma.o \
1100     machdep.o \
1101     posix_aio.o \
1102     pthr_attr.o \
1103     pthr_barrier.o \
1104     pthr_cond.o \
1105     pthr_mutex.o \
1106     pthr_rwlock.o \
1107     pthread.o \
1108     rand.o \
1109     rwlock.o \
1110     scalls.o \
1111     sched.o \
1112     sema.o \
1113     sigaction.o \
1114     sigev_thread.o \
1115     spawn.o \

```

```

1116     stack.o \
1117     synch.o \
1118     tdb_agent.o \
1119     thr.o \
1120     thread_interface.o \
1121     thread_pool.o \
1122     thrp_unwind.o \
1123     tls.o \
1124     tsd.o

1126 $(TIL:%=pics%) := CFLAGS64 += $(LIBCBASE)/threads/amd64.il

1128 # pics/mul64.o := CFLAGS64 += crt/mul64.il

1130 # large-file-aware components that should be built large

1132 #$(COMSYSOBS64:%=pics%) := \
1133 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1135 #$(SYSOBS64:%=pics%) := \
1136 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1138 #$(PORTGEN64:%=pics%) := \
1139 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1141 #$(PORTSTDIO64:%=pics%) := \
1142 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1144 #$(PORTSYS64:%=pics%) := \
1145 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1147 $(PORTSTDIO_W:%=pics%) := \
1148 CPPFLAGS += -D_WIDE

1150 $(PORTPRINT_W:%=pics%) := \
1151 CPPFLAGS += -D_WIDE

1153 $(PORTPRINT_C89:%=pics%) := \
1154 CPPFLAGS += -D_C89_INTMAX32

1156 $(PORTSTDIO_C89:%=pics%) := \
1157 CPPFLAGS += -D_C89_INTMAX32

1159 $(PORTI18N_COND:%=pics%) := \
1160 CPPFLAGS += -D_WCS_LONGLONG

1162 .KEEP_STATE:

1164 all: $(LIBS) $(LIB_PIC)

1166 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1167 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1168 lint := LINTFLAGS64 += -mm -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1170 lint:
1171     @echo $(LINT.c) ... $(LDLIBS)
1172     @$$(LINT.c) $(SRCS) $(LDLIBS)

1174 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1175 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1176 $(LINTLIB):= LINTFLAGS64=-nvx -m64

1178 # object files that depend on inline template
1179 $(TIL:%=pics%): $(LIBCBASE)/threads/amd64.il
1180 # pics/mul64.o: crt/mul64.il

```

```
1182 # include common libc targets
1183 include ../Makefile.targ

1185 # We need to strip out all CTF data from the static library
1186 $(LIB_PIC) := DIR = pics
1187 $(LIB_PIC): pics $$$(PICS)
1188     $(BUILD.AR)
1189     $(MCS) -d -n .SUNW_ctf $$@ > /dev/null 2>&1
1190     $(AR) -ts $$@ > /dev/null
1191     $(POST_PROCESS_A)

1193 ASSYMDEP_OBJS= \
1194     _lwp_mutex_unlock.o \
1195     _stack_grow.o \
1196     asm_subr.o \
1197     getcontext.o \
1198     setjmp.o \
1199     tls_get_addr.o \
1200     vforkx.o

1202 $(ASSYMDEP_OBJS:%=pics/%) : assym.h

1204 # assym.h build rules

1206 GENASSYM_C = genassym.c

1208 genassym: $(GENASSYM_C)
1209     $(NATIVECC) -Iinc -I$(LIBCDIR)/inc $(CPPFLAGS.native) \
1210     -o $$@ $(GENASSYM_C)

1212 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1214 assym.h: $(OFFSETS) genassym
1215     $(OFFSETS_CREATE) <$(OFFSETS) >$$@
1216     ./genassym >>$$@

1218 # derived C source and related explicit dependencies
1219 $(LIBCDIR)/port/gen/errlst.c + \
1220 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1221     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1223 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1225 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```

*****
23910 Sat Sep 28 12:17:12 2013
new/usr/src/lib/libc/i386/Makefile.com
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
24 #
25 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #

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

35 VALUES=     values-Xa.o

37 # objects are grouped by source directory

39 # local objects
40 STRETS=

42 CRTOBSJ=     \
43      cerror.o \
44      cerror64.o

46 DYNOBJS=     \
47      _rtbootld.o

49 FPOBJS=      \
50      _D_cplx_div.o \
51      _D_cplx_div_ix.o \
52      _D_cplx_div_rx.o \
53      _D_cplx_lr_div.o \
54      _D_cplx_lr_div_ix.o \
55      _D_cplx_lr_div_rx.o \
56      _D_cplx_mul.o \
57      _F_cplx_div.o \
58      _F_cplx_div_ix.o \
59      _F_cplx_div_rx.o \

```

```

60      _F_cplx_lr_div.o \
61      _F_cplx_lr_div_ix.o \
62      _F_cplx_lr_div_rx.o \
63      _F_cplx_mul.o \
64      _X_cplx_div.o \
65      _X_cplx_div_ix.o \
66      _X_cplx_div_rx.o \
67      _X_cplx_lr_div.o \
68      _X_cplx_lr_div_ix.o \
69      _X_cplx_lr_div_rx.o \
70      _X_cplx_mul.o \
71      _base_il.o \
72      fpgetmask.o \
73      fpgetround.o \
74      fpgetsticky.o \
75      fpsetmask.o \
76      fpsetround.o \
77      fpsetsticky.o \
78      fpstart.o

80 FPASMOBJS=   \
81      __xgetRD.o \
82      _xtoll.o \
83      _xtoull.o \
84      fpcw.o

86 ATOMICOBJS= \
87      atomic.o

89 XATTROBJS=   \
90      xattr_common.o

92 COMOBJS=     \
93      bcmp.o \
94      bcopy.o \
95      bsearch.o \
96      bzero.o \
97      qsort.o \
98      strtol.o \
99      strtoul.o \
100     strtoll.o \
101     strtoull.o

103 DTRACEOBJS= \
104     dtrace_data.o

106 GENOBJS=     \
107     _div64.o \
108     _divdi3.o \
109     _getsp.o \
110     _mul64.o \
111     abs.o \
112     alloca.o \
113     byteorder.o \
114     byteorder64.o \
115     cuexit.o \
116     ecvt.o \
117     errlst.o \
118     i386_data.o \
119     ladd.o \
120     ldivide.o \
121     lmul.o \
122     lock.o \
123     lshiftl.o \
124     lsign.o \
125     lsub.o \

```

```

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

148 # sysobjs that contain large-file interfaces
149 COMSYSOBS64=
150      fstatvfs64.o    \
151      getdents64.o   \
152      getrlimit64.o  \
153      lseek64.o      \
154      mmap64.o       \
155      pread64.o      \
156      pwrite64.o     \
157      setrlimit64.o \
158      statvfs64.o

160 SYSOBS64=

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

```

```

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

```

```

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

292 SYSOBJS=
293     __clock_gettime.o \|
294     __getcontext.o   \|
295     __uadmin.o       \|
296     __lwp_mutex_unlock.o \|
297     __stack_grow.o  \|
298     door.o           \|
299     forkx.o          \|
300     forkallx.o       \|
301     getcontext.o     \|
302     gettimeofday.o  \|
303     lwp_private.o   \|
304     nuname.o         \|
305     ptrace.o         \|
306     syscall.o        \|
307     sysi86.o         \|
308     tls_get_addr.o  \|
309     uadmin.o         \|
310     umount.o         \|
311     uname.o          \|
312     vforkx.o         \|
313     xstat.o          \|

315 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
316 PORTGEN64=
317     _xftw64.o        \|
318     attropen64.o     \|
319     ftw64.o          \|
320     mkstemp64.o     \|
321     nftw64.o         \|
322     tell64.o         \|
323     truncate64.o     \|

```

```

325 # objects from source under $(LIBCDIR)/port
326 PORTFP=
327     __flt_decim.o    \|
328     __flt_rounds.o  \|
329     __tbl_10_b.o     \|
330     __tbl_10_h.o     \|
331     __tbl_10_s.o     \|
332     __tbl_2_b.o      \|
333     __tbl_2_h.o      \|
334     __tbl_2_s.o      \|
335     __tbl_fdq.o      \|
336     __tbl_tens.o     \|
337     __x_power.o      \|
338     __base_sup.o     \|
339     aconvert.o       \|
340     decimal_bin.o   \|
341     double_decim.o  \|
342     econvert.o       \|
343     fconvert.o       \|
344     file_decim.o    \|
345     finite.o         \|
346     fp_data.o        \|
347     func_decim.o    \|
348     gconvert.o       \|
349     hex_bin.o        \|
350     ieee_globals.o  \|
351     pack_float.o     \|
352     sigfpe.o         \|
353     string_decim.o  \|

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

```

```

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

```

```

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

```



```

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

```

```

588 ttyname.o \
589 ttyslot.o \
590 ualarm.o \
591 ucred.o \
592 valloc.o \
593 vfmt.o \
594 vpfmt.o \
595 waitpid.o \
596 walkstack.o \
597 wdata.o \
598 xgetwidth.o \
599 xpg4.o \
600 xpg6.o \
602 PORTPRINT_W= \
603 doprnt_w.o \
605 PORTPRINT= \
606 asprintf.o \
607 doprnt.o \
608 fprintf.o \
609 printf.o \
610 snprintf.o \
611 sprintf.o \
612 vfprintf.o \
613 vprintf.o \
614 vsnprintf.o \
615 vsprintf.o \
616 vwprintf.o \
617 wprintf.o \
619 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
620 PORTPRINT_C89= \
621 vfprintf_c89.o \
622 fprintf_c89.o \
623 vsnprintf_c89.o \
624 vsprintf_c89.o \
625 vwprintf_c89.o \
627 PORTSTDIO_C89= \
628 vscanf_c89.o \
629 vwscanf_c89.o \
631 # portable stdio objects that contain large file interfaces.
632 # Note: fopen64 is a special case, as we build it small.
633 PORTSTDIO64= \
634 fopen64.o \
635 fpos64.o \
637 PORTSTDIO_W= \
638 doscan_w.o \
640 PORTSTDIO= \
641 __extensions.o \
642 __endopen.o \
643 __filbuf.o \
644 __findbuf.o \
645 __flsbuf.o \
646 __wrtchk.o \
647 clearerr.o \
648 ctermid.o \
649 ctermid_r.o \
650 cuserid.o \
651 data.o \
652 doscan.o \
653 fdopen.o \

```

```

654 feof.o \
655 ferror.o \
656 fgetc.o \
657 fgets.o \
658 fileno.o \
659 flockf.o \
660 flush.o \
661 fopen.o \
662 fpos.o \
663 fputc.o \
664 fputs.o \
665 fread.o \
666 fseek.o \
667 fseek.o \
668 ftell.o \
669 ftello.o \
670 fwrite.o \
671 getc.o \
672 getchar.o \
673 getline.o \
674 getpass.o \
675 gets.o \
676 getw.o \
677 mse.o \
678 popen.o \
679 putc.o \
680 putchar.o \
681 puts.o \
682 putw.o \
683 rewind.o \
684 scanf.o \
685 setbuf.o \
686 setbuffer.o \
687 setvbuf.o \
688 system.o \
689 tempnam.o \
690 tmpfile.o \
691 tmpnam_r.o \
692 ungetc.o \
693 vscanf.o \
694 vwscanf.o \
695 wscanf.o \
697 PORTI18N= \
698 getwchar.o \
699 putwchar.o \
700 putws.o \
701 strcasecmp.o \
702 strcasestr.o \
703 strncasecmp.o \
704 strtows.o \
705 wcsnlen.o \
706 wcstr.o \
707 wcstoimax.o \
708 wcstol.o \
709 wcstoul.o \
710 wcs wcs.o \
711 wmemchr.o \
712 wmemcmp.o \
713 wmemcpy.o \
714 wmemmove.o \
715 wmemset.o \
716 wscasecmp.o \
717 wscat.o \
718 wschr.o \
719 wscmp.o \

```

```

720 wscopy.o \
721 wcs spn.o \
722 wsdup.o \
723 wslen.o \
724 wncasecmp.o \
725 wncat.o \
726 wncmp.o \
727 wncpy.o \
728 wspbrk.o \
729 vsprintf.o \
730 wsrchr.o \
731 wsscanf.o \
732 wssp n.o \
733 wstod.o \
734 wstok.o \
735 wstol.o \
736 wstoll.o \
737 wxfrm.o \
738 gettext.o \
739 gettext_gnu.o \
740 gettext_real.o \
741 gettext_util.o \
742 isdigit.o \
743 plural_parser.o \
744 wdresolve.o \
745 _ctype.o \
746 isascii.o \
747 toascii.o \
749 PORTI18N_COND= \
750 wcstol_longlong.o \
751 wcstoul_longlong.o \
753 PORTLOCALE= \
754 big5.o \
755 btowc.o \
756 collate.o \
757 collcmp.o \
758 euc.o \
759 fnmatch.o \
760 fgetwc.o \
761 fgetws.o \
762 fix_grouping.o \
763 fputwc.o \
764 fputws.o \
765 fwide.o \
766 gb18030.o \
767 gb2312.o \
768 gbk.o \
769 getdate.o \
770 iswctype.o \
771 ldp art.o \
772 lmessages.o \
773 lnumeric.o \
774 lmonetary.o \
775 localeconv.o \
776 mbftowc.o \
777 mblen.o \
778 mbrlen.o \
779 mbrtowc.o \
780 mbsinit.o \
781 mbsnr towcs.o \
782 mbsrtowcs.o \
783 mbstowcs.o \
784 mbtowc.o \
785 mskanji.o \

```

```

786 nextwctype.o \
787 nl_langinfo.o \
788 none.o \
789 regcomp.o \
790 regfree.o \
791 regerror.o \
792 regexec.o \
793 rune.o \
794 runetype.o \
795 setlocale.o \
796 setrunelocale.o \
797 strcoll.o \
798 strfmon.o \
799 strftime.o \
800 strtptime.o \
801 strxfrm.o \
802 table.o \
803 timelocal.o \
804 tolower.o \
805 towlower.o \
806 ungetwc.o \
807 utf8.o \
808 wcrtoomb.o \
809 wcscoll.o \
810 wcsftime.o \
811 wcsnrtoombs.o \
812 wcsrtombs.o \
813 wcswidth.o \
814 wcstombs.o \
815 wcsxfrm.o \
816 wctob.o \
817 wctomb.o \
818 wctrans.o \
819 wctype.o \
820 wcwidth.o \
821 wscoll.o \
\
823 AIOBJS= \
824 aio.o \
825 aio_alloc.o \
826 posix_aio.o \
\
828 RTOBJS= \
829 clock_timer.o \
830 mqueue.o \
831 pos4obj.o \
832 sched.o \
833 sem.o \
834 shm.o \
835 sigev_thread.o \
\
837 TPOOLBJS= \
838 thread_pool.o \
\
840 THREADSOBJS= \
841 alloc.o \
842 assfail.o \
843 cancel.o \
844 door_calls.o \
845 pthr_attr.o \
846 pthr_barrier.o \
847 pthr_cond.o \
848 pthr_mutex.o \
849 pthr_rwlock.o \
850 pthread.o \
851 rwlock.o \

```

```

852 calls.o \
853 sema.o \
854 sigaction.o \
855 spawn.o \
856 synch.o \
857 tdb_agent.o \
858 thr.o \
859 thread_interface.o \
860 tls.o \
861 tsd.o \
\
863 THREADSMACHOBJS= \
864 machdep.o \
\
866 THREADSASMOBJS= \
867 asm_subr.o \
\
869 UNICODEOBJS= \
870 u8_textprep.o \
871 uconv.o \
\
873 UNWINDMACHOBJS= \
874 unwind.o \
\
876 UNWINDASMOBJS= \
877 unwind_frame.o \
\
879 # objects that implement the transitional large file API
880 PORTSYS64= \
881 lockf64.o \
882 stat64.o \
\
884 PORTSYS= \
885 _autofssys.o \
886 access.o \
887 acctctl.o \
888 BSD_signal.o \
889 chmod.o \
890 chown.o \
891 corectl.o \
892 exacctsyst.o \
893 execl.o \
894 execl.o \
895 execv.o \
896 fcntl.o \
897 getpagesizes.o \
898 getpeerucred.o \
899 inst_sync.o \
900 issetugid.o \
901 label.o \
902 link.o \
903 lockf.o \
904 lwp.o \
905 lwp_cond.o \
906 lwp_rwlock.o \
907 lwp_sigmask.o \
908 meminfosys.o \
909 mkdir.o \
910 mknod.o \
911 msgsys.o \
912 nfssys.o \
913 open.o \
914 pgrpsys.o \
915 posix_sigwait.o \
916 ppriv.o \
917 psetsys.o \

```

```

918 rctlsys.o \
919 readlink.o \
920 rename.o \
921 sbrk.o \
922 semsys.o \
923 set_errno.o \
924 sharefs.o \
925 shmsys.o \
926 sidsys.o \
927 siginterrupt.o \
928 signal.o \
929 sigpending.o \
930 sigstack.o \
931 stat.o \
932 symlink.o \
933 tasksys.o \
934 time.o \
935 time_util.o \
936 ucontext.o \
937 unlink.o \
938 ustat.o \
939 utimesys.o \
940 zone.o \

942 PORTREGEX= \
943 glob.o \
944 regcmp.o \
945 regex.o \
946 wordexp.o \

948 MOSTOBSJS= \
949 $(STRETS) \
950 $(CRTOBSJS) \
951 $(DYNOBJS) \
952 $(FPOBSJS) \
953 $(FPASMOBSJS) \
954 $(ATOMICOBJS) \
955 $(XATROBSJS) \
956 $(COMOBSJS) \
957 $(DTRACEOBSJS) \
958 $(GENOBSJS) \
959 $(PORTFP) \
960 $(PORTGEN) \
961 $(PORTGEN64) \
962 $(PORTI18N) \
963 $(PORTI18N_COND) \
964 $(PORTLOCALE) \
965 $(PORTPRINT) \
966 $(PORTPRINT_C89) \
967 $(PORTPRINT_W) \
968 $(PORTREGEX) \
969 $(PORTSTDIO) \
970 $(PORTSTDIO64) \
971 $(PORTSTDIO_C89) \
972 $(PORTSTDIO_W) \
973 $(PORTSYS) \
974 $(PORTSYS64) \
975 $(AIOOBSJS) \
976 $(RTOBSJS) \
977 $(TPOOLOBSJS) \
978 $(THREADSOBSJS) \
979 $(THREADSMACHOBSJS) \
980 $(THREADSASMOBSJS) \
981 $(UNICODEOBSJS) \
982 $(UNWINDMACHOBSJS) \
983 $(UNWINDASMOBSJS) \

```

```

984 $(COMSYSOBSJS) \
985 $(SYSOBSJS) \
986 $(COMSYSOBSJS64) \
987 $(SYSOBSJS64) \
988 $(VALUES) \

990 TRACEOBSJS= \
991 plockstat.o \

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

1005 LDPASS_OFF= $(POUND_SIGN)

1007 # include common library definitions
1008 include ../../Makefile.lib

1010 # we need to override the default SONAME here because we might
1011 # be building a variant object (still libc.so.1, but different filename)
1012 SONAME = libc.so.1

1014 CFLAGS += $(CCVERBOSE) $(CTF_FLAGS)

1016 # This is necessary to avoid problems with calling _ex_unwind().
1017 # We probably don't want any inlining anyway.
1018 XINLINE = -xinline=
1019 CFLAGS += $(XINLINE)

1021 CERRWARN += -_gcc=-Wno-parentheses
1022 CERRWARN += -_gcc=-Wno-switch
1023 CERRWARN += -_gcc=-Wno-uninitialized
1024 CERRWARN += -_gcc=-Wno-unused-value
1025 CERRWARN += -_gcc=-Wno-unused-label
1026 CERRWARN += -_gcc=-Wno-unused-variable
1027 CERRWARN += -_gcc=-Wno-type-limits
1028 CERRWARN += -_gcc=-Wno-char-subscripts
1029 CERRWARN += -_gcc=-Wno-clobbered
1030 CERRWARN += -_gcc=-Wno-unused-function
1031 CERRWARN += -_gcc=-Wno-address

1033 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1034 # enables ASSERT() checking in the threads portion of the library.
1035 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1036 THREAD_DEBUG =
1037 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1039 # Make string literals read-only to save memory.
1040 CFLAGS += $(XSTRCONST)

1042 ALTPICS= $(TRACEOBSJS:%=pics/%)

1044 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) \
1045 $(EXTPICS) $(LDLIBS)

1047 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1049 #

```

```

1050 # EXTN_CPPFLAGS and EXTN_CFLAGS set in enclosing Makefile
1051 #
1052 CFLAGS +=          $(EXTN_CFLAGS)
1053 CPPFLAGS+=        -D_REENTRANT -Di386 $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1054                  -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1055 ASFLAGS=          $(AS_PICFLAGS) -P -D_STDC -D_ASM $(CPPFLAGS) $(i386_AS_XARCH)

1057 # As a favor to the dtrace syscall provider, libc still calls the
1058 # old syscall traps that have been obsoleted by the *at() interfaces.
1059 # Delete this to compile libc using only the new *at() system call traps
1060 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1062 # Inform the run-time linker about libc specialized initialization
1063 RTLDINFO =        -z rtldinfo=tls_rtldinfo
1064 DYNFLAGS +=       $(RTLDINFO)

1066 # Force libc's internal references to be resolved immediately upon loading
1067 # in order to avoid critical region problems. Since almost all libc symbols
1068 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1069 DYNFLAGS +=       -znov

1071 DYNFLAGS +=       -e __rtboot
1072 DYNFLAGS +=       $(EXTN_DYNFLAGS)

1074 # Inform the kernel about the initial DTrace area (in case
1075 # libc is being used as the interpreter / runtime linker).
1076 DTRACE_DATA =     -zdtrace=dtrace_data
1077 DYNFLAGS +=       $(DTRACE_DATA)

1079 # DTrace needs an executable data segment.
1080 MAPFILE.NED=

1082 BUILD.s=         $(AS) $(ASFLAGS) $< -o $@

1084 # Override this top level flag so the compiler builds in its native
1085 # C99 mode. This has been enabled to support the complex arithmetic
1086 # added to libc.
1087 C99MODE=         $(C99_ENABLE)

1089 # libc method of building an archive
1090 # The "$(GREP) -v ' L '" part is necessary only until
1091 # lorder is fixed to ignore thread-local variables.
1092 BUILD.AR= $(RM) $@ ; \
1093           $(AR) q $@ `$(LORDER) $(MOSTOBSJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1095 # extra files for the clean target
1096 CLEANFILES=      \
1097                 $(LIBCDIR)/port/gen/errlst.c \
1098                 $(LIBCDIR)/port/gen/new_list.c \
1099                 assym.h \
1100                 genassym \
1101                 crt/_rtld.s \
1102                 crt/_rtbootld.s \
1103                 pics/_rtbootld.o \
1104                 pics/crti.o \
1105                 pics/crtn.o \
1106                 $(ALTPICS)

1108 CLOBBERFILES += $(LIB_PIC)

1110 # list of C source for lint
1111 SRCS=            \
1112                 $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c) \
1113                 $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c) \
1114                 $(COMOBSJS:%.o=$(SRC)/common/util/%.c) \
1115                 $(DTRACEOBSJS:%.o=$(SRC)/common/dtrace/%.c) \

```

```

1116                 $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c) \
1117                 $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c) \
1118                 $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c) \
1119                 $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c) \
1120                 $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c) \
1121                 $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c) \
1122                 $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c) \
1123                 $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c) \
1124                 $(AIOOBSJS:%.o=$(LIBCDIR)/port/aio/%.c) \
1125                 $(RTOBSJS:%.o=$(LIBCDIR)/port/rt/%.c) \
1126                 $(TPOOLBSJS:%.o=$(LIBCDIR)/port/tpool/%.c) \
1127                 $(THREADSOBSJS:%.o=$(LIBCDIR)/port/threads/%.c) \
1128                 $(THREADSMACHOBSJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1129                 $(UNICODEOBSJS:%.o=$(SRC)/common/unicode/%.c) \
1130                 $(UNWINDMACHOBSJS:%.o=$(LIBCDIR)/port/unwind/%.c) \
1131                 $(FPOBSJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c) \
1132                 $(LIBCBASE)/gen/ecvt.c \
1133                 $(LIBCBASE)/gen/makectxt.c \
1134                 $(LIBCBASE)/gen/signfolst.c \
1135                 $(LIBCBASE)/gen/siglongjmp.c \
1136                 $(LIBCBASE)/gen/strcmp.c \
1137                 $(LIBCBASE)/gen/sync_instruction_memory.c \
1138                 $(LIBCBASE)/sys/ptrace.c \
1139                 $(LIBCBASE)/sys/uadmin.c

1141 # conditional assignments
1142 $(DYNLIB) := CRTI = crt.i.o
1143 $(DYNLIB) := CRTN = crtn.o

1145 # Files which need the threads .il inline template
1146 TIL=
1147 aio.o \
1148 alloc.o \
1149 assfail.o \
1150 atexit.o \
1151 atfork.o \
1152 cancel.o \
1153 door_calls.o \
1154 err.o \
1155 errno.o \
1156 lwp.o \
1157 ma.o \
1158 machdep.o \
1159 posix_aio.o \
1160 pthr_attr.o \
1161 pthr_barrier.o \
1162 pthr_cond.o \
1163 pthr_mutex.o \
1164 pthr_rwlock.o \
1165 pthread.o \
1166 rand.o \
1167 rwlock.o \
1168 scalls.o \
1169 sched.o \
1170 sema.o \
1171 sigaction.o \
1172 sigev_thread.o \
1173 spawn.o \
1174 stack.o \
1175 synch.o \
1176 tdb_agent.o \
1177 thr.o \
1178 thread_interface.o \
1179 thread_pool.o \
1180 tls.o \
1181 tsd.o \

```

```

1182      unwind.o

1184 THREADS_INLINES = $(LIBCBASE)/threads/i386.il
1185 $(TIL:%=pics/%) := CFLAGS += $(THREADS_INLINES)

1187 # pics/mul64.o := CFLAGS += $(LIBCBASE)/crt/mul64.il

1189 # large-file-aware components that should be built large

1191 $(COMSYSOBS64:%=pics/%) := \
1192     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1194 $(SYSOBS64:%=pics/%) := \
1195     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1197 $(PORTGEN64:%=pics/%) := \
1198     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1200 $(PORTSTDIO64:%=pics/%) := \
1201     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1203 $(PORTSYS64:%=pics/%) := \
1204     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1206 $(PORTSTDIO_W:%=pics/%) := \
1207     CPPFLAGS += -D_WIDE

1209 $(PORTPRINT_W:%=pics/%) := \
1210     CPPFLAGS += -D_WIDE

1212 $(PORTPRINT_C89:%=pics/%) := \
1213     CPPFLAGS += -D_C89_INTMAX32

1215 $(PORTSTDIO_C89:%=pics/%) := \
1216     CPPFLAGS += -D_C89_INTMAX32

1218 $(PORTI18N_COND:%=pics/%) := \
1219     CPPFLAGS += -D_WCS_LONGLONG

1221 .KEEP_STATE:

1223 all: $(LIBS) $(LIB_PIC)

1225 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1226 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1227 lint := LINTFLAGS += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1229 lint:
1230     @echo $(LINT.c) ...
1231     @$$(LINT.c) $(SRCS) $(LDLIBS)

1233 $(LINTLIB):= SRCS=$(LIBCDIR)/port/l1ib-1c
1234 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1235 $(LINTLIB):= LINTFLAGS=-nvx

1237 # object files that depend on inline template
1238 $(TIL:%=pics/%): $(LIBCBASE)/threads/i386.il
1239 # pics/mul64.o: $(LIBCBASE)/crt/mul64.il

1241 # include common libc targets
1242 include $(LIBCDIR)/Makefile.targ

1244 # We need to strip out all CTF and DOF data from the static library
1245 $(LIB_PIC) := DIR = pics
1246 $(LIB_PIC): pics $$$(PICS)
1247     $(BUILD.AR)

```

```

1248     $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1249     $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1250     $(AR) -ts $@ > /dev/null
1251     $(POST_PROCESS_A)

1253 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.c
1254     $(CC) $(CPPFLAGS) $(CTF_FLAGS) -o -s $(C_PICFLAGS) \
1255     $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1256     $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
1257     $(RM) $(LIBCBASE)/crt/_rtld.s

1259 # partially built from C source
1260 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1261     $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
1262     $(CTFCONVERT_O)

1264 ASSYMDEP_OBJS= \
1265     _lwp_mutex_unlock.o \
1266     _stack_grow.o \
1267     getcontext.o \
1268     setjmp.o \
1269     tls_get_addr.o \
1270     vforkx.o

1272 $(ASSYMDEP_OBJS:%=pics%) := CPPFLAGS += -I.

1274 $(ASSYMDEP_OBJS:%=pics%): assym.h

1276 # assym.h build rules

1278 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1280 genassym: $(GENASSYM_C)
1281     $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1282     -D_EXTENSIONS_ $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1284 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1286 assym.h: $(OFFSETS) genassym
1287     $(OFFSETS_CREATE) <$(OFFSETS) >$@
1288     ./genassym >>$@

1290 # derived C source and related explicit dependencies
1291 $(LIBCDIR)/port/gen/errlst.c + \
1292 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1293     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1295 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1297 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c

```

new/usr/src/lib/libc/port/gen/cfmakeraw.c

1

```
*****
1366 Sat Sep 28 12:17:12 2013
new/usr/src/lib/libc/port/gen/cfmakeraw.c
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
```

```
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 2011 Nexenta Systems, Inc. All rights reserved.
24 */

26 #include "lint.h"
27 #include <sys/types.h>
28 #include <sys/termios.h>

30 /*
31  * set the structure pointed to by termios_p to raw mode
32 */

34 void
35 cfmakeraw(struct termios *termios_p)
36 {
37     termios_p->c_iflag &= ~(IGNBRK|BRKINT|PARMRK|ISTRIP|
38         INLCR|IGNCR|ICRNL|IXON);
39     termios_p->c_oflag &= ~OPOST;
40     termios_p->c_lflag &= ~(ECHO|ECHONL|ICANON|ISIG|IEXTEN);
41     termios_p->c_cflag &= ~(CSIZE|PARENB);
42     termios_p->c_cflag |= CS8;
43     termios_p->c_cc[VMIN] = 1;
44     termios_p->c_cc[VTIME] = 0;
45 }
46 #endif /* ! codereview */
```

```

*****
45688 Sat Sep 28 12:17:12 2013
new/usr/src/lib/libc/port/llib-1c
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24  * Copyright 2011 Nexenta Systems, Inc. All rights reserved.
25  * Copyright 2013 OmniTI Computer Consulting, Inc. All rights reserved.
26 */

28 /* LINTLIBRARY */
29 /* PROTOLIB1 */

31 #define __EXTENSIONS__

33 #include <aio.h>
34 #include <alloca.h>
35 #include <attr.h>
36 #include <atomic.h>
37 #include <ctype.h>
38 #include <deflt.h>
39 #include <dirent.h>
40 #include <dlfcn.h>
41 #include <door.h>
42 #include <err.h>
43 #include <sys/errno.h>
44 #include <euc.h>
45 #include <fcntl.h>
46 #include <float.h>
47 #include <fmtmsg.h>
48 #include <fnmatch.h>
49 #include <ftw.h>
50 #include <glob.h>
51 #include <getwidth.h>
52 #include <grp.h>
53 #include <iconv.h>
54 #include <langinfo.h>
55 #include <libgen.h>
56 #include <libw.h>
57 #include <locale.h>
58 #include <memory.h>
59 #include <mon.h>

```

```

60 #include <mqueue.h>
61 #include <nan.h>
62 #include <ndbm.h>
63 #include <limits.h>
64 #include <nln_types.h>
65 #include <poll.h>
66 #include <project.h>
67 #include <priv.h>
68 #include <pwd.h>
69 #include <rctl.h>
70 #include <regex.h>
71 #include <rpcsvc/ypclnt.h>
72 #include <sched.h>
73 #include <search.h>
74 #include <semaphore.h>
75 #include <setjmp.h>
76 #include <shadow.h>
77 #include <siginfo.h>
78 #include <signal.h>
79 #include <stdarg.h>
80 #include <ucred.h>
81 #include <sys/ucred.h>
82 #include <unistd.h>
83 #include <ulimit.h>
84 #include <utime.h>
85 #include <stddef.h>
86 #include <stdio.h>
87 #include <stdlib.h>
88 #include <string.h>
89 #include <stropts.h>
90 #include <synch.h>
91 #include <sys/acctctl.h>
92 #include <sys/acl.h>
93 #include <sys/asynch.h>
94 #include <sys/byteorder.h>
95 #include <sys/cladm.h>
96 #include <sys/corectl.h>
97 #include <sys/dl.h>
98 #include <sys/exacct.h>
99 #include <sys/fcntl.h>
100 #include <sys/file.h>
101 #include <sys/fs/namemode.h>
102 #include <sys/instance.h>
103 #include <sys/ipc.h>
104 #include <sys/lwp.h>
105 #include <sys/mkdev.h>
106 #include <sys/mman.h>
107 #include <sys/mnttab.h>
108 #include <sys/mount.h>
109 #include <sys/msg.h>
110 #include <sys/param.h>
111 #include <sys/priocntl.h>
112 #include <sys/procset.h>
113 #include <sys/processor.h>
114 #include <sys/pset.h>
115 #include <sys/rctl_impl.h>
116 #include <sys/sem.h>
117 #include <sys/shm.h>
118 #include <sys/sid.h>
119 #include <sys/signal.h>
120 #include <sys/stat.h>
121 #include <sys/statvfs.h>
122 #include <sys/strlog.h>
123 #include <sys/stropts.h>
124 #include <sys/syscall.h>
125 #include <sys/sysconfig.h>

```



```

126 #include <sys/syslog.h>
127 #include <sys/systeminfo.h>
128 #include <sys/task.h>
129 #include <sys/termio.h>
130 #include <sys/termios.h>
131 #include <sys/u8_textprep.h>
132 #include <sys/time.h>
133 #include <sys/timeb.h>
134 #include <sys/times.h>
135 #include <sys/types.h>
136 #include <sys/uadmin.h>
137 #include <sys/utsnam.h>
138 #include <sys/vfstab.h>
139 #include <sys/sendfile.h>
140 #include <sys/zone.h>
141 #include <termio.h>
142 #include <time.h>
143 #include <tzfile.h>
144 #include <ucontext.h>
145 #include <utmpx.h>
146 #include <values.h>
147 #include <wait.h>
148 #include <wchar.h>
149 #include <wctype.h>
150 #include <wider.h>
151 #include <wordexp.h>
152 #include <thread.h>
153 #include <pthread.h>
154 #include <schedctl.h>
155 #include <zone.h>
156 #include <port.h>
157 #include <spawn.h>
158 #include <inttypes.h>
159 #include <getopt.h>
160 #include <stdio_ext.h>
161 #if defined(__i386)
162 #include <sys/sysi86.h>
163 #endif
164 #if defined(__amd64)
165 #include <stack_unwind.h>
166 #endif

168 /*
169  * This really comes from the crt*.s startup modules.
170  */
171 char **environ;

173 /*
174  * This is a GNU/Linux/BSD compatibility interface,
175  * not declared in any header file.
176  */
177 const char *__progname;

179 /*
180  * POSIX versions of standard libc routines; these aren't extracted
181  * from the headers above since we cannot #define _POSIX_C_SOURCE.
182  */
183 int __posix_readdir_r(DIR * _RESTRICT_KYWD, struct dirent * _RESTRICT_KYWD,
184 struct dirent ** _RESTRICT_KYWD);
185 int __posix_getgrgid_r(gid_t, struct group *, char *, size_t, struct group **);
186 int __posix_getgrnam_r(const char *, struct group *, char *, size_t,
187 struct group **);
188 int __posix_getpwuid_r(uid_t, struct passwd *, char *, size_t,
189 struct passwd **);
190 int __posix_getpwnam_r(const char *, struct passwd *, char *, size_t,
191 struct passwd **);

```

```

192 int __posix_sigwait(const sigset_t * _RESTRICT_KYWD, int * _RESTRICT_KYWD);
193 char * __posix_asctime_r(const struct tm * _RESTRICT_KYWD, char * _RESTRICT_KYWD);
194 char * __posix_ctime_r(const time_t *, char *);
195 int __posix_ttyname_r(int, char *, size_t);
196 int __posix_getlogin_r(char *, int);

198 /*
199  * XPG4 versions of standard libc routines; these aren't extracted
200  * from the headers above since we cannot #define _XPG4_2.
201  */
202 int __xpg4_putmsg(int, const struct strbuf *, const struct strbuf *, int);
203 int __xpg4_putpmsg(int, const struct strbuf *, const struct strbuf *, int, int);

205 /*
206  * These aren't extracted from the headers above because:
207  * - We cannot #define _STRPTIME_DONTZERO
208  * - We cannot #define _XPG5
209  */
210 char * __strptime_dontzero(const char *, const char *, struct tm *);
211 long __sysconf_xpg5(int);
212 wchar_t * __wcstok_xpg5(wchar_t * _RESTRICT_KYWD,
213 const wchar_t * _RESTRICT_KYWD, wchar_t ** _RESTRICT_KYWD);
214 size_t __wcsftime_xpg5(wchar_t * _RESTRICT_KYWD, size_t,
215 const wchar_t * _RESTRICT_KYWD, const struct tm * _RESTRICT_KYWD);
216 wint_t __fgetwc_xpg5(__FILE *);
217 wint_t __getwc_xpg5(__FILE *);
218 wint_t __getwchar_xpg5(void);
219 wint_t __fputwc_xpg5(wint_t, __FILE *);
220 wint_t __putwc_xpg5(wint_t, __FILE *);
221 wint_t __putwchar_xpg5(wint_t);
222 wchar_t * __fgetws_xpg5(wchar_t * _RESTRICT_KYWD, int, __FILE * _RESTRICT_KYWD);
223 int __fputws_xpg5(const wchar_t * _RESTRICT_KYWD, __FILE * _RESTRICT_KYWD);
224 wint_t __ungetwc_xpg5(wint_t, __FILE *);

226 /*
227  * /usr/src/lib/libc/port/gen routines
228  */

230 /* _ctype.c */

232 /* _loc_data.c */

234 /* _locale.c */

236 /* _set_tab.c */
237 int __set_tab(const char *loc, int cat);

239 /* _xftw.c */
240 int __xftw(int ver, const char *path, int (*fn)(), int depth);

242 /* a64l.c */
243 long a64l(const char *);

245 /* abort.c */
246 void abort(void);

248 /* abs.c */
249 int abs(int arg);
250 long labs(long int arg);

252 /* assert.c */
253 void __assert(const char *assertion, const char *filename, int line_num);
254 void __assert_c99(const char *assertion, const char *filename, int line_num,
255 const char *funcname);

257 /* atexit.c */

```

```

258 int atexit(void(*func)());
259 void _exithandle(void);

261 /* atof.c */
262 double atof(const char *p);

264 /* atoi.c */
265 int atoi(const char *p);

267 /* atol.c */
268 long atol(const char *p);

270 /* basename.c */
271 char *basename(char *s);

273 /* bcmp.c */
274 int bcmp(const void *s1, const void *s2, size_t len);

276 /* bcopy.c */
277 void bcopy(const void *s1, void *s2, size_t len);

279 /* bsearch.c */
280 void *bsearch(const void *ky, const void *bs, size_t nel,
281             size_t width, int (*compar)());

283 /* bzero.c */
284 void bzero(void *sp, size_t len);

286 /* calloc.c */
287 void *calloc(size_t num, size_t size);

289 /* catclose.c */
290 int catclose(nl_catd catd);

292 /* catgets.c */
293 char *catgets(nl_catd catd, int set_num, int msg_num, const char *s);

295 /* catopen.c */
296 nl_catd catopen(const char *name, int mode);

298 /* cfgetispeed.c */
299 speed_t cfgetispeed(const struct termios *termios_p);

301 /* cfgetospeed.c */
302 speed_t cfgetospeed(const struct termios *termios_p);

304 /* cfree.c */
305 void cfree(void *p, size_t num, size_t size);

307 /* cfsetispeed.c */
308 int cfsetispeed(struct termios *termios_p, speed_t speed);

310 /* cfsetospeed.c */
311 int cfsetospeed(struct termios *termios_p, speed_t speed);

313 /* cfmakeraw.c */
314 void cfmakeraw(struct termios *termios_p);

316 #endif /* ! codereview */
317 /* cftime.c */
318 int cftime(char *buf, char *format, const time_t *t);
319 int asctime(char *buf, const char *format, const struct tm *tm);

321 /* clock.c */
322 clock_t clock(void);

```

```

324 /* closedir.c */
325 int closedir(DIR *dirp);

327 /* confstr.c */
328 size_t confstr(int name, char *buf, size_t length);

330 /* crypt.c */
331 void setkey(const char *key);
332 void encrypt(char *block, int fake);
333 char *crypt(const char *key, const char *salt);

335 /* csetlen.c */
336 int csetlen(int cset);
337 int csetcol(int cset);

339 /* ctime.c */
340 char *ctime(const time_t *t);
341 char *ctime_r(const time_t *, char *buf, int);
342 char *asctime(const struct tm *t);
343 char *asctime_r(const struct tm *, char *, int);

345 /* ctypefcns.c */
346 int isalpha(int c);
347 int isupper(int c);
348 int islower(int c);
349 int isdigit(int c);
350 int isxdigit(int c);
351 int isalnum(int c);
352 int isspace(int c);
353 int ispunct(int c);
354 int isprint(int c);
355 int isgraph(int c);
356 int iscntrl(int c);
357 int isascii(int c);
358 int _toupper(int c);
359 int _tolower(int c);
360 int toascii(int c);

362 /* daemon.c */
363 int daemon(int nochdir, int noclose);

365 /* directio.c */
366 int directio(int filedes, int advice);

368 /* dirname.c */
369 char *dirname(char *s);

371 /* div.c */
372 div_t div(int numer, int denom);
373 ldiv_t ldiv(long int numer, long int denom);

375 /* drand48.c */
376 double drand48(void);
377 double erand48(unsigned short *xsubi);
378 long lrand48(unsigned short *xsubi, unsigned int m);
379 long lrand48(void);
380 long mrand48(void);
381 void srand48(long seedval);
382 unsigned short *seed48(unsigned short seed16v[3]);
383 void lcong48(unsigned short param[7]);
384 long nrand48(unsigned short *xsubi);
385 long jrand48(unsigned short *xsubi);

387 /* dup.c */
388 int dup(int fildes);
389 int dup2(int fildes, int fildes2);

```

```

390 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 _errxrp(FILE *, int, const char *, ...);
402 void _verrxrp(FILE *, int, const char *, va_list);
403 void _warnfp(FILE *, const char *, ...);
404 void _vwarnfp(FILE *, const char *, va_list);
405 void _warnxrp(FILE *, const char *, ...);
406 void _vwarnxrp(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 /* ffs.c */
427 int ffs(int field);

429 /* fmtmsg.c */
430 int addseverity(int value, const char *string);
431 int fmtmsg(long class, const char *label, int severity, const char *text,
432           const char *action, const char *tag);

434 /* ftime.c */
435 int ftime(struct timeb *tp);

437 /* ftok.c */
438 key_t ftok(const char *path, int id);

440 /* gcvt.c */
441 char *gcvt(double number, int ndigit, char *buf);

443 /* getcwd.c */
444 char *getcwd(char *str, size_t size);

446 /* getdate.c */
447 struct tm *getdate(const char *expression);
448 #ifdef getdate_err
449 #undef getdate_err
450 #endif
451 int getdate_err;

453 /* getdate_data.c */

455 /* getdate_gd.c */

```

```

457 /* getdtblsize.c */
458 int getdtblsize(void);

460 /* getenv.c */
461 char *getenv(const char *name);

463 /* getexecname.c */
464 const char *getexecname(void);

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

474 /* gethostid.c */
475 long gethostid(void);

477 /* gethz.c */
478 int gethz(void);

480 /* getisax.c */
481 uint_t getisax(uint32_t *, uint_t);

483 /* getlogin.c */
484 char *getlogin(void);
485 char *getlogin_r(char *, int);

487 /* getmntent.c */
488 int getmntany(FILE *fd, struct mnttab *mgetp, struct mnttab *mrefp);
489 int getmntent(FILE *fd, struct mnttab *mp);

491 /* getnetgrent.c */
492 int setnetgrent(const char *grp);
493 int endnetgrent(void);
494 int getnetgrent(char **machinep, char **namep, char **domainp);

496 /* getopt.c */
497 int getopt(int argc, char *const *argv, const char *opts);

499 /* getopt_long.c */
500 int getopt_clip(int argc, char *const *argv, const char *optstring,
501               const struct option *long_options, int *long_index);
502 int getopt_long(int argc, char *const *argv, const char *optstring,
503               const struct option *long_options, int *long_index);
504 int getopt_long_only(int argc, char *const *argv, const char *optstring,
505                  const struct option *long_options, int *long_index);

507 /* getpagesize.c */
508 int getpagesize(void);

510 /* getpw.c */
511 int getpw(uid_t uid, char *buf);

513 /* getpwnam.c */
514 struct passwd *getpwnam(const char *name);
515 struct passwd *getpwuid(uid_t uid);
516 struct passwd *fgetpwent_r(FILE *, struct passwd *, char *, int);
517 struct passwd *getpwent_r(struct passwd *, char *, int);
518 struct passwd *getpwnam_r(const char *, struct passwd *, char *, int);
519 struct passwd *getpwuid_r(uid_t, struct passwd *, char *, int);

521 /* getrusage.c */

```

```

522 int getrusage(int who, struct rusage *rusage);

524 /* gettimeofday.c */
525 int gettimeofday(struct timeval *_RESTRICT_KYWD tp, void *_RESTRICT_KYWD);

527 /* getspent.c */
528 void setspent(void);
529 void endspent(void);
530 struct spwd *getspent(void);
531 struct spwd *getspent_r(struct spwd *, char *, int);
532 struct spwd *fgetspent(FILE *f);
533 struct spwd *fgetspent_r(FILE *, struct spwd *, char *, int);
534 struct spwd *getspnam(const char *name);
535 struct spwd *getspnam_r(const char *, struct spwd *, char *, int);
536 int putspent(const struct spwd *p, FILE *f);

538 /* getspent_r.c */
539 int str2spwd(const char *, int, void *, char *, int);

541 /* getsubopt.c */
542 int getsubopt(char **optionsp, char *const *tokens, char **valuep);

544 /* gettxt.c */
545 char *gettext(const char *msg_id, const char *dflt_str);

547 /* getusershell.c */
548 char *getusershell(void);
549 void endusershell(void);
550 void setusershell(void);

552 /* getut.c */
553 struct utmp *getutent(void);
554 struct utmp *getutid(const struct utmp *entry);
555 struct utmp *getutline(const struct utmp *entry);
556 struct utmp *pututline(const struct utmp *entry);
557 void setutent(void);
558 void endutent(void);
559 int utmpname(const char *newfile);
560 void updwtmp(const char *file, struct utmp *ut);
561 void getutmp(const struct utmpx *utx, struct utmp *ut);
562 void getutmpx(const struct utmp *ut, struct utmpx *utx);
563 struct utmp *makeut(struct utmp *utmp);

565 /* getutx.c */
566 struct utmpx *getutxent(void);
567 struct utmpx *getutxid(const struct utmpx *entry);
568 struct utmpx *getutxline(const struct utmpx *entry);
569 struct utmpx *pututxline(const struct utmpx *entry);
570 void setutxent(void);
571 void endutxent(void);
572 int utmpxname(const char *newfile);
573 void updwtmpx(const char *filex, struct utmpx *utx);
574 struct utmpx *makeutx(const struct utmpx *utmp);
575 struct utmpx *modutx(const struct utmpx *utp);

577 /* getvfsent.c */
578 int getvfsspec(FILE *fd, struct vfstab *vp, char *special);
579 int getvfile(FILE *fd, struct vfstab *vp, char *mountp);
580 int getvfscopy(FILE *fd, struct vfstab *vgetp, struct vfstab *vrefp);
581 int getvfscopy(FILE *fd, struct vfstab *vp);

583 /* getwd.c */
584 char *getwd(char *pathname);

586 /* getwidth.c */
587 void getwidth(eucwidth_t *eucstruct);

```

```

589 /* hsearch.c */
590 int hcreate(size_t size);
591 void hdestroy(void);
592 ENTRY *hsearch(ENTRY item, ACTION action);

594 /* iconv.c */
595 size_t iconv(iconv_t cd, const char **_RESTRICT_KYWD inbuf,
596             size_t *_RESTRICT_KYWD inbytesleft, char **_RESTRICT_KYWD outbuf,
597             size_t *_RESTRICT_KYWD outbytesleft);
598 int iconv_close(iconv_t cd);
599 iconv_t iconv_open(const char *tocode, const char *fromcode);

601 /* imaxabs.c */
602 intmax_t imaxabs(intmax_t j);

604 /* imaxdiv.c */
605 imaxdiv_t imaxdiv(intmax_t numer, intmax_t denom);

607 /* index.c */
608 char *index(const char *sp, int c);

610 /* initgroups.c */
611 int initgroups(const char *uname, gid_t agroup);

613 /* inetgr.c */
614 int inetgr(const char *group, const char *machine, const char *name,
615            const char *domain);

617 /* insque.c */
618 void insque(void *elem, void *pred);
619 void remque(void *elem);

621 /* isaexec.c */
622 int isaexec(const char *, char *const *, char *const *);

624 /* isastream.c */
625 int isastream(int fd);

627 /* isatty.c */
628 int isatty(int f);

630 /* killpg.c */
631 int killpg(pid_t pgrp, int sig);

633 /* l64a.c */
634 char *l64a(long lg);

636 /* lckpddf.c */
637 int lckpddf(void);
638 int ulckpddf(void);

640 /* lfind.c */
641 void * lfind(const void *ky, const void *bs, size_t *nel,
642             size_t width, int (*compar)());

644 /* localeconv.c */
645 struct lconv *localeconv(void);

647 /* lsearch.c */
648 void * lsearch(const void *ky, void *bs, size_t *nel,
649              size_t width, int (*compar)());

651 /* madvise.c */
652 int madvise(caddr_t addr, size_t len, int advice);

```

```

654 /* malloc.c */
655 void *malloc(size_t size);
656 void *realloc(void *old, size_t size);
657 void free(void *old);

659 /* mbstowcs.c */
660 size_t mbstowcs(wchar_t *_RESTRICT_KYWD pwcs, const char *_RESTRICT_KYWD s,
661               size_t n);

663 /* mbtowc.c */
664 int mbtowc(wchar_t *_RESTRICT_KYWD wchar, const char *_RESTRICT_KYWD s,
665           size_t n);
666 int mblen(const char *s, size_t n);

668 /* memalign.c */
669 void *memalign(size_t align, size_t nbytes);

671 /* memccpy.c */
672 void *memccpy(void *_RESTRICT_KYWDs, const void *_RESTRICT_KYWD s0, int c,
673             size_t n);

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

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

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

684 /* memmove.c */
685 void *memmove(void *s, const void *s0, size_t n);

687 /* memset.c */
688 void *memset(void *sp1, int c, size_t n);

690 /* mkdev.c */
691 dev_t __makedev(const int version, const major_t majdev,
692               const minor_t mindev);
693 major_t __major(const int version, const dev_t devnum);
694 minor_t __minor(const int version, const dev_t devnum);

696 /* mkfifo.c */
697 int mkfifo(const char *path, mode_t mode);

699 /* mktemp.c */
700 char *mktemp(char *as);

702 /* mlock.c */
703 int mlock(caddr_t addr, size_t len);

705 /* mlockall.c */
706 int mlockall(int flags);

708 /* mon.c */
709 void monitor(int (*alowpc)(), int (*ahighpc)(), WORD *buffer,
710             size_t bufsize, size_t nfunc);

712 /* msync.c */
713 int msync(caddr_t addr, size_t len, int flags);

715 /* munlock.c */
716 int munlock(caddr_t addr, size_t len);

718 /* munlockall.c */
719 int munlockall(void);

```

```

721 /* ndbm.c */
722 void dbm_setdefwrite(DBM *db);
723 int dbm_flush(DBM *db);
724 int dbm_flushpag(DBM *db);
725 DBM *dbm_open(const char *file, int flags, mode_t mode);
726 void dbm_close(DBM *db);
727 int dbm_close_status(DBM *db);
728 datum dbm_fetch(DBM *db, datum key);
729 int dbm_delete(DBM *db, datum key);
730 int dbm_store(DBM *db, datum key, datum dat, int replace);
731 datum dbm_firstkey(DBM *db);
732 datum dbm_nextkey(DBM *db);
733 datum dbm_do_nextkey(DBM *db, datum inkey);

735 /* new_list.c */

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

740 /* nl_langinfo.c */
741 char *nl_langinfo(nl_item item);

743 /* opendir.c */
744 DIR *opendir(const char *filename);

746 /* opt_data.c */

748 /* perror.c */
749 void perror(const char *s);

751 /* pipe.c */
752 int pipe(int *fds);

754 /* psiginfo.c */
755 void psiginfo(siginfo_t *sip, char *s);

757 /* psignal.c */
758 void psignal(int sig, const char *s);

760 /* pt.c */
761 char *ptsname(int fd);
762 int unlockpt(int fd);
763 int grantpt(int fd);

765 /* putenv.c */
766 int putenv(char *change);
767 int setenv(const char *envname, const char *envval, int overwrite);
768 int unsetenv(const char *name);

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

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

776 /* raise.c */
777 int raise(int sig);

779 /* rand.c */
780 void srand(unsigned x);
781 int rand(void);
782 int rand_r(unsigned int *);

784 /* random.c */
785 void srandom(unsigned x);

```

```

786 char *initstate(unsigned seed, char *arg_state, size_t n);
787 char *setstate(const char *arg_state);
788 long random(void);

790 /* rctlops.c */
791 int rctl_walk(int (*callback)(const char *, void *), void *walk_data);
792 hrttime_t rctlblk_get_firing_time(rctlblk_t *rblk);
793 uint_t rctlblk_get_global_action(rctlblk_t *rblk);
794 uint_t rctlblk_get_global_flags(rctlblk_t *rblk);
795 uint_t rctlblk_get_local_action(rctlblk_t *rblk, int *signalp);
796 uint_t rctlblk_get_local_flags(rctlblk_t *rblk);
797 id_t rctlblk_get_recipient_pid(rctlblk_t *rblk);
798 rctl_priv_t rctlblk_get_privilege(rctlblk_t *rblk);
799 rctl_qty_t rctlblk_get_value(rctlblk_t *rblk);
800 void rctlblk_set_local_action(rctlblk_t *rblk, uint_t action, int signal);
801 void rctlblk_set_local_flags(rctlblk_t *rblk, uint_t flags);
802 void rctlblk_set_privilege(rctlblk_t *rblk, rctl_priv_t priv);
803 void rctlblk_set_value(rctlblk_t *rblk, rctl_qty_t val);
804 size_t rctlblk_size(void);

806 /* readdir.c */
807 struct dirent *readdir(DIR *dirp);

809 /* realpath.c */
810 char *realpath(const char *_RESTRICT_KYWD raw, char *_RESTRICT_KYWD canon);

812 /* regex.c */
813 char *re_comp(const char *sp);
814 int re_exec(const char *p1);

816 /* rindex.c */
817 char *rindex(const char *sp, int c);

819 /* rename.c */
820 int remove(const char *filename);
821 int rename(const char *old, const char *new);

823 /* rewinddir.c */
824 #undef rewinddir
825 void rewinddir(DIR *dirp);

827 /* scandir.c */
828 int alphasort(const struct dirent **, const struct dirent **);
829 int scandir(const char *dirname, struct dirent *(*namelist[]),
830             int (*select)(const struct dirent *),
831             int (*dcomp)(const struct dirent **, const struct dirent **));

833 /* scrwidth.c */
834 int scrwidth(wchar_t c);

836 /* seekdir.c */
837 void seekdir(DIR *dirp, long loc);

839 /* select.c */
840 int pselect(int nfd,
841             fd_set * _RESTRICT_KYWD readfds,
842             fd_set * _RESTRICT_KYWD writefds,
843             fd_set * _RESTRICT_KYWD errorfds,
844             const struct timespec * _RESTRICT_KYWD timeout,
845             const sigset_t * _RESTRICT_KYWD sigmask);
846 int select(int nfd,
847            fd_set * _RESTRICT_KYWD readfds,
848            fd_set * _RESTRICT_KYWD writefds,
849            fd_set * _RESTRICT_KYWD errorfds,
850            struct timeval * _RESTRICT_KYWD timeout);

```

```

852 /* setlocale.c */
853 char *setlocale(int cat, const char *loc);

855 /* setpriority.c */
856 int getpriority(int which, id_t who);
857 int setpriority(int which, id_t who, int prio);

859 /* settimeofday.c */
860 int settimeofday(struct timeval *tp, void *);

862 /* sigflag.c */
863 int sigflag(int sig, int flag, int on);

865 /* siglist.c */

867 /* sigsend.c */
868 int sigsend(idtype_t idtype, id_t id, int sig);

870 /* sigsetops.c */
871 int sigfillset(sigset_t *set);
872 int sigemptyset(sigset_t *set);
873 int sigaddset(sigset_t *set, int sig);
874 int sigdelset(sigset_t *set, int sig);
875 int sigismember(const sigset_t *set, int sig);

877 /* sleeps.c */
878 unsigned sleep(unsigned sleep_tm);

880 /* ssignal.c */
881 int (*ssignal(int sig, int (*fn)()) ())();
882 int gsignal(int sig);

884 /* str2id.c */

886 /* str2sig.c */
887 int str2sig(const char *s, int *sigp);
888 int sig2str(int i, char *s);

890 /* strcat.c */
891 char *strcat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

893 /* strchr.c */
894 char *strchr(const char *sp, int c);

896 /* strcmp.c */
897 int strcmp(const char *s1, const char *s2);

899 /* strcpy.c */
900 char *strcpy(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2);

902 /* strcspn.c */
903 size_t strcspn(const char *string, const char *charset);

905 /* strdup.c */
906 char *strdup(const char *s1);

908 /* strerror.c */
909 char *strerror(int errnum);
910 int strerror_r(int errnum, char *strerrbuf, size_t buflen);

912 /* strftime.c */
913 size_t strftime(char * _RESTRICT_KYWD s, size_t maxsize,
914                const char * _RESTRICT_KYWD format,
915                const struct tm * _RESTRICT_KYWD tm);

917 /* strlen.c */

```

```

918 size_t strlen(const char *s);

920 /* strcat.c */
921 char *strncat(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2, size_t n);

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

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

929 /* strpbrk.c */
930 char *strpbrk(const char *string, const char *brkset);

932 /* strrchr.c */
933 char *strrchr(const char *sp, int c);

935 /* strsep.c */
936 char *strsep(char **stringp, const char *delim);

938 /* strspn.c */
939 size_t strspn(const char *string, const char *charset);

941 /* strstr.c */
942 char *strstr(const char *as1, const char *as2);

944 /* strtod.c */
945 double strtod(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
946 float strtodf(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);
947 long double strtold(const char *_RESTRICT_KYWD cp, char **_RESTRICT_KYWD ptr);

949 /* strtoumax.c */
950 intmax_t strtoumax(const char *_RESTRICT_KYWD nptr,
951                   char **_RESTRICT_KYWD endptr, int base);

953 /* strtok.c */
954 char *strtok(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD sepset);
955 char *strtok_r(char *_RESTRICT_KYWD, const char *_RESTRICT_KYWD,
956               char **_RESTRICT_KYWD);

958 /* strtol.c */
959 long strtol(const char *_RESTRICT_KYWD str, char **_RESTRICT_KYWD nptr,
960             int base);

962 /* strtoul.c */
963 unsigned long strtoul(const char *_RESTRICT_KYWD str,
964                       char **_RESTRICT_KYWD nptr, int base);

966 /* strtoumax.c */
967 uintmax_t strtoumax(const char *_RESTRICT_KYWD nptr,
968                     char **_RESTRICT_KYWD endptr, int base);

970 /* strxfrm.c */
971 size_t strxfrm(char *_RESTRICT_KYWD s1, const char *_RESTRICT_KYWD s2,
972               size_t n);
973 int strcoll(const char *s1, const char *s2);

975 /* swab.c */
976 void swab(const char *_RESTRICT_KYWD from, char *_RESTRICT_KYWD to, ssize_t n);

978 /* swapctl.c */
979 int swapctl(int cmd, void *arg);

981 /* sysconf.c */
982 long sysconf(int name);

```

```

984 /* syslog.c */
985 /* VARARGS2 */
986 void syslog(int pri, const char *fmt, ...);
987 void vsyslog(int pri, const char *fmt, va_list ap);
988 void openlog(const char *ident, int logstat, int logfac);
989 void closelog(void);
990 int setlogmask(int pmask);

992 /* tcdrain.c */
993 int tcdrain(int fildes);

995 /* tcflow.c */
996 int tcflow(int fildes, int action);

998 /* tcflush.c */
999 int tcflush(int fildes, int queue_selector);

1001 /* tcgetattr.c */
1002 int tcgetattr(int fildes, struct termios *termios_p);

1004 /* tcgetpgrp.c */
1005 pid_t tcgetpgrp(int fd);

1007 /* tcgetsid.c */
1008 pid_t tcgetsid(int fd);

1010 /* tcsendbreak.c */
1011 int tcsendbreak(int fildes, int duration);

1013 /* tcsetattr.c */
1014 int tcsetattr(int fildes, int optional_actions,
1015              const struct termios *termios_p);

1017 /* tcsetpgrp.c */
1018 int tcsetpgrp(int fd, pid_t pgrp);

1020 /* tell.c */
1021 long tell(int f);

1023 /* telldir.c */
1024 long telldir(DIR *dirp);

1026 /* tfind.c */
1027 void *tfind(const void *ky, void *const *rtp, int (*compar)());

1029 /* time_comm.c */
1030 struct tm *localtime(const time_t *timep);
1031 struct tm *localtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
1032 struct tm *gmtime(const time_t *clock);
1033 struct tm *gmtime_r(const time_t *_RESTRICT_KYWD, struct tm *_RESTRICT_KYWD);
1034 double difftime(time_t time1, time_t time0);
1035 time_t mktime(struct tm *timeptr);
1036 void _ltzset(time_t tim);
1037 void tzset(void);

1039 /* time_data.c */

1041 /* time_gdata.c */

1043 /* tolower.c */
1044 int tolower(int c);

1046 /* toupper.c */
1047 int toupper(int c);

1049 /* truncate.c */

```

```

1050 int ftruncate(int fildes, off_t len);
1051 int truncate(const char *path, off_t len);

1053 /* tsearch.c */
1054 void *tsearch(const void *ky, void **rtp, int (*compar)());
1055 void *tdelete(const void *ky, void **rtp, int (*compar)());
1056 void twalk(const void *rt, void (*action)());

1058 /* ttyname.c */
1059 char *ttyname(int f);
1060 char *ttyname_dev(dev_t rdev, char *buffer, size_t buflen);
1061 char *ttyname_r(int, char *, int);

1063 /* ttyslot.c */
1064 int ttyslot(void);

1066 /* ualarm.c */
1067 unsigned ualarm(unsigned usecs, unsigned reload);

1069 /* ulimit.c */
1070 /* VARARGS1 */
1071 long ulimit(int cmd, ...);

1073 /* scalls.c */
1074 int usleep(unsigned n);

1076 /* valloc.c */
1077 void *valloc(size_t size);

1079 /* waitpid.c */
1080 pid_t wait(int *stat_loc);
1081 pid_t waitpid(pid_t pid, int *stat_loc, int options);
1082 pid_t wait3(int *status, int options, struct rusage *rp);
1083 pid_t wait4(pid_t pid, int *status, int options, struct rusage *rusage);

1085 /* wcstombs.c */
1086 size_t wcstombs(char *_RESTRICT_KYWD s, const wchar_t *_RESTRICT_KYWD pwcs,
1087                size_t n);

1089 /* wctomb.c */
1090 int wctomb(char *s, wchar_t wchar);

1092 /* wdata.c */

1094 /* wisprint.c */
1095 int wisprint(wchar_t c);

1097 /* xgetwidth.c */
1098 void _xgetwidth(void);

1100 /*
1101  * /usr/src/lib/libc/port/intl routines
1102  */

1104 /* gettext.c */
1105 char *bindtextdomain(const char *domain, const char *binding);
1106 char *dcgettext(const char *domain, const char *msg_id, const int category);
1107 char *dgettext(const char *domain, const char *msg_id);
1108 char *gettext(const char *msg_id);
1109 char *textdomain(const char *domain);

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

1115 /* fprintf.c */

```

```

1116 /* VARARGS2 */
1117 int fprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format, ...);

1119 /* printf.c */
1120 /* VARARGS1 */
1121 int printf(const char *_RESTRICT_KYWD format, ...);

1123 /* snprintf.c */
1124 /* VARARGS2 */
1125 int snprintf(char *_RESTRICT_KYWD string, size_t n,
1126              const char *_RESTRICT_KYWD format, ...);

1128 /* sprintf.c */
1129 /* VARARGS2 */
1130 int sprintf(char *_RESTRICT_KYWD string,
1131             const char *_RESTRICT_KYWD format, ...);

1133 /* vfprintf.c */
1134 /* VARARGS2 */
1135 int vfprintf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD format,
1136             va_list);

1138 /* vprintf.c */
1139 /* VARARGS1 */
1140 int vprintf(const char *_RESTRICT_KYWD format, va_list);

1142 /* vsnprintf.c */
1143 /* VARARGS2 */
1144 int vsnprintf(char *_RESTRICT_KYWD string, size_t n,
1145              const char *_RESTRICT_KYWD format, va_list);

1147 /* vsprintf.c */
1148 /* VARARGS2 */
1149 int vsprintf(char *_RESTRICT_KYWD string, const char *_RESTRICT_KYWD format,
1150             va_list);

1152 /*
1153  * /usr/src/lib/libc/port/regex routines
1154  */

1156 /* glob.c */
1157 extern int glob(const char *restrict pattern, int flags,
1158               int (*errfunc)(const char *epath, int eerrno), glob_t *restrict pglob);
1159 extern void globfree(glob_t *pglob);

1161 /* regex.c */
1162 char *regcomp(const char *regexp, const char *stringp, ...);
1163 #ifdef __locl
1164 #undef __locl
1165 #endif
1166 char *__locl;

1168 /* regcmp.c */
1169 char *regcmp(const char *regexp, ...);
1170 #ifdef __i_size
1171 #undef __i_size
1172 #endif
1173 int __i_size;

1175 /*
1176  * /usr/src/lib/libc/port/stdio routines
1177  */

1179 /* _filbuf.c */
1180 int _filbuf(FILE *iop);

```



```

1182 /* _flsbuf.c */
1183 int _flsbuf(int ch, FILE *iop);

1185 /* _wrtchk.c */
1186 int _wrtchk(FILE *iop);

1188 /* clearerr.c */
1189 void clearerr(FILE *iop);

1191 /* ctermid.c */
1192 char *ctermid(char *s);
1193 char *ctermid_r(char *s);

1195 /* cuserid.c */
1196 char *cuserid(char *s);

1198 /* data.c */

1200 /* doscan.c */
1201 int _doscan(FILE *iop, const char *fmt, va_list va_alist);

1203 /* fdopen.c */
1204 FILE *fdopen(int fd, const char *type);

1206 /* feof.c */
1207 int feof(FILE *iop);

1209 /* ferror.c */
1210 int ferror(FILE *iop);

1212 /* fgetc.c */
1213 int fgetc(FILE *iop);

1215 /* fgets.c */
1216 char *fgets(char *_RESTRICT_KYWD buf, int size, FILE *_RESTRICT_KYWD iop);

1218 /* fileno.c */
1219 int _fileno(FILE *iop);

1221 /* flush.c */
1222 void _cleanup(void);
1223 FILE *_findiop(void);
1224 typedef unsigned char Uchar;
1225 void _setbufend(FILE *iop, Uchar *end);
1226 Uchar *_realbufend(FILE *iop);
1227 void _bufsync(FILE *iop, Uchar *bufend);
1228 int _xflsbuf(FILE *iop);
1229 int fflush(FILE *iop);
1230 int fclose(FILE *iop);

1232 /* fopen.c */
1233 FILE *fopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type);
1234 FILE *freopen(const char *_RESTRICT_KYWD name, const char *_RESTRICT_KYWD type,
1235 FILE *_RESTRICT_KYWD iop);

1237 /* fpos.c */
1238 int fgetpos(FILE *_RESTRICT_KYWD stream, fpos_t *_RESTRICT_KYWD pos);
1239 int fsetpos(FILE *stream, const fpos_t *pos);

1241 /* fputc.c */
1242 int fputc(int ch, FILE *iop);

1244 /* fputs.c */
1245 int fputs(const char *_RESTRICT_KYWD ptr, FILE *_RESTRICT_KYWD iop);

1247 /* fread.c */

```

```

1248 size_t fread(void *_RESTRICT_KYWD ptr, size_t size, size_t count,
1249 FILE *_RESTRICT_KYWD iop);

1251 /* fseek.c */
1252 int fseek(FILE *iop, long offset, int ptrname);

1254 /* ftell.c */
1255 long ftell(FILE *iop);

1257 /* fwrite.c */
1258 size_t fwrite(const void *_RESTRICT_KYWD ptr1, size_t size, size_t count,
1259 FILE *_RESTRICT_KYWD iop);

1261 /*getc.c */
1262 int getc(FILE *iop);

1264 /* getchar.c */
1265 int getchar(void);

1267 /* getpass.c */
1268 char *getpass(const char *prompt);

1270 /* getpass.c */
1271 char *getpassphrase(const char *prompt);

1273 /* gets.c */
1274 char *gets(char *buf);

1276 /* getw.c */
1277 int getw(FILE *stream);

1279 /* popen.c */
1280 FILE *popen(const char *cmd, const char *mode);
1281 int pclose(FILE *ptr);

1283 /*putc.c */
1284 int putc(int ch, FILE *iop);

1286 /* putchar.c */
1287 int putchar(int ch);

1289 /* puts.c */
1290 int puts(const char *ptr);

1292 /* putw.c */
1293 int putw(int w, FILE *stream);

1295 /* rewind.c */
1296 void rewind(FILE *iop);

1298 /* scanf.c */
1299 /* VARARGS1 */
1300 int scanf(const char *_RESTRICT_KYWD fmt, ...);

1302 /* VARARGS2 */
1303 int fscanf(FILE *_RESTRICT_KYWD iop, const char *_RESTRICT_KYWD fmt, ...);

1305 /* VARARGS2 */
1306 int sscanf(const char *_RESTRICT_KYWD str, const char *_RESTRICT_KYWD fmt, ...);

1308 /* setbuf.c */
1309 void setbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf);

1311 /* setvbuf.c */
1312 int setvbuf(FILE *_RESTRICT_KYWD iop, char *_RESTRICT_KYWD abuf, int type,
1313 size_t size);

```

```

1315 /* system.c */
1316 int system(const char *s);

1318 /* tmpnam.c */
1319 char *tmpnam(const char *dir, const char *pfx);

1321 /* tmpfile.c */
1322 FILE *tmpfile(void);

1324 /* tmpnam.c */
1325 char *tmpnam(char *s);
1326 char *tmpnam_r(char *);

1328 /* ungetc.c */
1329 int ungetc(int c, FILE *iop);

1331 /*
1332  * /usr/src/lib/libc/port/sys routines
1333  */

1335 /* exactsys.c */
1336 size_t getacct(idtype_t idtype, id_t id, void *buf, size_t bufsize);
1337 int putacct(idtype_t idtype, id_t id, void *buf, size_t bufsize, int flags);
1338 int wracct(idtype_t idtype, id_t id, int flags);

1340 /* execl.c */
1341 /* VARARGS1 */
1342 int execl(const char *name, const char *, ...);

1344 /* execl_e.c */
1345 int execl_e(const char *, const char *file, ...);

1347 /* execv.c */
1348 int execv(const char *file, char *const *argv);

1350 /* lockf.c */
1351 int lockf(int fildes, int function, off_t size);

1353 /* meminfosys.c */
1354 int meminfo(const uint64_t *inaddr, int addr_count, const uint_t *info_req,
1355             int info_count, uint64_t *outdata, uint_t *validity);

1357 /* msgsys.c */
1358 int msgget(key_t key, int msgflg);
1359 int msqctl(int msqid, int cmd, struct msqid_ds *buf);
1360 ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);
1361 int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);

1363 /* nfssys.c */
1364 /*
1365 int exportfs(char *dir, struct export *ep);
1366 int nfs_getfh(char *path, fh_t *fhp);
1367 int nfssvc(int fd);
1368 */

1370 /* psetsys.c */
1371 int pset_create(psetid_t *npset);
1372 int pset_destroy(psetid_t pset);
1373 int pset_assign(psetid_t pset, processorid_t cpu, psetid_t *opset);
1374 int pset_assign_forced(psetid_t pset, processorid_t cpu, psetid_t *opset);
1375 int pset_info(psetid_t pset, int *type, u_int *numcpus, processorid_t *cpulist);
1376 int pset_bind(psetid_t pset, idtype_t idtype, id_t id, psetid_t *opset);
1377 int pset_bind_lwp(psetid_t pset, id_t id, pid_t, psetid_t *opset);
1378

```

```

1380 /* rctlsys.c */
1381 int getrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
1382             int flags);
1383 int setrctl(const char *name, rctlblk_t *old_rblk, rctlblk_t *new_rblk,
1384             int flags);
1385 /* (private functions) */
1386 int setprojctl(const char *name, rctlblk_t *new_rblk, size_t size, int flags);
1387 int rctlctl(const char *, rctlblk_t *, int);
1388 size_t rctlolist(char *, size_t);

1391 /* semsys.c */
1392 int semctl(int semid, int semnum, int cmd, ...);
1393 int semget(key_t key, int nsems, int semflg);
1394 int semop(int semid, struct sembuf *sops, size_t nsops);

1396 /* shmsys.c */
1397 void *shmat(int shmid, const void *shmaddr, int shmflg);
1398 int shmctl(int shmid, int cmd, struct shmids *buf);
1399 #if defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4)
1400 int shmdt(const void *);
1401 #else
1402 int shmdt(char *);
1403 #endif /* defined(_XOPEN_SOURCE) && (_XOPEN_VERSION - 0 == 4) */
1404 int shmget(key_t key, size_t size, int shmflg);

1406 /* tasksys.c */
1407 taskid_t settaskid(projid_t project, uint_t flags);
1408 taskid_t gettaskid(void);
1409 projid_t getprojid(void);

1411 /*
1412  * /usr/src/lib/libc/port/widec routines
1413  */

1415 /* fgetws.c */
1416 wchar_t *fgetws(wchar_t *RESTRICT_KYWD ptr, int size,
1417                 FILE *RESTRICT_KYWD iop);

1419 /* fputwc.c */
1420 wint_t fputwc(wint_t wc, FILE *iop);
1421 wint_t putwc(wint_t wc, FILE *iop);

1423 /* fputws.c */
1424 int fputws(const wchar_t *RESTRICT_KYWD ptr, FILE *RESTRICT_KYWD iop);

1426 /* getwchar.c */
1427 wint_t getwchar(void);

1429 /* getwidth.c */
1430 void getwidth(eucwidth_t *eucstruct);

1432 /* getws.c */
1433 wchar_t *getws(wchar_t *ptr);

1435 /* iswctype.c */
1436 int iswctype(wint_t wc, wctype_t charclass);
1437 int iswalph(wint_t c);
1438 int iswupper(wint_t c);
1439 int iswlower(wint_t c);
1440 int iswdigit(wint_t c);
1441 int iswxdigit(wint_t c);
1442 int iswalnum(wint_t c);
1443 int iswspace(wint_t c);
1444 int iswpunct(wint_t c);
1445 int iswprint(wint_t c);

```

```

1446 int iswgraph(wint_t c);
1447 int iswcntrl(wint_t c);
1448 int isphonogram(wint_t c);
1449 int isideogram(wint_t c);
1450 int isenglish(wint_t c);
1451 int isnumber(wint_t c);
1452 int isspecial(wint_t c);

1454 /* libwcollate.c */

1456 /* putwchar.c */
1457 wint_t putwchar(wint_t c);

1459 /* putws.c */
1460 int putws(const wchar_t *ptr);

1462 /* scrwidth.c */

1464 /* strtows.c */
1465 wchar_t *strtows(wchar_t *s1, char *s2);
1466 char *wstostr(char *s1, wchar_t *s2);

1468 /* trwctype.c */
1469 wint_t towupper(wint_t c);
1470 wint_t towlower(wint_t c);

1472 /* ungetwc.c */
1473 wint_t ungetwc(wint_t wc, FILE *iop);

1475 /* wcollate.c */
1476 size_t wcsxfrm(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1477               size_t n);
1478 int wscoll(const wchar_t *s1, const wchar_t *s2);

1480 /* wcsftime.c */
1481 #if !defined(__amd64) /* XX64 - fix me */
1482 size_t wcsftime(wchar_t *wcs, size_t maxsize,
1483                const char *format, const struct tm *timeptr);
1484 #endif /* __amd64 */

1486 /* wcstring.c */
1487 wint_t fgetwc(FILE *iop);
1488 wint_t getwc(FILE *iop);
1489 int wcwidth(wchar_t wc);
1490 int wcswidth(const wchar_t *pwcs, size_t n);

1492 /* wcsvcs.c */
1493 wchar_t *wcsvcs(const wchar_t *ws1, const wchar_t *ws2);

1495 /* wcsxfrm.c - empty file! */

1497 /* wcsxfrm.xpg4.c */

1499 /* wisprint.c */
1500 int wisprint(wchar_t c);

1502 /* wscasecmp.c */
1503 int wscasecmp(const wchar_t *s1, const wchar_t *s2);

1505 /* wscat.c */
1506 wchar_t *wscat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
1507 wchar_t *wscat(wchar_t *s1, const wchar_t *s2);

1509 /* wschr.c */
1510 wchar_t *wschr(const wchar_t *sp, wchar_t c);
1511 wchar_t *wschr(const wchar_t *sp, wchar_t c);

```

```

1513 /* wscmp.c */
1514 int wscmp(const wchar_t *s1, const wchar_t *s2);
1515 int wscmp(const wchar_t *s1, const wchar_t *s2);

1517 /* wscol.c */
1518 int wscol(const wchar_t *s1);

1520 /* wscpy.c */
1521 wchar_t *wscpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2);
1522 wchar_t *wscpy(wchar_t *s1, const wchar_t *s2);

1524 /* wscspn.c */
1525 size_t wscspn(const wchar_t *string, const wchar_t *charset);
1526 size_t wscspn(const wchar_t *string, const wchar_t *charset);

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

1531 /* wslen.c */
1532 size_t wslen(const wchar_t *s);
1533 size_t wslen(const wchar_t *s);

1535 /* wsnccasecmp.c */
1536 int wsnccasecmp(const wchar_t *s1, const wchar_t *s2, size_t n);

1538 /* wsnccat.c */
1539 wchar_t *wsnccat(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1540                 size_t n);
1541 wchar_t *wsnccat(wchar_t *s1, const wchar_t *s2, size_t n);

1543 /* wsnccmp.c */
1544 int wsnccmp(const wchar_t *s1, const wchar_t *s2, size_t n);
1545 int wsnccmp(const wchar_t *s1, const wchar_t *s2, size_t n);

1547 /* wsnncpy.c */
1548 wchar_t *wsnncpy(wchar_t *_RESTRICT_KYWD s1, const wchar_t *_RESTRICT_KYWD s2,
1549                 size_t n);
1550 wchar_t *wsnncpy(wchar_t *s1, const wchar_t *s2, size_t n);

1552 /* wspbrc.c */
1553 wchar_t *wspbrc(const wchar_t *string, const wchar_t *brkset);
1554 wchar_t *wspbrc(const wchar_t *string, const wchar_t *brkset);

1556 /* wsprintf.c */
1557 int wsprintf(wchar_t *wstring, const char *format, ...);

1559 /* wsrchr.c */
1560 wchar_t *wsrchr(const wchar_t *sp, wchar_t c);
1561 wchar_t *wsrchr(const wchar_t *sp, wchar_t c);

1563 /* wsscang.c */
1564 int wsscang(wchar_t *s, const char *format, ...);

1566 /* wssize.c */

1568 /* wsspnc.c */
1569 size_t wsspnc(const wchar_t *string, const wchar_t *charset);
1570 size_t wsspnc(const wchar_t *string, const wchar_t *charset);

1572 /* wstod.c */
1573 double wstod(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
1574 float wstof(const wchar_t *_RESTRICT_KYWD cp, wchar_t **_RESTRICT_KYWD ptr);
1575 long double wstold(const wchar_t *_RESTRICT_KYWD cp,
1576                   wchar_t **_RESTRICT_KYWD ptr);
1577 double wstod(const wchar_t *cp, wchar_t **ptr);

```

```

1579 /* wstok.c */
1580 #if !defined(__amd64)          /* XX64 - fix me */
1581 wchar_t *wcstok(wchar_t *string, const wchar_t *sepset);
1582 wchar_t *wstok(wchar_t *string, const wchar_t *sepset);
1583 #endif /* __amd64 */

1585 /* wcstol.c */
1586 long wcstol(const wchar_t *_RESTRICT_KYWD str, wchar_t **_RESTRICT_KYWD ptr,
1587             int base);
1588 long long wcstoll(const wchar_t *_RESTRICT_KYWD str,
1589                  wchar_t **_RESTRICT_KYWD ptr, int base);

1591 /* wcstoul.c */
1592 unsigned long wcstoul(const wchar_t *_RESTRICT_KYWD str,
1593                       wchar_t **_RESTRICT_KYWD ptr, int base);
1594 unsigned long long wcstoull(const wchar_t *_RESTRICT_KYWD str,
1595                              wchar_t **_RESTRICT_KYWD ptr, int base);

1597 /* wcstoimax.c */
1598 intmax_t wcstoimax(const wchar_t *_RESTRICT_KYWD nptr,
1599                   wchar_t **_RESTRICT_KYWD endptr, int base);
1600 uintmax_t wcstoumax(const wchar_t *_RESTRICT_KYWD nptr,
1601                     wchar_t **_RESTRICT_KYWD endptr, int base);

1603 /* wstol.c */
1604 long wstol(const wchar_t *str, wchar_t **ptr, int base);

1606 /* wstoll.c */
1607 long long wstoll(const wchar_t *str, wchar_t **ptr, int base);
1608 long long watoll(const wchar_t *p);

1610 /* wsxfrm.c */
1611 size_t wsxfrm(wchar_t *s1, const wchar_t *s2, size_t n);
1612 int wscoll(const wchar_t *s1, const wchar_t *s2);

1614 /*
1615  * /usr/src/lib/libc/port/gen/event_port.c
1616  */
1617 int port_dispatch(int port, int flags, int source, int events, uintptr_t object,
1618                  void *user);

1620 /*
1621  * /usr/src/lib/libc/$MACH/gen routines
1622  */

1624 /* alloca.s */

1626 void *__builtin_alloca(size_t);

1628 /*
1629  * modctl(int arg, ...) and utssys(...) are not available from a header
1630  * file, but our utilities which make use of it should be able to be
1631  * lint clean.
1632  */
1633 int modctl(int arg, ...);
1634 int utssys(void *buf, int arg, int type, void *outbp);

1637 typedef float single;
1638 typedef unsigned extended[3];
1639 typedef long double quadruple;
1640 typedef unsigned fp_exception_field_type;

1642 typedef char decimal_string[512];

```

```

1644 enum fp_class_type {
1645     fp_zero = 0,
1646     fp_subnormal = 1,
1647     fp_normal = 2,
1648     fp_infinity = 3,
1649     fp_quiet = 4,
1650     fp_signaling = 5
1651 };

1653 enum fp_direction_type {
1654     fp_nearest = 0,
1655     fp_tozero = 1,
1656     fp_positive = 2,
1657     fp_negative = 3
1658 };

1660 typedef struct {
1661     enum fp_class_type fpclass;
1662     int sign;
1663     int exponent;
1664     decimal_string ds;
1665     int more;
1666     int ndigits;
1667 } decimal_record;

1669 enum decimal_form {
1670     fixed_form,
1671     floating_form
1672 };

1674 typedef struct {
1675     enum fp_direction_type rd;
1676     enum decimal_form df;
1677     int ndigits;
1678 } decimal_mode;

1680 enum decimal_string_form {
1681     invalid_form,
1682     whitespace_form,
1683     fixed_int_form,
1684     fixed_intdot_form,
1685     fixed_dotfrac_form,
1686     fixed_intdotfrac_form,
1687     floating_int_form,
1688     floating_intdot_form,
1689     floating_dotfrac_form,
1690     floating_intdotfrac_form,
1691     inf_form,
1692     infinity_form,
1693     nan_form,
1694     nanstring_form
1695 };

1697 typedef int sigfpe_code_type;

1699 typedef void (*sigfpe_handler_type)();

1701 extern sigfpe_handler_type sigfpe(sigfpe_code_type, sigfpe_handler_type);

1703 extern void single_to_decimal(single *, decimal_mode *, decimal_record *,
1704                               fp_exception_field_type *);

1706 extern void double_to_decimal(double *, decimal_mode *, decimal_record *,
1707                               fp_exception_field_type *);
1708 extern void extended_to_decimal(extended *, decimal_mode *,
1709                                 decimal_record *, fp_exception_field_type *);

```

```

1710 extern void quadruple_to_decimal(quadruple *, decimal_mode *,
1711     decimal_record *, fp_exception_field_type *);
1712 extern void decimal_to_single(single *, decimal_mode *, decimal_record *,
1713     fp_exception_field_type *);
1714 extern void decimal_to_double(double *, decimal_mode *, decimal_record *,
1715     fp_exception_field_type *);
1716 extern void decimal_to_extended(extended *, decimal_mode *,
1717     decimal_record *, fp_exception_field_type *);
1718 extern void decimal_to_quadruple(quadruple *, decimal_mode *,
1719     decimal_record *, fp_exception_field_type *);
1720 extern void string_to_decimal(char **, int, int, decimal_record *,
1721     enum decimal_string_form *, char **);
1722 extern void func_to_decimal(char **, int, int, decimal_record *,
1723     enum decimal_string_form *, char **,
1724     int (*)(void), int *, int (*)(int));
1725 extern void file_to_decimal(char **, int, int, decimal_record *,
1726     enum decimal_string_form *, char **,
1727     FILE *, int *);
1728 extern char *seconvert(single *, int, int *, int *, char *);
1729 extern char *sfconvert(single *, int, int *, int *, char *);
1730 extern char *sgconvert(single *, int, int, char *);
1731 extern char *econvert(double, int, int *, int *, char *);
1732 extern char *fconvert(double, int, int *, int *, char *);
1733 extern char *gconvert(double, int, int, char *);
1734 extern char *qeconvert(quadruple *, int, int *, int *, char *);
1735 extern char *qfconvert(quadruple *, int, int *, int *, char *);
1736 extern char *qgconvert(quadruple *, int, int, char *);

1738 extern void __assert(const char *, const char *, int);

1740 extern int setjmp(jmp_buf);
1741 extern void longjmp(jmp_buf, int);
1742 extern int sigsetjmp(sigjmp_buf, int);
1743 extern void siglongjmp(sigjmp_buf, int);

1745 int uname(struct utsname *);
1746 int _uname(struct utsname *);

1748 int errno;
1749 int *__errno()
1750 { return (&errno); }

1752 extern int getloadavg(double [], int);

1754 extern long pcsample(uintptr_t [], long);

1756 int fstat(int, struct stat *);
1757 int stat(const char *_RESTRICT_KYWD, struct stat *_RESTRICT_KYWD);
1758 int lstat(const char *_RESTRICT_KYWD, struct stat *_RESTRICT_KYWD);
1759 int mknod(const char *, mode_t, dev_t);

1761 extern int __init_daemon_priv(int, uid_t uid, gid_t gid, ...);
1762 extern void __fini_daemon_priv(const char *, ...);
1763 extern int __init_suid_priv(int, ...);
1764 extern int __priv_bracket(priv_op_t);
1765 extern void __priv_relinquish(void);
1766 extern const char * __priv_getsetbynum(const void *, int);
1767 extern char * __priv_set_to_str(void *, const priv_set_t *, char, int);

1769 /* private interface to get the groups list for a certain user */
1770 int _getgroupsbymember(const char *, gid_t[], int, int);

1772 /* private interface for use only by java */
1773 volatile sc_shared_t *volatile *_thr_schedctl(void);

1775 /* private interface to unmount all autofs mounts */

```

```

1776 int _autofssys(enum autofssys_op, void *);

1778 /* label.c */
1779 extern int is_system_labeled(void);

1781 extern int uconv_u16tou32(const uint16_t *, size_t *, uint32_t *, size_t *,
1782     int);
1783 extern int uconv_u16tou8(const uint16_t *, size_t *, uchar_t *, size_t *, int);
1784 extern int uconv_u32tou16(const uint32_t *, size_t *, uint16_t *, size_t *,
1785     int);
1786 extern int uconv_u32tou8(const uint32_t *, size_t *, uchar_t *, size_t *, int);
1787 extern int uconv_u8tou16(const uchar_t *, size_t *, uint16_t *, size_t *, int);
1788 extern int uconv_u8tou32(const uchar_t *, size_t *, uint32_t *, size_t *, int);
1789 extern int u8_validate(char *, size_t, char **, int, int *);
1790 extern int u8_strcmp(const char *, const char *, size_t, int, size_t, int *);
1791 extern size_t u8_textprep_str(char *, size_t *, char *, size_t *, int, size_t,
1792     int *);

1794 /* private locale interfaces */
1795 wint_t __nextwctype(wint_t, wctype_t);
1796 int __iswruone(wint_t);

```

```

*****
54814 Sat Sep 28 12:17:12 2013
new/usr/src/lib/libc/port/mapfile-vers
1060 terminos missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 # Copyright 2010 Nexenta Systems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # Copyright (c) 2012 by Delphix. All rights reserved.
28 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
29 #
30 #
31 # MAPFILE HEADER START
32 #
33 # WARNING: STOP NOW. DO NOT MODIFY THIS FILE.
34 # Object versioning must comply with the rules detailed in
35 #
36 #     usr/src/lib/README.mapfiles
37 #
38 # You should not be making modifications here until you've read the most current
39 # copy of that file. If you need help, contact a gatekeeper for guidance.
40 #
41 # MAPFILE HEADER END
42 #
43 #
44 $mapfile_version 2
45 #
46 #
47 # All function names added to this or any other libc mapfile
48 # must be placed under the 'protected:' designation.
49 # The 'global:' designation is used *only* for data
50 # items and for the members of the malloc() family.
51 #
52 #
53 #
54 # README README README README README README: how to update this file
55 # 1) each version of Solaris/OpenSolaris gets a version number.
56 # (Actually since Solaris is actually a series of OpenSolaris releases
57 # we'll just use OpenSolaris for this exercise.)
58 #     OpenSolaris 2008.11 gets 1.23
59 #     OpenSolaris 2009.04 gets 1.24

```

```

60 #     etc.
61 # 2) each project integration uses a unique version number.
62 #     PSARC/2008/123 gets 1.24.1
63 #     PSARC/2008/456 gets 1.24.2
64 #     etc.
65 #
66 #
67 #
68 # Mnemonic conditional input identifiers:
69 #
70 # - amd64, i386, sparc32, sparcv9: Correspond to ISA subdirectories used to
71 #   hold per-platform code. Note however that we use 'sparc32' instead of
72 #   'sparc'. Since '_sparc' is predefined to apply to, all sparc platforms,
73 #   naming the 32-bit version 'sparc' would be too likely to cause errors.
74 #
75 # - lf64: Defined on platforms that offer the 32-bit largefile APIs
76 #
77 $if _ELF32
78 $add lf64
79 $endif
80 $if _sparc && _ELF32
81 $add sparc32
82 $endif
83 $if _sparc && _ELF64
84 $add sparcv9
85 $endif
86 $if _x86 && _ELF32
87 $add i386
88 $endif
89 $if _x86 && _ELF64
90 $add amd64
91 $endif
92 #
93 SYMBOL_VERSION ILLUMOS_0.6 {
94     protected:
95     cfmakeraw;
96 } ILLUMOS_0.5;
97 #
98 #endif /* ! codereview */
99 SYMBOL_VERSION ILLUMOS_0.5 { # common C++ ABI exit handlers
100     protected:
101     __cxa_atexit;
102     __cxa_finalize;
103 } ILLUMOS_0.4;
104 #
105 SYMBOL_VERSION ILLUMOS_0.4 { # Illumos additions
106     protected:
107     pipe2;
108     dup3;
109     mkostemp;
110     mkostemps;
111 #
112 $if lf64
113     mkostemp64;
114     mkostemps64;
115 $endif
116 } ILLUMOS_0.3;
117 #
118 SYMBOL_VERSION ILLUMOS_0.3 { # Illumos additions
119     protected:
120     assfail3;
121 } ILLUMOS_0.2;
122 #
123 SYMBOL_VERSION ILLUMOS_0.2 { # Illumos additions
124     protected:
125     posix_spawn_pipe_np;

```

```

126 } ILLUMOS_0.1;

128 SYMBOL_VERSION ILLUMOS_0.1 { # Illumos additions
129     protected:
130         timegm;
131 } SUNW_1.23;

133 SYMBOL_VERSION SUNW_1.23 { # SunOS 5.11 (Solaris 11)
134     global:
135         _nl_domain_bindings;
136         _nl_msg_cat_cntr;

138 $if _ELF32
139     dl_iterate_phdr { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
140 $elif sparcv9
141     dl_iterate_phdr { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
142 $elif amd64
143     dl_iterate_phdr { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
144 $else
145 $error unknown platform
146 $endif

148     protected:

150 $if sparc32
151     __align_cpy_1;
152 $endif

154     addrtoymstr;
155     aio_cancel;
156     aiocancel;
157     aio_error;
158     aio_fsync;
159     aio_read;
160     aioread;
161     aio_return;
162     aio_suspend;
163     aiowait;
164     aio_waitn;
165     aio_write;
166     aiowrite;
167     asprintf;
168     assfail;
169     backtrace;
170     backtrace_symbols;
171     backtrace_symbols_fd;
172     canonicalize_file_name;
173     clearenv;
174     clock_getres;
175     clock_gettime;
176     clock_nanosleep;
177     clock_settime;
178     daemon;
179     dirfd;
180     door_bind;
181     door_call;
182     door_create;
183     door_cred;
184     door_getparam;
185     door_info;
186     door_return;
187     door_revoke;
188     door_server_create;
189     door_setparam;
190     door_ucred;
191     door_unbind;

```

```

192     door_xcreate;
193     err;
194     errx;
195     faccessat;
196     fchmodat;
197     fcloseall;
198     fdatasync;
199     ffsl;
200     ffsl1;
201     fgetattr;
202     fls;
203     flsl;
204     flsl1;
205     forkallx;
206     forkx;
207     fsetattr;
208     getattrat;
209     getdelim;
210     getline;
211     get_nprocs;
212     get_nprocs_conf;
213     getprogname;
214     htonl;
215     htonll;
216     htons;
217     linkat;
218     lio_listio;
219     memmem;
220     mkdirat;
221     mkdtemp;
222     mkfifoat;
223     mknodat;
224     mkstemp;
225     mmapobj;
226     mq_close;
227     mq_getattr;
228     mq_notify;
229     mq_open;
230     mq_receive;
231     mq_reltimedreceive_np;
232     mq_reltimedsend_np;
233     mq_send;
234     mq_setattr;
235     mq_timedreceive;
236     mq_timedsend;
237     mq_unlink;
238     nanosleep;
239     ntohl;
240     ntohll;
241     ntohs;
242     posix_fadvise;
243     posix_fallocate;
244     posix_madvise;
245     posix_memalign;
246     posix_spawn_file_actions_addclosefrom_np;
247     posix_spawnattr_getsigignore_np;
248     posix_spawnattr_setsigignore_np;
249     ppoll;
250     priv_basicset;
251     pthread_key_create_once_np;
252     pthread_mutexattr_getrobust;
253     pthread_mutexattr_setrobust;
254     pthread_mutex_consistent;
255     readlinkat;
256     sched_getparam;
257     sched_get_priority_max;

```

```

258 sched_get_priority_min;
259 sched_getscheduler;
260 sched_rr_get_interval;
261 sched_setparam;
262 sched_setscheduler;
263 sched_yield;
264 sem_close;
265 sem_destroy;
266 sem_getvalue;
267 sem_init;
268 sem_open;
269 sem_post;
270 sem_reltimedwait_np;
271 sem_timedwait;
272 sem_trywait;
273 sem_unlink;
274 sem_wait;
275 setattrat;
276 setprogname;
277 _sharefs;
278 shm_open;
279 shm_unlink;
280 sigqueue;
281 sigtimedwait;
282 sigwaitinfo;
283 smt_pause;
284 stpcpy;
285 stpncpy;
286 strcasecmp;
287 strchrnul;
288 strndup;
289 strlen;
290 strnstr;
291 strsep;
292 symlinkat;
293 thr_keycreate_once;
294 timer_create;
295 timer_delete;
296 timer_getoverrun;
297 timer_gettime;
298 timer_settime;
299 u8_strerror;
300 u8_validate;
301 uconv_u16tou32;
302 uconv_u16tou8;
303 uconv_u32tou16;
304 uconv_u32tou8;
305 uconv_u8tou16;
306 uconv_u8tou32;
307 vasprintf;
308 verr;
309 verrx;
310 vforkx;
311 vwarn;
312 vwarnx;
313 warn;
314 warnx;
315 wcpcpy;
316 wcpncpy;
317 wcsasecmp;
318 wcsdup;
319 wcsncasecmp;
320 wcsnlen;

322 $if lf64
323 aio_cancel64;

```

```

324 aio_error64;
325 aio_fsync64;
326 aio_read64;
327 aioread64;
328 aio_return64;
329 aio_suspend64;
330 aio_waitn64;
331 aio_write64;
332 aiowrite64;
333 lio_listio64;
334 mkstemp64;
335 posix_fadvise64;
336 posix_fallocate64;
337 $endif
338 } SUNW_1.22.6;

340 SYMBOL_VERSION SUNW_1.22.6 { # s10u9 - SunOS 5.10 (Solaris 10) patch addition
341     protected:
342         futimens;
343         utimensat;
344 } SUNW_1.22.5;

346 SYMBOL_VERSION SUNW_1.22.5 { # s10u8 - SunOS 5.10 (Solaris 10) patch addition
347     protected:
348         getpagesizes2;
349 } SUNW_1.22.4;

351 SYMBOL_VERSION SUNW_1.22.4 { # s10u7 - SunOS 5.10 (Solaris 10) patch addition
352     protected:
353         SUNW_1.22.4;
354 } SUNW_1.22.3;

356 SYMBOL_VERSION SUNW_1.22.3 { # SunOS 5.10 (Solaris 10) patch additions
357     protected:
358         mutex_consistent;
359         u8_textprep_str;
360         uucopy;
361         uucopystr;
362 } SUNW_1.22.2;

364 SYMBOL_VERSION SUNW_1.22.2 { # SunOS 5.10 (Solaris 10) patch additions
365     protected:
366         is_system_labeled;
367         ucred_getlabel;
368         _ucred_getlabel;
369 } SUNW_1.22.1;

371 SYMBOL_VERSION SUNW_1.22.1 { # SunOS 5.10 (Solaris 10) patch additions
372     protected:
373         atomic_add_8;
374         atomic_add_8_nv;
375         atomic_add_char { FLAGS = NODYNSORT };
376         atomic_add_char_nv { FLAGS = NODYNSORT };
377         atomic_add_int { FLAGS = NODYNSORT };
378         atomic_add_int_nv { FLAGS = NODYNSORT };
379         atomic_add_ptr { FLAGS = NODYNSORT };
380         atomic_add_ptr_nv { FLAGS = NODYNSORT };
381         atomic_add_short { FLAGS = NODYNSORT };
382         atomic_add_short_nv { FLAGS = NODYNSORT };
383         atomic_and_16;
384         atomic_and_16_nv;
385         atomic_and_32_nv;
386         atomic_and_64;
387         atomic_and_64_nv;
388         atomic_and_8;
389         atomic_and_8_nv;

```



```

390 atomic_and_uchar { FLAGS = NODYNSORT };
391 atomic_and_uchar_nv { FLAGS = NODYNSORT };
392 atomic_and_uint_nv { FLAGS = NODYNSORT };
393 atomic_and_ulong { FLAGS = NODYNSORT };
394 atomic_and_ulong_nv { FLAGS = NODYNSORT };
395 atomic_and_ushort { FLAGS = NODYNSORT };
396 atomic_and_ushort_nv { FLAGS = NODYNSORT };
397 atomic_cas_16;
398 atomic_cas_32;
399 atomic_cas_64;
400 atomic_cas_8;
401 atomic_cas_ptr { FLAGS = NODYNSORT };
402 atomic_cas_uchar { FLAGS = NODYNSORT };
403 atomic_cas_uint { FLAGS = NODYNSORT };
404 atomic_cas_ulong { FLAGS = NODYNSORT };
405 atomic_cas_ushort { FLAGS = NODYNSORT };
406 atomic_clear_long_excl { FLAGS = NODYNSORT };
407 atomic_dec_16;
408 atomic_dec_16_nv;
409 atomic_dec_32;
410 atomic_dec_32_nv;
411 atomic_dec_64;
412 atomic_dec_64_nv;
413 atomic_dec_8;
414 atomic_dec_8_nv;
415 atomic_dec_uchar { FLAGS = NODYNSORT };
416 atomic_dec_uchar_nv { FLAGS = NODYNSORT };
417 atomic_dec_uint { FLAGS = NODYNSORT };
418 atomic_dec_uint_nv { FLAGS = NODYNSORT };
419 atomic_dec_ulong { FLAGS = NODYNSORT };
420 atomic_dec_ulong_nv { FLAGS = NODYNSORT };
421 atomic_dec_ushort { FLAGS = NODYNSORT };
422 atomic_dec_ushort_nv { FLAGS = NODYNSORT };
423 atomic_inc_16;
424 atomic_inc_16_nv;
425 atomic_inc_32;
426 atomic_inc_32_nv;
427 atomic_inc_64;
428 atomic_inc_64_nv;
429 atomic_inc_8;
430 atomic_inc_8_nv;
431 atomic_inc_uchar { FLAGS = NODYNSORT };
432 atomic_inc_uchar_nv { FLAGS = NODYNSORT };
433 atomic_inc_uint { FLAGS = NODYNSORT };
434 atomic_inc_uint_nv { FLAGS = NODYNSORT };
435 atomic_inc_ulong { FLAGS = NODYNSORT };
436 atomic_inc_ulong_nv { FLAGS = NODYNSORT };
437 atomic_inc_ushort { FLAGS = NODYNSORT };
438 atomic_inc_ushort_nv { FLAGS = NODYNSORT };
439 atomic_or_16;
440 atomic_or_16_nv;
441 atomic_or_32_nv;
442 atomic_or_64;
443 atomic_or_64_nv;
444 atomic_or_8;
445 atomic_or_8_nv;
446 atomic_or_uchar { FLAGS = NODYNSORT };
447 atomic_or_uchar_nv { FLAGS = NODYNSORT };
448 atomic_or_uint_nv { FLAGS = NODYNSORT };
449 atomic_or_ulong { FLAGS = NODYNSORT };
450 atomic_or_ulong_nv { FLAGS = NODYNSORT };
451 atomic_or_ushort { FLAGS = NODYNSORT };
452 atomic_or_ushort_nv { FLAGS = NODYNSORT };
453 atomic_set_long_excl { FLAGS = NODYNSORT };
454 atomic_swap_16;
455 atomic_swap_32;

```

```

456 atomic_swap_64;
457 atomic_swap_8;
458 atomic_swap_ptr { FLAGS = NODYNSORT };
459 atomic_swap_uchar { FLAGS = NODYNSORT };
460 atomic_swap_uint { FLAGS = NODYNSORT };
461 atomic_swap_ulong { FLAGS = NODYNSORT };
462 atomic_swap_ushort { FLAGS = NODYNSORT };
463 membar_consumer;
464 membar_enter;
465 membar_exit;
466 membar_producer;

468 $if _ELF32
469 enable_extended_FILE_stdio;
470 $endif

472 $if i386
473 # Note: atomic_[and,dec,inc,or]_64_nv are also defined above. Here,
474 # we add the NODYNSORT attribute to them. On this platform, they are
475 # aliases for the non-_nv versions. If that is changed, these lines
476 # should be removed.
477 atomic_and_64_nv { FLAGS = NODYNSORT };
478 atomic_dec_64_nv { FLAGS = NODYNSORT };
479 atomic_inc_64_nv { FLAGS = NODYNSORT };
480 atomic_or_64_nv { FLAGS = NODYNSORT };
481 $endif
482 $if _sparc
483 # Note: atomic_OP_WIDTH_nv symbols are also defined above. Here,
484 # we add the NODYNSORT attribute to them. On this platform, they are
485 # aliases for the non-_nv versions. If that is changed, these lines
486 # should be removed.
487 atomic_add_8_nv { FLAGS = NODYNSORT };
488 atomic_and_8_nv { FLAGS = NODYNSORT };
489 atomic_and_16_nv { FLAGS = NODYNSORT };
490 atomic_and_32_nv { FLAGS = NODYNSORT };
491 atomic_and_64_nv { FLAGS = NODYNSORT };
492 atomic_dec_8_nv { FLAGS = NODYNSORT };
493 atomic_dec_16_nv { FLAGS = NODYNSORT };
494 atomic_dec_32_nv { FLAGS = NODYNSORT };
495 atomic_dec_64_nv { FLAGS = NODYNSORT };
496 atomic_inc_8_nv { FLAGS = NODYNSORT };
497 atomic_inc_16_nv { FLAGS = NODYNSORT };
498 atomic_inc_32_nv { FLAGS = NODYNSORT };
499 atomic_inc_64_nv { FLAGS = NODYNSORT };
500 atomic_or_8_nv { FLAGS = NODYNSORT };
501 atomic_or_16_nv { FLAGS = NODYNSORT };
502 atomic_or_32_nv { FLAGS = NODYNSORT };
503 atomic_or_64_nv { FLAGS = NODYNSORT };
504 $endif
505 } SUNW_1.22;

507 SYMBOL_VERSION SUNW_1.22 { # SunOS 5.10 (Solaris 10)
508 global:
509 $if _ELF32
510 dladdr { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
511 dladdr1 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
512 dlclose { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
513 dlclose { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
514 dlerror { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
515 dlopen { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
516 dlmalloc { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
517 dlopen { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
518 dlclose { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
519 $elif sparcv9
520 dladdr { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
521 dladdr1 { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };

```

```

522     dlclose      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
523     dlclose      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
524     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
525     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
526     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
527     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
528     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
529     $elif amd64
530     dladdr        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
531     dladdr1       { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
532     dlamd64getunwind { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
533     dlclose      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
534     dlclose      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
535     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
536     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
537     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
538     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
539     dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
540     $else
541     $error unknown platform
542     $endif

544     protected:
545     alphasort;
546     _alphasort;
547     atomic_add_16;
548     atomic_add_16_nv;
549     atomic_add_32;
550     atomic_add_32_nv;
551     atomic_add_64;
552     atomic_add_64_nv;
553     atomic_add_long      { FLAGS = NODYNSORT };
554     atomic_add_long_nv   { FLAGS = NODYNSORT };
555     atomic_and_32;
556     atomic_and_uint      { FLAGS = NODYNSORT };
557     atomic_or_32;
558     atomic_or_uint       { FLAGS = NODYNSORT };
559     _Exit;
560     getisax;
561     _getisax;
562     getopt_clip;
563     _getopt_clip;
564     getopt_long;
565     _getopt_long;
566     getopt_long_only;
567     _getopt_long_only;
568     getpeercred;
569     _getpeercred;
570     getpflags;
571     _getpflags;
572     getppriv;
573     _getppriv;
574     getprivimplinfo;
575     _getprivimplinfo;
576     getzoneid;
577     getzoneidbyname;
578     getzonenamebyid;
579     imaxabs;
580     imaxdiv;
581     isblank;
582     iswblank;
583     port_alert;
584     port_associate;
585     port_create;
586     port_dissociate;
587     port_get;

```

```

588     port_getn;
589     port_send;
590     port_sendn;
591     posix_openpt;
592     posix_spawn;
593     posix_spawnattr_destroy;
594     posix_spawnattr_getflags;
595     posix_spawnattr_getpgroup;
596     posix_spawnattr_getschedparam;
597     posix_spawnattr_getschedpolicy;
598     posix_spawnattr_getsigdefault;
599     posix_spawnattr_getsigmask;
600     posix_spawnattr_init;
601     posix_spawnattr_setflags;
602     posix_spawnattr_setpgroup;
603     posix_spawnattr_setschedparam;
604     posix_spawnattr_setschedpolicy;
605     posix_spawnattr_setsigdefault;
606     posix_spawnattr_setsigmask;
607     posix_spawn_file_actions_addclose;
608     posix_spawn_file_actions_adddup2;
609     posix_spawn_file_actions_addopen;
610     posix_spawn_file_actions_destroy;
611     posix_spawn_file_actions_init;
612     posix_spawnnp;
613     priv_addset;
614     _priv_addset;
615     priv_allocset;
616     _priv_allocset;
617     priv_copyset;
618     _priv_copyset;
619     priv_delset;
620     _priv_delset;
621     priv_emptyset;
622     _priv_emptyset;
623     priv_fillset;
624     _priv_fillset;
625     _priv_free_info;
626     priv_freeset;
627     _priv_freeset;
628     priv_getbyname;
629     _priv_getbyname;
630     __priv_getbyname;
631     priv_getbynum;
632     _priv_getbynum;
633     __priv_getbynum;
634     _priv_getdata;
635     priv_getsetbyname;
636     _priv_getsetbyname;
637     __priv_getsetbyname;
638     priv_getsetbynum;
639     _priv_getsetbynum;
640     __priv_getsetbynum;
641     priv_gettext;
642     _priv_gettext;
643     priv_ineffect;
644     _priv_ineffect;
645     priv_intersect;
646     _priv_intersect;
647     priv_inverse;
648     _priv_inverse;
649     priv_isset;
650     _priv_isset;
651     priv_isequalset;
652     _priv_isequalset;
653     priv_isfullset;

```

```

654  __priv_isfullset;
655  __priv_ismember;
656  __priv_ismember;
657  __priv_issubset;
658  __priv_issubset;
659  __priv_parse_info;
660  __priv_set;
661  __priv_set;
662  __priv_set_to_str;
663  __priv_set_to_str;
664  __priv_set_to_str;
665  __priv_str_to_set;
666  __priv_str_to_set;
667  __priv_union;
668  __priv_union;
669  pselect;
670  pthread_attr_getstack;
671  pthread_attr_setstack;
672  pthread_barrierattr_destroy;
673  pthread_barrierattr_getpshared;
674  pthread_barrierattr_init;
675  pthread_barrierattr_setpshared;
676  pthread_barrier_destroy;
677  pthread_barrier_init;
678  pthread_barrier_wait;
679  pthread_condattr_getclock;
680  pthread_condattr_setclock;
681  pthread_mutexattr_getrobust_np { FLAGS = NODYNSORT };
682  pthread_mutexattr_setrobust_np { FLAGS = NODYNSORT };
683  pthread_mutex_consistent_np { FLAGS = NODYNSORT };
684  pthread_mutex_reltimedlock_np;
685  pthread_mutex_timedlock;
686  pthread_rwlock_reltimedrdlock_np;
687  pthread_rwlock_reltimedwrlock_np;
688  pthread_rwlock_timedrdlock;
689  pthread_rwlock_timedwrlock;
690  pthread_setschedprio;
691  pthread_spin_destroy;
692  pthread_spin_init;
693  pthread_spin_lock;
694  pthread_spin_trylock;
695  pthread_spin_unlock;
696  rctlblk_set_recipient_pid;
697  scandir;
698  __scandir;
699  schedctl_exit;
700  schedctl_init;
701  schedctl_lookup;
702  sema_reltimedwait;
703  sema_timedwait;
704  setenv;
705  setpflags;
706  __setpflags;
707  setppriv;
708  __setppriv;
709  strerror_r;
710  strtouf;
711  strtoumax;
712  strtold;
713  strtoumax;
714  ucred_free;
715  __ucred_free;
716  ucred_get;
717  __ucred_get;
718  ucred_getegid;
719  __ucred_getegid;

```

```

720  ucred_geteuid;
721  __ucred_geteuid;
722  ucred_getgroups;
723  __ucred_getgroups;
724  ucred_getpflags;
725  __ucred_getpflags;
726  ucred_getpid;
727  __ucred_getpid;
728  ucred_getprivset;
729  __ucred_getprivset;
730  ucred_getprojid;
731  __ucred_getprojid;
732  ucred_getrgid;
733  __ucred_getrgid;
734  ucred_getruid;
735  __ucred_getruid;
736  ucred_getsgid;
737  __ucred_getsgid;
738  ucred_getsuid;
739  __ucred_getsuid;
740  ucred_getzoneid;
741  __ucred_getzoneid;
742  ucred_size;
743  __ucred_size;
744  unsetenv;
745  wcstof;
746  wcstoumax;
747  wcstold;
748  wcstoll;
749  wcstoull;
750  wcstoumax;

752 $if lf64
753     alphasort64;
754     __alphasort64;
755     pselect_large_fdset;
756     scandir64;
757     __scandir64;
758 $endif

760 $if _ELF64
761     walkcontext;
762 $endif

764 $if _sparc
765     # Note: atomic_add_[16,32,64]_nv are also defined above. Here, we add
766     # the NODYNSORT attribute to them. On this platform, they are aliases
767     # for the non-_nv versions. If that is changed, these lines should be
768     # removed.
769     atomic_add_16_nv { FLAGS = NODYNSORT };
770     atomic_add_32_nv { FLAGS = NODYNSORT };
771     atomic_add_64_nv { FLAGS = NODYNSORT };
772 $endif

774 $if i386
775     # Note: atomic_add_64_nv is also defined above. Here, we add the
776     # NODYNSORT attribute to it. On this platform, it is an alias for
777     # atomic_add_64. If that is changed, this line should be removed.
778     atomic_add_64_nv { FLAGS = NODYNSORT };
779 $endif

781 $if amd64
782     # Exception unwind APIs required by the amd64 ABI
783     __SUNW_Unwind_DeleteException;
784     __SUNW_Unwind_ForcedUnwind;
785     __SUNW_Unwind_GetCFA;

```

```

786     _SUNW_Unwind_GetGR;
787     _SUNW_Unwind_GetIP;
788     _SUNW_Unwind_GetLanguageSpecificData;
789     _SUNW_Unwind_GetRegionStart;
790     _SUNW_Unwind_RaiseException;
791     _SUNW_Unwind_Resume;
792     _SUNW_Unwind_SetGR;
793     _SUNW_Unwind_SetIP;
794     _UA_CLEANUP_PHASE;
795     _UA_FORCE_UNWIND;
796     _UA_HANDLER_FRAME;
797     _UA_SEARCH_PHASE;
798     _Unwind_DeleteException;
799     _Unwind_ForcedUnwind;
800     _Unwind_GetCFA;
801     _Unwind_GetGR;
802     _Unwind_GetIP;
803     _Unwind_GetLanguageSpecificData;
804     _Unwind_GetRegionStart;
805     _Unwind_RaiseException;
806     _Unwind_Resume;
807     _Unwind_SetGR;
808     _Unwind_SetIP;
809 $endif
810 } SUNW_1.21.3;

812 SYMBOL_VERSION SUNW_1.21.3 { # SunOS 5.9 (Solaris 9) patch additions
813     protected:
814     forkall;
815 } SUNW_1.21.2;

817 SYMBOL_VERSION SUNW_1.21.2 { # SunOS 5.9 (Solaris 9) patch additions
818     protected:
819     getustack;
820     _getustack;
821     setustack;
822     _setustack;
823     stack_getbounds;
824     _stack_getbounds;
825     _stack_grow;
826     stack_inbounds;
827     _stack_inbounds;
828     stack_setbounds;
829     _stack_setbounds;
830     stack_violation;
831     _stack_violation;

833 $if _sparc
834     __makecontext_v2;
835     __makecontext_v2;
836 $endif
837 } SUNW_1.21.1;

839 SYMBOL_VERSION SUNW_1.21.1 { # SunOS 5.9 (Solaris 9) patch additions
840     protected:
841     crypt_gensalt;
842 } SUNW_1.21;

844 SYMBOL_VERSION SUNW_1.21 { # SunOS 5.9 (Solaris 9)
845     protected:
846     attropen;
847     _attropen;
848     bind_textdomain_codeset;
849     closefrom;
850     _closefrom;
851     cond_reltimedwait;

```

```

852     dcgettext;
853     dngettext;
854     fchownat;
855     _fchownat;
856     fdopendir;
857     _fdopendir;
858     fdwalk;
859     _fdwalk;
860     fstatat;
861     _fstatat;
862     futimesat;
863     _futimesat;
864     getcpuid;
865     _getcpuid;
866     gethomegroup;
867     _gethomegroup { FLAGS = NODYSORT };
868     getpagesizes;
869     getrctl;
870     _getrctl;
871     issetugid;
872     _issetugid;
873     _lwp_cond_reltimedwait;
874     meminfo;
875     _meminfo;
876     ngettext;
877     openat;
878     _openat;
879     printstack;
880     priocntl;
881     priocntlset;
882     pset_getattr;
883     pset_getloadavg;
884     pset_list;
885     pset_setattr;
886     pthread_cond_reltimedwait_np;
887     rctlblk_get_enforced_value;
888     rctlblk_get_firing_time;
889     rctlblk_get_global_action;
890     rctlblk_get_global_flags;
891     rctlblk_get_local_action;
892     rctlblk_get_local_flags;
893     rctlblk_get_privilege;
894     rctlblk_get_recipient_pid;
895     rctlblk_get_value;
896     rctlblk_set_local_action;
897     rctlblk_set_local_flags;
898     rctlblk_set_privilege;
899     rctlblk_set_value;
900     rctlblk_size;
901     rctl_walk;
902     renameat;
903     setrctl;
904     _setrctl;
905     unlinkat;
906     _unlinkat;
907     vfscanf;
908     _vfscanf;
909     vfwscanf;
910     vscanf;
911     _vscanf;
912     vsscanf;
913     _vsscanf;
914     vswscanf;
915     vwscanf;

917 $if _ELF32

```

```

918     walkcontext;
919 $endif

921 $if lf64
922     attropen64;
923     _attropen64;
924     fstatat64;
925     _fstatat64;
926     openat64;
927     _openat64;
928 $endif
929 } SUNW_1.20.4;

931 SYMBOL_VERSION SUNW_1.20.4 { # SunOS 5.8 (Solaris 8) patch additions
932     protected:
933     semtimedop;
934     _semtimedop;
935 } SUNW_1.20.1;

937 SYMBOL_VERSION SUNW_1.20.1 { # SunOS 5.8 (Solaris 8) patch additions
938     protected:
939     getacct;
940     _getacct;
941     getprojid;
942     _getprojid;
943     gettaskid;
944     _gettaskid;
945     msgids;
946     _msgids;
947     msgsnap;
948     _msgsnap;
949     putacct;
950     _putacct;
951     semids;
952     _semids;
953     settaskid;
954     _settaskid;
955     shmids;
956     _shmids;
957     wracct;
958     _wracct;
959 } SUNW_1.20;

961 SYMBOL_VERSION SUNW_1.20 { # SunOS 5.8 (Solaris 8)
962     protected:
963     getextmntent;
964     resetmnttab;
965 } SUNW_1.19;

967 SYMBOL_VERSION SUNW_1.19 {
968     protected:
969     strlcat;
970     strlcpy;
971     umount2;
972     _umount2;
973 } SUNW_1.18.1;

975 SYMBOL_VERSION SUNW_1.18.1 {
976     protected:
977     __fsetlocking;
978 } SUNW_1.18;

980 SYMBOL_VERSION SUNW_1.18 { # SunOS 5.7 (Solaris 7)
981     protected:
982     btowc;
983     __fbufsize;

```

```

984     __flbf;
985     __flushlbf;
986     __fpending;
987     __fpurge;
988     __freadable;
989     __freading;
990     fwide;
991     fwprintf;
992     __fwritable;
993     __fwriting;
994     fwscanf;
995     getloadavg;
996     isaexec;
997     mbrlen;
998     mbrtowc;
999     mbsinit;
1000    mbsrtowcs;
1001    pcsample;
1002    pthread_attr_getguardsize;
1003    pthread_attr_setguardsize;
1004    pthread_getconcurrency;
1005    pthread_mutexattr_gettype;
1006    pthread_mutexattr_settype;
1007    pthread_rwlockattr_destroy;
1008    pthread_rwlockattr_getpshared;
1009    pthread_rwlockattr_init;
1010    pthread_rwlockattr_setpshared;
1011    pthread_rwlock_destroy;
1012    pthread_rwlock_init;
1013    pthread_rwlock_rdlock;
1014    pthread_rwlock_tryrdlock;
1015    pthread_rwlock_trywrlock;
1016    pthread_rwlock_unlock;
1017    pthread_rwlock_wrlock;
1018    pthread_setconcurrency;
1019    swprintf;
1020    wscanf;
1021    __sysconf_xpg5;
1022    vfwprintf;
1023    vswprintf;
1024    vwprintf;
1025    wcrctomb;
1026    wcsrtombs;
1027    wcsstr;
1028    wctob;
1029    wmemchr;
1030    wmemcmp;
1031    wmemcpy;
1032    wmemmove;
1033    wmemset;
1034    wprintf;
1035    wscanf;

1037 $if _ELF32
1038     select_large_fdset;
1039 $endif
1040 } SUNW_1.17;

1042 # The empty versions SUNW_1.2 through SUNW_1.17 must be preserved because
1043 # applications built on Solaris 2.6 Beta (when they did contain symbols)
1044 # may depend on them. All symbol content for SunOS 5.6 is now in SUNW_1.1

1046 SYMBOL_VERSION SUNW_1.17 {
1047     protected:
1048     SUNW_1.17;
1049 } SUNW_1.16;

```

```

1051 SYMBOL_VERSION SUNW_1.16 {
1052     protected:
1053     SUNW_1.16;
1054 } SUNW_1.15;

1056 SYMBOL_VERSION SUNW_1.15 {
1057     protected:
1058     SUNW_1.15;
1059 } SUNW_1.14;

1061 SYMBOL_VERSION SUNW_1.14 {
1062     protected:
1063     SUNW_1.14;
1064 } SUNW_1.13;

1066 SYMBOL_VERSION SUNW_1.13 {
1067     protected:
1068     SUNW_1.13;
1069 } SUNW_1.12;

1071 SYMBOL_VERSION SUNW_1.12 {
1072     protected:
1073     SUNW_1.12;
1074 } SUNW_1.11;

1076 SYMBOL_VERSION SUNW_1.11 {
1077     protected:
1078     SUNW_1.11;
1079 } SUNW_1.10;

1081 SYMBOL_VERSION SUNW_1.10 {
1082     protected:
1083     SUNW_1.10;
1084 } SUNW_1.9;

1086 SYMBOL_VERSION SUNW_1.9 {
1087     protected:
1088     SUNW_1.9;
1089 } SUNW_1.8;

1091 SYMBOL_VERSION SUNW_1.8 {
1092     protected:
1093     SUNW_1.8;
1094 } SUNW_1.7;

1096 SYMBOL_VERSION SUNW_1.7 {
1097     protected:
1098     SUNW_1.7;
1099 } SUNW_1.6;

1101 SYMBOL_VERSION SUNW_1.6 {
1102     protected:
1103     SUNW_1.6;
1104 } SUNW_1.5;

1106 SYMBOL_VERSION SUNW_1.5 {
1107     protected:
1108     SUNW_1.5;
1109 } SUNW_1.4;

1111 SYMBOL_VERSION SUNW_1.4 {
1112     protected:
1113     SUNW_1.4;
1114 } SUNW_1.3;

```

```

1116 SYMBOL_VERSION SUNW_1.3 {
1117     protected:
1118     SUNW_1.3;
1119 } SUNW_1.2;

1121 SYMBOL_VERSION SUNW_1.2 {
1122     protected:
1123     SUNW_1.2;
1124 } SUNW_1.1;

1126 SYMBOL_VERSION SUNW_1.1 {           # SunOS 5.6 (Solaris 2.6)
1127     global:
1128     __locl;
1129     protected:
1130     basename;
1131     bindtextdomain;
1132     bsd_signal;
1133     dbm_clearerr;
1134     dbm_error;
1135     dcgettext;
1136     dgettext;
1137     directio;
1138     dirname;
1139     endusershell;
1140     _exithandle;
1141     fgetwc;
1142     fgetws;
1143     fpgetround;
1144     fpsetround;
1145     fputwc;
1146     fputws;
1147     fseeko;
1148     ftello;
1149     ftrylockfile;
1150     getexecname;
1151     _getexecname;
1152     getpassphrase;
1153     gettext;
1154     getusershell;
1155     getwc;
1156     getwchar;
1157     getws;
1158     isenglish;
1159     isideogram;
1160     isnumber;
1161     isphonogram;
1162     isspecial;
1163     iswalnum;
1164     iswalph;
1165     iswcntrl;
1166     iswctype;
1167     iswdigit;
1168     iswgraph;
1169     iswlower;
1170     iswprint;
1171     iswpunct;
1172     iswspace;
1173     iswupper;
1174     iswxdigit;
1175     __locl;
1176     __longjmp;
1177     _lwp_sema_trywait;
1178     ntp_adjtime;
1179     _ntp_adjtime;
1180     ntp_gettime;
1181     _ntp_gettime;

```

```

1182  __posix_asctime_r;
1183  __posix_ctime_r;
1184  __posix_getgrgid_r;
1185  __posix_getgrnam_r;
1186  __posix_getlogin_r;
1187  __posix_getpwnam_r;
1188  __posix_getpwuid_r;
1189  __posix_sigwait;
1190  __posix_ttyname_r;
1191  pset_assign;
1192  pset_bind;
1193  pset_create;
1194  pset_destroy;
1195  pset_info;
1196  pthread_atfork;
1197  pthread_attr_destroy;
1198  pthread_attr_getdetachstate;
1199  pthread_attr_getinheritsched;
1200  pthread_attr_getschedparam;
1201  pthread_attr_getschedpolicy;
1202  pthread_attr_getscope;
1203  pthread_attr_getstackaddr;
1204  pthread_attr_getstacksize;
1205  pthread_attr_init;
1206  pthread_attr_setdetachstate;
1207  pthread_attr_setinheritsched;
1208  pthread_attr_setschedparam;
1209  pthread_attr_setschedpolicy;
1210  pthread_attr_setscope;
1211  pthread_attr_setstackaddr;
1212  pthread_attr_setstacksize;
1213  pthread_cancel;
1214  __pthread_cleanup_pop;
1215  __pthread_cleanup_push;
1216  pthread_create;
1217  pthread_detach;
1218  pthread_equal;
1219  pthread_exit;
1220  pthread_getschedparam;
1221  pthread_getspecific;
1222  pthread_join;
1223  pthread_key_create;
1224  pthread_key_delete;
1225  pthread_kill;
1226  pthread_once;
1227  pthread_self;
1228  pthread_setcancelstate;
1229  pthread_setcanceltype;
1230  pthread_setschedparam;
1231  pthread_setspecific;
1232  pthread_sigmask;
1233  pthread_testcancel;
1234  putwc;
1235  putwchar;
1236  puts;
1237  regcmp;
1238  regex;
1239  resolvepath;
1240  _resolvepath;
1241  rwlock_destroy      { FLAGS = NODYNSORT };
1242  _rwlock_destroy    { FLAGS = NODYNSORT };
1243  sema_destroy;
1244  _sema_destroy;
1245  _setjmp;
1246  setusershell;
1247  siginterrupt;

```

```

1248  sigstack;
1249  snprintf;
1250  strtows;
1251  sync_instruction_memory;
1252  textdomain;
1253  thr_main;
1254  towctrans;
1255  tolower;
1256  toupper;
1257  ungetwc;
1258  vsnprintf;
1259  watoll;
1260  wscat;
1261  wcschr;
1262  wcscmp;
1263  wscoll;
1264  wcscpy;
1265  wcsncpy;
1266  wcsftime;
1267  wcslen;
1268  wcsncat;
1269  wcsncmp;
1270  wcsncpy;
1271  wcsrchr;
1272  wcsrchr;
1273  wcssp;
1274  wcstod;
1275  wctok;
1276  wcstol;
1277  wcstoul;
1278  wcs wcs;
1279  wcs width;
1280  wcsxfrm;
1281  wctrans;
1282  wctype;
1283  wcwidth;
1284  wscasecmp;
1285  wscat;
1286  wcschr;
1287  wscmp;
1288  wscol;
1289  wscoll;
1290  wscpy;
1291  wscspn;
1292  wsdup;
1293  wslen;
1294  wsncasecmp;
1295  wsnat;
1296  wsncmp;
1297  wsncpy;
1298  wspbrk;
1299  wsprintf;
1300  wsrchr;
1301  wsscanf;
1302  wssp;
1303  wstod;
1304  wstok;
1305  wstol;
1306  wstoll;
1307  wstostr;
1308  wsxfrm;
1309  __xpg4_putmsg;
1310  __xpg4_putpmsg;

1312 $if lf64
1313  creat64;

```

```

1314     _creat64;
1315     fgetpos64;
1316     fopen64;
1317     freopen64;
1318     fseeko64;
1319     fsetpos64;
1320     fstat64;
1321     _fstat64;
1322     fstatvfs64;
1323     _fstatvfs64;
1324     ftello64;
1325     ftruncate64;
1326     _ftruncate64;
1327     ftw64;
1328     _ftw64;
1329     getdents64;
1330     _getdents64;
1331     getrlimit64;
1332     _getrlimit64;
1333     lockf64;
1334     _lockf64;
1335     lseek64;
1336     _lseek64;
1337     lstat64;
1338     _lstat64;
1339     mkstemp64;
1340     _mkstemp64;
1341     mmap64;
1342     _mmap64;
1343     nftw64;
1344     _nftw64;
1345     open64;
1346     _open64;
1347     __posix_readdir_r;
1348     pread64;
1349     _pread64;
1350     pwrite64;
1351     _pwrite64;
1352     readdir64;
1353     _readdir64;
1354     readdir64_r;
1355     _readdir64_r;
1356     setrlimit64;
1357     _setrlimit64;
1358     s_fcntl;
1359     _s_fcntl           { FLAGS = NODYNSORT };
1360     s_ioctl;
1361     stat64;
1362     _stat64;
1363     statvfs64;
1364     _statvfs64;
1365     tell64;
1366     _tell64;
1367     tmpfile64;
1368     truncate64;
1369     _truncate64;
1370     _xftw64;
1371 $endif

1373 $if _sparc
1374     __flt_rounds;
1375 $endif
1376 } SUNW_0.9;

1378 SYMBOL_VERSION SUNW_0.9 {           # SunOS 5.5 (Solaris 2.5)
1379     protected;
```

```

1380     acl;
1381     bcmp;
1382     bcopy;
1383     bzero;
1384     facl;
1385     ftime;
1386     getdtablesize;
1387     gethostid;
1388     gethostname;
1389     getpagesize;
1390     getpriority;
1391     getrusage;
1392     getwd;
1393     index;
1394     initstate;
1395     killpg;
1396     _nsc_trydoorcall;
1397     pthread_condattr_destroy;
1398     pthread_condattr_getpshared;
1399     pthread_condattr_init;
1400     pthread_condattr_setpshared;
1401     pthread_cond_broadcast;
1402     pthread_cond_destroy;
1403     pthread_cond_init;
1404     pthread_cond_signal;
1405     pthread_cond_timedwait;
1406     pthread_cond_wait;
1407     pthread_mutexattr_destroy;
1408     pthread_mutexattr_getprioceiling;
1409     pthread_mutexattr_getprotocol;
1410     pthread_mutexattr_getpshared;
1411     pthread_mutexattr_init;
1412     pthread_mutexattr_setprioceiling;
1413     pthread_mutexattr_setprotocol;
1414     pthread_mutexattr_setpshared;
1415     pthread_mutex_destroy;
1416     pthread_mutex_getprioceiling;
1417     pthread_mutex_init;
1418     pthread_mutex_lock;
1419     pthread_mutex_setprioceiling;
1420     pthread_mutex_trylock;
1421     pthread_mutex_unlock;
1422     random;
1423     reboot;
1424     re_comp;
1425     re_exec;
1426     rindex;
1427     setbuffer;
1428     sethostname;
1429     setlinebuf;
1430     setpriority;
1431     setregid;
1432     setreuid;
1433     setstate;
1434     srandom;
1435     thr_min_stack;
1436     thr_stksegment;
1437     ualarm;
1438     usleep;
1439     wait3;
1440     wait4;
1441 } SUNW_0.8;

1443 SYMBOL_VERSION SUNW_0.8 {           # SunOS 5.4 (Solaris 2.4)
1444     global:
1445     __xpg4           { FLAGS = NODIRECT };
```



```

1446     protected:
1447         addsev;
1448         cond_broadcast      { FLAGS = NODYNSORT };
1449         cond_destroy        { FLAGS = NODYNSORT };
1450         cond_init;
1451         cond_signal        { FLAGS = NODYNSORT };
1452         cond_timedwait;
1453         cond_wait;
1454         confstr;
1455         fnmatch;
1456         _getdate_err_addr;
1457         glob;
1458         globfree;
1459         iconv;
1460         iconv_close;
1461         iconv_open;
1462         lfmt;
1463         mutex_destroy      { FLAGS = NODYNSORT };
1464         mutex_init;
1465         mutex_lock         { FLAGS = NODYNSORT };
1466         mutex_trylock      { FLAGS = NODYNSORT };
1467         mutex_unlock       { FLAGS = NODYNSORT };
1468         pfmt;
1469         regcomp;
1470         regerror;
1471         regexec;
1472         regfree;
1473         rwlock_init;
1474         rw_rdlock          { FLAGS = NODYNSORT };
1475         rw_read_held;
1476         rw_tryrdlock       { FLAGS = NODYNSORT };
1477         rw_trywrlock       { FLAGS = NODYNSORT };
1478         rw_unlock          { FLAGS = NODYNSORT };
1479         rw_write_held;
1480         rw_wrlock          { FLAGS = NODYNSORT };
1481         sema_held;
1482         sema_init;
1483         sema_post;
1484         sema_trywait;
1485         sema_wait;
1486         setcat;
1487         sigfpe;
1488         strfmon;
1489         strptime;
1490         thr_continue;
1491         thr_create;
1492         thr_exit;
1493         thr_getconcurrency;
1494         thr_getprio;
1495         thr_getspecific;
1496         thr_join;
1497         thr_keycreate;
1498         thr_kill;
1499         thr_self           { FLAGS = NODYNSORT };
1500         thr_setconcurrency;
1501         thr_setprio;
1502         thr_setspecific;
1503         thr_sigsetmask;
1504         thr_suspend;
1505         thr_yield;
1506         vlfmt;
1507         vpfmt;
1508         wordexp;
1509         wordfree;
1510 } SUNW_0.7;

```

```

1512 SYMBOL_VERSION SUNW_0.7 {      # SunOS 5.3 (Solaris 2.3)
1513     global:
1514         altzone;
1515         _ctype;
1516         isnanf            { TYPE = FUNCTION; FILTER = libm.so.2 };
1517         lone;
1518         lten;
1519         lzero;
1520         memalign          { FLAGS = NODIRECT };
1521         modff             { TYPE = FUNCTION; FILTER = libm.so.2 };
1522         nss_default_finders;
1523         _sibuf;
1524         _sobuf;
1525         _sys_buslist;
1526         _sys_cldlist;
1527         _sys_fpelist;
1528         _sys_illlist;
1529         _sys_segvlst;
1530         _sys_siginfo;
1531         _sys_siglist;
1532         _sys_siglistn;
1533         _sys_siglistp;
1534         _sys_traplist;
1535         valloc            { FLAGS = NODIRECT };
1537 $if _ELF32
1538     _bufendtab;
1539     _lastbuf;
1540     sys_errlist;
1541     sys_nerr;
1542     _sys_nsig;
1543 $endif
1545     protected:
1546         a64l;
1547         adjtime;
1548         ascftime;
1549         _assert;
1550         atoll;
1551         brk;
1552         __builtin_alloca;
1553         cftime;
1554         closelog;
1555         csetcol;
1556         csetlen;
1557         ctermid_r;
1558         dbm_close;
1559         dbm_delete;
1560         dbm_fetch;
1561         dbm_firstkey;
1562         dbm_nextkey;
1563         dbm_open;
1564         dbm_store;
1565         decimal_to_double;
1566         decimal_to_extended;
1567         decimal_to_quadruple;
1568         decimal_to_single;
1569         double_to_decimal;
1570         drand48;
1571         econvert;
1572         ecvt;
1573         endnetgrent;
1574         endspent;
1575         endutent;
1576         endutxent;
1577         erand48;

```

```

1578     euccol;
1579     euclen;
1580     eucscol;
1581     extended_to_decimal;
1582     fchroot;
1583     fconvert;
1584     fcvt;
1585     ffs;
1586     fgetspent;
1587     fgetspent_r;
1588     _filbuf;
1589     file_to_decimal;
1590     finite;
1591     _flsbuf;
1592     fork1                { FLAGS = NODYNSORT };
1593     fpclass;
1594     fpgetmask;
1595     fpgetsticky;
1596     fpsetmask;
1597     fpsetsticky;
1598     fstatfs;
1599     ftruncate;
1600     ftw;
1601     func_to_decimal;
1602     gconvert;
1603     gcvt;
1604     getdents;
1605     gethrtime;
1606     gethrvtime;
1607     getmntany;
1608     getmntent;
1609     getnetgrent;
1610     getnetgrent_r;
1611     getpw;
1612     getspent;
1613     getspent_r;
1614     getspnam;
1615     getspnam_r;
1616     getutent;
1617     getutid;
1618     getutline;
1619     getutmp;
1620     getutmpx;
1621     getutxent;
1622     getutxid;
1623     getutxline;
1624     getvfsany;
1625     getvfsent;
1626     getvfsfile;
1627     getvfsspec;
1628     getwidth;
1629     gsignal;
1630     hasmntopt;
1631     inetgr;
1632     insque;
1633     _insque;
1634     jrand48;
1635     l64a;
1636     ladd;
1637     lckpwwdf;
1638     lcong48;
1639     ldivide;
1640     lexpl0;
1641     llabs;
1642     lldiv;
1643     llog10;

```

```

1644     llseek;
1645     lltostr;
1646     lmul;
1647     lrand48;
1648     lshift1;
1649     lsub;
1650     _lwp_cond_broadcast;
1651     _lwp_cond_signal;
1652     _lwp_cond_timedwait;
1653     _lwp_cond_wait;
1654     _lwp_continue;
1655     _lwp_info;
1656     _lwp_kill;
1657     _lwp_mutex_lock;
1658     _lwp_mutex_trylock;
1659     _lwp_mutex_unlock;
1660     _lwp_self;
1661     _lwp_sema_init;
1662     _lwp_sema_post;
1663     _lwp_sema_wait;
1664     _lwp_suspend;
1665     madvise;
1666     __major;
1667     __makedev;
1668     mincore;
1669     __minor;
1670     mkstemp;
1671     _mkstemp;
1672     mlockall;
1673     mrand48;
1674     munlockall;
1675     _mutex_held                { FLAGS = NODYNSORT };
1676     _mutex_lock                { FLAGS = NODYNSORT };
1677     nrand48;
1678     _nss_netdb_aliases;
1679     _nss_XbyY_buf_alloc;
1680     _nss_XbyY_buf_free;
1681     __nsw_extended_action;
1682     __nsw_freeconfig;
1683     __nsw_getconfig;
1684     openlog;
1685     plock;
1686     p_online;
1687     pread;
1688     __priocntl;
1689     __priocntlset;
1690     processor_bind;
1691     processor_info;
1692     psiginfo;
1693     psignal;
1694     putpwent;
1695     putspent;
1696     pututline;
1697     pututxline;
1698     pwrite;
1699     qeconvert;
1700     qecvt;
1701     qfconvert;
1702     qfcvt;
1703     qgconvert;
1704     qgcvt;
1705     quadruple_to_decimal;
1706     realpath;
1707     remque;
1708     _remque;
1709     _rw_read_held;

```

```

1710     _rw_write_held;
1711     seconvert;
1712     seed48;
1713     select;
1714     _sema_held;
1715     setegid;
1716     seteuid;
1717     setlogmask;
1718     setnetgrent;
1719     setspent;
1720     settimeofday;
1721     setutent;
1722     setutxent;
1723     sfconvert;
1724     sgconvert;
1725     sig2str;
1726     sigwait;
1727     single_to_decimal;
1728     srand48;
1729     ssignal;
1730     statfs;
1731     str2sig;
1732     strcasecmp;
1733     string_to_decimal;
1734     strncasecmp;
1735     strsignal;
1736     strtoll;
1737     strtoull;
1738     swapctl;
1739     _syscall;
1740     sysfs;
1741     syslog;
1742     _syslog;
1743     tmpnam_r;
1744     truncate;
1745     ttyslot;
1746     uadmin;
1747     ulckpwn;
1748     ulltostr;
1749     unordered;
1750     updwtmp;
1751     updwtmpx;
1752     ustat;
1753     utimes;
1754     utmpname;
1755     utmpxname;
1756     vfork;
1757     vhangup;
1758     vsyslog;
1759     yield;

1761 $if i386
1762     # Note: _syscall is also defined above. Here, we add the NODYNSORT
1763     # attribute to it. On this platform, it is an alias to syscall.
1764     # If that is changed, this lines should be removed.
1765     _syscall          { FLAGS = NODYNSORT };
1766 $endif

1768 # The 32-bit sparc ABI requires SISCD_2.3. On other platforms, those symbols
1769 # go directly into SUNW_0.7.
1770 $if sparc32
1771 } SISCD_2.3;

1773 SYMBOL_VERSION SISCD_2.3 {
1774 $endif

```

```

1776     global:
1777         errno          { FLAGS = NODIRECT };
1778         _iob;

1780     protected:
1781         addseverity;
1782         _addseverity;
1783         asctime_r;
1784         crypt;
1785         _crypt;
1786         ctime_r;
1787         encrypt;
1788         _encrypt;
1789         endgrent;
1790         endpwent;
1791         __errno;
1792         fgetgrent;
1793         fgetgrent_r;
1794         fgetpwent;
1795         fgetpwent_r;
1796         flockfile;
1797         funlockfile;
1798         getchar_unlocked;
1799         getc_unlocked;
1800         getgrent;
1801         getgrent_r;
1802         getgrgid_r;
1803         getgrnam_r;
1804         getitimer;
1805         _getitimer;
1806         getlogin_r;
1807         getpwent;
1808         getpwent_r;
1809         getpwnam_r;
1810         getpwuid_r;
1811         gettimeofday;
1812         _gettimeofday;
1813         gmtime_r;
1814         localtime_r;
1815         putchar_unlocked;
1816         putc_unlocked;
1817         rand_r;
1818         readdir_r;
1819         setgrent;
1820         setitimer;
1821         _setitimer;
1822         setkey;
1823         _setkey;
1824         setpwent;
1825         strtok_r;
1826         sysinfo;
1827         _sysinfo;
1828         ttyname_r;

1830 $if _ELF32
1831     __div64;
1832     __mul64;
1833     __rem64;
1834     __udiv64;
1835     __urem64;
1836 $endif

1838 $if sparc32
1839     __dtoll;
1840     __dtoull;
1841     __ftoll;

```

```

1842     __ftoll;
1843     __Q_lltoq;
1844     __Q_qtoll;
1845     __Q_qtoll;
1846     __Q_ulltoq;
1847     sbrk;
1848     _sbrk;
1849     __umul64                { FLAGS = NODYNSORT }; # Same address as __mul6
1850 $endif

1852 # On 32-bit platforms, the following symbols go into SYSVABI_1.3, but on
1853 # other platforms they go directly into the current version (which will be
1854 # either SUNW_0.7, or SISCD_2.3, depending on the similar issue described above.
1855 $if _ELF32
1856 } SYSVABI_1.3;

1858 SYMBOL_VERSION SYSVABI_1.3 {
1859 $endif

1861     global:
1862     __altzone;
1863     calloc                { FLAGS = NODIRECT };
1864     __ctype;
1865     daylight;
1866     __daylight;
1867     environ                { FLAGS = NODIRECT };
1868     __environ            { FLAGS = NODIRECT };
1869     free                  { FLAGS = NODIRECT };
1870     frexp                 { TYPE = FUNCTION; FILTER = libm.so.2 };
1871     getdate_err;
1872     __getdate_err;
1873     getenv;
1874     __huge_val;
1875     __iob;
1876     isnan                 { TYPE = FUNCTION; FILTER = libm.so.2 };
1877     __isnan              { TYPE = FUNCTION; FILTER = libm.so.2 };
1878     isnand                { TYPE = FUNCTION; FILTER = libm.so.2 };
1879     __isnand             { TYPE = FUNCTION; FILTER = libm.so.2 };
1880     ldexp                 { TYPE = FUNCTION; FILTER = libm.so.2 };
1881     logb                  { TYPE = FUNCTION; FILTER = libm.so.2 };
1882     malloc                { FLAGS = NODIRECT };
1883     memcmp;
1884     memcpy;
1885     memmove;
1886     memset;
1887     modf                  { TYPE = FUNCTION; FILTER = libm.so.2 };
1888     __modf               { TYPE = FUNCTION; FILTER = libm.so.2 };
1889     nextafter             { TYPE = FUNCTION; FILTER = libm.so.2 };
1890     __nextafter          { TYPE = FUNCTION; FILTER = libm.so.2 };
1891     __numeric;
1892     optarg;
1893     opterr;
1894     optind;
1895     optopt;
1896     realloc                { FLAGS = NODIRECT };
1897     scalb                 { TYPE = FUNCTION; FILTER = libm.so.2 };
1898     __scalb              { TYPE = FUNCTION; FILTER = libm.so.2 };
1899     timezone;
1900     __timezone;
1901     tzname;
1902     __tzname;
1903 $if i386
1904     __fp_hw;
1905 $endif

1907     protected:

```

```

1908     abort;
1909     abs;
1910     access;
1911     __access;
1912     acct;
1913     __acct;
1914     alarm;
1915     __alarm;
1916     asctime;
1917     __assert;
1918     atexit;
1919     atof;
1920     atoi;
1921     atol;
1922     bsearch;
1923     catclose;
1924     __catclose;
1925     catgets;
1926     __catgets;
1927     catopen;
1928     __catopen;
1929     cfgetispeed;
1930     __cfgetispeed;
1931     cfgetospeed;
1932     __cfgetospeed;
1933     cfsetispeed;
1934     __cfsetispeed;
1935     cfsetospeed;
1936     __cfsetospeed;
1937     chdir;
1938     __chdir;
1939     chmod;
1940     __chmod;
1941     chown;
1942     __chown;
1943     chroot;
1944     __chroot;
1945     cleanup;
1946     clearerr;
1947     clock;
1948     __close;
1949     close;
1950     closedir;
1951     __closedir;
1952     creat;
1953     __creat;
1954     ctermid;
1955     ctime;
1956     cuserid;
1957     __cuserid;
1958     difftime;
1959     div;
1960     dup;
1961     __dup;
1962     dup2;
1963     __dup2;
1964     execl;
1965     __execl;
1966     execlp;
1967     __execlp;
1968     execlp;
1969     __execlp;
1970     execv;
1971     __execv;
1972     execve;
1973     __execve;

```

```

1974     execvp;
1975     _execvp;
1976     exit;
1977     _exit;
1978     fattach;
1979     _fattach;
1980     fchdir;
1981     _fchdir;
1982     fchmod;
1983     _fchmod;
1984     fchown;
1985     _fchown;
1986     fclose;
1987     fcntl;
1988     _fcntl;
1989     fdetach;
1990     _fdetach;
1991     fdopen;
1992     _fdopen;
1993     feof;
1994     ferror;
1995     fflush;
1996     fgetc;
1997     fgetpos;
1998     fgets;
1999     __filbuf;
2000     fileno;
2001     _fileno;
2002     __flsbuf;
2003     fmtmsg;
2004     _fmtmsg;
2005     fopen;
2006     _fork;
2007     fork;
2008     fpathconf;
2009     _fpathconf;
2010     fprintf;
2011     fputc;
2012     fputs;
2013     fread;
2014     freopen;
2015     fscanf;
2016     fseek;
2017     fsetpos;
2018     fstat;
2019     _fstat;
2020     fstatvfs;
2021     _fstatvfs;
2022     fsync;
2023     _fsync;
2024     ftell;
2025     ftok;
2026     _ftok;
2027     fwrite;
2028     getc;
2029     getchar;
2030     getcontext;
2031     _getcontext;
2032     getcwd;
2033     _getcwd;
2034     getdate;
2035     _getdate;
2036     getegid;
2037     _getegid;
2038     geteuid;
2039     _geteuid;

```

```

2040     getgid;
2041     _getgid;
2042     getgrgid;
2043     getgrnam;
2044     getgroups;
2045     _getgroups;
2046     getlogin;
2047     getmsg;
2048     _getmsg;
2049     getopt;
2050     _getopt;
2051     getpass;
2052     _getpass;
2053     getpgid;
2054     _getpgid;
2055     getpgrp;
2056     _getpgrp;
2057     getpid;
2058     _getpid;
2059     getpmsg;
2060     _getpmsg;
2061     getppid;
2062     _getppid;
2063     getpwnam;
2064     getpwuid;
2065     getrlimit;
2066     _getrlimit;
2067     gets;
2068     getsid;
2069     _getsid;
2070     getsubopt;
2071     _getsubopt;
2072     gettxt;
2073     _gettext;
2074     getuid;
2075     _getuid;
2076     getw;
2077     _getw;
2078     gmtime;
2079     grantpt;
2080     _grantpt;
2081     hcreate;
2082     _hcreate;
2083     hdestroy;
2084     _hdestroy;
2085     hsearch;
2086     _hsearch;
2087     initgroups;
2088     _initgroups;
2089     ioctl;
2090     _ioctl;
2091     isalnum;
2092     isalpha;
2093     isascii;
2094     _isascii;
2095     isastream;
2096     _isastream;
2097     isatty;
2098     _isatty;
2099     iscntrl;
2100     isdigit;
2101     isgraph;
2102     islower;
2103     isprint;
2104     ispunct;
2105     isspace;

```

```

2106 isupper;
2107 isxdigit;
2108 kill;
2109 _kill;
2110 labs;
2111 lchown;
2112 _lchown;
2113 ldiv;
2114 lfind;
2115 _lfind;
2116 link;
2117 _link;
2118 localeconv;
2119 localtime;
2120 lockf;
2121 _lockf;
2122 longjmp;
2123 lsearch;
2124 _lsearch;
2125 lseek;
2126 _lseek;
2127 lstat;
2128 _lstat;
2129 makecontext;
2130 _makecontext;
2131 mblen;
2132 mbstowcs;
2133 mbtowc;
2134 memccpy;
2135 _memccpy;
2136 memchr;
2137 memcntl;
2138 _memcntl;
2139 mkdir;
2140 _mkdir;
2141 mkfifo;
2142 _mkfifo;
2143 mknod;
2144 _mknod;
2145 mktemp;
2146 _mktemp;
2147 mktime;
2148 mlock;
2149 _mlock;
2150 mmap;
2151 _mmap;
2152 monitor;
2153 _monitor;
2154 mount;
2155 _mount;
2156 mprotect;
2157 _mprotect;
2158 msgctl;
2159 _msgctl;
2160 msgget;
2161 _msgget;
2162 msgrcv;
2163 _msgrcv;
2164 msgsnd;
2165 _msgsnd;
2166 msync;
2167 _msync;
2168 munlock;
2169 _munlock;
2170 munmap;
2171 _munmap;

```

```

2172 nftw;
2173 _nftw;
2174 nice;
2175 _nice;
2176 nl_langinfo;
2177 _nl_langinfo;
2178 open;
2179 _open;
2180 opendir;
2181 _opendir;
2182 pathconf;
2183 _pathconf;
2184 pause;
2185 _pause;
2186 pclose;
2187 _pclose;
2188 perror;
2189 pipe;
2190 _pipe;
2191 poll;
2192 _poll;
2193 popen;
2194 _popen;
2195 printf;
2196 profil;
2197 _profil;
2198 ptsname;
2199 _ptsname;
2200 putc;
2201 putchar;
2202 putenv;
2203 _putenv;
2204 putmsg;
2205 _putmsg;
2206 putpmsg;
2207 _putpmsg;
2208 puts;
2209 putw;
2210 _putw;
2211 qsort;
2212 raise;
2213 rand;
2214 read;
2215 _read;
2216 readdir;
2217 _readdir;
2218 readlink;
2219 _readlink;
2220 readv;
2221 _readv;
2222 remove;
2223 rename;
2224 _rename;
2225 rewind;
2226 rewinddir;
2227 _rewinddir;
2228 rmdir;
2229 _rmdir;
2230 scanf;
2231 seekdir;
2232 _seekdir;
2233 semctl;
2234 _semctl;
2235 semget;
2236 _semget;
2237 semop;

```

```

2238     _semop;
2239     setbuf;
2240     setcontext;
2241     _setcontext           { FLAGS = NODYNSORT };
2242     setgid;
2243     _setgid;
2244     setgroups;
2245     _setgroups;
2246     setjmp;
2247     setlabel;
2248     setlocale;
2249     setpgid;
2250     _setpgid;
2251     setpgrp;
2252     _setpgrp;
2253     setrlimit;
2254     _setrlimit;
2255     setsid;
2256     _setsid;
2257     setuid;
2258     _setuid;
2259     setvbuf;
2260     shmat;
2261     _shmat;
2262     shmctl;
2263     _shmctl;
2264     shmdt;
2265     _shmdt;
2266     shmget;
2267     _shmget;
2268     sigaction;
2269     _sigaction           { FLAGS = NODYNSORT };
2270     sigaddset;
2271     _sigaddset;
2272     sigaltstack;
2273     _sigaltstack;
2274     sigdelset;
2275     _sigdelset;
2276     sigemptyset;
2277     _sigemptyset;
2278     sigfillset;
2279     _sigfillset;
2280     sighold;
2281     _sighold;
2282     sigignore;
2283     _sigignore;
2284     sigismember;
2285     _sigismember;
2286     siglongjmp;
2287     _siglongjmp;
2288     signal;
2289     sigpause;
2290     _sigpause;
2291     sigpending;
2292     _sigpending;
2293     sigprocmask;
2294     _sigprocmask;
2295     sigrelse;
2296     _sigrelse;
2297     sigsend;
2298     _sigsend;
2299     sigsendset;
2300     _sigsendset;
2301     sigset;
2302     _sigset;
2303     sigsetjmp;

```

```

2304     _sigsetjmp           { FLAGS = NODYNSORT };
2305     sigsuspend;
2306     _sigsuspend;
2307     sleep;
2308     _sleep;
2309     sprintf;
2310     srand;
2311     sscanf;
2312     stat;
2313     _stat;
2314     statvfs;
2315     _statvfs;
2316     stime;
2317     _stime;
2318     strcat;
2319     strchr;
2320     strcmp;
2321     strcoll;
2322     strcpy;
2323     strcspn;
2324     strdup;
2325     _strdup;
2326     strerror;
2327     strftime;
2328     strlen;
2329     strncat;
2330     strncmp;
2331     strncpy;
2332     strpbrk;
2333     strrchr;
2334     strspn;
2335     strstr;
2336     strtod;
2337     strtok;
2338     strtol;
2339     strtoul;
2340     strxfrm;
2341     swab;
2342     _swab;
2343     swapcontext;
2344     _swapcontext;
2345     symlink;
2346     _symlink;
2347     sync;
2348     _sync;
2349     sysconf;
2350     _sysconf;
2351     system;
2352     tcdrain;
2353     _tcdrain;
2354     tcflow;
2355     _tcflow;
2356     tcflush;
2357     _tcflush;
2358     tcgetattr;
2359     _tcgetattr;
2360     tcgetpgrp;
2361     _tcgetpgrp;
2362     tcgetsid;
2363     _tcgetsid;
2364     tcsendbreak;
2365     _tcsendbreak;
2366     tcsetattr;
2367     _tcsetattr;
2368     tcsetpgrp;
2369     _tcsetpgrp;

```

```

2370 tdelete;
2371 _tdelete;
2372 tell;
2373 _tell;
2374 telldir;
2375 _telldir;
2376 tempnam;
2377 _tempnam;
2378 tfind;
2379 _tfind;
2380 time;
2381 _time;
2382 times;
2383 _times;
2384 tmpfile;
2385 tmpnam;
2386 toascii;
2387 _toascii;
2388 tolower;
2389 _tolower;
2390 toupper;
2391 _toupper;
2392 tsearch;
2393 _tsearch;
2394 ttyname;
2395 twalk;
2396 _twalk;
2397 tzset;
2398 _tzset;
2399 ulimit;
2400 _ulimit;
2401 umask;
2402 _umask;
2403 umount;
2404 _umount;
2405 uname;
2406 _uname;
2407 ungetc;
2408 unlink;
2409 _unlink;
2410 unlockpt;
2411 _unlockpt;
2412 utime;
2413 _utime;
2414 vfprintf;
2415 vprintf;
2416 vsprintf;
2417 wait;
2418 _wait;
2419 waitid;
2420 _waitid;
2421 waitpid;
2422 _waitpid;
2423 wcstombs;
2424 wctomb;
2425 write;
2426 _write;
2427 writev;
2428 _writev;
2429 xftw;

2431 $if _ELF32
2432 ptrace;
2433 _ptrace;
2434 $endif

```

```

2436 $if i386
2437     _fxstat;
2438     _lxstat;
2439     _nuname;
2440     _nuname;
2441     _xmknod;
2442     _xstat;
2443 $endif

2445 $if !sparc32
2446     sbrk;
2447 $endif

2449 $if _sparc
2450     __dtou;
2451     __ftou;
2452 $endif

2454 $if sparc32
2455     .div;
2456     .mul;
2457     .rem;
2458     .stret1;
2459     .stret2;
2460     .stret4;
2461     # .stret4 and .stret8 are the same thing
2462     .stret8 { FLAGS = NODYNSORT };
2463     .udiv;
2464     .umul;
2465     .urem;
2466     __Q_add;
2467     __Q_cmp;
2468     __Q_cmpe;
2469     __Q_div;
2470     __Q_dtoq;
2471     __Q_feq;
2472     __Q_fge;
2473     __Q_fgt;
2474     __Q_fle;
2475     __Qflt;
2476     __Q_fne;
2477     __Q_itoq;
2478     __Q_mul;
2479     __Q_neg;
2480     __Q_qtod;
2481     __Q_qtoi;
2482     __Q_qtos;
2483     __Q_qtou;
2484     __Q_sqrt;
2485     __Q_stoq;
2486     __Q_sub;
2487     __Q_utoq;
2488 $endif

2490 $if sparcv9
2491     # __align_cpy_1 is an alias for memcpy. Filter it out of
2492     # the .SUNW_dynsymsort section
2493     __align_cpy_1 { FLAGS = NODYNSORT };
2494     __align_cpy_16;
2495     __align_cpy_2;
2496     __align_cpy_4;
2497     # __align_cpy_8 is same as __align_cpy_16
2498     __align_cpy_8 { FLAGS = NODYNSORT };
2499     __dtoul;
2500     __ftoul;
2501     __Qp_add;

```



```

2502     __Qp_cmp;
2503     __Qp_cmpe;
2504     __Qp_div;
2505     __Qp_dtoq;
2506     __Qp_feq;
2507     __Qp_fge;
2508     __Qp_fgt;
2509     __Qp_fle;
2510     __Qpflt;
2511     __Qp_fne;
2512     __Qp_itoq;
2513     __Qp_mul;
2514     __Qp_neg;
2515     __Qp_qtod;
2516     __Qp_qtoi;
2517     __Qp_qtos;
2518     __Qp_qtoui;
2519     __Qp_qtoux;
2520     __Qp_qtox;
2521     __Qp_sqrt;
2522     __Qp_stoq;
2523     __Qp_sub;
2524     __Qp_uitoq;
2525     __Qp_uxttoq;
2526     __Qp_xtoq;
2527     __sparc_utrap_install;
2528 $endif

2530 # On amd64, we also have SYSVABI_1.3, but it contains a small subset of
2531 # the symbols put in that version on other platforms.
2532 $if amd64
2533 } SYSVABI_1.3;

2535 SYMBOL_VERSION SYSVABI_1.3 {
2536 $endif
2537     global:
2538     $if !_sparc
2539         __flt_rounds;
2540 $endif

2542     protected:
2543         _ctermid;
2544         _getgrgid;
2545         _getgrnam;
2546         _getlogin;
2547         _getpwnam;
2548         _getpwuid;
2549         _ttyname;

2551 $if !sparc32
2552     _sbrk;
2553 $endif

2555 $if _x86
2556     __fpstart;
2557     __fpstart;
2558 $endif
2559 };

2563 # There should never be more than one SUNWprivate version.
2564 # Don't add any more. Add new private symbols to SUNWprivate_1.1

2566 SYMBOL_VERSION SUNWprivate_1.1 {
2567     global:

```

```

2568     __Argv { FLAGS = NODIRECT };
2569     cfree { FLAGS = NODIRECT };
2570     __cswidth;
2571     __ctype_mask;
2572     __environ_lock { FLAGS = NODIRECT };
2573     __inf_read;
2574     __inf_written;
2575     __i_size;
2576     __isnanf { TYPE = FUNCTION; FILTER = libm.so.2 };
2577     __iswrunes;
2578     __libc_threaded;
2579     __lib_version { FLAGS = NODIRECT };
2580     __logb { TYPE = FUNCTION; FILTER = libm.so.2 };
2581     __lone { FLAGS = NODYNSORT };
2582     __lten { FLAGS = NODYNSORT };
2583     __lzero { FLAGS = NODYNSORT };
2584     __malloc_lock;
2585     __memcmp;
2586     __memcpy { FLAGS = NODYNSORT };
2587     __memmove;
2588     __memset;
2589     __modff { TYPE = FUNCTION; FILTER = libm.so.2 };
2590     __nan_read;
2591     __nan_written;
2592     __nextwctype;
2593     __nis_debug_bind;
2594     __nis_debug_calls;
2595     __nis_debug_file;
2596     __nis_debug_rpc;
2597     __nis_prefsrv;
2598     __nis_preftype;
2599     __nis_server;
2600     __nss_default_finders;
2601     __progname { FLAGS = NODIRECT };
2602     __smbuf;
2603     __sp;
2604     __strdupa_str { FLAGS = NODIRECT };
2605     __strdupa_len { FLAGS = NODIRECT };
2606     __tdb_bootstrap;
2607     __threaded;
2608     __thr_probe_getfunc_addr;
2609     __trans_lower;
2610     __trans_upper;
2611     __ubedata;
2612     __xpg6 { FLAGS = NODIRECT };

2614 $if _ELF32
2615     __dladdr { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2616     __dladdr1 { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2617     __dlclose { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2618     __dldump { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2619     __dlerror { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2620     __dlinfo { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2621     __dlmopen { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2622     __dlopen { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2623     __dlsym { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2624     __ld_libc { TYPE = FUNCTION; FILTER = /usr/lib/ld.so.1 };
2625     __sys_errlist;
2626     __sys_errs;
2627     __sys_index;
2628     __sys_nerr { FLAGS = NODYNSORT };
2629     __sys_num_err;
2630 $elif sparcv9
2631     __dladdr { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2632     __dladdr1 { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2633     __dlclose { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };

```

```

2634     _dlldump      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2635     _dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2636     _dlinfo       { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2637     _dlmopen      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2638     _dlopen       { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2639     _dlsym        { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2640     _ld_libc      { TYPE = FUNCTION; FILTER = /usr/lib/sparcv9/ld.so.1 };
2641 $elif amd64
2642     _dladdr       { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2643     _dladdr1      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2644     _dlamd64getunwind { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2645     _dlclose      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2646     _dlldump      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2647     _dlerror      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2648     _dlinfo       { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2649     _dlmopen      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2650     _dlopen       { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2651     _dlsym        { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2652     _ld_libc      { TYPE = FUNCTION; FILTER = /usr/lib/amd64/ld.so.1 };
2653 $else
2654 $error unknown platform
2655 $endif

2657 $if _sparc
2658     __lyday_to_month;
2659     __mon_lengths;
2660     __yday_to_month;
2661 $endif
2662 $if i386
2663     __sse_hw;
2664 $endif

2666     protected:
2667     acctctl;
2668     allocids;
2669     __assert_c99;
2670     __assert_c99;
2671     __assfail;
2672     attr_count;
2673     attr_to_data_type;
2674     attr_to_name;
2675     attr_to_option;
2676     attr_to_xattr_view;
2677     __autofs_sys;
2678     __bufsync;
2679     __cladm;
2680     __class_quadruple;
2681     core_get_default_content;
2682     core_get_default_path;
2683     core_get_global_content;
2684     core_get_global_path;
2685     core_get_options;
2686     core_get_process_content;
2687     core_get_process_path;
2688     core_set_default_content;
2689     core_set_default_path;
2690     core_set_global_content;
2691     core_set_global_path;
2692     core_set_options;
2693     core_set_process_content;
2694     core_set_process_path;
2695     dbm_close_status;
2696     dbm_do_nextkey;
2697     dbm_setdefwrite;
2698     _D_cplx_div;
2699     _D_cplx_div_ix;

```

```

2700     _D_cplx_div_rx;
2701     _D_cplx_mul;
2702     defclose_r;
2703     defcntl;
2704     defcntl_r;
2705     defopen;
2706     defopen_r;
2707     defread;
2708     defread_r;
2709     delete;
2710     _dgettext;
2711     _doprnt;
2712     _doscan;
2713     _errfp;
2714     _errxfp;
2715     exportfs;
2716     _F_cplx_div;
2717     _F_cplx_div_ix;
2718     _F_cplx_div_rx;
2719     _F_cplx_mul;
2720     __fgetwc_xpg5;
2721     __fgetws_xpg5;
2722     findbuf;
2723     findiop;
2724     fini_daemon_priv;
2725     finite;
2726     fork1
2727     forkall
2728     fpclass;
2729     fpgetmask;
2730     fpgetround;
2731     fpgetsticky;
2732     fprintf;
2733     fpsetmask;
2734     fpsetround;
2735     fpsetsticky;
2736     fputwc_xpg5;
2737     fputws_xpg5;
2738     ftw;
2739     _gcvt;
2740     _getarg;
2741     __getcontext;
2742     _getdents;
2743     _get_exit_frame_monitor;
2744     _getfp;
2745     _getgroupsbymember;
2746     _getlogin_r;
2747     _getsp;
2748     _gettsp;
2749     getvmusage;
2750     __getwchar_xpg5;
2751     __getwc_xpg5;
2752     gtty;
2753     __idmap_flush_kcache;
2754     __idmap_reg;
2755     __idmap_unreg;
2756     __init_daemon_priv;
2757     __init_suid_priv;
2758     insert;
2759     inst_sync;
2760     __iswctype;
2761     klpd_create;
2762     klpd_getpath;
2763     klpd_getport;
2764     klpd_getucred;
2765     klpd_register;

```

```

2766 klpd_register_id;
2767 klpd_unregister;
2768 klpd_unregister_id;
2769 _lgrp_home_fast      { FLAGS = NODYNSORT };
2770 _lgrpsys;
2771 _ltostr;
2772 _lock_clear;
2773 _lock_try;
2774 _ltzset;
2775 lwp_self;
2776 makeut;
2777 makeutx;
2778 _mbftowc;
2779 mcfiller;
2780 mntopt;
2781 modctl;
2782 modutx;
2783 msgctl64;
2784 __multi_innetgr;
2785 __mutex_destroy      { FLAGS = NODYNSORT };
2786 mutex_held;
2787 __mutex_init         { FLAGS = NODYNSORT };
2788 __mutex_unlock      { FLAGS = NODYNSORT };
2789 name_to_attr;
2790 nfs_getfh;
2791 nfssvc;
2792 __nfssys;
2793 __nis_get_environment;
2794 __nss_db_state_destr;
2795 nss_default_key2str;
2796 nss_delete;
2797 nss_endent;
2798 nss_getent;
2799 __nss_initf_group;
2800 __nss_initf_netgroup;
2801 __nss_initf_passwd;
2802 __nss_initf_shadow;
2803 nss_packed_arg_init;
2804 nss_packed_context_init;
2805 nss_packed_getkey;
2806 nss_packed_set_status;
2807 nss_search;
2808 nss_setent;
2809 __nss_XbyY_fgets;
2810 __nsw_extended_action_v1;
2811 __nsw_freeconfig_v1;
2812 __nsw_getconfig_v1;
2813 __nthreads;
2814 __openatrrdirat;
2815 option_to_attr;
2816 __priv_bracket;
2817 __priv_relinquish;
2818 pset_assign_forced;
2819 pset_bind_lwp;
2820 __psignal;
2821 __pthread_setcleanupinit;
2822 __putwchar_xpg5;
2823 __putwc_xpg5;
2824 rctlctl;
2825 rctl1ist;
2826 __realbufend;
2827 __resume;
2828 __resume_ret;
2829 rpcsys;
2830 __sbrk_grow_aligned;
2831 scrwidth;

```

```

2832 semctl64;
2833 __semctl64;
2834 set_setcontext_enforcement;
2835 __setbufend;
2836 __set_errno;
2837 setprojctl;
2838 __setregid;
2839 __setreuid;
2840 setsigacthandler;
2841 shmctl64;
2842 __shmctl64;
2843 sigflag;
2844 __signal;
2845 __sigoff;
2846 __sigon;
2847 __so_accept;
2848 __so_bind;
2849 __sockconfig;
2850 __so_connect;
2851 __so_getpeername;
2852 __so_getsockname;
2853 __so_getsockopt;
2854 __so_listen;
2855 __so_recv;
2856 __so_recvfrom;
2857 __so_recvmsg;
2858 __so_send;
2859 __so_sendmsg;
2860 __so_sendto;
2861 __so_setsockopt;
2862 __so_shutdown;
2863 __so_socket;
2864 __so_socketpair;
2865 str2group;
2866 str2passwd;
2867 str2spwd;
2868 __strptime_dontzero;
2869 stty;
2870 syscall;
2871 __sysconfig;
2872 __systemcall;
2873 thr_continue_allmutators;
2874 __thr_continue_allmutators;
2875 thr_continue_mutator;
2876 __thr_continue_mutator;
2877 thr_getstate;
2878 __thr_getstate;
2879 thr_mutators_barrier;
2880 __thr_mutators_barrier;
2881 thr_probe_setup;
2882 __thr_schedctl;
2883 thr_setmutator;
2884 __thr_setmutator;
2885 thr_setstate;
2886 __thr_setstate;
2887 thr_sighndlrinfo;
2888 __thr_sighndlrinfo;
2889 __thr_slot_offset;
2890 thr_suspend_allmutators;
2891 __thr_suspend_allmutators;
2892 thr_suspend_mutator;
2893 __thr_suspend_mutator;
2894 thr_wait_mutator;
2895 __thr_wait_mutator;
2896 __tls_get_addr;
2897 tpool_create;

```

```

2898 tpool_dispatch;
2899 tpool_destroy;
2900 tpool_wait;
2901 tpool_suspend;
2902 tpool_suspended;
2903 tpool_resume;
2904 tpool_member;
2905 __tynam_dev;
2906 __ucred_alloc;
2907 ucred_getamask;
2908 __ucred_getamask;
2909 ucred_getasid;
2910 __ucred_getasid;
2911 ucred_getatid;
2912 __ucred_getatid;
2913 ucred_getauuid;
2914 __ucred_getauuid;
2915 __ulltostr;
2916 __uncached_getgrgid_r;
2917 __uncached_getgrnam_r;
2918 __uncached_getpwnam_r;
2919 __uncached_getpwuid_r;
2920 __ungetwc_xpg5;
2921 __unordered;
2922 utssys;
2923 __verrfp;
2924 __verrxfp;
2925 __vwarnfp;
2926 __vwarnxfp;
2927 __warnfp;
2928 __warnxfp;
2929 __wcsftime_xpg5;
2930 __wcstok_xpg5;
2931 wdbindf;
2932 wdchkind;
2933 wddelim;
2934 __wrtchk;
2935 __xflsbuf;
2936 __xgetwidth;
2937 zone_add_datalink;
2938 zone_boot;
2939 zone_check_datalink;
2940 zone_create;
2941 zone_destroy;
2942 zone_enter;
2943 zone_getattr;
2944 zone_get_id;
2945 zone_list;
2946 zone_list_datalink;
2947 zonept;
2948 zone_remove_datalink;
2949 zone_setattr;
2950 zone_shutdown;
2951 zone_version;

2953 $if _ELF32
2954 __divdi3;
2955 __file_set;
2956 __fprintf_c89;
2957 __fscanf_c89;
2958 __fwprintf_c89;
2959 __fwscanf_c89;
2960 __imaxabs_c89;
2961 __imaxdiv_c89;
2962 __moddi3;
2963 __printf_c89;

```

```

2964 __scanf_c89;
2965 __snprintf_c89;
2966 __sprintf_c89;
2967 __sscanf_c89;
2968 __strtoimax_c89;
2969 __strtoumax_c89;
2970 __swprintf_c89;
2971 __swscanf_c89;
2972 __udivdi3;
2973 __umoddi3;
2974 __vfprintf_c89;
2975 __vfscanf_c89;
2976 __vfwprintf_c89;
2977 __vfwscanf_c89;
2978 __vprintf_c89;
2979 __vscanf_c89;
2980 __vsnprintf_c89;
2981 __vsprintf_c89;
2982 __vsscanf_c89;
2983 __vswprintf_c89;
2984 __vswscanf_c89;
2985 __vwprintf_c89;
2986 __vwscanf_c89;
2987 __wcstoimax_c89;
2988 __wcstoumax_c89;
2989 __wprintf_c89;
2990 __wscanf_c89;
2991 $endif

2993 $if _sparc
2994 __cerror;
2995 install_utrap;
2996 __install_utrap;
2997 nop;
2998 __Q_cplx_div;
2999 __Q_cplx_div_ix;
3000 __Q_cplx_div_rx;
3001 __Q_cplx_lr_div;
3002 __Q_cplx_lr_div_ix;
3003 __Q_cplx_lr_div_rx;
3004 __Q_cplx_lr_mul;
3005 __Q_cplx_mul;
3006 __QgetRD;
3007 __xregs_clrptr;
3008 $endif

3010 $if sparc32
3011 __ashldi3;
3012 __ashrdi3;
3013 __cerror64;
3014 __cmpdi2;
3015 __floatdidf;
3016 __floatdisf;
3017 __floatundidf;
3018 __floatundisf;
3019 __lshrdi3;
3020 __muldi3;
3021 __ucmpdi2;
3022 $endif

3024 $if _x86
3025 __D_cplx_lr_div;
3026 __D_cplx_lr_div_ix;
3027 __D_cplx_lr_div_rx;
3028 __F_cplx_lr_div;
3029 __F_cplx_lr_div_ix;

```

```

3030     _F_cplx_lr_div_rx;
3031     __fltrounds;
3032     sysi86;
3033     __sysi86;
3034     _X_cplx_div;
3035     _X_cplx_div_ix;
3036     _X_cplx_div_rx;
3037     _X_cplx_lr_div;
3038     _X_cplx_lr_div_ix;
3039     _X_cplx_lr_div_rx;
3040     _X_cplx_mul;
3041     __xgetRD;
3042     __xtol;
3043     __xtoll;
3044     __xtoul;
3045     __xtoull;
3046 $endif

3048 $if i386
3049     __divrem64;
3050     __tls_get_addr;
3051     __udivrem64;
3052 $endif

3054 # The following functions should not be exported from libc,
3055 # but /lib/libm.so.2, some older versions of the Studio
3056 # compiler/debugger components, and some ancient programs
3057 # found in /usr/dist reference them. When we no longer
3058 # care about these old and broken binary objects, these
3059 # symbols should be deleted.
3060     brk { FLAGS = NODYNSORT };
3061     _cond_broadcast { FLAGS = NODYNSORT };
3062     _cond_init { FLAGS = NODYNSORT };
3063     _cond_signal { FLAGS = NODYNSORT };
3064     _cond_wait { FLAGS = NODYNSORT };
3065     _ecvt { FLAGS = NODYNSORT };
3066     _fcvt { FLAGS = NODYNSORT };
3067     _getc_unlocked { FLAGS = NODYNSORT };
3068     _llseek { FLAGS = NODYNSORT };
3069     _pthread_attr_getdetachstate { FLAGS = NODYNSORT };
3070     _pthread_attr_getinheritsched { FLAGS = NODYNSORT };
3071     _pthread_attr_getschedparam { FLAGS = NODYNSORT };
3072     _pthread_attr_getschedpolicy { FLAGS = NODYNSORT };
3073     _pthread_attr_getscope { FLAGS = NODYNSORT };
3074     _pthread_attr_getstackaddr { FLAGS = NODYNSORT };
3075     _pthread_attr_getstacksize { FLAGS = NODYNSORT };
3076     _pthread_attr_init { FLAGS = NODYNSORT };
3077     _pthread_condattr_getpshared { FLAGS = NODYNSORT };
3078     _pthread_condattr_init { FLAGS = NODYNSORT };
3079     _pthread_cond_init { FLAGS = NODYNSORT };
3080     _pthread_create { FLAGS = NODYNSORT };
3081     _pthread_getschedparam { FLAGS = NODYNSORT };
3082     _pthread_join { FLAGS = NODYNSORT };
3083     _pthread_key_create { FLAGS = NODYNSORT };
3084     _pthread_mutexattr_getprioceiling { FLAGS = NODYNSORT };
3085     _pthread_mutexattr_getprotocol { FLAGS = NODYNSORT };
3086     _pthread_mutexattr_getpshared { FLAGS = NODYNSORT };
3087     _pthread_mutexattr_init { FLAGS = NODYNSORT };
3088     _pthread_mutex_getprioceiling { FLAGS = NODYNSORT };
3089     _pthread_mutex_init { FLAGS = NODYNSORT };
3090     _pthread_sigmask { FLAGS = NODYNSORT };
3091     _rwlock_init { FLAGS = NODYNSORT };
3092     _rw_rdlock { FLAGS = NODYNSORT };
3093     _rw_unlock { FLAGS = NODYNSORT };
3094     _rw_wrlock { FLAGS = NODYNSORT };
3095     _sbrk_unlocked { FLAGS = NODYNSORT };

```

```

3096     _select { FLAGS = NODYNSORT };
3097     _sema_init { FLAGS = NODYNSORT };
3098     _sema_post { FLAGS = NODYNSORT };
3099     _sema_trywait { FLAGS = NODYNSORT };
3100     _sema_wait { FLAGS = NODYNSORT };
3101     _sysfs { FLAGS = NODYNSORT };
3102     _thr_create { FLAGS = NODYNSORT };
3103     _thr_exit { FLAGS = NODYNSORT };
3104     _thr_getprio { FLAGS = NODYNSORT };
3105     _thr_getspecific { FLAGS = NODYNSORT };
3106     _thr_join { FLAGS = NODYNSORT };
3107     _thr_keycreate { FLAGS = NODYNSORT };
3108     _thr_kill { FLAGS = NODYNSORT };
3109     _thr_main { FLAGS = NODYNSORT };
3110     _thr_self { FLAGS = NODYNSORT };
3111     _thr_setspecific { FLAGS = NODYNSORT };
3112     _thr_sigsetmask { FLAGS = NODYNSORT };
3113     _thr_stksegment { FLAGS = NODYNSORT };
3114     _ungetc_unlocked { FLAGS = NODYNSORT };

3116     local:
3117     __imax_lldiv { FLAGS = NODYNSORT };
3118     __ti_thr_self { FLAGS = NODYNSORT };
3119     *;

3121 $if lf64
3122     __seekdir64 { FLAGS = NODYNSORT };
3123     __telldir64 { FLAGS = NODYNSORT };
3124 $endif

3126 $if _sparc
3127     __cerror { FLAGS = NODYNSORT };
3128 $endif

3130 $if sparc32
3131     __cerror64 { FLAGS = NODYNSORT };
3132 $endif

3134 $if sparcv9
3135     __cleanup { FLAGS = NODYNSORT };
3136 $endif

3138 $if i386
3139     __syscall6 { FLAGS = NODYNSORT };
3140     __systemcall6 { FLAGS = NODYNSORT };
3141 $endif

3143 $if amd64
3144     __tls_get_addr { FLAGS = NODYNSORT };
3145 $endif
3146 };

```

```

*****
25719 Sat Sep 28 12:17:13 2013
new/usr/src/lib/libc/sparc/Makefile.com
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
24 #
25 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #

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

35 # objects are grouped by source directory

37 # Symbol capabilities objects.
38 EXTPICS=
39 $(LIBCDIR)/capabilities/sun4u/sparc/pics/symcap.o \
40 $(LIBCDIR)/capabilities/sun4u-opl/sparc/pics/symcap.o \
41 $(LIBCDIR)/capabilities/sun4u-us3-hwcap1/sparc/pics/symcap.o \
42 $(LIBCDIR)/capabilities/sun4u-us3-hwcap2/sparc/pics/symcap.o \
43 $(LIBCDIR)/capabilities/sun4v-hwcap1/sparc/pics/symcap.o \
44 $(LIBCDIR)/capabilities/sun4v-hwcap2/sparc/pics/symcap.o

46 # local objects
47 STRETS=
48      stret1.o
49      stret2.o
50      stret4.o

52 CRTOBSJ=
53      _ftou.o
54      cerror.o
55      cerror64.o
56      hwmuldiv.o

58 DYNOBJS=
59      _rtbootld.o

```

```

61 FPOBJS=
62      _D_cplx_div.o
63      _D_cplx_div_ix.o
64      _D_cplx_div_rx.o
65      _D_cplx_mul.o
66      _F_cplx_div.o
67      _F_cplx_div_ix.o
68      _F_cplx_div_rx.o
69      _F_cplx_mul.o
70      _Q_add.o
71      _Q_cmp.o
72      _Q_cmpe.o
73      _Q_cplx_div.o
74      _Q_cplx_div_ix.o
75      _Q_cplx_div_rx.o
76      _Q_cplx_lr_div.o
77      _Q_cplx_lr_div_ix.o
78      _Q_cplx_lr_div_rx.o
79      _Q_cplx_lr_mul.o
80      _Q_cplx_mul.o
81      _Q_div.o
82      _Q_dtoq.o
83      _Q_fcc.o
84      _Q_itoq.o
85      _Q_lltoq.o
86      _Q_mul.o
87      _Q_neg.o
88      _Q_qtod.o
89      _Q_qtoi.o
90      _Q_qtos.o
91      _Q_qtou.o
92      _Q_scl.o
93      _Q_set_except.o
94      _Q_sqrt.o
95      _Q_stoq.o
96      _Q_sub.o
97      _Q_ulltoq.o
98      _Q_utoq.o
99      __quad_mag.o

101 FPASMOBJS=
102      _Q_get_rp_rd.o
103      fpgetmask.o
104      fpgetrnd.o
105      fpgetsticky.o
106      fpsetmask.o
107      fpsetrnd.o
108      fpsetsticky.o

110 $(__GNUC)FPASMOBJS +=
111      __quad.o

113 ATOMICOBJS=
114      atomic.o

116 XATTROBJS=
117      xattr_common.o

119 COMOBJS=
120      bcmp.o
121      bcopy.o
122      bzero.o
123      bsearch.o
124      memccpy.o
125      qsort.o

```

```

126      strtol.o      \
127      strtoul.o    \
128      strtoll.o    \
129      strtoull.o   \

131 DTRACEOBSJ=      \
132      dtrace_data.o \

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

169 # sysobjs that contain large-file interfaces
170 COMSYSOBSJ64=     \
171      fstatvfs64.o \
172      getdents64.o \
173      getrlimit64.o \
174      lseek64.o    \
175      mmap64.o     \
176      pread64.o    \
177      pwrite64.o   \
178      setrlimit64.o \
179      statvfs64.o  \

181 SYSOBSJ64=       \

183 COMSYSOBSJ=      \
184      __clock_timer.o \
185      __getloadavg.o \
186      __rusagesys.o  \
187      __signotify.o \
188      __sigrt.o      \
189      __time.o       \
190      __lgrp_home_fast.o \
191      __lgrpsys.o    \

```

```

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

```

```

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

```

```

324      syscall.o         \
325      tls_get_addr.o    \
326      uadmin.o          \
327      umount.o          \
328      uname.o           \
329      vforkx.o          \
\
331 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
332 PORTGEN64= \
333     _xftw64.o         \
334     attropen64.o      \
335     ftw64.o           \
336     mkstemp64.o       \
337     nftw64.o          \
338     tell64.o          \
339     truncate64.o      \
\
341 # objects from source under $(LIBCDIR)/port
342 PORTFP= \
343     __flt_decim.o     \
344     __flt_rounds.o    \
345     __tbl_10_b.o      \
346     __tbl_10_h.o      \
347     __tbl_10_s.o      \
348     __tbl_2_b.o       \
349     __tbl_2_h.o       \
350     __tbl_2_s.o       \
351     __tbl_fdq.o       \
352     __tbl_tens.o      \
353     __x_power.o       \
354     __base_sup.o      \
355     aconvert.o        \
356     decimal_bin.o     \
357     double_decim.o    \
358     econvert.o        \
359     fconvert.o        \
360     file_decim.o      \
361     finite.o           \
362     fp_data.o         \
363     func_decim.o      \
364     gconvert.o        \
365     hex_bin.o         \
366     ieee_globals.o    \
367     pack_float.o      \
368     sigfpe.o          \
369     string_decim.o    \
370     ashldi3.o         \
371     ashrdi3.o         \
372     cmpdi2.o          \
373     divdi3.o          \
374     floatdidf.o       \
375     floatdisf.o       \
376     floatundidf.o     \
377     floatundisf.o     \
378     lshrdi3.o         \
379     moddi3.o          \
380     muldi3.o          \
381     qdivrem.o         \
382     ucmpdi2.o        \
383     udivdi3.o         \
384     umoddi3.o         \
\
386 PORTGEN= \
387     __env_data.o      \
388     __ftoll.o         \
389     __ftoull.o        \

```



```

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

```

```

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

```

```

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

```

```

588 strncat.o \
589 strndup.o \
590 strpbrk.o \
591 strrchr.o \
592 strsep.o \
593 strsignal.o \
594 strspn.o \
595 strstr.o \
596 strtod.o \
597 strtointmax.o \
598 strtok.o \
599 strtok_r.o \
600 strtoumax.o \
601 swab.o \
602 swapctl.o \
603 sysconf.o \
604 syslog.o \
605 tcdrain.o \
606 tcflow.o \
607 tcflush.o \
608 tcgetattr.o \
609 tcgetpgrp.o \
610 tcgetsid.o \
611 tcsendbreak.o \
612 tcsetattr.o \
613 tcsetpgrp.o \
614 tell.o \
615 telldir.o \
616 tfind.o \
617 time_data.o \
618 time_gdata.o \
619 tls_data.o \
620 truncate.o \
621 tsdalloc.o \
622 tsearch.o \
623 ttyname.o \
624 ttyslot.o \
625 ualarm.o \
626 ucred.o \
627 valloc.o \
628 vlfmt.o \
629 vpfmt.o \
630 waitpid.o \
631 walkstack.o \
632 wdata.o \
633 xgetwidth.o \
634 xpg4.o \
635 xpg6.o \
637 PORTPRINT_W= \
638     doprnt_w.o \
640 PORTPRINT= \
641     asprintf.o \
642     doprnt.o \
643     fprintf.o \
644     printf.o \
645     snprintf.o \
646     sprintf.o \
647     vfprintf.o \
648     vprintf.o \
649     vsnprintf.o \
650     vsprintf.o \
651     vwprintf.o \
652     wprintf.o \

```

```

654 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
655 PORTPRINT_C89= \
656     vfprintf_c89.o \
657     vprintf_c89.o \
658     vsnprintf_c89.o \
659     vsprintf_c89.o \
660     vwprintf_c89.o

662 PORTSTDIO_C89= \
663     vscanf_c89.o \
664     vwscanf_c89.o

666 # portable stdio objects that contain large file interfaces.
667 # Note: fopen64 is a special case, as we build it small.
668 PORTSTDIO64= \
669     fopen64.o \
670     fpos64.o

672 PORTSTDIO_W= \
673     doscan_w.o

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

```

```

720     setbuffer.o \
721     setvbuf.o \
722     system.o \
723     tempnam.o \
724     tmpfile.o \
725     tmpnam_r.o \
726     ungetc.o \
727     mse.o \
728     vscanf.o \
729     vwscanf.o \
730     wscanf.o

732 PORTI18N= \
733     getwchar.o \
734     putwchar.o \
735     putws.o \
736     strcasecmp.o \
737     strcasestr.o \
738     strncasecmp.o \
739     strtows.o \
740     wcsnlen.o \
741     wcstoimax.o \
742     wcstol.o \
743     wcstoul.o \
744     wcs wcs.o \
745     wscasecmp.o \
746     wscat.o \
747     wscr.o \
748     wscmp.o \
749     wscpy.o \
750     wscspn.o \
751     wsdup.o \
752     wslen.o \
753     wsncasecmp.o \
754     wscat.o \
755     wscmp.o \
756     wscnpy.o \
757     wspbrk.o \
758     wsprintf.o \
759     wsrchr.o \
760     wsscanf.o \
761     wssp.o \
762     wstod.o \
763     wstok.o \
764     wstol.o \
765     wstoll.o \
766     wsxfrm.o \
767     wmemchr.o \
768     wmemcmp.o \
769     wmemcpy.o \
770     wmemmove.o \
771     wmemset.o \
772     wcstr.o \
773     gettext.o \
774     gettext_real.o \
775     gettext_util.o \
776     gettext_gnu.o \
777     plural_parser.o \
778     wdresolve.o \
779     _ctype.o \
780     isascii.o \
781     isdigit.o \
782     toascii.o

784 PORTI18N_COND= \
785     wcstol_longlong.o

```

```

786      wcstoul_longlong.o
788  PORTLOCALE= \
789      big5.o \
790      btowc.o \
791      collate.o \
792      collcmp.o \
793      euc.o \
794      fnmatch.o \
795      fgetwc.o \
796      fgetws.o \
797      fix_grouping.o \
798      fputwc.o \
799      fputws.o \
800      fwide.o \
801      gb18030.o \
802      gb2312.o \
803      gbk.o \
804      getdate.o \
805      iswctype.o \
806      ldpart.o \
807      lmessages.o \
808      lnumeric.o \
809      lmonetary.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      strcoll.o \
833      strfmon.o \
834      strftime.o \
835      strptime.o \
836      strxfrm.o \
837      table.o \
838      timelocal.o \
839      tolower.o \
840      tolower.o \
841      ungetwc.o \
842      utf8.o \
843      wctomb.o \
844      wcscoll.o \
845      wcsftime.o \
846      wcsnrtombs.o \
847      wcsrtombs.o \
848      wcstombs.o \
849      wcswidth.o \
850      wcsxfrm.o \
851      wctob.o \

```

```

852      wctomb.o \
853      wctrans.o \
854      wctype.o \
855      wcwidth.o \
856      wscoll.o
858  AIOBJS= \
859      aio.o \
860      aio_alloc.o \
861      posix_aio.o
863  RTOBJS= \
864      clock_timer.o \
865      mqueue.o \
866      pos4obj.o \
867      sched.o \
868      sem.o \
869      shm.o \
870      sigev_thread.o
872  TPOOLBJS= \
873      thread_pool.o
875  THREADSOBJS= \
876      alloc.o \
877      assfail.o \
878      cancel.o \
879      door_calls.o \
880      pthr_attr.o \
881      pthr_barrier.o \
882      pthr_cond.o \
883      pthr_mutex.o \
884      pthr_rwlock.o \
885      pthread.o \
886      rwlock.o \
887      scalls.o \
888      sema.o \
889      sigaction.o \
890      spawn.o \
891      synch.o \
892      tdb_agent.o \
893      thr.o \
894      thread_interface.o \
895      tls.o \
896      tsd.o
898  THREADSMACHOBJS= \
899      machdep.o
901  THREADSASMOBJS= \
902      asm_subr.o
904  UNICODEOBJS= \
905      u8_textprep.o \
906      uconv.o
908  UNWINDMACHOBJS= \
909      unwind.o
911  UNWINDASMOBJS= \
912      unwind_frame.o
914  # objects that implement the transitional large file API
915  PORTSYS64= \
916      lockf64.o \
917      stat64.o

```

```

919 PORTSYS= \
920     _autofssys.o \
921     access.o \
922     acctl.o \
923    bsd_signal.o \
924     chmod.o \
925     chown.o \
926     corectl.o \
927     exaccts.o \
928     execl.o \
929     execl.o \
930     execv.o \
931     fcntl.o \
932     getpagesizes.o \
933     getpeerucd.o \
934     inst_sync.o \
935     issetugid.o \
936     label.o \
937     link.o \
938     lockf.o \
939     lwp.o \
940     lwp_cond.o \
941     lwp_rwlock.o \
942     lwp_sigmask.o \
943     meminfosys.o \
944     mkdir.o \
945     mknod.o \
946     msgsys.o \
947     nfssys.o \
948     open.o \
949     pgrpsys.o \
950     posix_sigwait.o \
951     ppriv.o \
952     psetsys.o \
953     rctlsys.o \
954     readlink.o \
955     rename.o \
956     sbrk.o \
957     semsys.o \
958     set_errno.o \
959     sharefs.o \
960     shmsys.o \
961     sidsys.o \
962     siginterrupt.o \
963     signal.o \
964     sigpending.o \
965     sigstack.o \
966     stat.o \
967     symlink.o \
968     tasksys.o \
969     time.o \
970     time_util.o \
971     ucontext.o \
972     unlink.o \
973     ustat.o \
974     utimesys.o \
975     zone.o

977 PORTREGEX= \
978     glob.o \
979     regcmp.o \
980     regex.o \
981     wordexp.o

983 VALUES= values-Xa.o

```

```

985 MOSTOBS= \
986     $(STRETS) \
987     $(CRTOBS) \
988     $(DYNOBJS) \
989     $(FPOBJS) \
990     $(FPASMOBJS) \
991     $(ATOMICOBJS) \
992     $(XATTOBJS) \
993     $(COMOBJS) \
994     $(DTRACEOBJS) \
995     $(GENOBJS) \
996     $(PRFOBJS) \
997     $(PORTFP) \
998     $(PORTGEN) \
999     $(PORTGEN64) \
1000    $(PORTI18N) \
1001    $(PORTI18N_COND) \
1002    $(PORTLOCALE) \
1003    $(PORTPRINT) \
1004    $(PORTPRINT_C89) \
1005    $(PORTPRINT_W) \
1006    $(PORTREGEX) \
1007    $(PORTSTDIO) \
1008    $(PORTSTDIO64) \
1009    $(PORTSTDIO_C89) \
1010    $(PORTSTDIO_W) \
1011    $(PORTSYS) \
1012    $(PORTSYS64) \
1013    $(AIOOBJS) \
1014    $(RTOBJS) \
1015    $(TPOOLBJS) \
1016    $(THREADSOBJS) \
1017    $(THREADSMACHOBJS) \
1018    $(THREADSASMOBJS) \
1019    $(UNICODEOBJS) \
1020    $(UNWINDMACHOBJS) \
1021    $(UNWINDASMOBJS) \
1022    $(COMSYSOBJS) \
1023    $(SYSOBJS) \
1024    $(COMSYSOBJS64) \
1025    $(SYSOBJS64) \
1026    $(VALUES)

1028 TRACEOBS= \
1029     plockstat.o

1031 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
1032 # modules whose source is provided in the $(SRC)/lib/common directory.
1033 # This must be done because otherwise the Sun C compiler would insert
1034 # its own versions of these modules and those versions contain code
1035 # to call out to C++ initialization functions. Such C++ initialization
1036 # functions can call back into libc before thread initialization is
1037 # complete and this leads to segmentation violations and other problems.
1038 # Since libc contains no C++ code, linking with the minimal crti.o and
1039 # crtn.o modules is safe and avoids the problems described above.
1040 OBJECTS= $(CRTI) $(MOSTOBS) $(CRTN)
1041 CRTSRCS= ../../common/sparc

1043 # include common library definitions
1044 include $(SRC)/lib/Makefile.lib

1046 # we need to override the default SONAME here because we might
1047 # be building a variant object (still libc.so.1, but different filename)
1048 SONAME = libc.so.1

```

```

1050 CFLAGS += $(CCVERBOSE)

1052 # This is necessary to avoid problems with calling _ex_unwind().
1053 # We probably don't want any inlining anyway.
1054 CFLAGS += -xinline=

1056 CERRWARN += -_gcc=-Wno-parentheses
1057 CERRWARN += -_gcc=-Wno-switch
1058 CERRWARN += -_gcc=-Wno-uninitialized
1059 CERRWARN += -_gcc=-Wno-unused-value
1060 CERRWARN += -_gcc=-Wno-unused-label
1061 CERRWARN += -_gcc=-Wno-unused-variable
1062 CERRWARN += -_gcc=-Wno-type-limits
1063 CERRWARN += -_gcc=-Wno-char-subscripts
1064 CERRWARN += -_gcc=-Wno-clobbered
1065 CERRWARN += -_gcc=-Wno-unused-function
1066 CERRWARN += -_gcc=-Wno-address

1068 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1069 # enables ASSERT() checking in the threads portion of the library.
1070 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1071 THREAD_DEBUG =
1072 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1074 # Make string literals read-only to save memory.
1075 CFLAGS += $(XSTRCONST)

1077 ALTPICCS = $(TRACEOBSJCS:=pics/%)

1079 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICCS) $(ALTPICCS) $(EXTPICCS)

1081 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1083 CFLAGS += $(EXTN_CFLAGS)
1084 CPPFLAGS = -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1085 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1086 ASFLAGS = $(EXTN_ASFLAGS) -K pic -P -D__STDC__ -D_ASM $(CPPFLAGS) $(sparc_

1088 # As a favor to the dtrace syscall provider, libc still calls the
1089 # old syscall traps that have been obsoleted by the *at() interfaces.
1090 # Delete this to compile libc using only the new *at() system call traps
1091 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1093 # Inform the run-time linker about libc specialized initialization
1094 RTLDINFO = -z rtldinfo=tls_rtldinfo
1095 DYNFLAGS += $(RTLDINFO)

1097 # Force libc's internal references to be resolved immediately upon loading
1098 # in order to avoid critical region problems. Since almost all libc symbols
1099 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1100 DYNFLAGS += -znw

1102 DYNFLAGS += -e __rtboot
1103 DYNFLAGS += $(EXTN_DYNFLAGS)

1105 # Inform the kernel about the initial DTrace area (in case
1106 # libc is being used as the interpreter / runtime linker).
1107 DTRACE_DATA = -zdtrace=dtrace_data
1108 DYNFLAGS += $(DTRACE_DATA)

1110 # DTrace needs an executable data segment.
1111 MAPFILE.NED=

1113 BUILD.s= $(AS) $(ASFLAGS) $< -o $@

1115 # Override this top level flag so the compiler builds in its native

```

```

1116 # C99 mode. This has been enabled to support the complex arithmetic
1117 # added to libc.
1118 C99MODE= $(C99_ENABLE)

1120 # libc method of building an archive
1121 # The "$(GREP) -v ' L ' " part is necessary only until
1122 # lorder is fixed to ignore thread-local variables.
1123 BUILD.AR= $(RM) $@ ; \
1124 $(AR) q $@ `$(LORDER) $(MOSTOBSJCS:=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1126 # extra files for the clean target
1127 CLEANFILES= \
1128 $(LIBCDIR)/port/gen/errlst.c \
1129 $(LIBCDIR)/port/gen/new_list.c \
1130 assym.h \
1131 genassym \
1132 $(LIBCBASE)/crt/_rtld.s \
1133 $(LIBCBASE)/crt/_rtbootld.s \
1134 pics/_rtbootld.o \
1135 pics/crti.o \
1136 pics/crtn.o \
1137 $(ALTPICCS)

1139 CLOBBERFILES += $(LIB_PIC)

1141 # list of C source for lint
1142 SRCS= \
1143 $(ATOMICOBJS:=.o=$(SRC)/common/atomic/%.c) \
1144 $(XATTROBJS:=.o=$(SRC)/common/xattr/%.c) \
1145 $(COMOBJS:=.o=$(SRC)/common/util/%.c) \
1146 $(DTRACEOBSJCS:=.o=$(SRC)/common/dtrace/%.c) \
1147 $(PORTFP:=.o=$(LIBCDIR)/port/fp/%.c) \
1148 $(PORTGEN:=.o=$(LIBCDIR)/port/gen/%.c) \
1149 $(PORTI18N:=.o=$(LIBCDIR)/port/i18n/%.c) \
1150 $(PORTLOCALE:=.o=$(LIBCDIR)/port/locale/%.c) \
1151 $(PORTPRINT:=.o=$(LIBCDIR)/port/print/%.c) \
1152 $(PORTREGEX:=.o=$(LIBCDIR)/port/regex/%.c) \
1153 $(PORTSTDIO:=.o=$(LIBCDIR)/port/stdio/%.c) \
1154 $(PORTSYS:=.o=$(LIBCDIR)/port/sys/%.c) \
1155 $(AIOOBSJCS:=.o=$(LIBCDIR)/port/aio/%.c) \
1156 $(RTOBJS:=.o=$(LIBCDIR)/port/rt/%.c) \
1157 $(TPOOLOBSJCS:=.o=$(LIBCDIR)/port/tpool/%.c) \
1158 $(THREADSOBSJCS:=.o=$(LIBCDIR)/port/threads/%.c) \
1159 $(THREADSMACHOBSJCS:=.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1160 $(UNICODEOBSJCS:=.o=$(SRC)/common/unicode/%.c) \
1161 $(UNWINDMACHOBSJCS:=.o=$(LIBCDIR)/port/unwind/%.c) \
1162 $(FPOBJS:=.o=$(LIBCDIR)/$(MACH)/fp/%.c) \
1163 $(LIBCBASE)/crt/_ftou.c \
1164 $(LIBCBASE)/gen/_xregs_clrptr.c \
1165 $(LIBCBASE)/gen/byteorder.c \
1166 $(LIBCBASE)/gen/ecvt.c \
1167 $(LIBCBASE)/gen/getctxt.c \
1168 $(LIBCBASE)/gen/lmul.c \
1169 $(LIBCBASE)/gen/makectxt.c \
1170 $(LIBCBASE)/gen/signfolst.c \
1171 $(LIBCBASE)/gen/siglongjmp.c \
1172 $(LIBCBASE)/gen/swapctxt.c \
1173 $(LIBCBASE)/sys/ptrace.c \
1174 $(LIBCBASE)/sys/uadmin.c

1176 # conditional assignments
1177 $(DYNLIB) := CRTI = crt_i.o
1178 $(DYNLIB) := CRTN = crtn.o

1180 # Files which need the threads .il inline template
1181 TIL= \

```

```

1182 aio.o \
1183 alloc.o \
1184 assfail.o \
1185 atexit.o \
1186 atfork.o \
1187 cancel.o \
1188 door_calls.o \
1189 err.o \
1190 errno.o \
1191 gettxt.o \
1192 lwp.o \
1193 ma.o \
1194 machdep.o \
1195 posix_aio.o \
1196 pthr_attr.o \
1197 pthr_barrier.o \
1198 pthr_cond.o \
1199 pthr_mutex.o \
1200 pthr_rwlock.o \
1201 pthread.o \
1202 rand.o \
1203 rwlock.o \
1204 scalls.o \
1205 sched.o \
1206 sema.o \
1207 sigaction.o \
1208 sigev_thread.o \
1209 spawn.o \
1210 stack.o \
1211 swaptxt.o \
1212 synch.o \
1213 tdb_agent.o \
1214 thr.o \
1215 thread_interface.o \
1216 thread_pool.o \
1217 tls.o \
1218 tsd.o \
1219 unwind.o

1221 $(TIL:%=pics/%) := CFLAGS += $(LIBCBASE)/threads/sparc.il

1223 # special kludge for inlines with 'cas':
1224 pics/rwlock.o pics/synch.o pics/lwp.o pics/door_calls.o := \
1225     sparc_CFLAGS += -gcc=-Wa,-xarch=v8plus

1227 # Files in port/fp subdirectory that need base.il inline template
1228 IL=
1229     __flt_decim.o \
1230     decimal_bin.o

1232 $(IL:%=pics/%) := CFLAGS += $(LIBCBASE)/fp/base.il

1234 # Files in fp subdirectory which need __quad.il inline template
1235 QIL=
1236     _Q_add.o \
1237     _Q_cmp.o \
1238     _Q_cmpe.o \
1239     _Q_div.o \
1240     _Q_dtoq.o \
1241     _Q_fcc.o \
1242     _Q_mul.o \
1243     _Q_qtod.o \
1244     _Q_qtoi.o \
1245     _Q_qtos.o \
1246     _Q_qtou.o \
1247     _Q_sqrt.o

```

```

1248     _Q_stoq.o \
1249     _Q_sub.o

1251 $(QIL:%=pics/%) := CFLAGS += $(LIBCDIR)/$(MACH)/fp/__quad.il
1252 pics/_Q%.o := sparc_COPTFLAG = -xO4 -dalign
1253 pics/__quad%.o := sparc_COPTFLAG = -xO4 -dalign

1255 # large-file-aware components that should be built large

1257 $(COMSYSOBS64:%=pics/%) := \
1258     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1260 $(SYSOBS64:%=pics/%) := \
1261     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1263 $(PORTGEN64:%=pics/%) := \
1264     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1266 $(PORTSTDIO64:%=pics/%) := \
1267     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1269 $(PORTSYS64:%=pics/%) := \
1270     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1272 $(PORTSTDIO_W:%=pics/%) := \
1273     CPPFLAGS += -D_WIDE

1275 $(PORTPRINT_W:%=pics/%) := \
1276     CPPFLAGS += -D_WIDE

1278 # printf/scanf functions to support c89-sized intmax_t variables
1279 $(PORTPRINT_C89:%=pics/%) := \
1280     CPPFLAGS += -D_C89_INTMAX32

1282 $(PORTSTDIO_C89:%=pics/%) := \
1283     CPPFLAGS += -D_C89_INTMAX32

1285 $(PORTI18N_COND:%=pics/%) := \
1286     CPPFLAGS += -D_WCS_LONGLONG

1288 # Files which need extra optimization
1289 pics/getenv.o := sparc_COPTFLAG = -xO4

1291 .KEEP_STATE:

1293 all: $(LIBS) $(LIB_PIC)

1295 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1296 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1297 lint := LINTFLAGS += -mn

1299 lint:
1300     @echo $(LINT.c) ... $(LDLIBS)
1301     @$$(LINT.c) $(SRCS) $(LDLIBS)

1303 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-1c
1304 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1305 $(LINTLIB):= LINTFLAGS=-nvx

1307 # object files that depend on inline template
1308 $(TIL:%=pics/%) := $(LIBCBASE)/threads/sparc.il
1309 $(IL:%=pics/%) := $(LIBCBASE)/fp/base.il
1310 $(QIL:%=pics/%) := $(LIBCDIR)/$(MACH)/fp/__quad.il

1312 # include common libc targets
1313 include $(LIBCDIR)/Makefile.targ

```

```

1315 # We need to strip out all CTF and DOF data from the static library
1316 $(LIB_PIC) := DIR = pics
1317 $(LIB_PIC): pics $$ (PICS)
1318     $(BUILD.AR)
1319     $(MCS) -d -n .SUNW_ctf $$@ > /dev/null 2>&l
1320     $(MCS) -d -n .SUNW_dof $$@ > /dev/null 2>&l
1321     $(AR) -ts $$@ > /dev/null
1322     $(POST_PROCESS_A)

1324 # special cases
1325 $(STRETS:%=pics%): $(LIBCBASE)/crt/stret.s
1326     $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=) $(LIBCBASE)/crt/stret.s -o $$@
1327     $(POST_PROCESS_O)

1329 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.
1330     $(CC) $(CPPFLAGS) $(CTF_FLAGS) -o -s -K pic \
1331     $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1332     $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $$@
1333     $(RM) $(LIBCBASE)/crt/_rtld.s

1335 # partially built from C source
1336 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1337     $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $$@
1338     $(CTFCONVERT_O)

1340 ASSYMDEP_OBJS= \
1341     _lwp_mutex_unlock.o \
1342     _stack_grow.o \
1343     asm_subr.o \
1344     setjmp.o \
1345     smt_pause.o \
1346     tls_get_addr.o \
1347     unwind_frame.o \
1348     vforkx.o

1350 $(ASSYMDEP_OBJS:%=pics%) := CPPFLAGS += -I.

1352 $(ASSYMDEP_OBJS:%=pics%): assym.h

1354 # assym.h build rules

1356 assym.h := CFLAGS += -g

1358 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1360 genassym: $(GENASSYM_C)
1361     $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1362     $(CPPFLAGS.native) -o $$@ $(GENASSYM_C)

1364 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1366 assym.h: $(OFFSETS) genassym
1367     $(OFFSETS_CREATE) <$(OFFSETS) >$$@
1368     ./genassym >>$$@

1370 # derived C source and related explicit dependencies
1371 $(LIBCDIR)/port/gen/errlst.c + \
1372 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1373     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1375 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1377 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c

```



```

*****
16334 Sat Sep 28 12:17:13 2013
new/usr/src/uts/common/sys/termios.h
1060 termios missing cfmakeraw
Reviewed by: Cyril Plisko <cyril.plisko@mountall.com>
Reviewed by: Albert Lee <trisk@opensolaris.org>
*****
_____unchanged_portion_omitted_____

94 /*
95  * POSIX termios functions
96  * These functions get mapped into ioctl's.
97  */

99 #ifndef _KERNEL

101 #if defined(__STDC__)

103 extern speed_t cfgetospeed(const struct termios *);
104 extern int cfsetospeed(struct termios *, speed_t);
105 extern speed_t cfgetispeed(const struct termios *);
106 extern int cfsetispeed(struct termios *, speed_t);
107 extern void cfmakeraw(struct termios *);
108 #endif /* ! codereview */
109 extern int tcgetattr(int, struct termios *);
110 extern int tcsetattr(int, int, const struct termios *);
111 extern int tcsetattr(int, int);
112 extern int tcdrain(int);
113 extern int tcflush(int, int);
114 extern int tcflow(int, int);

116 #else

118 extern speed_t cfgetospeed();
119 extern int cfsetospeed();
120 extern speed_t cfgetispeed();
121 extern int cfsetispeed();
122 extern void cfmakeraw();
123 #endif /* ! codereview */
124 extern int tcgetattr();
125 extern int tcsetattr();
126 extern int tcsetattr(int, int);
127 extern int tcdrain();
128 extern int tcflush();
129 extern int tcflow();

131 #endif /* __STDC__ */

133 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)

135 #if defined(__STDC__)
136 extern pid_t tcgetsid(int);
137 #else
138 extern pid_t tcgetsid();
139 #endif /* __STDC__ */

141 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) ... */

143 #endif

145 /* control characters */
146 #define VINTR 0
147 #define VQUIT 1
148 #define VERASE 2
149 #define VKILL 3
150 #define VEOF 4

```

```

151 #define VEOL 5
152 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
153 #define VEOL2 6
154 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
155 #define VMIN 4
156 #define VTIME 5
157 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
158 #define VSWTCH 7
159 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
160 #define VSTART 8
161 #define VSTOP 9
162 #define VSUSP 10
163 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
164 #define VDSUSP 11
165 #define VREPRINT 12
166 #define VDISCARD 13
167 #define VWERASE 14
168 #define VLNEXT 15
169 /* 16 thru 19 reserved for future use */

171 /*
172  * control characters form Xenix termio.h
173  */
174 #define VCEOF NCC /* RESERVED true EOF char (V7 compatability) */
175 #define VCEOL (NCC + 1) /* RESERVED true EOL char */

177 #define CNUL 0
178 #define CDEL 0177

180 /* S5 default control chars */
181 /* CINTR, CERASE and CKILL modified to SunOS traditional values */
182 #define CESC '\\\
183 #define CINTR CTRL('c')
184 #define CQUIT 034 /* FS, cntl | */
185 #define CERASE 0177 /* DEL */
186 #define CKILL CTRL('u')
187 #define CEOT 04
188 #define CEOL 0
189 #define CEOL2 0
190 #define CEOF 04 /* cntl d */
191 #define CSTART 021 /* cntl q */
192 #define CSTOP 023 /* cntl s */
193 #define CSWTCH 032 /* cntl z */
194 #define CNSWTCH 0
195 #define CSUSP CTRL('z')
196 #define CDSUSP CTRL('y')
197 #define CRPRNT CTRL('r')
198 #define CFLUSH CTRL('o')
199 #define CWERASE CTRL('w')
200 #define CLNEXT CTRL('v')
201 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

204 /* input modes */
205 #define IGNBRK 0000001
206 #define BRKINT 0000002
207 #define IGNPAR 0000004
208 #define PARMRK 0000010
209 #define INPCK 0000020
210 #define ISTRIP 0000040
211 #define INLCR 0000100
212 #define IGNCR 0000200
213 #define ICRNL 0000400
214 #if !defined(_POSIX_C_SOURCE) || \
215     (defined(_XOPEN_SOURCE) && !defined(_XPG6)) || \
216     defined(__EXTENSIONS__)

```

```

217 #define IUCLC    0001000
218 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
219 #define IXON     0002000
220 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
221     defined(__EXTENSIONS__)
222 #define IXANY   0004000
223 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
224 #define IXOFF   0010000
225 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
226 #define IMAXBEL 0020000
227 #define DOSMODE 0100000 /* for 386 compatibility */
228 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

230 /* output modes */
231 #define OPOST   0000001
232 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
233     defined(__EXTENSIONS__)
234 #if !defined(_XPG6) || defined(__EXTENSIONS__)
235 #define OLCUC   0000002
236 #endif
237 #define ONLCR   0000004
238 #define OCRNL   0000010
239 #define ONOCR   0000020
240 #define ONLRET  0000040
241 #define OFILL   0000100
242 #define OFDEL   0000200
243 #define NLDLY   0000400
244 #define NL0     0
245 #define NL1     0000400
246 #define CRDLY   0003000
247 #define CR0     0
248 #define CR1     0001000
249 #define CR2     0002000
250 #define CR3     0003000
251 #define TABDLY  0014000
252 #define TAB0    0
253 #define TAB1    0004000
254 #define TAB2    0010000
255 #define TAB3    0014000
256 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
257 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
258 #define XTABS   0014000
259 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
260 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
261     defined(__EXTENSIONS__)
262 #define BSDLY   0020000
263 #define BS0     0
264 #define BS1     0020000
265 #define VTDLY   0040000
266 #define VT0     0
267 #define VT1     0040000
268 #define FFDLY   0100000
269 #define FF0     0
270 #define FF1     0100000
271 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
272 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
273 #define PAGEOUT 0200000
274 #define WRAP    0400000

276 /* control modes */
277 #define CBAUD   0000017
278 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
279 #define CSIZE   0000060
280 #define CS5     0
281 #define CS6     0000020
282 #define CS7     0000040

```

```

283 #define CS8     0000060
284 #define CSTOPB 0000100
285 #define CREAD   0000200
286 #define PARENB  0000400
287 #define PARODD  0001000
288 #define HUPCL   0002000
289 #define CLOCAL  0004000
290 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
291 #define RCVLEN   0010000
292 #define XMTLEN   0020000
293 #define LOBLK    0040000
294 #define XCLUDE   0100000 /* *V7* exclusive use coming from XENIX */
295 #define CRTSXOFF 010000000000
296 #define CRTSCTS  020000000000
297 #define CIBAUD   03600000
298 #define PAREXT   04000000
299 #define CBAUDEXT 0100000000
300 #define CIBAUDEXT 0200000000

302 /*
303  * 4.4BSD hardware flow control flags
304  */
305 #define CRTS_IFLOW 010000000000
306 #define CCTS_OFLOW 020000000000

308 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

310 /* line discipline 0 modes */
311 #define ISIG     0000001
312 #define ICANON   0000002
313 #if !defined(_POSIX_C_SOURCE) || \
314     (defined(_XOPEN_SOURCE) && !defined(_XPG6)) || \
315     defined(__EXTENSIONS__)
316 #define XCASE    0000004
317 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
318 #define ECHO     0000010
319 #define ECHOE    0000020
320 #define ECHOK    0000040
321 #define ECHONL   0000100
322 #define NOFLSH   0000200
323 #define TOSTOP   0000400
324 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
325 #define ECHOCTL  0001000
326 #define ECHOPRT  0002000
327 #define ECHOKE   0004000
328 #define DEFECHO  0010000
329 #define FLUSHO   0020000
330 #define PENDIN   0040000
331 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

333 #define IEXTEN   0100000 /* POSIX flag - enable POSIX extensions */
334 #define TIOC     ('T' << 8)

336 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)

338 #define TIOC     _TIOC

340 #define TCGETA   (_TIOC | 1)
341 #define TCSETA   (_TIOC | 2)
342 #define TCSETAW  (_TIOC | 3)
343 #define TCSETAF  (_TIOC | 4)
344 #define TCSEBRK  (_TIOC | 5)
345 #define TCXONC   (_TIOC | 6)
346 #define TCFLSH   (_TIOC | 7)

348 /* Slots reserved for 386/XENIX compatibility - keyboard control */

```

```

350 #define TIOCKEON      (_TIOC|8)
351 #define TIOCKBOF     (_TIOC|9)
352 #define KBENABLED    (_TIOC|10)

354 #ifndef IOCTYPE
355 #define IOCTYPE 0xff00
356 #endif

358 #define TCDSSET      (_TIOC|32)
359 #define RTS_TOG     (_TIOC|33) /* 386 - "RTS" toggle define 8A1 protocol */

361 #define TIOCGWINSZ  (_TIOC|104)
362 #define TIOCSWINSZ  (_TIOC|103)

364 /*
365  * Softcarrier ioctls
366  */
367 #define TIOCGSOFTCAR (_TIOC|105)
368 #define TIOCSSOFTCAR (_TIOC|106)

371 /* termios ioctls */

373 #define TCGETS      (_TIOC|13)
374 #define TCSETS      (_TIOC|14)
375 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
376 #define TCSANOW     (_TIOC|14) /* same as TCSETS */
377 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
378 #define TCSETSW     (_TIOC|15)
379 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
380 #define TCSADRAIN   (_TIOC|15) /* same as TCSETSW */
381 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
382 #define TCSETSF     (_TIOC|16)

384 /*
385  * NTP PPS ioctls
386  */
387 #define TIOCGPPS    (_TIOC|125)
388 #define TIOCSPPS    (_TIOC|126)
389 #define TIOCGPPSEV  (_TIOC|127)

391 /* Argument filled in by TIOCGPPSEV */
392 struct ppsclockev {
393     struct timeval tv;
394     uint_t serial;
395 };

397 #if defined(_SYSCALL32)
398 struct ppsclockev32 {
399     struct timeval32 tv;
400     uint32_t serial;
401 };
402 #endif /* _SYSCALL32 */

404 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

406 #define TCSAFLUSH    (_TIOC|16) /* same as TCSETSF */

408 /* termios option flags */

410 #define TCIFLUSH     0 /* flush data received but not read */
411 #define TCDFLUSH     1 /* flush data written but not transmitted */
412 #define TCIOFLUSH    2 /* flush both data both input and output queues */

414 #define TCOOFF       0 /* suspend output */

```

```

415 #define TCOON        1 /* restart suspended output */
416 #define TCIOFF       2 /* suspend input */
417 #define TCION        3 /* restart suspended input */

419 /* TIOC ioctls for BSD, ptys, job control and modem control */

421 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
422 #define tIOC         ('t'<<8)
423 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

425 /* Slots for 386/XENIX compatibility */
426 /* BSD includes these ioctls in ttold.h */

428 #ifndef _SYS_TTOLD_H

430 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
431 #define TIOCGETD     (tIOC|0)
432 #define TIOCSETD     (tIOC|1)
433 #define TIOCHPCL     (tIOC|2)
434 #define TIOCGETP     (tIOC|8)
435 #define TIOCSETP     (tIOC|9)
436 #define TIOCSETN     (tIOC|10)
437 #define TIOCEXCL     (tIOC|13)
438 #define TIOCNXCL     (tIOC|14)
439 #define TIOCFLUSH    (tIOC|16)
440 #define TIOCSETC     (tIOC|17)
441 #define TIOCGETC     (tIOC|18)
442 /*
443  * BSD ioctls that are not the same as XENIX are included here.
444  * There are also some relevant ioctls from SUN/BSD sys/ttycom.h
445  * BSD pty ioctls like TIOCPKT are not supported in SVR4.
446  */

448 #define TIOCLBIS     (tIOC|127) /* bis local mode bits */
449 #define TIOCLBIC     (tIOC|126) /* bic local mode bits */
450 #define TIOCLSET     (tIOC|125) /* set entire local mode word */
451 #define TIOCLGET     (tIOC|124) /* get local modes */
452 #define TIOCSBRK     (tIOC|123) /* set break bit */
453 #define TIOCCBRK     (tIOC|122) /* clear break bit */
454 #define TIOCSDTR     (tIOC|121) /* set data terminal ready */
455 #define TIOCCDTR     (tIOC|120) /* clear data terminal ready */
456 #define TIOCSLTC     (tIOC|117) /* set local special chars */
457 #define TIOCGLTC     (tIOC|116) /* get local special chars */
458 #define TIOCOUTQ     (tIOC|115) /* driver output queue size */
459 #define TIOCNOTTY    (tIOC|113) /* void tty association */
460 #define TIOCSCTTY    (tIOC|132) /* get a ctty */
461 #define TIOCSTOP     (tIOC|111) /* stop output, like ^S */
462 #define TIOCSSTART   (tIOC|110) /* start output, like ^Q */
463 #define TIOCSLOOP    (tIOC|109) /* private to Sun; do not use */
464 #define TIOCCLOOP    (tIOC|108) /* private to Sun; do not use */

466 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

468 #endif /* end _SYS_TTOLD_H */

470 /* POSIX job control ioctls */

472 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
473 #define TIOCGPRP     (tIOC|20) /* get pgrp of tty */
474 #define TIOCSGRP     (tIOC|21) /* set pgrp of tty */
475 #define TIOCGSID     (tIOC|22) /* get session id on ctty */

477 /* Miscellaneous */
478 #define TIOCSTI      (tIOC|23) /* simulate terminal input */

480 /* Modem control */

```

```

481 #define TIOCMSET      (tIOC|26)    /* set all modem bits */
482 #define TIOCMBSIS    (tIOC|27)    /* bis modem bits */
483 #define TIOCMBIC     (tIOC|28)    /* bic modem bits */
484 #define TIOCMGET      (tIOC|29)    /* get all modem bits */
485 #define TIOCM_LE      TIOCM_LE     0001    /* line enable */
486 #define TIOCM_DTR     TIOCM_DTR    0002    /* data terminal ready */
487 #define TIOCM_RTS     TIOCM_RTS    0004    /* request to send */
488 #define TIOCM_ST      TIOCM_ST     0010    /* secondary transmit */
489 #define TIOCM_SR      TIOCM_SR     0020    /* secondary receive */
490 #define TIOCM_CTS     TIOCM_CTS    0040    /* clear to send */
491 #define TIOCM_CAR     TIOCM_CAR    0100    /* carrier detect */
492 #define TIOCM_CD      TIOCM_CAR
493 #define TIOCM_RNG     TIOCM_RNG    0200    /* ring */
494 #define TIOCM_RI      TIOCM_RNG
495 #define TIOCM_DSR     TIOCM_DSR    0400    /* data set ready */

497 /* pseudo-tty */

499 #define TIOCREMOTE    (tIOC|30)    /* remote input editing */
500 #define TIOCSIGNAL    (tIOC|31)    /* pty: send signal to slave */

503 /* Some more 386 xenix stuff */

505 #define LDIOC        ('D'<<8)

507 #define LDOPEN        (LDIOC|0)
508 #define LDCLOSE       (LDIOC|1)
509 #define LDCHG         (LDIOC|2)
510 #define LDGETT        (LDIOC|8)
511 #define LDSETT        (LDIOC|9)

513 /* Slots for 386 compatibility */

515 #define LDSMAP        (LDIOC|110)
516 #define LDGMAP        (LDIOC|111)
517 #define LDNMAP        (LDIOC|112)
518 #define LDEMAP        (LDIOC|113)
519 #define LDDMAP        (LDIOC|114)

521 /*
522 * These are retained for 386/XENIX compatibility.
523 */

525 #define DIOC          ('d'<<8)
526 #define DIOCGETP      (DIOC|8)      /* V7 */
527 #define DIOCSETP      (DIOC|9)      /* V7 */

529 /*
530 * Returns a non-zero value if there
531 * are characters in the input queue.
532 *
533 * XXX - somebody is confused here. V7 had no such "ioctl", although XENIX may
534 * have added it; 4BSD had FIONREAD, which returned the number of characters
535 * waiting, and was supposed to work on all descriptors (i.e., every driver
536 * should make a stab at implementing it).
537 */
538 #define FIORDCHK      (('f'<<8)|3)    /* V7 */
539 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

541 /*
542 * Speeds
543 */
544 #define B0            0
545 #define B50           1
546 #define B75           2

```

```

547 #define B110         3
548 #define B134         4
549 #define B150         5
550 #define B200         6
551 #define B300         7
552 #define B600         8
553 #define B1200        9
554 #define B1800       10
555 #define B2400       11
556 #define B4800       12
557 #define B9600       13
558 #define B19200      14
559 #define B38400      15
560 #define B57600      16
561 #define B76800      17
562 #define B115200     18
563 #define B153600     19
564 #define B230400     20
565 #define B307200     21
566 #define B460800     22
567 #define B921600     23

569 #ifndef _SYS_TTOLD_H

571 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
572 /* Windowing structure to support JWINSIZE/TIOCSWINSZ/TIOCGWINSZ */
573 struct winsize {
574     unsigned short ws_row; /* rows, in characters */
575     unsigned short ws_col; /* columns, in character */
576     unsigned short ws_xpixel; /* horizontal size, pixels */
577     unsigned short ws_ypixel; /* vertical size, pixels */
578 };
579 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

581 #endif /* end _SYS_TTOLD_H */

583 #ifdef __cplusplus
584 }
585 #endif

587 #endif /* _SYS_TERMIO_H */

```