

new/usr/src/lib/libc/i386/Makefile.com

1

```
*****
23870 Fri Oct 11 14:55:58 2013
new/usr/src/lib/libc/i386/Makefile.com
libc: only have one select implementation, and move the pollfds onto the heap if
*****

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 CRTOBS=      \
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 \
```

new/usr/src/lib/libc/i386/Makefile.com

2

```
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      makectxt.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      gttty.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 SYSOBSJS= \
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   cftime.o            \
382   clock.o             \
383   closedir.o          \
384   closefrom.o         \
385   confstr.o           \
386   crypt.o             \
387   csetlen.o           \
388   ctime.o             \
389   ctime_r.o           \
390   daemon.o            \
391   deflt.o

```

```

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

```

```

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

```

```

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

```

```

589      valloc.o          \
590      vlfmt.o           \
591      vpfmt.o           \
592      waitpid.o         \
593      walkstack.o       \
594      wdata.o           \
595      xgetwidth.o       \
596      xpg4.o            \
597      xpg6.o            \
599      PORTPRINT_W=      \
600      doprnt_w.o        \
602      PORTPRINT=        \
603      asprintf.o        \
604      doprnt.o          \
605      fprintf.o         \
606      printf.o          \
607      snprintf.o        \
608      sprintf.o         \
609      vfprintf.o        \
610      vprintf.o         \
611      vsnprintf.o       \
612      vsprintf.o        \
613      vwprintf.o        \
614      wprintf.o         \
616      # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
617      PORTPRINT_C89=    \
618      vfprintf_c89.o    \
619      vprintf_c89.o     \
620      vsnprintf_c89.o   \
621      vsprintf_c89.o    \
622      vwprintf_c89.o    \
624      PORTSTDIO_C89=    \
625      vscanf_c89.o      \
626      vwscanf_c89.o     \
628      # portable stdio objects that contain large file interfaces.
629      # Note: fopen64 is a special case, as we build it small.
630      PORTSTDIO64=      \
631      fopen64.o         \
632      fpos64.o          \
634      PORTSTDIO_W=      \
635      doscan_w.o        \
637      PORTSTDIO=        \
638      __extensions.o     \
639      _endopen.o         \
640      _filbuf.o          \
641      _findbuf.o         \
642      _flsbuf.o          \
643      _wrtchk.o          \
644      clearerr.o         \
645      ctermid.o          \
646      ctermid_r.o       \
647      cuserid.o          \
648      data.o             \
649      doscan.o           \
650      fdopen.o           \
651      feof.o             \
652      ferror.o           \
653      fgetc.o            \
654      fgets.o            \

```

```

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

694 PORTI18N=      \
695      getwchar.o   \
696      putwchar.o   \
697      putws.o      \
698      strcasecmp.o \
699      strcasestr.o \
700      strncasecmp.o \
701      strtows.o    \
702      wcsnlen.o    \
703      wcsstr.o     \
704      wcstoimax.o  \
705      wcstol.o     \
706      wcstoul.o    \
707      wcswcs.o     \
708      wmemchr.o    \
709      wmemcmp.o    \
710      wmemcpy.o    \
711      wmemmove.o   \
712      wmemset.o    \
713      wscasecmp.o  \
714      wscat.o      \
715      wschr.o      \
716      wscmp.o      \
717      wscpy.o      \
718      wscspn.o     \
719      wsdup.o      \
720      wslen.o      \

```

```

721      wncasecmp.o   \
722      wncat.o       \
723      wncmp.o       \
724      wncpy.o       \
725      wspbrk.o      \
726      wsprintf.o    \
727      wsrchr.o      \
728      wscanf.o      \
729      wsspn.o       \
730      wstod.o       \
731      wstok.o       \
732      wstol.o       \
733      wstoll.o      \
734      wsxfrm.o      \
735      gettext.o     \
736      gettext_gnu.o \
737      gettext_real.o \
738      gettext_util.o \
739      isdigit.o     \
740      plural_parser.o \
741      wdresolve.o   \
742      _ctype.o      \
743      isascii.o     \
744      toascii.o     \

746 PORTI18N_COND=  \
747      wcstol_longlong.o \
748      wcstoul_longlong.o \

750 PORTLOCALE=      \
751      big5.o        \
752      btowc.o       \
753      collate.o     \
754      collcmp.o     \
755      euc.o         \
756      fnmatch.o     \
757      fgetwc.o      \
758      fgetws.o      \
759      fix_grouping.o \
760      fputwc.o      \
761      fputws.o      \
762      fwide.o       \
763      gb18030.o     \
764      gb2312.o      \
765      gbk.o         \
766      getdate.o     \
767      iswctype.o    \
768      ldpair.o      \
769      lmessages.o   \
770      lnumeric.o    \
771      lmonetary.o   \
772      localeconv.o  \
773      mbftowc.o     \
774      mblen.o       \
775      mbrlen.o      \
776      mbrtowc.o     \
777      mbsinit.o     \
778      mbsnrtowcs.o  \
779      mbsrtowcs.o   \
780      mbstowcs.o    \
781      mbtowc.o      \
782      mskanji.o     \
783      nextwctype.o  \
784      nl_langinfo.o \
785      none.o        \
786      regcomp.o     \

```

```

787      regfree.o          \
788      regerror.o         \
789      regexec.o          \
790      rune.o             \
791      runetype.o         \
792      setlocale.o        \
793      setrunelocale.o    \
794      strcoll.o          \
795      strfmon.o          \
796      strftime.o         \
797      strptime.o         \
798      strxfrm.o          \
799      table.o            \
800      timelocal.o        \
801      tolower.o          \
802      towlower.o         \
803      ungetwc.o          \
804      utf8.o             \
805      wcrtime.o          \
806      wcscoll.o          \
807      wcsftime.o         \
808      wcsnrtombs.o       \
809      wcsrtombs.o        \
810      wcswidth.o         \
811      wcstombs.o         \
812      wcsxfrm.o          \
813      wctob.o            \
814      wctomb.o           \
815      wctrans.o          \
816      wctype.o           \
817      wcwidth.o          \
818      wscoll.o           \
820  AI00BJS=              \
821      aio.o              \
822      aio_alloc.o        \
823      posix_aio.o        \
825  RT0BJS=              \
826      clock_timer.o      \
827      mqueue.o           \
828      pos4obj.o          \
829      sched.o            \
830      sem.o              \
831      shm.o              \
832      sigev_thread.o     \
834  TP00LOBJS=           \
835      thread_pool.o      \
837  THREADSOBJS=         \
838      alloc.o            \
839      assfail.o          \
840      cancel.o           \
841      door_calls.o       \
842      pthr_attr.o        \
843      pthr_barrier.o     \
844      pthr_cond.o        \
845      pthr_mutex.o       \
846      pthr_rwlock.o      \
847      pthread.o          \
848      rwlock.o           \
849      scalls.o           \
850      sema.o             \
851      sigaction.o        \
852      spawn.o            \

```

```

853      synch.o            \
854      tdb_agent.o        \
855      thr.o              \
856      thread_interface.o \
857      tls.o              \
858      tsd.o              \
860  THREADSMACH0BJS=      \
861      machdep.o          \
863  THREADSASMOBJS=       \
864      asm_subr.o         \
866  UNICODE0BJS=         \
867      u8_textprep.o      \
868      uconv.o            \
870  UNWINDMACH0BJS=       \
871      unwind.o           \
873  UNWINDASMOBJS=       \
874      unwind_frame.o    \
876  # objects that implement the transitional large file API
877  PORTSYS64=            \
878      lockf64.o          \
879      stat64.o           \
881  PORTSYS=              \
882      _autofssys.o       \
883      access.o           \
884      acctctl.o          \
885      bsd_signal.o       \
886      chmod.o            \
887      chown.o            \
888      corectl.o          \
889      exacctsyz.o        \
890      execl.o            \
891      execle.o           \
892      execv.o            \
893      fcntl.o            \
894      getpagesizes.o     \
895      getpeerucrd.o      \
896      inst_sync.o        \
897      isetugid.o         \
898      label.o            \
899      link.o             \
900      lockf.o            \
901      lwp.o              \
902      lwp_cond.o         \
903      lwp_rwlock.o       \
904      lwp_sigmask.o      \
905      meminfosys.o       \
906      mkdir.o            \
907      mknod.o            \
908      msgsys.o           \
909      nfssys.o           \
910      open.o             \
911      pgrpsys.o          \
912      posix_sigwait.o    \
913      ppriv.o            \
914      psetsys.o          \
915      rctlsys.o          \
916      readlink.o         \
917      rename.o           \
918      sbrk.o             \

```

```

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

939  PORTREGEX=           \
940      glob.o            \
941      regcmp.o          \
942      regex.o           \
943      wordexp.o         \

945  MOSTOBSJ=            \
946      $(STRETS)         \
947      $(CRTOBSJ)        \
948      $(DYNOBJS)        \
949      $(FPOBJS)         \
950      $(FPASMOBJS)      \
951      $(ATOMICOBJS)     \
952      $(XATTROBJS)      \
953      $(COMOBJS)        \
954      $(DTRACEOBSJ)     \
955      $(GENOBJS)        \
956      $(PORTFP)         \
957      $(PORTGEN)        \
958      $(PORTGEN64)      \
959      $(PORTI18N)       \
960      $(PORTI18N_COND)  \
961      $(PORTLOCALE)     \
962      $(PORTPRINT)      \
963      $(PORTPRINT_C89)  \
964      $(PORTPRINT_W)    \
965      $(PORTREGEX)      \
966      $(PORTSTDIO)      \
967      $(PORTSTDIO64)    \
968      $(PORTSTDIO_C89)  \
969      $(PORTSTDIO_W)    \
970      $(PORTSYS)        \
971      $(PORTSYS64)      \
972      $(AI0OBJS)        \
973      $(RTOBJS)         \
974      $(TPOOL0BJS)      \
975      $(THREADSOBJS)    \
976      $(THREADSMACHOBJS) \
977      $(THREADSASMOBJS) \
978      $(UNICODEOBJS)    \
979      $(UNWINDMACHOBJS) \
980      $(UNWINDASMOBJS)  \
981      $(COMSYSOBJS)     \
982      $(SYSOBJS)        \
983      $(COMSYSOBJS64)   \
984      $(SYSOBJS64)      \

```

```

985      $(VALUES)

987  TRACEOBSJ=           \
988      plockstat.o

990  # NOTE: libc.so.1 must be linked with the minimal crt1.o and crtn.o
991  # modules whose source is provided in the $(SRC)/lib/common directory.
992  # This must be done because otherwise the Sun C compiler would insert
993  # its own versions of these modules and those versions contain code
994  # to call out to C++ initialization functions. Such C++ initialization
995  # functions can call back into libc before thread initialization is
996  # complete and this leads to segmentation violations and other problems.
997  # Since libc contains no C++ code, linking with the minimal crt1.o and
998  # crtn.o modules is safe and avoids the problems described above.
999  OBJECTS= $(CRTI) $(MOSTOBSJ) $(CRTN)
1000  CRTSRCS= ../../common/i386

1002  LDPASS_OFF=          $(POUND_SIGN)

1004  # include common library definitions
1005  include ../../Makefile.lib

1007  # we need to override the default SONAME here because we might
1008  # be building a variant object (still libc.so.1, but different filename)
1009  SONAME = libc.so.1

1011  CFLAGS += $(CCVERBOSE) $(CTF_FLAGS)

1013  # This is necessary to avoid problems with calling _ex_unwind().
1014  # We probably don't want any inlining anyway.
1015  XINLINE = -xinline=
1016  CFLAGS += $(XINLINE)

1018  CERRWARN += -_gcc=-Wno-parentheses
1019  CERRWARN += -_gcc=-Wno-switch
1020  CERRWARN += -_gcc=-Wno-uninitialized
1021  CERRWARN += -_gcc=-Wno-unused-value
1022  CERRWARN += -_gcc=-Wno-unused-label
1023  CERRWARN += -_gcc=-Wno-unused-variable
1024  CERRWARN += -_gcc=-Wno-type-limits
1025  CERRWARN += -_gcc=-Wno-char-subscripts
1026  CERRWARN += -_gcc=-Wno-clobbered
1027  CERRWARN += -_gcc=-Wno-unused-function
1028  CERRWARN += -_gcc=-Wno-address

1030  # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1031  # enables ASSERT() checking in the threads portion of the library.
1032  # This is automatically enabled for DEBUG builds, not for non-debug builds.
1033  THREAD_DEBUG =
1034  $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1036  # Make string literals read-only to save memory.
1037  CFLAGS += $(XSTRCONST)

1039  ALTPICS= $(TRACEOBSJ:%=pics/%)

1041  $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) \
1042      $(EXTPICS) $(LDLIBS)

1044  MAPFILES =          $(LIBCDIR)/port/mapfile-vers

1046  #
1047  # EXT_N_CPPFLAGS and EXT_N_CFLAGS set in enclosing Makefile
1048  #
1049  CFLAGS +=            $(EXT_N_CFLAGS)
1050  CPPFLAGS=            -D_REENTRANT -Di386 $(EXT_N_CPPFLAGS) $(THREAD_DEBUG) \

```

```

1051      -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1052 ASFLAGS=      $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) $(i386_AS_XARCH)

1054 # As a favor to the dtrace syscall provider, libc still calls the
1055 # old syscall traps that have been obsoleted by the *at() interfaces.
1056 # Delete this to compile libc using only the new *at() system call traps
1057 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1059 # Inform the run-time linker about libc specialized initialization
1060 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1061 DYNFLAGS +=      $(RTLDINFO)

1063 # Force libc's internal references to be resolved immediately upon loading
1064 # in order to avoid critical region problems. Since almost all libc symbols
1065 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1066 DYNFLAGS +=      -znw

1068 DYNFLAGS +=      -e __rtboot
1069 DYNFLAGS +=      $(EXTN_DYNFLAGS)

1071 # Inform the kernel about the initial DTrace area (in case
1072 # libc is being used as the interpreter / runtime linker).
1073 DTRACE_DATA =      -zdtrace=dtrace_data
1074 DYNFLAGS +=      $(DTRACE_DATA)

1076 # DTrace needs an executable data segment.
1077 MAPFILE.NED=

1079 BUILD.S=      $(AS) $(ASFLAGS) $< -o $@

1081 # Override this top level flag so the compiler builds in its native
1082 # C99 mode. This has been enabled to support the complex arithmetic
1083 # added to libc.
1084 C99MODE=      $(C99_ENABLE)

1086 # libc method of building an archive
1087 # The "$(GREP) -v ' L '" part is necessary only until
1088 # lorder is fixed to ignore thread-local variables.
1089 BUILD.AR=      $(RM) $@ ; \
1090      $(AR) q $@ '$(LORDER) $(MOSTOBSJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1092 # extra files for the clean target
1093 CLEANFILES=      \
1094      $(LIBCDIR)/port/gen/errlst.c      \
1095      $(LIBCDIR)/port/gen/new_list.c    \
1096      assym.h                            \
1097      genassym                           \
1098      crt/_rtld.s                        \
1099      crt/_rtbootld.s                   \
1100      pics/_rtbootld.o                  \
1101      pics/crti.o                       \
1102      pics/crtn.o                       \
1103      $(ALTPICS)

1105 CLOBBERFILES += $(LIB_PIC)

1107 # list of C source for lint
1108 SRCS=      \
1109      $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c) \
1110      $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)   \
1111      $(COMOBSJS:%.o=$(SRC)/common/util/%.c)     \
1112      $(DTRACEOBJS:%.o=$(SRC)/common/dtrace/%.c) \
1113      $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)       \
1114      $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)      \
1115      $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)    \
1116      $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c) \

```

```

1117      $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c) \
1118      $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c) \
1119      $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)  \
1120      $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)      \
1121      $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)       \
1122      $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)         \
1123      $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)   \
1124      $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c) \
1125      $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1126      $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c) \
1127      $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c) \
1128      $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)      \
1129      $(LIBCBASE)/gen/ecvt.c                    \
1130      $(LIBCBASE)/gen/makectxt.c                 \
1131      $(LIBCBASE)/gen/siginfold.c                 \
1132      $(LIBCBASE)/gen/siglongjmp.c                \
1133      $(LIBCBASE)/gen/strcmp.c                    \
1134      $(LIBCBASE)/gen/sync_instruction_memory.c   \
1135      $(LIBCBASE)/sys/ptrace.c                    \
1136      $(LIBCBASE)/sys/uadmin.c

1138 # conditional assignments
1139 $(DYNLIB) := CRTI = crt.i.o
1140 $(DYNLIB) := CRTN = crtn.o

1142 # Files which need the threads .il inline template
1143 TIL=      \
1144      aio.o \
1145      alloc.o \
1146      assfail.o \
1147      atexit.o \
1148      atfork.o \
1149      cancel.o \
1150      door_calls.o \
1151      err.o \
1152      errno.o \
1153      lwp.o \
1154      ma.o \
1155      machdep.o \
1156      posix_aio.o \
1157      pthr_attr.o \
1158      pthr_barrier.o \
1159      pthr_cond.o \
1160      pthr_mutex.o \
1161      pthr_rwlock.o \
1162      pthread.o \
1163      rand.o \
1164      rwlock.o \
1165      scalls.o \
1166      sched.o \
1167      sema.o \
1168      sigaction.o \
1169      sigev_thread.o \
1170      spawn.o \
1171      stack.o \
1172      synch.o \
1173      tdb_agent.o \
1174      thr.o \
1175      thread_interface.o \
1176      thread_pool.o \
1177      tls.o \
1178      tsd.o \
1179      unwind.o

1181 THREADS_INLINES = $(LIBCBASE)/threads/i386.il
1182 $(TIL:%=pics/%) := CFLAGS += $(THREADS_INLINES)

```

```

1184 # pics/mul64.o := CFLAGS += $(LIBCBASE)/crt/mul64.il
1186 # large-file-aware components that should be built large
1188 $(COMSYSOBS64:%=pics/%) := \
1189     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
1191 $(SYSOBS64:%=pics/%) := \
1192     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
1194 $(PORTGEN64:%=pics/%) := \
1195     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
1197 $(PORTSTDIO64:%=pics/%) := \
1198     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
1200 $(PORTSYS64:%=pics/%) := \
1201     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64
1203 $(PORTSTDIO_W:%=pics/%) := \
1204     CPPFLAGS += -D_WIDE
1206 $(PORTPRINT_W:%=pics/%) := \
1207     CPPFLAGS += -D_WIDE
1209 $(PORTPRINT_C89:%=pics/%) := \
1210     CPPFLAGS += -D_C89_INTMAX32
1212 $(PORTSTDIO_C89:%=pics/%) := \
1213     CPPFLAGS += -D_C89_INTMAX32
1215 $(PORTI18N_COND:%=pics/%) := \
1216     CPPFLAGS += -D_WCS_LONGLONG
1218 .KEEP_STATE:
1220 all: $(LIBS) $(LIB_PIC)
1222 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1223 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1224 lint := LINTFLAGS += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
1226 lint:
1227     @echo $(LINT.c) ...
1228     @$$(LINT.c) $(SRCS) $(LDLIBS)
1230 $(LINTLIB):= SRCS=$(LIBCDIR)/port/lib-lc
1231 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1232 $(LINTLIB):= LINTFLAGS=-nvx
1234 # object files that depend on inline template
1235 $(TIL:%=pics/%) := $(LIBCBASE)/threads/i386.il
1236 # pics/mul64.o: $(LIBCBASE)/crt/mul64.il
1238 # include common libc targets
1239 include $(LIBCDIR)/Makefile.targ
1241 # We need to strip out all CTF and DOF data from the static library
1242 $(LIB_PIC) := DIR = pics
1243 $(LIB_PIC): pics $(PICS)
1244     $(BUILD.AR)
1245     $(MCS) -d -n .SUNW_ctf $$@ > /dev/null 2>&1
1246     $(MCS) -d -n .SUNW_dof $$@ > /dev/null 2>&1
1247     $(AR) -ts $$@ > /dev/null
1248     $(POST_PROCESS_A)

```

```

1250 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.c
1251     $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S $(C_PICFLAGS) \
1252     $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1253     $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $$@
1254     $(RM) $(LIBCBASE)/crt/_rtld.s
1256 # partially built from C source
1257 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1258     $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $$@
1259     $(CTFCONVERT_0)
1261 ASSYMDEP_OBJS= \
1262     _lwp_mutex_unlock.o \
1263     _stack_grow.o \
1264     getcontext.o \
1265     setjmp.o \
1266     tls_get_addr.o \
1267     vforkx.o
1269 $(ASSYMDEP_OBJS:%=pics/%) := CPPFLAGS += -I.
1271 $(ASSYMDEP_OBJS:%=pics/%) : assym.h
1273 # assym.h build rules
1275 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c
1277 genassym: $(GENASSYM_C)
1278     $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1279     -D__EXTENSIONS__ $(CPPFLAGS.native) -o $$@ $(GENASSYM_C)
1281 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in
1283 assym.h: $(OFFSETS) genassym
1284     $(OFFSETS_CREATE) <$(OFFSETS) >$$@
1285     ./genassym >>$$@
1287 # derived C source and related explicit dependencies
1288 $(LIBCDIR)/port/gen/errlst.c + \
1289 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1290     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist
1292 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c
1294 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c

```

```

*****
9187 Fri Oct 11 14:55:58 2013
new/usr/src/lib/libc/port/gen/select.c
select: shortcut fd_sets_count if given 0 fds (such as for timing)
if nfds is low, fastpath to try to maintain throughput
libc: only have one select implementation, and move the pollfds onto the heap if
*****
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 /*
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26
27 /*      Copyright (c) 1988 AT&T */
28 /*      All Rights Reserved */
29
30 #pragma ident "%Z%M% %I% %E% SMI"
31
32 /*
33  * Emulation of select() system call using poll() system call.
34  *
35  * Assumptions:
36  *   polling for input only is most common.
37  *   polling for exceptional conditions is very rare.
38  *
39  * Note that is it not feasible to emulate all error conditions,
40  * in particular conditions that would return EFAULT are far too
41  * difficult to check for in a library routine.
42 */
43
44 #pragma weak _select = select
45
46 #include "lint.h"
47 #include <values.h>
48 #include <pthread.h>
49 #include <errno.h>
50 #include <stdlib.h>
51 #endif /* ! codereview */
52 #include <sys/time.h>
53 #include <sys/types.h>
54 #include <sys/select.h>
55 #include <sys/poll.h>
56 #include <alloca.h>
57 #include "libc.h"
58
59 /*

```

```

58 * STACK_PFD_LIM
59 *
60 * The limit at which pselect allocates pollfd structures in the heap,
61 * rather than on the stack. These limits match the historical behaviour
62 * with the *_large_fdset implementations.
63 *
64 * BULK_ALLOC_LIM
65 *
66 * The limit below which we'll just allocate nfds pollfds, rather than
67 * counting how many we actually need.
68 */
69 #if defined(_LP64)
70 #define STACK_PFD_LIM    FD_SETSIZE
71 #define BULK_ALLOC_LIM   8192
72 #else
73 #define STACK_PFD_LIM    1024
74 #define BULK_ALLOC_LIM   1024
75 #endif
76
77 /*
78  * The previous _large_fdset implementations are, unfortunately, baked into
79  * the ABI.
80 */
81 #pragma weak select_large_fdset = select
82 #pragma weak pselect_large_fdset = pselect
83
84 #define fd_set_size(nfds)    (((nfds) + (NFDBITS - 1)) / NFDBITS)
85
86 static nfds_t
87 fd_sets_count(int limit, fd_set *in, fd_set *out, fd_set *ex)
88 {
89     nfds_t total = 0;
90
91     if (limit <= 0)
92         return (0);
93
94     for (int i = 0; i < fd_set_size(limit); i++) {
95         long v = (in->fds_bits[i] | out->fds_bits[i] | ex->fds_bits[i]);
96
97         while (v != 0) {
98             v &= v - 1;
99             total++;
100         }
101     }
102
103     return (total);
104 }
105
106 #endif /* ! codereview */
107 int
108 pselect(int nfds, fd_set *in0, fd_set *out0, fd_set *ex0,
109         const timespec_t *tsp, const sigset_t *sigmask)
110 {
111     long *in, *out, *ex;
112     ulong_t m; /* bit mask */
113     int j; /* loop counter */
114     ulong_t b; /* bits to test */
115     int n, rv;
116     struct pollfd *pfd;
117     struct pollfd *p;
118     int lastj = -1;
119     nfds_t npfds = 0;
120     boolean_t heap_pfds = B_FALSE;
121 #endif /* ! codereview */
122
123     /* "zero" is read-only, it could go in the text segment */

```

```

124     static fd_set zero = { 0 };
125
126     /*
127      * Check for invalid conditions at outset.
128      * Required for spec1170.
129      * SUSV3: We must behave as a cancellation point even if we fail early.
130      */
131     if (nfds < 0 || nfds > FD_SETSIZE) {
132         pthread_testcancel();
133         errno = EINVAL;
134         return (-1);
135     }
136     p = pfd = (struct pollfd *)alloca(nfds * sizeof (struct pollfd));
137
138     if (tsp != NULL) {
139         /* check timespec validity */
140         if (tsp->tv_nsec < 0 || tsp->tv_nsec >= NANOSEC ||
141             tsp->tv_sec < 0) {
142             pthread_testcancel();
143             errno = EINVAL;
144             return (-1);
145         }
146     }
147
148     /*
149      * If any input args are null, point them at the null array.
150      */
151     if (in0 == NULL)
152         in0 = &zero;
153     if (out0 == NULL)
154         out0 = &zero;
155     if (ex0 == NULL)
156         ex0 = &zero;
157
158     if (nfds <= BULK_ALLOC_LIM) {
159         p = pfd = alloca(nfds * sizeof (struct pollfd));
160     } else {
161         npfds = fd_sets_count(nfds, in0, out0, ex0);
162
163         if (npfds > STACK_PFD_LIM) {
164             p = pfd = malloc(npfds * sizeof (struct pollfd));
165             if (p == NULL)
166                 return (-1);
167             heap_pfds = B_TRUE;
168         } else {
169             p = pfd = alloca(npfds * sizeof (struct pollfd));
170         }
171     }
172 #endif /* ! codereview */
173     /*
174      * For each fd, if any bits are set convert them into
175      * the appropriate pollfd struct.
176      */
177     in = (long *)in0->fds_bits;
178     out = (long *)out0->fds_bits;
179     ex = (long *)ex0->fds_bits;
180     for (n = 0; n < nfds; n += NFDBITS) {
181         b = (ulong_t)(in[n] | out[n] | ex[n]);
182         for (j = 0; m = 1; b != 0; j++, b >>= 1, m <= 1) {
183             if (b & 1) {
184                 p->fd = n + j;
185                 if (p->fd >= nfds)
186                     goto done;
187                 p->events = 0;
188                 if (*in & m)

```

```

189                 p->events |= POLLRDNORM;
190                 if (*out & m)
191                     p->events |= POLLWRNORM;
192                 if (*ex & m)
193                     p->events |= POLLRDBAND;
194                 p++;
195             }
196         }
197         in++;
198         out++;
199         ex++;
200     }
201 done:
202     /*
203      * Now do the poll.
204      */
205     npfds = (int)(p - pfd);
206     n = (int)(p - pfd); /* number of pollfd's */
207     do {
208         rv = _pollsys(pfd, npfds, tsp, sigmask);
209         rv = _pollsys(pfd, (nfds_t)n, tsp, sigmask);
210     } while (rv < 0 && errno == EAGAIN);
211
212     if (rv < 0) /* no need to set bit masks */
213         goto out;
214     return (rv);
215
216     if (rv == 0) {
217         /*
218          * Clear out bit masks, just in case.
219          * On the assumption that usually only
220          * one bit mask is set, use three loops.
221          */
222         if (in0 != &zero) {
223             in = (long *)in0->fds_bits;
224             for (n = 0; n < nfds; n += NFDBITS)
225                 *in++ = 0;
226         }
227         if (out0 != &zero) {
228             out = (long *)out0->fds_bits;
229             for (n = 0; n < nfds; n += NFDBITS)
230                 *out++ = 0;
231         }
232         if (ex0 != &zero) {
233             ex = (long *)ex0->fds_bits;
234             for (n = 0; n < nfds; n += NFDBITS)
235                 *ex++ = 0;
236         }
237         rv = 0;
238         goto out;
239         return (0);
240     }
241
242     /*
243      * Check for EINVAL error case first to avoid changing any bits
244      * if we're going to return an error.
245      */
246     for (p = pfd, n = npfds; n-- > 0; p++) {
247         for (p = pfd, j = n; j-- > 0; p++) {
248             /*
249              * select will return EBADF immediately if any fd's
250              * are bad. poll will complete the poll on the
251              * rest of the fd's and include the error indication
252              * in the returned bits. This is a rare case so we
253              * accept this difference and return the error after
254              * doing more work than select would've done.

```

```

250     */
251     if (p->revents & POLLNVAL) {
252         errno = EBADE;
253         rv = -1;
254         goto out;
119         return (-1);
255     }
256     /*
257     * We would like to make POLLHUP available to select,
258     * checking to see if we have pending data to be read.
259     * BUT until we figure out how not to break Xsun's
260     * dependencies on select's existing features...
261     * This is what we _thought_ would work ... sigh!
262     */
263     /*
264     * if ((p->revents & POLLHUP) &&
265     *     !(p->revents & (POLLRDNORM|POLLRDBAND))) {
266     *     errno = EINTR;
267     *     rv = -1;
268     *     goto out;
132     *     return (-1);
269     * }
270     */
271 }

273 /*
274 * Convert results of poll back into bits
275 * in the argument arrays.
276 *
277 * We assume POLLRDNORM, POLLWRNORM, and POLLRDBAND will only be set
278 * on return from poll if they were set on input, thus we don't
279 * worry about accidentally setting the corresponding bits in the
280 * zero array if the input bit masks were null.
281 *
282 * Must return number of bits set, not number of ready descriptors
283 * (as the man page says, and as poll() does).
284 */
285 rv = 0;
286 for (p = pfd, n = npfds; n-- > 0; p++) {
150     for (p = pfd; n-- > 0; p++) {
287         j = (int)(p->fd / NFDBITS);
288         /* have we moved into another word of the bit mask yet? */
289         if (j != lastj) {
290             /* clear all output bits to start with */
291             in = (long *)&in0->fds_bits[j];
292             out = (long *)&out0->fds_bits[j];
293             ex = (long *)&ex0->fds_bits[j];
294             /*
295              * In case we made "zero" read-only (e.g., with
296              * cc -R), avoid actually storing into it.
297              */
298             if (in0 != &zero)
299                 *in = 0;
300             if (out0 != &zero)
301                 *out = 0;
302             if (ex0 != &zero)
303                 *ex = 0;
304             lastj = j;
305         }
306         if (p->revents) {
307             m = 1L << (p->fd % NFDBITS);
308             if (p->revents & POLLRDNORM) {
309                 *in |= m;
310                 rv++;
311             }
312             if (p->revents & POLLWRNORM) {

```

```

313         *out |= m;
314         rv++;
315     }
316     if (p->revents & POLLRDBAND) {
317         *ex |= m;
318         rv++;
319     }
320     /*
321     * Only set this bit on return if we asked about
322     * input conditions.
323     */
324     if ((p->revents & (POLLHUP|POLLERR)) &&
325         (p->events & POLLRDNORM)) {
326         if ((*in & m) == 0)
327             rv++; /* wasn't already set */
328         *in |= m;
329     }
330     /*
331     * Only set this bit on return if we asked about
332     * output conditions.
333     */
334     if ((p->revents & (POLLHUP|POLLERR)) &&
335         (p->events & POLLWRNORM)) {
336         if ((*out & m) == 0)
337             rv++; /* wasn't already set */
338         *out |= m;
339     }
340     /*
341     * Only set this bit on return if we asked about
342     * output conditions.
343     */
344     if ((p->revents & (POLLHUP|POLLERR)) &&
345         (p->events & POLLRDBAND)) {
346         if ((*ex & m) == 0)
347             rv++; /* wasn't already set */
348         *ex |= m;
349     }
350 }
351 }
352 out:
353     if (heap_pfds)
354         free(pfd);
355 #endif /* ! codereview */
356     return (rv);
357 }

359 int
360 select(int nfds, fd_set *in0, fd_set *out0, fd_set *ex0, struct timeval *tv)
361 {
362     timespec_t ts;
363     timespec_t *tsp;

365     if (tv == NULL)
366         tsp = NULL;
367     else {
368         /* check timeval validity */
369         if (tv->tv_usec < 0 || tv->tv_usec >= MICROSEC) {
370             errno = EINVAL;
371             return (-1);
372         }
373         /*
374          * Convert timeval to timespec.
375          * To preserve compatibility with past behavior,
376          * when select was built upon poll(2), which has a
377          * minimum non-zero timeout of 1 millisecond, force
378          * a minimum non-zero timeout of 500 microseconds.

```

```
379         */
380         ts.tv_sec = tv->tv_sec;
381         ts.tv_nsec = tv->tv_usec * 1000;
382         if (ts.tv_nsec != 0 && ts.tv_nsec < 500000)
383             ts.tv_nsec = 500000;
384         tsp = &ts;
385     }
387     return (pselect(nfds, in0, out0, ex0, tsp, NULL));
388 }
```

```
*****
25679 Fri Oct 11 14:55:59 2013
new/usr/src/lib/libc/sparc/Makefile.com
libc: only have one select implementation, and move the pollfds onto the heap if
*****
```

```
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 CRTOBSJS=     \
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      getcctxt.o         \
145      ladd.o             \
146      lmul.o             \
147      lock.o             \
148      lshiftl.o          \
149      lsign.o            \
150      lsub.o             \
151      makectxt.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      __lgrp_sys.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      fac1.o             \
227      fchdir.o           \
228      fchroot.o          \
229      fdsync.o           \
230      fpathconf.o        \
231      fstatfs.o          \
232      fstatvfs.o         \
233      getcpuid.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      sigprocmask.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 SYSOBJJS=
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      cftime.o         \
414      clock.o          \
415      closedir.o       \
416      closefrom.o     \
417      confstr.o        \
418      crypt.o          \
419      csetlen.o        \
420      ctime.o          \
421      ctime_r.o        \
422      daemon.o         \
423      deflt.o          \
424      directio.o       \
425      dirname.o        \
426      div.o            \
427      drand48.o        \
428      dup.o            \
429      env_data.o       \
430      err.o            \
431      errno.o          \
432      euclen.o         \
433      event_port.o     \
434      execvp.o         \
435      fattach.o        \
436      fdetach.o        \
437      fdopendir.o     \
438      ffs.o            \
439      fls.o            \
440      fmtmsg.o         \
441      ftime.o          \
442      ftok.o           \
443      ftw.o            \
444      gcv.t.o          \
445      getauxv.o        \
446      getcwd.o         \
447      getdate_err.o    \
448      getdtablesize.o  \
449      getenv.o         \
450      getexecname.o    \
451      getgrnam.o       \
452      getgrnam_r.o     \
453      gethostid.o      \
454      gethostname.o    \
455      gethz.o          \
456      getisax.o        \
457      getloadavg.o     \

```

```

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

```

```

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

```

```

589      strsep.o              \
590      strsignal.o           \
591      strspn.o              \
592      strstr.o              \
593      strtod.o              \
594      strtointmax.o         \
595      strtok.o              \
596      strtok_r.o            \
597      strtoumax.o           \
598      swab.o                \
599      swapctl.o             \
600      sysconf.o             \
601      syslog.o              \
602      tcdrain.o             \
603      tcflow.o              \
604      tcflush.o             \
605      tcgetattr.o           \
606      tcgetpgrp.o           \
607      tcgetsid.o            \
608      tcsendbreak.o         \
609      tcsetattr.o           \
610      tcsetpgrp.o           \
611      tell.o                \
612      telldir.o             \
613      tfind.o               \
614      time_data.o           \
615      time_gdata.o          \
616      tls_data.o            \
617      truncate.o            \
618      tsdalloc.o            \
619      tsearch.o             \
620      ttyname.o             \
621      ttyslot.o             \
622      ualarm.o              \
623      ucred.o               \
624      valloc.o              \
625      vlfmt.o               \
626      vpfmt.o               \
627      waitpid.o             \
628      walkstack.o           \
629      wdata.o               \
630      xgetwidth.o           \
631      xpg4.o                \
632      xpg6.o                \
634      PORTPRINT_W=         \
635      doprnt_w.o            \
637      PORTPRINT=            \
638      asprintf.o            \
639      doprnt.o              \
640      fprintf.o             \
641      printf.o              \
642      snprintf.o            \
643      sprintf.o             \
644      vfprintf.o            \
645      vprintf.o             \
646      vsnprintf.o           \
647      vsprintf.o            \
648      vwprintf.o            \
649      wprintf.o             \
651      # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
652      PORTPRINT_C89=        \
653      vfprintf_c89.o        \
654      vprintf_c89.o         \

```

```

655      vsnprintf_c89.o      \
656      vsprintf_c89.o       \
657      vwprintf_c89.o       \

659 PORTSTDIO_C89=          \
660      vscanf_c89.o         \
661      wscanf_c89.o         \

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

669 PORTSTDIO_W=             \
670      doscan_w.o            \

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

```

```

721      tmpfile.o             \
722      tmpnam_r.o            \
723      ungetc.o              \
724      mse.o                 \
725      vscanf.o              \
726      wscanf.o              \
727      wscanf.o              \

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

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

785 PORTLOCALE=              \
786      big5.o                \

```

```

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

```

```

853      wscoll.o

855  AI00BJS=          \
856      aio.o          \
857      aio_alloc.o    \
858      posix_aio.o    \

860  RT0BJS=           \
861      clock_timer.o  \
862      mqueue.o       \
863      pos4obj.o      \
864      sched.o        \
865      sem.o          \
866      shm.o          \
867      sigev_thread.o \

869  TP00LOBJS=        \
870      thread_pool.o \

872  THREADS0BJS=      \
873      alloc.o        \
874      assfail.o      \
875      cancel.o       \
876      door_calls.o   \
877      pthr_attr.o    \
878      pthr_barrier.o  \
879      pthr_cond.o    \
880      pthr_mutex.o   \
881      pthr_rwlock.o  \
882      pthread.o      \
883      rwlock.o       \
884      scalls.o       \
885      sema.o         \
886      sigaction.o    \
887      spawn.o        \
888      synch.o        \
889      tdb_agent.o    \
890      thr.o           \
891      thread_interface.o \
892      tls.o          \
893      tsd.o

895  THREADSMACH0BJS=  \
896      machdep.o

898  THREADSASM0BJS=   \
899      asm_subr.o

901  UNICODE0BJS=      \
902      u8_textprep.o  \
903      uconv.o

905  UNWINDMACH0BJS=   \
906      unwind.o

908  UNWINDASM0BJS=    \
909      unwind_frame.o

911  # objects that implement the transitional large file API
912  PORTSYS64=        \
913      lockf64.o      \
914      stat64.o

916  PORTSYS=          \
917      _autofssys.o   \
918      access.o

```

```

919      acctctl.o          \
920      bsd_signal.o       \
921      chmod.o            \
922      chown.o            \
923      corectl.o          \
924      exacctsys.o        \
925      execl.o             \
926      execle.o           \
927      execv.o            \
928      fcntl.o            \
929      getpagesizes.o     \
930      getpeerucred.o     \
931      inst_sync.o        \
932      issetugid.o        \
933      label.o            \
934      link.o             \
935      lockf.o            \
936      lwp.o              \
937      lwp_cond.o         \
938      lwp_rwlock.o       \
939      lwp_sigmask.o      \
940      meminfosys.o       \
941      mkdir.o            \
942      mknod.o            \
943      msgsys.o           \
944      nfssys.o           \
945      open.o             \
946      pgrpsys.o          \
947      posix_sigwait.o    \
948      ppriv.o            \
949      psetsys.o          \
950      rctlsys.o          \
951      readlink.o         \
952      rename.o           \
953      sbrk.o             \
954      semsys.o           \
955      set_errno.o        \
956      sharefs.o          \
957      shmsys.o           \
958      sidsys.o           \
959      siginterrupt.o     \
960      signal.o           \
961      sigpending.o       \
962      sigstack.o         \
963      stat.o             \
964      symlink.o          \
965      tasksys.o          \
966      time.o             \
967      time_util.o        \
968      ucontext.o         \
969      unlink.o           \
970      ustat.o            \
971      utimesys.o         \
972      zone.o             \
974  PORTREGEX=            \
975      glob.o             \
976      regcmp.o           \
977      regex.o            \
978      wordexp.o          \
980  VALUES= values-Xa.o
982  MOSTOBSJS=            \
983      $(STRETS)          \
984      $(CRTOBSJS)        \

```

```

985      $(DYNOBJS)         \
986      $(FPOBJS)          \
987      $(FPASMOBJS)       \
988      $(ATOMICOBJS)      \
989      $(XATTOBJS)        \
990      $(COMOBJS)          \
991      $(DTRACEOBJS)      \
992      $(GENOBJS)         \
993      $(PRFOBJS)         \
994      $(PORTFP)          \
995      $(PORTGEN)         \
996      $(PORTGEN64)       \
997      $(PORTI18N)        \
998      $(PORTI18N_COND)   \
999      $(PORTLOCALE)      \
1000     $(PORTPRINT)       \
1001     $(PORTPRINT_C89)    \
1002     $(PORTPRINT_W)      \
1003     $(PORTREGEX)        \
1004     $(PORTSTDIO)        \
1005     $(PORTSTDIO64)      \
1006     $(PORTSTDIO_C89)    \
1007     $(PORTSTDIO_W)      \
1008     $(PORTSYS)          \
1009     $(PORTSYS64)        \
1010     $(AI0OBJS)          \
1011     $(RTOBJS)           \
1012     $(TPOOL0BJS)        \
1013     $(THREADSOBJS)      \
1014     $(THREADSMACH0BJS)   \
1015     $(THREADSASMOBJS)    \
1016     $(UNICODE0BJS)      \
1017     $(UNWINDMACH0BJS)    \
1018     $(UNWINDASMOBJS)     \
1019     $(COMSYS0BJS)        \
1020     $(SYS0BJS)           \
1021     $(COMSYS0BJS64)      \
1022     $(SYS0BJS64)        \
1023     $(VALUES)           \
1025  TRACE0BJS=            \
1026      plockstat.o        \
1028  # NOTE: libc.so.1 must be linked with the minimal crt.i.o and crtn.o
1029  # modules whose source is provided in the $(SRC)/lib/common directory.
1030  # This must be done because otherwise the Sun C compiler would insert
1031  # its own versions of these modules and those versions contain code
1032  # to call out to C++ initialization functions. Such C++ initialization
1033  # functions can call back into libc before thread initialization is
1034  # complete and this leads to segmentation violations and other problems.
1035  # Since libc contains no C++ code, linking with the minimal crt.i.o and
1036  # crtn.o modules is safe and avoids the problems described above.
1037  OBJECTS= $(CRTI) $(MOSTOBSJS) $(CRTN)
1038  CRTSRCS= ../../common/sparc
1040  # include common library definitions
1041  include $(SRC)/lib/Makefile.lib
1043  # we need to override the default SONAME here because we might
1044  # be building a variant object (still libc.so.1, but different filename)
1045  SONAME = libc.so.1
1047  CFLAGS += $(CCVERBOSE)
1049  # This is necessary to avoid problems with calling _ex_unwind().
1050  # We probably don't want any inlining anyway.

```

```

1051 CFLAGS += -xinline=

1053 CERRWARN += -_gcc=-Wno-parentheses
1054 CERRWARN += -_gcc=-Wno-switch
1055 CERRWARN += -_gcc=-Wno-uninitialized
1056 CERRWARN += -_gcc=-Wno-unused-value
1057 CERRWARN += -_gcc=-Wno-unused-label
1058 CERRWARN += -_gcc=-Wno-unused-variable
1059 CERRWARN += -_gcc=-Wno-type-limits
1060 CERRWARN += -_gcc=-Wno-char-subscripts
1061 CERRWARN += -_gcc=-Wno-clobbered
1062 CERRWARN += -_gcc=-Wno-unused-function
1063 CERRWARN += -_gcc=-Wno-address

1065 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1066 # enables ASSERT() checking in the threads portion of the library.
1067 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1068 THREAD_DEBUG =
1069 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1071 # Make string literals read-only to save memory.
1072 CFLAGS += $(XSTRCONST)

1074 ALTPICS= $(TRACEOBSJS:=pics/%)

1076 $(DYNLIB) := BUILD.SO = $(LD) -o $$@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1078 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1080 CFLAGS += $(EXTN_CFLAGS)
1081 CPPFLAGS= -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1082           -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1083 ASFLAGS= $(EXTN_ASFLAGS) -K pic -P -D__STDC__ -D_ASM $(CPPFLAGS) $(sparc_

1085 # As a favor to the dtrace syscall provider, libc still calls the
1086 # old syscall traps that have been obsoleted by the *at() interfaces.
1087 # Delete this to compile libc using only the new *at() system call traps
1088 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1090 # Inform the run-time linker about libc specialized initialization
1091 RTLDINFO = -z rtldinfo=tls_rtldinfo
1092 DYNFLAGS += $(RTLDINFO)

1094 # Force libc's internal references to be resolved immediately upon loading
1095 # in order to avoid critical region problems. Since almost all libc symbols
1096 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1097 DYNFLAGS += -znov

1099 DYNFLAGS += -e __rtboot
1100 DYNFLAGS += $(EXTN_DYNFLAGS)

1102 # Inform the kernel about the initial DTrace area (in case
1103 # libc is being used as the interpreter / runtime linker).
1104 DTRACE_DATA = -zdtrace=dtrace_data
1105 DYNFLAGS += $(DTRACE_DATA)

1107 # DTrace needs an executable data segment.
1108 MAPFILE.NED=

1110 BUILD.s= $(AS) $(ASFLAGS) $< -o $$@

1112 # Override this top level flag so the compiler builds in its native
1113 # C99 mode. This has been enabled to support the complex arithmetic
1114 # added to libc.
1115 C99MODE= $(C99_ENABLE)

```

```

1117 # libc method of building an archive
1118 # The "$(GREP) -v ' L '" part is necessary only until
1119 # lorder is fixed to ignore thread-local variables.
1120 BUILD.AR= $(RM) $@ ; \
1121           $(AR) q $@ '$(LORDER) $(MOSTOBSJS:=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1123 # extra files for the clean target
1124 CLEANFILES= \
1125             $(LIBCDIR)/port/gen/errlst.c \
1126             $(LIBCDIR)/port/gen/new_list.c \
1127             assym.h \
1128             genassym \
1129             $(LIBCBASE)/crt/_rtld.s \
1130             $(LIBCBASE)/crt/_rtbootld.s \
1131             pics/_rtbootld.o \
1132             pics/crti.o \
1133             pics/crtn.o \
1134             $(ALTPICS)

1136 CLOBBERFILES += $(LIB_PIC)

1138 # list of C source for lint
1139 SRCS= \
1140       $(ATOMICOBJS:=.o=$(SRC)/common/atomic/%.c) \
1141       $(XATTSOBSJS:=.o=$(SRC)/common/xattr/%.c) \
1142       $(COMOBSJS:=.o=$(SRC)/common/util/%.c) \
1143       $(DTRACEOBSJS:=.o=$(SRC)/common/dtrace/%.c) \
1144       $(PORTFP:=.o=$(LIBCDIR)/port/fp/%.c) \
1145       $(PORTGEN:=.o=$(LIBCDIR)/port/gen/%.c) \
1146       $(PORTI18N:=.o=$(LIBCDIR)/port/i18n/%.c) \
1147       $(PORTLOCALE:=.o=$(LIBCDIR)/port/locale/%.c) \
1148       $(PORTPRINT:=.o=$(LIBCDIR)/port/print/%.c) \
1149       $(PORTREGEX:=.o=$(LIBCDIR)/port/regex/%.c) \
1150       $(PORTSTDIO:=.o=$(LIBCDIR)/port/stdio/%.c) \
1151       $(PORTSYS:=.o=$(LIBCDIR)/port/sys/%.c) \
1152       $(AIOOBSJS:=.o=$(LIBCDIR)/port/aio/%.c) \
1153       $(RTOBSJS:=.o=$(LIBCDIR)/port/rt/%.c) \
1154       $(TPOOLOBSJS:=.o=$(LIBCDIR)/port/tpool/%.c) \
1155       $(THREADSOBSJS:=.o=$(LIBCDIR)/port/threads/%.c) \
1156       $(THREADSMACHOBSJS:=.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1157       $(UNICODEOBSJS:=.o=$(SRC)/common/unicode/%.c) \
1158       $(UNWINDMACHOBSJS:=.o=$(LIBCDIR)/port/unwind/%.c) \
1159       $(FPOBSJS:=.o=$(LIBCDIR)/$(MACH)/fp/%.c) \
1160       $(LIBCBASE)/crt/_ftou.c \
1161       $(LIBCBASE)/gen/_xregs_clrptr.c \
1162       $(LIBCBASE)/gen/byteorder.c \
1163       $(LIBCBASE)/gen/ecvt.c \
1164       $(LIBCBASE)/gen/getctxt.c \
1165       $(LIBCBASE)/gen/lmul.c \
1166       $(LIBCBASE)/gen/makectxt.c \
1167       $(LIBCBASE)/gen/siginfolst.c \
1168       $(LIBCBASE)/gen/siglongjmp.c \
1169       $(LIBCBASE)/gen/swapctxt.c \
1170       $(LIBCBASE)/sys/ptrace.c \
1171       $(LIBCBASE)/sys/uadmin.c

1173 # conditional assignments
1174 $(DYNLIB) := CRTI = crt.i.o
1175 $(DYNLIB) := CRTN = crtn.o

1177 # Files which need the threads .il inline template
1178 IIL= \
1179     aio.o \
1180     alloc.o \
1181     assfail.o \
1182     atexit.o

```

```

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

1218 $(TIL:%=pics/%) := CFLAGS += $(LIBCBASE)/threads/sparc.il

1220 # special kludge for inlines with 'cas':
1221 pics/rwlock.o pics/synch.o pics/lwp.o pics/door_calls.o := \
1222     sparc_CFLAGS += -_gcc=-Wa,-xarch=v8plus

1224 # Files in port/fp subdirectory that need base.il inline template
1225 IL= \
1226     __flt_decim.o      \
1227     decimal_bin.o      \

1229 $(IL:%=pics/%) := CFLAGS += $(LIBCBASE)/fp/base.il

1231 # Files in fp subdirectory which need __quad.il inline template
1232 QIL= \
1233     _Q_add.o           \
1234     _Q_cmp.o           \
1235     _Q_cmpe.o          \
1236     _Q_div.o           \
1237     _Q_dtoq.o          \
1238     _Q_fcc.o           \
1239     _Q_mul.o           \
1240     _Q_qtod.o          \
1241     _Q_qtoi.o          \
1242     _Q_qtos.o          \
1243     _Q_qtou.o          \
1244     _Q_sqrt.o          \
1245     _Q_stoq.o          \
1246     _Q_sub.o           \

1248 $(QIL:%=pics/%) := CFLAGS += $(LIBCDIR)/$(MACH)/fp/__quad.il

```

```

1249 pics/_Q.o := sparc_COPTFLAG = -x04 -dalign
1250 pics/__quad.o := sparc_COPTFLAG = -x04 -dalign

1252 # large-file-aware components that should be built large

1254 $(COMSYSOBS64:%=pics/%) := \
1255     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1257 $(SYSOBS64:%=pics/%) := \
1258     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1260 $(PORTGEN64:%=pics/%) := \
1261     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1263 $(PORTSTDIO64:%=pics/%) := \
1264     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1266 $(PORTSYS64:%=pics/%) := \
1267     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1269 $(PORTSTDIO_W:%=pics/%) := \
1270     CPPFLAGS += -D_WIDE

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

1275 # printf/scanf functions to support c89-sized intmax_t variables
1276 $(PORTPRINT_C89:%=pics/%) := \
1277     CPPFLAGS += -D_C89_INTMAX32

1279 $(PORTSTDIO_C89:%=pics/%) := \
1280     CPPFLAGS += -D_C89_INTMAX32

1282 $(PORTI18N_COND:%=pics/%) := \
1283     CPPFLAGS += -D_WCS_LONGLONG

1285 # Files which need extra optimization
1286 pics/getenv.o := sparc_COPTFLAG = -x04

1288 .KEEP_STATE:

1290 all: $(LIBS) $(LIB_PIC)

1292 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1293 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1294 lint := LINTFLAGS += -mn

1296 lint:
1297     @echo $(LINT.c) ... $(LDLIBS)
1298     @$$(LINT.c) $(SRCS) $(LDLIBS)

1300 $(LINTLIB): SRCS=$(LIBCDIR)/port/l1ib-lc
1301 $(LINTLIB): CPPFLAGS += -D_MSE_INT_H
1302 $(LINTLIB): LINTFLAGS=-nvx

1304 # object files that depend on inline template
1305 $(TIL:%=pics/%) := $(LIBCBASE)/threads/sparc.il
1306 $(IL:%=pics/%) := $(LIBCBASE)/fp/base.il
1307 $(QIL:%=pics/%) := $(LIBCDIR)/$(MACH)/fp/__quad.il

1309 # include common libc targets
1310 include $(LIBCDIR)/Makefile.targ

1312 # We need to strip out all CTF and DOF data from the static library
1313 $(LIB_PIC) := DIR = pics
1314 $(LIB_PIC): pics $$({PICS})

```

```
1315      $(BUILD.AR)
1316      $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1317      $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1318      $(AR) -ts $@ > /dev/null
1319      $(POST_PROCESS_A)

1321 # special cases
1322 $(STRETS:%=pics/%): $(LIBCBASE)/crt/stret.s
1323      $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=) $(LIBCBASE)/crt/stret.s -o $@
1324      $(POST_PROCESS_0)

1326 $(LIBCBASE)/crt/_rtbootld.s:      $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.
1327      $(CC) $(CPPFLAGS) $(CTF_FLAGS) -o -S -K pic \
1328      $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1329      $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
1330      $(RM) $(LIBCBASE)/crt/_rtld.s

1332 # partially built from C source
1333 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1334      $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
1335      $(CTFCONVERT_0)

1337 ASSYMDEP_OBJS=      \
1338      _lwp_mutex_unlock.o      \
1339      _stack_grow.o      \
1340      asm_subr.o      \
1341      setjmp.o      \
1342      smt_pause.o      \
1343      tls_get_addr.o      \
1344      unwind_frame.o      \
1345      vforkx.o

1347 $(ASSYMDEP_OBJS:%=pics/%):      :=      CPPFLAGS += -I.

1349 $(ASSYMDEP_OBJS:%=pics/%):      assym.h

1351 # assym.h build rules

1353 assym.h := CFLAGS += -g

1355 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1357 genassym: $(GENASSYM_C)
1358      $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1359      $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1361 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1363 assym.h: $(OFFSETS) genassym
1364      $(OFFSETS_CREATE) <$(OFFSETS) >$@
1365      ./genassym >>$@

1367 # derived C source and related explicit dependencies
1368 $(LIBCDIR)/port/gen/errlst.c + \
1369 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1370      cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1372 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1374 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

new/usr/src/uts/common/conf/param.c

1

```
*****
21328 Fri Oct 11 14:56:00 2013
new/usr/src/uts/common/conf/param.c
3772 consider raising default descriptor soft limit
Reviewed by: Piotr Jasiukajtis <estibi@me.com>
Reviewed by: Saso Kiselkov <skiselkov.ml@gmail.com>
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Joshua M. Clulow <josh@sysmgr.org>
*****
unchanged_portion_omitted_
401 int nexectype = sizeof (execsw) / sizeof (execsw[0]); /* # of exec types */
402 kmutex_t execsw_lock; /* Used for allocation of execsw entries */

404 /*
405  * symbols added to make changing proc.max-file-descriptor
406  * symbols added to make changing max-file-descriptors
407  * simple via /etc/system
408  */
408 #define RLIM_FD_CUR 0x10000
408 #define RLIM_FD_CUR 0x100
409 #define RLIM_FD_MAX 0x10000

411 uint_t rlim_fd_cur = RLIM_FD_CUR;
412 uint_t rlim_fd_max = RLIM_FD_MAX;

414 /*
415  * (Default resource limits were formerly declared here, but are now provided by
416  * the more general resource controls framework.)
417  */

419 /*
420  * STREAMS tunables
421  */
422 int nstrpush = 9; /* maximum # of modules/drivers on a stream */
423 ssize_t strctlsz = 1024; /* maximum size of user-generated M_PROTO */
424 ssize_t strmsgsz = 0x10000; /* maximum size of user-generated M_DATA */
425 /* for 'strmsgsz', zero means unlimited */

426 /*
427  * Filesystem tunables
428  */
429 int rstchown = 1; /* POSIX_CHOWN_RESTRICTED is enabled */
430 int ngroups_max = NGROUPS_MAX_DEFAULT;

432 /*
433  * generic scheduling stuff
434  */
435 * Configurable parameters for RT and TS are in the respective
436 * scheduling class modules.
437 */

439 pri_t maxclsypri = MAXCLSYSPRI;
440 pri_t minclsypri = MINCLSYSPRI;
441 char sys_name[] = "SYS";

443 extern pri_t sys_init(id_t, int, classfuncs_t **);
444 extern classfuncs_t sys_classfuncs;

446 sclass_t sclass[] = {
447     { "SYS", sys_init, &sys_classfuncs, STATIC_SCHED, 0 },
448     { "", NULL, NULL, NULL, 0 },
449     { "", NULL, NULL, NULL, 0 },
450     { "", NULL, NULL, NULL, 0 },
451     { "", NULL, NULL, NULL, 0 },
452     { "", NULL, NULL, NULL, 0 },
453     { "", NULL, NULL, NULL, 0 },
454     { "", NULL, NULL, NULL, 0 },

```

new/usr/src/uts/common/conf/param.c

2

```
455     { "", NULL, NULL, NULL, 0 },
456     { "", NULL, NULL, NULL, 0 },
457 };
unchanged_portion_omitted_

```

new/usr/src/uts/common/sys/select.h

1

```
*****
4774 Fri Oct 11 14:56:00 2013
new/usr/src/uts/common/sys/select.h
libc: only have one select implementation, and move the pollfds onto the heap if
*****
_____unchanged_portion_omitted_____
73 #endif /* _SIGSET_T */

75 #endif /* #if !defined(__XOPEN_OR_POSIX) || defined(__XPG6) ... */

77 /*
78  * Select uses bit masks of file descriptors in longs.
79  * These macros manipulate such bit fields.
80  * FD_SETSIZE may be defined by the user, but the default here
81  * should be >= RLIM_FD_MAX.
81  * should be >= NOFILE (param.h).
82  */
83 #ifndef FD_SETSIZE
84 #ifdef _LP64
84 #define FD_SETSIZE      65536
86 #else
87 #define FD_SETSIZE      1024
88 #endif /* _LP64 */
89 #elif FD_SETSIZE > 1024 && !defined(_LP64)
90 #ifdef __PRAGMA_REDEFINE_EXTNAME
91 #pragma redefine_extname      select      select_large_fdset
92 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG6) || defined(__EXTENSIONS__)
93 #pragma redefine_extname      pselect     pselect_large_fdset
94 #endif
95 #else /* __PRAGMA_REDEFINE_EXTNAME */
96 #define select      select_large_fdset
97 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG6) || defined(__EXTENSIONS__)
98 #define pselect     pselect_large_fdset
99 #endif
100 #endif /* __PRAGMA_REDEFINE_EXTNAME */
85 #endif /* FD_SETSIZE */

87 #if !defined(__XPG4_2) || defined(__EXTENSIONS__)
88 typedef long      fd_mask;
89 #endif
90 typedef long      fds_mask;

92 /*
93  * The value of _NBBY needs to be consistant with the value
94  * of NBBY in <sys/param.h>.
95  */
96 #define _NBBY 8
97 #if !defined(__XPG4_2) || defined(__EXTENSIONS__)
98 #ifndef NBBY
98 #define NBBY _NBBY
99 #define NBBY _NBBY
100 #endif
101 #endif /* !defined(__XPG4_2) || defined(__EXTENSIONS__) */

103 #if !defined(__XPG4_2) || defined(__EXTENSIONS__)
104 #define NFDBITS      (sizeof (fd_mask) * NBBY) /* bits per mask */
105 #endif
106 #define FD_NFDBITS      (sizeof (fds_mask) * _NBBY) /* bits per mask */

108 #define __howmany(x, y)      (((x)+((y)-1))/(y))
109 #if !defined(__XPG4_2) || defined(__EXTENSIONS__)
110 #ifndef howmany
111 #define howmany(x, y)      (((x)+((y)-1))/(y))
112 #endif
113 #endif /* !defined(__XPG4_2) || defined(__EXTENSIONS__) */

115 #if !defined(__XPG4_2) || defined(__EXTENSIONS__)
```

new/usr/src/uts/common/sys/select.h

2

```
116 typedef struct fd_set {
117 #else
118 typedef struct __fd_set {
119 #endif
120         long      fds_bits[__howmany(FD_SETSIZE, FD_NFDBITS)];
121 } fd_set;
_____unchanged_portion_omitted_____
```