```
**********************************************************
    2098 Fri Feb 12 11:52:28 2016
new/usr/src/cmd/awk/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License, Version 1.0 only
   6 # (the "License").  You may not use this file except in compliance
   7 # with the License.
   8 #
   9 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  10 # or http://www.opensolaris.org/os/licensing.
  11 # See the License for the specific language governing permissions
  12 # and limitations under the License.
  13 #
  14 # When distributing Covered Code, include this CDDL HEADER in each
  15 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  16 # If applicable, add the following below this CDDL HEADER, with the
  17 # fields enclosed by brackets "[]" replaced with your own identifying
  18 # information: Portions Copyright [yyyy] [name of copyright owner]
  19 #
  20 # CDDL HEADER END
  21 #
  22 #
  23 #
  24 # Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
  25 # Use is subject to license terms.
  26 #

  28 # NOTE: awk is oawk.

  30 PROG= nawk

  32 OBJ1= b.o lib.o main.o parse.o proctab.o run.o tran.o
  33 OBJ2= awk.g.o awk.lx.o
  34 OBJS= $(OBJ2) $(OBJ1)
  35 SRCS= $(OBJ1:%.o=%.c)

  37 include ../Makefile.cmd

  39 CERRWARN += -_gcc=-Wno-implicit-function-declaration
  40 CERRWARN += -_gcc=-Wno-unused-label
  41 CERRWARN += -_gcc=-Wno-parentheses
  42 CERRWARN += -_gcc=-Wno-unused-variable
  43 CERRWARN += -_gcc=-Wno-uninitialized

  45 #
  46 # Message catalog
  47 #
  48 POFILES= $(OBJS:%.o=%.po)
  49 POFILE= awk.po
  50 XGETFLAGS += -a -x awk.xcl
  51 #

  53 CPPFLAGS += -D_FILE_OFFSET_BITS=64
  54 YFLAGS += -d
  55 LDLIBS += -lm
  56 LINTFLAGS += -u
  57 CLEANFILES= maketab proctab.c awk.g.c awk.lx.c y.tab.h

  59 .KEEP_STATE:

  61 all: $(PROG)
```



```
  63 $(PROG): $(OBJS)
  64         $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
  65         $(POST_PROCESS)

  67 #
  68 # message catalog
  69 #

  71 $(POFILE):  y.tab.h $(POFILES)
  72         $(RM) $@
  73         cat $(POFILES) > $@
  74 #

  76 proctab.c: maketab
  77         rm -f $@; ./maketab > $@

  79 maketab: maketab.c
  80         $(NATIVECC) $(NATIVE_CFLAGS) maketab.c -o $@ $(LDLIBS)
  80         $(NATIVECC) -O maketab.c -o $@ $(LDLIBS)

  82 install: all $(ROOTPROG) $(ROOTLINK)

  84 clean:
  85         $(RM) $(OBJS) $(CLEANFILES)

  87 lint: awk.g.c lint_SRCS

  89 awk.g.c + y.tab.h: awk.g.y

  91 awk.g.o: awk.g.c

  93 awk.lx.c: awk.lx.l

  95 proctab.o: proctab.c
  96         $(COMPILE.c) proctab.c
  97         $(POST_PROCESS_O)

  99 include ../Makefile.targ
```

```
**********************************************************
    2693 Fri Feb 12 11:52:29 2016
new/usr/src/cmd/sgs/elfdump/Makefile.targ
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
#

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

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

%64.o:          ../common/%.c
                $(COMPILE.c) -D_ELF64 -o $@ $<
                $(POST_PROCESS_O)

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

all:            $(PROG)

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

check_struct_layout: gen_struct_layout gen_layout_obj.o
        sed -e '/^#include <struct_layout.h>/q' \
          ../common/struct_layout_$(ARCH).c | sed -e '$$d' \
          > struct_layout_$(ARCH).tmp
        ./gen_struct_layout gen_layout_obj.o $(ARCH) \
          >> struct_layout_$(ARCH).tmp
        @diff -u struct_layout_$(ARCH).tmp \
          ../common/struct_layout_$(ARCH).c ||\
        { echo "Error: struct_layout_$(ARCH).c needs update!" ; exit 1; }
        touch $@
```



```
# We need CTF data in this object.
gen_layout_obj.o := CFLAGS += $(CTF_FLAGS)
gen_layout_obj.o := CFLAGS64 += $(CTF_FLAGS)
gen_layout_obj.o := POST_PROCESS_O = $(CTFCONVERT_O)

gen_struct_layout:      ../common/gen_struct_layout.c
                $(NATIVECC) $(NATIVE_CFLAGS) -o $@ \
                $(NATIVECC) $(NATIVECC_CFLAGS) -o $@ \
                        ../common/gen_struct_layout.c -lctf

clean:
                $(RM) $(OBJS) $(CLEANFILES) gen_struct_layout \
                  check_struct_layout gen_layout_obj.o \
                  struct_layout_$(ARCH).tmp

delete:
                $(RM) $(PROG)

package \
install:        all $(VAR_SGSBINPROG) $(VAR_SGSCCSLINK)

.PARALLEL:      $(LINTOUT32) $(LINTOUT64)

lint:           $(LINTOUT32) $(LINTOUT64) $(SGSLINTOUT)

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

# Derived source and header files (messaging).

catalog:        $(BLTMESG)

chkmsg:         $(SRCS)
                sh $(CHKMSG) $(CHKMSGFLAGS) $(SRCS)

$(BLTDEFS) + \
$(BLTDATA) + \
$(BLTMESG):     $(SGSMSGALL)
                $(SGSMSG) $(SGSMSGFLAGS) $(SGSMSGALL)
```

```
**********************************************************
    1872 Fri Feb 12 11:52:29 2016
new/usr/src/cmd/sgs/tools/Makefile.com
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#
# Makefile to support tools used for linker development:
#
#  o    sgsmsg creates message headers/arrays/catalogs (a native tool).
#
# Note, these tools are not part of the product.
#
# cmd/sgs/tools/Makefile.com

include         $(SRC)/cmd/Makefile.cmd

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

SGSPROTO=       ../../proto/$(MACH)

COMOBJS=

NATOBJS=        piglatin.o

OBJECTS=        $(COMOBJS)  $(NATOBJS)
NATIVECC_CFLAGS = -O

AVLOBJ=         avl.o
TOOL_OBJS=      sgsmsg.o string_table.o findprime.o assfail.o
SGSMSG_OBJS=    $(TOOL_OBJS) $(AVLOBJ)
SGSMSG_SRCS=    $(TOOL_OBJS:%.o=../common/%.c) \
                $(AVLOBJ:%.o=$(VAR_AVLDIR)/%.c)

$(SGSMSG_OBJS) := NATIVE_CFLAGS += -I../../include $(VAR_TOOLS_CPPFLAGS)
$(SGSMSG_OBJS) := NATIVECC_CFLAGS += -I../../include $(VAR_TOOLS_CPPFLAGS)

PROGS=          $(COMOBJS:%.o=%)
NATIVE=         $(NATOBJS:%.o=%) sgsmsg
SRCS=           $(COMOBJS:%.o=../common/%.c)  $(NATOBJS:%.o=../common/%.c)
```



```
CPPFLAGS +=     $(VAR_TOOLS_CPPFLAGS)
LDFLAGS +=      $(CC_USE_PROTO)
CLEANFILES +=   $(LINTOUT) $(SGSMSG_OBJS)
LINTFLAGS=      -ax

ROOTDIR=        $(ROOT)/opt/SUNWonld
ROOTPROGS=      $(PROGS:%=$(ROOTDIR)/bin/%)

FILEMODE=       0755
```

```
**********************************************************
    1922 Fri Feb 12 11:52:30 2016
new/usr/src/cmd/sgs/tools/Makefile.targ
6648 illumos build should be explicit about C standards
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License, Version 1.0 only
# (the "License").  You may not use this file except in compliance
# with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# ident "%Z%%M% %I%     %E% SMI"
#
# Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/sgs/tools/Makefile.targ


$(ROOTDIR)/bin/%:       %
                $(INS.file)

$(ROOTDIR)/lib/%:       %
                $(INS.file)

sgsmsg:         $(SGSMSG_OBJS)
                $(NATIVECC) $(DEBUG) $(NATIVE_CFLAGS) \
                $(NATIVECC) $(DEBUG) $(NATIVECC_CFLAGS) \
                        -o $@ $(SGSMSG_OBJS)

sgsmsg.o:       ../common/sgsmsg.c
                $(NATIVECC) $(DEBUG) $(NATIVE_CFLAGS) -c ../common/sgsmsg.c
                $(NATIVECC) $(DEBUG) $(NATIVECC_CFLAGS) -c ../common/sgsmsg.c

string_table.o: ../common/string_table.c
                $(NATIVECC) $(DEBUG) $(NATIVE_CFLAGS) -c \
                $(NATIVECC) $(DEBUG) $(NATIVECC_CFLAGS) -c \
                    ../common/string_table.c

avl.o:          $(VAR_AVLDIR)/avl.c
                $(NATIVECC) $(DEBUG) $(NATIVE_CFLAGS) -c \
                $(NATIVECC) $(DEBUG) $(NATIVECC_CFLAGS) -c \
                    $(VAR_AVLDIR)/avl.c

piglatin:       ../common/piglatin.c
                $(NATIVECC) $(NATIVE_CFLAGS) -o $@ ../common/piglatin.c
                $(NATIVECC) $(NATIVECC_CFLAGS) -o $@ ../common/piglatin.c

%:              %.o
                $(LINK.c) -o $@ $< $(LDLIBS)
```



```
                $(POST_PROCESS)

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

catalog:        sgsmsg

clobber:        clean
                $(RM) $(PROGS) $(NATIVE) $(CLOBBERFILES)

clean:
                $(RM) $(OBJECTS) $(CLEANFILES)

lint:
                @ echo "piglatin.c:"                    >> $(LINTOUT)
                $(LINT.c) ../common/piglatin.c -lc      >> $(LINTOUT) 2>&1
                @ echo "sgsmsg"                         >> $(LINTOUT)
                $(LINT.c)  $(SGSMSG_SRCS) -lc           >> $(LINTOUT) 2>&1
```

```
**********************************************************
   29941 Fri Feb 12 11:52:30 2016
new/usr/src/cmd/sgs/tools/common/sgsmsg.c
6645 sgsmsg relies on undefined behaviour of loop-scoped variables
**********************************************************
______unchanged_portion_omitted_

 661 static int
 662 file()
 663 {
 664         char    buffer[LINE_MAX], * token;
 665         uint_t  bufsize;
 666         char    *token_buffer;
 667         int     escape = 0;
 668         int     len = 0;
 669 #endif /* ! codereview */

 671         if ((token_buffer = malloc(LINE_MAX)) == 0) {
 672                 (void) fprintf(stderr, Errmsg_nmem, strerror(errno));
 673                 return (1);
 674         }
 675         bufsize = LINE_MAX;

 677         line = 1;

 679         while ((token = fgets(buffer, LINE_MAX, fddesc)) != NULL) {
 680                 char    defn[PATH_MAX], * _defn, * str;
 668                 int     len;

 682                 switch (*token) {
 683                 case '#':
 684                 case '$':
 685                         if (escape) {
 686                                 (void) fprintf(stderr, Errmsg_malt, fldesc,
 687                                     line);
 688                                 return (1);
 689                         }

 691                         /*
 692                          * If a msgid has been output a msgstr must follow
 693                          * before we digest the new token.  A msgid is only set
 694                          * if fdmsgs is in use.
 695                          */
 696                         if (msgid) {
 697                                 msgid = 0;
 698                                 if (fprintf(fdmsgs, "msgstr\t\"\"\n") < 0) {
 699                                         (void) fprintf(stderr, Errmsg_wrte,
 700                                             fdmsgs, strerror(errno));
 701                                         return (1);
 702                                 }
 703                         }

 705                         /*
 706                          * Pass lines directly through to the output message
 707                          * file.
 708                          */
 709                         if (fdmsgs && (prtmsgs == 1)) {
 710                                 char    comment;

 712                                 if (cflag == 0)
 713                                         comment = '#';
 714                                 else
 715                                         comment = '$';

 717                                 if (fprintf(fdmsgs, "%c%s", comment,
 718                                     ++token) < 0) {
```



```
 719                                         (void) fprintf(stderr, Errmsg_wrte,
 720                                             fdmsgs, strerror(errno));
 721                                         return (1);
 722                                 }
 723                         }
 724                         break;

 726                 case '@':
 727                         if (escape) {
 728                                 (void) fprintf(stderr, Errmsg_malt, fldesc,
 729                                     line);
 730                                 return (1);
 731                         }

 733                         /*
 734                          * If a msgid has been output a msgstr must follow
 735                          * before we digest the new token.
 736                          */
 737                         if (msgid) {
 738                                 msgid = 0;
 739                                 if (fprintf(fdmsgs, "msgstr\t\"\"\n") < 0) {
 740                                         (void) fprintf(stderr, Errmsg_wrte,
 741                                             fdmsgs, strerror(errno));
 742                                         return (1);
 743                                 }
 744                         }

 746                         /*
 747                          * Determine whether we have one or more tokens.
 748                          */
 749                         token++;
 750                         while (isspace(*token))         /* rid any whitespace */
 751                                 token++;
 752                         _defn = token;                  /* definition start */
 753                         while (!(isspace(*token)))
 754                                 token++;
 755                         *token++ = 0;

 757                         while (isspace(*token))         /* rid any whitespace */
 758                                 token++;

 760                         /*
 761                          * Determine whether the single token is one of the
 762                          * reserved message output delimiters otherwise
 763                          * translate it as a message identifier.
 764                          */
 765                         if (*token == 0) {
 766                                 if (strcmp(_defn, start) == 0)
 767                                         prtmsgs = 1;
 768                                 else if (strcmp(_defn, end) == 0)
 769                                         prtmsgs = -1;
 770                                 else if (getmesgid(_defn) == 1)
 771                                         return (1);
 772                                 break;
 773                         }

 775                         /*
 776                          * Multiple tokens are translated by taking the first
 777                          * token as the message definition, and the rest of the
 778                          * line as the message itself.  A message line ending
 779                          * with an escape ('\') is expected to be continued on
 780                          * the next line.
 781                          */
 782                         if (prtmsgs != -1)
 783                                 prtmsgs = 1;
 784                         if (fdmsgs && (prtmsgs == 1)) {
```

```
785                             /*
786                              * For catgets(3c) make sure a message
787                              * identifier has been established (this is
788                              * normally a domain for gettext(3i), but for
789                              * sgsmsg use this could be argued as being
790                              * redundent).  Also make sure that the message
791                              * definitions haven't exceeeded the maximum
792                              * value allowed by gencat(1) before generating
793                              * any message file entries.
794                              */
795                             if (cflag == 1) {
796                                     if (setid == 0) {
797                                             (void) fprintf(stderr, "file "
798                                                 "%s: no message identifier "
799                                                 "has been established\n",
800                                                 fldesc);
801                                             return (1);
802                                     }
803                                     if (ptr > NL_MSGMAX) {
804                                             (void) fprintf(stderr, "file "
805                                                 "%s: message definition "
806                                                 "(%d) exceeds allowable "
807                                                 "limit (NL_MSGMAX)\n",
808                                                 fldesc, ptr);
809                                             return (1);
810                                     }
811                             }

813                             /*
814                              * For catgets(3c) write the definition and the
815                              * message string to the message file.  For
816                              * gettext(3i) write the message string as a
817                              * mesgid - indicate a mesgid has been output
818                              * so that a msgstr can follow.
819                              */
820                             if (cflag == 1) {
821                                     if (fprintf(fdmsgs, "%d\t%s", ptr,
822                                         token) < 0) {
823                                             (void) fprintf(stderr,
824                                                 Errmsg_wrte, flmsgs,
825                                                 strerror(errno));
826                                             return (1);
827                                     }
828                             } else {
829                                     if (fprintf(fdmsgs, "msgid\t\"") < 0) {
830                                             (void) fprintf(stderr,
831                                                 Errmsg_wrte, flmsgs,
832                                                 strerror(errno));
833                                             return (1);
834                                     }
835                                     msgid = 1;
836                             }
837                     }

839                     /*
840                      * The message itself is a quoted string as this makes
841                      * embedding spaces at the start (or the end) of the
842                      * string very easy.
843                      */
844                     if (*token != '"') {
845                             (void) fprintf(stderr, Errmsg_malt, fldesc,
846                                 line);
847                             return (1);
848                     }

850                     (void) strcpy(defn, _defn);
```



```
852                     /*
853                      * Write the tag to the lint definitions.
854                      */
855                     if (fdlint) {
856                             if (fprintf(fdlint, "\n#define\t%s\t",
857                                 _defn) < 0) {
858                                     (void) fprintf(stderr, Errmsg_wrte,
859                                         fllint, strerror(errno));
860                                     return (1);
861                             }
862                     }

864                     len = 0;

866                     /*
867                      * Write each character of the message string to the
868                      * data array.  Translate any escaped characters - use
869                      * the same specially recognized characters as defined
870                      * by gencat(1).
871                      */
872 message:
873                     if (*token == '"') {
874                             if (fdlint &&
875                                 (fprintf(fdlint, "%c", *token) < 0)) {
876                                     (void) fprintf(stderr, Errmsg_wrte,
877                                         fllint, strerror(errno));
878                                     return (1);
879                             }
880                             token++;
881                     }
882                     while (*token) {
883                             char    _token;

885                             if ((*token == '\\') && (escape == 0)) {
886                                     escape = 1;
887                                     if (fdlint && (*(token + 1) != '\n') &&
888                                         fprintf(fdlint, "%c", *token) < 0) {
889                                             (void) fprintf(stderr,
890                                                 Errmsg_wrte, fllint,
891                                                 strerror(errno));
892                                             return (1);
893                                     }
894                                     token++;
895                                     continue;
896                             }
897                             if (escape) {
898                                     if (*token == 'n')
899                                             _token = '\n';
900                                     else if (*token == 't')
901                                             _token = '\t';
902                                     else if (*token == 'v')
903                                             _token = '\v';
904                                     else if (*token == 'b')
905                                             _token = '\b';
906                                     else if (*token == 'f')
907                                             _token = '\f';
908                                     else if (*token == '\\')
909                                             _token = '\\';
910                                     else if (*token == '"')
911                                             _token = '"';
912                                     else if (*token == '\n')
913                                             break;
914                                     else
915                                             _token = *token;
```

```
917                                     if (fdmsgs && (prtmsgs == 1) &&
918                                         (fprintf(fdmsgs, "\\") < 0)) {
919                                             (void) fprintf(stderr,
920                                                 Errmsg_wrte, flmsgs,
921                                                 strerror(errno));
922                                             return (1);
923                                     }
924                             } else {
925                                     /*
926                                      * If this is the trailing quote then
927                                      * thats the last of the message string.
928                                      * Eat up any remaining white space and
929                                      * unless an escape character is found
930                                      * terminate the data string with a 0.
931                                      */
932                                     /* BEGIN CSTYLED */
933                                     if (*token == '"') {
934                                         if (fdlint && (fprintf(fdlint,
935                                             "%c", *token) < 0)) {
936                                             (void) fprintf(stderr,
937                                                 Errmsg_wrte, fllint,
938                                                 strerror(errno));
939                                             return (1);
940                                         }

942                                         if (fdmsgs && (prtmsgs == 1) &&
943                                             (fprintf(fdmsgs, "%c",
944                                             *token) < 0)) {
945                                             (void) fprintf(stderr,
946                                                 Errmsg_wrte, flmsgs,
947                                                 strerror(errno));
948                                             return (1);
949                                         }

951                                         while (*++token) {
952                                             if (*token == '\n')
953                                                     break;
954                                         }
955                                         _token = '\0';
956                                     } else
957                                         _token = *token;
958                                     /* END CSTYLED */
959                             }

961                             if (fdmsgs && (prtmsgs == 1) &&
962                                 (fprintf(fdmsgs, "%c", *token) < 0)) {
963                                     (void) fprintf(stderr, Errmsg_wrte,
964                                         flmsgs, strerror(errno));
965                                     return (1);
966                             }

968                             if (fdlint && fprintf(fdlint,
969                                 "%c", *token) < 0) {
970                                     (void) fprintf(stderr, Errmsg_wrte,
971                                         fllint, strerror(errno));
972                                     return (1);
973                             }

975                             if (len >= bufsize) {
976                                     bufsize += LINE_MAX;
977                                     if ((token_buffer = realloc(
978                                         token_buffer, bufsize)) == 0) {
979                                             (void) fprintf(stderr,
980                                                 Errmsg_nmem,
981                                                 strerror(errno));
982                                             return (1);
```



```
983                                     }
984                             }
985                             token_buffer[len] = _token;
986                             ptr++, token++, len++;
987                             escape = 0;

989                             if (_token == '\0')
990                                     break;
991                     }

993                     /*
994                      * After the complete message string has been processed
995                      * (including its continuation beyond one line), create
996                      * a string size definition.
997                      */
998                     if (escape == 0) {
999                             const char *form = "#define\t%s_SIZE\t%d\n";

1001                            token_buffer[len] = '\0';

1003                            message_append(defn, token_buffer);

1005                            if (fdlint && fprintf(fdlint, form, defn,
1006                                (len - 1)) < 0) {
1007                                    (void) fprintf(stderr, Errmsg_wrte,
1008                                        fllint, strerror(errno));
1009                                    return (1);
1010                            }
1011                    }
1012                    break;

1014            default:
1015                    /*
1016                     * Empty lines are passed through to the message file.
1017                     */
1018                    while (isspace(*token))
1019                            token++;

1021                    if (*token == 0) {
1022                            if (msgid || (fdmsgs && (prtmsgs == 1))) {
1023                                    /*
1024                                     * If a msgid has been output a msgstr
1025                                     * must follow before we digest the new
1026                                     * token.
1027                                     */
1028                                    if (msgid) {
1029                                            msgid = 0;
1030                                            str = "msgstr\t\"\"\n\n";
1031                                    } else
1032                                            str = "\n";

1034                                    if (fprintf(fdmsgs, str) < 0) {
1035                                            (void) fprintf(stderr,
1036                                                Errmsg_wrte, flmsgs,
1037                                                strerror(errno));
1038                                            return (1);
1039                                    }
1040                            }
1041                            break;
1042                    }

1044                    /*
1045                     * If an escape is in effect then any tokens are taken
1046                     * to be message continuations.
1047                     */
1048                    if (escape) {
```

```
1049                                    escape = 0;
1050                                    goto message;
1051                            }

1053                            (void) fprintf(stderr, "file %s: line %d: invalid "
1054                                "input does not start with #, $ or @\n", fldesc,
1055                                line);
1056                            return (1);
1057                    }
1058                    line++;
1059            }

1061            free(token_buffer);

1063            return (0);
1064 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2917 Fri Feb 12 11:52:31 2016
new/usr/src/cmd/zic/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
```

```
   1 #
   2 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
   3 # Use is subject to license terms.
   4 #

   6 NATIVE=native
   7 PROG= zic
   8 KSHPROG= tzselect
   9 NPROG= $(NATIVE)/$(PROG)

  11 # "etcetera" must be listed before "backward" because it sets up links that
  12 # "backward" needs.
  13 TZFILES= africa antarctica asia australasia etcetera europe \
  14          factory northamerica pacificnew solar87 solar88 solar89 \
  15          southamerica backward
  16 # Don't build the systemv timezones, but do install the source file.
  17 SRCFILES = $(TZFILES) systemv README
  18 TABFILES = continent.tab country.tab zone_sun.tab

  20 OBJS= zic.o scheck.o ialloc.o
  21 SRCS= $(OBJS:%.o=%.c)
  22 NOBJS= $(OBJS:%=$(NATIVE)/%)

  24 CLOBBERFILES= $(NOBJS) $(KSHPROG) $(NPROG) $(NATIVE)/zic.c $(NATIVE)/tzfile.h

  26 include ../Makefile.cmd

  28 NCPPFLAGS=      -I$(NATIVE)
  29 ROOTTZD=        $(ROOTSHLIB)/zoneinfo
  30 ROOTTZSRCD=     $(ROOTSHLIB)/zoneinfo/src
  31 ROOTTZTABD=     $(ROOTSHLIB)/zoneinfo/tab
  32 DIRS=  $(ROOTTZSRCD) $(ROOTTZTABD)

  34 CERRWARN +=     -_gcc=-Wno-unused-variable
  35 CERRWARN +=     -_gcc=-Wno-uninitialized

  37 ROOTLIBTZSRCFILES=      $(SRCFILES:%=$(ROOTTZSRCD)/%)
  38 ROOTLIBTZTABFILES=      $(TABFILES:%=$(ROOTTZTABD)/%)
  39 ROOTKSHPROG=    $(ROOTBIN)/$(KSHPROG)

  41 $(ROOTLIBTZSRCFILES):=  FILEMODE=       0644
  42 $(ROOTLIBTZTABFILES):=  FILEMODE=       0644

  44 #
  45 # Message catalog
  46 #
  47 POFILE2=tzselect.po
  48 POFILE3=zoneinfo.po
  49 POFILES3=$(TABFILES)
  50 TEXT_DOMAIN2=SUNW_OST_ZONEINFO
  51 MSGDOMAIN2=$(MSGROOT)/$(TEXT_DOMAIN2)

  53 # NB: $(POFILE2) is under SCM.
  54 CLOBBERFILES += $(POFILE) $(POFILE3)

  56 .KEEP_STATE:

  58 all: $(PROG) $(NPROG) $(KSHPROG)

  60 $(PROG): $(OBJS)
  61         $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
```



```
  62         $(POST_PROCESS)

  64 install: all $(DIRS) $(ROOTTZSRCD) $(ROOTTZTABD) $(ROOTUSRSBINPROG) \
  65                 $(ROOTLIBTZSRCFILES) $(ROOTLIBTZTABFILES) $(ROOTKSHPROG)
  66         $(NPROG) -d $(ROOTTZD) $(TZFILES)

  68 $(NPROG): $(NATIVE) $(NOBJS)
  69         $(NATIVECC) $(NATIVE_CFLAGS) -o $@ $(NOBJS)
  69         $(NATIVECC) -o $@ $(NOBJS)

  71 #
  72 # XXX
  73 # There is a .ksh rule in Makefile.targ, but we need our custom POFILE rules.
  74 #
  75 $(KSHPROG):     $(KSHPROG).ksh
  76         $(RM) $@
  77         $(CAT) $(KSHPROG).ksh > $@
  78         $(CHMOD) +x $@

  80 $(ROOTTZSRCD)/%: %
  81         $(INS.file)

  83 $(ROOTTZTABD)/%: %
  84         $(INS.file)

  86 $(DIRS):
  87         $(INS.dir)

  89 $(NATIVE):
  90         -@mkdir -p $@

  92 clean:
  93         $(RM) $(OBJS)

  95 lint: lint_SRCS

  97 $(NOBJS): $$(@F:.o=.c)
  98         $(NATIVECC) -O $(NATIVE_CFLAGS) $(NCPPFLAGS) -c -o $@ $(@F:.o=.c)
  98         $(NATIVECC) -O $(NCPPFLAGS) -c -o $@ $(@F:.o=.c)

 100 $(NATIVE)/zic.o: $(NATIVE)/tzfile.h

 102 #$(NATIVE)/scheck.o: scheck.c
 103 #       $(NATIVECC) -c -o $@ scheck.c

 105 #$(NATIVE)/ialloc.o: ialloc.c
 106 #       $(NATIVECC) -c -o $@ ialloc.c

 102 $(NATIVE)/tzfile.h: $(ROOT)/usr/include/tzfile.h
 103         $(RM) $@; cp $(ROOT)/usr/include/tzfile.h $@

 105 #
 106 # From ../Makefile.targ
 107 #
 108 clobber: clean
 109         -$(RM) $(PROG) $(CLOBBERFILES)

 111 lint_PROG:
 112         $(LINT.c) $(PROG).c $(LDLIBS)

 114 lint_SRCS:
 115         $(LINT.c) $(SRCS) $(LDLIBS)

 117 #
 118 # For message catalogue files
 119 #
```

```
120 _msg: $(MSGDOMAIN) $(POFILE) $(POFILE2) $(MSGDOMAIN2) $(POFILE3)
121         $(RM) $(MSGDOMAIN)/$(POFILE)
122         $(CP) $(POFILE) $(MSGDOMAIN)
123         $(RM) $(MSGDOMAIN)/$(POFILE2)
124         $(CP) $(POFILE2) $(MSGDOMAIN)
125         $(RM) $(MSGDOMAIN2)/$(POFILE3)
126         $(CP) $(POFILE3) $(MSGDOMAIN2)/$(POFILE3)

128 $(MSGDOMAIN) $(DCMSGDOMAIN) $(MSGDOMAIN2):
129         $(INS.dir)

131 $(POFILE3):     $(POFILES3)
132         $(RM) $@
133         @/bin/sh ./do_pofile.sh > $@
134
```

****
2554 Fri Feb 12 11:52:31 2016
new/usr/src/grub/grub-0.97/Makefile.solaris.defs
6648 illumos build should be explicit about C standards
****

```
#
# Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
include $(SRC)/Makefile.master

.KEEP_STATE:

VERSION         = 0.95

# $(POUND_SIGN) is just a #.  Used to turn a feature off.
# The $(POUND_SIGN)POUND_SIGN:sh construct is just so make doesn't have to
# spawn a shell and redefine POUND_SIGN if it is already defined (e.g.
# $(SRC)/Makefile.master was already included).
$(POUND_SIGN)POUND_SIGN:sh = echo '\043'


#
# Optional features (set to $(POUND_SIGN) to turn off):
#
# OPTION_UTILS:
#       Build the userland utilities.
#       NOTE: these utilities will be installed in
#             /platform/$(PLATFORM)/boot/grub, not /boot/grub.
#
# OPTION_DOCS:
#       Build and install some GRUB documentations, e.g. man and info files.
#       NOTE: these docs will be installed in
#             /platform/$(PLATFORM)/boot/grub, not /boot/grub.
#
# OPTION_FS:
#       Build additional filesystem support into GRUB. This will increase
#       the sizes of stage2 and pxegrub and may cause malfunction.
#
# STAGE2_NETBOOT:
#       Include netboot support in stage2 (on by default).
#       This only affects stage2, not pxegrub (diskless support).
#
# STAGE2_NETBOOT        = $(POUND_SIGN)
OPTION_UTILS    = $(POUND_SIGN)
OPTION_DOCS     = $(POUND_SIGN)
OPTION_FS       = $(POUND_SIGN)


BASE_CFLAGS     = -B$(GCC_ROOT)/bin/ -g $(CPPFLAGS) $(OPTFLAGS) -std=gnu89
BASE_CFLAGS     = -B$(GCC_ROOT)/bin/ -g $(CPPFLAGS) $(OPTFLAGS)
BASE_CCASFLAGS  = -B$(GCC_ROOT)/bin/ -g $(CPPFLAGS) $(OPTFLAGS)
BASE_LDFLAGS    =

CC              = $(GCC_ROOT)/bin/gcc
CFLAGS          = $(BASE_CFLAGS)
CCDEPMODE       = depmode=none

CCAS            = $(CC)
CCASFLAGS       = $(BASE_CCASFLAGS)

CPP             = $(CC) -E

# $(ENVCPPFLAGS4), if defined, should be something like "-Y I,somepath".
# We want to get rid of the "-Y I," part but make's pattern replacement
# macro reference doesn't allow spaces in it, e.g. $(ENVCPPFLAGS4:-Y I,%=%).
```



```
# So we'll get rid of the "-Y" now and then the "I," later.  By the way, this
# should also work in case $(ENVCPPFLAGS4) is "-YI,somepath" (no space).
t.ENVCPPFLAGS4  = $(ENVCPPFLAGS4:-Y%=%)

CPPFLAGS        = $(DEFS) $(INCLUDES) $(WARNFLAGS) \
                  $(ENVCPPFLAGS1:-I%=-isystem %) \
                  $(ENVCPPFLAGS2:-I%=-isystem %) \
                  $(ENVCPPFLAGS3:-I%=-isystem %) \
                  $(t.ENVCPPFLAGS4:I,%=-nostdinc -isystem %)

CCLD            = $(GNU_ROOT)/bin/gld
LDFLAGS         = $(BASE_LDFLAGS)
LINKFLAGS       = -g
LINK            = $(CCLD) $(LINKFLAGS) $(LDFLAGS)

DEFS            = -DHAVE_CONFIG_H -DSOLARIS_NETBOOT
OPTFLAGS        = -falign-jumps=1 -falign-loops=1 -falign-functions=1
#WARNFLAGS      = -Wall -Wmissing-prototypes -Wunused -Wshadow \
#                 -Wpointer-arith -Wundef

OBJCOPY         = $(GNU_ROOT)/bin/gobjcopy

AR              = /usr/ccs/bin/ar
RANLIB          = :

MKDIR           = mkdir -p
```

```
**********************************************************
    7317 Fri Feb 12 11:52:32 2016
new/usr/src/grub/grub-0.97/netboot/io.h
6648 illumos build should be explicit about C standards
**********************************************************
```

```
   1 #include <sys/ccompile.h>

   3 #endif /* ! codereview */
   4 #ifndef IO_H
   5 #define IO_H


   7 /* Amount of relocation etherboot is experiencing */
   8 extern unsigned long virt_offset;

  10 /* Don't require identity mapped physical memory,
  11  * osloader.c is the only valid user at the moment.
  12  */
  13 unsigned long virt_to_phys(volatile const void *virt_addr);
  14 void *phys_to_virt(unsigned long phys_addr);

  16 /* virt_to_bus converts an addresss inside of etherboot [_start, _end]
  17  * into a memory access cards can use.
  18  */
  19 #define virt_to_bus virt_to_phys


  22 /* bus_to_virt reverses virt_to_bus, the address must be output
  23  * from virt_to_bus to be valid.  This function does not work on
  24  * all bus addresses.
  25  */
  26 #define bus_to_virt phys_to_virt

  28 /* ioremap converts a random 32bit bus address into something
  29  * etherboot can access.
  30  */
  31 static inline void *ioremap(unsigned long bus_addr, unsigned long length __unuse
  32 {
  33         return bus_to_virt(bus_addr);
  34 }
_____unchanged_portion_omitted_

  42 /*
  43  * This file contains the definitions for the x86 IO instructions
  44  * inb/inw/inl/outb/outw/outl and the "string versions" of the same
  45  * (insb/insw/insl/outsb/outsw/outsl). You can also use "pausing"
  46  * versions of the single-IO instructions (inb_p/inw_p/..).
  47  *
  48  * This file is not meant to be obfuscating: it's just complicated
  49  * to (a) handle it all in a way that makes gcc able to optimize it
  50  * as well as possible and (b) trying to avoid writing the same thing
  51  * over and over again with slight variations and possibly making a
  52  * mistake somewhere.
  53  */

  55 /*
  56  * Thanks to James van Artsdalen for a better timing-fix than
  57  * the two short jumps: using outb's to a nonexistent port seems
  58  * to guarantee better timings even on fast machines.
  59  *
  60  * On the other hand, I'd like to be sure of a non-existent port:
  61  * I feel a bit unsafe about using 0x80 (should be safe, though)
  62  *
  63  *              Linus
  64  */
```



```
  66 #ifdef  SLOW_IO_BY_JUMPING
  67 #define __SLOW_DOWN_IO __asm__ __volatile__("jmp 1f\n1:\tjmp 1f\n1:")
  68 #else
  69 #define __SLOW_DOWN_IO __asm__ __volatile__("outb %al,$0x80")
  70 #endif

  72 #ifdef  REALLY_SLOW_IO
  73 #define SLOW_DOWN_IO { __SLOW_DOWN_IO; __SLOW_DOWN_IO; __SLOW_DOWN_IO; __SLOW_DO
  74 #else
  75 #define SLOW_DOWN_IO __SLOW_DOWN_IO
  76 #endif

  78 /*
  79  * readX/writeX() are used to access memory mapped devices. On some
  80  * architectures the memory mapped IO stuff needs to be accessed
  81  * differently. On the x86 architecture, we just read/write the
  82  * memory location directly.
  83  */
  84 #define readb(addr) (*(volatile unsigned char *) (addr))
  85 #define readw(addr) (*(volatile unsigned short *) (addr))
  86 #define readl(addr) (*(volatile unsigned int *) (addr))

  88 #define writeb(b,addr) ((*(volatile unsigned char *) (addr)) = (b))
  89 #define writew(b,addr) ((*(volatile unsigned short *) (addr)) = (b))
  90 #define writel(b,addr) ((*(volatile unsigned int *) (addr)) = (b))

  92 #define memcpy_fromio(a,b,c)    memcpy((a),(void *)(b),(c))
  93 #define memcpy_toio(a,b,c)      memcpy((void *)(a),(b),(c))

  95 /*
  96  * Force strict CPU ordering.
  97  * And yes, this is required on UP too when we're talking
  98  * to devices.
  99  *
 100  * For now, "wmb()" doesn't actually do anything, as all
 101  * Intel CPU's follow what Intel calls a *Processor Order*,
 102  * in which all writes are seen in the program order even
 103  * outside the CPU.
 104  *
 105  * I expect future Intel CPU's to have a weaker ordering,
 106  * but I'd also expect them to finally get their act together
 107  * and add some real memory barriers if so.
 108  *
 109  * Some non intel clones support out of order store. wmb() ceases to be a
 110  * nop for these.
 111  */

 113 #define mb()    __asm__ __volatile__ ("lock; addl $0,0(%%esp)": : :"memory")
 114 #define rmb()   mb()
 115 #define wmb()   mb();


 118 /*
 119  * Talk about misusing macros..
 120  */

 122 #define __OUT1(s,x) \
 123 extern void __out##s(unsigned x value, unsigned short port); \
 124 extern __GNU_INLINE \
 125 void __out##s(unsigned x value, unsigned short port) {
 119 extern inline void __out##s(unsigned x value, unsigned short port) {

 127 #define __OUT2(s,s1,s2) \
 128 __asm__ __volatile__ ("out" #s " %" s1 "0,%" s2 "1"

 130 #define __OUT(s,s1,x) \
```

```
__OUT1(s,x) __OUT2(s,s1,"w") : : "a" (value), "d" (port)); } \
__OUT1(s##c,x) __OUT2(s,s1,"") : : "a" (value), "id" (port)); } \
__OUT1(s##_p,x) __OUT2(s,s1,"w") : : "a" (value), "d" (port)); SLOW_DOWN_IO; } \
__OUT1(s##c_p,x) __OUT2(s,s1,"") : : "a" (value), "id" (port)); SLOW_DOWN_IO; }

#define __IN1(s,x) \
extern unsigned x __in##s(unsigned short port); \
extern __GNU_INLINE \
unsigned x __in##s(unsigned short port) { unsigned x _v;
extern inline unsigned x __in##s(unsigned short port) { unsigned x _v;

#define __IN2(s,s1,s2) \
__asm__ __volatile__ ("in" #s " %" s2 "1,%" s1 "0"

#define __IN(s,s1,x,i...) \
__IN1(s,x) __IN2(s,s1,"w") : "=a" (_v) : "d" (port) ,##i ); return _v; } \
__IN1(s##c,x) __IN2(s,s1,"") : "=a" (_v) : "id" (port) ,##i ); return _v; } \
__IN1(s##_p,x) __IN2(s,s1,"w") : "=a" (_v) : "d" (port) ,##i ); SLOW_DOWN_IO; re
__IN1(s##c_p,x) __IN2(s,s1,"") : "=a" (_v) : "id" (port) ,##i ); SLOW_DOWN_IO; r

#define __INS(s) \
extern void ins##s(unsigned short port, void * addr, unsigned long count); \
extern __GNU_INLINE \
void ins##s(unsigned short port, void * addr, unsigned long count)      \
extern inline void ins##s(unsigned short port, void * addr, unsigned long count)
{ __asm__ __volatile__ ("cld ; rep ; ins" #s \
: "=D" (addr), "=c" (count) : "d" (port),"0" (addr),"1" (count)); }

#define __OUTS(s) \
extern void outs##s(unsigned short port, const void * addr, unsigned long  count
extern __GNU_INLINE \
void outs##s(unsigned short port, const void * addr, unsigned long count) \
extern inline void outs##s(unsigned short port, const void * addr, unsigned long
{ __asm__ __volatile__ ("cld ; rep ; outs" #s \
: "=S" (addr), "=c" (count) : "d" (port),"0" (addr),"1" (count)); }

__IN(b,"", char)
__IN(w,"",short)
__IN(l,"", long)

__OUT(b,"b",char)
__OUT(w,"w",short)
__OUT(l,,int)

__INS(b)
__INS(w)
__INS(l)

__OUTS(b)
__OUTS(w)
__OUTS(l)

/*
 * Note that due to the way __builtin_constant_p() works, you
 *  - can't use it inside a inline function (it will never be true)
 *  - you don't have to worry about side effects within the __builtin..
 */
#define outb(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outbc((val),(port)) : \
        __outb((val),(port)))

#define inb(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inbc(port) : \
        __inb(port))
```



```
#define outb_p(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outbc_p((val),(port)) : \
        __outb_p((val),(port)))

#define inb_p(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inbc_p(port) : \
        __inb_p(port))

#define outw(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outwc((val),(port)) : \
        __outw((val),(port)))

#define inw(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inwc(port) : \
        __inw(port))

#define outw_p(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outwc_p((val),(port)) : \
        __outw_p((val),(port)))

#define inw_p(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inwc_p(port) : \
        __inw_p(port))

#define outl(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outlc((val),(port)) : \
        __outl((val),(port)))

#define inl(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inlc(port) : \
        __inl(port))

#define outl_p(val,port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __outlc_p((val),(port)) : \
        __outl_p((val),(port)))

#define inl_p(port) \
((__builtin_constant_p((port)) && (port) < 256) ? \
        __inlc_p(port) : \
        __inl_p(port))

#endif /* ETHERBOOT_IO_H */
```

```
**********************************************************
   22871 Fri Feb 12 11:52:32 2016
new/usr/src/lib/libc/amd64/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
```

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

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

# objects are grouped by source directory

# local objects
STRETS=

CRTOBJS=                        \
        cerror.o

DYNOBJS=

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

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



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

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

ATOMICOBJS=                     \
        atomic.o

CHACHAOBJS=                     \
        chacha.o

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

GENOBJS=                        \
        _getsp.o                \
        abs.o                   \
        alloca.o                \
        arc4random.o            \
        arc4random_uniform.o    \
        attrat.o                \
        byteorder.o             \
        cuexit.o                \
        ecvt.o                  \
        errlst.o                \
        amd64_data.o            \
        ldivide.o               \
        lock.o                  \
        makectxt.o              \
        memccpy.o               \
        memchr.o                \
        memcmp.o                \
        memcpy.o                \
        memset.o                \
```

```
128         new_list.o              \
129         proc64_id.o             \
130         proc64_support.o        \
131         setjmp.o                \
132         siginfolst.o            \
133         siglongjmp.o            \
134         strcmp.o                \
135         strcpy.o                \
136         strlen.o                \
137         strncmp.o               \
138         strncpy.o               \
139         strnlen.o               \
140         sync_instruction_memory.o

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

146 SYSOBJS64=

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



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

```
260         sigsuspend.o            \
261         statfs.o                \
262         statvfs.o               \
263         stty.o                  \
264         sync.o                  \
265         sysconfig.o             \
266         sysfs.o                 \
267         sysinfo.o               \
268         syslwp.o                \
269         times.o                 \
270         ulimit.o                \
271         umask.o                 \
272         umount2.o               \
273         utssys.o                \
274         uucopy.o                \
275         vhangup.o               \
276         waitid.o                \
277         write.o                 \
278         writev.o                \
279         yield.o

281 SYSOBJS=                        \
282         __clock_gettime.o       \
283         __getcontext.o          \
284         __uadmin.o              \
285         _lwp_mutex_unlock.o     \
286         _stack_grow.o           \
287         door.o                  \
288         forkx.o                 \
289         forkallx.o              \
290         getcontext.o            \
291         gettimeofday.o          \
292         lwp_private.o           \
293         nuname.o                \
294         syscall.o               \
295         sysi86.o                \
296         tls_get_addr.o          \
297         uadmin.o                \
298         umount.o                \
299         uname.o                 \
300         vforkx.o

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

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



```
326         finite.o                \
327         fp_data.o               \
328         func_decim.o            \
329         gconvert.o              \
330         hex_bin.o               \
331         ieee_globals.o          \
332         pack_float.o            \
333         sigfpe.o                \
334         string_decim.o

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

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



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

```
524         stack.o                 \
525         stpcpy.o                \
526         stpncpy.o               \
527         str2sig.o               \
528         strcase_charmap.o       \
529         strcat.o                \
530         strchr.o                \
531         strchrnul.o             \
532         strcspn.o               \
533         strdup.o                \
534         strerror.o              \
535         strlcat.o               \
536         strlcpy.o               \
537         strncat.o               \
538         strndup.o               \
539         strpbrk.o               \
540         strrchr.o               \
541         strsep.o                \
542         strsignal.o             \
543         strspn.o                \
544         strstr.o                \
545         strtod.o                \
546         strtoimax.o             \
547         strtok.o                \
548         strtok_r.o              \
549         strtoumax.o             \
550         swab.o                  \
551         swapctl.o               \
552         sysconf.o               \
553         syslog.o                \
554         tcdrain.o               \
555         tcflow.o                \
556         tcflush.o               \
557         tcgetattr.o             \
558         tcgetpgrp.o             \
559         tcgetsid.o              \
560         tcsendbreak.o           \
561         tcsetattr.o             \
562         tcsetpgrp.o             \
563         tell.o                  \
564         telldir.o               \
565         tfind.o                 \
566         time_data.o             \
567         time_gdata.o            \
568         tls_data.o              \
569         truncate.o              \
570         tsdalloc.o              \
571         tsearch.o               \
572         ttyname.o               \
573         ttyslot.o               \
574         ualarm.o                \
575         ucred.o                 \
576         valloc.o                \
577         vlfmt.o                 \
578         vpfmt.o                 \
579         waitpid.o               \
580         walkstack.o             \
581         wdata.o                 \
582         xgetwidth.o             \
583         xpg4.o                  \
584         xpg6.o

586 PORTPRINT_W=                   \
587         doprnt_w.o

589 PORTPRINT=                     \
```



```
590         asprintf.o              \
591         doprnt.o                \
592         fprintf.o               \
593         printf.o                \
594         snprintf.o              \
595         sprintf.o               \
596         vfprintf.o              \
597         vprintf.o               \
598         vsnprintf.o             \
599         vsprintf.o              \
600         vwprintf.o              \
601         wprintf.o

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

607 PORTSTDIO_W=                    \
608         doscan_w.o

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

```
656         setbuffer.o             \
657         setvbuf.o               \
658         system.o                \
659         tempnam.o               \
660         tmpfile.o               \
661         tmpnam_r.o              \
662         ungetc.o                \
663         vscanf.o                \
664         vwscanf.o               \
665         wscanf.o

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

713 PORTI18N_COND=                  \
714         wcstol_longlong.o       \
715         wcstoul_longlong.o

717 PORTLOCALE=                     \
718         big5.o                  \
719         btowc.o                 \
720         collate.o               \
721         collcmp.o               \
```



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

```
788         wctrans.o               \
789         wctype.o                \
790         wcwidth.o               \
791         wscol.o

793 AIOOBJS=                        \
794         aio.o                   \
795         aio_alloc.o             \
796         posix_aio.o

798 RTOBJS=                         \
799         clock_timer.o           \
800         mqueue.o                \
801         pos4obj.o               \
802         sched.o                 \
803         sem.o                   \
804         shm.o                   \
805         sigev_thread.o

807 TPOOLOBJS=                      \
808         thread_pool.o

810 THREADSOBJS=                    \
811         alloc.o                 \
812         assfail.o               \
813         cancel.o                \
814         door_calls.o            \
815         tmem.o                  \
816         pthr_attr.o             \
817         pthr_barrier.o          \
818         pthr_cond.o             \
819         pthr_mutex.o            \
820         pthr_rwlock.o           \
821         pthread.o               \
822         rwlock.o                \
823         scalls.o                \
824         sema.o                  \
825         sigaction.o             \
826         spawn.o                 \
827         synch.o                 \
828         tdb_agent.o             \
829         thr.o                   \
830         thread_interface.o      \
831         tls.o                   \
832         tsd.o

834 THREADSMACHOBJS=               \
835         machdep.o

837 THREADSASMOBJS=                \
838         asm_subr.o

840 UNICODEOBJS=                   \
841         u8_textprep.o           \
842         uconv.o

844 UNWINDMACHOBJS=                \
845         call_frame_inst.o       \
846         eh_frame.o              \
847         thrp_unwind.o           \
848         unwind.o

850 pics/unwind.o:= COPTFLAG64 =

852 UNWINDASMOBJS=                 \
853         unwind_frame.o
```



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

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

```
PORTREGEX=                      \
        glob.o                  \
        regcmp.o                \
        regex.o                 \
        wordexp.o

VALUES=                         \
        values-Xa.o

MOSTOBJS=                       \
        $(STRETS)               \
        $(CRTOBJS)              \
        $(DYNOBJS)              \
        $(FPOBJS)               \
        $(I386FPOBJS)           \
        $(FPASMOBJS)            \
        $(ATOMICOBJS)           \
        $(CHACHAOBJS)           \
        $(XATTROBJS)            \
        $(COMOBJS)              \
        $(GENOBJS)              \
        $(PORTFP)               \
        $(PORTGEN)              \
        $(PORTGEN64)            \
        $(PORTI18N)             \
        $(PORTI18N_COND)        \
        $(PORTLOCALE)           \
        $(PORTPRINT)            \
        $(PORTPRINT_W)          \
        $(PORTREGEX)            \
        $(PORTSTDIO)            \
        $(PORTSTDIO64)          \
        $(PORTSTDIO_W)          \
        $(PORTSYS)              \
        $(PORTSYS64)            \
        $(AIOOBJS)              \
        $(RTOBJS)               \
        $(TPOOLOBJS)            \
        $(THREADSOBJS)          \
        $(THREADSMACHOBJS)      \
        $(THREADSASMOBJS)       \
        $(UNICODEOBJS)          \
        $(UNWINDMACHOBJS)       \
        $(UNWINDASMOBJS)        \
        $(COMSYSOBJS)           \
        $(SYSOBJS)              \
        $(COMSYSOBJS64)         \
        $(SYSOBJS64)            \
        $(VALUES)

TRACEOBJS=                      \
        plockstat.o

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



```
# include common library definitions
include ../../Makefile.lib
include ../../Makefile.lib.64

CFLAGS64 += $(CTF_FLAGS)

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

CERRWARN += -_gcc=-Wno-parentheses
CERRWARN += -_gcc=-Wno-switch
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-unused-value
CERRWARN += -_gcc=-Wno-unused-label
CERRWARN += -_gcc=-Wno-unused-variable
CERRWARN += -_gcc=-Wno-type-limits
CERRWARN += -_gcc=-Wno-char-subscripts
CERRWARN += -_gcc=-Wno-clobbered
CERRWARN += -_gcc=-Wno-unused-function
CERRWARN += -_gcc=-Wno-address

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

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

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

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

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

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

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

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

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

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

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

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

```
1052 # libc method of building an archive
1053 # The "$(GREP) -v ' L '" part is necessary only until
1054 # lorder is fixed to ignore thread-local variables.
1055 BUILD.AR= $(RM) $@ ; \
1056         $(AR) q $@ `$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1058 # extra files for the clean target
1059 CLEANFILES=                     \
1060         $(LIBCDIR)/port/gen/errlst.c    \
1061         $(LIBCDIR)/port/gen/new_list.c  \
1062         assym.h                 \
1063         genassym                \
1064         crt/_rtld.s             \
1065         pics/crti.o             \
1066         pics/crtn.o             \
1067         $(ALTPICS)

1069 CLOBBERFILES += $(LIB_PIC)

1071 # list of C source for lint
1072 SRCS=                                           \
1073         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1074         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1075         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1076         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1077         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1078         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1079         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1080         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1081         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1082         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1083         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1084         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1085         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1086         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1087         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1088         $(THREADSMACHOBJS:%.o=threads/%.c)              \
1089         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1090         $(UNWINDMACHOBJS:%.o=unwind/%.c)                \
1091         $(FPOBJS:%.o=fp/%.c)                            \
1092         $(I386FPOBJS:%.o=$(LIBCDIR)/i386/fp/%.c)                \
1093         $(LIBCBASE)/gen/ecvt.c                          \
1094         $(LIBCBASE)/gen/makectxt.c                      \
1095         $(LIBCBASE)/gen/siginfolst.c                    \
1096         $(LIBCBASE)/gen/siglongjmp.c                    \
1097         $(LIBCBASE)/gen/sync_instruction_memory.c       \
1098         $(LIBCBASE)/sys/uadmin.c

1100 # conditional assignments
1101 # $(DYNLIB) $(LIB_PIC) := DYNOBJS = _rtbootld.o
1102 $(DYNLIB) := CRTI = crti.o
1103 $(DYNLIB) := CRTN = crtn.o

1105 # Files which need the threads .il inline template
1106 TIL=                            \
1107         aio.o                   \
1108         alloc.o                 \
1109         assfail.o               \
1110         atexit.o                \
1111         atfork.o                \
1112         cancel.o                \
1113         door_calls.o            \
1114         err.o                   \
1115         errno.o                 \
1116         lwp.o                   \
1117         ma.o                    \
```



```
1118         machdep.o               \
1119         posix_aio.o             \
1120         pthr_attr.o             \
1121         pthr_barrier.o          \
1122         pthr_cond.o             \
1123         pthr_mutex.o            \
1124         pthr_rwlock.o           \
1125         pthread.o               \
1126         rand.o                  \
1127         rwlock.o                \
1128         scalls.o                \
1129         sched.o                 \
1130         sema.o                  \
1131         sigaction.o             \
1132         sigev_thread.o          \
1133         spawn.o                 \
1134         stack.o                 \
1135         synch.o                 \
1136         tdb_agent.o             \
1137         thr.o                   \
1138         thread_interface.o      \
1139         thread_pool.o           \
1140         thrp_unwind.o           \
1141         tls.o                   \
1142         tmem.o                  \
1143         tsd.o

1145 $(TIL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/threads/amd64.il

1147 # pics/mul64.o := CFLAGS64 += crt/mul64.il

1149 # large-file-aware components that should be built large

1151 #$(COMSYSOBJS64:%=pics/%) := \
1152 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1154 #$(SYSOBJS64:%=pics/%) := \
1155 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1157 #$(PORTGEN64:%=pics/%) := \
1158 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1160 #$(PORTSTDIO64:%=pics/%) := \
1161 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1163 #$(PORTSYS64:%=pics/%) := \
1164 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1166 $(PORTSTDIO_W:%=pics/%) := \
1167         CPPFLAGS += -D_WIDE

1169 $(PORTPRINT_W:%=pics/%) := \
1170         CPPFLAGS += -D_WIDE

1172 $(PORTPRINT_C89:%=pics/%) := \
1173         CPPFLAGS += -D_C89_INTMAX32

1175 $(PORTSTDIO_C89:%=pics/%) := \
1176         CPPFLAGS += -D_C89_INTMAX32

1178 $(PORTI18N_COND:%=pics/%) := \
1179         CPPFLAGS += -D_WCS_LONGLONG

1181 pics/arc4random.o :=    CPPFLAGS += -I$(SRC)/common/crypto/chacha

1183 .KEEP_STATE:
```

```
1185 all: $(LIBS) $(LIB_PIC)

1187 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1188 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1189 lint    :=      LINTFLAGS64 += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1191 lint:
1192         @echo $(LINT.c) ... $(LDLIBS)
1193         @$(LINT.c) $(SRCS) $(LDLIBS)

1195 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1196 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1197 $(LINTLIB):= LINTFLAGS64=-nvx -m64

1199 # object files that depend on inline template
1200 $(TIL:%=pics/%): $(LIBCBASE)/threads/amd64.il
1201 # pics/mul64.o: crt/mul64.il

1203 # include common libc targets
1204 include ../Makefile.targ

1206 # We need to strip out all CTF data from the static library
1207 $(LIB_PIC) := DIR = pics
1208 $(LIB_PIC): pics $$(PICS)
1209         $(BUILD.AR)
1210         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1211         $(AR) -ts $@ > /dev/null
1212         $(POST_PROCESS_A)

1214 ASSYMDEP_OBJS=                  \
1215         _lwp_mutex_unlock.o     \
1216         _stack_grow.o           \
1217         asm_subr.o              \
1218         getcontext.o            \
1219         setjmp.o                \
1220         tls_get_addr.o          \
1221         vforkx.o

1223 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1225 # assym.h build rules

1227 GENASSYM_C = genassym.c

1229 genassym: $(GENASSYM_C)
1230         $(NATIVECC) $(NATIVE_CFLAGS) -Iinc -I$(LIBCDIR)/inc $(CPPFLAGS.native) \
1230         $(NATIVECC) -Iinc -I$(LIBCDIR)/inc $(CPPFLAGS.native)   \
1231                 -o $@ $(GENASSYM_C)

1233 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1235 assym.h: $(OFFSETS) genassym
1236         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1237         ./genassym >>$@

1239 # derived C source and related explicit dependencies
1240 $(LIBCDIR)/port/gen/errlst.c + \
1241 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1242         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1244 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1246 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   24521 Fri Feb 12 11:52:33 2016
new/usr/src/lib/libc/i386/Makefile.com
6648 illumos build should be explicit about C standards
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

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

VALUES=         values-Xa.o

# objects are grouped by source directory

# local objects
STRETS=

CRTOBJS=                        \
        cerror.o                \
        cerror64.o

DYNOBJS=                        \
        _rtbootld.o

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



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

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

ATOMICOBJS=                     \
        atomic.o

CHACHAOBJS=                     \
        chacha.o

XATTROBJS=                      \
        xattr_common.o

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

DTRACEOBJS=                     \
        dtrace_data.o

GENOBJS=                        \
        _div64.o                \
        _divdi3.o               \
        _getsp.o                \
        _mul64.o                \
        abs.o                   \
        alloca.o                \
        arc4random.o            \
        arc4random_uniform.o    \
        byteorder.o             \
        byteorder64.o           \
        cuexit.o                \
        ecvt.o                  \
        errlst.o                \
        i386_data.o             \
        ladd.o                  \
```

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

156 # sysobjs that contain large-file interfaces
157 COMSYSOBJS64=                   \
158         fstatvfs64.o            \
159         getdents64.o            \
160         getrlimit64.o           \
161         lseek64.o               \
162         mmap64.o                \
163         pread64.o               \
164         preadv64.o              \
165         pwrite64.o              \
166         pwritev64.o             \
167         setrlimit64.o           \
168         statvfs64.o

170 SYSOBJS64=

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



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

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

305 SYSOBJS=                       \
306         __clock_gettime.o       \
307         __getcontext.o          \
308         __uadmin.o              \
309         _lwp_mutex_unlock.o     \
310         _stack_grow.o           \
311         door.o                  \
312         forkx.o                 \
313         forkallx.o              \
314         getcontext.o            \
315         gettimeofday.o          \
316         lwp_private.o           \
317         nuname.o                \
318         ptrace.o                \
319         syscall.o               \
320         sysi86.o                \
321         tls_get_addr.o          \
322         uadmin.o                \
323         umount.o                \
324         uname.o                 \
325         vforkx.o                \
```



```
326         xstat.o

328 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
329 PORTGEN64=                      \
330         _xftw64.o               \
331         attropen64.o            \
332         ftw64.o                 \
333         mkstemp64.o             \
334         nftw64.o                \
335         tell64.o                \
336         truncate64.o

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

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

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



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

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



```
590         tcsendbreak.o           \
591         tcsetattr.o             \
592         tcsetpgrp.o             \
593         tell.o                  \
594         telldir.o               \
595         tfind.o                 \
596         time_data.o             \
597         time_gdata.o            \
598         tls_data.o              \
599         truncate.o              \
600         tsdalloc.o              \
601         tsearch.o               \
602         ttyname.o               \
603         ttyslot.o               \
604         ualarm.o                \
605         ucred.o                 \
606         valloc.o                \
607         vlfmt.o                 \
608         vpfmt.o                 \
609         waitpid.o               \
610         walkstack.o             \
611         wdata.o                 \
612         xgetwidth.o             \
613         xpg4.o                  \
614         xpg6.o

616 PORTPRINT_W=                    \
617         doprnt_w.o

619 PORTPRINT=                      \
620         asprintf.o              \
621         doprnt.o                \
622         fprintf.o               \
623         printf.o                \
624         snprintf.o              \
625         sprintf.o               \
626         vfprintf.o              \
627         vprintf.o               \
628         vsnprintf.o             \
629         vsprintf.o              \
630         vwprintf.o              \
631         wprintf.o

633 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
634 PORTPRINT_C89=                  \
635         vfprintf_c89.o          \
636         vprintf_c89.o           \
637         vsnprintf_c89.o         \
638         vsprintf_c89.o          \
639         vwprintf_c89.o

641 PORTSTDIO_C89=                  \
642         vscanf_c89.o            \
643         vwscanf_c89.o           \

645 # portable stdio objects that contain large file interfaces.
646 # Note: fopen64 is a special case, as we build it small.
647 PORTSTDIO64=                    \
648         fopen64.o               \
649         fpos64.o

651 PORTSTDIO_W=                    \
652         doscan_w.o

654 PORTSTDIO=                      \
655         __extensions.o          \
```

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

711 PORTI18N=                       \
712         getwchar.o              \
713         putwchar.o              \
714         putws.o                 \
715         strtows.o               \
716         wcsnlen.o               \
717         wcsstr.o                \
718         wcstoimax.o             \
719         wcstol.o                \
720         wcstoul.o               \
721         wcswcs.o                \
```



```
722         wmemchr.o               \
723         wmemcmp.o               \
724         wmemcpy.o               \
725         wmemmove.o              \
726         wmemset.o               \
727         wscat.o                 \
728         wschr.o                 \
729         wscmp.o                 \
730         wscpy.o                 \
731         wscspn.o                \
732         wsdup.o                 \
733         wslen.o                 \
734         wsncat.o                \
735         wsncmp.o                \
736         wsncpy.o                \
737         wspbrk.o                \
738         wsprintf.o              \
739         wsrchr.o                \
740         wsscanf.o               \
741         wsspn.o                 \
742         wstod.o                 \
743         wstok.o                 \
744         wstol.o                 \
745         wstoll.o                \
746         wsxfrm.o                \
747         gettext.o               \
748         gettext_gnu.o           \
749         gettext_real.o          \
750         gettext_util.o          \
751         plural_parser.o         \
752         wdresolve.o             \
753         _ctype.o                \
754         isascii.o               \
755         toascii.o

757 PORTI18N_COND=                  \
758         wcstol_longlong.o       \
759         wcstoul_longlong.o

761 PORTLOCALE=                    \
762         big5.o                  \
763         btowc.o                 \
764         collate.o               \
765         collcmp.o               \
766         euc.o                   \
767         fnmatch.o               \
768         fgetwc.o                \
769         fgetws.o                \
770         fix_grouping.o          \
771         fputwc.o                \
772         fputws.o                \
773         fwide.o                 \
774         gb18030.o               \
775         gb2312.o                \
776         gbk.o                   \
777         getdate.o               \
778         isdigit.o               \
779         iswctype.o              \
780         ldpart.o                \
781         lmessages.o             \
782         lnumeric.o              \
783         lmonetary.o             \
784         localeconv.o            \
785         localeimpl.o            \
786         mbftowc.o               \
787         mblen.o                 \
```

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

837 AIOOBJS=                       \
838         aio.o                   \
839         aio_alloc.o             \
840         posix_aio.o

842 RTOBJS=                        \
843         clock_timer.o           \
844         mqueue.o                \
845         pos4obj.o               \
846         sched.o                 \
847         sem.o                   \
848         shm.o                   \
849         sigev_thread.o

851 TPOOLOBJS=                     \
852         thread_pool.o
```



```
854 THREADSOBJS=                   \
855         alloc.o                 \
856         assfail.o               \
857         cancel.o                \
858         door_calls.o            \
859         tmem.o                  \
860         pthr_attr.o             \
861         pthr_barrier.o          \
862         pthr_cond.o             \
863         pthr_mutex.o            \
864         pthr_rwlock.o           \
865         pthread.o               \
866         rwlock.o                \
867         scalls.o                \
868         sema.o                  \
869         sigaction.o             \
870         spawn.o                 \
871         synch.o                 \
872         tdb_agent.o             \
873         thr.o                   \
874         thread_interface.o      \
875         tls.o                   \
876         tsd.o

878 THREADSMACHOBJS=               \
879         machdep.o

881 THREADSASMOBJS=                \
882         asm_subr.o

884 UNICODEOBJS=                   \
885         u8_textprep.o           \
886         uconv.o

888 UNWINDMACHOBJS=                \
889         unwind.o

891 UNWINDASMOBJS=                 \
892         unwind_frame.o

894 # objects that implement the transitional large file API
895 PORTSYS64=                     \
896         lockf64.o               \
897         stat64.o

899 PORTSYS=                       \
900         _autofssys.o            \
901         access.o                \
902         acctctl.o               \
903         bsd_signal.o            \
904         chmod.o                 \
905         chown.o                 \
906         corectl.o               \
907         epoll.o                 \
908         eventfd.o               \
909         exacctsys.o             \
910         execl.o                 \
911         execle.o                \
912         execv.o                 \
913         fcntl.o                 \
914         getpagesizes.o          \
915         getpeerucred.o          \
916         inst_sync.o             \
917         issetugid.o             \
918         label.o                 \
919         link.o                  \
```

```
 920         lockf.o                 \
 921         lwp.o                   \
 922         lwp_cond.o              \
 923         lwp_rwlock.o            \
 924         lwp_sigmask.o           \
 925         meminfosys.o            \
 926         mkdir.o                 \
 927         mknod.o                 \
 928         msgsys.o                \
 929         nfssys.o                \
 930         open.o                  \
 931         pgrpsys.o               \
 932         posix_sigwait.o         \
 933         ppriv.o                 \
 934         psetsys.o               \
 935         rctlsys.o               \
 936         readlink.o              \
 937         rename.o                \
 938         sbrk.o                  \
 939         semsys.o                \
 940         set_errno.o             \
 941         sharefs.o               \
 942         shmsys.o                \
 943         sidsys.o                \
 944         siginterrupt.o          \
 945         signal.o                \
 946         signalfd.o              \
 947         sigpending.o            \
 948         sigstack.o              \
 949         stat.o                  \
 950         symlink.o               \
 951         tasksys.o               \
 952         time.o                  \
 953         time_util.o             \
 954         timerfd.o               \
 955         ucontext.o              \
 956         unlink.o                \
 957         ustat.o                 \
 958         utimesys.o              \
 959         zone.o

 961 PORTREGEX=                      \
 962         glob.o                  \
 963         regcmp.o                \
 964         regex.o                 \
 965         wordexp.o

 967 PORTREGEX64=                    \
 968         glob64.o

 970 MOSTOBJS=                       \
 971         $(STRETS)               \
 972         $(CRTOBJS)              \
 973         $(DYNOBJS)              \
 974         $(FPOBJS)               \
 975         $(FPASMOBJS)            \
 976         $(ATOMICOBJS)           \
 977         $(CHACHAOBJS)           \
 978         $(XATTROBJS)            \
 979         $(COMOBJS)              \
 980         $(DTRACEOBJS)           \
 981         $(GENOBJS)              \
 982         $(PORTFP)               \
 983         $(PORTGEN)              \
 984         $(PORTGEN64)            \
 985         $(PORTI18N)             \
```



```
 986         $(PORTI18N_COND)        \
 987         $(PORTLOCALE)           \
 988         $(PORTPRINT)            \
 989         $(PORTPRINT_C89)        \
 990         $(PORTPRINT_W)          \
 991         $(PORTREGEX)            \
 992         $(PORTREGEX64)          \
 993         $(PORTSTDIO)            \
 994         $(PORTSTDIO64)          \
 995         $(PORTSTDIO_C89)        \
 996         $(PORTSTDIO_W)          \
 997         $(PORTSYS)              \
 998         $(PORTSYS64)            \
 999         $(AIOOBJS)              \
1000         $(RTOBJS)               \
1001         $(TPOOLOBJS)            \
1002         $(THREADSOBJS)          \
1003         $(THREADSMACHOBJS)      \
1004         $(THREADSASMOBJS)       \
1005         $(UNICODEOBJS)          \
1006         $(UNWINDMACHOBJS)       \
1007         $(UNWINDASMOBJS)        \
1008         $(COMSYSOBJS)           \
1009         $(SYSOBJS)              \
1010         $(COMSYSOBJS64)         \
1011         $(SYSOBJS64)            \
1012         $(VALUES)

1014 TRACEOBJS=                      \
1015         plockstat.o

1017 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
1018 # modules whose source is provided in the $(SRC)/lib/common directory.
1019 # This must be done because otherwise the Sun C compiler would insert
1020 # its own versions of these modules and those versions contain code
1021 # to call out to C++ initialization functions.  Such C++ initialization
1022 # functions can call back into libc before thread initialization is
1023 # complete and this leads to segmentation violations and other problems.
1024 # Since libc contains no C++ code, linking with the minimal crti.o and
1025 # crtn.o modules is safe and avoids the problems described above.
1026 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
1027 CRTSRCS= ../../common/i386

1029 LDPASS_OFF=     $(POUND_SIGN)

1031 # include common library definitions
1032 include ../../Makefile.lib

1034 # we need to override the default SONAME here because we might
1035 # be building a variant object (still libc.so.1, but different filename)
1036 SONAME = libc.so.1

1038 CFLAGS += $(CCVERBOSE) $(CTF_FLAGS)

1040 # This is necessary to avoid problems with calling _ex_unwind().
1041 # We probably don't want any inlining anyway.
1042 XINLINE = -xinline=
1043 CFLAGS += $(XINLINE)

1045 CERRWARN += -_gcc=-Wno-parentheses
1046 CERRWARN += -_gcc=-Wno-switch
1047 CERRWARN += -_gcc=-Wno-uninitialized
1048 CERRWARN += -_gcc=-Wno-unused-value
1049 CERRWARN += -_gcc=-Wno-unused-label
1050 CERRWARN += -_gcc=-Wno-unused-variable
1051 CERRWARN += -_gcc=-Wno-type-limits
```

```
1052 CERRWARN += -_gcc=-Wno-char-subscripts
1053 CERRWARN += -_gcc=-Wno-clobbered
1054 CERRWARN += -_gcc=-Wno-unused-function
1055 CERRWARN += -_gcc=-Wno-address

1057 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1058 # enables ASSERT() checking in the threads portion of the library.
1059 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1060 THREAD_DEBUG =
1061 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1063 # Make string literals read-only to save memory.
1064 CFLAGS += $(XSTRCONST)

1066 ALTPICS= $(TRACEOBJS:%=pics/%)

1068 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) \
1069                 $(EXTPICS) $(LDLIBS)

1071 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1073 #
1074 # EXTN_CPPFLAGS and EXTN_CFLAGS set in enclosing Makefile
1075 #
1076 CFLAGS +=       $(EXTN_CFLAGS)
1077 CPPFLAGS=       -D_REENTRANT -Di386 $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1078                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1079 ASFLAGS=        $(AS_PICFLAGS) -P -D__STDC__ -D_ASM $(CPPFLAGS) $(i386_AS_XARCH)

1081 # As a favor to the dtrace syscall provider, libc still calls the
1082 # old syscall traps that have been obsoleted by the *at() interfaces.
1083 # Delete this to compile libc using only the new *at() system call traps
1084 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1086 # Inform the run-time linker about libc specialized initialization
1087 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1088 DYNFLAGS +=     $(RTLDINFO)

1090 # Force libc's internal references to be resolved immediately upon loading
1091 # in order to avoid critical region problems.  Since almost all libc symbols
1092 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1093 DYNFLAGS +=     -znow

1095 DYNFLAGS +=     -e __rtboot
1096 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1098 # Inform the kernel about the initial DTrace area (in case
1099 # libc is being used as the interpreter / runtime linker).
1100 DTRACE_DATA =   -zdtrace=dtrace_data
1101 DYNFLAGS +=     $(DTRACE_DATA)

1103 # DTrace needs an executable data segment.
1104 MAPFILE.NED=

1106 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1108 # Override this top level flag so the compiler builds in its native
1109 # C99 mode.  This has been enabled to support the complex arithmetic
1110 # added to libc.
1111 C99MODE=        $(C99_ENABLE)

1113 # libc method of building an archive
1114 # The "$(GREP) -v ' L '" part is necessary only until
1115 # lorder is fixed to ignore thread-local variables.
1116 BUILD.AR= $(RM) $@ ; \
1117         $(AR) q $@ `$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR
```



```
1119 # extra files for the clean target
1120 CLEANFILES=                     \
1121         $(LIBCDIR)/port/gen/errlst.c    \
1122         $(LIBCDIR)/port/gen/new_list.c  \
1123         assym.h                 \
1124         genassym                \
1125         crt/_rtld.s             \
1126         crt/_rtbootld.s         \
1127         pics/_rtbootld.o        \
1128         pics/crti.o             \
1129         pics/crtn.o             \
1130         $(ALTPICS)

1132 CLOBBERFILES += $(LIB_PIC)

1134 # list of C source for lint
1135 SRCS=                                                   \
1136         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1137         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1138         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1139         $(DTRACEOBJS:%.o=$(SRC)/common/dtrace/%.c)      \
1140         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1141         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1142         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1143         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1144         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1145         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1146         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1147         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1148         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1149         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1150         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1151         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1152         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1153         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1154         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1155         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1156         $(LIBCBASE)/gen/ecvt.c                          \
1157         $(LIBCBASE)/gen/makectxt.c                      \
1158         $(LIBCBASE)/gen/siginfolst.c                    \
1159         $(LIBCBASE)/gen/siglongjmp.c                    \
1160         $(LIBCBASE)/gen/strcmp.c                        \
1161         $(LIBCBASE)/gen/sync_instruction_memory.c       \
1162         $(LIBCBASE)/sys/ptrace.c                        \
1163         $(LIBCBASE)/sys/uadmin.c

1165 # conditional assignments
1166 $(DYNLIB) := CRTI = crti.o
1167 $(DYNLIB) := CRTN = crtn.o

1169 # Files which need the threads .il inline template
1170 TIL=                            \
1171         aio.o                   \
1172         alloc.o                 \
1173         assfail.o               \
1174         atexit.o                \
1175         atfork.o                \
1176         cancel.o                \
1177         door_calls.o            \
1178         err.o                   \
1179         errno.o                 \
1180         lwp.o                   \
1181         ma.o                    \
1182         machdep.o               \
1183         posix_aio.o             \
```

```
1184         pthr_attr.o             \
1185         pthr_barrier.o          \
1186         pthr_cond.o             \
1187         pthr_mutex.o            \
1188         pthr_rwlock.o           \
1189         pthread.o               \
1190         rand.o                  \
1191         rwlock.o                \
1192         scalls.o                \
1193         sched.o                 \
1194         sema.o                  \
1195         sigaction.o             \
1196         sigev_thread.o          \
1197         spawn.o                 \
1198         stack.o                 \
1199         synch.o                 \
1200         tdb_agent.o             \
1201         thr.o                   \
1202         thread_interface.o      \
1203         thread_pool.o           \
1204         tls.o                   \
1205         tsd.o                   \
1206         tmem.o                  \
1207         unwind.o

1209 THREADS_INLINES = $(LIBCBASE)/threads/i386.il
1210 $(TIL:%=pics/%) := CFLAGS += $(THREADS_INLINES)

1212 # pics/mul64.o := CFLAGS += $(LIBCBASE)/crt/mul64.il

1214 # large-file-aware components that should be built large

1216 $(COMSYSOBJS64:%=pics/%) := \
1217         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1219 $(SYSOBJS64:%=pics/%) := \
1220         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1222 $(PORTGEN64:%=pics/%) := \
1223         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1225 $(PORTREGEX64:%=pics/%) := \
1226         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1228 $(PORTSTDIO64:%=pics/%) := \
1229         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1231 $(PORTSYS64:%=pics/%) := \
1232         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1234 $(PORTSTDIO_W:%=pics/%) := \
1235         CPPFLAGS += -D_WIDE

1237 $(PORTPRINT_W:%=pics/%) := \
1238         CPPFLAGS += -D_WIDE

1240 $(PORTPRINT_C89:%=pics/%) := \
1241         CPPFLAGS += -D_C89_INTMAX32

1243 $(PORTSTDIO_C89:%=pics/%) := \
1244         CPPFLAGS += -D_C89_INTMAX32

1246 $(PORTI18N_COND:%=pics/%) := \
1247         CPPFLAGS += -D_WCS_LONGLONG

1249 pics/arc4random.o :=    CPPFLAGS += -I$(SRC)/common/crypto/chacha
```



```
1251 .KEEP_STATE:

1253 all: $(LIBS) $(LIB_PIC)

1255 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1256 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1257 lint    :=      LINTFLAGS += -mn -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED

1259 lint:
1260         @echo $(LINT.c) ...
1261         @$(LINT.c) $(SRCS) $(LDLIBS)

1263 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1264 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1265 $(LINTLIB):= LINTFLAGS=-nvx

1267 # object files that depend on inline template
1268 $(TIL:%=pics/%): $(LIBCBASE)/threads/i386.il
1269 # pics/mul64.o: $(LIBCBASE)/crt/mul64.il

1271 # include common libc targets
1272 include $(LIBCDIR)/Makefile.targ

1274 # We need to strip out all CTF and DOF data from the static library
1275 $(LIB_PIC) := DIR = pics
1276 $(LIB_PIC): pics $$(PICS)
1277         $(BUILD.AR)
1278         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1279         $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1280         $(AR) -ts $@ > /dev/null
1281         $(POST_PROCESS_A)

1283 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.c
1284         $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S $(C_PICFLAGS) \
1285             $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1286         $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
1287         $(RM) $(LIBCBASE)/crt/_rtld.s

1289 # partially built from C source
1290 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1291         $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
1292         $(CTFCONVERT_O)

1294 ASSYMDEP_OBJS=                  \
1295         _lwp_mutex_unlock.o     \
1296         _stack_grow.o           \
1297         getcontext.o            \
1298         setjmp.o                \
1299         tls_get_addr.o          \
1300         vforkx.o

1302 $(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

1304 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1306 # assym.h build rules

1308 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1310 genassym: $(GENASSYM_C)
1311         $(NATIVECC) $(NATIVE_CFLAGS) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1311         $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc  \
1312                 -D__EXTENSIONS__ $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1314 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in
```

```
1316 assym.h: $(OFFSETS) genassym
1317         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1318         ./genassym >>$@

1320 # derived C source and related explicit dependencies
1321 $(LIBCDIR)/port/gen/errlst.c + \
1322 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1323         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1325 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1327 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   26307 Fri Feb 12 11:52:33 2016
new/usr/src/lib/libc/sparc/Makefile.com
6648 illumos build should be explicit about C standards
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

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

# objects are grouped by source directory

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

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

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

DYNOBJS=                        \
        _rtbootld.o
```



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

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

$(__GNUC)FPASMOBJS +=           \
        __quad.o

ATOMICOBJS=                     \
        atomic.o

CHACHAOBJS=                     \
        chacha.o

XATTROBJS=                      \
        xattr_common.o

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

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

136 DTRACEOBJS=                      \
137         dtrace_data.o

139 GENOBJS=                        \
140         _getsp.o                \
141         _xregs_clrptr.o         \
142         abs.o                   \
143         alloca.o                \
144         arc4random.o            \
145         arc4random_uniform.o    \
146         ascii_strcasecmp.o      \
147         byteorder.o             \
148         cuexit.o                \
149         ecvt.o                  \
150         errlst.o                \
151         getctxt.o               \
152         ladd.o                  \
153         lmul.o                  \
154         lock.o                  \
155         lshiftl.o               \
156         lsign.o                 \
157         lsub.o                  \
158         makectxt.o              \
159         memchr.o                \
160         memcmp.o                \
161         new_list.o              \
162         setjmp.o                \
163         siginfolst.o            \
164         siglongjmp.o            \
165         smt_pause.o             \
166         sparc_data.o            \
167         strchr.o                \
168         strcmp.o                \
169         strlcpy.o               \
170         strncmp.o               \
171         strncpy.o               \
172         strnlen.o               \
173         swapctxt.o              \
174         sync_instruction_memory.o

176 # sysobjs that contain large-file interfaces
177 COMSYSOBJS64=                   \
178         fstatvfs64.o            \
179         getdents64.o            \
180         getrlimit64.o           \
181         lseek64.o               \
182         mmap64.o                \
183         pread64.o               \
184         preadv64.o              \
185         pwrite64.o              \
186         pwritev64.o             \
187         setrlimit64.o           \
188         statvfs64.o

190 SYSOBJS64=

192 COMSYSOBJS=                     \
193         __clock_timer.o         \
```



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

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

325 SYSOBJS=                       \
```



```
326         __clock_gettime.o       \
327         __getcontext.o          \
328         _lwp_mutex_unlock.o     \
329         _stack_grow.o           \
330         __uadmin.o              \
331         door.o                  \
332         forkx.o                 \
333         forkallx.o              \
334         gettimeofday.o          \
335         ptrace.o                \
336         syscall.o               \
337         tls_get_addr.o          \
338         uadmin.o                \
339         umount.o                \
340         uname.o                 \
341         vforkx.o

343 # objects under $(LIBCDIR)/port which contain transitional large file interfaces
344 PORTGEN64=                      \
345         _xftw64.o               \
346         attropen64.o            \
347         ftw64.o                 \
348         mkstemp64.o             \
349         nftw64.o                \
350         tell64.o                \
351         truncate64.o

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

```
392         muldi3.o                \
393         qdivrem.o               \
394         ucmpdi2.o               \
395         udivdi3.o               \
396         umoddi3.o

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



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

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



```
590         stack.o                 \
591         stpcpy.o                \
592         stpncpy.o               \
593         str2sig.o               \
594         strcase_charmap.o       \
595         strcat.o                \
596         strchrnul.o             \
597         strcspn.o               \
598         strdup.o                \
599         strerror.o              \
600         strlcat.o               \
601         strncat.o               \
602         strndup.o               \
603         strpbrk.o               \
604         strrchr.o               \
605         strsep.o                \
606         strsignal.o             \
607         strspn.o                \
608         strstr.o                \
609         strtod.o                \
610         strtoimax.o             \
611         strtok.o                \
612         strtok_r.o              \
613         strtoumax.o             \
614         swab.o                  \
615         swapctl.o               \
616         sysconf.o               \
617         syslog.o                \
618         tcdrain.o               \
619         tcflow.o                \
620         tcflush.o               \
621         tcgetattr.o             \
622         tcgetpgrp.o             \
623         tcgetsid.o              \
624         tcsendbreak.o           \
625         tcsetattr.o             \
626         tcsetpgrp.o             \
627         tell.o                  \
628         telldir.o               \
629         tfind.o                 \
630         time_data.o             \
631         time_gdata.o            \
632         tls_data.o              \
633         truncate.o              \
634         tsdalloc.o              \
635         tsearch.o               \
636         ttyname.o               \
637         ttyslot.o               \
638         ualarm.o                \
639         ucred.o                 \
640         valloc.o                \
641         vlfmt.o                 \
642         vpfmt.o                 \
643         waitpid.o               \
644         walkstack.o             \
645         wdata.o                 \
646         xgetwidth.o             \
647         xpg4.o                  \
648         xpg6.o

650 PORTPRINT_W=                    \
651         doprnt_w.o

653 PORTPRINT=                      \
654         asprintf.o              \
655         doprnt.o                \
```

```
656         fprintf.o               \
657         printf.o                \
658         snprintf.o              \
659         sprintf.o               \
660         vfprintf.o              \
661         vprintf.o               \
662         vsnprintf.o             \
663         vsprintf.o              \
664         vwprintf.o              \
665         wprintf.o

667 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)
668 PORTPRINT_C89=                  \
669         vfprintf_c89.o          \
670         vprintf_c89.o           \
671         vsnprintf_c89.o         \
672         vsprintf_c89.o          \
673         vwprintf_c89.o

675 PORTSTDIO_C89=                 \
676         vscanf_c89.o            \
677         vwscanf_c89.o

679 # portable stdio objects that contain large file interfaces.
680 # Note: fopen64 is a special case, as we build it small.
681 PORTSTDIO64=                    \
682         fopen64.o               \
683         fpos64.o

685 PORTSTDIO_W=                    \
686         doscan_w.o

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



```
722         getpass.o               \
723         gets.o                  \
724         getw.o                  \
725         popen.o                 \
726         putc.o                  \
727         putchar.o               \
728         puts.o                  \
729         putw.o                  \
730         rewind.o                \
731         scanf.o                 \
732         setbuf.o                \
733         setbuffer.o             \
734         setvbuf.o               \
735         system.o                \
736         tempnam.o               \
737         tmpfile.o               \
738         tmpnam_r.o              \
739         ungetc.o                \
740         mse.o                   \
741         vscanf.o                \
742         vwscanf.o               \
743         wscanf.o

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

```
788         isascii.o               \
789         toascii.o

791 PORTI18N_COND=                  \
792         wcstol_longlong.o       \
793         wcstoul_longlong.o

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



```
854         utf8.o                  \
855         wcrtomb.o               \
856         wcscasecmp.o            \
857         wcscoll.o               \
858         wcsftime.o              \
859         wcsnrtombs.o            \
860         wcsrtombs.o             \
861         wcstombs.o              \
862         wcswidth.o              \
863         wcsxfrm.o               \
864         wctob.o                 \
865         wctomb.o                \
866         wctrans.o               \
867         wctype.o                \
868         wcwidth.o               \
869         wscol.o

871 AIOOBJS=                        \
872         aio.o                   \
873         aio_alloc.o             \
874         posix_aio.o

876 RTOBJS=                         \
877         clock_timer.o           \
878         mqueue.o                \
879         pos4obj.o               \
880         sched.o                 \
881         sem.o                   \
882         shm.o                   \
883         sigev_thread.o

885 TPOOLOBJS=                      \
886         thread_pool.o

888 THREADSOBJS=                    \
889         alloc.o                 \
890         assfail.o               \
891         cancel.o                \
892         door_calls.o            \
893         tmem.o                  \
894         pthr_attr.o             \
895         pthr_barrier.o          \
896         pthr_cond.o             \
897         pthr_mutex.o            \
898         pthr_rwlock.o           \
899         pthread.o               \
900         rwlock.o                \
901         scalls.o                \
902         sema.o                  \
903         sigaction.o             \
904         spawn.o                 \
905         synch.o                 \
906         tdb_agent.o             \
907         thr.o                   \
908         thread_interface.o      \
909         tls.o                   \
910         tsd.o

912 THREADSMACHOBJS=                \
913         machdep.o

915 THREADSASMOBJS=                 \
916         asm_subr.o

918 UNICODEOBJS=                    \
919         u8_textprep.o           \
```

```
920         uconv.o

922 UNWINDMACHOBJS=                 \
923         unwind.o

925 UNWINDASMOBJS=                 \
926         unwind_frame.o

928 # objects that implement the transitional large file API
929 PORTSYS64=                      \
930         lockf64.o               \
931         stat64.o

933 PORTSYS=                        \
934         _autofssys.o            \
935         access.o                \
936         acctctl.o               \
937         bsd_signal.o            \
938         chmod.o                 \
939         chown.o                 \
940         corectl.o               \
941         epoll.o                 \
942         eventfd.o               \
943         exacctsys.o             \
944         execl.o                 \
945         execle.o                \
946         execv.o                 \
947         fcntl.o                 \
948         getpagesizes.o          \
949         getpeerucred.o          \
950         inst_sync.o             \
951         issetugid.o             \
952         label.o                 \
953         link.o                  \
954         lockf.o                 \
955         lwp.o                   \
956         lwp_cond.o              \
957         lwp_rwlock.o            \
958         lwp_sigmask.o           \
959         meminfosys.o            \
960         mkdir.o                 \
961         mknod.o                 \
962         msgsys.o                \
963         nfssys.o                \
964         open.o                  \
965         pgrpsys.o               \
966         posix_sigwait.o         \
967         ppriv.o                 \
968         psetsys.o               \
969         rctlsys.o               \
970         readlink.o              \
971         rename.o                \
972         sbrk.o                  \
973         semsys.o                \
974         set_errno.o             \
975         sharefs.o               \
976         shmsys.o                \
977         sidsys.o                \
978         siginterrupt.o          \
979         signal.o                \
980         signalfd.o              \
981         sigpending.o            \
982         sigstack.o              \
983         stat.o                  \
984         symlink.o               \
985         tasksys.o               \
```



```
986         time.o                  \
987         time_util.o             \
988         timerfd.o               \
989         ucontext.o              \
990         unlink.o                \
991         ustat.o                 \
992         utimesys.o              \
993         zone.o

995 PORTREGEX=                      \
996         glob.o                  \
997         regcmp.o                \
998         regex.o                 \
999         wordexp.o

1001 PORTREGEX64=                   \
1002         glob64.o

1004 VALUES= values-Xa.o

1006 MOSTOBJS=                      \
1007         $(STRETS)              \
1008         $(CRTOBJS)             \
1009         $(DYNOBJS)             \
1010         $(FPOBJS)              \
1011         $(FPASMOBJS)           \
1012         $(ATOMICOBJS)          \
1013         $(CHACHAOBJS)          \
1014         $(XATTROBJS)           \
1015         $(COMOBJS)             \
1016         $(DTRACEOBJS)          \
1017         $(GENOBJS)             \
1018         $(PRFOBJS)             \
1019         $(PORTFP)              \
1020         $(PORTGEN)             \
1021         $(PORTGEN64)           \
1022         $(PORTI18N)            \
1023         $(PORTI18N_COND)       \
1024         $(PORTLOCALE)          \
1025         $(PORTPRINT)           \
1026         $(PORTPRINT_C89)       \
1027         $(PORTPRINT_W)         \
1028         $(PORTREGEX)           \
1029         $(PORTREGEX64)         \
1030         $(PORTSTDIO)           \
1031         $(PORTSTDIO64)         \
1032         $(PORTSTDIO_C89)       \
1033         $(PORTSTDIO_W)         \
1034         $(PORTSYS)             \
1035         $(PORTSYS64)           \
1036         $(AIOOBJS)             \
1037         $(RTOBJS)              \
1038         $(TPOOLOBJS)           \
1039         $(THREADSOBJS)         \
1040         $(THREADSMACHOBJS)     \
1041         $(THREADSASMOBJS)      \
1042         $(UNICODEOBJS)         \
1043         $(UNWINDMACHOBJS)      \
1044         $(UNWINDASMOBJS)       \
1045         $(COMSYSOBJS)          \
1046         $(SYSOBJS)             \
1047         $(COMSYSOBJS64)        \
1048         $(SYSOBJS64)           \
1049         $(VALUES)

1051 TRACEOBJS=                     \
```

```
1052         plockstat.o

1054 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
1055 # modules whose source is provided in the $(SRC)/lib/common directory.
1056 # This must be done because otherwise the Sun C compiler would insert
1057 # its own versions of these modules and those versions contain code
1058 # to call out to C++ initialization functions.  Such C++ initialization
1059 # functions can call back into libc before thread initialization is
1060 # complete and this leads to segmentation violations and other problems.
1061 # Since libc contains no C++ code, linking with the minimal crti.o and
1062 # crtn.o modules is safe and avoids the problems described above.
1063 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
1064 CRTSRCS= ../../common/sparc

1066 # include common library definitions
1067 include $(SRC)/lib/Makefile.lib

1069 # we need to override the default SONAME here because we might
1070 # be building a variant object (still libc.so.1, but different filename)
1071 SONAME = libc.so.1

1073 CFLAGS += $(CCVERBOSE)

1075 # This is necessary to avoid problems with calling _ex_unwind().
1076 # We probably don't want any inlining anyway.
1077 CFLAGS += -xinline=

1079 CERRWARN += -_gcc=-Wno-parentheses
1080 CERRWARN += -_gcc=-Wno-switch
1081 CERRWARN += -_gcc=-Wno-uninitialized
1082 CERRWARN += -_gcc=-Wno-unused-value
1083 CERRWARN += -_gcc=-Wno-unused-label
1084 CERRWARN += -_gcc=-Wno-unused-variable
1085 CERRWARN += -_gcc=-Wno-type-limits
1086 CERRWARN += -_gcc=-Wno-char-subscripts
1087 CERRWARN += -_gcc=-Wno-clobbered
1088 CERRWARN += -_gcc=-Wno-unused-function
1089 CERRWARN += -_gcc=-Wno-address

1091 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1092 # enables ASSERT() checking in the threads portion of the library.
1093 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1094 THREAD_DEBUG =
1095 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1097 # Make string literals read-only to save memory.
1098 CFLAGS += $(XSTRCONST)

1100 ALTPICS= $(TRACEOBJS:%=pics/%)

1102 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1104 MAPFILES =      $(LIBCDIR)/port/mapfile-vers

1106 CFLAGS +=       $(EXTN_CFLAGS)
1107 CPPFLAGS=       -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1108                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1109 ASFLAGS=        $(EXTN_ASFLAGS) -K pic -P -D__STDC__ -D_ASM $(CPPFLAGS) $(sparc_

1111 # As a favor to the dtrace syscall provider, libc still calls the
1112 # old syscall traps that have been obsoleted by the *at() interfaces.
1113 # Delete this to compile libc using only the new *at() system call traps
1114 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1116 # Inform the run-time linker about libc specialized initialization
1117 RTLDINFO =      -z rtldinfo=tls_rtldinfo
```



```
1118 DYNFLAGS +=     $(RTLDINFO)

1120 # Force libc's internal references to be resolved immediately upon loading
1121 # in order to avoid critical region problems.  Since almost all libc symbols
1122 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1123 DYNFLAGS +=     -znow

1125 DYNFLAGS +=     -e __rtboot
1126 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1128 # Inform the kernel about the initial DTrace area (in case
1129 # libc is being used as the interpreter / runtime linker).
1130 DTRACE_DATA =   -zdtrace=dtrace_data
1131 DYNFLAGS +=     $(DTRACE_DATA)

1133 # DTrace needs an executable data segment.
1134 MAPFILE.NED=

1136 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1138 # Override this top level flag so the compiler builds in its native
1139 # C99 mode.  This has been enabled to support the complex arithmetic
1140 # added to libc.
1141 C99MODE=        $(C99_ENABLE)

1143 # libc method of building an archive
1144 # The "$(GREP) -v ' L '" part is necessary only until
1145 # lorder is fixed to ignore thread-local variables.
1146 BUILD.AR= $(RM) $@ ; \
1147         $(AR) q $@ '$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1149 # extra files for the clean target
1150 CLEANFILES=                     \
1151         $(LIBCDIR)/port/gen/errlst.c    \
1152         $(LIBCDIR)/port/gen/new_list.c  \
1153         assym.h                 \
1154         genassym                \
1155         $(LIBCBASE)/crt/_rtld.s         \
1156         $(LIBCBASE)/crt/_rtbootld.s             \
1157         pics/_rtbootld.o        \
1158         pics/crti.o             \
1159         pics/crtn.o             \
1160         $(ALTPICS)

1162 CLOBBERFILES += $(LIB_PIC)

1164 # list of C source for lint
1165 SRCS=                                                   \
1166         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1167         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1168         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1169         $(DTRACEOBJS:%.o=$(SRC)/common/dtrace/%.c)      \
1170         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1171         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1172         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1173         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1174         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1175         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1176         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1177         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1178         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1179         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1180         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1181         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1182         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1183         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
```

```
1184         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1185         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1186         $(LIBCBASE)/crt/_ftou.c                         \
1187         $(LIBCBASE)/gen/_xregs_clrptr.c                 \
1188         $(LIBCBASE)/gen/byteorder.c                     \
1189         $(LIBCBASE)/gen/ecvt.c                          \
1190         $(LIBCBASE)/gen/getctxt.c                       \
1191         $(LIBCBASE)/gen/lmul.c                          \
1192         $(LIBCBASE)/gen/makectxt.c                      \
1193         $(LIBCBASE)/gen/siginfolst.c                    \
1194         $(LIBCBASE)/gen/siglongjmp.c                    \
1195         $(LIBCBASE)/gen/swapctxt.c                      \
1196         $(LIBCBASE)/sys/ptrace.c                        \
1197         $(LIBCBASE)/sys/uadmin.c

1199 # conditional assignments
1200 $(DYNLIB) := CRTI = crti.o
1201 $(DYNLIB) := CRTN = crtn.o

1203 # Files which need the threads .il inline template
1204 TIL=                            \
1205         aio.o                   \
1206         alloc.o                 \
1207         assfail.o               \
1208         atexit.o                \
1209         atfork.o                \
1210         cancel.o                \
1211         door_calls.o            \
1212         err.o                   \
1213         errno.o                 \
1214         getctxt.o               \
1215         lwp.o                   \
1216         ma.o                    \
1217         machdep.o               \
1218         posix_aio.o             \
1219         pthr_attr.o             \
1220         pthr_barrier.o          \
1221         pthr_cond.o             \
1222         pthr_mutex.o            \
1223         pthr_rwlock.o           \
1224         pthread.o               \
1225         rand.o                  \
1226         rwlock.o                \
1227         scalls.o                \
1228         sched.o                 \
1229         sema.o                  \
1230         sigaction.o             \
1231         sigev_thread.o          \
1232         spawn.o                 \
1233         stack.o                 \
1234         swapctxt.o              \
1235         synch.o                 \
1236         tdb_agent.o             \
1237         thr.o                   \
1238         thread_interface.o      \
1239         thread_pool.o           \
1240         tls.o                   \
1241         tsd.o                   \
1242         unwind.o

1244 $(TIL:%=pics/%) := CFLAGS += $(LIBCBASE)/threads/sparc.il

1246 # special kludge for inlines with 'cas':
1247 pics/rwlock.o pics/synch.o pics/lwp.o pics/door_calls.o := \
1248         sparc_CFLAGS += -_gcc=-Wa,-xarch=v8plus
```



```
1250 # Files in port/fp subdirectory that need base.il inline template
1251 IL=                             \
1252         __flt_decim.o           \
1253         decimal_bin.o

1255 $(IL:%=pics/%) := CFLAGS += $(LIBCBASE)/fp/base.il

1257 # Files in fp subdirectory which need __quad.il inline template
1258 QIL=                            \
1259         _Q_add.o                \
1260         _Q_cmp.o                \
1261         _Q_cmpe.o               \
1262         _Q_div.o                \
1263         _Q_dtoq.o               \
1264         _Q_fcc.o                \
1265         _Q_mul.o                \
1266         _Q_qtod.o               \
1267         _Q_qtoi.o               \
1268         _Q_qtos.o               \
1269         _Q_qtou.o               \
1270         _Q_sqrt.o               \
1271         _Q_stoq.o               \
1272         _Q_sub.o

1274 $(QIL:%=pics/%) := CFLAGS += $(LIBCDIR)/$(MACH)/fp/__quad.il
1275 pics/_Q%.o := sparc_COPTFLAG = -xO4 -dalign
1276 pics/__quad%.o := sparc_COPTFLAG = -xO4 -dalign

1278 # large-file-aware components that should be built large

1280 $(COMSYSOBJS64:%=pics/%) := \
1281         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1283 $(SYSOBJS64:%=pics/%) := \
1284         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1286 $(PORTGEN64:%=pics/%) := \
1287         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1289 $(PORTREGEX64:%=pics/%) := \
1290         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1292 $(PORTSTDIO64:%=pics/%) := \
1293         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1295 $(PORTSYS64:%=pics/%) := \
1296         CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1298 $(PORTSTDIO_W:%=pics/%) := \
1299         CPPFLAGS += -D_WIDE

1301 $(PORTPRINT_W:%=pics/%) := \
1302         CPPFLAGS += -D_WIDE

1304 # printf/scanf functions to support c89-sized intmax_t variables
1305 $(PORTPRINT_C89:%=pics/%) := \
1306         CPPFLAGS += -D_C89_INTMAX32

1308 $(PORTSTDIO_C89:%=pics/%) := \
1309         CPPFLAGS += -D_C89_INTMAX32

1311 $(PORTI18N_COND:%=pics/%) := \
1312         CPPFLAGS += -D_WCS_LONGLONG

1314 pics/arc4random.o :=    CPPFLAGS += -I$(SRC)/common/crypto/chacha
```

```
1316 # Files which need extra optimization
1317 pics/getenv.o := sparc_COPTFLAG = -xO4

1319 .KEEP_STATE:

1321 all: $(LIBS) $(LIB_PIC)

1323 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1324 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1325 lint    :=      LINTFLAGS += -mn

1327 lint:
1328         @echo $(LINT.c) ... $(LDLIBS)
1329         @$(LINT.c) $(SRCS) $(LDLIBS)

1331 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1332 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1333 $(LINTLIB):= LINTFLAGS=-nvx

1335 # object files that depend on inline template
1336 $(TIL:%=pics/%): $(LIBCBASE)/threads/sparc.il
1337 $(IL:%=pics/%): $(LIBCBASE)/fp/base.il
1338 $(QIL:%=pics/%): $(LIBCDIR)/$(MACH)/fp/__quad.il

1340 # include common libc targets
1341 include $(LIBCDIR)/Makefile.targ

1343 # We need to strip out all CTF and DOF data from the static library
1344 $(LIB_PIC) := DIR = pics
1345 $(LIB_PIC): pics $$(PICS)
1346         $(BUILD.AR)
1347         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1348         $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1349         $(AR) -ts $@ > /dev/null
1350         $(POST_PROCESS_A)

1352 # special cases
1353 $(STRETS:%=pics/%): $(LIBCBASE)/crt/stret.s
1354         $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=%) $(LIBCBASE)/crt/stret.s -o $@
1355         $(POST_PROCESS_O)

1357 $(LIBCBASE)/crt/_rtbootld.s:    $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.
1358         $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S -K pic \
1359             $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1360         $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $@
1361         $(RM) $(LIBCBASE)/crt/_rtld.s

1363 # partially built from C source
1364 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1365         $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $@
1366         $(CTFCONVERT_O)

1368 ASSYMDEP_OBJS=                  \
1369         _lwp_mutex_unlock.o     \
1370         _stack_grow.o           \
1371         asm_subr.o              \
1372         setjmp.o                \
1373         smt_pause.o             \
1374         tls_get_addr.o          \
1375         unwind_frame.o          \
1376         vforkx.o

1378 $(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

1380 $(ASSYMDEP_OBJS:%=pics/%): assym.h
```



```
1382 # assym.h build rules

1384 assym.h := CFLAGS += -g

1386 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1388 genassym: $(GENASSYM_C)
1389         $(NATIVECC) $(NATIVE_CFLAGS) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1389         $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1390                 $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1392 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1394 assym.h: $(OFFSETS) genassym
1395         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1396         ./genassym >>$@

1398 # derived C source and related explicit dependencies
1399 $(LIBCDIR)/port/gen/errlst.c + \
1400 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1401         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1403 pics/errlst.o: $(LIBCDIR)/port/gen/errlst.c

1405 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
   24834 Fri Feb 12 11:52:34 2016
new/usr/src/lib/libc/sparcv9/Makefile.com
6648 illumos build should be explicit about C standards
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2016 Gary Mills
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2015, Joyent, Inc.  All rights reserved.
# Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2013 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Use is subject to license terms.
#

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

# objects are grouped by source directory

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

# local objects
STRETS=

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

DYNOBJS=

FPOBJS=                         \
```



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

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

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

$(__GNUC)FPASMOBJS +=           \
        __quad.o

ATOMICOBJS=                     \
        atomic.o

CHACHAOBJS=                     \
        chacha.o

XATTROBJS=                      \
        xattr_common.o

COMOBJS=                        \
```

```
128         bcmp.o                  \
129         bcopy.o                 \
130         bsearch.o               \
131         bzero.o                 \
132         memccpy.o               \
133         qsort.o                 \
134         strtol.o                \
135         strtoul.o               \
136         strtoll.o               \
137         strtoull.o

139 GENOBJS=                        \
140         _getsp.o                \
141         _xregs_clrptr.o         \
142         abs.o                   \
143         alloca.o                \
144         arc4random.o            \
145         arc4random_uniform.o    \
146         ascii_strcasecmp.o      \
147         byteorder.o             \
148         cuexit.o                \
149         ecvt.o                  \
150         getctxt.o               \
151         lock.o                  \
152         makectxt.o              \
153         memchr.o                \
154         memcmp.o                \
155         new_list.o              \
156         setjmp.o                \
157         siginfolst.o            \
158         siglongjmp.o            \
159         smt_pause.o             \
160         sparc_data.o            \
161         strchr.o                \
162         strcmp.o                \
163         strlcpy.o               \
164         strncmp.o               \
165         strncpy.o               \
166         strnlen.o               \
167         swapctxt.o              \
168         sync_instruction_memory.o

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

174 SYSOBJS64=

176 COMSYSOBJS=                     \
177         __clock_timer.o         \
178         __getloadavg.o          \
179         __rusagesys.o           \
180         __signotify.o           \
181         __sigrt.o               \
182         __time.o                \
183         _lgrp_home_fast.o       \
184         _lgrpsys.o              \
185         _nfssys.o               \
186         _portfs.o               \
187         _pset.o                 \
188         _rpcsys.o               \
189         _sigaction.o            \
190         _so_accept.o            \
191         _so_bind.o              \
192         _so_connect.o           \
193         _so_getpeername.o       \
```



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

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

309 SYSOBJS=                        \
310         __clock_gettime.o       \
311         __getcontext.o          \
312         __uadmin.o              \
313         _lwp_mutex_unlock.o     \
314         _stack_grow.o           \
315         door.o                  \
316         forkx.o                 \
317         forkallx.o              \
318         gettimeofday.o          \
319         sparc_utrap_install.o   \
320         syscall.o               \
321         tls_get_addr.o          \
322         uadmin.o                \
323         umount.o                \
324         uname.o                 \
325         vforkx.o
```



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

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

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

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



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

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



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

610 PORTPRINT_W=                    \
611         doprnt_w.o

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

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

631 PORTSTDIO_W=                    \
632         doscan_w.o

634 PORTSTDIO=                      \
635         __extensions.o          \
636         _endopen.o              \
637         _filbuf.o               \
638         _findbuf.o              \
639         _flsbuf.o               \
640         _wrtchk.o               \
641         clearerr.o              \
642         ctermid.o               \
643         ctermid_r.o             \
644         cuserid.o               \
645         data.o                  \
646         doscan.o                \
647         fdopen.o                \
648         feof.o                  \
649         ferror.o                \
650         fgetc.o                 \
651         fgets.o                 \
652         fileno.o                \
653         flockf.o                \
654         flush.o                 \
655         fopen.o                 \
```

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

691 PORTI18N=                       \
692         getwchar.o              \
693         putwchar.o              \
694         putws.o                 \
695         strtows.o               \
696         wcsnlen.o               \
697         wcstoimax.o             \
698         wcstol.o                \
699         wcstoul.o               \
700         wcswcs.o                \
701         wscat.o                 \
702         wschr.o                 \
703         wscmp.o                 \
704         wscpy.o                 \
705         wscspn.o                \
706         wsdup.o                 \
707         wslen.o                 \
708         wsncat.o                \
709         wsncmp.o                \
710         wsncpy.o                \
711         wspbrk.o                \
712         wsprintf.o              \
713         wsrchr.o                \
714         wsscanf.o               \
715         wsspn.o                 \
716         wstod.o                 \
717         wstok.o                 \
718         wstol.o                 \
719         wstoll.o                \
720         wsxfrm.o                \
721         wmemchr.o               \
```



```
722         wmemcmp.o               \
723         wmemcpy.o               \
724         wmemmove.o              \
725         wmemset.o               \
726         wcsstr.o                \
727         gettext.o               \
728         gettext_real.o          \
729         gettext_util.o          \
730         gettext_gnu.o           \
731         plural_parser.o         \
732         wdresolve.o             \
733         _ctype.o                \
734         isascii.o               \
735         toascii.o

737 PORTI18N_COND=                  \
738         wcstol_longlong.o       \
739         wcstoul_longlong.o

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

```
788         strcasestr.o            \
789         strcoll.o               \
790         strfmon.o               \
791         strftime.o              \
792         strncasecmp.o           \
793         strptime.o              \
794         strxfrm.o               \
795         table.o                 \
796         timelocal.o             \
797         tolower.o               \
798         towlower.o              \
799         ungetwc.o               \
800         utf8.o                  \
801         wcrtomb.o               \
802         wcscasecmp.o            \
803         wcscoll.o               \
804         wcsftime.o              \
805         wcsnrtombs.o            \
806         wcsrtombs.o             \
807         wcstombs.o              \
808         wcswidth.o              \
809         wcsxfrm.o               \
810         wctob.o                 \
811         wctomb.o                \
812         wctrans.o               \
813         wctype.o                \
814         wcwidth.o               \
815         wscol.o

817 AIOOBJS=                       \
818         aio.o                   \
819         aio_alloc.o             \
820         posix_aio.o

822 RTOBJS=                        \
823         clock_timer.o           \
824         mqueue.o                \
825         pos4obj.o               \
826         sched.o                 \
827         sem.o                   \
828         shm.o                   \
829         sigev_thread.o

831 TPOOLOBJS=                     \
832         thread_pool.o

834 THREADSOBJS=                   \
835         alloc.o                 \
836         assfail.o               \
837         cancel.o                \
838         door_calls.o            \
839         tmem.o                  \
840         pthr_attr.o             \
841         pthr_barrier.o          \
842         pthr_cond.o             \
843         pthr_mutex.o            \
844         pthr_rwlock.o           \
845         pthread.o               \
846         rwlock.o                \
847         scalls.o                \
848         sema.o                  \
849         sigaction.o             \
850         spawn.o                 \
851         synch.o                 \
852         tdb_agent.o             \
853         thr.o                   \
```



```
854         thread_interface.o      \
855         tls.o                   \
856         tsd.o

858 THREADSMACHOBJS=               \
859         machdep.o

861 THREADSASMOBJS=                \
862         asm_subr.o

864 UNICODEOBJS=                   \
865         u8_textprep.o           \
866         uconv.o

868 UNWINDMACHOBJS=                \
869         unwind.o

871 UNWINDASMOBJS=                 \
872         unwind_frame.o

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

878 PORTSYS=                        \
879         _autofssys.o            \
880         access.o                \
881         acctctl.o               \
882         bsd_signal.o            \
883         chmod.o                 \
884         chown.o                 \
885         corectl.o               \
886         eventfd.o               \
887         epoll.o                 \
888         exacctsys.o             \
889         execl.o                 \
890         execle.o                \
891         execv.o                 \
892         fcntl.o                 \
893         getpagesizes.o          \
894         getpeerucred.o          \
895         inst_sync.o             \
896         issetugid.o             \
897         label.o                 \
898         link.o                  \
899         lockf.o                 \
900         lwp.o                   \
901         lwp_cond.o              \
902         lwp_rwlock.o            \
903         lwp_sigmask.o           \
904         meminfosys.o            \
905         mkdir.o                 \
906         mknod.o                 \
907         msgsys.o                \
908         nfssys.o                \
909         open.o                  \
910         pgrpsys.o               \
911         posix_sigwait.o         \
912         ppriv.o                 \
913         psetsys.o               \
914         rctlsys.o               \
915         readlink.o              \
916         rename.o                \
917         sbrk.o                  \
918         semsys.o                \
919         set_errno.o             \
```

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

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

946 VALUES= values-Xa.o

948 MOSTOBJS=                       \
949         $(STRETS)               \
950         $(CRTOBJS)              \
951         $(DYNOBJS)              \
952         $(FPOBJS)               \
953         $(FPOBJS64)             \
954         $(FPASMOBJS)            \
955         $(ATOMICOBJS)           \
956         $(CHACHAOBJS)           \
957         $(XATTROBJS)            \
958         $(COMOBJS)              \
959         $(GENOBJS)              \
960         $(PRFOBJS)              \
961         $(PORTFP)               \
962         $(PORTGEN)              \
963         $(PORTGEN64)            \
964         $(PORTI18N)             \
965         $(PORTI18N_COND)        \
966         $(PORTLOCALE)           \
967         $(PORTPRINT)            \
968         $(PORTPRINT_W)          \
969         $(PORTREGEX)            \
970         $(PORTSTDIO)            \
971         $(PORTSTDIO64)          \
972         $(PORTSTDIO_W)          \
973         $(PORTSYS)              \
974         $(PORTSYS64)            \
975         $(AIOOBJS)              \
976         $(RTOBJS)               \
977         $(TPOOLOBJS)            \
978         $(THREADSOBJS)          \
979         $(THREADSMACHOBJS)      \
980         $(THREADSASMOBJS)       \
981         $(UNICODEOBJS)          \
982         $(UNWINDMACHOBJS)       \
983         $(UNWINDASMOBJS)        \
984         $(COMSYSOBJS)           \
985         $(SYSOBJS)              \
```



```
986         $(COMSYSOBJS64)         \
987         $(SYSOBJS64)            \
988         $(VALUES)

990 TRACEOBJS=                      \
991         plockstat.o

993 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o
994 # modules whose source is provided in the $(SRC)/lib/common directory.
995 # This must be done because otherwise the Sun C compiler would insert
996 # its own versions of these modules and those versions contain code
997 # to call out to C++ initialization functions.  Such C++ initialization
998 # functions can call back into libc before thread initialization is
999 # complete and this leads to segmentation violations and other problems.
1000 # Since libc contains no C++ code, linking with the minimal crti.o and
1001 # crtn.o modules is safe and avoids the problems described above.
1002 OBJECTS= $(CRTI) $(MOSTOBJS) $(CRTN)
1003 CRTSRCS= ../../common/sparcv9

1005 # include common library definitions
1006 include $(SRC)/lib/Makefile.lib
1007 include $(SRC)/lib/Makefile.lib.64

1009 # we need to override the default SONAME here because we might
1010 # be building a variant object (still libc.so.1, but different filename)
1011 SONAME = libc.so.1

1013 CFLAGS64 += $(CCVERBOSE)

1015 # This is necessary to avoid problems with calling _ex_unwind().
1016 # We probably don't want any inlining anyway.
1017 CFLAGS64 += -xinline=

1019 CERRWARN += -_gcc=-Wno-parentheses
1020 CERRWARN += -_gcc=-Wno-switch
1021 CERRWARN += -_gcc=-Wno-uninitialized
1022 CERRWARN += -_gcc=-Wno-unused-value
1023 CERRWARN += -_gcc=-Wno-unused-label
1024 CERRWARN += -_gcc=-Wno-unused-variable
1025 CERRWARN += -_gcc=-Wno-type-limits
1026 CERRWARN += -_gcc=-Wno-char-subscripts
1027 CERRWARN += -_gcc=-Wno-clobbered
1028 CERRWARN += -_gcc=-Wno-unused-function
1029 CERRWARN += -_gcc=-Wno-address

1031 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1032 # enables ASSERT() checking in the threads portion of the library.
1033 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1034 THREAD_DEBUG =
1035 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1037 # Make string literals read-only to save memory.
1038 CFLAGS64 += $(XSTRCONST)

1040 ALTPICS= $(TRACEOBJS:%=pics/%)

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

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

1046 sparcv9_C_PICFLAGS= -K PIC
1047 CFLAGS64 +=     $(EXTN_CFLAGS)
1048 CPPFLAGS=       -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1049                 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1050 ASFLAGS=        $(EXTN_ASFLAGS) -K PIC -P -D__STDC__ -D_ASM -D__sparcv9 $(CPPFLA
1051                 $(sparcv9_AS_XARCH)
```

```
1053 # As a favor to the dtrace syscall provider, libc still calls the
1054 # old syscall traps that have been obsoleted by the *at() interfaces.
1055 # Delete this to compile libc using only the new *at() system call traps
1056 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1058 # Inform the run-time linker about libc specialized initialization
1059 RTLDINFO =      -z rtldinfo=tls_rtldinfo
1060 DYNFLAGS +=     $(RTLDINFO)

1062 # Force libc's internal references to be resolved immediately upon loading
1063 # in order to avoid critical region problems.  Since almost all libc symbols
1064 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1065 DYNFLAGS +=     -znow

1067 DYNFLAGS +=     $(EXTN_DYNFLAGS)

1069 BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

1071 # Override this top level flag so the compiler builds in its native
1072 # C99 mode.  This has been enabled to support the complex arithmetic
1073 # added to libc.
1074 C99MODE=        $(C99_ENABLE)

1076 # libc method of building an archive
1077 # The "$(GREP) -v ' L '" part is necessary only until
1078 # lorder is fixed to ignore thread-local variables.
1079 BUILD.AR= $(RM) $@ ; \
1080         $(AR) q $@ '$(LORDER) $(MOSTOBJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1082 # extra files for the clean target
1083 CLEANFILES=                     \
1084         $(LIBCDIR)/port/gen/errlst.c    \
1085         $(LIBCDIR)/port/gen/new_list.c  \
1086         assym.h                 \
1087         genassym                \
1088         pics/crti.o             \
1089         pics/crtn.o             \
1090         $(ALTPICS)

1092 CLOBBERFILES += $(LIB_PIC)

1094 # list of C source for lint
1095 SRCS=                                                   \
1096         $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c)      \
1097         $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c)        \
1098         $(COMOBJS:%.o=$(SRC)/common/util/%.c)           \
1099         $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c)                    \
1100         $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c)                  \
1101         $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c)                \
1102         $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c)            \
1103         $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c)              \
1104         $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c)              \
1105         $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c)              \
1106         $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c)                  \
1107         $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c)                  \
1108         $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c)                    \
1109         $(TPOOLOBJS:%.o=$(LIBCDIR)/port/tpool/%.c)              \
1110         $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c)          \
1111         $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c)   \
1112         $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c)    \
1113         $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c)        \
1114         $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c)                 \
1115         $(FPOBJS64:%.o=$(LIBCBASE)/fp/%.c)              \
1116         $(LIBCBASE)/crt/_ftou.c                         \
1117         $(LIBCBASE)/gen/_xregs_clrptr.c                 \
```



```
1118         $(LIBCBASE)/gen/byteorder.c                     \
1119         $(LIBCBASE)/gen/ecvt.c                          \
1120         $(LIBCBASE)/gen/getctxt.c                       \
1121         $(LIBCBASE)/gen/makectxt.c                      \
1122         $(LIBCBASE)/gen/siginfolst.c                    \
1123         $(LIBCBASE)/gen/siglongjmp.c                    \
1124         $(LIBCBASE)/gen/swapctxt.c

1126 # conditional assignments
1127 $(DYNLIB) := CRTI = crti.o
1128 $(DYNLIB) := CRTN = crtn.o

1130 # Files which need the threads .il inline template
1131 TIL=                            \
1132         aio.o                   \
1133         alloc.o                 \
1134         assfail.o               \
1135         atexit.o                \
1136         atfork.o                \
1137         cancel.o                \
1138         door_calls.o            \
1139         err.o                   \
1140         errno.o                 \
1141         getctxt.o               \
1142         lwp.o                   \
1143         ma.o                    \
1144         machdep.o               \
1145         posix_aio.o             \
1146         pthr_attr.o             \
1147         pthr_barrier.o          \
1148         pthr_cond.o             \
1149         pthr_mutex.o            \
1150         pthr_rwlock.o           \
1151         pthread.o               \
1152         rand.o                  \
1153         rwlock.o                \
1154         scalls.o                \
1155         sched.o                 \
1156         sema.o                  \
1157         sigaction.o             \
1158         sigev_thread.o          \
1159         spawn.o                 \
1160         stack.o                 \
1161         swapctxt.o              \
1162         synch.o                 \
1163         tdb_agent.o             \
1164         thr.o                   \
1165         thread_interface.o      \
1166         thread_pool.o           \
1167         tls.o                   \
1168         tsd.o                   \
1169         unwind.o

1171 $(TIL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/threads/sparcv9.il

1173 # Files in fp, port/fp subdirectories that need base.il inline template
1174 IL=                             \
1175         __flt_decim.o           \
1176         decimal_bin.o

1178 $(IL:%=pics/%) := CFLAGS64 += $(LIBCBASE)/fp/base.il

1180 # Files in fp subdirectory which need __quad.il inline template
1181 QIL=                            \
1182         _Q_add.o                \
1183         _Q_cmp.o                \
```

```
1184         _Q_cmpe.o               \
1185         _Q_div.o                \
1186         _Q_dtoq.o               \
1187         _Q_fcc.o                \
1188         _Q_mul.o                \
1189         _Q_qtod.o               \
1190         _Q_qtoi.o               \
1191         _Q_qtos.o               \
1192         _Q_qtou.o               \
1193         _Q_sqrt.o               \
1194         _Q_stoq.o               \
1195         _Q_sub.o                \
1196         _Qp_qtox.o              \
1197         _Qp_qtoux.o

1199 $(QIL:%=pics/%) := CFLAGS64 += $(LIBCDIR)/$(MACH)/fp/__quad.il
1200 pics/_Qp%.o := CFLAGS64 += -I$(LIBCDIR)/$(MACH)/fp
1201 pics/_Q%.o := sparcv9_COPTFLAG = -xO4 -xchip=ultra

1203 # Files in crt subdirectory which need muldiv64.il inline template
1204 #CIL=   mul64.o divrem64.o
1205 #$(CIL:%=pics/%) := CFLAGS += $(LIBCBASE)/crt/mul64.il

1207 # large-file-aware components that should be built large

1209 #$(COMSYSOBJS64:%=pics/%) := \
1210 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1212 #$(SYSOBJS64:%=pics/%) := \
1213 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1215 #$(PORTGEN64:%=pics/%) := \
1216 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1218 #$(PORTSTDIO64:%=pics/%) := \
1219 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1221 #$(PORTSYS64:%=pics/%) := \
1222 #       CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1224 $(PORTSTDIO_W:%=pics/%) := \
1225         CPPFLAGS += -D_WIDE

1227 $(PORTPRINT_W:%=pics/%) := \
1228         CPPFLAGS += -D_WIDE

1230 $(PORTI18N_COND:%=pics/%) := \
1231         CPPFLAGS += -D_WCS_LONGLONG

1233 pics/arc4random.o :=    CPPFLAGS += -I$(SRC)/common/crypto/chacha

1235 # Files which need extra optimization
1236 pics/getenv.o := sparcv9_COPTFLAG = -xO4

1238 .KEEP_STATE:

1240 all: $(LIBS) $(LIB_PIC)

1242 lint    :=      CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1243 lint    :=      CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1244 lint    :=      LINTFLAGS64 += -mn

1246 lint:
1247         @echo $(LINT.c) ... $(LDLIBS)
1248         @$(LINT.c) $(SRCS) $(LDLIBS)
```



```
1250 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-lc
1251 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1252 $(LINTLIB):= LINTFLAGS64=-nvx -m64

1254 # object files that depend on inline template
1255 $(TIL:%=pics/%): $(LIBCBASE)/threads/sparcv9.il
1256 $(IL:%=pics/%): $(LIBCBASE)/fp/base.il
1257 $(QIL:%=pics/%): $(LIBCDIR)/$(MACH)/fp/__quad.il
1258 #$(CIL:%=pics/%): $(LIBCBASE)/crt/muldiv64.il

1260 # include common libc targets
1261 include $(LIBCDIR)/Makefile.targ

1263 # We need to strip out all CTF and DOF data from the static library
1264 $(LIB_PIC) := DIR = pics
1265 $(LIB_PIC): pics $$(PICS)
1266         $(BUILD.AR)
1267         $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1268         $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1269         $(AR) -ts $@ > /dev/null
1270         $(POST_PROCESS_A)

1272 # special cases
1273 #$(STRETS:%=pics/%): crt/stret.s
1274 #       $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=%) crt/stret.s -o $@
1275 #       $(POST_PROCESS_O)

1277 #crt/_rtbootld.s:       crt/_rtboot.s crt/_rtld.c
1278 #       $(CC) $(CPPFLAGS) -O -S -K pic crt/_rtld.c -o crt/_rtld.s
1279 #       $(CAT) crt/_rtboot.s crt/_rtld.s > $@
1280 #       $(RM) crt/_rtld.s

1282 ASSYMDEP_OBJS=                  \
1283         _lwp_mutex_unlock.o     \
1284         _stack_grow.o           \
1285         asm_subr.o              \
1286         setjmp.o                \
1287         smt_pause.o             \
1288         tls_get_addr.o          \
1289         unwind_frame.o          \
1290         vforkx.o

1292 $(ASSYMDEP_OBJS:%=pics/%)       :=      CPPFLAGS += -I.

1294 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1296 # assym.h build rules

1298 assym.h := CFLAGS64 += -g

1300 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1302 genassym: $(GENASSYM_C)
1303         $(NATIVECC) $(NATIVE_CFLAGS) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1303         $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1304                 $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1306 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1308 assym.h: $(OFFSETS) genassym
1309         $(OFFSETS_CREATE) <$(OFFSETS) >$@
1310         ./genassym >>$@

1312 # derived C source and related explicit dependencies
1313 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1314         cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist
```

```
1316 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c
```

```
**********************************************************
    2917 Fri Feb 12 11:52:34 2016
new/usr/src/lib/libm/Makefile.libm.com
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#

LIBMDIR         = $(SRC)/lib/libm

LIBMSRC         = $(LIBMDIR)/common

CPP_CMD         = $(CC) -E -Xs

ASSUFFIX_sparc  = S
ASSUFFIX_i386   = s
ASSUFFIX        = $(ASSUFFIX_$(MACH))

# With studio C99MODE of neither enabled nor disabled is "no_lib", whereby we
# expect C99-the-language, but don't modify the behaviour of library routines.
# This is VERY IMPORTANT, as -xc99=%all, for instance, would link us with
# C99MODE of neither enabled nor disabled is "no_lib", whereby we expect
# C99-the-language, but don't modify the behaviour of library routines.  This
# is VERY IMPORTANT, as -xc99=%all, for instance, would link us with
# values-xpg6, which would introduce an __xpg6 to our object with the C99
# flags set, causing us to default C99 libm behaviour on, breaking
# compatibility.
#
# We must then, unfortunately, defeat the GNU compiler _defaulting_ to C99, by
# in that case setting it back to gnu89, which _also_ accepts C99 syntax as
# far as is important.
#endif /* ! codereview */
C99MODE         =
CFLAGS          += -_gcc=-std=gnu89
CFLAGS64        += -_gcc=-std=gnu89
#endif /* ! codereview */

M4FLAGS         = -D__STDC__ -DPIC

LDBLDIR_sparc   = Q
LDBLDIR_i386    = LD
LDBLDIR         = $(LDBLDIR_$(MACH))

LM_IL           = $(LIBMDIR)/$(TARGET_ARCH)/src/locallibm.il

CFLAGS          += $(C_PICFLAGS) $(XSTRCONST) $(LM_IL)
CFLAGS64        += $(C_PICFLAGS) $(XSTRCONST) $(LM_IL)
sparc_CFLAGS    += -Wa,-xarch=v8plus

CPPFLAGS        += -I$(LIBMSRC)/C \
                -I$(LIBMSRC)/$(LDBLDIR) -I$(LIBMDIR)/$(TARGET_ARCH)/src

# GCC needs __C99FEATURES__ such that the implementations of isunordered,
# isgreaterequal, islessequal, etc, exist.  This is basically equivalent to
```



```
# providing no -xc99 to Studio, in that it gets us the C99 language features,
# but not values-xpg6, the reason for which is outlined with C99MODE.
# but not values-xpg6, the reason for which is outline with C99MODE.
CFLAGS          += -_gcc=-D__C99FEATURES__
CFLAGS64        += -_gcc=-D__C99FEATURES__

# libm depends on integer overflow characteristics
CFLAGS          += -_gcc=-fno-strict-overflow
CFLAGS64        += -_gcc=-fno-strict-overflow

$(DYNLIB)       := LDLIBS += -lc

$(LINTLIB)      := SRCS = $(LIBMSRC)/$(LINTSRC)

CLEANFILES      += pics/*.s pics/*.S

FPDEF_amd64     = -DARCH_amd64
FPDEF_sparc     = -DCG89 -DARCH_v8plus -DFPADD_TRAPS_INCOMPLETE_ON_NAN
FPDEF_sparcv9   = -DARCH_v9 -DFPADD_TRAPS_INCOMPLETE_ON_NAN
FPDEF           = $(FPDEF_$(TARGET_ARCH))

ASFLAGS         = -P -D_ASM $(FPDEF)

XARCH_sparc     = v8plus
XARCH_sparcv9   = v9
XARCH_i386      = f80387
XARCH_amd64     = amd64
XARCH           = $(XARCH_$(TARGET_ARCH))

ASOPT_sparc     = -xarch=$(XARCH) $(AS_PICFLAGS)
ASOPT_sparcv9   = -xarch=$(XARCH) $(AS_PICFLAGS)
ASOPT_i386      =
ASOPT_amd64     = -xarch=$(XARCH) $(AS_PICFLAGS)
ASOPT           = $(ASOPT_$(TARGET_ARCH))

ASFLAGS         += $(ASOPT)

CPPFLAGS_sparc = -DFPADD_TRAPS_INCOMPLETE_ON_NAN \
        -DFDTOS_TRAPS_INCOMPLETE_IN_FNS_MODE

CPPFLAGS        += $(CPPFLAGS_$(MACH))
ASFLAGS         += $(CPPFLAGS)
```

```
**********************************************************
    3988 Fri Feb 12 11:52:34 2016
new/usr/src/lib/libm/amd64/src/libm_inlines.h
6648 illumos build should be explicit about C standards
**********************************************************
```

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

/*
 * Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
 */
/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2011, Richard Lowe.
 */

/* Functions in this file are duplicated in locallibm.il.  Keep them in sync */

#ifndef _LIBM_INLINES_H
#define _LIBM_INLINES_H

#ifdef __GNUC__

#ifdef __cplusplus
extern "C" {
#endif

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

extern __GNU_INLINE float
extern __inline__ float
__inline_sqrtf(float a)
{
        float ret;

        __asm__ __volatile__("sqrtss %1, %0\n\t" : "=x" (ret) : "x" (a));
        return (ret);
}

extern __GNU_INLINE double
extern __inline__ double
__inline_sqrt(double a)
{
```



```
        double ret;

        __asm__ __volatile__("sqrtsd %1, %0\n\t" : "=x" (ret) : "x" (a));
        return (ret);
}

extern __GNU_INLINE double
extern __inline__ double
__ieee754_sqrt(double a)
{
        return (__inline_sqrt(a));
}

/*
 * 00 - 24 bits
 * 01 - reserved
 * 10 - 53 bits
 * 11 - 64 bits
 */
extern __GNU_INLINE int
extern __inline__ int
__swapRP(int i)
{
        int ret;
        uint16_t cw;

        __asm__ __volatile__("fstcw %0\n\t" : "=m" (cw));

        ret = (cw >> 8) & 0x3;
        cw = (cw & 0xfcff) | ((i & 0x3) << 8);

        __asm__ __volatile__("fldcw %0\n\t" : : "m" (cw));

        return (ret);
}

/*
 * 00 - Round to nearest, with even preferred
 * 01 - Round down
 * 10 - Round up
 * 11 - Chop
 */
extern __GNU_INLINE enum fp_direction_type
extern __inline__ enum fp_direction_type
__swap87RD(enum fp_direction_type i)
{
        int ret;
        uint16_t cw;

        __asm__ __volatile__("fstcw %0\n\t" : "=m" (cw));

        ret = (cw >> 10) & 0x3;
        cw = (cw & 0xf3ff) | ((i & 0x3) << 10);

        __asm__ __volatile__("fldcw %0\n\t" : : "m" (cw));

        return (ret);
}

extern __GNU_INLINE int
extern __inline__ int
abs(int i)
{
        int ret;
        __asm__ __volatile__(
            "movl    %1, %0\n\t"
```

```
122             "negl    %1\n\t"
123             "cmovnsl %1, %0\n\t"
124             : "=r" (ret), "+r" (i)
125             :
126             : "cc");
127         return (ret);
128 }

130 extern __GNU_INLINE double
130 extern __inline__ double
131 copysign(double d1, double d2)
132 {
133         double tmpd;

135         __asm__ __volatile__(
136             "movd %3, %1\n\t"
137             "andpd %1, %0\n\t"
138             "andnpd %2, %1\n\t"
139             "orpd %1, %0\n\t"
140             : "+&x" (d1), "=&x" (tmpd)
141             : "x" (d2), "r" (0x7fffffffffffffff));

143         return (d1);
144 }

146 extern __GNU_INLINE double
146 extern __inline__ double
147 fabs(double d)
148 {
149         double tmp;

151         __asm__ __volatile__(
152             "movd  %2, %1\n\t"
153             "andpd %1, %0"
154             : "+x" (d), "=&x" (tmp)
155             : "r" (0x7fffffffffffffff));

157         return (d);
158 }

160 extern __GNU_INLINE float
160 extern __inline__ float
161 fabsf(float d)
162 {
163         __asm__ __volatile__(
164             "andpd %1, %0"
165             : "+x" (d)
166             : "x" (0x7fffffff));

168         return (d);
169 }

171 extern __GNU_INLINE int
171 extern __inline__ int
172 finite(double d)
173 {
174         long ret = 0x7fffffffffffffff;
175         uint64_t tmp;

177         __asm__ __volatile__(
178             "movq %2, %1\n\t"
179             "andq %1, %0\n\t"
180             "movq $0x7ff0000000000000, %1\n\t"
181             "subq %1, %0\n\t"
182             "shrq $63, %0\n\t"
183             : "+r" (ret), "=r" (tmp)
```



```
184             : "x" (d)
185             : "cc");

187         return (ret);
188 }

190 extern __GNU_INLINE int
190 extern __inline__ int
191 signbit(double d)
192 {
193         long ret;
194         __asm__ __volatile__(
195             "movmskpd %1, %0\n\t"
196             "andq     $1, %0\n\t"
197             : "=r" (ret)
198             : "x" (d)
199             : "cc");
200         return (ret);
201 }

203 extern __GNU_INLINE double
203 extern __inline__ double
204 sqrt(double d)
205 {
206         return (__inline_sqrt(d));
207 }

209 extern __GNU_INLINE float
209 extern __inline__ float
210 sqrtf(float f)
211 {
212         return (__inline_sqrtf(f));
213 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
   12775 Fri Feb 12 11:52:35 2016
new/usr/src/lib/libm/common/m9x/fenv_inlines.h
6648 illumos build should be explicit about C standards
**********************************************************
______unchanged_portion_omitted_

  42 extern __GNU_INLINE void
  42 extern __inline__ void
  43 __fenv_getcwsw(unsigned int *value)
  44 {
  45         union fp_cwsw *u = (union fp_cwsw *)value;

  47         __asm__ __volatile__(
  48             "fstsw %0\n\t"
  49             "fstcw %1\n\t"
  50             : "=m" (u->words.cw), "=m" (u->words.sw));
  51 }

  53 extern __GNU_INLINE void
  53 extern __inline__ void
  54 __fenv_setcwsw(const unsigned int *value)
  55 {
  56         union fp_cwsw cwsw;
  57         short fenv[16];

  59         cwsw.cwsw = *value;

  61         __asm__ __volatile__(
  62             "fstenv %0\n\t"
  63             "movw   %4,%1\n\t"
  64             "movw   %3,%2\n\t"
  65             "fldenv %0\n\t"
  66             "fwait\n\t"
  67             : "=m" (fenv), "=m" (fenv[0]), "=m" (fenv[2])
  68             : "r" (cwsw.words.cw), "r" (cwsw.words.sw)
  69             /* For practical purposes, we clobber the whole FPU */
  70             : "cc", "st", "st(1)", "st(2)", "st(3)", "st(4)", "st(5)",
  71               "st(6)", "st(7)");
  72 }

  74 extern __GNU_INLINE void
  74 extern __inline__ void
  75 __fenv_getmxcsr(unsigned int *value)
  76 {
  77         __asm__ __volatile__("stmxcsr %0" : "=m" (*value));
  78 }

  80 extern __GNU_INLINE void
  80 extern __inline__ void
  81 __fenv_setmxcsr(const unsigned int *value)
  82 {
  83         __asm__ __volatile__("ldmxcsr %0" : : "m" (*value));
  84 }

  86 extern __GNU_INLINE long double
  86 extern __inline__ long double
  87 f2xm1(long double x)
  88 {
  89         long double ret;

  91         __asm__ __volatile__("f2xm1" : "=t" (ret) : "0" (x) : "cc");
  92         return (ret);
  93 }

  95 extern __GNU_INLINE long double
```



```
  95 extern __inline__ long double
  96 fyl2x(long double y, long double x)
  97 {
  98         long double ret;

 100         __asm__ __volatile__("fyl2x"
 101             : "=t" (ret)
 102             : "0" (x), "u" (y)
 103             : "st(1)", "cc");
 104         return (ret);
 105 }

 107 extern __GNU_INLINE long double
 107 extern __inline__ long double
 108 fptan(long double x)
 109 {
 110         /*
 111          * fptan pushes 1.0 then the result on completion, so we want to pop
 112          * the FP stack twice, so we need a dummy value into which to pop it.
 113          */
 114         long double ret;
 115         long double dummy;

 117         __asm__ __volatile__("fptan"
 118             : "=t" (dummy), "=u" (ret)
 119             : "0" (x)
 120             : "cc");
 121         return (ret);
 122 }

 124 extern __GNU_INLINE long double
 124 extern __inline__ long double
 125 fpatan(long double x, long double y)
 126 {
 127         long double ret;

 129         __asm__ __volatile__("fpatan"
 130             : "=t" (ret)
 131             : "0" (y), "u" (x)
 132             : "st(1)", "cc");
 133         return (ret);
 134 }

 136 extern __GNU_INLINE long double
 136 extern __inline__ long double
 137 fxtract(long double x)
 138 {
 139         __asm__ __volatile__("fxtract" : "+t" (x) : : "cc");
 140         return (x);
 141 }

 143 extern __GNU_INLINE long double
 143 extern __inline__ long double
 144 fprem1(long double idend, long double div)
 145 {
 146         __asm__ __volatile__("fprem1" : "+t" (div) : "u" (idend) : "cc");
 147         return (div);
 148 }

 150 extern __GNU_INLINE long double
 150 extern __inline__ long double
 151 fprem(long double idend, long double div)
 152 {
 153         __asm__ __volatile__("fprem" : "+t" (div) : "u" (idend) : "cc");
 154         return (div);
 155 }
```

```
157 extern __GNU_INLINE long double
157 extern __inline__ long double
158 fyl2xp1(long double y, long double x)
159 {
160         long double ret;

162         __asm__ __volatile__("fyl2xp1"
163             : "=t" (ret)
164             : "0" (x), "u" (y)
165             : "st(1)", "cc");
166         return (ret);
167 }

169 extern __GNU_INLINE long double
169 extern __inline__ long double
170 fsqrt(long double x)
171 {
172         __asm__ __volatile__("fsqrt" : "+t" (x) : : "cc");
173         return (x);
174 }

176 extern __GNU_INLINE long double
176 extern __inline__ long double
177 fsincos(long double x)
178 {
179         long double dummy;

181         __asm__ __volatile__("fsincos" : "+t" (x), "=u" (dummy) : : "cc");
182         return (x);
183 }

185 extern __GNU_INLINE long double
185 extern __inline__ long double
186 frndint(long double x)
187 {
188         __asm__ __volatile__("frndint" : "+t" (x) : : "cc");
189         return (x);
190 }

192 extern __GNU_INLINE long double
192 extern __inline__ long double
193 fscale(long double x, long double y)
194 {
195         long double ret;

197         __asm__ __volatile__("fscale" : "=t" (ret) : "0" (y), "u" (x) : "cc");
198         return (ret);
199 }

201 extern __GNU_INLINE long double
201 extern __inline__ long double
202 fsin(long double x)
203 {
204         __asm__ __volatile__("fsin" : "+t" (x) : : "cc");
205         return (x);
206 }

208 extern __GNU_INLINE long double
208 extern __inline__ long double
209 fcos(long double x)
210 {
211         __asm__ __volatile__("fcos" : "+t" (x) : : "cc");
212         return (x);
213 }
```



```
215 extern __GNU_INLINE void
215 extern __inline__ void
216 sse_cmpeqss(float *f1, float *f2, int *i1)
217 {
218         __asm__ __volatile__(
219             "cmpeqss %2, %1\n\t"
220             "movss   %1, %0"
221             : "=m" (*i1), "+x" (*f1)
222             : "x" (*f2)
223             : "cc");
224 }

226 extern __GNU_INLINE void
226 extern __inline__ void
227 sse_cmpltss(float *f1, float *f2, int *i1)
228 {
229         __asm__ __volatile__(
230             "cmpltss %2, %1\n\t"
231             "movss   %1, %0"
232             : "=m" (*i1), "+x" (*f1)
233             : "x" (*f2)
234             : "cc");
235 }

237 extern __GNU_INLINE void
237 extern __inline__ void
238 sse_cmpless(float *f1, float *f2, int *i1)
239 {
240         __asm__ __volatile__(
241             "cmpless %2, %1\n\t"
242             "movss   %1, %0"
243             : "=m" (*i1), "+x" (*f1)
244             : "x" (*f2)
245             : "cc");
246 }

248 extern __GNU_INLINE void
248 extern __inline__ void
249 sse_cmpunordss(float *f1, float *f2, int *i1)
250 {
251         __asm__ __volatile__(
252             "cmpunordss %2, %1\n\t"
253             "movss      %1, %0"
254             : "=m" (*i1), "+x" (*f1)
255             : "x" (*f2)
256             : "cc");
257 }

259 extern __GNU_INLINE void
259 extern __inline__ void
260 sse_minss(float *f1, float *f2, float *f3)
261 {
262         __asm__ __volatile__(
263             "minss %2, %1\n\t"
264             "movss %1, %0"
265             : "=m" (*f3), "+x" (*f1)
266             : "x" (*f2));
267 }

269 extern __GNU_INLINE void
269 extern __inline__ void
270 sse_maxss(float *f1, float *f2, float *f3)
271 {
272         __asm__ __volatile__(
273             "maxss %2, %1\n\t"
274             "movss %1, %0"
```

```
275             : "=m" (*f3), "+x" (*f1)
276             : "x" (*f2));
277 }

279 extern __GNU_INLINE void
279 extern __inline__ void
280 sse_addss(float *f1, float *f2, float *f3)
281 {
282         __asm__ __volatile__(
283             "addss %2, %1\n\t"
284             "movss %1, %0"
285             : "=m" (*f3), "+x" (*f1)
286             : "x" (*f2));
287 }

289 extern __GNU_INLINE void
289 extern __inline__ void
290 sse_subss(float *f1, float *f2, float *f3)
291 {
292         __asm__ __volatile__(
293             "subss %2, %1\n\t"
294             "movss %1, %0"
295             : "=m" (*f3), "+x" (*f1)
296             : "x" (*f2));
297 }

299 extern __GNU_INLINE void
299 extern __inline__ void
300 sse_mulss(float *f1, float *f2, float *f3)
301 {
302         __asm__ __volatile__(
303             "mulss %2, %1\n\t"
304             "movss %1, %0"
305             : "=m" (*f3), "+x" (*f1)
306             : "x" (*f2));
307 }

309 extern __GNU_INLINE void
309 extern __inline__ void
310 sse_divss(float *f1, float *f2, float *f3)
311 {
312         __asm__ __volatile__(
313             "divss %2, %1\n\t"
314             "movss %1, %0"
315             : "=m" (*f3), "+x" (*f1)
316             : "x" (*f2));
317 }

319 extern __GNU_INLINE void
319 extern __inline__ void
320 sse_sqrtss(float *f1, float *f2)
321 {
322         double tmp;

324         __asm__ __volatile__(
325             "sqrtss %2, %1\n\t"
326             "movss  %1, %0"
327             : "=m" (*f2), "=x" (tmp)
328             : "m" (*f1));
329 }

331 extern __GNU_INLINE void
331 extern __inline__ void
332 sse_ucomiss(float *f1, float *f2)
333 {
334         __asm__ __volatile__("ucomiss %1, %0" : : "x" (*f1), "x" (*f2));
```



```
336 }

338 extern __GNU_INLINE void
338 extern __inline__ void
339 sse_comiss(float *f1, float *f2)
340 {
341         __asm__ __volatile__("comiss %1, %0" : : "x" (*f1), "x" (*f2));
342 }

344 extern __GNU_INLINE void
344 extern __inline__ void
345 sse_cvtss2sd(float *f1, double *d1)
346 {
347         double tmp;

349         __asm__ __volatile__(
350             "cvtss2sd %2, %1\n\t"
351             "movsd    %1, %0"
352             : "=m" (*d1), "=x" (tmp)
353             : "m" (*f1));
354 }

356 extern __GNU_INLINE void
356 extern __inline__ void
357 sse_cvtsi2ss(int *i1, float *f1)
358 {
359         double tmp;

361         __asm__ __volatile__(
362             "cvtsi2ss %2, %1\n\t"
363             "movss    %1, %0"
364             : "=m" (*f1), "=x" (tmp)
365             : "m" (*i1));
366 }

368 extern __GNU_INLINE void
368 extern __inline__ void
369 sse_cvttss2si(float *f1, int *i1)
370 {
371         int tmp;

373         __asm__ __volatile__(
374             "cvttss2si %2, %1\n\t"
375             "movl      %1, %0"
376             : "=m" (*i1), "=r" (tmp)
377             : "m" (*f1));
378 }

380 extern __GNU_INLINE void
380 extern __inline__ void
381 sse_cvtss2si(float *f1, int *i1)
382 {
383         int tmp;

385         __asm__ __volatile__(
386             "cvtss2si %2, %1\n\t"
387             "movl     %1, %0"
388             : "=m" (*i1), "=r" (tmp)
389             : "m" (*f1));
390 }

392 #if defined(__amd64)
393 extern __GNU_INLINE void
393 extern __inline__ void
394 sse_cvtsi2ssq(long long *ll1, float *f1)
```

```
395 {
396         double tmp;

398         __asm__ __volatile__(
399             "cvtsi2ssq %2, %1\n\t"
400             "movss     %1, %0"
401             : "=m" (*f1), "=x" (tmp)
402             : "m" (*ll1));
403 }

405 extern __GNU_INLINE void
405 extern __inline__ void
406 sse_cvttss2siq(float *f1, long long *ll1)
407 {
408         uint64_t tmp;

410         __asm__ __volatile__(
411             "cvttss2siq %2, %1\n\t"
412             "movq       %1, %0"
413             : "=m" (*ll1), "=r" (tmp)
414             : "m" (*f1));
415 }

417 extern __GNU_INLINE void
417 extern __inline__ void
418 sse_cvtss2siq(float *f1, long long *ll1)
419 {
420         uint64_t tmp;

422         __asm__ __volatile__(
423             "cvtss2siq %2, %1\n\t"
424             "movq      %1, %0"
425             : "=m" (*ll1), "=r" (tmp)
426             : "m" (*f1));
427 }

429 #endif

431 extern __GNU_INLINE void
431 extern __inline__ void
432 sse_cmpeqsd(double *d1, double *d2, long long *ll1)
433 {
434         __asm__ __volatile__(
435             "cmpeqsd %2,%1\n\t"
436             "movsd   %1,%0"
437             : "=m" (*ll1), "+x" (*d1)
438             : "x" (*d2));
439 }

441 extern __GNU_INLINE void
441 extern __inline__ void
442 sse_cmpltsd(double *d1, double *d2, long long *ll1)
443 {
444         __asm__ __volatile__(
445             "cmpltsd %2,%1\n\t"
446             "movsd   %1,%0"
447             : "=m" (*ll1), "+x" (*d1)
448             : "x" (*d2));
449 }

451 extern __GNU_INLINE void
451 extern __inline__ void
452 sse_cmplesd(double *d1, double *d2, long long *ll1)
453 {
454         __asm__ __volatile__(
455             "cmplesd %2,%1\n\t"
```



```
456             "movsd   %1,%0"
457             : "=m" (*ll1), "+x" (*d1)
458             : "x" (*d2));
459 }

461 extern __GNU_INLINE void
461 extern __inline__ void
462 sse_cmpunordsd(double *d1, double *d2, long long *ll1)
463 {
464         __asm__ __volatile__(
465             "cmpunordsd %2,%1\n\t"
466             "movsd      %1,%0"
467             : "=m" (*ll1), "+x" (*d1)
468             : "x" (*d2));
469 }


472 extern __GNU_INLINE void
472 extern __inline__ void
473 sse_minsd(double *d1, double *d2, double *d3)
474 {
475         __asm__ __volatile__(
476             "minsd %2,%1\n\t"
477             "movsd %1,%0"
478             : "=m" (*d3), "+x" (*d1)
479             : "x" (*d2));
480 }

482 extern __GNU_INLINE void
482 extern __inline__ void
483 sse_maxsd(double *d1, double *d2, double *d3)
484 {
485         __asm__ __volatile__(
486             "maxsd %2,%1\n\t"
487             "movsd %1,%0"
488             : "=m" (*d3), "+x" (*d1)
489             : "x" (*d2));
490 }

492 extern __GNU_INLINE void
492 extern __inline__ void
493 sse_addsd(double *d1, double *d2, double *d3)
494 {
495         __asm__ __volatile__(
496             "addsd %2,%1\n\t"
497             "movsd %1,%0"
498             : "=m" (*d3), "+x" (*d1)
499             : "x" (*d2));
500 }

502 extern __GNU_INLINE void
502 extern __inline__ void
503 sse_subsd(double *d1, double *d2, double *d3)
504 {
505         __asm__ __volatile__(
506             "subsd %2,%1\n\t"
507             "movsd %1,%0"
508             : "=m" (*d3), "+x" (*d1)
509             : "x" (*d2));
510 }

512 extern __GNU_INLINE void
512 extern __inline__ void
513 sse_mulsd(double *d1, double *d2, double *d3)
514 {
515         __asm__ __volatile__(
```

```
516             "mulsd %2,%1\n\t"
517             "movsd %1,%0"
518             : "=m" (*d3), "+x" (*d1)
519             : "x" (*d2));
520 }

522 extern __GNU_INLINE void
522 extern __inline__ void
523 sse_divsd(double *d1, double *d2, double *d3)
524 {
525         __asm__ __volatile__(
526             "divsd %2,%1\n\t"
527             "movsd %1,%0"
528             : "=m" (*d3), "+x" (*d1)
529             : "x" (*d2));
530 }

532 extern __GNU_INLINE void
532 extern __inline__ void
533 sse_sqrtsd(double *d1, double *d2)
534 {
535         double tmp;

537         __asm__ __volatile__(
538             "sqrtsd %2, %1\n\t"
539             "movsd %1, %0"
540             : "=m" (*d2), "=x" (tmp)
541             : "m" (*d1));
542 }

544 extern __GNU_INLINE void
544 extern __inline__ void
545 sse_ucomisd(double *d1, double *d2)
546 {
547         __asm__ __volatile__("ucomisd %1, %0" : : "x" (*d1), "x" (*d2));
548 }

550 extern __GNU_INLINE void
550 extern __inline__ void
551 sse_comisd(double *d1, double *d2)
552 {
553         __asm__ __volatile__("comisd %1, %0" : : "x" (*d1), "x" (*d2));
554 }

556 extern __GNU_INLINE void
556 extern __inline__ void
557 sse_cvtsd2ss(double *d1, float *f1)
558 {
559         double tmp;

561         __asm__ __volatile__(
562             "cvtsd2ss %2,%1\n\t"
563             "movss    %1,%0"
564             : "=m" (*f1), "=x" (tmp)
565             : "m" (*d1));
566 }

568 extern __GNU_INLINE void
568 extern __inline__ void
569 sse_cvtsi2sd(int *i1, double *d1)
570 {
571         double tmp;
572         __asm__ __volatile__(
573             "cvtsi2sd %2,%1\n\t"
574             "movsd    %1,%0"
575             : "=m" (*d1), "=x" (tmp)
```



```
576             : "m" (*i1));
577 }

579 extern __GNU_INLINE void
579 extern __inline__ void
580 sse_cvttsd2si(double *d1, int *i1)
581 {
582         int tmp;

584         __asm__ __volatile__(
585             "cvttsd2si %2,%1\n\t"
586             "movl      %1,%0"
587             : "=m" (*i1), "=r" (tmp)
588             : "m" (*d1));
589 }

591 extern __GNU_INLINE void
591 extern __inline__ void
592 sse_cvtsd2si(double *d1, int *i1)
593 {
594         int tmp;

596         __asm__ __volatile__(
597             "cvtsd2si %2,%1\n\t"
598             "movl     %1,%0"
599             : "=m" (*i1), "=r" (tmp)
600             : "m" (*d1));
601 }

603 #if defined(__amd64)
604 extern __GNU_INLINE void
604 extern __inline__ void
605 sse_cvtsi2sdq(long long *ll1, double *d1)
606 {
607         double tmp;

609         __asm__ __volatile__(
610             "cvtsi2sdq %2,%1\n\t"
611             "movsd     %1,%0"
612             : "=m" (*d1), "=x" (tmp)
613             : "m" (*ll1));
614 }

616 extern __GNU_INLINE void
616 extern __inline__ void
617 sse_cvttsd2siq(double *d1, long long *ll1)
618 {
619         uint64_t tmp;

621         __asm__ __volatile__(
622             "cvttsd2siq %2,%1\n\t"
623             "movq       %1,%0"
624             : "=m" (*ll1), "=r" (tmp)
625             : "m" (*d1));
626 }

628 extern __GNU_INLINE void
628 extern __inline__ void
629 sse_cvtsd2siq(double *d1, long long *ll1)
630 {
631         uint64_t tmp;

633         __asm__ __volatile__(
634             "cvtsd2siq %2,%1\n\t"
635             "movq      %1,%0"
636             : "=m" (*ll1), "=r" (tmp)
```

```
637             : "m" (*d1));
638 }
639 #endif

641 #elif defined(__sparc)
642 extern __GNU_INLINE void
642 extern __inline__ void
643 __fenv_getfsr(unsigned long *l)
644 {
645         __asm__ __volatile__(
646 #if defined(__sparcv9)
647             "stx %%fsr,%0\n\t"
648 #else
649             "st  %%fsr,%0\n\t"
650 #endif
651             : "=m" (*l));
652 }

654 extern __GNU_INLINE void
654 extern __inline__ void
655 __fenv_setfsr(const unsigned long *l)
656 {
657         __asm__ __volatile__(
658 #if defined(__sparcv9)
659             "ldx %0,%%fsr\n\t"
660 #else
661             "ld %0,%%fsr\n\t"
662 #endif
663             : : "m" (*l) : "cc");
664 }

666 extern __GNU_INLINE void
666 extern __inline__ void
667 __fenv_getfsr32(unsigned int *l)
668 {
669         __asm__ __volatile__("st %%fsr,%0\n\t" : "=m" (*l));
670 }

672 extern __GNU_INLINE void
672 extern __inline__ void
673 __fenv_setfsr32(const unsigned int *l)
674 {
675         __asm__ __volatile__("ld %0,%%fsr\n\t" : : "m" (*l));
676 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    7094 Fri Feb 12 11:52:35 2016
new/usr/src/lib/libm/i386/src/libm_inlines.h
6648 illumos build should be explicit about C standards
**********************************************************
```

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

/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2011, Richard Lowe
 */

/* Functions in this file are duplicated in locallibm.il.  Keep them in sync */

#ifndef _LIBM_INLINES_H
#define _LIBM_INLINES_H

#ifdef __GNUC__

#ifdef __cplusplus
extern "C" {
#endif

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

#define _LO_WORD(x)     ((uint32_t *)&x)[0]
#define _HI_WORD(x)     ((uint32_t *)&x)[1]
#define _HIER_WORD(x)   ((uint32_t *)&x)[2]

extern __GNU_INLINE double
extern __inline__ double
__inline_sqrt(double a)
{
        double ret;

        __asm__ __volatile__("fsqrt\n\t" : "=t" (ret) : "0" (a) : "cc");
        return (ret);
}

extern __GNU_INLINE double
extern __inline__ double
__ieee754_sqrt(double a)
```



```
{
        return (__inline_sqrt(a));
}

extern __GNU_INLINE float
extern __inline__ float
__inline_sqrtf(float a)
{
        float ret;

        __asm__ __volatile__("fsqrt\n\t" : "=t" (ret) : "0" (a) : "cc");
        return (ret);
}

extern __GNU_INLINE double
extern __inline__ double
__inline_rint(double a)
{
        __asm__ __volatile__(
            "andl $0x7fffffff,%1\n\t"
            "cmpl $0x43300000,%1\n\t"
            "jae  1f\n\t"
            "frndint\n\t"
            "1: fwait\n\t"
            : "+t" (a), "+&r" (_HI_WORD(a))
            :
            : "cc");

        return (a);
}

/*
 * 00 - 24 bits
 * 01 - reserved
 * 10 - 53 bits
 * 11 - 64 bits
 */
extern __GNU_INLINE int
extern __inline__ int
__swapRP(int i)
{
        int ret;
        uint16_t cw;

        __asm__ __volatile__("fstcw %0\n\t" : "=m" (cw));

        ret = (cw >> 8) & 0x3;
        cw = (cw & 0xfcff) | ((i & 0x3) << 8);

        __asm__ __volatile__("fldcw %0\n\t" : : "m" (cw));

        return (ret);
}

/*
 * 00 - Round to nearest, with even preferred
 * 01 - Round down
 * 10 - Round up
 * 11 - Chop
 */
extern __GNU_INLINE enum fp_direction_type
extern __inline__ enum fp_direction_type
__swap87RD(enum fp_direction_type i)
{
        int ret;
        uint16_t cw;
```

```
123         __asm__ __volatile__("fstcw %0\n\t" : "=m" (cw));

125         ret = (cw >> 10) & 0x3;
126         cw = (cw & 0xf3ff) | ((i & 0x3) << 10);

128         __asm__ __volatile__("fldcw %0\n\t" : : "m" (cw));

130         return (ret);
131 }

133 extern __GNU_INLINE double
133 extern __inline__ double
134 ceil(double d)
135 {
136         /*
137          * Let's set a Rounding Control (RC) bits from x87 FPU Control Word
138          * to fp_positive and save old bits in rd.
139          */
140         short rd = __swap87RD(fp_positive);

142         /*
143          * The FRNDINT instruction returns a floating-point value that is the
144          * integral value closest to the source value in the direction of the
145          * rounding mode specified in the RC field of the x87 FPU control word.
146          *
147          * Rounds the source value in the ST(0) register to the nearest
148          * integral value, depending on the current rounding mode
149          * (setting of the RC field of the FPU control word),
150          * and stores the result in ST(0).
151          */
152         __asm__ __volatile__("frndint" : "+t" (d) : : "cc");

154         /* restore old RC bits */
155         __swap87RD(rd);

157         return (d);
158 }

160 extern __GNU_INLINE double
160 extern __inline__ double
161 copysign(double d1, double d2)
162 {
163         __asm__ __volatile__(
164             "andl $0x7fffffff,%0\n\t"   /* %0 <-- hi_32(abs(d)) */
165             "andl $0x80000000,%1\n\t"   /* %1[31] <-- sign_bit(d2) */
166             "orl  %1,%0\n\t"            /* %0 <-- hi_32(copysign(x,y)) */
167             : "+&r" (_HI_WORD(d1)), "+r" (_HI_WORD(d2))
168             :
169             : "cc");

171         return (d1);
172 }

174 extern __GNU_INLINE double
174 extern __inline__ double
175 fabs(double d)
176 {
177         __asm__ __volatile__("fabs\n\t" : "+t" (d) : : "cc");
178         return (d);
179 }

181 extern __GNU_INLINE float
181 extern __inline__ float
182 fabsf(float d)
183 {
```



```
184         __asm__ __volatile__("fabs\n\t" : "+t" (d) : : "cc");
185         return (d);
186 }

188 extern __GNU_INLINE long double
188 extern __inline__ long double
189 fabsl(long double d)
190 {
191         __asm__ __volatile__("fabs\n\t" : "+t" (d) : : "cc");
192         return (d);
193 }

195 extern __GNU_INLINE int
195 extern __inline__ int
196 finite(double d)
197 {
198         int ret = _HI_WORD(d);

200         __asm__ __volatile__(
201             "notl %0\n\t"
202             "andl $0x7ff00000,%0\n\t"
203             "negl %0\n\t"
204             "shrl $31,%0\n\t"
205             : "+r" (ret)
206             :
207             : "cc");
208         return (ret);
209 }

211 extern __GNU_INLINE double
211 extern __inline__ double
212 floor(double d)
213 {
214         short rd = __swap87RD(fp_negative);

216         __asm__ __volatile__("frndint" : "+t" (d), "+r" (rd) : : "cc");
217         __swap87RD(rd);

219         return (d);
220 }

222 /*
223  *      branchless __isnan
224  *      ((0x7ff00000-[((lx|-lx)>>31)&1]|ahx)>>31)&1 = 1 iff x is NaN
225  */
226 extern __GNU_INLINE int
226 extern __inline__ int
227 isnan(double d)
228 {
229         int ret;

231         __asm__ __volatile__(
232             "movl %1,%%ecx\n\t"
233             "negl %%ecx\n\t"                    /* ecx <-- -lo_32(x) */
234             "orl  %%ecx,%1\n\t"
235             "shrl $31,%1\n\t"                   /* 1 iff lx != 0 */
236             "andl $0x7fffffff,%2\n\t"   /* ecx <-- hi_32(abs(x)) */
237             "orl  %2,%1\n\t"
238             "subl $0x7ff00000,%1\n\t"
239             "negl %1\n\t"
240             "shrl $31,%1\n\t"
241             : "=r" (ret)
242             : "0" (_HI_WORD(d)), "r" (_LO_WORD(d))
243             : "ecx");

245         return (ret);
```

```
246 }

248 extern __GNU_INLINE int
248 extern __inline__ int
249 isnanf(float f)
250 {
251         __asm__ __volatile__(
252             "andl $0x7fffffff,%0\n\t"
253             "negl %0\n\t"
254             "addl $0x7f800000,%0\n\t"
255             "shrl $31,%0\n\t"
256             : "+r" (f)
257             :
258             : "cc");

260         return (f);
261 }

263 extern __GNU_INLINE double
263 extern __inline__ double
264 rint(double a) {
265     return (__inline_rint(a));
266 }

268 extern __GNU_INLINE double
268 extern __inline__ double
269 scalbn(double d, int n)
270 {
271         double dummy;

273         __asm__ __volatile__(
274             "fildl %2\n\t"      /* Convert N to extended */
275             "fxch\n\t"
276             "fscale\n\t"
277             : "+t" (d), "=u" (dummy)
278             : "m" (n)
279             : "cc");

281         return (d);
282 }

284 extern __GNU_INLINE int
284 extern __inline__ int
285 signbit(double d)
286 {
287         return (_HI_WORD(d) >> 31);
288 }

290 extern __GNU_INLINE int
290 extern __inline__ int
291 signbitf(float f)
292 {
293         return ((*(uint32_t *)&f) >> 31);
294 }

296 extern __GNU_INLINE double
296 extern __inline__ double
297 sqrt(double d)
298 {
299         return (__inline_sqrt(d));
300 }

302 extern __GNU_INLINE float
302 extern __inline__ float
303 sqrtf(float f)
304 {
```



```
305         return (__inline_sqrtf(f));
306 }

308 extern __GNU_INLINE long double
308 extern __inline__ long double
309 sqrtl(long double ld)
310 {
311         __asm__ __volatile__("fsqrt" : "+t" (ld) : : "cc");
312         return (ld);
313 }

315 extern __GNU_INLINE int
315 extern __inline__ int
316 isnanl(long double ld)
317 {
318         int ret = _HIER_WORD(ld);

320         __asm__ __volatile__(
321             "andl  $0x00007fff,%0\n\t"
322             "jz    1f\n\t"              /* jump if exp is all 0 */
323             "xorl  $0x00007fff,%0\n\t"
324             "jz    2f\n\t"              /* jump if exp is all 1 */
325             "testl $0x80000000,%1\n\t"
326             "jz    3f\n\t"              /* jump if leading bit is 0 */
327             "xorl  %0,%0\n\t"
328             "jmp   1f\n\t"
329             "2:\n\t"                    /* note that %0 = 0 from before */
330             "cmpl  $0x80000000,%1\n\t"  /* what is first half of significand? */
331             "jnz   3f\n\t"              /* jump if not equal to 0x80000000 */
332             "testl $0xffffffff,%2\n\t"  /* is second half of significand 0? */
333             "jnz   3f\n\t"              /* jump if not equal to 0 */
334             "jmp   1f\n\t"
335             "3:\n\t"
336             "movl  $1,%0\n\t"
337             "1:\n\t"
338             : "+&r" (ret)
339             : "r" (_HI_WORD(ld)), "r" (_LO_WORD(ld))
340             : "cc");

342         return (ret);
343 }
```

*_____unchanged_portion_omitted_*

```
**********************************************************
    7057 Fri Feb 12 11:52:36 2016
new/usr/src/lib/libm/sparc/src/libm_inlines.h
6648 illumos build should be explicit about C standards
**********************************************************
```

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

/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2011, Richard Lowe.
 */

/* Functions in this file are duplicated in locallibm.il.  Keep them in sync */

#ifndef _LIBM_INLINES_H
#define _LIBM_INLINES_H

#ifdef __GNUC__

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

#ifdef __cplusplus
extern "C" {
#endif

extern __GNU_INLINE double
extern __inline__ double
__inline_sqrt(double d)
{
        double ret;

        __asm__ __volatile__("fsqrtd %1,%0\n\t" : "=e" (ret) : "e" (d));
        return (ret);
}

extern __GNU_INLINE float
extern __inline__ float
__inline_sqrtf(float f)
{
        float ret;

        __asm__ __volatile__("fsqrts %1,%0\n\t" : "=f" (ret) : "f" (f));
```



```
        return (ret);
}

extern __GNU_INLINE enum fp_class_type
extern __inline__ enum fp_class_type
fp_classf(float f)
{
        enum fp_class_type ret;
        uint32_t tmp;

        /* XXX: Separate input and output */
        __asm__ __volatile__(
            "sethi  %%hi(0x80000000),%1\n\t"
            "andncc %2,%1,%0\n\t"
            "bne    1f\n\t"
            "nop\n\t"
            "mov    0,%0\n\t"
            "ba 2f\n\t"         /* x is 0 */
            "nop\n\t"
            "1:\n\t"
            "sethi  %%hi(0x7f800000),%1\n\t"
            "andcc  %0,%1,%%g0\n\t"
            "bne    1f\n\t"
            "nop\n\t"
            "mov    1,%0\n\t"
            "ba     2f\n\t"     /* x is subnormal */
            "nop\n\t"
            "1:\n\t"
            "cmp    %0,%1\n\t"
            "bge    1f\n\t"
            "nop\n\t"
            "mov    2,%0\n\t"
            "ba     2f\n\t"     /* x is normal */
            "nop\n\t"
            "1:\n\t"
            "bg     1f\n\t"
            "nop\n\t"
            "mov    3,%0\n\t"
            "ba     2f\n\t"     /* x is __infinity */
            "nop\n\t"
            "1:\n\t"
            "sethi  %%hi(0x00400000),%1\n\t"
            "andcc  %0,%1,%%g0\n\t"
            "mov    4,%0\n\t"   /* x is quiet NaN */
            "bne    2f\n\t"
            "nop\n\t"
            "mov    5,%0\n\t"   /* x is signaling NaN */
            "2:\n\t"
            : "=r" (ret), "=&r" (tmp)
            : "r" (f)
            : "cc");
        return (ret);
}

#define _HI_WORD(x)     ((uint32_t *)&x)[0]
#define _LO_WORD(x)     ((uint32_t *)&x)[1]

extern __GNU_INLINE enum fp_class_type
extern __inline__ enum fp_class_type
fp_class(double d)
{
        enum fp_class_type ret;
        uint32_t tmp;

        __asm__ __volatile__(
            "sethi %%hi(0x80000000),%1\n\t"     /* %1 gets 80000000 */
```

```
124             "andn  %2,%1,%0\n\t"                /* %2-%0 gets abs(x) */
125             "orcc  %0,%3,%%g0\n\t"              /* set cc as x is zero/nonzero *
126             "bne   1f\n\t"                      /* branch if x is nonzero */
127             "nop\n\t"
128             "mov   0,%0\n\t"
129             "ba    2f\n\t"                      /* x is 0 */
130             "nop\n\t"
131             "1:\n\t"
132             "sethi %%hi(0x7ff00000),%1\n\t"     /* %1 gets 7ff00000 */
133             "andcc %0,%1,%%g0\n\t"              /* cc set by __exp field of x */
134             "bne   1f\n\t"                      /* branch if normal or max __exp
135             "nop\n\t"
136             "mov   1,%0\n\t"
137             "ba    2f\n\t"                      /* x is subnormal */
138             "nop\n\t"
139             "1:\n\t"
140             "cmp   %0,%1\n\t"
141             "bge   1f\n\t"                      /* branch if x is max __exp */
142             "nop\n\t"
143             "mov   2,%0\n\t"
144             "ba    2f\n\t"                      /* x is normal */
145             "nop\n\t"
146             "1:\n\t"
147             "andn  %0,%1,%0\n\t"                /* o0 gets msw __significand fie
148             "orcc  %0,%3,%%g0\n\t"              /* set cc by OR __significand */
149             "bne   1f\n\t"                      /* Branch if __nan */
150             "nop\n\t"
151             "mov   3,%0\n\t"
152             "ba    2f\n\t"                      /* x is __infinity */
153             "nop\n\t"
154             "1:\n\t"
155             "sethi %%hi(0x00080000),%1\n\t"
156             "andcc %0,%1,%%g0\n\t"              /* set cc by quiet/sig bit */
157             "be    1f\n\t"                      /* Branch if signaling */
158             "nop\n\t"
159             "mov   4,%0\n\t"                    /* x is quiet NaN */
160             "ba    2f\n\t"
161             "nop\n\t"
162             "1:\n\t"
163             "mov   5,%0\n\t"                    /* x is signaling NaN */
164             "2:\n\t"
165             : "=&r" (ret), "=&r" (tmp)
166             : "r" (_HI_WORD(d)), "r" (_LO_WORD(d))
167             : "cc");

169         return (ret);
170 }

172 extern __GNU_INLINE int
172 extern __inline__ int
173 __swapEX(int i)
174 {
175         int ret;
176         uint32_t fsr;
177         uint32_t tmp1, tmp2;

179         __asm__ __volatile__(
180             "and  %4,0x1f,%2\n\t" /* tmp1 = %2 = %o1 */
181             "sll  %2,5,%2\n\t"  /* shift input to aexc bit location */
182             ".volatile\n\t"
183             "st   %%fsr,%1\n\t"
184             "ld   %1,%0\n\t"    /* %0 = fsr */
185             "andn %0,0x3e0,%3\n\t" /* tmp2 = %3 = %o2 */
186             "or   %2,%3,%2\n\t" /* %2 = new fsr */
187             "st   %2,%1\n\t"
188             "ld   %1,%%fsr\n\t"
```



```
189             "srl  %0,5,%0\n\t"
190             "and  %0,0x1f,%0\n\t" /* %0 = ret = %o0 */
191             ".nonvolatile\n\t"
192             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2)
193             : "r" (i)
194             : "cc");

196         return (ret);
197 }

199 /*
200  * On the SPARC, __swapRP is a no-op; always return 0 for backward
201  * compatibility
202  */
203 /* ARGSUSED */
204 extern __GNU_INLINE enum fp_precision_type
204 extern __inline__ enum fp_precision_type
205 __swapRP(enum fp_precision_type i)
206 {
207         return (0);
208 }

210 extern __GNU_INLINE enum fp_direction_type
210 extern __inline__ enum fp_direction_type
211 __swapRD(enum fp_direction_type d)
212 {
213         enum fp_direction_type ret;
214         uint32_t fsr;
215         uint32_t tmp1, tmp2, tmp3;

217         __asm__ __volatile__(
218             "and  %5,0x3,%0\n\t"
219             "sll  %0,30,%2\n\t"         /* shift input to RD bit location */
220             ".volatile\n\t"
221             "st   %%fsr,%1\n\t"
222             "ld   %1,%0\n\t"            /* %0 = fsr */
223             "set  0xc0000000,%4\n\t"    /* mask of rounding direction bits */
224             "andn %0,%4,%3\n\t"
225             "or   %2,%3,%2\n\t"         /* %2 = new fsr */
226             "st   %2,%1\n\t"
227             "ld   %1,%%fsr\n\t"
228             "srl  %0,30,%0\n\t"
229             "and  %0,0x3,%0\n\t"
230             ".nonvolatile\n\t"
231             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2), "=r" (tmp3)
232             : "r" (d)
233             : "cc");

235         return (ret);
236 }

238 extern __GNU_INLINE int
238 extern __inline__ int
239 __swapTE(int i)
240 {
241         int ret;
242         uint32_t fsr, tmp1, tmp2;

244         __asm__ __volatile__(
245             "and  %4,0x1f,%0\n\t"
246             "sll  %0,23,%2\n\t"         /* shift input to TEM bit location */
247             ".volatile\n\t"
248             "st   %%fsr,%1\n\t"
249             "ld   %1,%0\n\t"            /* %0 = fsr */
250             "set  0x0f800000,%3\n\t"    /* mask of TEM (Trap Enable Mode bits) *
251             "andn %0,%3,%3\n\t"
```

```
252             "or   %2,%3,%2\n\t"         /* %2 = new fsr */
253             "st   %2,%1\n\t"
254             "ld   %1,%%fsr\n\t"
255             "srl  %0,23,%0\n\t"
256             "and  %0,0x1f,%0\n\t"
257             ".nonvolatile\n\t"
258             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2)
259             : "r" (i)
260             : "cc");

262         return (ret);
263 }

265 extern __GNU_INLINE double
265 extern __inline__ double
266 sqrt(double d)
267 {
268         return (__inline_sqrt(d));
269 }

271 extern __GNU_INLINE float
271 extern __inline__ float
272 sqrtf(float f)
273 {
274         return (__inline_sqrtf(f));
275 }

277 extern __GNU_INLINE double
277 extern __inline__ double
278 fabs(double d)
279 {
280         double ret;

282         __asm__ __volatile__("fabsd %1,%0\n\t" : "=e" (ret) : "e" (d));
283         return (ret);
284 }

286 extern __GNU_INLINE float
286 extern __inline__ float
287 fabsf(float f)
288 {
289         float ret;

291         __asm__ __volatile__("fabss %1,%0\n\t" : "=f" (ret) : "f" (f));
292         return (ret);
293 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    6499 Fri Feb 12 11:52:36 2016
new/usr/src/lib/libm/sparcv9/src/libm_inlines.h
6648 illumos build should be explicit about C standards
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */

/*
 * Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

/*
 * Copyright 2011, Richard Lowe.
 */

/* Functions in this file are duplicated in locallibm.il.  Keep them in sync */

#ifndef _LIBM_INLINES_H
#define _LIBM_INLINES_H

#ifdef __GNUC__

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

#ifdef __cplusplus
extern "C" {
#endif

extern __GNU_INLINE enum fp_class_type
extern __inline__ enum fp_class_type
fp_classf(float f)
{
        enum fp_class_type ret;
        int fint;               /* scratch for f as int */
        uint64_t tmp;

        __asm__ __volatile__(
            "fabss  %3,%3\n\t"
            "st     %3,%1\n\t"
            "ld     %1,%0\n\t"
            "orcc   %%g0,%0,%%g0\n\t"
            "be,pn  %%icc,2f\n\t"
            "nop\n\t"
            "1:\n\t"
            "sethi  %%hi(0x7f800000),%2\n\t"
            "andcc  %0,%2,%%g0\n\t"
            "bne,pt %%icc,1f\n\t"
            "nop\n\t"
            "or     %%g0,1,%0\n\t"
            "ba     2f\n\t"     /* subnormal */
            "nop\n\t"
            "1:\n\t"
            "subcc  %0,%2,%%g0\n\t"
            "bge,pn %%icc,1f\n\t"
            "nop\n\t"
            "or     %%g0,2,%0\n\t"
            "ba     2f\n\t"     /* normal */
            "nop\n\t"
            "1:\n\t"
            "bg,pn  %%icc,1f\n\t"
            "nop\n\t"
            "or     %%g0,3,%0\n\t"
            "ba     2f\n\t"     /* infinity */
            "nop\n\t"
            "1:\n\t"
            "sethi  %%hi(0x00400000),%2\n\t"
            "andcc  %0,%2,%%g0\n\t"
            "or     %%g0,4,%0\n\t"
            "bne,pt %%icc,2f\n\t" /* quiet NaN */
            "nop\n\t"
            "or     %%g0,5,%0\n\t" /* signalling NaN */
            "2:\n\t"
            : "=r" (ret), "=m" (fint), "=r" (tmp), "+f" (f)
            :
            : "cc");

        return (ret);
}

extern __GNU_INLINE enum fp_class_type
extern __inline__ enum fp_class_type
fp_class(double d)
{
        enum fp_class_type ret;
        uint64_t dint;          /* Scratch for d-as-long */
        uint64_t tmp;

        __asm__ __volatile__(
            "fabsd  %3,%3\n\t"
            "std    %3,%1\n\t"
            "ldx    %1,%0\n\t"
            "orcc   %%g0,%0,%%g0\n\t"
            "be,pn  %%xcc,2f\n\t"
            "nop\n\t"
            "sethi  %%hi(0x7ff00000),%2\n\t"
            "sllx   %2,32,%2\n\t"
            "andcc  %0,%2,%%g0\n\t"
            "bne,pt %%xcc,1f\n\t"
            "nop\n\t"
            "or     %%g0,1,%0\n\t"
            "ba     2f\n\t"
            "nop\n\t"
            "1:\n\t"
            "subcc  %0,%2,%%g0\n\t"
            "bge,pn %%xcc,1f\n\t"
            "nop\n\t"
            "or     %%g0,2,%0\n\t"
            "ba     2f\n\t"
            "nop\n\t"
            "1:\n\t"
            "andncc %0,%2,%0\n\t"
```

```
126             "bne,pn %%xcc,1f\n\t"
127             "nop\n\t"
128             "or     %%g0,3,%0\n\t"
129             "ba     2f\n\t"
130             "nop\n\t"
131             "1:\n\t"
132             "sethi  %%hi(0x00080000),%2\n\t"
133             "sllx   %2,32,%2\n\t"
134             "andcc  %0,%2,%%g0\n\t"
135             "or     %%g0,4,%0\n\t"
136             "bne,pt %%xcc,2f\n\t"
137             "nop\n\t"
138             "or     %%g0,5,%0\n\t"
139             "2:\n\t"
140             : "=r" (ret), "=m" (dint), "=r" (tmp), "+e" (d)
141             :
142             : "cc");

144         return (ret);
145 }

147 extern __GNU_INLINE float
147 extern __inline__ float
148 __inline_sqrtf(float f)
149 {
150         float ret;

152         __asm__ __volatile__("fsqrts %1,%0\n\t" : "=f" (ret) : "f" (f));
153         return (ret);
154 }

156 extern __GNU_INLINE double
156 extern __inline__ double
157 __inline_sqrt(double d)
158 {
159         double ret;

161         __asm__ __volatile__("fsqrtd %1,%0\n\t" : "=f" (ret) : "f" (d));
162         return (ret);
163 }

165 extern __GNU_INLINE int
165 extern __inline__ int
166 __swapEX(int i)
167 {
168         int ret;
169         uint32_t fsr;
170         uint64_t tmp1, tmp2;

172         __asm__ __volatile__(
173             "and  %4,0x1f,%2\n\t"
174             "sll  %2,5,%2\n\t"  /* shift input to aexc bit location */
175             ".volatile\n\t"
176             "st   %%fsr,%1\n\t"
177             "ld   %1,%0\n\t"    /* %0 = fsr */
178             "andn %0,0x3e0,%3\n\t"
179             "or   %2,%3,%2\n\t" /* %2 = new fsr */
180             "st   %2,%1\n\t"
181             "ld   %1,%%fsr\n\t"
182             "srl  %0,5,%0\n\t"
183             "and  %0,0x1f,%0\n\t"
184             ".nonvolatile\n\t"
185             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2)
186             : "r" (i)
187             : "cc");
```



```
189         return (ret);
190 }

192 /*
193  * On the SPARC, __swapRP is a no-op; always return 0 for backward
194  * compatibility
195  */
196 /* ARGSUSED */
197 extern __GNU_INLINE enum fp_precision_type
197 extern __inline__ enum fp_precision_type
198 __swapRP(enum fp_precision_type i)
199 {
200         return (0);
201 }

203 extern __GNU_INLINE enum fp_direction_type
203 extern __inline__ enum fp_direction_type
204 __swapRD(enum fp_direction_type d)
205 {
206         enum fp_direction_type ret;
207         uint32_t fsr;
208         uint64_t tmp1, tmp2, tmp3;

210         __asm__ __volatile__(
211             "and   %5,0x3,%0\n\t"
212             "sll   %0,30,%2\n\t"        /* shift input to RD bit location */
213             ".volatile\n\t"
214             "st    %%fsr,%1\n\t"
215             "ld    %1,%0\n\t"           /* %0 = fsr */
216             /* mask of rounding direction bits */
217             "sethi %%hi(0xc0000000),%4\n\t"
218             "andn  %0,%4,%3\n\t"
219             "or    %2,%3,%2\n\t"        /* %2 = new fsr */
220             "st    %2,%1\n\t"
221             "ld    %1,%%fsr\n\t"
222             "srl   %0,30,%0\n\t"
223             "and   %0,0x3,%0\n\t"
224             ".nonvolatile\n\t"
225             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2), "=r" (tmp3)
226             : "r" (d)
227             : "cc");

229         return (ret);
230 }

232 extern __GNU_INLINE int
232 extern __inline__ int
233 __swapTE(int i)
234 {
235         int ret;
236         uint32_t fsr;
237         uint64_t tmp1, tmp2, tmp3;

239         __asm__ __volatile__(
240             "and   %5,0x1f,%0\n\t"
241             "sll   %0,23,%2\n\t"        /* shift input to TEM bit location */
242             ".volatile\n\t"
243             "st    %%fsr,%1\n\t"
244             "ld    %1,%0\n\t"           /* %0 = fsr */
245             /* mask of TEM (Trap Enable Mode bits) */
246             "sethi %%hi(0x0f800000),%4\n\t"
247             "andn  %0,%4,%3\n\t"
248             "or    %2,%3,%2\n\t"        /* %2 = new fsr */
249             "st    %2,%1\n\t"
250             "ld    %1,%%fsr\n\t"
251             "srl   %0,23,%0\n\t"
```

```
252             "and   %0,0x1f,%0\n\t"
253             ".nonvolatile\n\t"
254             : "=r" (ret), "=m" (fsr), "=r" (tmp1), "=r" (tmp2), "=r" (tmp3)
255             : "r" (i)
256             : "cc");

258         return (ret);
259 }


262 extern __GNU_INLINE double
262 extern __inline__ double
263 sqrt(double d)
264 {
265         return (__inline_sqrt(d));
266 }

268 extern __GNU_INLINE float
268 extern __inline__ float
269 sqrtf(float f)
270 {
271         return (__inline_sqrtf(f));
272 }

274 extern __GNU_INLINE double
274 extern __inline__ double
275 fabs(double d)
276 {
277         double ret;

279         __asm__ __volatile__("fabsd %1,%0\n\t" : "=e" (ret) : "e" (d));
280         return (ret);
281 }

283 extern __GNU_INLINE float
283 extern __inline__ float
284 fabsf(float f)
285 {
286         float ret;

288         __asm__ __volatile__("fabss %1,%0\n\t" : "=f" (ret) : "f" (f));
289         return (ret);
290 }
```

_____unchanged_portion_omitted_

```
**********************************************************
    6177 Fri Feb 12 11:52:37 2016
new/usr/src/lib/libsqlite/Makefile.com
6648 illumos build should be explicit about C standards
**********************************************************
```

```
   1 #
   2 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
   3 # Use is subject to license terms.
   4 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
   5 #

   7 # Make the SO name unlikely to conflict with any other
   8 # libsqlite that might also be found on the system.
   9 LIBRARY = libsqlite-sys.a

  11 VERS = .2.8.15
  12 OBJECTS = \
  13         attach.o        \
  14         auth.o          \
  15         btree.o         \
  16         btree_rb.o      \
  17         build.o         \
  18         copy.o          \
  19         date.o          \
  20         delete.o        \
  21         encode.o        \
  22         expr.o          \
  23         func.o          \
  24         hash.o          \
  25         insert.o        \
  26         main.o          \
  27         opcodes.o       \
  28         os.o            \
  29         pager.o         \
  30         parse.o         \
  31         pragma.o        \
  32         printf.o        \
  33         random.o        \
  34         select.o        \
  35         table.o         \
  36         tokenize.o      \
  37         trigger.o       \
  38         update.o        \
  39         util.o          \
  40         vacuum.o        \
  41         vdbe.o          \
  42         vdbeaux.o       \
  43         where.o

  45 include $(SRC)/lib/Makefile.lib

  47 # install this library in the root filesystem
  48 include $(SRC)/lib/Makefile.rootfs

  50 SRCDIR = ../src
  51 TOOLDIR = ../tool
  52 $(DYNLIB) := LDLIBS += -lc
  53 LIBS = $(DYNLIB) $(LINTLIB) $(NATIVERELOC)

  55 $(LINTLIB) :=   SRCS = ../$(LINTSRC)

  57 # generated sources
  58 GENSRC = opcodes.c parse.c

  60 # all sources
  61 SRCS = \
```



```
  62         $(GENSRC) \
  63         $(SRCDIR)/attach.c      \
  64         $(SRCDIR)/auth.c        \
  65         $(SRCDIR)/btree.c       \
  66         $(SRCDIR)/btree_rb.c    \
  67         $(SRCDIR)/build.c       \
  68         $(SRCDIR)/copy.c        \
  69         $(SRCDIR)/date.c        \
  70         $(SRCDIR)/delete.c      \
  71         $(SRCDIR)/encode.c      \
  72         $(SRCDIR)/expr.c        \
  73         $(SRCDIR)/func.c        \
  74         $(SRCDIR)/hash.c        \
  75         $(SRCDIR)/insert.c      \
  76         $(SRCDIR)/main.c        \
  77         $(SRCDIR)/os.c          \
  78         $(SRCDIR)/pager.c       \
  79         $(SRCDIR)/pragma.c      \
  80         $(SRCDIR)/printf.c      \
  81         $(SRCDIR)/random.c      \
  82         $(SRCDIR)/select.c      \
  83         $(SRCDIR)/table.c       \
  84         $(SRCDIR)/tokenize.c    \
  85         $(SRCDIR)/update.c      \
  86         $(SRCDIR)/util.c        \
  87         $(SRCDIR)/vacuum.c      \
  88         $(SRCDIR)/vdbe.c        \
  89         $(SRCDIR)/vdbeaux.c     \
  90         $(SRCDIR)/where.c       \
  91         $(SRCDIR)/trigger.c

  93 MYCPPFLAGS = -D_REENTRANT -DTHREADSAFE=1 -DHAVE_USLEEP=1 -I. -I.. -I$(SRCDIR)
  94 CPPFLAGS += $(MYCPPFLAGS)

  96 CERRWARN += -_gcc=-Wno-implicit-function-declaration
  97 CERRWARN += -_gcc=-Wno-uninitialized
  98 CERRWARN += -_gcc=-Wno-unused-function
  99 CERRWARN += -_gcc=-Wno-unused-label

 101 MAPFILES = ../mapfile-sqlite

 103 # headers generated here
 104 GENHDR = opcodes.h parse.h

 106 # Header files used by all library source files.
 107 #
 108 HDR = \
 109         $(GENHDR) \
 110         $(SRCDIR)/btree.h       \
 111         $(SRCDIR)/config.h      \
 112         $(SRCDIR)/hash.h        \
 113         $(SRCDIR)/os.h          \
 114         ../sqlite.h             \
 115         $(SRCDIR)/sqliteInt.h   \
 116         $(SRCDIR)/vdbe.h        \
 117         $(SRCDIR)/vdbeInt.h

 119 #
 120 # Sources used for test harness
 121 #
 122 TESTSRC = \
 123         $(SRCDIR)/tclsqlite.c   \
 124         $(SRCDIR)/btree.c       \
 125         $(SRCDIR)/func.c        \
 126         $(SRCDIR)/os.c          \
 127         $(SRCDIR)/pager.c       \
```

```
128         $(SRCDIR)/test1.c       \
129         $(SRCDIR)/test2.c       \
130         $(SRCDIR)/test3.c       \
131         $(SRCDIR)/md5.c

133 TESTOBJS = $(TESTSRC:$(SRCDIR)/%.c=%.o)

135 TESTCLEAN = $(TESTOBJS) test.db test.tcl test1.bt test2.db testdb

137 #
138 # Native variant (needed by cmd/configd)
139 #
140 NATIVERELOC = libsqlite-native.o
141 NATIVEPROGS = testfixture
142 NATIVEOBJS = $(OBJS:%.o=%-native.o)

144 NATIVETARGETS = $(NATIVEPROGS) $(NATIVEOBJS) $(NATIVERELOC)

146 $(NATIVETARGETS) :=     CC = $(NATIVECC)
147 $(NATIVETARGETS) :=     LD = $(NATIVELD)
148 $(NATIVETARGETS) :=     CFLAGS = $(NATIVE_CFLAGS)
149 $(NATIVETARGETS) :=     CPPFLAGS = $(MYCPPFLAGS)
150 $(NATIVETARGETS) :=     LDFLAGS =
151 $(NATIVETARGETS) :=     LDLIBS = -lc

153 $(OBJS) :=              CFLAGS += $(CTF_FLAGS)
154 $(OBJS) :=              CTFCONVERT_POST = $(CTFCONVERT_O)

156 TCLBASE = /usr/sfw
157 TCLVERS = tcl8.3

159 testfixture := MYCPPFLAGS += -I$(TCLBASE)/include -DTCLSH -DSQLITE_TEST=1
160 #
161 # work around compiler issues
162 #
163 testfixture := CFLAGS += \
164         -erroff=E_ARRAY_OF_INCOMPLETE \
165         -erroff=E_ARG_INCOMPATIBLE_WITH_ARG

167 testfixture := LDLIBS += -R$(TCLBASE)/lib -L$(TCLBASE)/lib -l$(TCLVERS) -lm -ldl

169 CLEANFILES += \
170         $(NATIVETARGETS) \
171         $(TESTCLEAN)    \
172         lemon           \
173         lemon.o         \
174         lempar.c        \
175         opcodes.c       \
176         opcodes.h       \
177         parse_tmp.c     \
178         parse_tmp.h     \
179         parse_tmp.out   \
180         parse_tmp.y     \
181         parse.c         \
182         parse.h

184 ENCODING  = ISO8859


187 .PARALLEL: $(OBJS) $(OBJS:%.o=%-native.o)
188 .KEEP_STATE:

190 # This is the default Makefile target.  The objects listed here
191 # are what get build when you type just "make" with no arguments.
192 #
193 all:            $(LIBS)
```



```
194 install:        all \
195                 $(ROOTLIBDIR)/$(DYNLIB) \
196                 $(ROOTLIBDIR)/$(LINTLIB) \
197                 $(ROOTLIBDIR)/$(NATIVERELOC)

199 lint:

201 all_h: $(GENHDR)

203 $(ROOTLIBDIR)/$(NATIVERELOC)    := FILEMODE= 644
204 $(ROOTLINTDIR)/$(LINTLIB)      := FILEMODE= 644

206 $(ROOTLINK): $(ROOTLIBDIR) $(ROOTLIBDIR)/$(DYNLIB)
207         $(INS.liblink)

209 $(ROOTLINTDIR)/%: ../%
210         $(INS.file)

212 native: $(NATIVERELOC)

214 $(NATIVERELOC): objs .WAIT $(OBJS:%.o=%-native.o)
215         $(LD) -r -o $(NATIVERELOC) $(OBJS:%.o=%-native.o)

217 opcodes.h: $(SRCDIR)/vdbe.c
218         @echo "Generating $@"; \
219          $(RM) -f $@ ; \
220          echo '/* Automatically generated file.  Do not edit */' > $@ ; \
221          grep '^case OP_' $(SRCDIR)/vdbe.c | \
222             sed -e 's/://' | \
223             $(AWK) '{printf "#define %-30s %3d\n", $$2, ++cnt}' >> $@

225 opcodes.c: $(SRCDIR)/vdbe.c
226         @echo "Generating $@"; \
227          $(RM) -f $@ ; \
228          echo '/* Automatically generated file.  Do not edit */' > $@ ; \
229          echo 'char *sqliteOpcodeNames[] = { "???", ' >> $@ ; \
230          grep '^case OP_' $(SRCDIR)/vdbe.c | \
231             sed -e 's/^.*OP_/  "/' -e 's/:.*$$/", /' >> $@ ; \
232          echo '};' >> $@

234 testfixture: FRC
235         @if [ -f $(TCLBASE)/include/tcl.h ]; then \
236                 unset SUNPRO_DEPENDENCIES; \
237                 echo $(LINK.c) -o testfixture $(TESTSRC) $(LIBRARY) $(LDLIBS) ;\
238                 exec $(LINK.c) -o testfixture $(TESTSRC) $(LIBRARY) $(LDLIBS) ;\
239         else \
240                 echo "$(TCLBASE)/include/tcl.h: not found."; \
241                 exit 1; \
242         fi

244 parse.h parse.c : $(SRCDIR)/parse.y $(TOOLDIR)/lemon.c $(TOOLDIR)/lempar.c
245         -$(RM) parse_tmp.y lempar.c
246         $(CP) $(SRCDIR)/parse.y parse_tmp.y
247         $(CP) $(TOOLDIR)/lempar.c lempar.c
248         $(NATIVECC) $(NATIVE_CFLAGS) -o lemon $(TOOLDIR)/lemon.c
248         $(NATIVECC) -o lemon $(TOOLDIR)/lemon.c
249         ./lemon parse_tmp.y
250         -$(RM) parse.c parse.h
251         $(CP) parse_tmp.h parse.h
252         $(CP) parse_tmp.c parse.c

254 objs/%-native.o: $(SRCDIR)/%.c $(GENHDR)
255         $(COMPILE.c) -o $@ $<
256         $(POST_PROCESS_O)

258 objs/%-native.o: %.c $(GENHDR)
```

```
259         $(COMPILE.c) -o $@ $<
260         $(POST_PROCESS_O)

262 objs/parse-native.o: parse.c $(GENHDR)
263         $(COMPILE.c) -o $@ parse.c
264         $(POST_PROCESS_O)

266 objs/%.o pics/%.o: $(SRCDIR)/%.c $(GENHDR)
267         $(COMPILE.c) -o $@ $<
268         $(POST_PROCESS_O)

270 objs/%.o pics/%.o: %.c $(GENHDR)
271         $(COMPILE.c) -o $@ $<
272         $(POST_PROCESS_O)

274 # need direct rules for generated files
275 objs/opcodes.o pics/opcodes.o: opcodes.c $(GENHDR)
276         $(COMPILE.c) -o $@ opcodes.c
277         $(POST_PROCESS_O)

279 objs/parse.o pics/parse.o: parse.c $(GENHDR)
280         $(COMPILE.c) -o $@ parse.c
281         $(POST_PROCESS_O)

283 include $(SRC)/lib/Makefile.targ

285 FRC:
```

****
1156 Fri Feb 12 11:52:37 2016
new/usr/src/test/zfs-tests/tests/functional/checksum/Makefile.subdirs
6648 illumos build should be explicit about C standards
****

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

  12 #
  13 # Copyright (c) 2015 by Delphix. All rights reserved.
  14 #

  16 include $(SRC)/cmd/Makefile.cmd

  18 ROOTOPTPKG = $(ROOT)/opt/zfs-tests
  19 TESTDIR = $(ROOTOPTPKG)/tests/functional/checksum

  21 OBJS = $(PROG:%=%.o)
  22 SRCS = $(OBJS:%.o=../%.c)

  24 CMD32 = $(PROG:%=$(TESTDIR)/%.$(MACH))
  25 CMD64 = $(PROG:%=$(TESTDIR)/%.$(MACH64))
  26 CMDS = $(CMD32) $(CMD64)
  27 $(CMDS) := FILEMODE = 0555

  29 C99MODE = $(C99_ENABLE)
  30 CFLAGS += -I $(SRC)/uts/common -D_KERNEL
  29 C99MODE = -xc99=%all
  30 CFLAGS = -I $(SRC)/uts/common -D_KERNEL
  31 LDLIBS += -lmd
  32 LINTFLAGS += -u
  33 LINTFLAGS64 += -u

  35 all: $(PROG) $(TESTDIR)

  37 lint: lint_SRCS

  39 clobber: clean

  41 clean:
  42         $(RM) $(PROG)

  44 include $(SRC)/cmd/Makefile.targ

  46 $(TESTDIR):
  47         $(INS.dir)

  49 $(TESTDIR)/%.$(MACH): %
  50         $(INS.rename)

  52 $(TESTDIR)/%.$(MACH64): %
  53         $(INS.rename)

  55 %: ../%.c
  56         $(LINK.c) -o $@ $< $(LDLIBS)
  57         $(POST_PROCESS)
```

```
**********************************************************
   22600 Fri Feb 12 11:52:38 2016
new/usr/src/uts/common/io/audio/drv/audioemu10k/dsp/asm10k.c
6648 illumos build should be explicit about C standards
**********************************************************
```

*unchanged_portion_omitted_*

```
 435 static void
 436 compile_mono(char **tokens, int cnt)
 437 {
 438         char *parm, *def;
 439         int n, num;
 440         char tmp[128];

 441         CHECK_COUNT(tokens, cnt, 3, 3);

 443         parm = tokens[1];
 444         def = tokens[2];

 446         n = fle.parms.ngpr;
 447         if (n >= MAX_GPR_PARMS) {
 448                 error("Too many GPR parameters");
 449                 return;
 450         }

 452         if (sscanf(def, "%d", &num) != 1) {
 453                 error("Bad integer value near '%s'", def);
 454                 return;
 455         }

 457         (void) strcpy(fle.parms.gpr[n].name, parm);
 458         fle.parms.gpr[n].num = ngpr;
 459         fle.parms.gpr[n].def = num;
 460         fle.parms.ngpr = n + 1;

 462         add_init(ngpr, num, parm);

 464         add_symbol(parm, SY_PARM, gpr_base + ngpr++);
 465 }
```

*unchanged_portion_omitted_*

```
 604 static void
 605 compile_asm(char **tokens, int cnt)
 606 {
 608         char *parms[4];
 607         sym_t *symbols[4];
 608 #define EMIT(o, r, a, x, y) \
 609         fle.code[pc*2] =  ((x) << 10) | (y);                    \
 610         fle.code[pc*2+1] = ((o) << 20) | ((r) << 10) | a; pc++
 611 #define EMIT_AUDIGY(o, r, a, x, y) \
 612         fle.code[pc*2] =  ((x) << 12) | (y);                    \
 613         fle.code[pc*2+1] = ((o) << 24) | ((r) << 12) | a; pc++

 615         int i, nerr = 0;
 617         int n = 0, nerr = 0;
 616         int ninputs = 0;

 618         CHECK_COUNT(tokens, cnt, 5, 5);

 620         for (i = 0; i < 4; i++) {
 621                 if ((symbols[i] = find_symbol(tokens[i+1])) == NULL) {
 622                         (void) fprintf(stderr, "%s\n", tokens[i+1]);
 623                         nerr++;
 624                         error("Undefined symbol '%s'", tokens[i + 1]);
 625                         continue;
 626                 }
```



```
 628                 if (symbols[i]->type == SY_INPUT)
 629                         ninputs++;

 631                 if (symbols[i]->type == SY_ACCUM && i != 1)
 632                         error("Bad usage of 'accum' operand.");
 633         }

 635         if (nerr > 0)
 636                 return;

 638         if (ninputs > 1) {
 639                 error("Attempt to access more than one input "
 640                     "GPRs by the same instruction");
 641         }

 643         for (i = 0; instructions[i].name != NULL; i++)
 644                 if (strcasecmp(tokens[0], instructions[i].name) == 0)  {

 646                         if (is_audigy) {
 647                                 EMIT_AUDIGY(instructions[i].opcode,
 648                                     symbols[0]->arg,
 649                                     symbols[1]->arg,
 650                                     symbols[2]->arg,
 651                                     symbols[3]->arg);
 652                         } else {
 653                                 EMIT(instructions[i].opcode,
 654                                     symbols[0]->arg,
 655                                     symbols[1]->arg,
 656                                     symbols[2]->arg,
 657                                     symbols[3]->arg);
 658                         }

 660                         return;
 661                 }

 663         error("Unrecognized instruction '%s'", tokens[0]);
 664 }
```

*unchanged_portion_omitted_*

```
 811 static void
 812 produce_map(char *name)
 813 {
 816         char fname[1024];
 814         int i;
 815         FILE *f;

 817         if ((f = fopen(name, "w")) == NULL) {
 818                 perror(name);
 819                 return;
 820         }

 822         (void) fprintf(f, "%d\n", pc);

 824         for (i = 0; i < nsyms; i++) {
 825                 (void) fprintf(f, "%04x %x %s\n",
 826                     symtab[i].arg, symtab[i].type, symtab[i].name);
 827         }

 829         (void) fclose(f);
 830         if (verbose) {
 831                 (void) fprintf(stderr,
 832                     "No errors detected - Map written to %s\n", name);
 833         }
 834 }
```

*unchanged_portion_omitted_*

```
 954 int
 955 main(int argc, char *argv[])
 956 {
 957         char *outfile;
 960         char line[4096], *p, *s, *outfile;
 961         char *iline;
 958         int i;
 959         FILE *input;
 960         char *tokens[10];
 961         int tokcnt;
 962         char *mapfile = NULL;
 963         char *header = NULL;
 964         char *prefix = NULL;

 966         outfile = NULL;
 967         infile = NULL;
 968         input = NULL;
 969         progname = argv[0];

 971         while ((i = getopt(argc, argv, "m:h:o:i:P:021v")) != EOF) {
 972                 switch (i) {
 973                 case 'o':
 974                         outfile = optarg;
 975                         break;
 976                 case 'i':
 977                         infile = strdup(optarg);
 978                         break;
 979                 case 'm':
 980                         mapfile = optarg;
 981                         break;
 982                 case 'P':
 983                         prefix = optarg;
 984                         break;
 985                 case 'h':
 986                         header = optarg;
 987                         break;
 988                 case '0':
 989                         parms_only = 1;
 990                         break;
 991                 case '2':
 992                         is_audigy = 1;
 993                         break;
 994                 case '1':
 995                         is_audigy = 0;
 996                         break;
 997                 case 'v':
 998                         verbose++;
 999                         break;
1000                 default:
1001                         (void) fprintf(stderr,
1002                             "usage: %s [-m <map>] [-h <header>] "
1003                             "[-o <binary>] [-i <source>] [-2|-1]",
1004                             progname);
1005                         exit(-1);
1006                         break;
1007                 }
1008         }

1010         if ((outfile == NULL) && (mapfile == NULL) && (header == NULL)) {
1011                 outfile = "dsp.bin";
1012         }

1014         if (infile) {
1015                 input = fopen(infile, "r");
1016                 if (input == NULL) {
```



```
1017                         perror(infile);
1018                         exit(-1);
1019                 }
1020         } else {
1021                 infile = strdup("<stdin>");
1022                 input = stdin;
1023         }

1025         if (is_audigy) {
1026                 gpr_base = 0x400;
1027                 input_base = 0x40;
1028                 output_base = 0x60;
1029                 if (verbose)
1030                         (void) fprintf(stderr, "Compiling for SB Audigy\n");
1031         } else {
1032                 if (verbose)
1033                         (void) fprintf(stderr, "Compiling for SB Live\n");
1034         }

1036         init_compiler();

1038         while ((tokcnt = getaline(input, tokens)) != -1) {
1039                 /* skip empty lines */
1040                 if (tokcnt == 0) {
1041                         continue;
1042                 }

1044                 if (strcmp(tokens[0], "#") == 0) {
1045                         int     num;
1046                         if ((tokcnt >= 3) &&
1047                             (sscanf(tokens[1], "%d", &num) == 1)) {
1048                                 lineno = num;
1049                                 free(infile);
1050                                 infile = strdup(tokens[2]);
1051                                 /* we don't want to count the # directive */
1052                                 lineno--;
1053                         }

1055                         /* unknown # directive? muddle on... */
1056                         continue;
1057                 }
1058                 if (*tokens[0] == '.') {
1059                         compile_directive(tokens, tokcnt);
1060                 } else {
1061                         compile_asm(tokens, tokcnt);
1062                 }
1063         }

1065         if (lineno < 1) {
1066                 error("Empty input");
1067         }

1069         if (errors == 0) {
1070                 if (verbose) {
1071                         (void) fprintf(stderr,
1072                             "%d instructions out of 512 assembled\n", pc);
1073                 }

1075                 if (outfile)
1076                         produce_output(outfile);
1077                 if (mapfile)
1078                         produce_map(mapfile);
1079                 if (header)
1080                         produce_header(header, prefix);
1081         }
```

```
1083         if (errors > 0) {
1084                 (void) fprintf(stderr, "%d errors - compile failed\n", errors);
1085                 exit(-1);
1086         }

1088         return (0);
1089 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   14154 Fri Feb 12 11:52:38 2016
new/usr/src/uts/i86pc/Makefile.rules
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright (c) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
#

#       This Makefile defines the build rules for the directory uts/i86pc
#       and its children. These are the source files which are i86pc
#       "implementation architecture" dependent.
#
#       The following two-level ordering must be maintained in this file.
#         Lines are sorted first in order of decreasing specificity based on
#         the first directory component.  That is, i86pc rules come before
#         intel rules come before common rules.
#
#         Lines whose initial directory components are equal are sorted
#         alphabetically by the remaining components.

#
#       Section 1a: C object build rules
#
$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/cpu/amd_opteron/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/cpu/authenticamd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/cpu/generic_cpu/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/cpu/genuineintel/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```



```
$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/acpi_drv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/fipe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/acpi/acpidev/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/acpi/acpinex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

SBD_IOCTL       = $(UTSBASE)/i86pc/sys/sbd_ioctl.h
DRMACH_IO       = $(UTSBASE)/i86pc/io/acpi/drmach_acpi
DRMACH_GENERR   = $(DRMACH_IO)/sbdgenerr
DR_IO           = $(UTSBASE)/i86pc/io/dr
DR_GENERR       = $(DR_IO)/sbdgenerr

$(DRMACH_GENERR):       $(DR_IO)/sbdgenerr.pl
        $(RM) $@
        $(CAT) $(DR_IO)/sbdgenerr.pl > $@
        $(CHMOD) +x $@

$(DRMACH_IO)/drmach_err.c:      $(DRMACH_GENERR) $(SBD_IOCTL)
        $(RM) $@
        $(DRMACH_GENERR) EX86 < $(SBD_IOCTL) > $(DRMACH_IO)/drmach_err.c

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/acpi/drmach_acpi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/amd_iommu/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(DR_GENERR):                   $(DR_IO)/sbdgenerr.pl
        $(RM) $@
        $(CAT) $(DR_IO)/sbdgenerr.pl > $@
        $(CHMOD) +x $@

$(DR_IO)/dr_err.c:              $(DR_GENERR) $(SBD_IOCTL)
        $(RM) $@
        $(DR_GENERR) ESBD < $(SBD_IOCTL) > $(DR_IO)/dr_err.c

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/dr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/ioat/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
```

```
128         $(CTFCONVERT_O)

130 $(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/pciex/hotplug/%.c
131         $(COMPILE.c) -o $@ $<
132         $(CTFCONVERT_O)

134 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pcplusmp/%.c
135         $(COMPILE.c) -o $@ $<
136         $(CTFCONVERT_O)

138 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pcplusmp/%.s
139         $(COMPILE.s) -o $@ $<

141 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/apix/%.c
142         $(COMPILE.c) -o $@ $<
143         $(CTFCONVERT_O)

145 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/ppm/%.c
146         $(COMPILE.c) -o $@ $<
147         $(CTFCONVERT_O)

149 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/ppm/%.s
150         $(COMPILE.s) -o $@ $<

152 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/psm/%.c
153         $(COMPILE.c) -o $@ $<
154         $(CTFCONVERT_O)

156 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/psm/%.s
157         $(COMPILE.s) -o $@ $<

159 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/tzmon/%.c
160         $(COMPILE.c) -o $@ $<
161         $(CTFCONVERT_O)

163 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/ml/%.s
164         $(COMPILE.s) -o $@ $<

166 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/os/%.c
167         $(COMPILE.c) -_gcc=-fno-stack-protector -o $@ $<
168         $(CTFCONVERT_O)

170 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/os/cpupm/%.c
171         $(COMPILE.c) -o $@ $<
172         $(CTFCONVERT_O)

174 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/boot/%.c
175         $(COMPILE.c) -o $@ $<
176         $(CTFCONVERT_O)

178 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/vm/%.c
179         $(COMPILE.c) -o $@ $<
180         $(CTFCONVERT_O)

182 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/%.c
183         $(COMPILE.c) -o $@ $<
184         $(CTFCONVERT_O)

186 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppm/%.c
187         $(COMPILE.c) -o $@ $<
188         $(CTFCONVERT_O)

190 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/%.c
191         $(COMPILE.c) -o $@ $<
192         $(CTFCONVERT_O)
```



```
194 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/os/%.c
195         $(COMPILE.c) -o $@ $<
196         $(CTFCONVERT_O)

198 $(OBJS_DIR)/%.o:                $(SRC)/common/dis/i386/%.c
199         $(COMPILE.c) -o $@ $<
200         $(CTFCONVERT_O)

202 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/gfx_private/%.c
203         $(COMPILE.c) -o $@ $<
204         $(CTFCONVERT_O)

206 $(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/xsvc/%.c
207         $(COMPILE.c) -o $@ $<
208         $(CTFCONVERT_O)

210 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/xen/os/%.c
211         $(COMPILE.c) -o $@ $<
212         $(CTFCONVERT_O)

214 #
215 # dboot stuff is always 32 bit, linked to run with phys_addr == virt_addr
216 #
217 DBOOT_OBJS_DIR  = dboot/$(OBJS_DIR)
218 DBOOT_MACH_32   = -D_BOOT_TARGET_i386
219 DBOOT_MACH_64   = -D_BOOT_TARGET_amd64
220 DBOOT_DEFS      = -D_BOOT $(DBOOT_MACH_$(CLASS))
221 DBOOT_DEFS      += -D_MACHDEP -D_KMEMUSER -U_KERNEL -D_I32LPx
222 DBOOT_FLAGS     = $(CCVERBOSE) $(C99MODE) $(CERRWARN) $(CCNOAUTOINLINE)
222 DBOOT_FLAGS     = $(CCVERBOSE) $(CERRWARN) $(CCNOAUTOINLINE)

224 DBOOT_CC_INCL   = -I$(SRC)/common -I$(SRC)/common/util $(INCLUDE_PATH)
225 DBOOT_AS_INCL   = $(AS_INC_PATH)

227 DBOOT_AS        = $(ONBLD_TOOLS)/bin/$(MACH)/aw

229 DBOOT_LINTS_DIR = $(DBOOT_OBJS_DIR)
230 DBOOT_LINTFLAGS_i86pc   = $(LINTFLAGS_i386_32) $(LINTTAGS_i386_32)

232 DBOOT_LINTFLAGS = $(DBOOT_LINTFLAGS_$(PLATFORM)) $(LINTTAGS) $(C99LMODE) \
233         $(CPPFLAGS) $(DBOOT_DEFS)
234 DBOOT_LOCAL_LINTFLAGS   = -c -dirout=$(DBOOT_LINTS_DIR) -I$(SRC)/common \
235         $(DBOOT_LINTFLAGS) $(DBOOT_LINTTAGS)

237 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/boot/%.c
238         $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

240 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/dboot/%.c
241         $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

243 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/intel/ia32/%.s
244         $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

246 $(DBOOT_OBJS_DIR)/%.o:          $(COMMONBASE)/crypto/sha1/%.c
247         $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<
247         $(i386_CC) $(CERRWARN) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

249 $(DBOOT_OBJS_DIR)/%.o:          $(COMMONBASE)/util/%.c
250         $(i386_CC) $(DBOOT_FLAGS) -O $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

252 $(DBOOT_OBJS_DIR)/%.o:          $(COMMONBASE)/util/i386/%.s
253         $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

255 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/dboot/%.s
256         $(DBOOT_AS) -P -D_ASM $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<
```

```
258 #
259 # Stuff to build bios_call.o for the kernel.
260 #
261 MAPFILE_BIOS    = $(UTSBASE)/i86pc/conf/Mapfile.bios
262 $(OBJS_DIR)/bios_call.o:    $(UTSBASE)/i86pc/ml/bios_call_src.s
263         $(COMPILE.s) -o $(OBJS_DIR)/bios_call_src.o \
264                 $(UTSBASE)/i86pc/ml/bios_call_src.s
265         $(LD) -dn -M $(MAPFILE_BIOS)  \
266                 -o $(OBJS_DIR)/bios_call_src $(OBJS_DIR)/bios_call_src.o
267         @echo "  .data"                         > $(OBJS_DIR)/bios_call.s
268         @echo "  .globl bios_image"             >> $(OBJS_DIR)/bios_call.s
269         @echo "bios_image:"                     >> $(OBJS_DIR)/bios_call.s
270         $(ELFEXTRACT) $(OBJS_DIR)/bios_call_src >> $(OBJS_DIR)/bios_call.s
271         @echo "  .align 4"                      >> $(OBJS_DIR)/bios_call.s
272         @echo "  .globl bios_size"              >> $(OBJS_DIR)/bios_call.s
273         @echo "bios_size:"                      >> $(OBJS_DIR)/bios_call.s
274         @echo "  .long . - bios_image"          >> $(OBJS_DIR)/bios_call.s
275         $(COMPILE.s) -o $@ $(OBJS_DIR)/bios_call.s

277 #
278 # Stuff to build fb_swtch.o for the kernel.
279 #
280 MAPFILE_FBSWTCH = $(UTSBASE)/i86pc/conf/Mapfile.fb_swtch
281 $(OBJS_DIR)/fb_swtch.o:    $(UTSBASE)/i86pc/ml/fb_swtch_src.s
282         $(COMPILE.s) -o $(OBJS_DIR)/fb_swtch_src.o \
283                 $(UTSBASE)/i86pc/ml/fb_swtch_src.s
284         $(LD) -dn -M $(MAPFILE_FBSWTCH)  \
285                 -o $(OBJS_DIR)/fb_swtch_src $(OBJS_DIR)/fb_swtch_src.o
286         @echo "  .data"                         > $(OBJS_DIR)/fb_swtch.s
287         @echo "  .globl fb_swtch_image"         >> $(OBJS_DIR)/fb_swtch.s
288         @echo "fb_swtch_image:"                 >> $(OBJS_DIR)/fb_swtch.s
289         $(ELFEXTRACT) $(OBJS_DIR)/fb_swtch_src  >> $(OBJS_DIR)/fb_swtch.s
290         @echo "  .align 4"                      >> $(OBJS_DIR)/fb_swtch.s
291         @echo "  .globl fb_swtch_size"          >> $(OBJS_DIR)/fb_swtch.s
292         @echo "fb_swtch_size:"                  >> $(OBJS_DIR)/fb_swtch.s
293         @echo "  .long . - fb_swtch_image"      >> $(OBJS_DIR)/fb_swtch.s
294         $(COMPILE.s) -o $@ $(OBJS_DIR)/fb_swtch.s

296 # ridiculous contortions ---
297 ATOMIC_SUBDIR_32        = i386
298 ATOMIC_SUBDIR_64        = amd64
299 ATOMIC_SUBDIR           = $(ATOMIC_SUBDIR_$(CLASS))

301 $(OBJS_DIR)/%.o:                $(SRC)/common/atomic/$(ATOMIC_SUBDIR)/%.s
302         $(COMPILE.s) -o $@ $<

304 #
305 # dtrace stubs
306 #

308 $(OBJS_DIR)/dtracestubs.s:      $(UNIX_O) $(LIBS)
309         $(NM) -u $(UNIX_O) $(LIBS) | $(GREP) __dtrace_probe_ | $(SORT) | \
310             $(UNIQ) | $(AWK) '{ \
311             printf("\t.globl %s\n\t.type %s,@function\n%s:\n", \
312             $$1, $$1, $$1); }' > $(OBJS_DIR)/dtracestubs.s

314 $(DTRACESTUBS): $(DTRACESTUBS_O)
315         $(BUILD.SO) $(DTRACESTUBS_O)

317 #
318 #       Section 1b: Lint 'object' build rules
319 #
320 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/conf/%.c
321         @($(LHEAD) $(LINT.c) $< $(LTAIL))

323 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/cpu/amd_opteron/%.c
```



```
324         @($(LHEAD) $(LINT.c) $< $(LTAIL))

326 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/cpu/authenticamd/%.c
327         @($(LHEAD) $(LINT.c) $< $(LTAIL))

329 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/cpu/generic_cpu/%.c
330         @($(LHEAD) $(LINT.c) $< $(LTAIL))

332 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/cpu/genuineintel/%.c
333         @($(LHEAD) $(LINT.c) $< $(LTAIL))

335 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/%.c
336         @($(LHEAD) $(LINT.c) $< $(LTAIL))

338 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/acpi_drv/%.c
339         @($(LHEAD) $(LINT.c) $< $(LTAIL))

341 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/fipe/%.c
342         @($(LHEAD) $(LINT.c) $< $(LTAIL))

344 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/acpi/acpidev/%.c
345         @($(LHEAD) $(LINT.c) $< $(LTAIL))

347 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/acpi/acpinex/%.c
348         @($(LHEAD) $(LINT.c) $< $(LTAIL))

350 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/acpi/drmach_acpi/%.c
351         @($(LHEAD) $(LINT.c) $< $(LTAIL))

353 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/amd_iommu/%.c
354         @($(LHEAD) $(LINT.c) $< $(LTAIL))

356 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/dr/%.c
357         @($(LHEAD) $(LINT.c) $< $(LTAIL))

359 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/ioat/%.c
360         @($(LHEAD) $(LINT.c) $< $(LTAIL))

362 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pci/%.c
363         @($(LHEAD) $(LINT.c) $< $(LTAIL))

365 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pciex/%.c
366         @($(LHEAD) $(LINT.c) $< $(LTAIL))

368 $(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/pciex/hotplug/%.c
369         @($(LHEAD) $(LINT.c) $< $(LTAIL))

371 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pcplusmp/%.c
372         @($(LHEAD) $(LINT.c) $< $(LTAIL))

374 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pcplusmp/%.s
375         @($(LHEAD) $(LINT.s) $< $(LTAIL))

377 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/apix/%.c
378         @($(LHEAD) $(LINT.c) $< $(LTAIL))

380 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/apix/%.s
381         @($(LHEAD) $(LINT.s) $< $(LTAIL))

383 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/ppm/%.c
384         @($(LHEAD) $(LINT.c) $< $(LTAIL))

386 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/ppm/%.s
387         @($(LHEAD) $(LINT.s) $< $(LTAIL))

389 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/psm/%.c
```

```
390         @($(LHEAD) $(LINT.c) $< $(LTAIL))

392 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/psm/%.s
393         @($(LHEAD) $(LINT.s) $< $(LTAIL))

395 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/tzmon/%.c
396         @($(LHEAD) $(LINT.c) $< $(LTAIL))

398 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/ml/%.s
399         @($(LHEAD) $(LINT.s) $< $(LTAIL))

401 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/os/%.c
402         @($(LHEAD) $(LINT.c) $< $(LTAIL))

404 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/os/cpupm/%.c
405         @($(LHEAD) $(LINT.c) $< $(LTAIL))

407 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/boot/%.c
408         @($(LHEAD) $(LINT.c) $< $(LTAIL))

410 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/vm/%.c
411         @($(LHEAD) $(LINT.c) $< $(LTAIL))

413 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/%.c
414         @($(LHEAD) $(LINT.c) $< $(LTAIL))

416 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ppm/%.c
417         @($(LHEAD) $(LINT.c) $< $(LTAIL))

419 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/%.c
420         @($(LHEAD) $(LINT.c) $< $(LTAIL))

422 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/os/%.c
423         @($(LHEAD) $(LINT.c) $< $(LTAIL))

425 $(LINTS_DIR)/%.ln:              $(SRC)/common/dis/i386/%.c
426         @($(LHEAD) $(LINT.c) $< $(LTAIL))

428 $(LINTS_DIR)/%.ln:              $(SRC)/common/atomic/%.c
429         @($(LHEAD) $(LINT.c) $< $(LTAIL))

431 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/gfx_private/%.c
432         @($(LHEAD) $(LINT.c) $< $(LTAIL))

434 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/xsvc/%.c
435         @($(LHEAD) $(LINT.c) $< $(LTAIL))

437 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/xen/os/%.c
438         @($(LHEAD) $(LINT.c) $< $(LTAIL))

440 #
441 # bios call has a funky name change while building
442 #
443 $(LINTS_DIR)/bios_call.ln:      $(UTSBASE)/i86pc/ml/bios_call_src.s
444         @($(LHEAD) $(LINT.s) $(UTSBASE)/i86pc/ml/bios_call_src.s  $(LTAIL))
445         @mv $(LINTS_DIR)/bios_call_src.ln $(LINTS_DIR)/bios_call.ln

447 #
448 # So does fb_swtch
449 #
450 $(LINTS_DIR)/fb_swtch.ln:       $(UTSBASE)/i86pc/ml/fb_swtch_src.s
451         @($(LHEAD) $(LINT.s) $(UTSBASE)/i86pc/ml/fb_swtch_src.s  $(LTAIL))
452         @mv $(LINTS_DIR)/fb_swtch_src.ln $(LINTS_DIR)/fb_swtch.ln

454 #
455 # dboot always compiles in 32-bit mode, so force lint to be 32-bit mode too.
```



```
456 #
457 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/dboot/%.c
458         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

460 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/dboot/%.s
461         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

463 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/crypto/sha1/%.c
464         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

466 $(DBOOT_LINTS_DIR)/%.ln: $(UTSBASE)/i86pc/boot/%.c
467         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

469 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/util/%.c
470         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

472 $(DBOOT_LINTS_DIR)/%.ln: $(COMMONBASE)/util/i386/%.s
473         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))
```

```
**********************************************************
    2286 Fri Feb 12 11:52:39 2016
new/usr/src/uts/i86pc/genassym/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#
#       This makefile drives the production of assym.h through
#       compile time intialized data.
#
#       i86pc implementation architecture dependent
#


#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

ASSYM_H         = $(DSF_DIR)/$(OBJS_DIR)/assym.h
KDI_ASSYM_H     = $(DSF_DIR)/$(OBJS_DIR)/kdi_assym.h
GENASSYM        = $(DSF_DIR)/$(OBJS_DIR)/genassym

#
#       Include common rules.
#
include $(UTSBASE)/i86pc/Makefile.i86pc

#
#       Define targets
#
ALL_TARGET      = $(ASSYM_H) $(KDI_ASSYM_H)

#
# This is DSF_DIR.  Use a short path.
#
DSF_DIR         = .


#
#       Overrides
#
CLEANFILES      = $(GENASSYM) Nothing_to_remove
CLOBBERFILES    = $(ASSYM_H) $(KDI_ASSYM_H) $(CLEANFILES) Nothing_to_remove
```



```
#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

clean.lint:

install:        def

CPPFLAGS += -I../../i86pc/io/ppm

#
# Create assym.h
#
$(GENASSYM): $(GENASSYM_SRC)
        $(NATIVECC) $(NATIVE_CFLAGS) $(ALWAYS_DEFS) $(GENASSYM_DEFS) \
           $(NATIVE_INC_PATH) -o $@ $(GENASSYM_SRC)
        $(NATIVECC) $(ALWAYS_DEFS) $(GENASSYM_DEFS) $(NATIVE_INC_PATH) \
           -o $@ $(GENASSYM_SRC)

$(ASSYM_H): $(OFFSETS_SRC) $(PLATFORM_OFFSETS_SRC) $(GENASSYM)
        $(OFFSETS_CREATE) <$(OFFSETS_SRC) >$@
        $(OFFSETS_CREATE) <$(PLATFORM_OFFSETS_SRC) >>$@
        $(GENASSYM) >>$@

$(KDI_ASSYM_H): $(KDI_OFFSETS_SRC) $(GENASSYM)
        $(OFFSETS_CREATE) <$(KDI_OFFSETS_SRC) >$@
        $(GENASSYM) >>$@

#
#       Include common targets.
#
include $(UTSBASE)/i86pc/Makefile.targ
```

****
6396 Fri Feb 12 11:52:39 2016
new/usr/src/uts/i86pc/ml/genassym.c
6648 illumos build should be explicit about C standards
****

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright (c) 1992, 2010, Oracle and/or its affiliates. All rights reserved.
 */

#ifndef _GENASSYM
#define _GENASSYM
#endif

#define exit    kern_exit

#include <sys/types.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/elf_notes.h>
#include <sys/thread.h>
#include <sys/rwlock.h>
#include <sys/proc.h>
#include <sys/cpuvar.h>
#include <sys/clock.h>
#include <sys/trap.h>
#include <sys/modctl.h>
#include <sys/traptrace.h>
#include <vm/seg.h>
#include <sys/avintr.h>
#include <sys/pic.h>
#include <sys/pit.h>
#include <sys/fp.h>
#include <sys/disp.h>
#include <sys/archsystm.h>
#include <sys/x86_archext.h>
#include <sys/sunddi.h>
#include <sys/mach_mmu.h>

#if defined(__xpv)
#include <sys/hypervisor.h>
#endif

#undef  exit            /* unhide exit, see comment above */
extern void exit(int);

/*
```



```
 * Proactively discourage anyone from referring to structures or
 * member offsets in this program.
 */
#define struct  struct...
#define OFFSET  OFFSET...

int
main(int argc, char *argv[])
{
        printf("#define\tT_AST 0x%x\n", T_AST);

        printf("#define\tLOCK_LEVEL 0x%x\n", LOCK_LEVEL);
        printf("#define\tCLOCK_LEVEL 0x%x\n", CLOCK_LEVEL);
        printf("#define\tDISP_LEVEL 0x%x\n", DISP_LEVEL);
        printf("#define\tPIL_MAX 0x%x\n", PIL_MAX);
        printf("#define\tHIGH_LEVELS 0x%x\n", HIGH_LEVELS);
        printf("#define\tCPU_INTR_ACTV_HIGH_LEVEL_MASK 0x%x\n",
            CPU_INTR_ACTV_HIGH_LEVEL_MASK);

        printf("#define\tPIC_NSEOI 0x%x\n", PIC_NSEOI);
        printf("#define\tPIC_SEOI_LVL7 0x%x\n", PIC_SEOI_LVL7);

        printf("#define\tNANOSEC 0x%llx\n", NANOSEC);
        printf("#define\tNANOSEC 0x%x\n", NANOSEC);
        printf("#define\tADJ_SHIFT 0x%x\n", ADJ_SHIFT);

        printf("#define\tSSLEEP 0x%x\n", SSLEEP);
        printf("#define\tSRUN 0x%x\n", SRUN);
        printf("#define\tSONPROC 0x%x\n", SONPROC);

        printf("#define\tT_INTR_THREAD 0x%x\n", T_INTR_THREAD);
        printf("#define\tFREE_THREAD 0x%x\n", TS_FREE);
        printf("#define\tTS_FREE 0x%x\n", TS_FREE);
        printf("#define\tTS_ZOMB 0x%x\n", TS_ZOMB);
        printf("#define\tTP_MSACCT 0x%x\n", TP_MSACCT);
        printf("#define\tTP_WATCHPT 0x%x\n", TP_WATCHPT);
        printf("#define\tONPROC_THREAD 0x%x\n", TS_ONPROC);

        printf("#define\tS_READ 0x%x\n", (int)S_READ);
        printf("#define\tS_WRITE 0x%x\n", (int)S_WRITE);
        printf("#define\tS_EXEC 0x%x\n", (int)S_EXEC);
        printf("#define\tS_OTHER 0x%x\n", (int)S_OTHER);

        printf("#define\tNORMALRETURN 0x%x\n", (int)NORMALRETURN);
        printf("#define\tLWP_USER 0x%x\n", LWP_USER);
        printf("#define\tLWP_SYS 0x%x\n", LWP_SYS);
        printf("#define\tLMS_USER 0x%x\n", LMS_USER);
        printf("#define\tLMS_SYSTEM 0x%x\n", LMS_SYSTEM);

        printf("#define\tSSE_MXCSR_EFLAGS 0x%x\n", SSE_MXCSR_EFLAGS);

        printf("#define\tFP_487 0x%x\n", FP_487);
        printf("#define\tFP_486 0x%x\n", FP_486);
        printf("#define\tFPU_CW_INIT 0x%x\n", FPU_CW_INIT);
        printf("#define\tFPU_EN 0x%x\n", FPU_EN);
        printf("#define\tFPU_VALID 0x%x\n", FPU_VALID);

        printf("#define\tFP_NO 0x%x\n", FP_NO);
        printf("#define\tFP_SW 0x%x\n", FP_SW);
        printf("#define\tFP_HW 0x%x\n", FP_HW);
        printf("#define\tFP_287 0x%x\n", FP_287);
        printf("#define\tFP_387 0x%x\n", FP_387);
        printf("#define\t__FP_SSE 0x%x\n", __FP_SSE);

        printf("#define\tFP_FNSAVE 0x%x\n", FP_FNSAVE);
        printf("#define\tFP_FXSAVE 0x%x\n", FP_FXSAVE);
```

```
127         printf("#define\tFP_XSAVE 0x%x\n", FP_XSAVE);

129         printf("#define\tAV_INT_SPURIOUS 0x%x\n", AV_INT_SPURIOUS);

131         printf("#define\tCPU_READY 0x%x\n", CPU_READY);
132         printf("#define\tCPU_QUIESCED 0x%x\n", CPU_QUIESCED);

134         printf("#define\tMCMD_PORT 0x%x\n", MCMD_PORT);
135         printf("#define\tSCMD_PORT 0x%x\n", SCMD_PORT);
136         printf("#define\tMIMR_PORT 0x%x\n", MIMR_PORT);
137         printf("#define\tSIMR_PORT 0x%x\n", SIMR_PORT);

139         printf("#define\tDMP_NOSYNC 0x%x\n", DMP_NOSYNC);

141         printf("#define\tRW_WRITER\t0x%x\n", RW_WRITER);
142         printf("#define\tRW_READER\t0x%x\n", RW_READER);

144         printf("#define\tNSYSCALL 0x%x\n", NSYSCALL);

146         printf("#define\tSE_32RVAL1 0x%x\n", SE_32RVAL1);
147         printf("#define\tSE_32RVAL2 0x%x\n", SE_32RVAL2);
148         printf("#define\tSE_64RVAL 0x%x\n", SE_64RVAL);

150         printf("#define\tMAXSYSARGS 0x%x\n", MAXSYSARGS);

152         /* Hack value just to allow clock to be kicked */
153         printf("#define\tNSEC_PER_CLOCK_TICK 0x%llx\n", NANOSEC / 100);
153         printf("#define\tNSEC_PER_CLOCK_TICK 0x%x\n", NANOSEC / 100);

155         printf("#define\tNSEC_PER_COUNTER_TICK 0x%llx\n", NANOSEC / PIT_HZ);
155         printf("#define\tNSEC_PER_COUNTER_TICK 0x%x\n", NANOSEC / PIT_HZ);

157         printf("#define\tPITCTR0_PORT 0x%x\n", PITCTR0_PORT);
158         printf("#define\tPITCTL_PORT 0x%x\n", PITCTL_PORT);
159         printf("#define\tPIT_COUNTDOWN 0x%x\n",
160             PIT_C0 | PIT_LOADMODE | PIT_NDIVMODE);

162         printf("#define\tNBPW 0x%x\n", NBPW);

164         printf("#define\tDDI_ACCATTR_IO_SPACE 0x%x\n", DDI_ACCATTR_IO_SPACE);
165         printf("#define\tDDI_ACCATTR_DIRECT 0x%x\n", DDI_ACCATTR_DIRECT);
166         printf("#define\tDDI_ACCATTR_CPU_VADDR 0x%x\n", DDI_ACCATTR_CPU_VADDR);
167         printf("#define\tDDI_DEV_AUTOINCR 0x%x\n", DDI_DEV_AUTOINCR);

169         printf("#define\tMMU_STD_PAGESIZE 0x%x\n", (uint_t)MMU_STD_PAGESIZE);
170         printf("#define\tMMU_STD_PAGEMASK 0x%x\n", (uint_t)MMU_STD_PAGEMASK);
171         printf("#define\tFOUR_MEG 0x%x\n", (uint_t)FOUR_MEG);

173         printf("#define\tTRAPTR_NENT 0x%x\n", TRAPTR_NENT);

175         printf("#define\tCPU_DTRACE_NOFAULT 0x%x\n", CPU_DTRACE_NOFAULT);
176         printf("#define\tCPU_DTRACE_BADADDR 0x%x\n", CPU_DTRACE_BADADDR);
177         printf("#define\tCPU_DTRACE_ILLOP 0x%x\n", CPU_DTRACE_ILLOP);

179         printf("#define\tMODS_NOUNLOAD 0x%x\n", MODS_NOUNLOAD);
180         printf("#define\tMODS_WEAK 0x%x\n", MODS_WEAK);
181         printf("#define\tMODS_INSTALLED 0x%x\n", MODS_INSTALLED);

183         printf("#define\tKPREEMPT_SYNC 0x%x\n", KPREEMPT_SYNC);

185 #if defined(__xpv)
186         printf("#define\tSHUTDOWN_reboot 0x%x\n", SHUTDOWN_reboot);
187         printf("#define\tSCHEDOP_block 0x%x\n", SCHEDOP_block);
188         printf("#define\tVGCF_IN_KERNEL 0x%x\n", VGCF_IN_KERNEL);
189 #endif
190         return (0);
```



```
191 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7993 Fri Feb 12 11:52:40 2016
new/usr/src/uts/i86xpv/Makefile.rules
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

#
#       This Makefile defines the build rules for the directory uts/i86xpv
#       and its children. These are the source files which are i86xpv
#       "implementation architecture" dependent.
#
#       The following two-level ordering must be maintained in this file.
#
#       - Lines are sorted first in order of decreasing specificity based on
#         the first directory component.  That is, sun4u rules come before
#         sparc rules come before common rules.
#
#       - Lines whose initial directory components are equal are sorted
#         alphabetically by the remaining components.
#


#
#       We share many files with the i86pc implementation to reduce
#       the amount of merge work
#


#
#       Section 1a: C object build rules
#
$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/cpu/generic_cpu/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/io/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```



```
$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/ioat/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/pciex/hotplug/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/io/psm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/ml/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/os/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/vm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86xpv/boot/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/boot/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/gfx_private/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/i86pc/io/xsvc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

#       We need this one to make sure we share dtrace_subr.c with i86pc
#       Otherwise we pick up common/os/dtrace_subr.c instead :(
#       Note that only the non-commented versions of this hack end up
#       in the lint section below.

$(OBJS_DIR)/dtrace_subr.o:      $(UTSBASE)/i86pc/os/dtrace_subr.c
        $(COMPILE.c) -o $@ $(UTSBASE)/i86pc/os/dtrace_subr.c
        $(CTFCONVERT_O)

#
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/cpr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/xen/io/%.c
```

```
128         $(COMPILE.c) -o $@ $<
129         $(CTFCONVERT_O)

131 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/xen/os/%.c
132         $(COMPILE.c) -o $@ $<
133         $(CTFCONVERT_O)

135 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/xen/dtrace/%.c
136         $(COMPILE.c) -o $@ $<
137         $(CTFCONVERT_O)

139 #
140 #       Section 1b: Lint 'object' build rules
141 #
142 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/conf/%.c
143         @($(LHEAD) $(LINT.c) $< $(LTAIL))

145 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/io/%.c
146         @($(LHEAD) $(LINT.c) $< $(LTAIL))

148 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/io/psm/%.c
149         @($(LHEAD) $(LINT.c) $< $(LTAIL))

151 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/ml/%.s
152         @($(LHEAD) $(LINT.s) $< $(LTAIL))

154 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/os/%.c
155         @($(LHEAD) $(LINT.c) $< $(LTAIL))

157 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/vm/%.c
158         @($(LHEAD) $(LINT.c) $< $(LTAIL))

160 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86xpv/boot/%.c
161         @($(LHEAD) $(LINT.c) $< $(LTAIL))

163 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/boot/%.c
164         @($(LHEAD) $(LINT.c) $< $(LTAIL))

166 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/xen/dtrace/%.c
167         @($(LHEAD) $(LINT.c) $< $(LTAIL))

169 #       See note above.

171 $(LINTS_DIR)/dtrace_subr.ln:    $(UTSBASE)/i86pc/os/dtrace_subr.c
172         @($(LHEAD) $(LINT.c) $(UTSBASE)/i86pc/os/dtrace_subr.c $(LTAIL))

174 #
175 # stuff for dboot
176 #
177 # This is different from i86pc - uses 64 bit compiler for 64 bit
178 #
179 # Note that we *don't* want to use the definitions that have been augmented
180 # by various bits of the context of the kernel build environment; so we
181 # start constructing flags and things afresh.

183 DBOOT_OBJS_DIR= dboot/$(OBJS_DIR)

185 DBOOT_XARCH_32 = $(i386_XARCH)
186 DBOOT_XARCH_64 = $(amd64_XARCH)
187 DBOOT_CFLAGS = $(DBOOT_XARCH_$(CLASS)) $(C99MODE) $(CERRWARN) \
188     $(CCNOAUTOINLINE) -O
187 DBOOT_CFLAGS = $(DBOOT_XARCH_$(CLASS)) $(CERRWARN) $(CCNOAUTOINLINE) -O

190 DBOOT_AS_XARCH_32 = $(i386_AS_XARCH)
191 DBOOT_AS_XARCH_64 = $(amd64_AS_XARCH)
192 DBOOT_ASFLAGS = $(DBOOT_AS_XARCH_$(CLASS)) -P -D_ASM
```



```
194 DBOOT_LINTFLAGS_i86xpv  = $(LINTFLAGS_i386_$(CLASS)) $(LINTTAGS_i386_$(CLASS))

196 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/common/xen/os/%.c
197         $(CC) $(DBOOT_CFLAGS) $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

199 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86xpv/boot/%.c
200         $(CC) $(DBOOT_CFLAGS) $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

202 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/boot/%.c
203         $(CC) $(DBOOT_CFLAGS) $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

205 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/dboot/%.c
206         $(CC) $(DBOOT_CFLAGS)  $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

208 $(DBOOT_OBJS_DIR)/%.o:          $(COMMONBASE)/util/%.c
209         $(CC) $(DBOOT_CFLAGS) $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

211 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86xpv/os/%.c
212         $(CC) $(DBOOT_CFLAGS) $(DBOOT_DEFS) $(DBOOT_CC_INCL) -c -o $@ $<

214 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/intel/ia32/ml/%.s
215         $(AS) $(DBOOT_ASFLAGS) $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

217 $(DBOOT_OBJS_DIR)/%.o:          $(COMMONBASE)/util/i386/%.s
218         $(AS) $(DBOOT_ASFLAGS) $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

220 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86xpv/ml/%.s
221         $(AS) $(DBOOT_ASFLAGS) $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

223 $(DBOOT_OBJS_DIR)/%.o:          $(UTSBASE)/i86pc/dboot/%.s
224         $(AS) $(DBOOT_ASFLAGS) $(DBOOT_DEFS) $(DBOOT_AS_INCL) -o $@ $<

226 include $(UTSBASE)/i86pc/Makefile.rules

228 DBOOT_DEFS += -D__xpv

230 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/cpr/%.c
231         @($(LHEAD) $(LINT.c) $< $(LTAIL))

233 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/ioat/%.c
234         @($(LHEAD) $(LINT.c) $< $(LTAIL))

236 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pci/%.c
237         @($(LHEAD) $(LINT.c) $< $(LTAIL))

239 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/pciex/%.c
240         @($(LHEAD) $(LINT.c) $< $(LTAIL))

242 $(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/pciex/hotplug/%.c
243         @($(LHEAD) $(LINT.c) $< $(LTAIL))

245 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/gfx_private/%.c
246         @($(LHEAD) $(LINT.c) $< $(LTAIL))

248 $(LINTS_DIR)/%.ln:              $(UTSBASE)/i86pc/io/xsvc/%.c
249         @($(LHEAD) $(LINT.c) $< $(LTAIL))

251 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/%.c
252         @($(LHEAD) $(LINT.c) $< $(LTAIL))

254 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/xen/io/%.c
255         @($(LHEAD) $(LINT.c) $< $(LTAIL))

257 $(LINTS_DIR)/%.ln:              $(UTSBASE)/common/xen/os/%.c
258         @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
260 $(DBOOT_LINTS_DIR)/%.ln:                $(UTSBASE)/intel/ia32/ml/%.s
261         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

263 $(DBOOT_LINTS_DIR)/%.ln:                $(UTSBASE)/common/xen/os/%.c
264         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

266 $(DBOOT_LINTS_DIR)/%.ln:                $(UTSBASE)/i86xpv/os/%.c
267         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

269 $(DBOOT_LINTS_DIR)/%.ln:                $(UTSBASE)/i86xpv/ml/%.s
270         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))

272 $(DBOOT_LINTS_DIR)/%.ln:                $(UTSBASE)/i86xpv/boot/%.c
273         @($(LHEAD) $(DBOOT_LINT) $(DBOOT_LOCAL_LINTFLAGS) $< $(LTAIL))
```

```
**********************************************************
    2258 Fri Feb 12 11:52:40 2016
new/usr/src/uts/i86xpv/genassym/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
```

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

#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# ident "%Z%%M% %I%     %E% SMI"
#
#       This makefile drives the production of assym.h through
#       compile time intialized data.
#
#       i86xpv implementation architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..

ASSYM_H         = $(DSF_DIR)/$(OBJS_DIR)/assym.h
KDI_ASSYM_H     = $(DSF_DIR)/$(OBJS_DIR)/kdi_assym.h
GENASSYM        = $(DSF_DIR)/$(OBJS_DIR)/genassym

#
#       Include common rules.
#
include $(UTSBASE)/i86xpv/Makefile.i86xpv

#
#       Define targets
#
ALL_TARGET      = $(ASSYM_H) $(KDI_ASSYM_H)

#
# This is DSF_DIR.  Use a short path.
#
DSF_DIR         = .


#
#       Overrides
#
```



```
CLEANFILES      = $(GENASSYM) Nothing_to_remove
CLOBBERFILES    = $(ASSYM_H) $(KDI_ASSYM_H) $(CLEANFILES) Nothing_to_remove

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

clean.lint:

install:        def

#
# Create assym.h
#
$(GENASSYM): $(GENASSYM_SRC)
        $(NATIVECC) $(NATIVE_CFLAGS) $(ALWAYS_DEFS) $(GENASSYM_DEFS) \
            $(NATIVE_INC_PATH) -o $@ $(GENASSYM_SRC)
        $(NATIVECC) $(ALWAYS_DEFS) $(GENASSYM_DEFS) $(NATIVE_INC_PATH) \
           -o $@ $(GENASSYM_SRC)

$(ASSYM_H): $(OFFSETS_SRC) $(PLATFORM_OFFSETS_SRC) $(GENASSYM)
        $(OFFSETS_CREATE) <$(OFFSETS_SRC) >$@
        $(OFFSETS_CREATE) <$(PLATFORM_OFFSETS_SRC) >>$@
        $(GENASSYM) >>$@

$(KDI_ASSYM_H): $(KDI_OFFSETS_SRC) $(GENASSYM)
        $(OFFSETS_CREATE) <$(KDI_OFFSETS_SRC) >$@
        $(GENASSYM) >>$@

#
#       Include common targets.
#
include $(UTSBASE)/i86xpv/Makefile.targ
```

```
**********************************************************
    2831 Fri Feb 12 11:52:41 2016
new/usr/src/uts/intel/audioemu10k/Makefile
6648 illumos build should be explicit about C standards
**********************************************************
```

```
   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 # uts/intel/audioemu10k/Makefile
  23 #
  24 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  25 # Use is subject to license terms.
  26 #
  27 #       This makefile drives the production of the audioemu10k driver.
  28 #

  30 #
  31 #       Path to the base of the uts directory tree (usually /usr/src/uts).
  32 #
  33 UTSBASE = ../..

  35 #
  36 #       Define the module and object file sets.
  37 #
  38 MODULE          = audioemu10k
  39 OBJECTS         = $(AUDIOEMU10K_OBJS:%=$(OBJS_DIR)/%)
  40 LINTS           = $(AUDIOEMU10K_OBJS:%.o=$(LINTS_DIR)/%.ln)
  41 ROOTMODULE      = $(ROOT_DRV_DIR)/$(MODULE)
  42 DSP_SRCDIR      = $(UTSBASE)/common/io/audio/drv/audioemu10k/dsp
  43 DSP_HNAMES      = emu10k_gpr.h emu10k1_dsp.h emu10k2_dsp.h
  44 DSP_HDRS        = $(DSP_HNAMES:%=$(OBJS_DIR)/%)
  45 DSP_SNAMES      = emu10k.dsp emu10k1.mac emu10k2.mac
  46 DSP_SRCS        = $(DSP_SNAMES:%=$(DSP_SRCDIR)/%)
  47 ASM10K          = $(OBJS_DIR)/asm10k

  49 #
  50 #       Include common rules.
  51 #
  52 include $(UTSBASE)/intel/Makefile.intel

  54 #
  55 #       Define targets
  56 #
  57 ALL_TARGET      = $(BINARY)
  58 LINT_TARGET     = $(MODULE).lint
  59 INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

  61 LDFLAGS         += -dy -Ndrv/audio -Nmisc/ac97
```



```
  63 CERRWARN        += -_gcc=-Wno-uninitialized

  65 $(OBJS_DIR)/%.o                 := CPPFLAGS += -I$(OBJS_DIR)
  66 $(OBJS_DIR)/%.ln                := CPPFLAGS += -I$(OBJS_DIR)
  67 $(OBJS_DIR)/emu10k_gpr.h        := ASM10KFLAGS = -v -0 -P gpr
  68 $(OBJS_DIR)/emu10k1_dsp.h       := ASM10KFLAGS = -v -1 -P emu10k1
  69 $(OBJS_DIR)/emu10k2_dsp.h       := MODEL10K = SBLIVE
  70 $(OBJS_DIR)/emu10k2_dsp.h       := ASM10KFLAGS = -v -2 -P emu10k2
  71 $(OBJS_DIR)/emu10k2_dsp.h       := MODEL10K = AUDIGY

  73 #
  74 #       Default build targets.
  75 #
  76 .KEEP_STATE:

  78 def:            $(DEF_DEPS)

  80 all:            $(ALL_DEPS)

  82 clean:          $(CLEAN_DEPS)

  84 clobber:        $(CLOBBER_DEPS)

  86 lint:           $(LINT_DEPS)

  88 modlintlib:     $(MODLINTLIB_DEPS)

  90 clean.lint:     $(CLEAN_LINT_DEPS)

  92 install:        $(INSTALL_DEPS)

  94 $(BINARY):      $(OBJS_DIR)/asm10k $(DSP_HDRS)

  96 $(ASM10K):      $(DSP_SRCDIR)/asm10k.c
  97         $(NATIVECC) $(NATIVE_CFLAGS) -o $@ $(DSP_SRCDIR)/asm10k.c
  97         $(CC) -o $@ $(DSP_SRCDIR)/asm10k.c

  99 $(DSP_HDRS): $(ASM10K) $(DSP_SRCS)
 100         $(CPP) -D$(MODEL10K) -I$(DSP_SRCDIR) $(DSP_SRCDIR)/emu10k.dsp | \
 101         $(OBJS_DIR)/asm10k $(ASM10KFLAGS) -h $@

 103 CLOBBERFILES += $(ASM10K) $(DSP_HDRS)

 105 #
 106 #       Include common targets.
 107 #
 108 include $(UTSBASE)/intel/Makefile.targ
```