

```

*****
35635 Tue Oct 7 15:48:33 2014
new/usr/src/Makefile.master
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2014 Gary Mills
27 #
28 #
29 #
30 # Makefile.master, global definitions for system source
31 #
32 ROOT= /proto
33 #
34 #
35 # Adjunct root, containing an additional proto area to be used for headers
36 # and libraries.
37 #
38 ADJUNCT_PROTO=
39 #
40 #
41 # Adjunct for building things that run on the build machine.
42 #
43 NATIVE_ADJUNCT= /usr
44 #
45 #
46 # RELEASE_BUILD should be cleared for final release builds.
47 # NOT_RELEASE_BUILD is exactly what the name implies.
48 #
49 # __GNUC toggles the building of ON components using gcc and related tools.
50 # Normally set to '#', set it to '' to do gcc build.
51 #
52 # The declaration POUND_SIGN is always '#'. This is needed to get around the
53 # make feature that '#' is always a comment delimiter, even when escaped or
54 # quoted. We use this macro expansion method to get POUND_SIGN rather than
55 # always breaking out a shell because the general case can cause a noticeable
56 # slowdown in build times when so many Makefiles include Makefile.master.
57 #
58 # While the majority of users are expected to override the setting below
59 # with an env file (via nightly or bldenv), if you aren't building that way
60 # (ie, you're using "ws" or some other bootstrapping method) then you need
61 # this definition in order to avoid the subshell invocation mentioned above.

```

```

62 #
63 PRE_POUND= pre\#
64 POUND_SIGN= $(PRE_POUND:pre\%=%)
65 #
66 NOT_RELEASE_BUILD=
67 RELEASE_BUILD= $(POUND_SIGN)
68 $(RELEASE_BUILD)NOT_RELEASE_BUILD= $(POUND_SIGN)
69 PATCH_BUILD= $(POUND_SIGN)
70 #
71 #
72 # SPARC_BLD is '#' for an Intel build.
73 # INTEL_BLD is '#' for a Sparc build.
74 SPARC_BLD_1= $(MACH:i386=$(POUND_SIGN))
75 SPARC_BLD= $(SPARC_BLD_1:sparc=)
76 INTEL_BLD_1= $(MACH:sparc=$(POUND_SIGN))
77 INTEL_BLD= $(INTEL_BLD_1:i386=)
78 #
79 # The variables below control the compilers used during the build.
80 # There are a number of permutations.
81 #
82 # __GNUC and __SUNC control (and indicate) the primary compiler. Whichever
83 # one is not POUND_SIGN is the primary, with the other as the shadow. They
84 # may also be used to control entirely compiler-specific Makefile assignments.
85 # __GNUC and GCC are the default.
86 #
87 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
88 # There is no Sun C analogue.
89 #
90 # The following version-specific options are operative regardless of which
91 # compiler is primary, and control the versions of the given compilers to be
92 # used. They also allow compiler-version specific Makefile fragments.
93 #
94 #
95 __SUNC= $(POUND_SIGN)
96 $(__SUNC)__GNUC= $(POUND_SIGN)
97 __GNUC64= $(__GNUC)
98 #
99 # CLOSED is the root of the tree that contains source which isn't released
100 # as open source
101 CLOSED= $(SRC)/../closed
102 #
103 # BUILD_TOOLS is the root of all tools including compilers.
104 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.
105 #
106 BUILD_TOOLS= /ws/onnv-tools
107 ONBLD_TOOLS= $(BUILD_TOOLS)/onbld
108 #
109 JAVA_ROOT= /usr/java
110 #
111 SFW_ROOT= /usr/sfw
112 SFWINCDIR= $(SFW_ROOT)/include
113 SFWLIBDIR= $(SFW_ROOT)/lib
114 SFWLIBDIR64= $(SFW_ROOT)/lib/$(MACH64)
115 #
116 GCC_ROOT= /opt/gcc/4.4.4
117 GCCLIBDIR= $(GCC_ROOT)/lib
118 GCCLIBDIR64= $(GCC_ROOT)/lib/$(MACH64)
119 #
120 DOCBOOK_XSL_ROOT= /usr/share/sgml/docbook/xsl-stylesheets
121 #
122 RPCGEN= /usr/bin/rpcgen
123 STABS= $(ONBLD_TOOLS)/bin/$(MACH)/stabs
124 ELFXTRACT= $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
125 MBH_PATCH= $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
126 ECHO= echo
127 INS= install

```

```

128 TRUE= true
129 SYMLINK= /usr/bin/ln -s
130 LN= /usr/bin/ln
131 CHMOD= /usr/bin/chmod
132 MV= /usr/bin/mv -f
133 RM= /usr/bin/rm -f
134 CUT= /usr/bin/cut
135 NM= /usr/ccs/bin/nm
136 DIFF= /usr/bin/diff
137 GREP= /usr/bin/grep
138 EGREP= /usr/bin/egrep
139 ELFWRAP= /usr/bin/elfwrap
140 KSH93= /usr/bin/ksh93
141 SED= /usr/bin/sed
142 NAWK= /usr/bin/nawk
143 CP= /usr/bin/cp -f
144 MCS= /usr/ccs/bin/mcs
145 CAT= /usr/bin/cat
146 ELFDUMP= /usr/ccs/bin/elfdump
147 M4= /usr/ccs/bin/m4
148 STRIP= /usr/ccs/bin/strip
149 LEX= /usr/ccs/bin/lex
150 FLEX= $(SFW_ROOT)/bin/flex
151 YACC= /usr/ccs/bin/yacc
152 CPP= /usr/lib/cpp
153 JAVAC= $(JAVA_ROOT)/bin/javac
154 JAVAH= $(JAVA_ROOT)/bin/javah
155 JAVADOC= $(JAVA_ROOT)/bin/javadoc
156 RMIC= $(JAVA_ROOT)/bin/rmic
157 JAR= $(JAVA_ROOT)/bin/jar
158 CTFCONVERT= $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
159 CTFMERGE= $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
160 CTFSTABS= $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
161 CTFSTRIP= $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
162 NDRGEN= $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
163 GENOFFSETS= $(ONBLD_TOOLS)/bin/genoffsets
164 CTFCVTPTBL= $(ONBLD_TOOLS)/bin/ctfcvtptbl
165 CTFINDMOD= $(ONBLD_TOOLS)/bin/ctffindmod
166 XREF= $(ONBLD_TOOLS)/bin/xref
167 FIND= /usr/bin/find
168 PERL= /usr/bin/perl
169 PERL_VERSION= 5.10.0
170 PERL_PKGVERS= -510
171 PYTHON_26= /usr/bin/python2.6
172 PYTHON= $(PYTHON_26)
173 SORT= /usr/bin/sort
174 TOUCH= /usr/bin/touch
175 WC= /usr/bin/wc
176 XARGS= /usr/bin/xargs
177 ELFEDIT= /usr/bin/elfedit
178 ELFSIGN= /usr/bin/elfsign
179 DTRACE= /usr/sbin/dtrace -xnolib
180 UNIQ= /usr/bin/uniq
181 TAR= /usr/bin/tar
182 ASTBINDIR= /usr/ast/bin
183 MSGCC= $(ASTBINDIR)/msgcc

185 FILEMODE= 644
186 DIRMODE= 755

188 #
189 # The version of the patch makeup table optimized for build-time use. Used
190 # during patch builds only.
191 $(PATCH_BUILD)PMTMO_FILE=$(SRC)/patch_makeup_table.mo

193 # Declare that nothing should be built in parallel.

```

```

194 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
195 .NO_PARALLEL:

197 # For stylistic checks
198 #
199 # Note that the X and C checks are not used at this time and may need
200 # modification when they are actually used.
201 #
202 CSTYLE= $(ONBLD_TOOLS)/bin/cstyle
203 CSTYLE_TAIL=
204 HDRCHK= $(ONBLD_TOOLS)/bin/hdrchk
205 HDRCHK_TAIL=
206 JSTYLE= $(ONBLD_TOOLS)/bin/jstyle

208 DOT_H_CHECK= \
209     @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL); \
210     $(HDRCHK) $< $(HDRCHK_TAIL)

212 DOT_X_CHECK= \
213     @$(ECHO) "checking $<"; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
214     $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

216 DOT_C_CHECK= \
217     @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL)

219 MANIFEST_CHECK= \
220     @$(ECHO) "checking $<"; \
221     SVCCFG_DTD=$(SRC)/cmd/svc/dtd/service_bundle.dtd.1 \
222     SVCCFG_REPOSITORY=$(SRC)/cmd/svc/seed/global.db \
223     SVCCFG_CONFIGD_PATH=$(SRC)/cmd/svc/configd/svc.configd-native \
224     $(SRC)/cmd/svc/svccfg/svccfg-native validate $<

226 INS.file= $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
227 INS.dir= $(INS) -s -d -m $(DIRMODE) $@
228 # installs and renames at once
229 #
230 INS.rename= $(INS.file); $(MV) $(@D)/$(<F) $@

232 # install a link
233 INSLINKTARGET= $<
234 INS.link= $(RM) $@; $(LN) $(INSLINKTARGET) $@
235 INS.symlink= $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

237 #
238 # Python bakes the mtime of the .py file into the compiled .pyc and
239 # rebuilds if the baked-in mtime != the mtime of the source file
240 # (rather than only if it's less than), thus when installing python
241 # files we must make certain to not adjust the mtime of the source
242 # (.py) file.
243 #
244 INS.pyfile= $(INS.file); $(TOUCH) -r $< $@

246 # MACH must be set in the shell environment per uname -p on the build host
247 # More specific architecture variables should be set in lower makefiles.
248 #
249 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
250 # architectures on which we do not build 64-bit versions.
251 # (There are no such architectures at the moment.)
252 #
253 # Set BUILD64=# in the environment to disable 64-bit amd64
254 # builds on i386 machines.

256 MACH64_1= $(MACH:sparc=sparcv9)
257 MACH64= $(MACH64_1:i386=amd64)

259 MACH32_1= $(MACH:sparc=sparcv7)

```

```

260 MACH32=          $(MACH32_1:i386=i86)

262 sparc_BUILD64=
263 i386_BUILD64=
264 BUILD64=         $$((MACH)_BUILD64)

266 #
267 # C compiler mode. Future compilers may change the default on us,
268 # so force extended ANSI mode globally. Lower level makefiles can
269 # override this by setting CCMODE.
270 #
271 CCMODE=            -Xa
272 CCMODE64=         -Xa

274 #
275 # C compiler verbose mode. This is so we can enable it globally,
276 # but turn it off in the lower level makefiles of things we cannot
277 # (or aren't going to) fix.
278 #
279 CCVERBOSE=        -v

281 # set this to the secret flag "-Wc,-Qiselect-v9abiwarn=1" to get warnings
282 # from the compiler about places the -xarch=v9 may differ from -xarch=v9c.
283 V9ABIWARN=

285 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
286 # symbols (used to detect conflicts between objects that use global registers)
287 # we disable this now for safety, and because genunix doesn't link with
288 # this feature (the v9 default) enabled.
289 #
290 # REGSYM is separate since the C++ driver syntax is different.
291 CCREGSYM=          -Wc,-Qiselect-regsym=0
292 CCCREGSYM=         -Qoption cg -Qiselect-regsym=0

294 # Prevent the removal of static symbols by the SPARC code generator (cg).
295 # The x86 code generator (ube) does not remove such symbols and as such
296 # using this workaround is not applicable for x86.
297 #
298 CCSTATICSYM=      -Wc,-Qassembler-ounrefsym=0
299 #
300 # generate 32-bit addresses in the v9 kernel. Saves memory.
301 CCABS32=          -Wc,-xcode=abs32
302 #
303 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
304 # system calls.
305 CC32BITCALLERS=   -_gcc=-massume-32bit-callers

307 # GCC, especially, is increasingly beginning to auto-inline functions and
308 # sadly does so separately not under the general -fno-inline-functions
309 # Additionally, we wish to prevent optimisations which cause GCC to clone
310 # functions -- in particular, these may cause unhelpful symbols to be
311 # emitted instead of function names
312 CCNOAUTOINLINE=   -_gcc=-fno-inline-small-functions \
313                   -_gcc=-fno-inline-functions-called-once \
314                   -_gcc=-fno-ipa-cp

316 # One optimization the compiler might perform is to turn this:
317 #     #pragma weak foo
318 #     extern int foo;
319 #     if (&foo)
320 #         foo = 5;
321 # into
322 #     foo = 5;
323 # Since we do some of this (foo might be referenced in common kernel code
324 # but provided only for some cpu modules or platforms), we disable this
325 # optimization.

```

```

326 #
327 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
328 i386_CCUNBOUND =
329 CCUNBOUND =       $$((MACH)_CCUNBOUND)

331 #
332 # compiler '-xarch' flag. This is here to centralize it and make it
333 # overridable for testing.
334 sparc_XARCH=      -m32
335 sparcv9_XARCH=    -m64
336 i386_XARCH=
337 amd64_XARCH=      -m64 -Ui386 -U__i386

339 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
340 sparc_AS_XARCH=   -xarch=v8plus
341 sparcv9_AS_XARCH= -xarch=v9
342 i386_AS_XARCH=
343 amd64_AS_XARCH=   -xarch=amd64 -P -Ui386 -U__i386

345 #
346 # These flags define what we need to be 'standalone' i.e. -not- part
347 # of the rather more cosy userland environment. This basically means
348 # the kernel.
349 #
350 # XX64 future versions of gcc will make -mmodel=kernel imply -mno-red-zone
351 #
352 sparc_STAND_FLAGS= -_gcc=-ffreestanding
353 sparcv9_STAND_FLAGS= -_gcc=-ffreestanding
354 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
355 # additions to SSE (SSE2, AVX ,etc.)
356 NO_SIMD=          -_gcc=-mno-mmx -_gcc=-mno-sse
357 i386_STAND_FLAGS= -_gcc=-ffreestanding $(NO_SIMD)
358 amd64_STAND_FLAGS= -xmodel=kernel $(NO_SIMD)

360 SAVEARGS=         -Wu,-save_args
361 amd64_STAND_FLAGS += $(SAVEARGS)

363 STAND_FLAGS_32 = $$((MACH)_STAND_FLAGS)
364 STAND_FLAGS_64 = $$((MACH64)_STAND_FLAGS)

366 #
367 # disable the incremental linker
368 ILDOFF=           -xildoff
369 #
370 XDEPEND=          -xdepend
371 XFFLAG=           -xF=%all
372 KESS=             -xs
373 XSTRCONST=        -xstrconst

375 #
376 # turn warnings into errors (C)
377 CERRWARN = -errtags=yes -errwarn=%all
378 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
379 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

381 CERRWARN += -_gcc=-Wno-missing-braces
382 CERRWARN += -_gcc=-Wno-sign-compare
383 CERRWARN += -_gcc=-Wno-unknown-pragmas
384 CERRWARN += -_gcc=-Wno-unused-parameter
385 CERRWARN += -_gcc=-Wno-missing-field-initializers

387 # Unfortunately, this option can misfire very easily and unfixably.
388 CERRWARN += -_gcc=-Wno-array-bounds

390 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
391 # -nd builds

```

```

392 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
393 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

395 #
396 # turn warnings into errors (C++)
397 CCERRWARN= -xwe

399 # C99 mode
400 C99_ENABLE= -xc99=%all
401 C99_DISABLE= -xc99=%none
402 C99MODE= $(C99_DISABLE)
403 C99LMODE= $(C99MODE:-xc99%=-Xc99%)

405 # In most places, assignments to these macros should be appended with +=
406 # (CPPFLAGS.master allows values to be prepended to CPPFLAGS).
407 sparc_CFLAGS= $(sparc_XARCH) $(CCSTATICSYM)
408 sparcv9_CFLAGS= $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
409 $(CCSTATICSYM)
410 i386_CFLAGS= $(i386_XARCH)
411 amd64_CFLAGS= $(amd64_XARCH)

413 sparc_ASFLAGS= $(sparc_AS_XARCH)
414 sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
415 i386_ASFLAGS= $(i386_AS_XARCH)
416 amd64_ASFLAGS= $(amd64_AS_XARCH)

418 # Default adjustments to specified gcc optimization level
419 #
420 # For -O and -O1
421 ADJUST_OPT1=
422 # For -O2
423 ADJUST_OPT2= -_gcc=-fno-strict-aliasing \
424 -_gcc=-fno-unit-at-a-time \
425 -_gcc=-fno-optimize-sibling-calls
418 sparc_COPTFLAG= -xO3
419 sparcv9_COPTFLAG= -xO3
420 i386_COPTFLAG= -O
421 amd64_COPTFLAG= -xO3

427 sparc_ADJCOPT= $(ADJUST_OPT2)
428 sparcv9_ADJCOPT= $(ADJUST_OPT2)
429 i386_ADJCOPT= $(ADJUST_OPT1)
430 amd64_ADJCOPT= $(ADJUST_OPT2)
423 COPTFLAG= $(($(MACH)_COPTFLAG)
424 COPTFLAG64= $(($(MACH64)_COPTFLAG)

432 # Symbols that can be redefined in individual Makefiles
433 ADJCOPT= $(($(MACH)_ADJCOPT)
434 ADJCOPT64= $(($(MACH64)_ADJCOPT)

436 # These symbols are sometimes redefined in individual Makefiles
437 sparc_COPTFLAG= -_cc=-xO3 -_gcc=-O2
438 sparcv9_COPTFLAG= -_cc=-xO3 -_gcc=-O2
439 i386_COPTFLAG= -_cc=-O -_gcc=-O
440 amd64_COPTFLAG= -_cc=-xO3 -_gcc=-O2

442 # Documented symbols that can be redefined in individual Makefiles
443 COPTFLAG= $(ADJCOPT) $(($(MACH)_COPTFLAG)
444 COPTFLAG64= $(ADJCOPT64) $(($(MACH64)_COPTFLAG)

446 # When -g is used, the compiler globalizes static objects
447 # (gives them a unique prefix). Disable that.
448 CNOGLOBAL= -W0,-noglobal

450 # Direct the Sun Studio compiler to use a static globalization prefix based on t
451 # name of the module rather than something unique. Otherwise, objects

```

```

452 # will not build deterministically, as subsequent compilations of identical
453 # source will yeild objects that always look different.
454 #
455 # In the same spirit, this will also remove the date from the N_OPT stab.
456 CGLOBALSTATIC= -W0,-xglobalstatic

458 # Sometimes we want all symbols and types in debugging information even
459 # if they aren't used.
460 CALLSYMS= -W0,-xdbggen=no%usedonly

462 #
463 # Default debug format for Sun Studio 11 is dwarf, so force it to
464 # generate stabs.
465 #
466 DEBUGFORMAT= -xdebugformat=stabs

468 #
469 # Flags used to build in debug mode for ctf generation. Bugs in the Devpro
470 # compilers currently prevent us from building with cc-emitted DWARF.
471 #
472 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
473 CTF_FLAGS_i386 = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)

475 CTF_FLAGS_sparcv9 = $(CTF_FLAGS_sparc)
476 CTF_FLAGS_amd64 = $(CTF_FLAGS_i386)

478 # Sun Studio produces broken userland code when saving arguments.
479 $(__GNUCC)CTF_FLAGS_amd64 += $(SAVEARGS)

481 CTF_FLAGS_32 = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
482 CTF_FLAGS_64 = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
483 CTF_FLAGS = $(CTF_FLAGS_32)

485 #
486 # Flags used with genoffsets
487 #
488 GOFLAGS = -_noecho \
489 $(CALLSYMS) \
490 $(CDWARFSTR)

492 OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
493 $(CC) $(GOFLAGS) $(CFLAGS) $(CPPFLAGS)

495 OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
496 $(CC) $(GOFLAGS) $(CFLAGS64) $(CPPFLAGS)

498 #
499 # tradeoff time for space (smaller is better)
500 #
501 sparc_SPACEFLAG = -xspace -W0,-Lt
502 sparcv9_SPACEFLAG = -xspace -W0,-Lt
503 i386_SPACEFLAG = -xspace
504 amd64_SPACEFLAG =

506 SPACEFLAG = $(($(MACH)_SPACEFLAG)
507 SPACEFLAG64 = $(($(MACH64)_SPACEFLAG)

509 #
510 # The Sun Studio 11 compiler has changed the behaviour of integer
511 # wrap arounds and so a flag is needed to use the legacy behaviour
512 # (without this flag panics/hangs could be exposed within the source).
513 #
514 sparc_IROPTFLAG = -W2,-xwrap_int
515 sparcv9_IROPTFLAG = -W2,-xwrap_int
516 i386_IROPTFLAG =
517 amd64_IROPTFLAG =

```

```

519 IROPTFLAG           = $($MACH)_IROPTFLAG)
520 IROPTFLAG64         = $($MACH64)_IROPTFLAG)

522 sparc_XREGSFLAG     = -xregs=no%appl
523 sparcv9_XREGSFLAG   = -xregs=no%appl
524 i386_XREGSFLAG      =
525 amd64_XREGSFLAG     =

527 XREGSFLAG           = $($MACH)_XREGSFLAG)
528 XREGSFLAG64         = $($MACH64)_XREGSFLAG)

530 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
531 # avoids stripping it.
532 SOURCEDEBUG         = $(POUND_SIGN)
533 SRCDBGBLD           = $(SOURCEDEBUG=yes=)

535 #
536 # These variables are intended ONLY for use by developers to safely pass extra
537 # flags to the compilers without unintentionally overriding Makefile-set
538 # flags. They should NEVER be set to any value in a Makefile.
539 #
540 # They come last in the associated FLAGS variable such that they can
541 # explicitly override things if necessary, there are gaps in this, but it's
542 # the best we can manage.
543 #
544 CUSERFLAGS           =
545 CUSERFLAGS64         = $(CUSERFLAGS)
546 CCUSERFLAGS         =
547 CCUSERFLAGS64       = $(CCUSERFLAGS)

549 CSOURCEDEBUGFLAGS   =
550 CCSOURCEDEBUGFLAGS  =
551 $(SRCDBGBLD)CSOURCEDEBUGFLAGS = -g -xs
552 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS = -g -xs

554 CFLAGS=              $(COPTFLAG) $($MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
555                      $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
556                      $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
557                      $(CUSERFLAGS)
558 CFLAGS64=            $(COPTFLAG64) $($MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
559                      $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
560                      $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
561                      $(CUSERFLAGS64)
562 #
563 # Flags that are used to build parts of the code that are subsequently
564 # run on the build machine (also known as the NATIVE_BUILD).
565 #
566 NATIVE_CFLAGS=       $(COPTFLAG) $($NATIVE_MACH)_CFLAGS) $(CCMODE) \
567                      $(ILDOFF) $(CERRWARN) $(C99MODE) $(NATIVE_MACH)_CCUNBOUND) \
568                      $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
569                      $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

571 DTEXTDOM=-DTEXT_DOMAIN="\$(TEXT_DOMAIN)\\" # For messaging.
572 DTS_ERRNO=-D_TS_ERRNO
573 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
574                $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
575                $(ADJUNCT_PROTO:%=-I%/usr/include)
576 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
577                $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
578 CPPFLAGS=            $(CPPFLAGS.master)
579 AS_CPPFLAGS=         $(CPPFLAGS.master)
580 JAVAFLAGS=          -deprecation

582 #
583 # For source message catalogue

```

```

584 #
585 .SUFFIXES: $(SUFFIXES) .i .po
586 MSGROOT= $(ROOT)/catalog
587 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
588 MSGDOMAINDPOFILE = $(MSGDOMAIN)/$(POFILE)
589 DCMMSGDOMAINDPOFILE = $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
590 DCMMSGDOMAINDPOFILE = $(DCMSGDOMAINDPOFILE)/$(DCFILE:.dc=.po)

592 CLOBBERFILES += $(POFILE) $(POFILES)
593 COMPILE.cpp= $(CC) -E -C $(CFLAGS) $(CPPFLAGS)
594 XGETTEXT= /usr/bin/xgettext
595 XGETTEXTFLAGS= -c TRANSLATION_NOTE
596 GNUXGETTEXT= /usr/gnu/bin/xgettext
597 GNUXGETTEXTFLAGS= --add-comments=TRANSLATION_NOTE --keyword=_ \
598                  --strict --no-location --omit-header
599 BUILD.po= $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $<.i ;\
600           $(RM) $@ ;\
601           $(SED) "/^domain/d" < $(<F).po > $@ ;\
602           $(RM) $(<F).po $<.i

604 #
605 # This is overwritten by local Makefile when PROG is a list.
606 #
607 POFILE= $(PROG).po

609 sparc_CCFLAGS=       -cg92 -compat=4 \
610                      -Qoption ccfe -messages=no%anachronism \
611                      $(CCERRWARN)
612 sparcv9_CCFLAGS=    $(sparcv9_XARCH) -dalign -compat=5 \
613                      -Qoption ccfe -messages=no%anachronism \
614                      -Qoption ccfe -features=no%conststrings \
615                      $(CCREGSYM) \
616                      $(CCERRWARN)
617 i386_CCFLAGS=       -compat=4 \
618                      -Qoption ccfe -messages=no%anachronism \
619                      -Qoption ccfe -features=no%conststrings \
620                      $(CCERRWARN)
621 amd64_CCFLAGS=      $(amd64_XARCH) -compat=5 \
622                      -Qoption ccfe -messages=no%anachronism \
623                      -Qoption ccfe -features=no%conststrings \
624                      $(CCERRWARN)

626 sparc_CCOPTFLAG=    -_CC=-O -_g++=-O
627 sparcv9_CCOPTFLAG=  -_CC=-O -_g++=-O
628 i386_CCOPTFLAG=     -_CC=-O -_g++=-O
629 amd64_CCOPTFLAG=    -_CC=-O -_g++=-O
630 sparc_CCOPTFLAG=    -O
631 sparcv9_CCOPTFLAG=  -O
632 i386_CCOPTFLAG=     -O
633 amd64_CCOPTFLAG=    -O

631 ADJCCOPT=           $(ADJUST_OPT1)
632 ADJCCOPT64=         $(ADJUST_OPT1)
631 CCOPTFLAG=          $(MACH)_CCOPTFLAG)
632 CCOPTFLAG64=        $(MACH64)_CCOPTFLAG)
631 CCFLAGS=            $(COPTFLAG) $(MACH)_CCFLAGS) $(CSOURCEDEBUGFLAGS) \
632                      $(CUSERFLAGS)
631 CCFLAGS64=          $(COPTFLAG64) $(MACH64)_CCFLAGS) $(CSOURCEDEBUGFLAGS) \
632                      $(CCUSERFLAGS64)

634 CCOPTFLAG=          $(ADJCCOPT) $(MACH)_CCOPTFLAG)
635 CCOPTFLAG64=        $(ADJCCOPT) $(MACH64)_CCOPTFLAG)
636 CCFLAGS=            $(COPTFLAG) $(MACH)_CCFLAGS) \
637                      $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)
638 CCFLAGS64=          $(COPTFLAG64) $(MACH64)_CCFLAGS) \
639                      $(CSOURCEDEBUGFLAGS) $(CCUSERFLAGS64)

```

```

641 #
642 #
643 #
644 ELFWRAP_FLAGS =
645 ELFWRAP_FLAGS64 = -64

647 #
648 # Various mapfiles that are used throughout the build, and delivered to
649 # /usr/lib/ld.
650 #
651 MAPFILE.NED_i386 = $(SRC)/common/mapfiles/common/map.noexdata
652 MAPFILE.NED_sparc =
653 MAPFILE.NED = $(MAPFILE.NED_$(MACH))
654 MAPFILE.PGA = $(SRC)/common/mapfiles/common/map.pagealign
655 MAPFILE.NES = $(SRC)/common/mapfiles/common/map.noexstk
656 MAPFILE.FLT = $(SRC)/common/mapfiles/common/map.filter
657 MAPFILE.LEX = $(SRC)/common/mapfiles/common/map.lex.yy

659 #
660 # Generated mapfiles that are compiler specific, and used throughout the
661 # build. These mapfiles are not delivered in /usr/lib/ld.
662 #
663 MAPFILE.NGB_sparc= $(SRC)/common/mapfiles/gen/sparc_cc_map.noexglobs
664 $(__GNUC64)MAPFILE.NGB_sparc= \
665 $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexglobs
666 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexglobs
667 $(__GNUC64)MAPFILE.NGB_sparcv9= \
668 $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexglobs
669 MAPFILE.NGB_i386= $(SRC)/common/mapfiles/gen/i386_cc_map.noexglobs
670 $(__GNUC64)MAPFILE.NGB_i386= \
671 $(SRC)/common/mapfiles/gen/i386_gcc_map.noexglobs
672 MAPFILE.NGB_amd64= $(SRC)/common/mapfiles/gen/amd64_cc_map.noexglobs
673 $(__GNUC64)MAPFILE.NGB_amd64= \
674 $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexglobs
675 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH))

677 #
678 # A generic interface mapfile name, used by various dynamic objects to define
679 # the interfaces and interposers the object must export.
680 #
681 MAPFILE.INT = mapfile-intf

683 #
684 # LDLIBS32 and LDLIBS64 can be set in the environment to override the following
685 # assignments.
686 #
687 # These environment settings make sure that no libraries are searched outside
688 # of the local workspace proto area:
689 # LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
690 # LDLIBS64=-YP,$ROOT/lib:$MACH64:$ROOT/usr/lib:$MACH64
691 #
692 LDLIBS32 = $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
693 LDLIBS32 += $(ADJUNCT_PROTO:%=-L%/usr/lib -L%/lib)
694 LDLIBS.cmd = $(LDLIBS32)
695 LDLIBS.lib = $(LDLIBS32)

697 LDLIBS64 = $(ENVLDLIBS1:%=%/$(MACH64)) \
698 $(ENVLDLIBS2:%=%/$(MACH64)) \
699 $(ENVLDLIBS3:%=%/$(MACH64))
700 LDLIBS64 += $(ADJUNCT_PROTO:%=-L%/usr/lib/$(MACH64) -L%/lib/$(MACH64))

702 #
703 # Define compilation macros.
704 #
705 COMPILE.c= $(CC) $(CFLAGS) $(CPPFLAGS) -c

```

```

706 COMPILE64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) -c
707 COMPILE.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
708 COMPILE64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
709 COMPILE.s= $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
710 COMPILE64.s= $(AS) $(ASFLAGS) $(MACH64)_AS_XARCH $(AS_CPPFLAGS)
711 COMPILE.d= $(DTRACE) -G -32
712 COMPILE64.d= $(DTRACE) -G -64
713 COMPILE.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
714 COMPILE64.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

716 CLASSPATH= .
717 COMPILE.java= $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

719 #
720 # Link time macros
721 #
722 CCNEEDED = -lC
723 CCEXTNEEDED = -lCrun -lCstd
724 $(__GNUC)CCNEEDED = -L$(GCCLIBDIR) -lstc++ -lgcc_s
725 $(__GNUC)CCEXTNEEDED = $(CCNEEDED)

727 LINK.c= $(CC) $(CFLAGS) $(CPPFLAGS) $(LDLFLAGS)
728 LINK64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDLFLAGS)
729 NORUNPATH= -norunpath -nolib
730 LINK.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
731 $(LDLFLAGS) $(CCNEEDED)
732 LINK64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
733 $(LDLFLAGS) $(CCNEEDED)

735 #
736 # lint macros
737 #
738 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
739 # ON is built with a version of lint that has the fix for 4484186.
740 #
741 ALWAYS_LINT_DEFS = -errtags=yes -s
742 ALWAYS_LINT_DEFS += -erroff=E_PTRDIFF_OVERFLOW
743 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_NARROW_CONV
744 ALWAYS_LINT_DEFS += -U__PRAGMA_REDEFINE_EXTNAME
745 ALWAYS_LINT_DEFS += $(C99LMODE)
746 ALWAYS_LINT_DEFS += -errsecurity=$(SECLEVEL)
747 ALWAYS_LINT_DEFS += -erroff=E_SEC_CREAT_WITHOUT_EXCL
748 ALWAYS_LINT_DEFS += -erroff=E_SEC_FORBIDDEN_WARN_CREAT
749 # XX64 -- really only needed for amd64 lint
750 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_INT_TO_SMALL_INT
751 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_CONST_TO_SMALL_INT
752 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_TO_SMALL_INT
753 ALWAYS_LINT_DEFS += -erroff=E_CAST_TO_PTR_FROM_INT
754 ALWAYS_LINT_DEFS += -erroff=E_COMP_INT_WITH_LARGE_INT
755 ALWAYS_LINT_DEFS += -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
756 ALWAYS_LINT_DEFS += -erroff=E_PASS_INT_TO_SMALL_INT
757 ALWAYS_LINT_DEFS += -erroff=E_PTR_CONV_LOSES_BITS

759 # This forces lint to pick up note.h and sys/note.h from Devpro rather than
760 # from the proto area. The note.h that ON delivers would disable NOTE().
761 ONLY_LINT_DEFS = -I$(SPRO_VROOT)/prod/include/lint

763 SECLEVEL= core
764 LINT.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
765 $(ALWAYS_LINT_DEFS)
766 LINT64.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
767 $(ALWAYS_LINT_DEFS)
768 LINT.s= $(LINT.c)

770 # For some future builds, NATIVE_MACH and MACH might be different.
771 # Therefore, NATIVE_MACH needs to be redefined in the

```

```

772 # environment as 'uname -p' to override this macro.
773 #
774 # For now at least, we cross-compile amd64 on i386 machines.
775 NATIVE_MACH= $(MACH:amd64=i386)

777 # Define native compilation macros
778 #

780 # Base directory where compilers are loaded.
781 # Defined here so it can be overridden by developer.
782 #
783 SPRO_ROOT= $(BUILD_TOOLS)/SUNWsprio
784 SPRO_VROOT= $(SPRO_ROOT)/SS12
785 GNU_ROOT= $(SFW_ROOT)

787 # Till SS12ul formally becomes the NV CBE, LINT is hard
788 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
789 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
790 # i386_LINT, amd64_LINT.
791 # Reset them when SS12ul is rolled out.
792 #

794 # Specify platform compiler versions for languages
795 # that we use (currently only c and c++).
796 #
797 sparc_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _cc
798 $(__GNUC)sparc_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _gcc
799 sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _CC
800 $(__GNUC)sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _g++
801 sparc_CPP= /usr/ccs/lib/cpp
802 sparc_AS= /usr/ccs/bin/as -xregsym=no
803 sparc_LD= /usr/ccs/bin/ld
804 sparc_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

806 sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _cc
807 $(__GNUC64)sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _gcc
808 sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _CC
809 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _g++
810 sparcv9_CPP= /usr/ccs/lib/cpp
811 sparcv9_AS= /usr/ccs/bin/as -xregsym=no
812 sparcv9_LD= /usr/ccs/bin/ld
813 sparcv9_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

815 i386_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _cc
816 $(__GNUC)i386_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _gcc
817 i386_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _CC
818 $(__GNUC)i386_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _g++
819 i386_CPP= /usr/ccs/lib/cpp
820 i386_AS= /usr/ccs/bin/as
821 $(__GNUC)i386_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
822 i386_LD= /usr/ccs/bin/ld
823 i386_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

825 amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _cc
826 $(__GNUC64)amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _gcc
827 amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _CC
828 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw _g++
829 amd64_CPP= /usr/ccs/lib/cpp
830 amd64_AS= $(ONBLD_TOOLS)/bin/$(MACH)/aw
831 amd64_LD= /usr/ccs/bin/ld
832 amd64_LINT= $(SPRO_ROOT)/sunstudio12.1/bin/lint

834 NATIVECC= $( $(NATIVE_MACH)_CC )
835 NATIVECCC= $( $(NATIVE_MACH)_CCC )
836 NATIVECPP= $( $(NATIVE_MACH)_CPP )
837 NATIVEAS= $( $(NATIVE_MACH)_AS )

```

```

838 NATIVELD= $( $(NATIVE_MACH)_LD )
839 NATIVELINT= $( $(NATIVE_MACH)_LINT )

841 #
842 # Makefile.master.64 overrides these settings
843 #
844 CC= $(NATIVECC)
845 CCC= $(NATIVECCC)
846 CPP= $(NATIVECPP)
847 AS= $(NATIVEAS)
848 LD= $(NATIVELD)
849 LINT= $(NATIVELINT)

851 # The real compilers used for this build
852 CW_CC_CMD= $(CC) _compiler
853 CW_CCC_CMD= $(CCC) _compiler
854 REAL_CC= $(CW_CC_CMD:sh)
855 REAL_CCC= $(CW_CCC_CMD:sh)

857 # Pass -Y flag to cpp (method of which is release-dependent)
858 CCYFLAG= -Y I,

860 BDIRECT= -Bdirect
861 BDYNAMIC= -Bdynamic
862 BLOCAL= -Blocal
863 BNODIRECT= -Bnodirect
864 BREDUCE= -Breduce
865 BSTATIC= -Bstatic

867 ZDEFS= -zdefs
868 ZDIRECT= -zdirect
869 ZIGNORE= -zignore
870 ZINITFIRST= -zinitfirst
871 ZINTERPOSE= -zinterpose
872 ZLAZYLOAD= -zlazyload
873 ZLOADFLTR= -zloadfltr
874 ZMULDEFS= -zmuldefs
875 ZNODEFAULTLIB= -znofaultlib
876 ZNODEFS= -znofdefs
877 ZNODELETE= -znodelete
878 ZNODLOPEN= -znodlopen
879 ZNODUMP= -znodump
880 ZNOLAZYLOAD= -znolazyload
881 ZNOLDYNSYM= -znolddynsym
882 ZNORELOC= -znoreloc
883 ZNOVERSION= -znoversion
884 ZRECORD= -zrecord
885 ZREDLOCSYM= -zredlocsymb
886 ZTEXT= -ztext
887 ZVERBOSE= -zverbose

889 GSHARED= -G
890 CCMT= -mt

892 # Handle different PIC models on different ISAs
893 # (May be overridden by lower-level Makefiles)

895 sparc_C_PICFLAGS = -K pic
896 sparcv9_C_PICFLAGS = -K pic
897 i386_C_PICFLAGS = -K pic
898 amd64_C_PICFLAGS = -K pic
899 C_PICFLAGS = $( $(MACH)_C_PICFLAGS )
900 C_PICFLAGS64 = $( $(MACH64)_C_PICFLAGS )

902 sparc_C_BIGPICFLAGS = -K PIC
903 sparcv9_C_BIGPICFLAGS = -K PIC

```

```

904 i386_C_BIGPICFLAGS = -K PIC
905 amd64_C_BIGPICFLAGS = -K PIC
906 C_BIGPICFLAGS = $( $(MACH)_C_BIGPICFLAGS )
907 C_BIGPICFLAGS64 = $( $(MACH64)_C_BIGPICFLAGS )

909 # CC requires there to be no space between '-K' and 'pic' or 'PIC'.
910 sparc_CC_PICFLAGS = -Kpic
911 sparcv9_CC_PICFLAGS = -KPIC
912 i386_CC_PICFLAGS = -Kpic
913 amd64_CC_PICFLAGS = -Kpic
914 CC_PICFLAGS = $( $(MACH)_CC_PICFLAGS )
915 CC_PICFLAGS64 = $( $(MACH64)_CC_PICFLAGS )

917 AS_PICFLAGS= $(C_PICFLAGS)
918 AS_BIGPICFLAGS= $(C_BIGPICFLAGS)

920 #
921 # Default label for CTF sections
922 #
923 CTFCVTFLAGS= -i -L VERSION
924 $(SRCSBGBLD)CTFCVTFLAGS += -g

926 #
927 # Override to pass module-specific flags to ctfmerge. Currently used only by
928 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
929 # stripping.
930 #
931 CTFMRGFLAGS=
932 $(SRCSBGBLD)CTFMRGFLAGS += -g

935 CTFCONVERT_O = $(CTFCONVERT) $(CTFCVTFLAGS) $@

937 ELFSIGN_O= $(TRUE)
938 ELFSIGN_CRYPTO= $(ELFSIGN_O)
939 ELFSIGN_OBJECT= $(ELFSIGN_O)

941 # Rules (normally from make.rules) and macros which are used for post
942 # processing files. Normally, these do stripping of the comment section
943 # automatically.
944 # RELEASE_CM: Should be edited to reflect the release.
945 # POST_PROCESS_O: Post-processing for '.o' files.
946 # POST_PROCESS_A: Post-processing for '.a' files (currently null).
947 # POST_PROCESS_SO: Post-processing for '.so' files.
948 # POST_PROCESS: Post-processing for executable files (no suffix).
949 # Note that these macros are not completely generalized as they are to be
950 # used with the file name to be processed following.
951 #
952 # It is left as an exercise to Release Engineering to embellish the generation
953 # of the release comment string.
954 #
955 # If this is a standard development build:
956 # compress the comment section (mcs -c)
957 # add the standard comment (mcs -a $(RELEASE_CM))
958 # add the development specific comment (mcs -a $(DEV_CM))
959 #
960 # If this is an installation build:
961 # delete the comment section (mcs -d)
962 # add the standard comment (mcs -a $(RELEASE_CM))
963 # add the development specific comment (mcs -a $(DEV_CM))
964 #
965 # If this is an release build:
966 # delete the comment section (mcs -d)
967 # add the standard comment (mcs -a $(RELEASE_CM))
968 #
969 # The following list of macros are used in the definition of RELEASE_CM

```

```

970 # which is used to label all binaries in the build:
971 #
972 # RELEASE Specific release of the build, eg: 5.2
973 # RELEASE_MAJOR Major version number part of $(RELEASE)
974 # RELEASE_MINOR Minor version number part of $(RELEASE)
975 # VERSION Version of the build (alpha, beta, Generic)
976 # PATCHID If this is a patch this value should contain
977 # the patchid value (eg: "Generic 100832-01"), otherwise
978 # it will be set to $(VERSION)
979 # RELEASE_DATE Date of the Release Build
980 # PATCH_DATE Date the patch was created, if this is blank it
981 # will default to the RELEASE_DATE
982 #
983 RELEASE_MAJOR= 5
984 RELEASE_MINOR= 11
985 RELEASE= $(RELEASE_MAJOR).$(RELEASE_MINOR)
986 VERSION= SunOS Development
987 PATCHID= $(VERSION)
988 RELEASE_DATE= release date not set
989 PATCH_DATE= $(RELEASE_DATE)
990 RELEASE_CM= "@$(POUND_SIGN)SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
991 DEV_CM= "@$(POUND_SIGN)SunOS Internal Development: non-nightly build"

993 PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
994 $(RELEASE_BUILD)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM)

996 STRIP_STABS= :
997 $(RELEASE_BUILD)STRIP_STABS= $(STRIP) -x $@
998 $(SRCSBGBLD)STRIP_STABS= :

1000 POST_PROCESS_O= $(PROCESS_COMMENT) $@
1001 POST_PROCESS_A=
1002 POST_PROCESS_SO= $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
1003 $(ELFSIGN_OBJECT)
1004 POST_PROCESS= $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
1005 $(ELFSIGN_OBJECT)

1007 #
1008 # chk4ubin is a tool that inspects a module for a symbol table
1009 # ELF section size which can trigger an OBP bug on older platforms.
1010 # This problem affects only specific sun4u bootable modules.
1011 #
1012 CHK4UBIN= $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
1013 CHK4UBINFLAGS=
1014 CHK4UBINARY= $(CHK4UBIN) $(CHK4UBINFLAGS) $@

1016 #
1017 # PKGARCHIVE specifies the default location where packages should be
1018 # placed if built.
1019 #
1020 $(RELEASE_BUILD)PKGARCHIVESUFFIX= -nd
1021 PKGARCHIVE=$(SRC)/../../../../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)

1023 #
1024 # The repositories will be created with these publisher settings. To
1025 # update an image to the resulting repositories, this must match the
1026 # publisher name provided to "pkg set-publisher."
1027 #
1028 PKGPUBLISHER_REDIST= on-nightly
1029 PKGPUBLISHER_NONREDIST= on-extra

1031 # Default build rules which perform comment section post-processing.
1032 #
1033 .c:
1034 $(LINK.c) -o $@ $< $(LDLIBS)
1035 $(POST_PROCESS)

```



```

1036 .c.o:
1037     $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1038     $(POST_PROCESS_O)
1039 .c.a:
1040     $(COMPILE.c) -o $$ $<
1041     $(PROCESS_COMMENT) $$
1042     $(AR) $(ARFLAGS) $$@ $$
1043     $(RM) $$
1044 .s.o:
1045     $(COMPILE.s) -o $$@ $<
1046     $(POST_PROCESS_O)
1047 .s.a:
1048     $(COMPILE.s) -o $$ $<
1049     $(PROCESS_COMMENT) $$
1050     $(AR) $(ARFLAGS) $$@ $$
1051     $(RM) $$
1052 .cc:
1053     $(LINK.cc) -o $$@ $< $(LDLIBS)
1054     $(POST_PROCESS)
1055 .cc.o:
1056     $(COMPILE.cc) $(OUTPUT_OPTION) $<
1057     $(POST_PROCESS_O)
1058 .cc.a:
1059     $(COMPILE.cc) -o $$ $<
1060     $(AR) $(ARFLAGS) $$@ $$
1061     $(PROCESS_COMMENT) $$
1062     $(RM) $$
1063 .y:
1064     $(YACC.y) $<
1065     $(LINK.c) -o $$@ y.tab.c $(LDLIBS)
1066     $(POST_PROCESS)
1067     $(RM) y.tab.c
1068 .y.o:
1069     $(YACC.y) $<
1070     $(COMPILE.c) -o $$@ y.tab.c $(CTFCONVERT_HOOK)
1071     $(POST_PROCESS_O)
1072     $(RM) y.tab.c
1073 .l:
1074     $(RM) $*.c
1075     $(LEX.l) $< > $*.c
1076     $(LINK.c) -o $$@ $*.c -ll $(LDLIBS)
1077     $(POST_PROCESS)
1078     $(RM) $*.c
1079 .l.o:
1080     $(RM) $*.c
1081     $(LEX.l) $< > $*.c
1082     $(COMPILE.c) -o $$@ $*.c $(CTFCONVERT_HOOK)
1083     $(POST_PROCESS_O)
1084     $(RM) $*.c

1086 .bin.o:
1087     $(COMPILE.b) -o $$@ $<
1088     $(POST_PROCESS_O)

1090 .java.class:
1091     $(COMPILE.java) $<

1093 # Bourne and Korn shell script message catalog build rules.
1094 # We extract all gettext strings with sed(1) (being careful to permit
1095 # multiple gettext strings on the same line), weed out the dups, and
1096 # build the catalogue with awk(1).

1098 .sh.po .ksh.po:
1099     $(SED) -n -e ":a" \
1100     -e "h" \
1101     -e "s/.*gettext *\([^\"]*\)*\([^\"]*\).*\/\1/p" \

```

```

1102     -e "x" \
1103     -e "s/\(.*\)gettext *\[^\"]*\\"*\([^\"]*\).*\/\1\2/" \
1104     -e "t a" \
1105     $< | sort -u | awk '{ print "msgid\t" $$0 "\nmsgstr" }' > $$@

1107 #
1108 # Python and Perl executable and message catalog build rules.
1109 #
1110 .SUFFIXES: .pl .pm .py .pyc

1112 .pl:
1113     $(RM) $$@;
1114     $(SED) -e "s@TEXT_DOMAIN@"$(TEXT_DOMAIN)"@" $< > $$@;
1115     $(CHMOD) +x $$@

1117 .py:
1118     $(RM) $$@; $(CAT) $< > $$@; $(CHMOD) +x $$@

1120 .py.pyc:
1121     $(RM) $$@
1122     $(PYTHON) -mpy_compile $<
1123     @[ $(<)c = $@ ] || $(MV) $(<)c $$@

1125 .py.po:
1126     $(GNUXGETTEXT) $(GNUXGETTEXTFLAGS) -d $(<F:%.py=) $< ;

1128 .pl.po .pm.po:
1129     $(XGETTEXT) $(XGETTEXTFLAGS) -d $(<F) $< ;
1130     $(RM) $$@ ;
1131     $(SED) "/^domain/d" < $(<F).po > $$@ ;
1132     $(RM) $(<F).po

1134 #
1135 # When using xgettext, we want messages to go to the default domain,
1136 # rather than the specified one. This special version of the
1137 # COMPILER.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1138 # causing xgettext to put all messages into the default domain.
1139 #
1140 CPPFORPO=$(COMPILE.cpp:"$(TEXT_DOMAIN)"=TEXT_DOMAIN)

1142 .c.i:
1143     $(CPPFORPO) $< > $$@

1145 .h.i:
1146     $(CPPFORPO) $< > $$@

1148 .y.i:
1149     $(YACC) -d $<
1150     $(CPPFORPO) y.tab.c > $$@
1151     $(RM) y.tab.c

1153 .l.i:
1154     $(LEX) $<
1155     $(CPPFORPO) lex.yy.c > $$@
1156     $(RM) lex.yy.c

1158 .c.po:
1159     $(CPPFORPO) $< > $<.i
1160     $(BUILD.po)

1162 .y.po:
1163     $(YACC) -d $<
1164     $(CPPFORPO) y.tab.c > $<.i
1165     $(BUILD.po)
1166     $(RM) y.tab.c

```

**new/usr/src/Makefile.master**

19

```
1168 .l.po:
1169     $(LEX) $<
1170     $(CPPFORPO) lex.yy.c > $<.i
1171     $(BUILD.po)
1172     $(RM) lex.yy.c

1174 #
1175 # Rules to perform stylistic checks
1176 #
1177 .SUFFIXES: .x .xml .check .xmlchk

1179 .h.check:
1180     $(DOT_H_CHECK)

1182 .x.check:
1183     $(DOT_X_CHECK)

1185 .xml.xmlchk:
1186     $(MANIFEST_CHECK)

1188 #
1189 # Include rules to render automated sccs get rules "safe".
1190 #
1191 include $(SRC)/Makefile.noget
```

```

*****
3062 Tue Oct 7 15:48:34 2014
new/usr/src/cmd/dtrace/test/tst/Makefile.com
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #

28 include $(SRC)/cmd/Makefile.cmd

30 .KEEP_STATE:

32 ROOTOPTPKG = $(ROOT)/opt/SUNWdtrt
33 ROOTTST = $(ROOTOPTPKG)/tst
34 SUBDIR :sh= basename `pwd`
35 TSTDIR = $(ROOTTST)/$(SUBDIR)
36 DSTYLE = $(ROOTOPTPKG)/bin/dstyle

38 CSRCS :sh= find . -name SCCS -prune -o -name *.c -print | cut -b3-
39 SSRCS :sh= find . -name SCCS -prune -o -name *.s -print | cut -b3-
40 DSRCS :sh= find . -name SCCS -prune -o -name *.d -print | cut -b3-

42 TSTS :sh= find . -name tst.*.d -o -name err.*.d -o \
43     -name tst.*.d.out -o -name err.*.d.out -o -name tst.*.ksh \
44     -o -name err.*.ksh -o -name tst.*.ksh.out -o -name drp.*.d \
45     -o -name get.*.pl

47 EXES :sh= find . -name SCCS -prune -o \( -name *.exe -o -name *.pl \) -print \
48     | cut -b3-
49 EXES += $(CSRCS:%.c=%.exe)
50 EXES += $(SSRCS:%.s=%.exe)

52 ROOT_TSTS = $(TSTS:%=$(TSTDIR)/%)
53 ROOT_EXES = $(EXES:%=$(TSTDIR)/%)

55 $(ROOT_TSTS) := FILEMODE = 0444
56 $(ROOT_EXES) := FILEMODE = 0555

58 # The DTrace tests rely on "normal" behaviour from the compiler which
59 # aggressive optimization of small, simple, one compilation-unit programs may
60 # utterly subvert. We force the compiler to not optimize rather than engage
61 # in an arms race with increasingly belligerent optimizers.

```

```

62 COPTFLAG=      _cc=-xO0  _gcc=-O0
61 COPTFLAG=      -xO0

64 CERRWARN +=    _gcc=-Wno-switch
65 CERRWARN +=    _gcc=-Wno-unused-variable
66 CERRWARN +=    _gcc=-Wno-implicit-function-declaration
67 CERRWARN +=    _gcc=-Wno-unused-function
68 CERRWARN +=    _gcc=-Wno-unused-variable

70 all: $(EXES)

72 clean lint:

74 clobber: FRC
75     -$(RM) $(CSRCS:%.c=%.exe) $(CSRCS:%.c=%.o)
76     -$(RM) $(SSRCS:%.s=%.exe) $(SSRCS:%.s=%.o)
77     -$(RM) $(DSRCS:%.d=%.o)
78     -$(RM) $(CLOBBERFILES)

80 install: $(ROOT_TSTS) $(ROOT_EXES)

82 $(ROOT_TSTS): $(TSTDIR)

84 $(ROOT_EXES): $(TSTDIR)

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

89 $(TSTDIR)/%: %
90     $(INS) -s -d -m $(DIRMODE) $(@D)
91     $(INS.file)

93 %.exe: %.c
94     $(LINK.c) -o $@ $< $(LDLIBS)
95     $(POST_PROCESS) ; $(STRIP_STABS)

97 %.exe: %.o
98     $(LINK.c) -o $@ $< $(LDLIBS)
99     $(POST_PROCESS) ; $(STRIP_STABS)

101 %.o: %.c
102     $(COMPILE.c) -o $@ $<
103     $(POST_PROCESS_O)

105 %.o: %.s
106     $(COMPILE.s) -o $@ $<
107     $(POST_PROCESS_O)

109 scripts: FRC
110     @cd ../cmd/scripts; pwd; $(MAKE) install

112 dstyle: FRC
113     @if [ -n "$(DSRCS)" ]; then $(DSTYLE) $(DSRCS); fi

115 FRC:

```

```

*****
5569 Tue Oct 7 15:48:34 2014
new/usr/src/cmd/dtrace/test/tst/common/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #
27 #
28 # Copyright (c) 2012 by Delphix. All rights reserved.
29 # Copyright (c) 2013, Joyent, Inc. All rights reserved.
30 # Copyright 2014 Gary Mills
31 #
32 #
33 #
34 # WARNING: Do not include Makefile.ctf here. That will cause tests to
35 # break.
36 #
37 #
38 include $(SRC)/Makefile.master
39 include ../Makefile.com
40 #
41 SNOOPDIR = $(SRC)/cmd/cmd-inet/usr.sbin/snoop
42 SNOOPOBJS = nfs4_xdr.o
43 SNOOPSRCS = ${SNOOPOBJS:.o=.c}
44 CLOBBERFILES += nfs/$(SNOOPOBJS)
45 #
46 RPCSVCDIR = $(SRC)/head/rpcsvc
47 RPCSVCOBJS = nfs_prot.o
48 RPCSVCSRCS = ${RPCSVCOBJS:%o=%c}
49 CLOBBERFILES += nfs/$(RPCSVCOBJS) $(RPCSVCDIR)/$(RPCSVCSRCS)
50 CLOBBERFILES += usdt/forker.h usdt/lazyprobe.h
51 #
52 fasttrap/tst.fasttrap.exe := LDLIBS += -ldtrace
53 fasttrap/tst.stack.exe := LDLIBS += -ldtrace
54 #
55 sysevent/tst.post.exe := LDLIBS += -lsysevent
56 sysevent/tst.post_chan.exe := LDLIBS += -lsysevent
57 #
58 ustack/tst.bigstack.exe := COPTFLAG = $(ADJUST_OPT1) -_cc=-x01 -_gcc=-O1
59 ustack/tst.bigstack.exe := COPTFLAG += -x01
60 #
61 GCC = $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc

```

```

62 nfs/%.o: $(SNOOPDIR)/%.c
63     $(COMPILE.c) -o $@ $< -I$(SNOOPDIR)
64     $(POST_PROCESS_O)
65 nfs/tst.call.exe: nfs/tst.call.o nfs/$(SNOOPOBJS)
66     $(LINK.c) -o $@ nfs/tst.call.o nfs/$(SNOOPOBJS) $(LDLIBS) -lnsl
67     $(POST_PROCESS) ; $(STRIP_STABS)
68 $(RPCSVCDIR)/%.c: $(RPCSVCDIR)/%.x
69     $(RPCGEN) -Cc $< > $@
70 nfs/$(RPCSVCOBJS): $(RPCSVCDIR)/$(RPCSVCSRCS)
71     $(COMPILE.c) -o $@ $(RPCSVCDIR)/$(RPCSVCSRCS)
72     $(POST_PROCESS_O)
73 nfs/tst.call3.exe: nfs/tst.call3.o nfs/$(RPCSVCOBJS)
74     $(LINK.c) -o $@ nfs/tst.call3.o nfs/$(RPCSVCOBJS) \
75     $(LDLIBS) -lnsl -lrpcsvc
76     $(POST_PROCESS) ; $(STRIP_STABS)
77 #
78 pid/tst.gcc.exe: pid/tst.gcc.c
79     $(GCC) -o pid/tst.gcc.exe pid/tst.gcc.c $(LDLFLAGS)
80     $(POST_PROCESS) ; $(STRIP_STABS)
81 #
82 json/tst.usdt.o: json/usdt.h
83 #
84 json/usdt.h: json/usdt.d
85     $(DTRACE) -h -s json/usdt.d -o json/usdt.h
86 #
87 json/usdt.o: json/usdt.d json/tst.usdt.o
88     $(COMPILE.d) -o json/usdt.o -s json/usdt.d json/tst.usdt.o
89 #
90 json/tst.usdt.exe: json/tst.usdt.o json/usdt.o
91     $(LINK.c) -o json/tst.usdt.exe json/tst.usdt.o json/usdt.o $(LDLIBS)
92     $(POST_PROCESS) ; $(STRIP_STABS)
93 #
94 #
95 # Tests that use the next three programs rely on the binaries having
96 # valid CTF data.
97 #
98 uctf/tst.aouttype.exe: uctf/tst.aouttype.c
99     $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.aouttype.o uctf/tst.aouttype.c
100    $(CTFCONVERT) -i -L VERSION uctf/tst.aouttype.o
101    $(LINK.c) -o uctf/tst.aouttype.exe uctf/tst.aouttype.o $(LDLIBS)
102    $(CTFMERGE) -L VERSION -o $@ uctf/tst.aouttype.o
103    $(POST_PROCESS) ; $(STRIP_STABS)
104 #
105 uctf/tst.chasestrings.exe: uctf/tst.chasestrings.c
106    $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.chasestrings.o uctf/tst.chasestrin
107    $(CTFCONVERT) -i -L VERSION uctf/tst.chasestrings.o
108    $(LINK.c) -o uctf/tst.chasestrings.exe uctf/tst.chasestrings.o $(LDLIBS)
109    $(CTFMERGE) -L VERSION -o $@ uctf/tst.chasestrings.o
110    $(POST_PROCESS) ; $(STRIP_STABS)
111 #
112 uctf/tst.printtype.exe: uctf/tst.printtype.c
113    $(COMPILE.c) $(CTF_FLAGS) -o uctf/tst.printtype.o uctf/tst.printtype.c
114    $(CTFCONVERT) -i -L VERSION uctf/tst.printtype.o
115    $(LINK.c) -o uctf/tst.printtype.exe uctf/tst.printtype.o $(LDLIBS)
116    $(CTFMERGE) -L VERSION -o $@ uctf/tst.printtype.o
117    $(POST_PROCESS) ; $(STRIP_STABS)
118 #
119 #
120 # This program should never have any ctf data in it.
121 #
122 uctf/tst.libtype.exe:
123    $(LINK.c) -o uctf/tst.libtype.exe uctf/tst.libtype.c $(LDLIBS)
124    $(POST_PROCESS) ; $(STRIP_STABS)
125 #
126 usdt/tst.args.exe: usdt/tst.args.o usdt/args.o

```

```
127     $(LINK.c) -o usdt/tst.args.exe usdt/tst.args.o usdt/args.o $(LDLIBS)
128     $(POST_PROCESS) ; $(STRIP_STABS)

130 usdt/args.o: usdt/args.d usdt/tst.args.o
131     $(COMPILE.d) -o usdt/args.o -s usdt/args.d usdt/tst.args.o

133 usdt/tst.argmap.exe: usdt/tst.argmap.o usdt/argmap.o
134     $(LINK.c) -o usdt/tst.argmap.exe \
135     usdt/tst.argmap.o usdt/argmap.o $(LDLIBS)
136     $(POST_PROCESS) ; $(STRIP_STABS)

138 usdt/argmap.o: usdt/argmap.d usdt/tst.argmap.o
139     $(COMPILE.d) -o usdt/argmap.o -s usdt/argmap.d usdt/tst.argmap.o

141 usdt/tst.forker.exe: usdt/tst.forker.o usdt/forker.o
142     $(LINK.c) -o usdt/tst.forker.exe \
143     usdt/tst.forker.o usdt/forker.o $(LDLIBS)
144     $(POST_PROCESS) ; $(STRIP_STABS)

146 usdt/forker.o: usdt/forker.d usdt/tst.forker.o
147     $(COMPILE.d) -o usdt/forker.o -s usdt/forker.d usdt/tst.forker.o

149 usdt/tst.forker.o: usdt/forker.h

151 usdt/forker.h: usdt/forker.d
152     $(DTRACE) -h -s usdt/forker.d -o usdt/forker.h

154 usdt/tst.lazyprobe.exe: usdt/tst.lazyprobe.o usdt/lazyprobe.o
155     $(LINK.c) -o usdt/tst.lazyprobe.exe \
156     usdt/tst.lazyprobe.o usdt/lazyprobe.o $(LDLIBS)
157     $(POST_PROCESS) ; $(STRIP_STABS)

159 usdt/lazyprobe.o: usdt/lazyprobe.d usdt/tst.lazyprobe.o
160     $(COMPILE.d) -xlazyload -o usdt/lazyprobe.o \
161     -s usdt/lazyprobe.d usdt/tst.lazyprobe.o

163 usdt/tst.lazyprobe.o: usdt/lazyprobe.h

165 usdt/lazyprobe.h: usdt/lazyprobe.d
166     $(DTRACE) -h -s usdt/lazyprobe.d -o usdt/lazyprobe.h

168 SUBDIRS = java_api
169 include ../../Makefile.subdirs
```

```

*****
25900 Tue Oct 7 15:48:34 2014
new/usr/src/lib/libc/sparc/Makefile.com
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2012, Joyent, Inc. All rights reserved.
24 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2013 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2014 Gary Mills
27 #
28 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
29 # Use is subject to license terms.
30 #

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

38 # objects are grouped by source directory

40 # Symbol capabilities objects.
41 EXTPICS= \
42 $(LIBCDIR)/capabilities/sun4u/sparc/pics/symcap.o \
43 $(LIBCDIR)/capabilities/sun4u-opl/sparc/pics/symcap.o \
44 $(LIBCDIR)/capabilities/sun4u-us3-hwcap1/sparc/pics/symcap.o \
45 $(LIBCDIR)/capabilities/sun4u-us3-hwcap2/sparc/pics/symcap.o \
46 $(LIBCDIR)/capabilities/sun4v-hwcap1/sparc/pics/symcap.o \
47 $(LIBCDIR)/capabilities/sun4v-hwcap2/sparc/pics/symcap.o

49 # local objects
50 STRETS= \
51 stret1.o \
52 stret2.o \
53 stret4.o

55 CRTOBS= \
56 _ftou.o \
57 cerror.o \
58 cerror64.o \
59 hwmuldiv.o

61 DYNOBJS= \

```

```

62 _rtbootld.o

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

104 FPASMOBJS= \
105 _Q_get_rp_rd.o \
106 fpgetmask.o \
107 fpgetrnd.o \
108 fpgetsticky.o \
109 fpsetmask.o \
110 fpsetrnd.o \
111 fpsetsticky.o

113 $(__GNUC)FPASMOBJS += \
114 __quad.o

116 ATOMICOBJS= \
117 atomic.o

119 XATTROBJS= \
120 xattr_common.o

122 COMOBJS= \
123 bcmp.o \
124 bcopy.o \
125 bzero.o \
126 bsearch.o \
127 memccpy.o

```

```

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

134 DTRACEOBS=        \
135      dtrace_data.o  \

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

172 # sysobjs that contain large-file interfaces
173 COMSYSOBS64=      \
174      fstatvfs64.o \
175      getdents64.o \
176      getrlimit64.o \
177      lseek64.o    \
178      mmap64.o     \
179      pread64.o    \
180      pwrite64.o   \
181      setrlimit64.o \
182      statvfs64.o

184 SYSOBS64=

186 COMSYSOBS=       \
187      __clock_timer.o \
188      __getloadavg.o  \
189      __rusagesys.o  \
190      __signotify.o  \
191      __sigrt.o      \
192      __time.o       \
193      __lgrp_home_fast.o \

```

```

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

```

```

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

```

```

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

```



```

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

```

```

458      gethz.o      \
459      getisax.o    \
460      getloadavg.o \
461      getlogin.o   \
462      getmntent.o  \
463      getnetgrent.o \
464      get_nprocs.o \
465      getopt.o     \
466      getopt_long.o \
467      getpagesize.o \
468      getpw.o      \
469      getpwnam.o   \
470      getpwnam_r.o \
471      getrusage.o  \
472      getspent.o   \
473      getspent_r.o \
474      getsubopt.o  \
475      gettxt.o     \
476      getusershell.o \
477      getut.o      \
478      getutx.o     \
479      getvfsent.o  \
480      getwd.o      \
481      getwidth.o   \
482      getxby_door.o \
483      gtxt.o       \
484      hsearch.o    \
485      iconv.o      \
486      imaxabs.o    \
487      imaxdiv.o    \
488      index.o      \
489      initgroups.o \
490      insque.o     \
491      isaexec.o    \
492      isastream.o  \
493      isatty.o     \
494      killpg.o     \
495      klpdlib.o    \
496      l64a.o       \
497      lckpwdf.o    \
498      lconstants.o \
499      ldivide.o    \
500      lexpl0.o     \
501      lfind.o      \
502      lfmt.o       \
503      lfmt_log.o   \
504      llabs.o      \
505      lldiv.o      \
506      llogl0.o     \
507      lltostr.o    \
508      localtime.o  \
509      lsearch.o    \
510      madvise.o    \
511      malloc.o     \
512      memalign.o   \
513      memmem.o     \
514      mkdev.o      \
515      mkdtemp.o    \
516      mkfifo.o     \
517      mkstemp.o    \
518      mktemp.o     \
519      mlock.o      \
520      mlockall.o   \
521      mon.o        \
522      msync.o      \
523      munlock.o    \

```

```

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

```

```

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

```

```

655 # c89 variants to support 32-bit size of c89 u/intmax_t (32-bit libc only)

```

```

656 PORTPRINT_C89= \
657     vfprintf_c89.o \
658     vprintf_c89.o \
659     vsnprintf_c89.o \
660     vsprintf_c89.o \
661     vwprintf_c89.o \
663 PORTSTDIO_C89= \
664     vscanf_c89.o \
665     vwscanf_c89.o \
667 # portable stdio objects that contain large file interfaces.
668 # Note: fopen64 is a special case, as we build it small.
669 PORTSTDIO64= \
670     fopen64.o \
671     fpos64.o \
673 PORTSTDIO_W= \
674     doscan_w.o \
676 PORTSTDIO= \
677     __extensions.o \
678     _endopen.o \
679     _filbuf.o \
680     _findbuf.o \
681     _flsbuf.o \
682     _wrtchk.o \
683     clearerr.o \
684     ctermid.o \
685     ctermid_r.o \
686     cuserid.o \
687     data.o \
688     doscan.o \
689     fdopen.o \
690     feof.o \
691     ferror.o \
692     fgetc.o \
693     fgets.o \
694     fileno.o \
695     flockf.o \
696     flush.o \
697     fopen.o \
698     fpos.o \
699     fputc.o \
700     fputs.o \
701     fread.o \
702     fseek.o \
703     fseek.o \
704     ftell.o \
705     ftello.o \
706     fwrite.o \
707     getc.o \
708     getchar.o \
709     getline.o \
710     getpass.o \
711     gets.o \
712     getw.o \
713     popen.o \
714     putc.o \
715     putchar.o \
716     puts.o \
717     putw.o \
718     rewind.o \
719     scanf.o \
720     setbuf.o \
721     setbuffer.o \

```

```

722     setvbuf.o \
723     system.o \
724     tempnam.o \
725     tmpfile.o \
726     tmpnam_r.o \
727     ungetc.o \
728     mse.o \
729     vscanf.o \
730     vwscanf.o \
731     wscanf.o \
733 PORTI18N= \
734     getwchar.o \
735     putwchar.o \
736     putws.o \
737     strtows.o \
738     wcsnlen.o \
739     wcstolmax.o \
740     wcstol.o \
741     wcstoul.o \
742     wscwcs.o \
743     wscat.o \
744     wschr.o \
745     wscmp.o \
746     wscpy.o \
747     wscspn.o \
748     wsdup.o \
749     wslen.o \
750     wsncat.o \
751     wsncmp.o \
752     wsncpy.o \
753     wspbkr.o \
754     wsprintf.o \
755     wsrchr.o \
756     wscanf.o \
757     wsspn.o \
758     wstod.o \
759     wstok.o \
760     wstol.o \
761     wstoll.o \
762     wsxfrm.o \
763     wmemchr.o \
764     wmemcmp.o \
765     wmemcpy.o \
766     wmemmove.o \
767     wmemset.o \
768     wcstr.o \
769     gettext.o \
770     gettext_real.o \
771     gettext_util.o \
772     gettext_gnu.o \
773     plural_parser.o \
774     wdresolve.o \
775     _ctype.o \
776     isascii.o \
777     toascii.o \
779 PORTI18N_COND= \
780     wcstol_longlong.o \
781     wcstoul_longlong.o \
783 PORTLOCALE= \
784     big5.o \
785     btowc.o \
786     collate.o \
787     collcmp.o \

```

```

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

```

```

854      wctrans.o           \
855      wctype.o            \
856      wcwidth.o           \
857      wscoll.o            \
\
859      AIOOBJS=            \
860      aio.o                \
861      aio_alloc.o         \
862      posix_aio.o         \
\
864      RTOBJS=             \
865      clock_timer.o       \
866      mqueue.o            \
867      pos4obj.o           \
868      sched.o             \
869      sem.o               \
870      shm.o               \
871      sigev_thread.o      \
\
873      TPOOLBJS=           \
874      thread_pool.o       \
\
876      THREADSOBJS=       \
877      alloc.o             \
878      assfail.o           \
879      cancel.o            \
880      door_calls.o        \
881      tmem.o              \
882      pthr_attr.o         \
883      pthr_barrier.o      \
884      pthr_cond.o         \
885      pthr_mutex.o        \
886      pthr_rwlock.o       \
887      pthread.o           \
888      rwlock.o            \
889      scalls.o            \
890      sema.o              \
891      sigaction.o         \
892      spawn.o             \
893      synch.o             \
894      tdb_agent.o         \
895      thr.o                \
896      thread_interface.o  \
897      tls.o                \
898      tsd.o                \
\
900      THREADSMACHOBJS=   \
901      machdep.o           \
\
903      THREADSASMOBJS=    \
904      asm_subr.o          \
\
906      UNICODEOBJS=       \
907      u8_textprep.o      \
908      uconv.o             \
\
910      UNWINDMACHOBJS=    \
911      unwind.o           \
\
913      UNWINDASMOBJS=     \
914      unwind_frame.o     \
\
916      # objects that implement the transitional large file API
917      PORTSYS64=          \
918      lockf64.o           \
919      stat64.o            \

```

```

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

```

```

987 MOSTOBSJ= \
988     $(STRETS) \
989     $(CRTOBSJ) \
990     $(DYNOBJS) \
991     $(FPOBJS) \
992     $(FPASMOBJS) \
993     $(ATOMICOBJS) \
994     $(XATTROBJS) \
995     $(COMOBJS) \
996     $(DTRACEOBSJ) \
997     $(GENOBJS) \
998     $(PRFOBJS) \
999     $(PORTFP) \
1000    $(PORTGEN) \
1001    $(PORTGEN64) \
1002    $(PORTI18N) \
1003    $(PORTI18N_COND) \
1004    $(PORTLOCALE) \
1005    $(PORTPRINT) \
1006    $(PORTPRINT_C89) \
1007    $(PORTPRINT_W) \
1008    $(PORTREGEX) \
1009    $(PORTSTDIO) \
1010    $(PORTSTDIO64) \
1011    $(PORTSTDIO_C89) \
1012    $(PORTSTDIO_W) \
1013    $(PORTSYS) \
1014    $(PORTSYS64) \
1015    $(AIOOBJS) \
1016    $(RTOBJS) \
1017    $(TPOOLOBSJ) \
1018    $(THREADSOBJS) \
1019    $(THREADSMACHOBJS) \
1020    $(THREADSASMOBJS) \
1021    $(UNICODEOBJS) \
1022    $(UNWINDMACHOBJS) \
1023    $(UNWINDASMOBJS) \
1024    $(COMSYSOBJS) \
1025    $(SYSOBJS) \
1026    $(COMSYSOBJS64) \
1027    $(SYSOBJS64) \
1028    $(VALUES) \
1030 TRACEOBSJ= \
1031     plockstat.o \
1033 # NOTE: libc.so.1 must be linked with the minimal crti.o and crtn.o \
1034 # modules whose source is provided in the $(SRC)/lib/common directory. \
1035 # This must be done because otherwise the Sun C compiler would insert \
1036 # its own versions of these modules and those versions contain code \
1037 # to call out to C++ initialization functions. Such C++ initialization \
1038 # functions can call back into libc before thread initialization is \
1039 # complete and this leads to segmentation violations and other problems. \
1040 # Since libc contains no C++ code, linking with the minimal crti.o and \
1041 # crtn.o modules is safe and avoids the problems described above. \
1042 OBJECTS= $(CRTI) $(MOSTOBSJ) $(CRTN) \
1043 CRTSRCS= ../../common/sparc \
1045 # include common library definitions \
1046 include $(SRC)/lib/Makefile.lib \
1048 # we need to override the default SONAME here because we might \
1049 # be building a variant object (still libc.so.1, but different filename) \
1050 SONAME = libc.so.1

```

```

1052 CFLAGS += $(CCVERBOSE)

1054 # This is necessary to avoid problems with calling _ex_unwind().
1055 # We probably don't want any inlining anyway.
1056 CFLAGS += -xinline=

1058 CERRWARN += _gcc=-Wno-parentheses
1059 CERRWARN += _gcc=-Wno-switch
1060 CERRWARN += _gcc=-Wno-uninitialized
1061 CERRWARN += _gcc=-Wno-unused-value
1062 CERRWARN += _gcc=-Wno-unused-label
1063 CERRWARN += _gcc=-Wno-unused-variable
1064 CERRWARN += _gcc=-Wno-type-limits
1065 CERRWARN += _gcc=-Wno-char-subscripts
1066 CERRWARN += _gcc=-Wno-clobbered
1067 CERRWARN += _gcc=-Wno-unused-function
1068 CERRWARN += _gcc=-Wno-address

1070 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1071 # enables ASSERT() checking in the threads portion of the library.
1072 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1073 THREAD_DEBUG =
1074 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1076 # Make string literals read-only to save memory.
1077 CFLAGS += $(XSTRCONST)

1079 ALTPICS= $(TRACEOBSJ:%=pics%)

1081 $(DYNLIB) := BUILD.SO = $(LD) -o $$@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1083 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1085 CFLAGS += $(EXTN_CFLAGS)
1086 CPPFLAGS= -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1087 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1088 ASFLAGS= $(EXTN_ASFLAGS) -K pic -P -D__STDC__ -D_ASM $(CPPFLAGS) $(sparc_

1090 # As a favor to the dtrace syscall provider, libc still calls the
1091 # old syscall traps that have been obsoleted by the *at() interfaces.
1092 # Delete this to compile libc using only the new *at() system call traps
1093 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1095 # Inform the run-time linker about libc specialized initialization
1096 RTLDINFO = -z rtldinfo=tls_rtldinfo
1097 DYNFLAGS += $(RTLDINFO)

1099 # Force libc's internal references to be resolved immediately upon loading
1100 # in order to avoid critical region problems. Since almost all libc symbols
1101 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1102 DYNFLAGS += -znw

1104 DYNFLAGS += -e __rtboot
1105 DYNFLAGS += $(EXTN_DYNFLAGS)

1107 # Inform the kernel about the initial DTrace area (in case
1108 # libc is being used as the interpreter / runtime linker).
1109 DTRACE_DATA = -zdtrace=dtrace_data
1110 DYNFLAGS += $(DTRACE_DATA)

1112 # DTrace needs an executable data segment.
1113 MAPFILE.NED=

1115 BUILD.s= $(AS) $(ASFLAGS) $< -o $$@

1117 # Override this top level flag so the compiler builds in its native

```

```

1118 # C99 mode. This has been enabled to support the complex arithmetic
1119 # added to libc.
1120 C99MODE= $(C99_ENABLE)

1122 # libc method of building an archive
1123 # The "$(GREP) -v ' L ' " part is necessary only until
1124 # lorder is fixed to ignore thread-local variables.
1125 BUILD.AR= $(RM) $$@ ; \
1126 $(AR) q $$@ '$(LORDER) $(MOSTOBSJ:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1128 # extra files for the clean target
1129 CLEANFILES= \
1130 $(LIBCDIR)/port/gen/errlst.c \
1131 $(LIBCDIR)/port/gen/new_list.c \
1132 assym.h \
1133 genassym \
1134 $(LIBCBASE)/crt/_rtld.s \
1135 $(LIBCBASE)/crt/_rtbootld.s \
1136 pics/_rtbootld.o \
1137 pics/crti.o \
1138 pics/crtn.o \
1139 $(ALTPICS)

1141 CLOBBERFILES += $(LIB_PIC)

1143 # list of C source for lint
1144 SRCS= \
1145 $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c) \
1146 $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c) \
1147 $(COMOBSJ:%.o=$(SRC)/common/util/%.c) \
1148 $(DTRACEOBSJ:%.o=$(SRC)/common/dtrace/%.c) \
1149 $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c) \
1150 $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c) \
1151 $(PORTI18N:%.o=$(LIBCDIR)/port/i18n/%.c) \
1152 $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c) \
1153 $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c) \
1154 $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c) \
1155 $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c) \
1156 $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c) \
1157 $(AIOOBSJ:%.o=$(LIBCDIR)/port/aio/%.c) \
1158 $(RTOBSJ:%.o=$(LIBCDIR)/port/rt/%.c) \
1159 $(TPOOLOBSJ:%.o=$(LIBCDIR)/port/tpool/%.c) \
1160 $(THREADSOBSJ:%.o=$(LIBCDIR)/port/threads/%.c) \
1161 $(THREADSMACHOBSJ:%.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1162 $(UNICODEOBSJ:%.o=$(SRC)/common/unicode/%.c) \
1163 $(UNWINDMACHOBSJ:%.o=$(LIBCDIR)/port/unwind/%.c) \
1164 $(FPOBSJ:%.o=$(LIBCDIR)/$(MACH)/fp/%.c) \
1165 $(LIBCBASE)/crt/_ftou.c \
1166 $(LIBCBASE)/gen/_xregs_clrptr.c \
1167 $(LIBCBASE)/gen/byteorder.c \
1168 $(LIBCBASE)/gen/ecvt.c \
1169 $(LIBCBASE)/gen/getctxt.c \
1170 $(LIBCBASE)/gen/lmul.c \
1171 $(LIBCBASE)/gen/makectxt.c \
1172 $(LIBCBASE)/gen/signfolst.c \
1173 $(LIBCBASE)/gen/siglongjmp.c \
1174 $(LIBCBASE)/gen/swapctxt.c \
1175 $(LIBCBASE)/sys/ptrace.c \
1176 $(LIBCBASE)/sys/uadmin.c

1178 # conditional assignments
1179 $(DYNLIB) := CRTI = crt.i.o
1180 $(DYNLIB) := CRTN = crtn.o

1182 # Files which need the threads .il inline template
1183 TIL= \

```

```

1184 aio.o \
1185 alloc.o \
1186 assfail.o \
1187 atexit.o \
1188 atfork.o \
1189 cancel.o \
1190 door_calls.o \
1191 err.o \
1192 errno.o \
1193 getctxt.o \
1194 lwp.o \
1195 ma.o \
1196 machdep.o \
1197 posix_aio.o \
1198 pthr_attr.o \
1199 pthr_barrier.o \
1200 pthr_cond.o \
1201 pthr_mutex.o \
1202 pthr_rwlock.o \
1203 pthread.o \
1204 rand.o \
1205 rwlock.o \
1206 scalls.o \
1207 sched.o \
1208 sema.o \
1209 sigaction.o \
1210 sigev_thread.o \
1211 spawn.o \
1212 stack.o \
1213 swapctxt.o \
1214 synch.o \
1215 tdb_agent.o \
1216 thr.o \
1217 thread_interface.o \
1218 thread_pool.o \
1219 tls.o \
1220 tsd.o \
1221 unwind.o \

1223 $(TIL:%=pics/%) := CFLAGS += $(LIBCBASE)/threads/sparc.il

1225 # special kludge for inlines with 'cas':
1226 pics/rwlock.o pics/synch.o pics/lwp.o pics/door_calls.o := \
1227     sparc_CFLAGS += -_gcc=-Wa,-xarch=v8plus

1229 # Files in port/fp subdirectory that need base.il inline template
1230 IL=
1231     __flt_decim.o \
1232     decimal_bin.o

1234 $(IL:%=pics/%) := CFLAGS += $(LIBCBASE)/fp/base.il

1236 # Files in fp subdirectory which need __quad.il inline template
1237 QIL=
1238     _Q_add.o \
1239     _Q_cmp.o \
1240     _Q_cmpe.o \
1241     _Q_div.o \
1242     _Q_dtoq.o \
1243     _Q_fcc.o \
1244     _Q_mul.o \
1245     _Q_qtod.o \
1246     _Q_qtoi.o \
1247     _Q_qtos.o \
1248     _Q_qtou.o \
1249     _Q_sqrt.o \

```

```

1250     _Q_stoq.o \
1251     _Q_sub.o

1253 $(QIL:%=pics/%) := CFLAGS += $(LIBCDIR)/$(MACH)/fp/__quad.il
1254 pics/_Q%.o := sparc_COPTFLAG = -_cc=-x04 -_gcc=-O2 -dalign
1255 pics/__quad%.o := sparc_COPTFLAG = -_cc=-x04 -_gcc=-O2 -dalign
1253 pics/_Q%.o := sparc_COPTFLAG = -x04 -dalign
1254 pics/__quad%.o := sparc_COPTFLAG = -x04 -dalign

1257 # large-file-aware components that should be built large

1259 $(COMSYSOBS64:%=pics/%) := \
1260     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1262 $(SYSOBS64:%=pics/%) := \
1263     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1265 $(PORTGEN64:%=pics/%) := \
1266     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1268 $(PORTSTDIO64:%=pics/%) := \
1269     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1271 $(PORTSYS64:%=pics/%) := \
1272     CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1274 $(PORTSTDIO_W:%=pics/%) := \
1275     CPPFLAGS += -D_WIDE

1277 $(PORTPRINT_W:%=pics/%) := \
1278     CPPFLAGS += -D_WIDE

1280 # printf/scanf functions to support c89-sized intmax_t variables
1281 $(PORTPRINT_C89:%=pics/%) := \
1282     CPPFLAGS += -D_C89_INTMAX32

1284 $(PORTSTDIO_C89:%=pics/%) := \
1285     CPPFLAGS += -D_C89_INTMAX32

1287 $(PORTI18N_COND:%=pics/%) := \
1288     CPPFLAGS += -D_WCS_LOGLONG

1290 # Files which need extra optimization
1291 pics/getenv.o := sparc_COPTFLAG = -_cc=-x04 -_gcc=-O2
1290 pics/getenv.o := sparc_COPTFLAG = -x04

1293 .KEEP_STATE:

1295 all: $(LIBS) $(LIB_PIC)

1297 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1298 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1299 lint := LINTFLAGS += -mn

1301 lint:
1302     @echo $(LINT.c) ... $(LDLIBS)
1303     @$$(LINT.c) $(SRCS) $(LDLIBS)

1305 $(LINTLIB) := SRCS=$(LIBCDIR)/port/l1ib-lc
1306 $(LINTLIB) := CPPFLAGS += -D_MSE_INT_H
1307 $(LINTLIB) := LINTFLAGS=-nvx

1309 # object files that depend on inline template
1310 $(TIL:%=pics/%) := $(LIBCBASE)/threads/sparc.il
1311 $(IL:%=pics/%) := $(LIBCBASE)/fp/base.il
1312 $(QIL:%=pics/%) := $(LIBCDIR)/$(MACH)/fp/__quad.il

```

```

1314 # include common libc targets
1315 include $(LIBCDIR)/Makefile.targ

1317 # We need to strip out all CTF and DOF data from the static library
1318 $(LIB_PIC) := DIR = pics
1319 $(LIB_PIC): pics $$ (PICS)
1320     $(BUILD.AR)
1321     $(MCS) -d -n .SUNW_ctf $$@ > /dev/null 2>&1
1322     $(MCS) -d -n .SUNW_dof $$@ > /dev/null 2>&1
1323     $(AR) -ts $$@ > /dev/null
1324     $(POST_PROCESS_A)

1326 # special cases
1327 $(STRETS:%=pics/%): $(LIBCBASE)/crt/stret.s
1328     $(AS) $(ASFLAGS) -DSTRET$@F:stret%.o=%) $(LIBCBASE)/crt/stret.s -o $$@
1329     $(POST_PROCESS_O)

1331 $(LIBCBASE)/crt/_rtbootld.s: $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.
1332     $(CC) $(CPPFLAGS) $(CTF_FLAGS) -O -S -K pic \
1333     $(LIBCBASE)/crt/_rtld.c -o $(LIBCBASE)/crt/_rtld.s
1334     $(CAT) $(LIBCBASE)/crt/_rtboot.s $(LIBCBASE)/crt/_rtld.s > $$@
1335     $(RM) $(LIBCBASE)/crt/_rtld.s

1337 # partially built from C source
1338 pics/_rtbootld.o: $(LIBCBASE)/crt/_rtbootld.s
1339     $(AS) $(ASFLAGS) $(LIBCBASE)/crt/_rtbootld.s -o $$@
1340     $(CTFCONVERT_O)

1342 ASSYMDEP_OBJS= \
1343     _lwp_mutex_unlock.o \
1344     _stack_grow.o \
1345     asm_subr.o \
1346     setjmp.o \
1347     smt_pause.o \
1348     tls_get_addr.o \
1349     unwind_frame.o \
1350     vforkx.o

1352 $(ASSYMDEP_OBJS:%=pics/%) := CPPFLAGS += -I.

1354 $(ASSYMDEP_OBJS:%=pics/%): assym.h

1356 # assym.h build rules

1358 assym.h := CFLAGS += -g

1360 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1362 genassym: $(GENASSYM_C)
1363     $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1364     $(CPPFLAGS.native) -o $$@ $(GENASSYM_C)

1366 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1368 assym.h: $(OFFSETS) genassym
1369     $(OFFSETS_CREATE) <$(OFFSETS) >$$@
1370     ./genassym >>$$@

1372 # derived C source and related explicit dependencies
1373 $(LIBCDIR)/port/gen/errlst.c + \
1374 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1375     cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

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

```

```

1379 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c

```



```

*****
24548 Tue Oct 7 15:48:34 2014
new/usr/src/lib/libc/sparcv9/Makefile.com
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2012, Joyent, Inc. All rights reserved.
24 # Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2013 Garrett D'Amore <garrett@damore.org>
26 # Copyright 2014 Gary Mills
27 #
28 # Copyright 2011 Nexenta Systems, Inc. All rights reserved.
29 # Use is subject to license terms.
30 #

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

38 # objects are grouped by source directory

40 # Symbol capabilities objects.
41 EXTPICS=     \
42             $(LIBCDIR)/capabilities/sun4u/sparcv9/pics/symcap.o \
43             $(LIBCDIR)/capabilities/sun4u-opl/sparcv9/pics/symcap.o \
44             $(LIBCDIR)/capabilities/sun4u-us3-hwcap1/sparcv9/pics/symcap.o \
45             $(LIBCDIR)/capabilities/sun4u-us3-hwcap2/sparcv9/pics/symcap.o \
46             $(LIBCDIR)/capabilities/sun4v-hwcap1/sparcv9/pics/symcap.o \
47             $(LIBCDIR)/capabilities/sun4v-hwcap2/sparcv9/pics/symcap.o

49 # local objects
50 STRETS=

52 CRTOBS=     \
53             __align_cpy_2.o \
54             __align_cpy_4.o \
55             __align_cpy_8.o \
56             _ftou.o \
57             cerror.o

59 DYNOBJS=

61 FPOBJS=     \

```

```

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

97 FPOBJS64=  \
98             _Qp_qtox.o \
99             _Qp_qtoux.o \
100            _Qp_xtou.o \
101            _Qp_uxtou.o \
102            __dtoul.o \
103            __ftoul.o

105 FPASMOBJS= \
106             _Q_get_rp_rd.o \
107             __quad_mag64.o \
108             fpgetmask.o \
109             fpgetrnd.o \
110             fpgetsticky.o \
111             fpsetmask.o \
112             fpsetrnd.o \
113             fpsetsticky.o

115 $(__GNUC)FPASMOBJS += \
116             __quad.o

118 ATOMICOBJS= \
119             atomic.o

121 XATTROBJS= \
122             xattr_common.o

124 COMOBJS=   \
125             bcmp.o \
126             bcopy.o \
127             bsearch.o \

```

```

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

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

165 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
166 # This macro should ALWAYS be empty; native APIs are already 'large file'.
167 COMSYSOBSJ64=

169 SYSOBSJ64=

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

```

```

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

```

```

260      priocntlset.o      \
261      processor_bind.o   \
262      processor_info.o   \
263      profil.o           \
264      putmsg.o           \
265      putpmsg.o          \
266      pwrite.o           \
267      read.o             \
268      readv.o            \
269      resolvepath.o     \
270      seteguid.o         \
271      setgid.o           \
272      setgroups.o        \
273      setitimer.o        \
274      setreid.o          \
275      setrlimit.o        \
276      setuid.o           \
277      sigaltstk.o        \
278      sigprocmsk.o       \
279      sigsendset.o       \
280      sigsuspend.o        \
281      statfs.o           \
282      statvfs.o          \
283      stty.o             \
284      sync.o             \
285      sysconfig.o        \
286      sysfs.o            \
287      sysinfo.o          \
288      syslwp.o           \
289      times.o            \
290      ulimit.o           \
291      umask.o            \
292      umount2.o          \
293      utssys.o           \
294      uucopy.o           \
295      vhangup.o          \
296      waitid.o           \
297      write.o            \
298      writev.o           \
299      yield.o            \
\
301 SYSOBJS=
302      __clock_gettime.o  \
303      __getcontext.o     \
304      __uadmin.o         \
305      _lwp_mutex_unlock.o \
306      _stack_grow.o      \
307      door.o             \
308      forkx.o            \
309      forkallx.o         \
310      gettimeofday.o     \
311      sparc_utrap_install.o \
312      syscall.o          \
313      tls_get_addr.o     \
314      uadmin.o           \
315      umount.o           \
316      uname.o            \
317      vforkx.o           \
\
319 # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
320 # This macro should ALWAYS be empty; native APIs are already 'large file'.
321 PORTGEN64=
\
323 # objects from source under $(LIBCDIR)/port
324 PORTFP=
325      __flt_decim.o      \

```

```

326      __flt_rounds.o     \
327      __tbl_10_b.o       \
328      __tbl_10_h.o       \
329      __tbl_10_s.o       \
330      __tbl_2_b.o        \
331      __tbl_2_h.o        \
332      __tbl_2_s.o        \
333      __tbl_fdq.o        \
334      __tbl_tens.o       \
335      __x_power.o        \
336      _base_sup.o        \
337      aconvert.o         \
338      decimal_bin.o      \
339      double_decim.o     \
340      econvert.o         \
341      fconvert.o         \
342      file_decim.o       \
343      finite.o           \
344      fp_data.o          \
345      func_decim.o       \
346      gconvert.o         \
347      hex_bin.o          \
348      ieee_globals.o     \
349      pack_float.o       \
350      sigfpe.o           \
351      string_decim.o     \
\
353 PORTGEN=
354      _env_data.o        \
355      _xftw.o            \
356      a64l.o             \
357      abort.o            \
358      addsev.o           \
359      ascii_strncasecmp.o \
360      assert.o           \
361      attrat.o           \
362      atof.o             \
363      atoi.o             \
364      atol.o             \
365      atoll.o            \
366      attropen.o         \
367      atexit.o           \
368      atfork.o           \
369      basename.o         \
370      calloc.o           \
371      catgets.o          \
372      catopen.o          \
373      cfgetispeed.o      \
374      cfgetospeed.o     \
375      cfree.o            \
376      cfsetispeed.o     \
377      cfsetospeed.o     \
378      cftime.o           \
379      clock.o            \
380      closedir.o         \
381      closefrom.o        \
382      confstr.o          \
383      crypt.o            \
384      csetlen.o          \
385      ctime.o            \
386      ctime_r.o          \
387      daemon.o           \
388      deflt.o            \
389      directio.o         \
390      dirname.o          \
391      div.o              \

```

```

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

```

```

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

```

```

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

```

```

590     vlfmt.o            \|
591     vpfmt.o            \|
592     waitpid.o          \|
593     walkstack.o        \|
594     wdata.o            \|
595     xgetwidth.o        \|
596     xpg4.o             \|
597     xpg6.o             \|
599     PORTPRINT_W=      \|
600         doprnt_w.o    \|
602     PORTPRINT=        \|
603         asprintf.o    \|
604         doprnt.o      \|
605         fprintf.o     \|
606         printf.o      \|
607         snprintf.o    \|
608         sprintf.o     \|
609         vfprintf.o    \|
610         vprintf.o     \|
611         vsnprintf.o   \|
612         vsprintf.o    \|
613         vwprintf.o    \|
614         wprintf.o     \|
616     # Preserved solely to ease maintenance of 32-bit and 64-bit library builds
617     # This macro should ALWAYS be empty; native APIs are already 'large file'.
618     PORTSTDIO64=
620     PORTSTDIO_W=      \|
621         doscan_w.o    \|
623     PORTSTDIO=        \|
624         __extensions.o \|
625         __endopen.o    \|
626         __filbuf.o     \|
627         __findbuf.o    \|
628         __flsbuf.o     \|
629         __wrtchk.o     \|
630         clearerr.o     \|
631         ctermid.o     \|
632         ctermid_r.o   \|
633         cuserid.o     \|
634         data.o         \|
635         doscan.o       \|
636         fdopen.o       \|
637         feof.o         \|
638         ferrord.o      \|
639         fgetc.o        \|
640         fgets.o        \|
641         fileno.o       \|
642         flockf.o       \|
643         flush.o        \|
644         fopen.o        \|
645         fpos.o         \|
646         fputc.o        \|
647         fputs.o        \|
648         fread.o        \|
649         fseek.o        \|
650         fseeko.o       \|
651         ftell.o        \|
652         ftello.o       \|
653         fwrite.o       \|
654         getc.o         \|
655         getchar.o     \|

```

```

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

680 PORTI18N=      \|
681      getwchar.o    \|
682      putwchar.o   \|
683      puts.o        \|
684      strtows.o     \|
685      wcsnlen.o     \|
686      wcstoimax.o   \|
687      wcstol.o      \|
688      wcstoul.o     \|
689      wcswcs.o      \|
690      wscat.o       \|
691      wscr.o        \|
692      wscmp.o       \|
693      wscpy.o       \|
694      wscspn.o      \|
695      wsdup.o       \|
696      wslen.o       \|
697      wscat.o       \|
698      wscmp.o       \|
699      wscpy.o       \|
700      wscspn.o      \|
701      wsdup.o       \|
702      wslen.o       \|
703      wscat.o       \|
704      wscmp.o       \|
705      wscpy.o       \|
706      wscspn.o      \|
707      wsdup.o       \|
708      wslen.o       \|
709      wscat.o       \|
710      wscmp.o       \|
711      wscpy.o       \|
712      wscspn.o      \|
713      wsdup.o       \|
714      wslen.o       \|
715      wscat.o       \|
716      wscmp.o       \|
717      wscpy.o       \|
718      wscspn.o      \|
719      wsdup.o       \|
720      wslen.o       \|
721      wscat.o       \|

```

```

722      _ctype.o      \|
723      isascii.o     \|
724      toascii.o     \|

726 PORTI18N_COND= \|
727      wcstol_longlong.o \|
728      wcstoul_longlong.o \|

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

```

```

788      ungetwc.o      \
789      utf8.o        \
790      wcrtoomb.o    \
791      wscasecmp.o   \
792      wscoll.o      \
793      wcsftime.o    \
794      wcsnrtombs.o  \
795      wcsrtombs.o   \
796      wcstombs.o    \
797      wcswidth.o    \
798      wcsxfrm.o     \
799      wctob.o       \
800      wctomb.o      \
801      wctrans.o     \
802      wctype.o      \
803      wcwidth.o    \
804      wscoll.o      \

806 AIOOBS=          \
807      aio.o         \
808      aio_alloc.o   \
809      posix_aio.o   \

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

820 TPOOLBS=         \
821      thread_pool.o \

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

847 THREADSMACHOBS= \
848      machdep.o    \

850 THREADSASMOBS=   \
851      asm_subr.o   \

853 UNICODEOBS=     \

```

```

854      u8_textprep.o \
855      uconv.o        \

857 UNWINDMACHOBS=   \
858      unwind.o      \

860 UNWINDASMOBS=    \
861      unwind_frame.o \

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

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

```

```

920      unlink.o      \
921      ustat.o       \
922      utimesys.o   \
923      zone.o        \

925 PORTREGEX=      \
926      glob.o       \
927      regcmp.o     \
928      regex.o      \
929      wordexp.o    \

931 VALUES= values-Xa.o

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

974 TRACEOBSJS=    \
975      plockstat.o \

977 # NOTE: libc.so.1 must be linked with the minimal crt1.o and crtn.o
978 # modules whose source is provided in the $(SRC)/lib/common directory.
979 # This must be done because otherwise the Sun C compiler would insert
980 # its own versions of these modules and those versions contain code
981 # to call out to C++ initialization functions. Such C++ initialization
982 # functions can call back into libc before thread initialization is
983 # complete and this leads to segmentation violations and other problems.
984 # Since libc contains no C++ code, linking with the minimal crt1.o and
985 # crtn.o modules is safe and avoids the problems described above.

```

```

986 OBJECTS= $(CRTI) $(MOSTOBSJS) $(CRTN)
987 CRTSRCS= ../../common/sparcv9

989 # include common library definitions
990 include $(SRC)/lib/Makefile.lib
991 include $(SRC)/lib/Makefile.lib.64

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

997 CFLAGS64 += $(CCVERBOSE)

999 # This is necessary to avoid problems with calling _ex_unwind().
1000 # We probably don't want any inlining anyway.
1001 CFLAGS64 += -xinline=

1003 CERRWARN += -_gcc=-Wno-parentheses
1004 CERRWARN += -_gcc=-Wno-switch
1005 CERRWARN += -_gcc=-Wno-uninitialized
1006 CERRWARN += -_gcc=-Wno-unused-value
1007 CERRWARN += -_gcc=-Wno-unused-label
1008 CERRWARN += -_gcc=-Wno-unused-variable
1009 CERRWARN += -_gcc=-Wno-type-limits
1010 CERRWARN += -_gcc=-Wno-char-subscripts
1011 CERRWARN += -_gcc=-Wno-clobbered
1012 CERRWARN += -_gcc=-Wno-unused-function
1013 CERRWARN += -_gcc=-Wno-address

1015 # Setting THREAD_DEBUG = -DTHREAD_DEBUG (make THREAD_DEBUG=-DTHREAD_DEBUG ...)
1016 # enables ASSERT() checking in the threads portion of the library.
1017 # This is automatically enabled for DEBUG builds, not for non-debug builds.
1018 THREAD_DEBUG =
1019 $(NOT_RELEASE_BUILD)THREAD_DEBUG = -DTHREAD_DEBUG

1021 # Make string literals read-only to save memory.
1022 CFLAGS64 += $(XSTRCONST)

1024 ALTPICS= $(TRACEOBSJS:%=pics/%)

1026 $(DYNLIB) := BUILD.SO = $(LD) -o $@ -G $(DYNFLAGS) $(PICS) $(ALTPICS) $(EXTPICS)

1028 MAPFILES = $(LIBCDIR)/port/mapfile-vers

1030 sparcv9_C_PICFLAGS= -K PIC
1031 CFLAGS64 += $(EXTN_CFLAGS)
1032 CPPFLAGS= -D_REENTRANT -Dsparc $(EXTN_CPPFLAGS) $(THREAD_DEBUG) \
1033 -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc $(CPPFLAGS.master)
1034 ASFLAGS= $(EXTN_ASFLAGS) -K PIC -P -D__STDC__ -D__ASM -D__sparcv9 $(CPPFLA
1035 $(sparcv9_AS_XARCH)

1037 # As a favor to the dtrace syscall provider, libc still calls the
1038 # old syscall traps that have been obsoleted by the *at() interfaces.
1039 # Delete this to compile libc using only the new *at() system call traps
1040 CPPFLAGS += -D_RETAIN_OLD_SYSCALLS

1042 # Inform the run-time linker about libc specialized initialization
1043 RTLDINFO = -z rtldinfo=tls_rtldinfo
1044 DYNFLAGS += $(RTLDINFO)

1046 # Force libc's internal references to be resolved immediately upon loading
1047 # in order to avoid critical region problems. Since almost all libc symbols
1048 # are marked 'protected' in the mapfiles, this is a minimal set (15 to 20).
1049 DYNFLAGS += -znw

1051 DYNFLAGS += $(EXTN_DYNFLAGS)

```



```

1053 BUILD.s=      $(AS) $(ASFLAGS) $< -o $@

1055 # Override this top level flag so the compiler builds in its native
1056 # C99 mode. This has been enabled to support the complex arithmetic
1057 # added to libc.
1058 C99MODE=       $(C99_ENABLE)

1060 # libc method of building an archive
1061 # The "$(GREP) -v ' L '" part is necessary only until
1062 # lorder is fixed to ignore thread-local variables.
1063 BUILD.AR=      $(RM) $@ ; \
1064               $(AR) q $@ `$(LORDER) $(MOSTOBSJS:%=$(DIR)/%) | $(GREP) -v ' L ' | $(TSOR

1066 # extra files for the clean target
1067 CLEANFILES=   \
1068               $(LIBCDIR)/port/gen/errlst.c  \
1069               $(LIBCDIR)/port/gen/new_list.c \
1070               assym.h                       \
1071               genassym                      \
1072               pics/crti.o                   \
1073               pics/crtn.o                   \
1074               $(ALTPICS)

1076 CLOBBERFILES += $(LIB_PIC)

1078 # list of C source for lint
1079 SRCS=
1080 $(ATOMICOBJS:%.o=$(SRC)/common/atomic/%.c) \
1081 $(XATTROBJS:%.o=$(SRC)/common/xattr/%.c) \
1082 $(COMOBJS:%.o=$(SRC)/common/util/%.c) \
1083 $(PORTFP:%.o=$(LIBCDIR)/port/fp/%.c) \
1084 $(PORTGEN:%.o=$(LIBCDIR)/port/gen/%.c) \
1085 $(PORTI18N:%.o=$(LIBCDIR)/port/il8n/%.c) \
1086 $(PORTLOCALE:%.o=$(LIBCDIR)/port/locale/%.c) \
1087 $(PORTPRINT:%.o=$(LIBCDIR)/port/print/%.c) \
1088 $(PORTREGEX:%.o=$(LIBCDIR)/port/regex/%.c) \
1089 $(PORTSTDIO:%.o=$(LIBCDIR)/port/stdio/%.c) \
1090 $(PORTSYS:%.o=$(LIBCDIR)/port/sys/%.c) \
1091 $(AIOOBJS:%.o=$(LIBCDIR)/port/aio/%.c) \
1092 $(RTOBJS:%.o=$(LIBCDIR)/port/rt/%.c) \
1093 $(TPOOLBJS:%.o=$(LIBCDIR)/port/tpool/%.c) \
1094 $(THREADSOBJS:%.o=$(LIBCDIR)/port/threads/%.c) \
1095 $(THREADSMACHOBJS:%.o=$(LIBCDIR)/$(MACH)/threads/%.c) \
1096 $(UNICODEOBJS:%.o=$(SRC)/common/unicode/%.c) \
1097 $(UNWINDMACHOBJS:%.o=$(LIBCDIR)/port/unwind/%.c) \
1098 $(FPOBJS:%.o=$(LIBCDIR)/$(MACH)/fp/%.c) \
1099 $(FPOBJS64:%.o=$(LIBCBASE)/fp/%.c) \
1100 $(LIBCBASE)/crt/_ftou.c \
1101 $(LIBCBASE)/gen/_xregs_clrptr.c \
1102 $(LIBCBASE)/gen/byteorder.c \
1103 $(LIBCBASE)/gen/ecvt.c \
1104 $(LIBCBASE)/gen/getctxt.c \
1105 $(LIBCBASE)/gen/makectxt.c \
1106 $(LIBCBASE)/gen/signfolst.c \
1107 $(LIBCBASE)/gen/siglongjmp.c \
1108 $(LIBCBASE)/gen/swapctxt.c

1110 # conditional assignments
1111 $(DYNLIB) := CRTI = crt1.o
1112 $(DYNLIB) := CRTN = crtn.o

1114 # Files which need the threads .il inline template
1115 TIL= \
1116     aio.o \
1117     alloc.o

```

```

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

1155 $(TIL:%=pics/) := CFLAGS64 += $(LIBCBASE)/threads/sparcv9.il

1157 # Files in fp, port/fp subdirectories that need base.il inline template
1158 IL= \
1159     __flt_decim.o \
1160     decimal_bin.o

1162 $(IL:%=pics/) := CFLAGS64 += $(LIBCBASE)/fp/base.il

1164 # Files in fp subdirectory which need __quad.il inline template
1165 QIL= \
1166     __Q_add.o \
1167     __Q_cmp.o \
1168     __Q_cmpe.o \
1169     __Q_div.o \
1170     __Q_dtoq.o \
1171     __Q_fcc.o \
1172     __Q_mul.o \
1173     __Q_qtod.o \
1174     __Q_qtoi.o \
1175     __Q_qtos.o \
1176     __Q_qtou.o \
1177     __Q_sqrt.o \
1178     __Q_stoq.o \
1179     __Q_sub.o \
1180     __Qp_qtox.o \
1181     __Qp_qtoux.o

1183 $(QIL:%=pics/) := CFLAGS64 += $(LIBCDIR)/$(MACH)/fp/__quad.il

```

```

1184 pics/_Op%.o := CFLAGS64 += -I$(LIBCDIR)/$(MACH)/fp
1185 pics/_O%.o := sparcv9_COPTFLAG = -_cc=-xO4 -_gcc=-O2 -xchip=ultra
1184 pics/_Q%.o := sparcv9_COPTFLAG = -xO4 -xchip=ultra

1187 # Files in crt subdirectory which need muldiv64.il inline template
1188 #CIL= mul64.o divrem64.o
1189 #$(CIL:%=pics/%) := CFLAGS += $(LIBCBASE)/crt/mul64.il

1191 # large-file-aware components that should be built large

1193 #$(COMSYSOBS64:%=pics/%) := \
1194 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1196 #$(SYSOBS64:%=pics/%) := \
1197 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1199 #$(PORTGEN64:%=pics/%) := \
1200 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1202 #$(PORTSTDIO64:%=pics/%) := \
1203 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1205 #$(PORTSYS64:%=pics/%) := \
1206 # CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64

1208 $(PORTSTDIO_W:%=pics/%) := \
1209 CPPFLAGS += -D_WIDE

1211 $(PORTPRINT_W:%=pics/%) := \
1212 CPPFLAGS += -D_WIDE

1214 $(PORTI18N_COND:%=pics/%) := \
1215 CPPFLAGS += -D_WCS_LOGLONG

1217 # Files which need extra optimization
1218 pics/getenv.o := sparcv9_COPTFLAG = -_cc=-xO4 -_gcc=-O2
1217 pics/getenv.o := sparcv9_COPTFLAG = -xO4

1220 .KEEP_STATE:

1222 all: $(LIBS) $(LIB_PIC)

1224 lint := CPPFLAGS += -I$(LIBCDIR)/$(MACH)/fp
1225 lint := CPPFLAGS += -D_MSE_INT_H -D_LCONV_C99
1226 lint := LINTFLAGS64 += -mn

1228 lint:
1229 @echo $(LINT.c) ... $(LDLIBS)
1230 @$$(LINT.c) $(SRCS) $(LDLIBS)

1232 $(LINTLIB):= SRCS=$(LIBCDIR)/port/llib-1c
1233 $(LINTLIB):= CPPFLAGS += -D_MSE_INT_H
1234 $(LINTLIB):= LINTFLAGS64=-nvx -m64

1236 # object files that depend on inline template
1237 $(TIL:%=pics/%) := $(LIBCBASE)/threads/sparcv9.il
1238 $(IL:%=pics/%) := $(LIBCBASE)/fp/base.il
1239 $(QIL:%=pics/%) := $(LIBCDIR)/$(MACH)/fp/__quad.il
1240 #$(CIL:%=pics/%) := $(LIBCBASE)/crt/muldiv64.il

1242 # include common libc targets
1243 include $(LIBCDIR)/Makefile.targ

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

```

```

1248 $(BUILD.AR)
1249 $(MCS) -d -n .SUNW_ctf $@ > /dev/null 2>&1
1250 $(MCS) -d -n .SUNW_dof $@ > /dev/null 2>&1
1251 $(AR) -ts $@ > /dev/null
1252 $(POST_PROCESS_A)

1254 # special cases
1255 #$(STRETS:%=pics/%) := crt/stret.s
1256 # $(AS) $(ASFLAGS) -DSTRET$(@F:stret%.o=) crt/stret.s -o $@
1257 # $(POST_PROCESS_O)

1259 #crt/_rtbootld.s: crt/_rtboot.s crt/_rtld.c
1260 # $(CC) $(CPPFLAGS) -O -S -K pic crt/_rtld.c -o crt/_rtld.s
1261 # $(CAT) crt/_rtboot.s crt/_rtld.s > $@
1262 # $(RM) crt/_rtld.s

1264 ASSYMDEP_OBJS= \
1265 _lwp_mutex_unlock.o \
1266 _stack_grow.o \
1267 asm_subr.o \
1268 setjmp.o \
1269 smt_pause.o \
1270 tls_get_addr.o \
1271 unwind_frame.o \
1272 vforkx.o

1274 $(ASSYMDEP_OBJS:%=pics/%) := CPPFLAGS += -I.

1276 $(ASSYMDEP_OBJS:%=pics/%) := assym.h

1278 # assym.h build rules

1280 assym.h := CFLAGS64 += -g

1282 GENASSYM_C = $(LIBCDIR)/$(MACH)/genassym.c

1284 genassym: $(GENASSYM_C)
1285 $(NATIVECC) -I$(LIBCBASE)/inc -I$(LIBCDIR)/inc \
1286 $(CPPFLAGS.native) -o $@ $(GENASSYM_C)

1288 OFFSETS = $(LIBCDIR)/$(MACH)/offsets.in

1290 assym.h: $(OFFSETS) genassym
1291 $(OFFSETS_CREATE) <$(OFFSETS) >$@
1292 ./genassym >>$@

1294 # derived C source and related explicit dependencies
1295 $(LIBCDIR)/port/gen/new_list.c: $(LIBCDIR)/port/gen/errlist $(LIBCDIR)/port/gen/
1296 cd $(LIBCDIR)/port/gen; pwd; $(AWK) -f errlist.awk < errlist

1298 pics/new_list.o: $(LIBCDIR)/port/gen/new_list.c

```

new/usr/src/lib/libsum/Makefile.com

1

```
*****
2952 Tue Oct 7 15:48:34 2014
new/usr/src/lib/libsum/Makefile.com
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 #
25 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #
28 #
29 SHELL=/usr/bin/ksh93
30 #
31 LIBRARY=      libsum.a
32 VERS=        .1
33 #
34 OBJECTS= \
35     sumlib.o
36 #
37 include ../../Makefile.astmsg
38 #
39 include ../../Makefile.lib
40 #
41 # mapfile-vers does not live with the sources in in common/ to make
42 # automated code updates easier.
43 MAPFILES=    ../mapfile-vers
44 #
45 # Set common AST build flags (e.g. C99/XPG6, needed to support the math stuff)
46 include ../../../../Makefile.ast
47 #
48 LIBS =       $(DYNLIB) $(LINTLIB)
49 #
50 LDLIBS += \
51     -last \
52     -lmd \
53     -lc
54 #
55 $(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)
56 #
57 SRCDIR =     ../common
58 #
59 # We use "=" here since using $(CPPFLAGS.master) is very tricky in our
60 # case - it MUST come as the last element but future changes in -D options
61 # may then cause silent breakage in the AST sources because the last -D
```

new/usr/src/lib/libsum/Makefile.com

2

```
62 # option specified overrides previous -D options so we prefer the current
63 # way to explicitly list each single flag.
64 # Notes:
65 # - "-D_BLD_DLL" comes from ${mam_cc_DLL} in Mamfile
66 CPPFLAGS = \
67     $(DTEXTDOM) $(DTS_ERRNO) \
68     -Isrc/lib/libsum \
69     -I$(ROOT)/usr/include/ast \
70     -I$(ROOT)/usr/include \
71     -D_PACKAGE_ast \
72     -D_BLD_DLL
73 #
74 CFLAGS += \
75     $(ASTCFLAGS)
76 CFLAGS64 += \
77     $(ASTCFLAGS64)
78 #
79 CERRWARN      += -_gcc=-Wno-parentheses
80 # This codepath is performance-critical
81 sparc_COPTFLAG=-_cc=-xO5 -_gcc=-O2 -_cc=-xprefetch=auto,explicit
82 sparcv9_COPTFLAG=-_cc=-xO5 -_gcc=-O2 -_cc=-xprefetch=auto,explicit
83 # No optimization for gcc
84 i386_COPTFLAG=-_cc=-xO5 -_gcc=-O0 -_cc=-xprefetch=auto,explicit
85 amd64_COPTFLAG=-_cc=-xO5 -_gcc=-O0 -_cc=-xprefetch=auto,explicit
86 amd64_ADJCOPT=$(ADJUST_OPT1)
87 sparc_COPTFLAG=-xO5 -_cc=-xprefetch=auto,explicit
88 sparcv9_COPTFLAG=-xO5 -_cc=-xprefetch=auto,explicit
89 i386_COPTFLAG=-_cc=-xO5 -_cc=-xprefetch=auto,explicit
90 amd64_COPTFLAG=-_cc=-xO5 -_cc=-xprefetch=auto,explicit
91 # Suppress this one warning as the fix would break old gcc, which upstream
92 # still supports. This needs a better fix (later).
93 pics/sumlib.o := CERRWARN += -erroff=E_CONST_PROMOTED_UNSIGNED_LONG
94 #
95 .KEEP_STATE:
96 #
97 all: $(LIBS)
98 #
99 # libsum is not lint-clean yet; fake up a target. (You can use
100 # "make lintcheck" to actually run lint; please send all lint fixes
101 # upstream (to AT&T) so the next update will pull them into ON.)
102 lint:
103     @ print "usr/src/lib/libsum is not lint-clean: skipping"
104 #
105 include ../../../../Makefile.targ
```

new/usr/src/lib/pkcs11/libsoftcrypto/amd64/Makefile

1

```
*****
2880 Tue Oct 7 15:48:34 2014
new/usr/src/lib/pkcs11/libsoftcrypto/amd64/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
25 #
26 #
27 LIBRARY = libsoftcrypto.a
28 VERS = .1
29 #
30 include ../Makefile.com
31 #
32 AES_PSM_OBJS = aes_amd64.o aes_intel.o aeskey.o
33 ARCFOUR_PSM_OBJS = arcfour-x86_64.o
34 BIGNUM_PSM_OBJS = bignum_amd64.o bignum_amd64_asm.o
35 #
36 include $(SRC)/lib/Makefile.lib
37 include $(SRC)/lib/Makefile.lib.64
38 #
39 CERRWARN += -_gcc=-Wno-type-limits
40 CERRWARN += -_gcc=-Wno-parentheses
41 CERRWARN += -_gcc=-Wno-uninitialized
42 #
43 AES_PSM_SRC = $(AES_DIR)/$(MACH64)/aes_amd64.s \
44 $(AES_DIR)/$(MACH64)/aes_intel.s \
45 $(AES_DIR)/$(MACH64)/aeskey.c
46 ARCFOUR_PSM_SRC = arcfour-x86_64.s
47 BIGNUM_PSM_SRC = $(BIGNUM_DIR)/$(MACH64)/bignum_amd64.c \
48 $(BIGNUM_DIR)/$(MACH64)/bignum_amd64_asm.s
49 #
50 # Sources need to be redefined after Makefile.lib inclusion.
51 SRCS = $(AES_SRC) $(ARCFOUR_SRC) $(BIGNUM_SRC) $(BLOWFISH_SRC) \
52 $(DES_SRC) $(MODES_SRC) $(DH_SRC) $(DSA_SRC) $(RSA_SRC) \
53 $(PAD_SRC)
54 #
55 SRCDIR = $(SRC)/lib/pkcs11/libsoftcrypto/common
56 #
57 LIBS = $(DYNLIB) $(LINTLIB)
58 MAPFILEDIR = ../common
59 #
60 COPTFLAG = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
61 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
```

new/usr/src/lib/pkcs11/libsoftcrypto/amd64/Makefile

2

```
62 CFLAGS += -xcrossfile
63 CFLAGS += -x04 -xcrossfile
64 CPPFLAGS += -I$(CRYPTODIR) -I$(UTSDIR) -D_POSIX_PTHREAD_SEMANTICS
65 ASFLAGS += $(AS_PICFLAGS) -P -D__STDC__ -D_ASM
66 BIGNUM_FLAGS += -DPSR_MUL
67 CLEANFILES += arcfour-x86_64.s
68 LDLIBS += -lcryptoutil -lc
69 LINTFLAGS64 += $(EXTRA_LINT_FLAGS)
70 #
71 #
72 all: $(LIBS)
73 #
74 lint: $(SRCS)
75 #
76 install: all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)
77 #
78 pics/%.o: $(AES_DIR)/$(MACH64)/%.c
79 $(COMPILE.c) $(AES_FLAGS) -o $@ $<
80 $(POST_PROCESS_O)
81 #
82 pics/%.o: $(AES_DIR)/$(MACH64)/%.s
83 $(COMPILE.s) $(AES_FLAGS) -o $@ $<
84 $(POST_PROCESS_O)
85 #
86 pics/%.o: $(BIGNUM_DIR)/$(MACH64)/%.c
87 $(COMPILE.c) $(BIGNUM_FLAGS) -o $@ $<
88 $(POST_PROCESS_O)
89 #
90 pics/%.o: $(BIGNUM_DIR)/$(MACH64)/%.s
91 $(COMPILE64.s) $(BIGNUM_FLAGS) -o $@ $<
92 $(POST_PROCESS_O)
93 #
94 include ../Makefile.targ
95 #
96 arcfour-x86_64.s: $(ARCFOUR_DIR)/amd64/arcfour-x86_64.pl
97 $(PERL) $? $@
98 #
99 pics/%.o: arcfour-x86_64.s
100 $(COMPILE64.s) $(ARCFOUR_FLAGS) -o $@ $<
101 $(POST_PROCESS_O)
```

new/usr/src/lib/pkcs11/libsoftcrypto/capabilities/sun4v/Makefile.com

1

\*\*\*\*\*

1606 Tue Oct 7 15:48:34 2014

new/usr/src/lib/pkcs11/libsoftcrypto/capabilities/sun4v/Makefile.com

5196 The cw wrapper restricts gcc to -O2

\*\*\*\*\*

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
25 #
26 #
27 PLATFORM = sun4v
28 #
29 include ../../Makefile.com
30 #
31 # Redefine the objects required for this capabilities group.
32 OBJECTS = $(ARCFOUR_OBJS) $(BIGNUM_OBJS)
33 #
34 include $(SRC)/lib/Makefile.lib
35 #
36 AS_CPPFLAGS += -D__STDC__ -D_ASM -DPIC -D_REENTRANT -D$(MACH)
37 ASFLAGS += $(AS_PICFLAGS) -P
38 COPTFLAG = $(ADJUST_OPT2) -_cc=-x05 -_gcc=-O2
39 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x05 -_gcc=-O2
40 CFLAGS += $(CCVERBOSE) -xbuiltin=%all -dalign
41 CFLAGS += $(CCVERBOSE) -x05 -xbuiltin=%all -dalign
42 CERRWARN += -_gcc=-Wno-parentheses
43 CERRWARN += -_gcc=-Wno-uninitialized
44 CERRWARN += -_gcc=-Wno-unused-variable
45 CERRWARN += -_gcc=-Wno-unused-function
46 CPPFLAGS += -D$(PLATFORM) -I$(CRYPTODIR) -I$(UTSDIR) \
47 -D_POSIX_PTHREAD_SEMANTICS
48 BIGNUM_FLAGS += -DUMUL64 -DNO_BIG_ONE -DNO_BIG_TWO
```

```

*****
45264 Tue Oct 7 15:48:35 2014
new/usr/src/tools/cw/cw.c
5196 The cw wrapper restricts gcc to -O2
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2014 Gary Mills
24  *
25  * Copyright 2010 Sun Microsystems, Inc. All rights reserved.
26  * Use is subject to license terms.
27  */

29 /*
30  * Wrapper for the GNU C compiler to make it accept the Sun C compiler
31  * arguments where possible.
32  *
33  * Since the translation is inexact, this is something of a work-in-progress.
34  *
35  */

37 /* If you modify this file, you must increment CW_VERSION */
38 #define CW_VERSION "1.30"
36 #define CW_VERSION "1.29"

40 /*
41  * -# Verbose mode
42  * -### Show compiler commands built by driver, no compilation
43  * -A<name[(tokens)]> Preprocessor predicate assertion
44  * -B[static|dynamic]> Specify dynamic or static binding
45  * -C Prevent preprocessor from removing comments
46  * -c Compile only - produce .o files, suppress linking
47  * -cg92 Alias for -xtarget=ss1000
48  * -D<name=[token]> Associate name with token as if by #define
49  * -d[y|n] dynamic [-dy] or static [-dn] option to linker
50  * -E Compile source through preprocessor only, output to stdout
51  * -errorf=<t> Suppress warnings specified by tags t(%none, %all, <tag list>)
52  * -errtags=<a> Display messages with tags a(no, yes)
53  * -errwarn=<t> Treats warnings specified by tags t(%none, %all, <tag list>)
54  * as errors
55  * -fast Optimize using a selection of options
56  * -fd Report old-style function definitions and declarations
57  * -features=zla Allow zero-length arrays
58  * -flags Show this summary of compiler options
59  * -fnonstd Initialize floating-point hardware to non-standard preferences
60  * -fns[=<yes|no>] Select non-standard floating point mode

```

```

61  * -fprecision=<p> Set FP rounding precision mode p(single, double, extended)
62  * -fround=<r> Select the IEEE rounding mode in effect at startup
63  * -fsimple[=<n>] Select floating-point optimization preferences <n>
64  * -fsingle Use single-precision arithmetic (-Xt and -Xs modes only)
65  * -ftrap=<t> Select floating-point trapping mode in effect at startup
66  * -fstore force floating pt. values to target precision on assignment
67  * -G Build a dynamic shared library
68  * -g Compile for debugging
69  * -H Print path name of each file included during compilation
70  * -h <name> Assign <name> to generated dynamic shared library
71  * -I<dir> Add <dir> to preprocessor #include file search path
72  * -i Passed to linker to ignore any LD_LIBRARY_PATH setting
73  * -keeptmp Keep temporary files created during compilation
74  * -KPIC Compile position independent code with 32-bit addresses
75  * -Kpic Compile position independent code
76  * -L<dir> Pass to linker to add <dir> to the library search path
77  * -l<name> Link with library lib<name>.a or lib<name>.so
78  * -mc Remove duplicate strings from .comment section of output files
79  * -mr Remove all strings from .comment section of output files
80  * -mr,"string" Remove all strings and append "string" to .comment section
81  * -mt Specify options needed when compiling multi-threaded code
82  * -native Find available processor, generate code accordingly
83  * -nofstore Do not force floating pt. values to target precision
84  * on assignment
85  * -nolib Same as -xnolib
86  * -noqueue Disable queuing of compiler license requests
87  * -norunpath Do not build in a runtime path for shared libraries
88  * -O Use default optimization level (-xO2 or -xO3. Check man page.)
89  * -o <outputfile> Set name of output file to <outputfile>
90  * -P Compile source through preprocessor only, output to .i file
91  * -PIC Alias for -KPIC or -xcode=pic32
92  * -p Compile for profiling with prof
93  * -pic Alias for -Kpic or -xcode=pic13
94  * -Q[y|n] Emit/don't emit identification info to output file
95  * -qp Compile for profiling with prof
96  * -R<dir[:dir]> Build runtime search path list into executable
97  * -S Compile and only generate assembly code (.s)
98  * -s Strip symbol table from the executable file
99  * -t Turn off duplicate symbol warnings when linking
100 * -U<name> Delete initial definition of preprocessor symbol <name>
101 * -V Report version number of each compilation phase
102 * -v Do stricter semantic checking
103 * -W<c>,<arg> Pass <arg> to specified component <c> (a,l,m,p,0,2,h,i,u)
104 * -w Suppress compiler warning messages
105 * -Xa Compile assuming ANSI C conformance, allow K & R extensions
106 * (default mode)
107 * -Xc Compile assuming strict ANSI C conformance
108 * -Xs Compile assuming (pre-ANSI) K & R C style code
109 * -Xt Compile assuming K & R conformance, allow ANSI C
110 * -x386 Generate code for the 80386 processor
111 * -x486 Generate code for the 80486 processor
112 * -xarch=<a> Specify target architecture instruction set
113 * -xbuiltin[=<b>] When profitable inline, or substitute intrinsic functions
114 * for system functions, b={%all,%none}
115 * -xCC Accept C++ style comments
116 * -xchar_byte_order=<o> Specify multi-char byte order <o> (default, high, low)
117 * -xchip=<c> Specify the target processor for use by the optimizer
118 * -xcode=<c> Generate different code for forming addresses
119 * -xcrossfile[=<n>] Enable optimization and inlining across source files,
120 * n={0|1}
121 * -xe Perform only syntax/semantic checking, no code generation
122 * -xF Compile for later mapfile reordering or unused section
123 * elimination
124 * -xhelp=<f> Display on-line help information f(flags, readme, errors)
125 * -xildoff Cancel -xildon
126 * -xildon Enable use of the incremental linker, ild

```

```

127 * -xinline=[<a>,...,<a>] Attempt inlining of specified user routines,
128 * <a>={%auto,func,no%func}
129 * -xlibmieee Force IEEE 754 return values for math routines in
130 * exceptional cases
131 * -xlibmil Inline selected libm math routines for optimization
132 * -xlic_lib=sunperf Link in the Sun supplied performance libraries
133 * -xlicinfo Show license server information
134 * -xM Generate makefile dependencies
135 * -xMl Generate makefile dependencies, but exclude /usr/include
136 * -xmaxopt=[off,1,2,3,4,5] maximum optimization level allowed on #pragma opt
137 * -xnolib Do not link with default system libraries
138 * -xnolibmil Cancel -xlibmil on command line
139 * -xO<n> Generate optimized code (n={1|2|3|4|5})
140 * -xP Print prototypes for function definitions
141 * -xpentium Generate code for the pentium processor
142 * -xpg Compile for profiling with gprof
143 * -xprofile=<p> Collect data for a profile or use a profile to optimize
144 * <p>={{collect,use}[:<path>],tcov}
145 * -xregs=<r> Control register allocation
146 * -xs Allow debugging without object (.o) files
147 * -xsb Compile for use with the WorkShop source browser
148 * -xsbfast Generate only WorkShop source browser info, no compilation
149 * -xsfpconst Represent unsuffixed floating point constants as single
150 * precision
151 * -xspace Do not do optimizations that increase code size
152 * -xstrconst Place string literals into read-only data segment
153 * -xtarget=<t> Specify target system for optimization
154 * -xtemp=<dir> Set directory for temporary files to <dir>
155 * -xtime Report the execution time for each compilation phase
156 * -xtransition Emit warnings for differences between K&R C and ANSI C
157 * -xtrigraphs[=<yes|no>] Enable|disable trigraph translation
158 * -xunroll=n Enable unrolling loops n times where possible
159 * -Y<c>,<dir> Specify <dir> for location of component <c> (a,l,m,p,0,h,i,u)
160 * -YA,<dir> Change default directory searched for components
161 * -YI,<dir> Change default directory searched for include files
162 * -YP,<dir> Change default directory for finding libraries files
163 * -YS,<dir> Change default directory for startup object files
164 */

166 /*
167 * Translation table:
168 */
169 /*
170 * -# -v
171 * -### error
172 * -A<name[(tokens)]> pass-thru
173 * -B<[static|dynamic]> pass-thru (syntax error for anything else)
174 * -C pass-thru
175 * -c pass-thru
176 * -cg92 -m32 -mcpu=v8 -mtune=supersparc (SPARC only)
177 * -D<name[=token]> pass-thru
178 * -dy or -dn -Wl,-dy or -Wl,-dn
179 * -E pass-thru
180 * -erroff=E_EMPTY_TRANSLATION_UNIT ignore
181 * -errtags=%all -Wall
182 * -errwarn=%all -Werror else -Wno-error
183 * -fast error
184 * -fd error
185 * -features=zla ignore
186 * -flags --help
187 * -fnonstd error
188 * -fns[=<yes|no>] error
189 * -fprecision=<p> error
190 * -fround=<r> error
191 * -fsimple[=<n>] error
192 * -fsingle[=<n>] error

```

```

193 * -ftrap=<t> error
194 * -fstore error
195 * -G pass-thru
196 * -g pass-thru
197 * -H pass-thru
198 * -h <name> pass-thru
199 * -I<dir> pass-thru
200 * -i pass-thru
201 * -keeptmp -save-temps
202 * -KPIC -fPIC
203 * -Kpic -fpic
204 * -L<dir> pass-thru
205 * -l<name> pass-thru
206 * -mc error
207 * -mr error
208 * -mr,"string" error
209 * -mt -D_REENTRANT
210 * -native error
211 * -nofstore error
212 * -nolib -nodefaultlibs
213 * -noqueue ignore
214 * -norunpath ignore
215 * -O -O1 (Check the man page to be certain)
216 * -o <outputfile> pass-thru
217 * -p -E -o filename.i (or error)
218 * -PIC -fPIC (C++ only)
219 * -p pass-thru
220 * -pic -fpic (C++ only)
221 * -Q[y|n] error
222 * -qp -p
223 * -R<dir[:dir]> pass-thru
224 * -S pass-thru
225 * -s -Wl,-s
226 * -t -Wl,-t
227 * -U<name> pass-thru
228 * -V --version
229 * -v -Wall
230 * -Wa,<arg> pass-thru
231 * -Wp,<arg> pass-thru except -xc99=<a>
232 * -Wl,<arg> pass-thru
233 * -W{m,0,2,h,i,u} error/ignore
234 * -Wu,-xmodel=kernel -ffreestanding -mcmmodel=kernel -mno-red-zone
235 * -xmodel=kernel -ffreestanding -mcmmodel=kernel -mno-red-zone
236 * -Wu,-save_args msave-args
237 * -w pass-thru
238 * -Xa -std=iso9899:199409 or -ansi
239 * -Xc -ansi -pedantic
240 * -Xt error
241 * -Xs -traditional -std=c89
242 * -x386 -march=i386 (x86 only)
243 * -x486 -march=i486 (x86 only)
244 * -xarch=<a> table
245 * -xbuiltin[=<b>] -fbuiltin (-fno-builtin otherwise)
246 * -xCC ignore
247 * -xchar_byte_order=<o> error
248 * -xchip=<c> table
249 * -xcode=<c> table
250 * -xdebugformat=<format> ignore (always use dwarf-2 for gcc)
251 * -xcrossfile[=<n>] ignore
252 * -xe error
253 * -xF error
254 * -xhelp=<f> error
255 * -xildoff ignore
256 * -xildon ignore
257 * -xinline ignore
258 * -xlibmieee error

```

```

259 * -xlibmil          error
260 * -xlic_lib=sunperf error
261 * -xM               -M
262 * -xM1             -MM
263 * -xmaxopt=[...]   error
264 * -xnolib          -ndefaultlibs
265 * -xnolibmil       error
266 * -xO<n>           -O<n>
267 * -xP             error
268 * -xpentium        -march=pentium (x86 only)
269 * -xpg            error
270 * -xprofile=<p>    error
271 * -xregs=<r>      table
272 * -xs             error
273 * -xsb           error
274 * -xsbfast      error
275 * -xsfpconst    error
276 * -xspace       ignore (-not -Os)
277 * -xstrconst    ignore
278 * -xtarget=<t>   table
279 * -xtemp=<dir>  error
280 * -xtime        error
281 * -xtransition -Wtransition
282 * -xtrigraphs=<yes|no> -trigraphs -notrigraphs
283 * -xunroll=n    error
284 * -W0,-xdbggen=no%usedonly -fno-eliminate-unused-debug-symbols
285 *               -fno-eliminate-unused-debug-types
286 * -Y<c>,<dir>   error
287 * -YA,<dir>    error
288 * -YI,<dir>    -nostdinc -I<dir>
289 * -YP,<dir>    error
290 * -YS,<dir>    error
291 */

293 #include <stdio.h>
294 #include <sys/types.h>
295 #include <unistd.h>
296 #include <string.h>
297 #include <stdlib.h>
298 #include <ctype.h>
299 #include <fcntl.h>
300 #include <errno.h>
301 #include <stdarg.h>
302 #include <sys/utsname.h>
303 #include <sys/param.h>
304 #include <sys/isa_defs.h>
305 #include <sys/wait.h>
306 #include <sys/stat.h>

308 #define CW_F_CXX      0x01
309 #define CW_F_SHADOW  0x02
310 #define CW_F_EXEC    0x04
311 #define CW_F_ECHO    0x08
312 #define CW_F_XLATE   0x10
313 #define CW_F_PROG    0x20

315 typedef enum cw_compiler {
316     CW_C_CC = 0,
317     CW_C_GCC
318 } cw_compiler_t;

```

unchanged portion omitted

```

537 static void
538 optim_disable(struct aelist *h, int level)
539 {
540     if (level >= 2) {

```

```

541         newae(h, "-fno-strict-aliasing");
542         newae(h, "-fno-unit-at-a-time");
543         newae(h, "-fno-optimize-sibling-calls");
544     }
545 }

539 /* ARGSUSED */
540 static void
541 Xamode(struct aelist *h)
542 {
543 }

```

unchanged portion omitted

```

625 static void
626 do_gcc(cw_ictx_t *ctx)
627 {
628     int c;
629     int pic = 0, nolibs = 0;
630     int in_output = 0, seen_o = 0, c_files = 0;
631     cw_op_t op = CW_O_LINK;
632     char *model = NULL;
633     int mflag = 0;

635     if (ctx->i_flags & CW_F_PROG) {
636         newae(ctx->i_ae, "--version");
637         return;
638     }

640     newae(ctx->i_ae, "-fident");
641     newae(ctx->i_ae, "-finline");
642     newae(ctx->i_ae, "-fno-inline-functions");
643     newae(ctx->i_ae, "-fno-builtin");
644     newae(ctx->i_ae, "-fno-asm");
645     newae(ctx->i_ae, "-fdiagnostics-show-option");
646     newae(ctx->i_ae, "-ndefaultlibs");

648 #if defined(__sparc)
649     /*
650     * The SPARC ldd and std instructions require 8-byte alignment of
651     * their address operand. gcc correctly uses them only when the
652     * ABI requires 8-byte alignment; unfortunately we have a number of
653     * pieces of buggy code that doesn't conform to the ABI. This
654     * flag makes gcc work more like Studio with -xmemalign=4.
655     */
656     newae(ctx->i_ae, "-mno-integer-ldd-std");
657 #endif

659     /*
660     * This is needed because 'u' is defined
661     * under a conditional on 'sun'. Should
662     * probably just remove the conditional,
663     * or make it be dependent on '__sun'.
664     *
665     * -Dunix is also missing in enhanced ANSI mode
666     */
667     newae(ctx->i_ae, "-D__sun");

669     /*
670     * Walk the argument list, translating as we go ..
671     */

673     while (--ctx->i_oldargc > 0) {
674         char *arg = **ctx->i_oldargv;
675         size_t arglen = strlen(arg);

677         if (*arg == '-') {

```



```

678     arglen--;
679 } else {
680     /*
681     * Discard inline files that gcc doesn't grok
682     */
683     if (!in_output && arglen > 3 &&
684         strcmp(arg + arglen - 3, ".il") == 0)
685         continue;
687     if (!in_output && arglen > 2 &&
688         arg[arglen - 2] == '.' &&
689         (arg[arglen - 1] == 'S' || arg[arglen - 1] == 's' ||
690          arg[arglen - 1] == 'c' || arg[arglen - 1] == 'i'))
691         c_files++;
693     /*
694     * Otherwise, filenames and partial arguments
695     * are passed through for gcc to chew on. However,
696     * output is always discarded for the secondary
697     * compiler.
698     */
699     if ((ctx->i_flags & CW_F_SHADOW) && in_output)
700         newae(ctx->i_ae, ctx->i_discard);
701     else
702         newae(ctx->i_ae, arg);
703     in_output = 0;
704     continue;
705 }
707 if (ctx->i_flags & CW_F_CXX) {
708     if (strncmp(arg, "-compat=", 8) == 0) {
709         /* discard -compat=4 and -compat=5 */
710         continue;
711     }
712     if (strcmp(arg, "-Qoption") == 0) {
713         /* discard -Qoption and its two arguments */
714         if (ctx->i_oldargc < 3)
715             error(arg);
716         ctx->i_oldargc -= 2;
717         ctx->i_oldargv += 2;
718         continue;
719     }
720     if (strcmp(arg, "-xwe") == 0) {
721         /* turn warnings into errors */
722         newae(ctx->i_ae, "-Werror");
723         continue;
724     }
725     if (strcmp(arg, "-noex") == 0) {
726         /* no exceptions */
727         newae(ctx->i_ae, "-fno-exceptions");
728         /* no run time type descriptor information */
729         newae(ctx->i_ae, "-fno-rtti");
730         continue;
731     }
732     if (strcmp(arg, "-pic") == 0) {
733         newae(ctx->i_ae, "-fpic");
734         pic = 1;
735         continue;
736     }
737     if (strcmp(arg, "-PIC") == 0) {
738         newae(ctx->i_ae, "-fPIC");
739         pic = 1;
740         continue;
741     }
742     if (strcmp(arg, "-norunpath") == 0) {
743         /* gcc has no corresponding option */

```

```

744         continue;
745     }
746     if (strcmp(arg, "-nolib") == 0) {
747         /* -nodefaultlibs is on by default */
748         nolibc = 1;
749         continue;
750     }
751     #if defined(__sparc)
752     if (strcmp(arg, "-cg92") == 0) {
753         mflag |= xlate_xtb(ctx->i_ae, "v8");
754         xlate(ctx->i_ae, "super", xchip_tbl);
755         continue;
756     }
757     #endif /* __sparc */
758 }
760 switch ((c = arg[1])) {
761 case '-':
762     if (strcmp(arg, "-noecho") == 0)
763         ctx->i_flags &= ~CW_F_ECHO;
764     else if (strncmp(arg, "-cc=", 5) == 0 ||
765             strncmp(arg, "-_CC=", 5) == 0)
766         /* EMPTY */;
767     else if (strncmp(arg, "-gcc=", 6) == 0 ||
768             strncmp(arg, "-g++=", 6) == 0)
769         newae(ctx->i_ae, arg + 6);
770     else
771         error(arg);
772     break;
773 case '#':
774     if (arglen == 1) {
775         newae(ctx->i_ae, "-v");
776         break;
777     }
778     error(arg);
779     break;
780 case 'g':
781     newae(ctx->i_ae, "-gdwarf-2");
782     break;
783 case 'E':
784     if (arglen == 1) {
785         newae(ctx->i_ae, "-xc");
786         newae(ctx->i_ae, arg);
787         op = CW_O_PREPROCESS;
788         nolibc = 1;
789         break;
790     }
791     error(arg);
792     break;
793 case 'c':
794 case 'S':
795     if (arglen == 1) {
796         op = CW_O_COMPILE;
797         nolibc = 1;
798     }
799     /* FALLTHROUGH */
800 case 'C':
801 case 'H':
802 case 'p':
803     if (arglen == 1) {
804         newae(ctx->i_ae, arg);
805         break;
806     }
807     error(arg);
808     break;
809 case 'A':

```

```

810     case 'h':
811     case 'I':
812     case 'i':
813     case 'L':
814     case 'l':
815     case 'R':
816     case 'U':
817     case 'u':
818     case 'w':
819         newae(ctx->i_ae, arg);
820         break;
821     case 'o':
822         seen_o = 1;
823         if (arglen == 1) {
824             in_output = 1;
825             newae(ctx->i_ae, arg);
826         } else if (ctx->i_flags & CW_F_SHADOW) {
827             newae(ctx->i_ae, "-o");
828             newae(ctx->i_ae, ctx->i_discard);
829         } else {
830             newae(ctx->i_ae, arg);
831         }
832         break;
833     case 'D':
834         newae(ctx->i_ae, arg);
835         /*
836          * XXX Clearly a hack ... do we need _KADB too?
837          */
838         if (strcmp(arg, "-D_KERNEL") == 0 ||
839             strcmp(arg, "-D_BOOT") == 0)
840             newae(ctx->i_ae, "-ffreestanding");
841         break;
842     case 'd':
843         if (arglen == 2) {
844             if (strcmp(arg, "-dy") == 0) {
845                 newae(ctx->i_ae, "-Wl,-dy");
846                 break;
847             }
848             if (strcmp(arg, "-dn") == 0) {
849                 newae(ctx->i_ae, "-Wl,-dn");
850                 break;
851             }
852         }
853         if (strcmp(arg, "-dalign") == 0) {
854             /*
855              * -dalign forces alignment in some cases;
856              * gcc does not need any flag to do this.
857              */
858             break;
859         }
860         error(arg);
861         break;
862     case 'e':
863         if (strcmp(arg,
864             "--errorf=E_EMPTY_TRANSLATION_UNIT") == 0) {
865             /*
866              * Accept but ignore this -- gcc doesn't
867              * seem to complain about empty translation
868              * units
869              */
870             break;
871         }
872         /* XX64 -- ignore all -errorf= options, for now */
873         if (strncmp(arg, "--errorf=", 8) == 0)
874             break;
875         if (strcmp(arg, "-errtags=yes") == 0) {

```

```

876             warnings(ctx->i_ae);
877             break;
878         }
879         if (strcmp(arg, "-errwarn=all") == 0) {
880             newae(ctx->i_ae, "-Werror");
881             break;
882         }
883         error(arg);
884         break;
885     case 'f':
886         if (strcmp(arg, "-flags") == 0) {
887             newae(ctx->i_ae, "--help");
888             break;
889         }
890         if (strncmp(arg, "--features=zla", 13) == 0) {
891             /*
892              * Accept but ignore this -- gcc allows
893              * zero length arrays.
894              */
895             break;
896         }
897         error(arg);
898         break;
899     case 'G':
900         newae(ctx->i_ae, "-shared");
901         nolibc = 1;
902         break;
903     case 'k':
904         if (strcmp(arg, "-keeptmp") == 0) {
905             newae(ctx->i_ae, "-save-temps");
906             break;
907         }
908         error(arg);
909         break;
910     case 'K':
911         if (arglen == 1) {
912             if ((arg = **++ctx->i_oldargv) == NULL ||
913                 *arg == '\0')
914                 error("-K");
915             ctx->i_oldargc--;
916         } else {
917             arg += 2;
918         }
919         if (strcmp(arg, "pic") == 0) {
920             newae(ctx->i_ae, "-fpic");
921             pic = 1;
922             break;
923         }
924         if (strcmp(arg, "PIC") == 0) {
925             newae(ctx->i_ae, "-fPIC");
926             pic = 1;
927             break;
928         }
929         error("-K");
930         break;
931     case 'm':
932         if (strcmp(arg, "-mt") == 0) {
933             newae(ctx->i_ae, "-D_REENTRANT");
934             break;
935         }
936         if (strcmp(arg, "-m64") == 0) {
937             newae(ctx->i_ae, "-m64");
938             #if defined(__x86)
939             newae(ctx->i_ae, "-mtune=opteron");
940             #endif
941             mflag |= M64;

```

```

942         break;
943     }
944     if (strcmp(arg, "-m32") == 0) {
945         newae(ctx->i_ae, "-m32");
946         mflag |= M32;
947         break;
948     }
949     error(arg);
950     break;
951 case 'B': /* linker options */
952 case 'M':
953 case 'z':
954     {
955         char *opt;
956         size_t len;
957         char *s;
958
959         if (arglen == 1) {
960             opt = *++ctx->i_oldargv;
961             if (opt == NULL || *opt == '\0')
962                 error(arg);
963             ctx->i_oldargc--;
964         } else {
965             opt = arg + 2;
966         }
967         len = strlen(opt) + 7;
968         if ((s = malloc(len)) == NULL)
969             nomem();
970         (void) snprintf(s, len, "-Wl,-%c%s", c, opt);
971         newae(ctx->i_ae, s);
972         free(s);
973     }
974     break;
975 case 'n':
976     if (strcmp(arg, "-noqueue") == 0) {
977         /*
978          * Horrid license server stuff - n/a
979          */
980         break;
981     }
982     error(arg);
983     break;
984 case 'O':
985     if (arglen == 1) {
986         newae(ctx->i_ae, "-O");
987         break;
988     }
989     error(arg);
990     break;
991 case 'P':
992     /*
993     * We could do '-E -o filename.i', but that's hard,
994     * and we don't need it for the case that's triggering
995     * this addition. We'll require the user to specify
996     * -o in the Makefile. If they don't they'll find out
997     * in a hurry.
998     */
999     newae(ctx->i_ae, "-E");
1000     op = CW_O_PREPROCESS;
1001     nolibc = 1;
1002     break;
1003 case 'q':
1004     if (strcmp(arg, "-qp") == 0) {
1005         newae(ctx->i_ae, "-p");
1006         break;
1007     }

```

```

1008         error(arg);
1009         break;
1010 case 's':
1011     if (arglen == 1) {
1012         newae(ctx->i_ae, "-Wl,-s");
1013         break;
1014     }
1015     error(arg);
1016     break;
1017 case 't':
1018     if (arglen == 1) {
1019         newae(ctx->i_ae, "-Wl,-t");
1020         break;
1021     }
1022     error(arg);
1023     break;
1024 case 'V':
1025     if (arglen == 1) {
1026         ctx->i_flags &= ~CW_F_ECHO;
1027         newae(ctx->i_ae, "--version");
1028         break;
1029     }
1030     error(arg);
1031     break;
1032 case 'v':
1033     if (arglen == 1) {
1034         warnings(ctx->i_ae);
1035         break;
1036     }
1037     error(arg);
1038     break;
1039 case 'W':
1040     if (strcmp(arg, "-Wp,-xc99", 9) == 0) {
1041         /*
1042          * gcc's preprocessor will accept c99
1043          * regardless, so accept and ignore.
1044          */
1045         break;
1046     }
1047     if (strcmp(arg, "-Wa,", 4) == 0 ||
1048         strcmp(arg, "-Wp,", 4) == 0 ||
1049         strcmp(arg, "-Wl,", 4) == 0) {
1050         newae(ctx->i_ae, arg);
1051         break;
1052     }
1053     if (strcmp(arg, "-W0,-xc99=pragma") == 0) {
1054         /* (undocumented) enables _Pragma */
1055         break;
1056     }
1057     if (strcmp(arg, "-W0,-xc99=%none") == 0) {
1058         /*
1059          * This is a polite way of saying
1060          * "no c99 constructs allowed!"
1061          * For now, just accept and ignore this.
1062          */
1063         break;
1064     }
1065     if (strcmp(arg, "-W0,-noglobal") == 0 ||
1066         strcmp(arg, "-W0,-xglobalstatic") == 0) {
1067         /*
1068          * gcc doesn't prefix local symbols
1069          * in debug mode, so this is not needed.
1070          */
1071         break;
1072     }
1073     if (strcmp(arg, "-W0,-Lt") == 0) {

```

```

1074     /*
1075     * Generate tests at the top of loops.
1076     * There is no direct gcc equivalent, ignore.
1077     */
1078     break;
1079 }
1080 if (strcmp(arg, "-W0,-xdbggen=no%usedonly") == 0) {
1081     newae(ctx->i_ae,
1082         "-fno-eliminate-unused-debug-symbols");
1083     newae(ctx->i_ae,
1084         "-fno-eliminate-unused-debug-types");
1085     break;
1086 }
1087 if (strcmp(arg, "-W2,-xwrap_int") == 0) {
1088     /*
1089     * Use the legacy behaviour (pre-SS11)
1090     * for integer wrapping.
1091     * gcc does not need this.
1092     */
1093     break;
1094 }
1095 if (strcmp(arg, "-W2,-Rcond_elim") == 0) {
1096     /*
1097     * Elimination and expansion of conditionals;
1098     * gcc has no direct equivalent.
1099     */
1100     break;
1101 }
1102 if (strcmp(arg, "-Wd,-xsafe=unboundsym") == 0) {
1103     /*
1104     * Prevents optimizing away checks for
1105     * unbound weak symbol addresses. gcc does
1106     * not do this, so it's not needed.
1107     */
1108     break;
1109 }
1110 if (strncmp(arg, "-Wc,-xcode=", 11) == 0) {
1111     xlate(ctx->i_ae, arg + 11, xcode_ttbl);
1112     if (strncmp(arg + 11, "pic", 3) == 0)
1113         pic = 1;
1114     break;
1115 }
1116 if (strncmp(arg, "-Wc,-Qiselect", 13) == 0) {
1117     /*
1118     * Prevents insertion of register symbols.
1119     * gcc doesn't do this, so ignore it.
1120     */
1121     break;
1122 }
1123 if (strcmp(arg, "-Wc,-Qassembler-ounrefsym=0") == 0) {
1124     /*
1125     * Prevents optimizing away of static variables.
1126     * gcc does not do this, so it's not needed.
1127     */
1128     break;
1129 }
1130 #if defined(__x86)
1131 if (strcmp(arg, "-Wu,-xmodel=kernel") == 0) {
1132     newae(ctx->i_ae, "-ffreestanding");
1133     newae(ctx->i_ae, "-mno-red-zone");
1134     model = "-mcmmodel=kernel";
1135     nolibc = 1;
1136     break;
1137 }
1138 if (strcmp(arg, "-Wu,-save_args") == 0) {
1139     newae(ctx->i_ae, "-msave-args");

```

```

1140         break;
1141     }
1142 #endif /* __x86 */
1143     error(arg);
1144     break;
1145 case 'X':
1146     if (strcmp(arg, "-Xa") == 0 ||
1147         strcmp(arg, "-Xt") == 0) {
1148         Xamode(ctx->i_ae);
1149         break;
1150     }
1151     if (strcmp(arg, "-Xc") == 0) {
1152         Xcmode(ctx->i_ae);
1153         break;
1154     }
1155     if (strcmp(arg, "-Xs") == 0) {
1156         Xsmode(ctx->i_ae);
1157         break;
1158     }
1159     error(arg);
1160     break;
1161 case 'x':
1162     if (arglen == 1)
1163         error(arg);
1164     switch (arg[2]) {
1165 #if defined(__x86)
1166     case '3':
1167         if (strcmp(arg, "-x386") == 0) {
1168             newae(ctx->i_ae, "-march=i386");
1169             break;
1170         }
1171         error(arg);
1172         break;
1173     case '4':
1174         if (strcmp(arg, "-x486") == 0) {
1175             newae(ctx->i_ae, "-march=i486");
1176             break;
1177         }
1178         error(arg);
1179         break;
1180 #endif /* __x86 */
1181     case 'a':
1182         if (strncmp(arg, "-xarch=", 7) == 0) {
1183             mflag |= xlate_xtb(ctx->i_ae, arg + 7);
1184             break;
1185         }
1186         error(arg);
1187         break;
1188     case 'b':
1189         if (strncmp(arg, "-xbuiltin=", 10) == 0) {
1190             if (strcmp(arg + 10, "%all")
1191                 newae(ctx->i_ae, "-fbuiltin");
1192             break;
1193         }
1194         error(arg);
1195         break;
1196     case 'C':
1197         /* Accept C++ style comments -- ignore */
1198         if (strcmp(arg, "-xCC") == 0)
1199             break;
1200         error(arg);
1201         break;
1202     case 'c':
1203         if (strncmp(arg, "-xc99=%all", 10) == 0) {
1204             newae(ctx->i_ae, "-std=gnu99");
1205             break;

```

```

1206     }
1207     if (strncmp(arg, "-xc99=%none", 11) == 0) {
1208         newae(ctx->i_ae, "-std=gnu89");
1209         break;
1210     }
1211     if (strncmp(arg, "-xchip=", 7) == 0) {
1212         xlate(ctx->i_ae, arg + 7, xchip_tbl);
1213         break;
1214     }
1215     if (strncmp(arg, "-xcode=", 7) == 0) {
1216         xlate(ctx->i_ae, arg + 7, xcode_tbl);
1217         if (strncmp(arg + 7, "pic", 3) == 0)
1218             pic = 1;
1219         break;
1220     }
1221     if (strncmp(arg, "-xcache=", 8) == 0)
1222         break;
1223     if (strncmp(arg, "-xcrossfile", 11) == 0)
1224         break;
1225     error(arg);
1226     break;
1227 case 'd':
1228     if (strcmp(arg, "-xdepend") == 0)
1229         break;
1230     if (strncmp(arg, "-xdebugformat=", 14) == 0)
1231         break;
1232     error(arg);
1233     break;
1234 case 'F':
1235     /*
1236     * Compile for mapfile reordering, or unused
1237     * section elimination, syntax can be -xF or
1238     * more complex, like -xF=%all -- ignore.
1239     */
1240     if (strncmp(arg, "-xF", 3) == 0)
1241         break;
1242     error(arg);
1243     break;
1244 case 'i':
1245     if (strncmp(arg, "-xinline", 8) == 0)
1246         /* No inlining; ignore */
1247         break;
1248     if (strcmp(arg, "-xildon") == 0 ||
1249         strcmp(arg, "-xildoff") == 0)
1250         /* No incremental linking; ignore */
1251         break;
1252     error(arg);
1253     break;
1254 #if defined(__x86)
1255 case 'm':
1256     if (strcmp(arg, "-xmodel=kernel") == 0) {
1257         newae(ctx->i_ae, "-ffreestanding");
1258         newae(ctx->i_ae, "-mno-red-zone");
1259         model = "-mcmmodel=kernel";
1260         nolibc = 1;
1261         break;
1262     }
1263     error(arg);
1264     break;
1265 #endif /* __x86 */
1266 case 'M':
1267     if (strcmp(arg, "-xM") == 0) {
1268         newae(ctx->i_ae, "-M");
1269         break;
1270     }
1271     if (strcmp(arg, "-xM1") == 0) {

```

```

1272         newae(ctx->i_ae, "-MM");
1273         break;
1274     }
1275     error(arg);
1276     break;
1277 case 'n':
1278     if (strcmp(arg, "-xnolib") == 0) {
1279         nolibc = 1;
1280         break;
1281     }
1282     error(arg);
1283     break;
1284 case 'O':
1285     if (strncmp(arg, "-xO", 3) == 0) {
1286         size_t len = strlen(arg);
1287         char *s;
1288         int c = *(arg + 3);
1289         int level;
1290
1291         if (len != 4 || !isdigit(c))
1292             error(arg);
1293
1294         if ((s = malloc(len)) == NULL)
1295             nomem();
1296
1297         level = atoi(arg + 3);
1298         if (level > 5)
1299             error(arg);
1300         if (level >= 2) {
1301             /*
1302             * For gcc-3.4.x at -O2 we
1303             * need to disable optimizations
1304             * that break ON.
1305             */
1306             optim_disable(ctx->i_ae, level);
1307             /*
1308             * limit -xO3 to -O2 as well.
1309             */
1310             level = 2;
1311         }
1312         (void) snprintf(s, len, "-O%d", level);
1313         newae(ctx->i_ae, s);
1314         free(s);
1315         break;
1316     }
1317     error(arg);
1318     break;
1319 case 'p':
1320     if (strcmp(arg, "-xpentium") == 0) {
1321         newae(ctx->i_ae, "-march=pentium");
1322         break;
1323     }
1324     if (strcmp(arg, "-xpg") == 0) {
1325         newae(ctx->i_ae, "-pg");
1326         break;
1327     }
1328     error(arg);
1329     break;
1330 case 'r':
1331     if (strncmp(arg, "-xregs=", 7) == 0) {
1332         xlate(ctx->i_ae, arg + 7, xregs_tbl);
1333         break;
1334     }
1335     error(arg);
1336     break;
1337 case 's':

```

```

1326         if (strcmp(arg, "-xs") == 0 ||
1327             strcmp(arg, "-xspace") == 0 ||
1328             strcmp(arg, "-xstrconst") == 0)
1329             break;
1330         error(arg);
1331         break;
1332     case 't':
1333         if (strcmp(arg, "-xtransition") == 0) {
1334             newae(ctx->i_ae, "-Wtransition");
1335             break;
1336         }
1337         if (strcmp(arg, "-xtrigraphs=yes") == 0) {
1338             newae(ctx->i_ae, "-trigraphs");
1339             break;
1340         }
1341         if (strcmp(arg, "-xtrigraphs=no") == 0) {
1342             newae(ctx->i_ae, "-notrigraphs");
1343             break;
1344         }
1345         if (strncmp(arg, "-xtarget=", 9) == 0) {
1346             xlate(ctx->i_ae, arg + 9, xtarget_tbl);
1347             break;
1348         }
1349         error(arg);
1350         break;
1351     case 'e':
1352     case 'h':
1353     case 'l':
1354     default:
1355         error(arg);
1356         break;
1357     }
1358     break;
1359 case 'Y':
1360     if (arglen == 1) {
1361         if ((arg = **++ctx->i_oldargv) == NULL ||
1362             *arg == '\0')
1363             error("-Y");
1364         ctx->i_oldargc--;
1365         arglen = strlen(arg + 1);
1366     } else {
1367         arg += 2;
1368     }
1369     /* Just ignore -YS,... for now */
1370     if (strncmp(arg, "S,", 2) == 0)
1371         break;
1372     if (strncmp(arg, "l,", 2) == 0) {
1373         char *s = strdup(arg);
1374         s[0] = '-';
1375         s[1] = 'B';
1376         newae(ctx->i_ae, s);
1377         free(s);
1378         break;
1379     }
1380     if (strncmp(arg, "I,", 2) == 0) {
1381         char *s = strdup(arg);
1382         s[0] = '-';
1383         s[1] = 'I';
1384         newae(ctx->i_ae, "-nostdinc");
1385         newae(ctx->i_ae, s);
1386         free(s);
1387         break;
1388     }
1389     error(arg);
1390     break;
1391 case 'Q':

```

```

1392         /*
1393         * We could map -Qy into -Wl,-Qy etc.
1394         */
1395     default:
1396         error(arg);
1397         break;
1398     }
1399 }
1400
1401 if (c_files > 1 && (ctx->i_flags & CW_F_SHADOW) &&
1402     op != CW_O_PREPROCESS) {
1403     (void) fprintf(stderr, "%s: error: multiple source files are "
1404                    "allowed only with -E or -P\n", progname);
1405     exit(2);
1406 }
1407
1408 /*
1409 * Make sure that we do not have any unintended interactions between
1410 * the xarch options passed in and the version of the Studio compiler
1411 * used.
1412 */
1413 if ((mflag & (SS11|SS12)) == (SS11|SS12)) {
1414     (void) fprintf(stderr,
1415                    "Conflicting \"-xarch=\" flags (both Studio 11 and 12)\n");
1416     exit(2);
1417 }
1418
1419 switch (mflag) {
1420     case 0: /* FALLTHROUGH */
1421     case M32:
1422         #if defined(__sparc)
1423             /*
1424             * Only -m32 is defined and so put in the missing xarch
1425             * translation.
1426             */
1427             newae(ctx->i_ae, "-mcpu=v8");
1428             newae(ctx->i_ae, "-mno-v8plus");
1429         #endif
1430         break;
1431     case M64:
1432         #if defined(__sparc)
1433             /*
1434             * Only -m64 is defined and so put in the missing xarch
1435             * translation.
1436             */
1437             newae(ctx->i_ae, "-mcpu=v9");
1438         #endif
1439         break;
1440     case SS12:
1441         #if defined(__sparc)
1442             /* no -m32/-m64 flag used - this is an error for sparc builds */
1443             (void) fprintf(stderr, "No -m32/-m64 flag defined\n");
1444             exit(2);
1445         #endif
1446         break;
1447     case SS11:
1448         /* FALLTHROUGH */
1449     case (SS11|M32):
1450     case (SS11|M64):
1451         break;
1452     case (SS12|M32):
1453     #if defined(__sparc)
1454         /*
1455         * Need to add in further 32 bit options because with SS12
1456         * the xarch=sparcvis option can be applied to 32 or 64

```

```
1458         * bit, and so the translation table (xtbl) cannot handle
1459         * that.
1460         */
1461         newae(ctx->i_ae, "-mv8plus");
1462 #endif
1463         break;
1464     case (SSI2|M64):
1465         break;
1466     default:
1467         (void) fprintf(stderr,
1468             "Incompatible -xarch= and/or -m32/-m64 options used.\n");
1469         exit(2);
1470     }
1471     if (op == CW_O_LINK && (ctx->i_flags & CW_F_SHADOW))
1472         exit(0);
1473
1474     if (model && !pic)
1475         newae(ctx->i_ae, model);
1476     if (!nolibc)
1477         newae(ctx->i_ae, "-lc");
1478     if (!seen_o && (ctx->i_flags & CW_F_SHADOW)) {
1479         newae(ctx->i_ae, "-o");
1480         newae(ctx->i_ae, ctx->i_discard);
1481     }
1482 }
unchanged_portion_omitted
```

```

*****
2488 Tue Oct 7 15:48:35 2014
new/usr/src/uts/intel/arcfour/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # This makefile drives the production of the ARCFOUR KEF provider.
27 #
28 # Intel implementation architecture dependent
29 #
30 #
31 #
32 # Path to the base of the uts directory tree (usually /usr/src/uts).
33 #
34 UTSBASE = ../../
35 COM_DIR = $(COMMONBASE)/crypto/arcfour
36 #
37 #
38 # Define the module and object file sets.
39 #
40 MODULE = arcfour
41 LINTS = $(ARCFOURPROV_OBJS:%.o=$(LINTS_DIR)/%.ln)
42 ARCFOURPROV_OBJS_32 =
43 ARCFOURPROV_OBJS_64 = arcfour-x86_64.o
44 ARCFOURPROV_OBJS += $(ARCFOURPROV_OBJS_$(CLASS))
45 OBJECTS = $(ARCFOURPROV_OBJS:%=$(OBJSDIR)/%)
46 ROOTMODULE = $(ROOTCRYPTO_DIR)/$(MODULE)
47 #
48 #
49 # Include common rules.
50 #
51 include $(UTSBASE)/intel/Makefile.intel
52 #
53 # set signing mode
54 ELFSIGN_MOD = $(ELFSIGN_CRYPT)
55 #
56 #
57 # Define targets
58 #
59 ALL_TARGET = $(BINARY)
60 LINT_TARGET = $(MODULE).lint
61 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

```

```

63 #
64 # Overrides
65 #
66 COPTFLAG = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
67 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
68 CPPFLAGS += -I$(COM_DIR)
69 CFLAGS += -xcrossfile
66 CFLAGS += -x04 -xcrossfile
70 CLEANFILES += arcfour-x86_64.s
71 #
72 #
73 # Default build targets.
74 #
75 .KEEP_STATE:
76 #
77 def: $(DEF_DEPS)
78 #
79 all: $(ALL_DEPS)
80 #
81 clean: $(CLEAN_DEPS)
82 #
83 clobber: $(CLOBBER_DEPS)
84 #
85 lint: $(LINT_DEPS)
86 #
87 modlintlib: $(MODLINTLIB_DEPS)
88 #
89 clean.lint: $(CLEAN_LINT_DEPS)
90 #
91 install: $(INSTALL_DEPS)
92 #
93 #
94 # Include common targets.
95 #
96 include $(UTSBASE)/intel/Makefile.targ
97 #
98 $(OBJSDIR)/arcfour-x86_64.o: arcfour-x86_64.s
99 $(COMPILE.s) -o $@ ${@F:.o=.s}
100 $(POST_PROCESS_O)
101 #
102 $(OBJSDIR)/arcfour-x86_64.ln: arcfour-x86_64.s
103 @($(LHEAD) $(LINT.s) ${@F:.ln=.s}) $(LTAIL)
104 #
105 arcfour-x86_64.s: $(COM_DIR)/amd64/arcfour-x86_64.pl
106 $(PERL) $? $@

```



new/usr/src/uts/intel/bignum/Makefile.64

1

```
*****
2535 Tue Oct 7 15:48:35 2014
new/usr/src/uts/intel/bignum/Makefile.64
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # Configuration and targets for bignum module
27 # specific to AMD 64-bit architecture, amd64.
28 #
29 # Bignum configuration (BIGNUM_CFG):
30 # PSR_MUL:
31 # There is a processor-specific implementation bignum multiply functions
32 # HWCAP:
33 # There are multiple implementations of bignum functions, and the
34 # appropriate one must be chosen at run time, based on testing
35 # hardware capabilities.
36 #
37 # -DPSR_MUL:
38 # For AMD64, there is a processor-specific implementation of
39 # the bignum multiply functions, which takes advantage of the
40 # 64x64->128 bit multiply instruction.
41 #
42 # -UHWCAP:
43 # There is only one implementation, because the 128 bit multiply using
44 # general-purpose registers is faster than any MMX or SSE2 implementation.
45 #
46 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-xO4 -_gcc=-O2
47 BIGNUM_CFG = -DPSR_MUL
48 CFLAGS += -xcrossfile
49 CFLAGS += -xO4 -xcrossfile
49 $(OBJDIR)/bignumimpl.o $(LINTSDIR)/bignumimpl.ln := \
50 CPPFLAGS += $(BIGNUM_CFG)
51 $(OBJDIR)/bignum_amd64.o $(LINTSDIR)/bignum_amd64.ln := \
52 CPPFLAGS += $(BIGNUM_CFG)
53 #
54 $(OBJDIR)/bignum_amd64.o: $(BIGNUMDIR)/amd64/bignum_amd64.c
55 $(COMPILE.c) -o $@ $(BIGNUM_CFG) $(BIGNUMDIR)/amd64/bignum_amd64.c
56 $(CTFCONVERT_O)
57 $(POST_PROCESS_O)
58 #
59 $(OBJDIR)/bignum_amd64_asm.o: $(BIGNUMDIR)/amd64/bignum_amd64_asm.s
60 $(COMPILE.s) -P -o $@ $(BIGNUM_CFG) \
```

new/usr/src/uts/intel/bignum/Makefile.64

2

```
61 $(BIGNUMDIR)/amd64/bignum_amd64_asm.s
62 $(POST_PROCESS_O)
63 #
64 $(LINTSDIR)/bignum_amd64.ln: $(BIGNUMDIR)/amd64/bignum_amd64.c
65 @$(LHEAD) $(LINT.c) $(BIGNUMDIR)/amd64/bignum_amd64.c $(LTAIL))
66 #
67 $(LINTSDIR)/bignum_amd64_asm.ln: $(BIGNUMDIR)/amd64/bignum_amd64_asm.s
68 @$(LHEAD) $(LINT.s) $(BIGNUMDIR)/amd64/bignum_amd64_asm.s $(LTAIL))
```

new/usr/src/uts/intel/myril0ge/Makefile

1

```
*****
2013 Tue Oct 7 15:48:35 2014
new/usr/src/uts/intel/myril0ge/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 #

26 #
27 # Path to the base of the uts directory tree (usually /usr/src/uts).
28 #
29 UTSBASE = ../../

31 #
32 # Define the module and object file sets.
33 #
34 MODULE = myril0ge

36 OBJECTS += $(MYRI10GE_OBJS:%=$(OBJS_DIR)/%)

38 LINTS += $(MYRI10GE_OBJS:%.o=$(LINTS_DIR)/%.ln)

40 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)

42 #
43 # Include common rules.
44 #
45 include $(UTSBASE)/intel/Makefile.intel

47 #
48 # Define targets
49 #
50 ALL_TARGET = $(BINARY)
51 LINT_TARGET = $(MODULE).lint
52 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

54 #
55 # Overrides
56 #
57 LINTTAGS += -DSOLARIS_S11 -erroff=E_STATIC_UNUSED

59 #COPTIMIZE = -_cc=-xO4 -_gcc=-O2
58 #COPTIMIZE = -xO4
```

new/usr/src/uts/intel/myril0ge/Makefile

2

```
61 INC_PATH += -I$(UTSBASE)/common/io/myril0ge -I$(UTSBASE)/common/io/myril0ge/fir
63 CFLAGS += -DSOLARIS_S11

65 #
66 # Driver depends on MAC
67 #
68 LDFLAGS += -dy -N misc/mac -N drv/ip

70 #
71 # Default build targets.
72 #
73 .KEEP_STATE:

75 def: $(DEF_DEPS)

77 all: $(ALL_DEPS)

79 clean: $(CLEAN_DEPS)

81 clobber: $(CLOBBER_DEPS)

83 lint: $(LINT_DEPS)

85 modlintlib: $(MODLINTLIB_DEPS)

87 clean.lint: $(CLEAN_LINT_DEPS)

89 install: $(INSTALL_DEPS)

91 #
92 # Include common targets.
93 #
94 include $(UTSBASE)/intel/Makefile.targ
```

```

*****
2310 Tue Oct 7 15:48:35 2014
new/usr/src/uts/intel/ntxn/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 #
28 # This makefile drives the production of the Neterion Xframe
29 # 10G Ethernet (XGE) driver module in x86 systems
30 #
31 #
32 #
33 # Paths to the base of the uts directory trees
34 #
35 UTBASE = ../../
36 #
37 #
38 # Define the module and object file sets.
39 #
40 MODULE = ntxn
41 OBJECTS = $(NTXN_OBJS:%=$(OBJDIR)/%)
42 LINTS = $(NTXN_OBJS:%.o=$(LINTSDIR)/%.ln)
43 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
44 CONF_SRCDIR = $(UTSBASE)/common/io/ntxn
45 #
46 #
47 # Include common rules.
48 #
49 include $(UTSBASE)/intel/Makefile.intel
50 #
51 #
52 # Define targets
53 #
54 ALL_TARGET = $(BINARY) $(SRC_CONF_FILE)
55 LINT_TARGET = $(MODULE).lint
56 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONF_FILE)
57 #
58 INC_PATH += -I$(CONF_SRCDIR)
59 #
60 COPTFLAG = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
61 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2

```

```

62 CFLAGS += -DSOLARIS11
63 CFLAGS64 += -DSOLARIS11 -I./
64 CFLAGS += -DSOLARIS11 -xO4
65 CFLAGS64 += -DSOLARIS11 -xO -I./
66 #
67 # Driver depends on MAC & IP
68 LDFLAGS += -dy -N misc/mac -N drv/ip
69 #
70 # Lint flag
71 #
72 LINTFLAGS += $(NTXN_DFLAGS) $(NTXN_KFLAGS)
73 LINTFLAGS += -DSOLARIS11
74 #
75 CERRWARN += -_gcc=-Wno-parentheses
76 CERRWARN += -_gcc=-Wno-unused-label
77 CERRWARN += -_gcc=-Wno-type-limits
78 #
79 #
80 #
81 # Default build targets.
82 #
83 .KEEP_STATE:
84 #
85 def: $(DEF_DEPS)
86 #
87 all: $(ALL_DEPS)
88 #
89 clean: $(CLEAN_DEPS)
90 #
91 clobber: $(CLOBBER_DEPS)
92 #
93 lint: $(LINT_DEPS)
94 #
95 modlintlib: $(MODLINTLIB_DEPS)
96 #
97 clean.lint: $(CLEAN_LINT_DEPS)
98 #
99 install: $(INSTALL_DEPS)
100 #
101 #
102 # Include common targets.
103 #
104 include $(UTSBASE)/intel/Makefile.targ

```

```

*****
3688 Tue Oct 7 15:48:35 2014
new/usr/src/uts/intel/xge/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # This makefile drives the production of the Neterion Xframe
28 # 10G Ethernet (XGE) driver module in x86 systems
29 #
30 #
31 #
32 # Paths to the base of the uts directory trees
33 #
34 UTSBASE = ../../
35 #
36 #
37 # Define the module and object file sets.
38 #
39 MODULE = xge
40 OBJECTS = $(XGE_HAL_OBJS:%=$(OBJS_DIR)/%) $(XGE_OBJS:%=$(OBJS_DIR)/%)
41 LINTS = $(XGE_HAL_OBJS%.o=$(LINTS_DIR)/%.ln) $(XGE_OBJS%.o=$(LINTS_D
42 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
43 #
44 #
45 # Include common rules.
46 #
47 include $(UTSBASE)/intel/Makefile.intel
48 #
49 #
50 # Define targets
51 #
52 ALL_TARGET = $(BINARY)
53 LINT_TARGET = $(MODULE).lint
54 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)
55 #
56 #
57 # GENERAL PURPOSE HAL FLAGS: Tuning HAL for Solaris specific modes
58 #
59 HAL_CFLAGS = -DXGE_HAL_USE_MGMT_AUX
60 #
61 #

```

```

62 # TRACE SECTION: Possible values for MODULE, TRACE and ERR masks:
63 #
64 # XGE_COMPONENT_HAL_CONFIG 0x1
65 # XGE_COMPONENT_HAL_FIFO 0x2
66 # XGE_COMPONENT_HAL_RING 0x4
67 # XGE_COMPONENT_HAL_CHANNEL 0x8
68 # XGE_COMPONENT_HAL_DEVICE 0x10
69 # XGE_COMPONENT_HAL_MM 0x20
70 # XGE_COMPONENT_HAL_QUEUE 0x40
71 # XGE_COMPONENT_HAL_STATS 0x100
72 # XGE_COMPONENT_OSDEP 0x1000
73 # XGE_COMPONENT_LL 0x2000
74 # XGE_COMPONENT_TOE 0x4000
75 # XGE_COMPONENT_RDMA 0x8000
76 # XGE_COMPONENT_ALL 0xffffffff
77 #TRACE_CFLAGS = -DXGE_DEBUG_MODULE_MASK=0xffffffff \
78 # -DXGE_DEBUG_TRACE_MASK=0xffffffff \
79 # -DXGE_DEBUG_ERR_MASK=0xffffffff
80 TRACE_CFLAGS = -DXGE_DEBUG_MODULE_MASK=0x00003010 \
81 # -DXGE_DEBUG_TRACE_MASK=0x00000000 \
82 # -DXGE_DEBUG_ERR_MASK=0x00003010
83 #
84 XGE_CFLAGS = $(HAL_CFLAGS) $(TRACE_CFLAGS) $(CCVERBOSE) \
85 # -I$(UTSBASE)/common/io/xge/hal/include \
86 # -I$(UTSBASE)/common/io/xge/hal/xgehal \
87 # -I$(UTSBASE)/common/io/xge/drv -DSOLARIS
88 #
89 COPTFLAG = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
90 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
91 CFLAGS += $(XGE_CFLAGS) -xcrossfile -xc99=%all
92 CFLAGS64 += $(XGE_CFLAGS) -xcrossfile -xc99=%all
93 #
94 #
95 # Driver depends on MAC & IP
96 #
97 LDFLAGS += -dy -N misc/mac -N drv/ip
98 #
99 # Lint flag
100 #
101 LINTFLAGS += $(XGE_CFLAGS) -xc99=%all
102 #
103 #
104 # For now, disable these lint checks; maintainers should endeavor
105 # to investigate and remove these for maximum lint coverage.
106 # Please do not carry these forward to new Makefiles.
107 #
108 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
109 LINTTAGS += -erroff=E_STATIC_UNUSED
110 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
111 #
112 CERRWARN += -_gcc=-Wno-parentheses
113 CERRWARN += -_gcc=-Wno-unused-variable
114 CERRWARN += -_gcc=-Wno-unused-label
115 CERRWARN += -_gcc=-Wno-empty-body
116 CERRWARN += -_gcc=-Wno-uninitialized
117 #
118 #
119 #
120 # Default build targets.
121 #
122 .KEEP_STATE:
123 #
124 def: $(DEF_DEPS)

```

**new/usr/src/uts/intel/xge/Makefile**

**3**

```
126 all:          $(ALL_DEPS)
128 clean:        $(CLEAN_DEPS)
130 clobber:      $(CLOBBER_DEPS)
132 lint:         $(LINT_DEPS)
134 modlintlib:   $(MODLINTLIB_DEPS)
136 clean.lint:   $(CLEAN_LINT_DEPS)
138 install:      $(INSTALL_DEPS)

140 #
141 #   Include common targets.
142 #
143 include $(UTSBASE)/intel/Makefile.targ
```

new/usr/src/uts/sparc/Makefile.sparc

1

```
*****
13396 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sparc/Makefile.sparc
5196 The cw wrapper restricts gcc to -O2
*****

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

22 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright (c) 2013 Andrew Stormont. All rights reserved.
24 # Copyright 2014 Gary Mills

26 #
27 # This makefile contains the common definitions for all sparc
28 # implementation architecture independent modules.
29 #

31 #
32 # Define supported builds
33 #
34 DEF_BUILDS = $(DEF_BUILDS64)
35 ALL_BUILDS = $(ALL_BUILDS64)

37 #
38 # Everybody needs to know how to build modstubs.o and to locate unix.o.
39 # Note that unix.o must currently be selected from among the possible
40 # "implementation architectures". Note further, that unix.o is only
41 # used as an optional error check for undefines so (theoretically)
42 # any "implementation architectures" could be used. We choose sun4u
43 # because it is the reference port.
44 #
45 UNIX_DIR = $(UTSBASE)/sun4u/unix
46 GENLIB_DIR = $(UTSBASE)/sun4u/genunix
47 IPDRV_DIR = $(UTSBASE)/sparc/ip
48 MODSTUBS_DIR = $(UNIX_DIR)
49 DSF_DIR = $(UNIX_DIR)
50 LINTS_DIR = $(OBJS_DIR)
51 LINT_LIB_DIR = $(UTSBASE)/sparc/lint-libs/$(OBJS_DIR)

53 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
54 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
55 GENLIB = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/libgenunix.so

57 LINT_LIB_32 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lunix.ln
58 GEN_LINT_LIB_32 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln

60 LINT_LIB_64 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lunix.ln
```

new/usr/src/uts/sparc/Makefile.sparc

2

```
61 GEN_LINT_LIB_64 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln
63 LINT_LIB = $(LINT_LIB_$(CLASS))
64 GEN_LINT_LIB = $(GEN_LINT_LIB_$(CLASS))

66 LINT32_DIRS = $(LINT32_BUILDS:=$(UTSBASE)/sparc/lint-libs/%)
67 LINT32_FILES = $(LINT32_DIRS:=%/llib-l$(MODULE).ln)

69 LINT64_DIRS = $(LINT64_BUILDS:=$(UTSBASE)/sparc/lint-libs/%)
70 LINT64_FILES = $(LINT64_DIRS:=%/llib-l$(MODULE).ln)

72 #
73 # Include the makefiles which define build rule templates, the
74 # collection of files per module, and a few specific flags. Note
75 # that order is significant, just as with an include path. The
76 # first build rule template which matches the files name will be
77 # used. By including these in order from most machine dependent
78 # to most machine independent, we allow a machine dependent file
79 # to be used in preference over a machine independent version
80 # (Such as a machine specific optimization, which preserves the
81 # interfaces.)
82 #
83 include $(UTSBASE)/sparc/Makefile.files
84 include $(UTSBASE)/sparc/v9/Makefile.files
85 include $(UTSBASE)/sun/Makefile.files
86 include $(UTSBASE)/common/Makefile.files

88 #
89 # ----- TRANSITIONAL SECTION -----
90 #

92 #
93 # Not everything which *should* be a module is a module yet. The
94 # following is a list of such objects which are currently part of
95 # genunix but which might someday become kmods. This must be
96 # defined before we include Makefile.uts, or else genunix's build
97 # won't be as parallel as we might like.
98 #
99 NOT_YET_KMODS = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)

101 #
102 # ----- END OF TRANSITIONAL SECTION -----
103 #
104 # Include machine independent rules. Note that this does not imply
105 # that the resulting module from rules in Makefile.uts is machine
106 # independent. Only that the build rules are machine independent.
107 #
108 include $(UTSBASE)/Makefile.uts

110 #
111 # machine specific optimization, override default in Makefile.master
112 #
113 XARCH_32 = -xarch=v8
114 XARCH_64 = -m64
115 XARCH = $(XARCH_$(CLASS))

117 COPTIMIZE_32 = -_cc=-xO3 -_gcc=-O2
118 COPTIMIZE_64 = -_cc=-xO3 -_gcc=-O2
119 COPTIMIZE_32 = -xO3
120 COPTIMIZE_64 = -xO3
121 COPTIMIZE = $(COPTIMIZE_$(CLASS))

121 CCMODE = -Xa

123 CFLAGS_32 = -xcg92
124 CFLAGS_64 = -xchip=ultra $(CCABS32) $(CCREGSYM)
```

```

125 CFLAGS          = $(CFLAGS_$(CLASS))

127 CFLAGS          += $(XARCH)
128 CFLAGS          += $(COPTIMIZE)
129 CFLAGS          += $(EXTRA_CFLAGS)
130 CFLAGS          += $(XAOPT)
131 CFLAGS          += $(INLINES) -D_ASM_INLINES
132 CFLAGS          += $(CCMODE)
133 CFLAGS          += $(SPACEFLAG)
134 CFLAGS          += $(CERRWARN)
135 CFLAGS          += $(CTF_FLAGS_$(CLASS))
136 CFLAGS          += $(C99MODE)
137 CFLAGS          += $(CCUNBOUND)
138 CFLAGS          += $(CCSTATICSYM)
139 CFLAGS          += $(CC32BITCALLERS)
140 CFLAGS          += $(CCNOAUTOINLINE)
141 CFLAGS          += $(IROPTFLAG)
142 CFLAGS          += $(CGLOBALSTATIC)
143 CFLAGS          += -xregs=no%float
144 CFLAGS          += -xstrconst
145 CFLAGS          += $(CSOURCEDEBUGFLAGS)
146 CFLAGS          += $(USERFLAGS)

148 ASFLAGS         += $(XARCH)

150 LINT_DEFS_32    =
151 LINT_DEFS_64    = -m64
152 LINT_DEFS       += $(LINT_DEFS_$(CLASS))

154 #
155 #   The following must be defined for all implementations:
156 #
157 #   MODSTUBS:      Module stubs source file.
158 #
159 MODSTUBS         = $(UTSBASE)/sparc/ml/modstubs.s

161 #
162 #   Define the actual specific platforms - obviously none.
163 #
164 MACHINE_DEFS     =

166 #
167 #   Debugging level
168 #
169 #   Special knowledge of which special debugging options effect which
170 #   file is used to optimize the build if these flags are changed.
171 #
172 #   XXX: The above could possibly be done for more flags and files, but
173 #   is left as an experiment to the interested reader. Be forewarned,
174 #   that excessive use could lead to maintenance difficulties.
175 #
176 DEBUG_DEFS_OBJ32 =
177 DEBUG_DEFS_DBG32 = -DDEBUG
178 DEBUG_DEFS_OBJ64 =
179 DEBUG_DEFS_DBG64 = -DDEBUG
180 DEBUG_DEFS       = $(DEBUG_DEFS_$(BUILD_TYPE))

182 DEBUG_COND_OBJ32 = $(POUND_SIGN)
183 DEBUG_COND_DBG32 =
184 DEBUG_COND_OBJ64 = $(POUND_SIGN)
185 DEBUG_COND_DBG64 =
186 IF_DEBUG_OBJ     = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

188 $(IF_DEBUG_OBJ)syscall.o      :=      DEBUG_DEFS      += -DSYSCALLTRACE
189 $(IF_DEBUG_OBJ)clock.o       :=      DEBUG_DEFS      += -DKSLICE=1

```

```

191 # Comment these out if you don't want dispatcher lock statistics.

193 # $(IF_DEBUG_OBJ)disp_lock.o      := DEBUG_DEFS      += -DDISP_LOCK_STATS

195 #
196 #   Collect the preprocessor definitions to be associated with *all*
197 #   files.
198 #
199 ALL_DEFS         = $(MACHINE_DEFS) $(DEBUG_DEFS) $(OPTION_DEFS)
200 #
201 #
202 #   The kernels modules which are "implementation architecture"
203 #   specific for this machine are enumerated below. Note that most
204 #   of these modules must exist (in one form or another) for each
205 #   architecture.
206 #
207 #   Common Drivers (usually pseudo drivers) (/kernel/drv):
208 #
209 DRV_KMODS        += aggr arp audio bl blkdev bofi clone cn conskbd consms cpuid
210 DRV_KMODS        += crypto cryptoadm devinfo dump
211 DRV_KMODS        += dtrace fasttrap fbt lockstat profile sdt systrace dcpc
212 DRV_KMODS        += fssnap icmp icmp6 ip ip6 ipnet ipsecah
213 DRV_KMODS        += ipsecesp iptun iwscn keysock kmdb kstat ksyms llc1
214 DRV_KMODS        += lofi
215 DRV_KMODS        += log logindmux kssl mm nca physmem pm poll pool
216 DRV_KMODS        += pseudo ptc ptm pts ptsl ramdisk random rsm rts sad
217 DRV_KMODS        += simnet softmac sPPP sPpPttun sy sysevent sysmsg
218 DRV_KMODS        += spdsock
219 DRV_KMODS        += tcp tcp6 tl tnf ttymux udp udp6 wc winlock zcons
220 DRV_KMODS        += ippctl
221 DRV_KMODS        += dld
222 DRV_KMODS        += ipd
223 DRV_KMODS        += ipf
224 DRV_KMODS        += rpcib
225 DRV_KMODS        += dlpistub
226 DRV_KMODS        += vnic
227 DRV_KMODS        += xge
228 DRV_KMODS        += rds
229 DRV_KMODS        += rdsv3
230 DRV_KMODS        += chxge
231 DRV_KMODS        += smbsrv
232 DRV_KMODS        += vscan
233 DRV_KMODS        += nsmb
234 DRV_KMODS        += fm
235 DRV_KMODS        += nulldriver
236 DRV_KMODS        += bridge trill
237 DRV_KMODS        += bpf
238 DRV_KMODS        += dca

240 #
241 #   Hardware Drivers in common space
242 #
244 DRV_KMODS        += afe
245 DRV_KMODS        += audio1575
246 DRV_KMODS        += audioens
247 DRV_KMODS        += audiols
248 DRV_KMODS        += audiopl6x
249 DRV_KMODS        += audiopci
250 DRV_KMODS        += audiots
251 DRV_KMODS        += e1000g
252 DRV_KMODS        += efe
253 DRV_KMODS        += hxge
254 DRV_KMODS        += mxfe
255 DRV_KMODS        += rge
256 DRV_KMODS        += rtls

```

```

257 DRV_KMODS      += sfe
258 DRV_KMODS      += aac
259 DRV_KMODS      += igb
260 DRV_KMODS      += ixgbe
261 DRV_KMODS      += vr
262 DRV_KMODS      += mr_sas
263 DRV_KMODS      += yge

265 #
266 #      Machine Specific Driver Modules (/kernel/drv):
267 #
268 DRV_KMODS      += audiocs
269 DRV_KMODS      += bge dmfe eri fas hme qfe
270 DRV_KMODS      += openepr options sd ses st
271 DRV_KMODS      += ssd
272 DRV_KMODS      += ecpp
273 DRV_KMODS      += hid hubd ehci ohci uhci usb_mid usb_ia scsa2usb usbprn ugen
274 DRV_KMODS      += usbser usbsacm usbsksp usbsprl
275 DRV_KMODS      += usb_as usb_ac
276 DRV_KMODS      += usbskel
277 DRV_KMODS      += usbvc
278 DRV_KMODS      += usbftdi
279 DRV_KMODS      += wusb_df hwahc hwarc wusb_ca
280 DRV_KMODS      += usbecm
281 DRV_KMODS      += hcil394 avl394 scsa1394 dcaml394
282 DRV_KMODS      += sbp2
283 DRV_KMODS      += ib ibp eibnx eoib rdsib sdp iser daplt hermon tavor sol_ucma
284 DRV_KMODS      += sol_umad
285 DRV_KMODS      += pci_pci pcieb pcieb_bcm
286 DRV_KMODS      += i8042 kb8042 mouse8042
287 DRV_KMODS      += fcode
288 DRV_KMODS      += mpt_sas
289 DRV_KMODS      += social
290 DRV_KMODS      += sgen
291 DRV_KMODS      += myril0ge
292 DRV_KMODS      += smp
293 DRV_KMODS      += dad
294 DRV_KMODS      += scsi_vhci
295 DRV_KMODS      += fcp
296 DRV_KMODS      += fcip
297 DRV_KMODS      += fcsm
298 DRV_KMODS      += fp
299 DRV_KMODS      += qlc
300 DRV_KMODS      += qlge
301 DRV_KMODS      += stmf
302 DRV_KMODS      += stmf_sbd
303 DRV_KMODS      += fct
304 DRV_KMODS      += fcoe
305 DRV_KMODS      += fcoet
306 DRV_KMODS      += fcoei
307 DRV_KMODS      += qlt
308 DRV_KMODS      += iscsit
309 DRV_KMODS      += pppt
310 DRV_KMODS      += ncall nctl sdbc nskern sv
311 DRV_KMODS      += ii rdc rdcsrv rdcstub
312 DRV_KMODS      += iscsi
313 DRV_KMODS      += emlxs
314 DRV_KMODS      += oce
315 DRV_KMODS      += srpt
316 DRV_KMODS      += pmcs
317 DRV_KMODS      += pmcs8001fw

319 #
320 #      I/O framework test drivers
321 #
322 DRV_KMODS      += pshot

```

```

323 DRV_KMODS      += gen_drv
324 DRV_KMODS      += tvhci tphci tclient
325 DRV_KMODS      += emul64

327 #
328 #      PCMCIA specific module(s)
329 #
330 DRV_KMODS      += pcs
331 MISC_KMODS     += busra cardbus dada pcmcia
332 DRV_KMODS      += pcic

334 # Add lvm
335 #
336 DRV_KMODS      += md
337 MISC_KMODS     += md_mirror md_stripe md_hotspares md_raid md_trans md_notify
338 MISC_KMODS     += md_sp

340 #
341 #      Exec Class Modules (/kernel/exec):
342 #
343 EXEC_KMODS     += aoutexec elfexec intpexec shbinexec javaexec

345 #
346 #      Scheduling Class Modules (/kernel/sched):
347 #
348 SCHED_KMODS    += RT TS RT_DPTBL TS_DPTBL IA FSS FX FX_DPTBL SDC

350 #
351 #      File System Modules (/kernel/fs):
352 #
353 FS_KMODS       += dev devfs fdfs fifofs hsfhs lofs namefs nfs pcfs tmpfs zfs
354 FS_KMODS       += zut specfs udfs ufs autofs cacheofs procfs sockfs mntfs
355 FS_KMODS       += ctfs objfs sharefs dcfs smbfs

357 #
358 #      Streams Modules (/kernel/strmod):
359 #
360 STRMOD_KMODS   += bufmod connld dedump ldterm ms pckt pfmod
361 STRMOD_KMODS   += pipemod ptem redirmod rpcmod rlmod telmod timod
362 STRMOD_KMODS   += sppsasyn sppscomp
363 STRMOD_KMODS   += tirdwr ttcompat
364 STRMOD_KMODS   += usbbkm usbms usbwcm usb_ah
365 STRMOD_KMODS   += drcompat
366 STRMOD_KMODS   += cryptmod
367 STRMOD_KMODS   += vuid3ps2

369 #
370 #      'System' Modules (/kernel/sys):
371 #
372 SYS_KMODS      += c2audit
373 SYS_KMODS      += exacctsys
374 SYS_KMODS      += inst_sync kaio msgsys semsys shmsys sysacct pipe
375 SYS_KMODS      += doorfs pset acctctl portfs

377 #
378 #      'User' Modules (/kernel/misc):
379 #
380 MISC_KMODS     += ac97
381 MISC_KMODS     += bignum
382 MISC_KMODS     += consconfig gld ipc nfs_dlboot nfssrv scsi
383 MISC_KMODS     += strplumb swapgeneric tlimod
384 MISC_KMODS     += rpcsec rpcsec_gss kgssapi kmecch_dummy
385 MISC_KMODS     += kmecch_krb5
386 MISC_KMODS     += fssnap_if
387 MISC_KMODS     += hidparser kbtrans usba usbal0 usbs49_fw
388 MISC_KMODS     += sl394

```



```

389 MISC_KMODS      += hpcsvc pcihp
390 MISC_KMODS      += rsmops
391 MISC_KMODS      += kcf
392 MISC_KMODS      += ksocket
393 MISC_KMODS      += ibcm
394 MISC_KMODS      += ibdm
395 MISC_KMODS      += ibdma
396 MISC_KMODS      += ibmf
397 MISC_KMODS      += ibtl
398 MISC_KMODS      += sol_ofs
399 MISC_KMODS      += idm
400 MISC_KMODS      += idmap
401 MISC_KMODS      += hook
402 MISC_KMODS      += neti
403 MISC_KMODS      += ctf
404 MISC_KMODS      += mac dls
405 MISC_KMODS      += cmlb
406 MISC_KMODS      += tem
407 MISC_KMODS      += pcicfg fcodem fcpci
408 MISC_KMODS      += scsi_vhci_f_sym scsi_vhci_f_tpgs scsi_vhci_f_asym_sun
409 MISC_KMODS      += scsi_vhci_f_sym_hds
410 MISC_KMODS      += scsi_vhci_f_tape scsi_vhci_f_tpgs_tape
411 MISC_KMODS      += fctl
412 MISC_KMODS      += emlxs_fw
413 MISC_KMODS      += qlc_fw_2200
414 MISC_KMODS      += qlc_fw_2300
415 MISC_KMODS      += qlc_fw_2400
416 MISC_KMODS      += qlc_fw_2500
417 MISC_KMODS      += qlc_fw_6322
418 MISC_KMODS      += qlc_fw_8100
419 MISC_KMODS      += spuni
420 MISC_KMODS      += hwa1480_fw uwba
421 MISC_KMODS      += mii

423 MISC_KMODS      += klmmod klmops

425 #
426 #       Software Cryptographic Providers (/kernel/crypto):
427 #
428 CRYPTO_KMODS     += aes
429 CRYPTO_KMODS     += arcfour
430 CRYPTO_KMODS     += blowfish
431 CRYPTO_KMODS     += des
432 CRYPTO_KMODS     += md4
433 CRYPTO_KMODS     += md5
434 CRYPTO_KMODS     += ecc
435 CRYPTO_KMODS     += rsa
436 CRYPTO_KMODS     += sha1
437 CRYPTO_KMODS     += sha2
438 CRYPTO_KMODS     += swrand

440 #
441 # IP Policy Modules (/kernel/ipp):
442 #
443 IPP_KMODS         += dlcosmk
444 IPP_KMODS         += flowacct
445 IPP_KMODS         += ipgpc
446 IPP_KMODS         += dscpmk
447 IPP_KMODS         += tokenmt
448 IPP_KMODS         += tswtclmt

450 #
451 # 'Dacf' modules (/kernel/dacf)
452 DACF_KMODS       += consconfig_dacf

454 #

```

```

455 #       SVVS Testing Modules (/kernel/strmod):
456 #
457 #       These are streams and driver modules which are not to be
458 #       delivered with a released system. However, during development
459 #       it is convenient to build and install the SVVS kernel modules.
460 #
461 SVVS_KMODS        += lmodb lmode lmodr lmodt svvslo tidg tivc tmutex

463 #
464 #       Modules eXcluded from the product:
465 #
466 XMODS             +=

468 #
469 #       'Dacf' Modules (/kernel/dacf):
470 #
471 DACF_KMODS        += net_dacf

473 #
474 #       MAC-Type Plugin Modules (/kernel/mac)
475 #
476 MAC_KMODS         += mac_6to4
477 MAC_KMODS         += mac_ether
478 MAC_KMODS         += mac_ipv4
479 MAC_KMODS         += mac_ipv6
480 MAC_KMODS         += mac_wifi
481 MAC_KMODS         += mac_ib

483 #
484 # socketmod (kernel/socketmod)
485 #
486 SOCKET_KMODS      += sockpfp
487 SOCKET_KMODS      += socksctp
488 SOCKET_KMODS      += socksdp
489 SOCKET_KMODS      += sockrds
490 SOCKET_KMODS      += kssif

492 #
493 #       kiconv modules (/kernel/kiconv):
494 #
495 KICONV_KMODS      += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc

497 #
498 # Ensure that the variable member of the cpu_t (cpu_m) is defined
499 # for the lint builds so as not to cause lint errors during the
500 # global cross check.
501 #
502 $(LINTFLAGSSUPPRESS)LINTFLAGS += -D_MACHDEP -I$(UTSBASE)/sun4 \
503                                -I$(UTSBASE)/sun4u -I$(UTSBASE)/sfmmu

```

```

*****
2576 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sparc/dld/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #

27 #
28 # Path to the base of the uts directory tree (usually /usr/src/uts).
29 #
30 UTSBASE = ../../

32 #
33 # Define the module and object file sets.
34 #
35 MODULE          = dld
36 OBJECTS         = $(DLD_OBJS:%=$(OBJS_DIR)/%)
37 LINTS           = $(DLD_OBJS:%.o=$(LINTS_DIR)/%.ln)
38 ROOTMODULE     = $(ROOT_DRV_DIR)/$(MODULE)
39 CONF_SRCDIR    = $(UTSBASE)/common/io/dld
40 ROOTDEVLINK    = $(ROOT)/dev/$(MODULE)

42 #
43 # Include common rules.
44 #
45 include $(UTSBASE)/sparc/Makefile.sparc

47 #
48 # Define targets
49 #
50 ALL_TARGET      = $(BINARY) $(SRC_CONFILE)
51 LINT_TARGET     = $(MODULE).lint
52 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFILE)

54 #
55 # Overrides
56 #
57 CFLAGS          += $(CCVERBOSE)
58 $(RELEASE_BUILD)CFLAGS += -xinline=auto -xcrossfile
59 $(RELEASE_BUILD)COPTIMIZE = -_cc=-xO5 -_gcc=-O2
58 $(RELEASE_BUILD)COPTIMIZE = -xO5
60 LDFLAGS        += -dy -N misc/dls -N misc/mac

```

```

61 INC_PATH          += -I$(UTSBASE)/common/io/bpf

63 #
64 # For now, disable these lint checks; maintainers should endeavor
65 # to investigate and remove these for maximum lint coverage.
66 # Please do not carry these forward to new Makefiles.
67 #
68 LINTTAGS          += -erroff=E_BAD_PTR_CAST_ALIGN
69 LINTTAGS          += -erroff=E_PTRDIFF_OVERFLOW
70 LINTTAGS          += -erroff=E_ASSIGN_NARROW_CONV
71 LINTTAGS          += -erroff=E_SUSPICIOUS_COMPARISON

73 CERRWARN          += -_gcc=-Wno-uninitialized
74 CERRWARN          += -_gcc=-Wno-type-limits
75 CERRWARN          += -_gcc=-Wno-parentheses

77 #
78 #      Default build targets.
79 #
80 .KEEP_STATE:

82 def:              $(DEF_DEPS)

84 all:              $(ALL_DEPS)

86 clean:            $(CLEAN_DEPS)

88 clobber:          $(CLOBBER_DEPS)

90 lint:             $(LINT_DEPS)

92 modlintlib:      $(MODLINTLIB_DEPS)

94 clean.lint:      $(CLEAN_LINT_DEPS)

96 install:         $(INSTALL_DEPS) $(ROOTDEVLINK)

98 $(ROOTDEVLINK):
99     -$(RM) $@; $(SYMLINK) ../devices/pseudo/$(@F)@0:ctl $@

101 #
102 #      Include common targets.
103 #
104 include $(UTSBASE)/sparc/Makefile.targ

```

```

*****
2293 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sparc/dls/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #

27 #
28 # Path to the base of the uts directory tree (usually /usr/src/uts).
29 #
30 UTSBASE = ../../

32 #
33 # Define the module and object file sets.
34 #
35 MODULE          = dls
36 OBJECTS         = $(DLS_OBJS:%=$(OBJS_DIR)/%)
37 LINTS           = $(DLS_OBJS:%.o=$(LINTS_DIR)/%.ln)
38 ROOTMODULE     = $(ROOT_MISC_DIR)/$(MODULE)

40 #
41 # Include common rules.
42 #
43 include $(UTSBASE)/sparc/Makefile.sparc

45 #
46 # Define targets
47 #
48 ALL_TARGET      = $(BINARY)
49 LINT_TARGET     = $(MODULE).lint
50 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

52 #
53 # Overrides.
54 #
55 CFLAGS          += $(CCVERBOSE)
56 $(RELEASE_BUILD)CFLAGS += -xinline=auto -xcrossfile
57 $(RELEASE_BUILD)COPTIMIZE = -_cc=-xO5 -_gcc=-O2
56 $(RELEASE_BUILD)COPTIMIZE = -xO5
58 LDFLAGS        += -dy -N misc/mac
59 INC_PATH       += -I$(UTSBASE)/common/io/bpf

```

```

61 #
62 # For now, disable these lint checks; maintainers should endeavor
63 # to investigate and remove these for maximum lint coverage.
64 # Please do not carry these forward to new Makefiles.
65 #
66 LINTTAGS        += -erroff=E_BAD_PTR_CAST_ALIGN
67 LINTTAGS        += -erroff=E_PTRDIFF_OVERFLOW
68 LINTTAGS        += -erroff=E_ASSIGN_NARROW_CONV
69 LINTTAGS        += -erroff=E_STATIC_UNUSED

71 CERRWARN        += -_gcc=-Wno-uninitialized

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 #
95 #           Include common targets.
96 #
97 include $(UTSBASE)/sparc/Makefile.targ

```

```

*****
2326 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sparc/mac/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #
27 # This makefile drives the production of the mac driver
28 # kernel module.
29 #
30 # sparc architecture dependent
31 #
32 #
33 # Path to the base of the uts directory tree (usually /usr/src/uts).
34 #
35 UTSBASE = ../../

37 #
38 # Define the module and object file sets.
39 #
40 MODULE = mac
41 OBJECTS = $(MAC_OBJS:%=$(OBJSDIR)/%)
42 LINTS = $(MAC_OBJS:%.o=$(LINTSDIR)/%.ln)
43 ROOTMODULE = $(ROOTMISC_DIR)/$(MODULE)

45 #
46 # Include common rules.
47 #
48 include $(UTSBASE)/sparc/Makefile.sparc

50 #
51 # Define targets
52 #
53 ALL_TARGET = $(BINARY)
54 LINT_TARGET = $(MODULE).lint
55 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

57 #
58 # Overrides.
59 #
60 CFLAGS += $(CCVERBOSE)
61 $(RELEASE_BUILD)CFLAGS += -xinline=auto -xcrossfile

```

```

62 $(RELEASE_BUILD)COPTIMIZE = -_cc=-xO5 -_gcc=-O2
61 $(RELEASE_BUILD)COPTIMIZE = -xO5
63 LDFLAGS += -dy
64 INC_PATH += -I$(UTSBASE)/common/io/bpf

66 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
67 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN

69 CERRWARN += -_gcc=-Wno-unused-label
70 CERRWARN += -_gcc=-Wno-uninitialized
71 CERRWARN += -_gcc=-Wno-parentheses
72 CERRWARN += -_gcc=-Wno-type-limits
73 CERRWARN += -_gcc=-Wno-switch
74 CERRWARN += -_gcc=-Wno-unused-variable

76 #
77 # Default build targets.
78 #
79 .KEEP_STATE:

81 def: $(DEF_DEPS)

83 all: $(ALL_DEPS)

85 clean: $(CLEAN_DEPS)

87 clobber: $(CLOBBER_DEPS)

89 lint: $(LINT_DEPS)

91 modlintlib: $(MODLINTLIB_DEPS)

93 clean.lint: $(CLEAN_LINT_DEPS)

95 install: $(INSTALL_DEPS)

97 #
98 # Include common targets.
99 #
100 include $(UTSBASE)/sparc/Makefile.targ

```

new/usr/src/uts/sparc/myril0ge/Makefile

1

\*\*\*\*\*

2039 Tue Oct 7 15:48:36 2014

new/usr/src/uts/sparc/myril0ge/Makefile

5196 The cw wrapper restricts gcc to -O2

\*\*\*\*\*

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2014 Gary Mills
23 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #

27 #
28 # Path to the base of the uts directory tree (usually /usr/src/uts).
29 #
30 UTSBASE = ../../

32 #
33 # Define the module and object file sets.
34 #
35 MODULE = myril0ge

37 OBJECTS += $(MYRI10GE_OBJS:%=$(OBJSDIR)/%)

39 LINTS += $(MYRI10GE_OBJS:%.o=$(LINTSDIR)/%.ln)

41 ROOTMODULE = $(ROOTDRVDIR)/$(MODULE)

43 #
44 # Include common rules.
45 #
46 include $(UTSBASE)/sparc/Makefile.sparc

48 #
49 # Define targets
50 #
51 ALL_TARGET = $(BINARY)
52 LINT_TARGET = $(MODULE).lint
53 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

55 #
56 # Overrides
57 #
58 LINTTAGS += -DSOLARIS_S11 -erroff=E_STATIC_UNUSED

60 #OPTIMIZE = -_cc=-xO4 -_gcc=-O2
59 #OPTIMIZE = -xO4
```

new/usr/src/uts/sparc/myril0ge/Makefile

2

```
62 INC_PATH += -I$(UTSBASE)/common/io/myril0ge -I$(UTSBASE)/common/io/myril0ge/fir

64 CFLAGS += -dalign -DSOLARIS_S11

66 #
67 # Driver depends on MAC
68 #
69 LDFLAGS += -dy -N misc/mac -N drv/ip

71 #
72 # Default build targets.
73 #
74 .KEEP_STATE:

76 def: $(DEF_DEPS)

78 all: $(ALL_DEPS)

80 clean: $(CLEAN_DEPS)

82 clobber: $(CLOBBER_DEPS)

84 lint: $(LINT_DEPS)

86 modlintlib: $(MODLINTLIB_DEPS)

88 clean.lint: $(CLEAN_LINT_DEPS)

90 install: $(INSTALL_DEPS)

92 #
93 # Include common targets.
94 #
95 include $(UTSBASE)/sparc/Makefile.targ
```

```

*****
3688 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sparc/xge/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2014 Gary Mills
24 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # This makefile drives the production of the Neterion Xframe
28 # 10G Ethernet (XGE) driver module in x86 systems
29 #
30 #
31 #
32 # Paths to the base of the uts directory trees
33 #
34 UTSBASE = ../../
35 #
36 #
37 # Define the module and object file sets.
38 #
39 MODULE = xge
40 OBJECTS = $(XGE_HAL_OBJS:%=$(OBJS_DIR)/%) $(XGE_OBJS:%=$(OBJS_DIR)/%)
41 LINTS = $(XGE_HAL_OBJS%.o=$(LINTS_DIR)/%.ln) $(XGE_OBJS%.o=$(LINTS_D
42 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
43 #
44 #
45 # Include common rules.
46 #
47 include $(UTSBASE)/sparc/Makefile.sparc
48 #
49 #
50 # Define targets
51 #
52 ALL_TARGET = $(BINARY)
53 LINT_TARGET = $(MODULE).lint
54 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)
55 #
56 #
57 # GENERAL PURPOSE HAL FLAGS: Tuning HAL for Solaris specific modes
58 #
59 HAL_CFLAGS = -DXGE_HAL_USE_MGMT_AUX
60 #
61 #

```

```

62 # TRACE SECTION: Possible values for MODULE, TRACE and ERR masks:
63 #
64 # XGE_COMPONENT_HAL_CONFIG 0x1
65 # XGE_COMPONENT_HAL_FIFO 0x2
66 # XGE_COMPONENT_HAL_RING 0x4
67 # XGE_COMPONENT_HAL_CHANNEL 0x8
68 # XGE_COMPONENT_HAL_DEVICE 0x10
69 # XGE_COMPONENT_HAL_MM 0x20
70 # XGE_COMPONENT_HAL_QUEUE 0x40
71 # XGE_COMPONENT_HAL_STATS 0x100
72 # XGE_COMPONENT_OSDEP 0x1000
73 # XGE_COMPONENT_LL 0x2000
74 # XGE_COMPONENT_TOE 0x4000
75 # XGE_COMPONENT_RDMA 0x8000
76 # XGE_COMPONENT_ALL 0xffffffff
77 #TRACE_CFLAGS = -DXGE_DEBUG_MODULE_MASK=0xffffffff \
78 # -DXGE_DEBUG_TRACE_MASK=0xffffffff \
79 # -DXGE_DEBUG_ERR_MASK=0xffffffff
80 TRACE_CFLAGS = -DXGE_DEBUG_MODULE_MASK=0x00003010 \
81 # -DXGE_DEBUG_TRACE_MASK=0x00000000 \
82 # -DXGE_DEBUG_ERR_MASK=0x00003010
83 #
84 XGE_CFLAGS = $(HAL_CFLAGS) $(TRACE_CFLAGS) $(CCVERBOSE) \
85 # -I$(UTSBASE)/common/io/xge/hal/include \
86 # -I$(UTSBASE)/common/io/xge/hal/xgehal \
87 # -I$(UTSBASE)/common/io/xge/drv -DSOLARIS
88 #
89 COPTFLAG = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
90 COPTFLAG64 = $(ADJUST_OPT2) -_cc=-x04 -_gcc=-O2
91 CFLAGS += $(XGE_CFLAGS) -xcrossfile -xc99=%all
92 CFLAGS64 += $(XGE_CFLAGS) -xcrossfile -xc99=%all
93 #
94 #
95 # Driver depends on MAC & IP
96 #
97 LDFLAGS += -dy -N misc/mac -N drv/ip
98 #
99 # Lint flag
100 #
101 LINTFLAGS += $(XGE_CFLAGS) -xc99=%all
102 #
103 #
104 # For now, disable these lint checks; maintainers should endeavor
105 # to investigate and remove these for maximum lint coverage.
106 # Please do not carry these forward to new Makefiles.
107 #
108 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
109 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
110 LINTTAGS += -erroff=E_STATIC_UNUSED
111 #
112 CERRWARN += -_gcc=-Wno-parentheses
113 CERRWARN += -_gcc=-Wno-unused-variable
114 CERRWARN += -_gcc=-Wno-unused-label
115 CERRWARN += -_gcc=-Wno-empty-body
116 CERRWARN += -_gcc=-Wno-uninitialized
117 #
118 #
119 #
120 # Default build targets.
121 #
122 .KEEP_STATE:
123 #
124 def: $(DEF_DEPS)

```

**new/usr/src/uts/sparc/xge/Makefile**

3

```
126 all:          $(ALL_DEPS)
128 clean:        $(CLEAN_DEPS)
130 clobber:      $(CLOBBER_DEPS)
132 lint:         $(LINT_DEPS)
134 modlintlib:   $(MODLINTLIB_DEPS)
136 clean.lint:   $(CLEAN_LINT_DEPS)
138 install:      $(INSTALL_DEPS)

140 #
141 #   Include common targets.
142 #
143 include $(UTSBASE)/sparc/Makefile.targ
```

```

*****
13447 Tue Oct 7 15:48:36 2014
new/usr/src/uts/sun4u/Makefile.sun4u
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 #
23 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright (c) 2013 Andrew Stormont. All rights reserved.
26 # Copyright 2014 Gary Mills
27 #
28 # This makefile contains the common definitions for the sun4u unix
29 # and all sun4u implementation architecture dependent modules.
30 #

32 #
33 # Machine type (implementation architecture):
34 #
35 PLATFORM = sun4u
36 PROMIF = ieee1275
37 PSMBASE = $(UTSBASE)/../psm

39 #
40 # uname -m value
41 #
42 UNAME_M = $(PLATFORM)

44 #
45 # Definitions for the platform-specific /platform directories.
46 #
47 # PLATFORMS designates those sun4u machines which have no platform
48 # specific code.
49 #
50 # IMPLEMENTATIONS is used to designate sun4u machines which do have
51 # platform specific modules (perhaps including their own unix). All
52 # code specific to a given implementation resides in the appropriately
53 # named subdirectory. This requires these platforms to have their
54 # own Makefiles to define ROOT_PLAT_DIRS, USR_PLAT_DIRS, etc.
55 #
56 # So if we had an implementation named 'foo', we would need the following
57 # Makefiles in the foo subdirectory:
58 #
59 # sun4u/foo/Makefile
60 # sun4u/foo/Makefile.foo
61 # sun4u/foo/Makefile.targ

```

```

62 #
63 #
64 #
65 # /usr/platform/$(IMPLEMENTED_PLATFORM) is created as a directory that
66 # all the $(LINKED_PLATFORMS) link to.
67 #
68 IMPLEMENTED_PLATFORM = SUNW,Ultra-2

70 LINKED_PLATFORMS += SUNW,Ultra-30
71 LINKED_PLATFORMS += SUNW,Ultra-60

73 #
74 # all PLATFORMS that do not belong in the $(IMPLEMENTATIONS) list
75 # ie. all desktop platforms
76 #
77 PLATFORMS = $(IMPLEMENTED_PLATFORM)
78 PLATFORMS += $(LINKED_PLATFORMS)

80 ROOT_PLAT_DIRS = $(PLATFORMS:%=$(ROOT_PLAT_DIR)/%)
81 USR_PLAT_DIRS = $(PLATFORMS:%=$(USR_PLAT_DIR)/%)

83 USR_DESKTOP_DIR = $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM)
84 USR_DESKTOP_INC_DIR = $(USR_DESKTOP_DIR)/include
85 USR_DESKTOP_SBIN_DIR = $(USR_DESKTOP_DIR)/sbin
86 USR_DESKTOP_LIB_DIR = $(USR_DESKTOP_DIR)/lib

88 #
89 # Welcome to SPARC V9.
90 #

92 #
93 # Define supported builds
94 #
95 DEF_BUILDS = $(DEF_BUILDS64)
96 ALL_BUILDS = $(ALL_BUILDS64)

98 #
99 # Everybody needs to know how to build modstubs.o and to locate unix.o
100 #
101 UNIX_DIR = $(UTSBASE)/$(PLATFORM)/unix
102 GENLIB_DIR = $(UTSBASE)/$(PLATFORM)/genunix
103 MODSTUBS_DIR = $(UNIX_DIR)
104 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
105 LINTS_DIR = $(OBJSDIR)
106 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJSDIR)
107
108 DTRACESTUBS_O = $(OBJSDIR)/dtracestubs.o
109 DTRACESTUBS = $(OBJSDIR)/libdtracestubs.so

111 UNIX_O = $(UNIX_DIR)/$(OBJSDIR)/unix.o
112 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJSDIR)/modstubs.o
113 GENLIB = $(GENLIB_DIR)/$(OBJSDIR)/libgenunix.so

115 LINT_LIB = $(LINT_LIB_DIR)/llib-lunix.ln
116 GEN_LINT_LIB = $(LINT_LIB_DIR)/llib-lgenunix.ln

118 LINT64_DIRS = $(LINT64_BUILDS:%=$(UTSBASE)/$(PLATFORM)/lint-libs/%)
119 LINT64_FILES = $(LINT64_DIRS:%=%/llib-l$(MODULE).ln)

121 #
122 # cpu and platform modules need to know how to build their own symcheck mo
123 #
124 PLATMOD = platmod
125 PLATLIB = $(PLATDIR)/$(OBJSDIR)/libplatmod.so

127 CPUNAME = cpu

```



```

128 CPULIB          = $(CPU_DIR)/$(OBJS_DIR)/libcpu.so
130 SYM_MOD         = $(OBJS_DIR)/unix.sym

132 #
133 #       Include the makefiles which define build rule templates, the
134 #       collection of files per module, and a few specific flags. Note
135 #       that order is significant, just as with an include path. The
136 #       first build rule template which matches the files name will be
137 #       used. By including these in order from most machine dependent
138 #       to most machine independent, we allow a machine dependent file
139 #       to be used in preference over a machine independent version
140 #       (Such as a machine specific optimization, which preserves the
141 #       interfaces.)
142 #
143 include $(UTSBASE)/sun4/Makefile.files
144 include $(UTSBASE)/$(PLATFORM)/Makefile.files
145 include $(UTSBASE)/sfmmu/Makefile.files
146 include $(UTSBASE)/sparc/v9/Makefile.files
147 include $(UTSBASE)/sparc/Makefile.files
148 include $(UTSBASE)/sun/Makefile.files
149 include $(SRC)/psm/promif/$(PROMIF)/common/Makefile.files
150 include $(SRC)/psm/promif/$(PROMIF)/$(PLATFORM)/Makefile.files
151 include $(UTSBASE)/common/Makefile.files

153 #
154 #       Include machine independent rules. Note that this does not imply
155 #       that the resulting module from rules in Makefile.uts is machine
156 #       independent. Only that the build rules are machine independent.
157 #
158 include $(UTSBASE)/Makefile.uts

160 # These come after Makefile.uts.
161 IMPLEMENTATIONS  = tazmo
162 IMPLEMENTATIONS += starfire
163 IMPLEMENTATIONS += javelin
164 IMPLEMENTATIONS += darwin
165 IMPLEMENTATIONS += quasar
166 IMPLEMENTATIONS += grover
167 IMPLEMENTATIONS += enchilada
168 IMPLEMENTATIONS += taco
169 IMPLEMENTATIONS += mpxu
170 IMPLEMENTATIONS += excalibur
171 IMPLEMENTATIONS += montecarlo
172 IMPLEMENTATIONS += serengeti
173 IMPLEMENTATIONS += littleneck
174 IMPLEMENTATIONS += starcat
175 IMPLEMENTATIONS += daktari
176 IMPLEMENTATIONS += cherrystone
177 IMPLEMENTATIONS += fjlite
178 IMPLEMENTATIONS += snowbird
179 IMPLEMENTATIONS += schumacher
180 IMPLEMENTATIONS += blade
181 IMPLEMENTATIONS += boston
182 IMPLEMENTATIONS += seattle
183 IMPLEMENTATIONS += chicago
184 IMPLEMENTATIONS += sunfire
185 IMPLEMENTATIONS += lw8
186 IMPLEMENTATIONS += makaha
187 IMPLEMENTATIONS += opl
188 IMPLEMENTATIONS += lw2plus

190 #
191 #       machine specific optimization, override default in Makefile.master
192 #
193 CC_XARCH          = -m64 -xarch=sparcvis

```

```

194 AS_XARCH         = -xarch=v9a
195 COPTIMIZE        = -_cc=-xO3 -_gcc=-O2
196 COPTIMIZE        = -xO3
197 CCMODE           = -Xa

198 CFLAGS           = -xchip=ultra $(CCABS32) $(CCREGSYM)
199 CFLAGS           += $(CC_XARCH)
200 CFLAGS           += $(COPTIMIZE)
201 CFLAGS           += $(EXTRA_CFLAGS)
202 CFLAGS           += $(XAOPT)
203 CFLAGS           += $(INLINES) -D_ASM_INLINES
204 CFLAGS           += $(CCMODE)
205 CFLAGS           += $(SPACEFLAG)
206 CFLAGS           += $(CERRWARN)
207 CFLAGS           += $(CTF_FLAGS_$(CLASS))
208 CFLAGS           += $(C99MODE)
209 CFLAGS           += $(CCUNBOUND)
210 CFLAGS           += $(CCNOAUTOINLINE)
211 CFLAGS           += $(CCSTATICSYM)
212 CFLAGS           += $(CC32BITCALLERS)
213 CFLAGS           += $(IROPTFLAG)
214 CFLAGS           += $(CGLOBALSTATIC)
215 CFLAGS           += -xregs=no%float
216 CFLAGS           += -xstrconst
217 CFLAGS           += $(CSOURCEDEBUGFLAGS)
218 CFLAGS           += $(CUSERFLAGS)

220 ASFLAGS          += $(AS_XARCH)

222 AS_INC_PATH      += -I$(DSF_DIR)/$(OBJS_DIR)

224 LINT_KMODS       += $(GENUNIX_KMODS)

226 LINT_DEFS        = -m64

228 #
229 #       The following must be defined for all implementations:
230 #
231 #       MAPFILE:          ld mapfile for the build of kernel/unix.
232 #       MODSTUBS:        Module stubs source file.
233 #       GENCONST_SRC:    genconst.c
234 #       OFFSETS:         offsets.in
235 #       PLATFORM_OFFSETS: Platform specific mach_offsets.in
236 #       FDOFFSETS:       fd_offsets.in
237 #
238 MAPFILE          = $(UTSBASE)/sun4/conf/Mapfile
239 MODSTUBS         = $(UTSBASE)/sparc/ml/modstubs.s
240 GENCONST_SRC     = $(UTSBASE)/sun4/ml/genconst.c
241 OFFSETS         = $(UTSBASE)/sun4/ml/offsets.in
242 PLATFORM_OFFSETS = $(UTSBASE)/sun4u/ml/mach_offsets.in
243 FDOFFSETS        = $(UTSBASE)/sun/io/fd_offsets.in

245 #
246 #       Define the actual specific platforms
247 #

249 MACHINE_DEFS    = -D$(PLATFORM) -D_MACHDEP -DSFMMU

251 #
252 #       Software workarounds for hardware "features"
253 #

255 include $(UTSBASE)/$(PLATFORM)/Makefile.workarounds

257 #
258 #       Debugging level

```

```

259 #
260 #     Special knowledge of which special debugging options effect which
261 #     file is used to optimize the build if these flags are changed.
262 #
263 #     XXX: The above could possibly be done for more flags and files, but
264 #     is left as an experiment to the interested reader. Be forewarned,
265 #     that excessive use could lead to maintenance difficulties.
266 #
267 #     Note: kslice can be enabled for the sun4u, but is disabled by default
268 #     in all cases.
269 #
271 DEBUG_DEFS_OBJ64      =
272 DEBUG_DEFS_DBG64      = -DDEBUG
273 DEBUG_DEFS             = $(DEBUG_DEFS_$(BUILD_TYPE))
274
275 DEBUG_COND_OBJ64      = $(POUND_SIGN)
276 DEBUG_COND_DBG64      =
277 IF_DEBUG_OBJ          = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/
278
279 $(IF_DEBUG_OBJ)trap.o      :=      DEBUG_DEFS      += -DTRAPDEBUG
280 $(IF_DEBUG_OBJ)mach_trap.o :=      DEBUG_DEFS      += -DTRAPDEBUG
281 $(IF_DEBUG_OBJ)syscall_trap.o :=    DEBUG_DEFS      += -DSYSCALLTRACE
282 $(IF_DEBUG_OBJ)clock.o    :=      DEBUG_DEFS      += -DKSLICE=0
283
284 IF_TRAPTRACE_OBJ = $(IF_DEBUG_OBJ)
285 # comment this out for a non-debug kernel with TRAPTRACE
286 #IF_TRAPTRACE_OBJ = $(OBJS_DIR)/
287
288 $(IF_TRAPTRACE_OBJ)mach_locore.o :=      DEBUG_DEFS      += -DTRAPTRACE
289 $(IF_TRAPTRACE_OBJ)m1setup.o     :=      DEBUG_DEFS      += -DTRAPTRACE
290 $(IF_TRAPTRACE_OBJ)syscall_trap.o :=      DEBUG_DEFS      += -DTRAPTRACE
291 $(IF_TRAPTRACE_OBJ)startup.o     :=      DEBUG_DEFS      += -DTRAPTRACE
292 $(IF_TRAPTRACE_OBJ)mach_startup.o :=      DEBUG_DEFS      += -DTRAPTRACE
293 $(IF_TRAPTRACE_OBJ)mp_startup.o  :=      DEBUG_DEFS      += -DTRAPTRACE
294 $(IF_TRAPTRACE_OBJ)cpu_states.o  :=      DEBUG_DEFS      += -DTRAPTRACE
295 $(IF_TRAPTRACE_OBJ)mach_cpu_states.o :=      DEBUG_DEFS      += -DTRAPTRACE
296 $(IF_TRAPTRACE_OBJ)interrupt.o   :=      DEBUG_DEFS      += -DTRAPTRACE
297 $(IF_TRAPTRACE_OBJ)mach_interrupt.o :=      DEBUG_DEFS      += -DTRAPTRACE
298 $(IF_TRAPTRACE_OBJ)sfmmu_asm.o   :=      DEBUG_DEFS      += -DTRAPTRACE
299 $(IF_TRAPTRACE_OBJ)trap_table.o  :=      DEBUG_DEFS      += -DTRAPTRACE
300 $(IF_TRAPTRACE_OBJ)xc.o          :=      DEBUG_DEFS      += -DTRAPTRACE
301 $(IF_TRAPTRACE_OBJ)mach_xc.o     :=      DEBUG_DEFS      += -DTRAPTRACE
302 $(IF_TRAPTRACE_OBJ)wbuf.o        :=      DEBUG_DEFS      += -DTRAPTRACE
303 $(IF_TRAPTRACE_OBJ)trap.o        :=      DEBUG_DEFS      += -DTRAPTRACE
304 $(IF_TRAPTRACE_OBJ)mach_trap.o   :=      DEBUG_DEFS      += -DTRAPTRACE
305 $(IF_TRAPTRACE_OBJ)x_call.o      :=      DEBUG_DEFS      += -DTRAPTRACE
306 $(IF_TRAPTRACE_OBJ)spitfire_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
307 $(IF_TRAPTRACE_OBJ)us3_common_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
308 $(IF_TRAPTRACE_OBJ)us3_cheetah_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
309 $(IF_TRAPTRACE_OBJ)us3_cheetahplus_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
310 $(IF_TRAPTRACE_OBJ)us3_jalapeno_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
311 $(IF_TRAPTRACE_OBJ)opl_olympus_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
312
313 # Comment these out if you don't want dispatcher lock statistics.
314
315 #$(IF_DEBUG_OBJ)lock_prim.o      := DEBUG_DEFS      += -DDISP_LOCK_STATS
316 #$(IF_DEBUG_OBJ)disp.o          := DEBUG_DEFS      += -DDISP_LOCK_STATS
317
318 # Comment these out if you don't want dispatcher debugging
319
320 #$(IF_DEBUG_OBJ)lock_prim.o      := DEBUG_DEFS      += -DDISP_DEBUG
321
322 #
323 #     Collect the preprocessor definitions to be associated with *all*
324 #     files.

```

```

325 #
326 ALL_DEFS              = $(MACHINE_DEFS) $(WORKAROUND_DEFS) $(DEBUG_DEFS) \
327                        $(OPTION_DEFS)
328 GENCONST_DEFS        = $(MACHINE_DEFS) $(OPTION_DEFS)
329
330 #
331 # ----- TRANSITIONAL SECTION -----
332 #
333
334 #
335 #     Not everything which *should* be a module is a module yet. The
336 #     following is a list of such objects which are currently part of
337 #     the base kernel but should soon become kmods.
338 #
339 MACH_NOT_YET_KMODS    = $(AUTOCONF_OBJS)
340
341 #
342 # ----- END OF TRANSITIONAL SECTION -----
343 #
344
345 #
346 #     The kernels modules which are "implementation architecture"
347 #     specific for this machine are enumerated below. Note that most
348 #     of these modules must exist (in one form or another) for each
349 #     architecture.
350 #
351 #     Common Drivers (usually pseudo drivers) (/kernel/drv):
352 #
353
354 #
355 #     Machine Specific Driver Modules (/kernel/drv):
356 #
357 #     XXX: How many of these are really machine specific?
358 #
359 DRV_KMODS             += bbc_beeper
360 DRV_KMODS             += cpc
361 DRV_KMODS             += fd
362 DRV_KMODS             += rootnex sbusmem upa64s zs zsh
363 DRV_KMODS             += sbus
364 DRV_KMODS             += pcisch pcipsy simba
365 DRV_KMODS             += px
366 DRV_KMODS             += ebus
367 DRV_KMODS             += su
368 DRV_KMODS             += tod
369 DRV_KMODS             += power
370 DRV_KMODS             += epic
371 DRV_KMODS             += grbeep
372 DRV_KMODS             += pcf8584 max1617 seeprom tda8444 pca9556
373 DRV_KMODS             += ics951601 adm1031
374 DRV_KMODS             += lm75 ltc1427 pcf8591 pcf8574 ssc050 ssc100
375 DRV_KMODS             += pic16f819
376 DRV_KMODS             += pic16f747
377 DRV_KMODS             += adm1026
378 DRV_KMODS             += us
379 DRV_KMODS             += ppm schppm jbusppm
380 DRV_KMODS             += mc-us3
381 DRV_KMODS             += mc-us3i
382 DRV_KMODS             += smbus
383 DRV_KMODS             += db21554
384 DRV_KMODS             += gpio_87317
385 DRV_KMODS             += isadma
386 DRV_KMODS             += sbbc
387 DRV_KMODS             += pmubus
388 DRV_KMODS             += pmugpio
389 DRV_KMODS             += pmc
390 DRV_KMODS             += trapstat

```

## new/usr/src/uts/sun4u/Makefile.sun4u

7

```

391 DRV_KMODS      += rmc_comm
392 DRV_KMODS      += rmcadm
393 DRV_KMODS      += rmcclmv
394 DRV_KMODS      += sf
395 DRV_KMODS      += nxge
396 DRV_KMODS      += i2bsc
397 DRV_KMODS      += mem_cache

399 #
400 #      Exec Class Modules (/kernel/exec):
401 #
402 EXEC_KMODS      +=

404 #
405 #      Scheduling Class Modules (/kernel/sched):
406 #
407 SCHED_KMODS    +=

409 #
410 #      File System Modules (/kernel/fs):
411 #
412 FS_KMODS       +=

414 #
415 #      Streams Modules (/kernel/strmod):
416 #
417 STRMOD_KMODS   += kb

419 #
420 #      'System' Modules (/kernel/sys):
421 #
422 SYS_KMODS      +=

424 #
425 #      'User' Modules (/kernel/misc):
426 #
427 MISC_KMODS     += bignum
428 MISC_KMODS     += obpsym bootdev vis cpr platmod md5 sha1 i2c_svc
429 MISC_KMODS     += sbd

431 MISC_KMODS     += opl_cfg
432 MISC_KMODS     += zulumv
433 MISC_KMODS     += gptwo_cpu gptwocfg
434 MISC_KMODS     += pcie

436 #
437 #      Brand modules
438 #
439 BRAND_KMODS    += snl_brand s10_brand

441 #
442 #      Software Cryptographic Providers (/kernel/crypto):
443 #
444 CRYPTO_KMODS   += aes
445 CRYPTO_KMODS   += arcfour
446 CRYPTO_KMODS   += des

448 #
449 #      generic-unix module (/kernel/genunix):
450 #
451 GENUNIX_KMODS  += genunix

453 #      'User' "Modules" excluded from the Full Kernel lint target:
454 #

456 #

```

## new/usr/src/uts/sun4u/Makefile.sun4u

8

```

457 #      Modules eXcluded from the product:
458 #
459 XMODS          +=

461 #
462 #      cpu modules
463 #
464 CPU_KMODS      += cheetah cheetahplus jalapeno serrano spitfire hummingbird

466 #
467 #      sun4u 'TOD' Modules (/platform/.../kernel/tod):
468 #
469 TOD_KMODS      += todods1287 todods1337 todmostek todstarfire
470 TOD_KMODS      += todm5819 todblade todhq4802 todsg todopl
471 TOD_KMODS      += todm5819p_rmc todstarcat

473 #
474 #      Performance Counter BackEnd Modules (/usr/kernel/pcbe):
475 #
476 PCBE_KMODS     += us234_pcbe
477 PCBE_KMODS     += opl_pcbe

```

```

*****
12164 Tue Oct 7 15:48:37 2014
new/usr/src/uts/sun4v/Makefile.sun4v
5196 The cw wrapper restricts gcc to -O2
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright 2010 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright (c) 2013 Andrew Stormont. All rights reserved.
26 # Copyright 2014 Gary Mills
27 #
28 # This makefile contains the common definitions for the sun4v unix
29 # and all sun4v implementation architecture dependent modules.
30 #
31 #
32 #
33 # Machine type (implementation architecture):
34 #
35 PLATFORM = sun4v
36 LINKED_PLATFORMS += SUNW,Sun-Fire-T1000
37 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T5120
38 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T5220
39 LINKED_PLATFORMS += SUNW,T5140
40 LINKED_PLATFORMS += SUNW,T5240
41 LINKED_PLATFORMS += SUNW,T5440
42 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T1000
43 LINKED_PLATFORMS += SUNW,Sun-Blade-T6300
44 LINKED_PLATFORMS += SUNW,Sun-Blade-T6320
45 LINKED_PLATFORMS += SUNW,Netra-CP3260
46 LINKED_PLATFORMS += SUNW,Netra-T5220
47 LINKED_PLATFORMS += SUNW,USBRDT-5240
48 LINKED_PLATFORMS += SUNW,Netra-T5440
49 LINKED_PLATFORMS += SUNW,Sun-Blade-T6340
50 PROMIF = ieee1275
51 PSMBASE = $(UTSBASE)/../psm
52 #
53 #
54 # uname -m value
55 #
56 UNAME_M = $(PLATFORM)
57 #
58 #
59 # Definitions for the platform-specific /platform directories.
60 #
61 # PLATFORMS designates those sun4v machines which have no platform

```

```

62 # specific code.
63 #
64 # IMPLEMENTATIONS is used to designate sun4v machines which have
65 # platform specific modules. All code specific to a given implementation
66 # resides in the appropriately named subdirectory. This requires
67 # these platforms to have their own Makefiles to define ROOT_PLAT_DIRS,
68 # USR_PLAT_DIRS, etc.
69 # The number of IMPLEMENTATIONS should not grow!
70 #
71 # So if we had an implementation named 'foo', we would need the following
72 # Makefiles in the foo subdirectory:
73 #
74 # sun4v/foo/Makefile
75 # sun4v/foo/Makefile.foo
76 # sun4v/foo/Makefile.targ
77 #
78 #
79 #
80 # all PLATFORMS that do not belong in the $(IMPLEMENTATIONS) list.
81 # This list should be empty. A platform without platform modules
82 # is a plain, generic sun4v platform.
83 #
84 #IMPLEMENTED_PLATFORM =
85 #PLATFORMS = $(IMPLEMENTED_PLATFORM)
86 #
87 IMPLEMENTATIONS = ontario montoya huron maramba
88 #
89 #ROOT_PLAT_DIRS = $(PLATFORMS:%=$(ROOT_PLAT_DIR)/%)
90 #USR_PLAT_DIRS = $(PLATFORMS:%=$(USR_PLAT_DIR)/%)
91 #
92 #USR_DESKTOP_DIR = $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM)
93 #USR_DESKTOP_INC_DIR = $(USR_DESKTOP_DIR)/include
94 #USR_DESKTOP_SBIN_DIR = $(USR_DESKTOP_DIR)/sbin
95 #USR_DESKTOP_LIB_DIR = $(USR_DESKTOP_DIR)/lib
96 #
97 #
98 # Define supported builds
99 #
100 DEF_BUILDS = $(DEF_BUILDS64)
101 ALL_BUILDS = $(ALL_BUILDS64)
102 #
103 #
104 # Everybody needs to know how to build modstubs.o and to locate unix.o
105 #
106 UNIX_DIR = $(UTSBASE)/$(PLATFORM)/unix
107 GENLIB_DIR = $(UTSBASE)/$(PLATFORM)/genunix
108 MODSTUBS_DIR = $(UNIX_DIR)
109 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
110 LINTS_DIR = $(OBJS_DIR)
111 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJS_DIR)
112 #
113 DTRACESTUBS_O = $(OBJS_DIR)/dtracestubs.o
114 DTRACESTUBS = $(OBJS_DIR)/libdtracestubs.so
115 #
116 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
117 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
118 GENLIB = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
119 #
120 LINT_LIB = $(LINT_LIB_DIR)/llib-lunix.ln
121 GEN_LINT_LIB = $(LINT_LIB_DIR)/llib-lgenunix.ln
122 #
123 LINT64_DIRS = $(LINT64_BUILDS:%=$(UTSBASE)/$(PLATFORM)/lint-libs/%)
124 LINT64_FILES = $(LINT64_DIRS:%=%/llib-l$(MODULE).ln)
125 #
126 #
127 # cpu and platform modules need to know how to build their own symcheck mo

```

```

128 #
129 PLATMOD          = platmod
130 PLATLIB          = $(PLAT_DIR)/$(OBJS_DIR)/libplatmod.so

132 CPUNAME         = cpu
133 CPULIB          = $(CPU_DIR)/$(OBJS_DIR)/libcpu.so

135 SYM_MOD         = $(OBJS_DIR)/unix.sym

137 #
138 #   Include the makefiles which define build rule templates, the
139 #   collection of files per module, and a few specific flags. Note
140 #   that order is significant, just as with an include path. The
141 #   first build rule template which matches the files name will be
142 #   used. By including these in order from most machine dependent
143 #   to most machine independent, we allow a machine dependent file
144 #   to be used in preference over a machine independent version
145 #   (Such as a machine specific optimization, which preserves the
146 #   interfaces.)
147 #
148 include $(UTSBASE)/sun4/Makefile.files
149 include $(UTSBASE)/$(PLATFORM)/Makefile.files
150 include $(UTSBASE)/sfmmu/Makefile.files
151 include $(UTSBASE)/sparc/v9/Makefile.files
152 include $(UTSBASE)/sparc/Makefile.files
153 include $(UTSBASE)/sun/Makefile.files
154 include $(SRC)/psm/promif/$(PROMIF)/common/Makefile.files
155 include $(SRC)/psm/promif/$(PROMIF)/$(PLATFORM)/Makefile.files
156 include $(UTSBASE)/common/Makefile.files

158 #
159 #   Include machine independent rules. Note that this does not imply
160 #   that the resulting module from rules in Makefile.uts is machine
161 #   independent. Only that the build rules are machine independent.
162 #
163 include $(UTSBASE)/Makefile.uts

165 CTFMERGE_GUDIR = sun4v

167 #
168 #   machine specific optimization, override default in Makefile.master
169 #
170 CC_XARCH        = -m64 -xarch=sparcvis
171 AS_XARCH        = -xarch=v9v
172 COPTIMIZE       = -_cc=-xO3 -_gcc=-O2
171 COPTIMIZE       = -xO3
173 CCMODE          = -Xa

175 CFLAGS          = -xchip=ultra $(CCABS32) $(CCREGSYM)
176 CFLAGS          += $(CC_XARCH)
177 CFLAGS          += $(COPTIMIZE)
178 CFLAGS          += $(EXTRA_CFLAGS)
179 CFLAGS          += $(XAOPT)
180 CFLAGS          += $(INLINES) -D_ASM_INLINES
181 CFLAGS          += $(CCMODE)
182 CFLAGS          += $(SPACEFLAG)
183 CFLAGS          += $(CERRWARN)
184 CFLAGS          += $(CTF_FLAGS_$(CLASS))
185 CFLAGS          += $(C99MODE)
186 CFLAGS          += $(CCUNBOUND)
187 CFLAGS          += $(CCNOAUTOINLINE)
188 CFLAGS          += $(CCSTATCSYM)
189 CFLAGS          += $(CC32BITCALLERS)
190 CFLAGS          += $(IROPTFLAG)
191 CFLAGS          += $(CGLOBALSTATIC)
192 CFLAGS          += -xregs=no%float

```

```

193 CFLAGS          += -xstrconst
194 CFLAGS          += $(CSOURCEDEBUGFLAGS)
195 CFLAGS          += $(USERFLAGS)

197 CPPFLAGS        += -DGLREG

199 ASFLAGS         += $(AS_XARCH) -DGLREG

201 AS_INC_PATH     += -I$(DSF_DIR)/$(OBJS_DIR)

203 LINT_KMODS      += $(GENUNIX_KMODS)

205 LINT_DEFS       = -m64

207 #
208 #   The following must be defined for all implementations:
209 #
210 #   MAPFILE:          ld mapfile for the build of kernel/unix.
211 #   MODSTUBS:        Module stubs source file.
212 #   GENCONST_SRC:    genconst.c
213 #   OFFSETS:         offsets.in
214 #   PLATFORM_OFFSETS: Platform specific mach_offsets.in
215 #   FDOFFSETS:       fd_offsets.in
216 #
217 MAPFILE         = $(UTSBASE)/sun4/conf/Mapfile
218 MODSTUBS        = $(UTSBASE)/sparc/ml/modstubs.s
219 GENCONST_SRC    = $(UTSBASE)/sun4/ml/genconst.c
220 OFFSETS         = $(UTSBASE)/sun4/ml/offsets.in
221 PLATFORM_OFFSETS = $(UTSBASE)/sun4v/ml/mach_offsets.in
222 FDOFFSETS       = $(UTSBASE)/sun/io/fd_offsets.in

224 #
225 #   Define the actual specific platforms
226 #

228 MACHINE_DEFS    = -D$(PLATFORM) -D_MACHDEP -DSFMMU
229 MACHINE_DEFS    += -DMAX_MEM_NODES=8

231 #
232 #   Software workarounds for hardware "features"
233 #

235 include $(UTSBASE)/$(PLATFORM)/Makefile.workarounds

237 #
238 #   Debugging level
239 #
240 #   Special knowledge of which special debugging options effect which
241 #   file is used to optimize the build if these flags are changed.
242 #
243 #   XXX: The above could possibly be done for more flags and files, but
244 #   is left as an experiment to the interested reader. Be forewarned,
245 #   that excessive use could lead to maintenance difficulties.
246 #
247 #   Note: kslicc can be enabled for the sun4v, but is disabled by default
248 #   in all cases.
249 #

251 DEBUG_DEFS_OBJ64 =
252 DEBUG_DEFS_DBG64 = -DDEBUG
253 DEBUG_DEFS        = $(DEBUG_DEFS_$(BUILD_TYPE))

255 DEBUG_COND_OBJ64 = $(POUND_SIGN)
256 DEBUG_COND_DBG64 =
257 IF_DEBUG_OBJ      = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

```

```

259 $(IF_DEBUG_OBJ)trap.o := DEBUG_DEFS += -DTRAPDEBUG
260 $(IF_DEBUG_OBJ)mach_trap.o := DEBUG_DEFS += -DTRAPDEBUG
261 $(IF_DEBUG_OBJ)syscall_trap.o := DEBUG_DEFS += -DSYSCALLTRACE
262 $(IF_DEBUG_OBJ)clock.o := DEBUG_DEFS += -DKSLICE=0

264 IF_TRAPTRACE_OBJ = $(IF_DEBUG_OBJ)
265 # comment this out for a non-debug kernel with TRAPTRACE
266 #IF_TRAPTRACE_OBJ = $(OBJS_DIR)/

268 $(IF_TRAPTRACE_OBJ)mach_locore.o := DEBUG_DEFS += -DTRAPTRACE
269 $(IF_TRAPTRACE_OBJ)mlsetup.o := DEBUG_DEFS += -DTRAPTRACE
270 $(IF_TRAPTRACE_OBJ)syscall_trap.o := DEBUG_DEFS += -DTRAPTRACE
271 $(IF_TRAPTRACE_OBJ)startup.o := DEBUG_DEFS += -DTRAPTRACE
272 $(IF_TRAPTRACE_OBJ)mach_startup.o := DEBUG_DEFS += -DTRAPTRACE
273 $(IF_TRAPTRACE_OBJ)mp_startup.o := DEBUG_DEFS += -DTRAPTRACE
274 $(IF_TRAPTRACE_OBJ)cpu_states.o := DEBUG_DEFS += -DTRAPTRACE
275 $(IF_TRAPTRACE_OBJ)mach_cpu_states.o := DEBUG_DEFS += -DTRAPTRACE
276 $(IF_TRAPTRACE_OBJ)interrupt.o := DEBUG_DEFS += -DTRAPTRACE
277 $(IF_TRAPTRACE_OBJ)mach_interrupt.o := DEBUG_DEFS += -DTRAPTRACE
278 $(IF_TRAPTRACE_OBJ)sfmmu_asm.o := DEBUG_DEFS += -DTRAPTRACE
279 $(IF_TRAPTRACE_OBJ)trap_table.o := DEBUG_DEFS += -DTRAPTRACE
280 $(IF_TRAPTRACE_OBJ)xc.o := DEBUG_DEFS += -DTRAPTRACE
281 $(IF_TRAPTRACE_OBJ)mach_xc.o := DEBUG_DEFS += -DTRAPTRACE
282 $(IF_TRAPTRACE_OBJ)wbuf.o := DEBUG_DEFS += -DTRAPTRACE
283 $(IF_TRAPTRACE_OBJ)trap.o := DEBUG_DEFS += -DTRAPTRACE
284 $(IF_TRAPTRACE_OBJ)mach_trap.o := DEBUG_DEFS += -DTRAPTRACE
285 $(IF_TRAPTRACE_OBJ)x_call.o := DEBUG_DEFS += -DTRAPTRACE

287 # Comment these out if you don't want dispatcher lock statistics.

289 #$(IF_DEBUG_OBJ)lock_prim.o := DEBUG_DEFS += -DDISP_LOCK_STATS
290 #$(IF_DEBUG_OBJ)disp.o := DEBUG_DEFS += -DDISP_LOCK_STATS

292 # Comment these out if you don't want dispatcher debugging

294 #$(IF_DEBUG_OBJ)lock_prim.o := DEBUG_DEFS += -DDISP_DEBUG

296 #
297 # Collect the preprocessor definitions to be associated with *all*
298 # files.
299 #
300 ALL_DEFS = $(MACHINE_DEFS) $(WORKAROUND_DEFS) $(DEBUG_DEFS) \
301 $(OPTION_DEFS)
302 GENCONST_DEFS = $(MACHINE_DEFS) $(OPTION_DEFS)

304 #
305 # ----- TRANSITIONAL SECTION -----
306 #

308 #
309 # Not everything which *should* be a module is a module yet. The
310 # following is a list of such objects which are currently part of
311 # the base kernel but should soon become kmods.
312 #
313 MACH_NOT_YET_KMODS = $(AUTOCONF_OBJS)

315 #
316 # ----- END OF TRANSITIONAL SECTION -----
317 #

319 #
320 # The kernels modules which are "implementation architecture"
321 # specific for this machine are enumerated below. Note that most
322 # of these modules must exist (in one form or another) for each
323 # architecture.
324 #

```

```

325 # Common Drivers (usually pseudo drivers) (/kernel/drv):
326 #

328 #
329 # Machine Specific Driver Modules (/kernel/drv):
330 #
331 DRV_KMODS += bge
332 DRV_KMODS += cnex
333 DRV_KMODS += cpc
334 DRV_KMODS += drctl
335 DRV_KMODS += ds_pri
336 DRV_KMODS += ds_snmp
337 DRV_KMODS += ebus
338 DRV_KMODS += fpc
339 DRV_KMODS += glvc
340 DRV_KMODS += mdesc
341 DRV_KMODS += niumx
342 DRV_KMODS += ntwdt
343 DRV_KMODS += nxge
344 DRV_KMODS += n2piupc
345 DRV_KMODS += iospc
346 DRV_KMODS += n2rng
347 DRV_KMODS += px
348 DRV_KMODS += qcn
349 DRV_KMODS += rootnex
350 DRV_KMODS += su
351 DRV_KMODS += tpm
352 DRV_KMODS += trapstat
353 DRV_KMODS += vcc
354 DRV_KMODS += vdc
355 DRV_KMODS += vds
356 DRV_KMODS += vldc
357 DRV_KMODS += vlids
358 DRV_KMODS += vnet
359 DRV_KMODS += vnex
360 DRV_KMODS += vsw

362 #
363 # Exec Class Modules (/kernel/exec):
364 #
365 EXEC_KMODS +=

367 #
368 # Scheduling Class Modules (/kernel/sched):
369 #
370 SCHED_KMODS +=

372 #
373 # File System Modules (/kernel/fs):
374 #
375 FS_KMODS +=

377 #
378 # Streams Modules (/kernel/strmod):
379 #
380 # STRMOD_KMODS += kb

382 #
383 # 'System' Modules (/kernel/sys):
384 #
385 SYS_KMODS +=

387 #
388 # 'User' Modules (/kernel/misc):
389 #
390 MISC_KMODS += bootdev

```

```
391 MISC_KMODS      += dr_cpu
392 MISC_KMODS      += dr_io
393 MISC_KMODS      += dr_mem
394 MISC_KMODS      += ds
395 MISC_KMODS      += fault_iso
396 MISC_KMODS      += ldc
397 MISC_KMODS      += obpsym
398 MISC_KMODS      += platmod
399 MISC_KMODS      += platsvc
400 MISC_KMODS      += vis
401 MISC_KMODS      += pcie

403 #      md5 optimized for Niagara
404 #
405 MISC_KMODS      += md5

407 #
408 #      Brand modules
409 #
410 BRAND_KMODS     += snl_brand s10_brand

412 #
413 #      Software Cryptographic Providers (/kernel/crypto):
414 #
415 CRYPTO_KMODS   += arcfour

417 #
418 #      generic-unix module (/kernel/genunix):
419 #
420 GENUNIX_KMODS  += genunix

422 #
423 #      Modules eXcluded from the product:
424 #
425 XMODS          +=

427 #
428 #      cpu modules
429 #
430 CPU_KMODS      += generic niagara niagara2 vfalls kt

432 LINT_CPU_KMODS += generic

434 #
435 #      Performance Counter BackEnd Modules (/usr/kernel/pcbe):
436 #
437 PCBE_KMODS     += niagara_pcbe
438 PCBE_KMODS     += niagara2_pcbe
439 PCBE_KMODS     += vfalls_pcbe
440 PCBE_KMODS     += kt_pcbe
```

```

*****
2284 Tue Oct 7 15:48:37 2014
new/usr/src/uts/sun4v/arcfour/Makefile
5196 The cw wrapper restricts gcc to -O2
*****
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/sun4v/arcfour/Makefile
23 #
24 # Copyright 2014 Gary Mills
25 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
26 # Use is subject to license terms.
27 #
28 # This makefile drives the production of the arcfour kernel module.
29 #
30 # sun4v implementation architecture dependent
31 #
32 #
33 #
34 # Path to the base of the uts directory tree (usually /usr/src/uts).
35 #
36 UTSBASE = ../../
37 COM_DIR = $(COMMONBASE)/crypto/arcfour
38 #
39 #
40 # Define the module and object file sets.
41 #
42 MODULE = arcfour
43 OBJECTS = $(ARCFOUR_OBJS:%=$(OBJS_DIR)/%)
44 LINTS = $(ARCFOUR_OBJS:%.o=$(LINTS_DIR)/%.ln)
45 ROOTMODULE = $(ROOT_PSM_CRYPTODIR)/$(MODULE)
46 #
47 #
48 # Include common rules.
49 #
50 include $(UTSBASE)/sun4v/Makefile.sun4v
51 #
52 # set signing mode
53 ELFSIGN_MOD = $(ELFSIGN_CRYPTODIR)
54 #
55 #
56 # Define targets
57 #
58 ALL_TARGET = $(BINARY)
59 LINT_TARGET = $(MODULE).lint
60 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)

```

```

62 #
63 # lint pass one enforcement
64 #
65 CFLAGS += $(CCVERBOSE) -I$(COM_DIR)
66 LINTFLAGS += -I$(COM_DIR)
67 #
68 #
69 # Turn on doubleword alignment for 64 bit registers
70 #
71 CFLAGS += -dalign
72 #
73 #
74 # For niagara specific optimizations
75 #
76 COPTIMIZE = -_cc=-xO5 -_gcc=-O2 -xbuiltin=%all
77 COPTIMIZE = -xO5 -xbuiltin=%all
78 #
79 # Override the default -xspace setting
80 #
81 sparc_SPACEFLAG = -W0,-Lt
82 #
83 #
84 # Default build targets.
85 #
86 .KEEP_STATE:
87 #
88 def: $(DEF_DEPS)
89 #
90 all: $(ALL_DEPS)
91 #
92 clean: $(CLEAN_DEPS)
93 #
94 clobber: $(CLOBBER_DEPS)
95 #
96 lint: $(LINT_DEPS)
97 #
98 modlintlib: $(MODLINTLIB_DEPS)
99 #
100 clean.lint: $(CLEAN_LINT_DEPS)
101 #
102 install: $(INSTALL_DEPS)
103 #
104 #
105 # Include common targets.
106 #
107 include $(UTSBASE)/sun4v/Makefile.targ

```