

new/usr/src/Makefile

1

```
*****
7277 Thu Aug 15 11:59:20 2013
new/usr/src/Makefile
4027 remove CLOSED_BUILD
*****
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 #
26 #
27 #
28 # Makefile for system source
29 #
30 # include global definitions
31 include Makefile.master
32 #
33 # the Targetdirs file is the AT&T target.dirs file in a makefile format.
34 # it defines TARGETDIRS and ROOTDIRS.
35 include Targetdirs
36 #
37 COMMON_SUBDIRS= uts lib cmd ucplib ucblib ucblib psm man test
38 sparc_SUBDIRS= stand
39 i386_SUBDIRS= grub
40 #
41 #
42 # sparc needs to build stand before psm
43 #
44 $(SPARC_BLD)psm: stand
45 #
46 SUBDIRS= $(COMMON_SUBDIRS) $($ (MACH)_SUBDIRS)
47 #
48 HDRSUBDIRS= uts head lib cmd
49 #
50 # UCB headers are bug-for-bug compatible and not checkable against the header
51 # standards.
52 #
53 CHKHDRSUBDIRS= head uts lib
54 #
55 #
56 # Headers that can be built in parallel
57 #
58 PARALLEL_HEADERS = sysheaders userheaders libheaders cmdheaders
59 #
60 #
61 # Directories that can be built in parallel
```

new/usr/src/Makefile

2

```
62 #
63 PARALLEL_DIRS = uts lib man
64 #
65 # The check target also causes smf(5) service manifests to be validated.
66 CHKMFSTSUBDIRS= cmd
67 #
68 MSGSUBDIRS= cmd ucblib lib
69 DOMAINS= \
70 SUNW_OST_ADMIN \
71 SUNW_OST_NETRPC \
72 SUNW_OST_OSCMD \
73 SUNW_OST_OSLIB \
74 SUNW_OST_UCBCMD \
75 SUNW_OST_ZONEINFO
76 #
77 MSGDDIRS= $(DOMAINS:%=$(MSGROOT)/%)
78 MSGDIRS= $(MSGROOT) $(MSGDDIRS) $(MSGROOT)/LC_TIME
79 #
80 all := TARGET= all
81 install := TARGET= install
82 install1 := TARGET= install
83 install2 := TARGET= install
84 install_h := TARGET= install_h
85 clean := TARGET= clean
86 clobber := TARGET= clobber
87 check := TARGET= check
88 #
89 .KEEP_STATE:
90 #
91 #
92 # Note: install does not cause a build in pkg. To build packages,
93 # cd pkg and do a 'make install'
94 #
95 #
96 all: mapfiles closedbins sgs .WAIT $(SUBDIRS) pkg
97 #
98 #
99 # The _msg build is a two-step process. First, the _msg dependency
100 # causes recursive makes in $(MSGSUBDIRS), which stages raw message
101 # files in $(ROOT)/catalog. Second, the action from the install
102 # target rule causes those messages to be post-processed from where
103 # they were staged in $(ROOT)/catalog, and the results placed into the
104 # proto area.
105 #
106 # The stage-licenses target causes the license files needed for
107 # packaging to be pulled from $(SRC) and $(CLOSED) and staged in
108 # $(ROOT)/licenses.
109 #
110 install: install1 install2 _msg stage-licenses
111 @cd msg; pwd; $(MAKE) _msg
112 @rm -rf "$(ROOT)/catalog"
113 #
114 stage-licenses: install2
115 @cd pkg; pwd; $(MAKE) stage-licenses
116 #
117 install1: mapfiles closedbins sgs
118 #
119 install2: install1 $(SUBDIRS)
120 #
121 _msg: _msgdirs rootdirs install2 FRC
122 @for m in $(MSGSUBDIRS); do \
123 cd $$m; pwd; $(MAKE) _msg; cd ..; \
124 done
125 #
126 mapfiles: bldtools
127 @cd common/mapfiles; pwd; $(MAKE) install
```

```

129 clean clobber: $(SUBDIRS) head pkg

131 closedbins: bldtools $(ROOTDIRS) FRC
132   @CLOSED_ROOT=$(ON_CLOSED_BINS/root_$(MACH)${RELEASE_BUILD+nd}"; \
133   if [ "$CLOSED_IS_PRESENT" = no ]; then \
134     if [ ! -d "$CLOSED_ROOT" ]; then \
135       $(ECHO) "Error: ON_CLOSED_BINS must point to closed" \
136         "binaries."; \
137       $(ECHO) "Error: if closed sources are not present," \
138         "ON_CLOSED_BINS must point to closed binaries."; \
139       $(ECHO) "root_$(MACH)${RELEASE_BUILD+nd} is not" \
140         "present in $$ON_CLOSED_BINS."; \
141       exit 1; \
142     fi; \
143     $(ECHO) "Copying closed binaries from $$CLOSED_ROOT"; \
144     (cd $$CLOSED_ROOT; \
145       $(TAR) cfX - $(CODEMGR_WS)/exception_lists/closed-bins .) | \
146       (cd $(ROOT); $(TAR) xBpf -); \
147     ( cd $(ROOT); $(CTFSTRIP) $(cd $$CLOSED_ROOT; $(FIND) \
148       ./kernel ./usr/kernel ./platform/*/kernel -type f -a -perm -
149       $(EGREP) -vf $(CODEMGR_WS)/exception_lists/closed-bins) ); \
150   fi

151 #
152 # Declare what parts can be built in parallel
153 # DUMMY at the end is used in case macro expansion produces an empty string to
154 # prevent everything going in parallel
155 #
156 .PARALLEL: $(PARALLEL_HEADERS) DUMMY
157 .PARALLEL: $(PARALLEL_DIRS) DUMMY

158 $(SUBDIRS) head pkg: FRC
159   @cd $@; pwd; $(MAKE) $(TARGET)

161 # librpcsvc has a dependency on headers installed by
162 # userheaders, hence the .WAIT before libheaders.
163 sgs: rootdirs .WAIT sysheaders userheaders .WAIT \
164   libheaders cmdheaders

166 #
167 # Top-level setup target to setup the development environment that includes
168 # headers, tools and generated mapfiles. For open-only builds (i.e.: source
169 # trees w/o usr/closed), this also depends on the closedbins target (above)
170 # in order to properly seed the proto area. Note, although the tools are
171 # dependent on a number of constant mapfiles, the tools themselves are
172 # required to build the generated mapfiles.
173 #
174 setup: closedbins bldtools sgs mapfiles

176 bldtools:
177   @cd tools; pwd; $(MAKE) install

179 # /var/mail/:saved is a special case because of the colon in the name.
180 #
181 rootdirs: $(ROOTDIRS)
182   $(INS) -d -m 775 $(ROOT)/var/mail/:saved

184 lint: FRC
185   $(MAKE) -f Makefile.lint

187 _msgdirs: $(MSGDIRS)

189 $(ROOTDIRS) $(MSGDIRS):
190   $(INS.dir)

```

```

192 userheaders: FRC
193   @cd head; pwd; $(MAKE) install_h

195 libheaders: bldtools
196   @cd lib; pwd; $(MAKE) install_h

198 sysheaders: FRC
199   @cd uts; pwd; $(MAKE) install_h

201 cmdheaders: FRC
202   @cd cmd/fm; pwd; $(MAKE) install_h
203   @cd cmd/mdb; pwd; $(MAKE) install_h

205 check: $(CHKHDRSUBDIRS) $(CHKMFSTSUBDIRS)

207 #
208 # Cross-reference customization: skip all of the subdirectories that
209 # don't contain actual source code.
210 #
211 $(CLOSED_BUILD)XRDIRS += ../closed
212 XRPRUNE = pkg prototypes
213 XRINCDIRS = uts/common head ucbread
214 $(CLOSED_BUILD)XRINCDIRS = uts/common ../closed/uts/common head ucbread

216 cscope.out tags: FRC
217   $(XREF) -f -x $@

219 #
220 # Targets for reporting compiler versions; nightly uses these.
221 #

222 cc-version:
223   @if $(MACH_CC) -_versions >/dev/null 2>/dev/null; then \
224     $(ECHO) 32-bit compiler; \
225     $(ECHO) $(MACH_CC); \
226     $(MACH_CC) -_versions 2>&1 | \
227     $(EGREP) '^ (cw|cc|gcc|primary|shadow)'; \
228   else \
229     __COMPILER=$(MACH_CC) -_compiler 2>/dev/null || $(TRUE); \
230     if [ -z "$__COMPILER" ]; then \
231       $(ECHO) No 32-bit compiler found; \
232       exit 1; \
233     else \
234       $(ECHO) 32-bit compiler; \
235       $(ECHO) $(MACH_CC); \
236       $(ECHO) $__COMPILER; \
237       $(MACH_CC) -V 2>&1 | head -1; \
238     fi; \
239   fi

241 cc64-version:
242   @if $(MACH64_CC) -_versions >/dev/null 2>/dev/null; then \
243     $(ECHO) 64-bit compiler; \
244     $(ECHO) $(MACH64_CC); \
245     $(MACH64_CC) -_versions 2>&1 | \
246     $(EGREP) '^ (cw|cc|gcc|primary|shadow)'; \
247   else \
248     __COMPILER=$(MACH64_CC) -_compiler 2>/dev/null || $(TRUE); \
249     if [ -z "$__COMPILER" ]; then \
250       $(ECHO) No 64-bit compiler found; \
251       exit 1; \
252     else \
253       $(ECHO) 64-bit compiler; \
254       $(ECHO) $(MACH64_CC); \
255     fi; \

```

new/usr/src/Makefile

5

```
256             $(ECHO) $$__COMPILER;           \
257             $$((MACH64)_CC) -V 2>&1 | head -1; \
258             fi;                               \
259     fi;

261 java-version:
262     @if [ -x "$(JAVAC)" ]; then              \
263         $(ECHO) "$(JAVAC)";                 \
264         $(JAVA_ROOT)/bin/java -fullversion 2>&1 | head -1; \
265     else
266         $(ECHO) No Java compiler found;     \
267         exit 1;                             \
268     fi
```

```

*****
35365 Thu Aug 15 11:59:20 2013
new/usr/src/Makefile.master
4027 remove CLOSED_BUILD
*****
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 #
26 #
27 #
28 # Makefile.master, global definitions for system source
29 #
30 ROOT=          /proto
31 #
32 #
33 # RELEASE_BUILD should be cleared for final release builds.
34 # NOT_RELEASE_BUILD is exactly what the name implies.
35 #
36 # INTERNAL_RELEASE_BUILD is a subset of RELEASE_BUILD. It mostly controls
37 # identification strings. Enabling RELEASE_BUILD automatically enables
38 # INTERNAL_RELEASE_BUILD.
39 #
40 # CLOSED_BUILD controls whether we try to build files under
41 # usr/closed. (" means to build closed code, "# means don't try to
42 # build it.) Skipping the closed code implies doing an export release
43 # build.
44 #
45 # STRIP_COMMENTS toggles comment section stripping. Generally the same setting
46 # as INTERNAL_RELEASE_BUILD.
47 #
48 # __GNUC toggles the building of ON components using gcc and related tools.
49 # Normally set to '#', set it to '' to do gcc build.
50 #
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 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #

```

```

58 PRE_POUND=          pre\#
59 POUND_SIGN=        $(PRE_POUND:pre\#=%)

61 NOT_RELEASE_BUILD=
62 INTERNAL_RELEASE_BUILD=  $(POUND_SIGN)
63 RELEASE_BUILD=        $(POUND_SIGN)
64 $(RELEASE_BUILD)NOT_RELEASE_BUILD=  $(POUND_SIGN)
65 $(RELEASE_BUILD)INTERNAL_RELEASE_BUILD=
66 PATCH_BUILD=        $(POUND_SIGN)

73 # If CLOSED_IS_PRESENT is not set, assume the closed tree is present.
74 CLOSED_BUILD_1=  $(CLOSED_IS_PRESENT:yes=)
75 CLOSED_BUILD=    $(CLOSED_BUILD_1:no=$(POUND_SIGN))

68 # SPARC_BLD is '#' for an Intel build.
69 # INTEL_BLD is '#' for a Sparc build.
70 SPARC_BLD_1=    $(MACH:i386=$(POUND_SIGN))
71 SPARC_BLD=      $(SPARC_BLD_1:sparc=)
72 INTEL_BLD_1=    $(MACH:sparc=$(POUND_SIGN))
73 INTEL_BLD=      $(INTEL_BLD_1:i386=)

75 STRIP_COMMENTS=  $(INTERNAL_RELEASE_BUILD)

77 # Are we building tonic closedbins? Unless you have used the
78 # -O flag to nightly or bldenv, leave the definition of TONICBUILD
79 # as $(POUND_SIGN).
80 #
81 # IF YOU CHANGE CLOSEDROOT, you MUST change install.bin
82 # to match the new definition.
83 TONICBUILD=      $(POUND_SIGN)
84 $(TONICBUILD)CLOSEDROOT= $(ROOT)-closed

87 # The variables below control the compilers used during the build.
88 # There are a number of permutations.
89 #
90 # __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
91 # one is not POUND_SIGN is the primary, with the other as the shadow.  They
92 # may also be used to control entirely compiler-specific Makefile assignments.
93 # __SUNC and Sun Studio are the default.
94 #
95 # __GNUC64 indicates that the 64bit build should use the GNU C compiler.
96 # There is no Sun C analogue.
97 #
98 # The following version-specific options are operative regardless of which
99 # compiler is primary, and control the versions of the given compilers to be
100 # used.  They also allow compiler-version specific Makefile fragments.
101 #
102 #
103 __GNUC=          $(POUND_SIGN)
104 $(__GNUC)__SUNC=  $(POUND_SIGN)
105 __GNUC64=        $(__GNUC)

107 # CLOSED is the root of the tree that contains source which isn't released
108 # as open source
109 CLOSED=          $(SRC)/../closed

111 # BUILD_TOOLS is the root of all tools including compilers.
112 # ONBLD_TOOLS is the root of all the tools that are part of SUNWonbld.

114 BUILD_TOOLS=     /ws/onnv-tools
115 ONBLD_TOOLS=     $(BUILD_TOOLS)/onbld

117 JAVA_ROOT=      /usr/java

```

new/usr/src/Makefile.master

```

119 SFW_ROOT=          /usr/sfw
120 SFWINCDIR=         $(SFW_ROOT)/include
121 SFWLIBDIR=         $(SFW_ROOT)/lib
122 SFWLIBDIR64=       $(SFW_ROOT)/lib/$(MACH64)

124 GCC_ROOT=          /opt/gcc/4.4.4
125 GCCLIBDIR=         $(GCC_ROOT)/lib
126 GCCLIBDIR64=       $(GCC_ROOT)/lib/$(MACH64)

128 DOCBOOK_XSL_ROOT= /usr/share/sgml/docbook/xsl-stylesheets

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

```

3

new/usr/src/Makefile.master

```

185 DTRACE=             /usr/sbin/dtrace -xnolib
186 UNIQ=                /usr/bin/uniq
187 TAR=                 /usr/bin/tar

189 FILEMODE=           644
190 DIRMODE=            755

192 #
193 # The version of the patch makeup table optimized for build-time use. Used
194 # during patch builds only.
195 $(PATCH_BUILD)PMTMO_FILE=$(SRC)/patch_makeup_table.mo

197 # Declare that nothing should be built in parallel.
198 # Individual Makefiles can use the .PARALLEL target to declare otherwise.
199 .NO_PARALLEL:

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

212 DOT_H_CHECK=         \
213     @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL); \
214     $(HDRCHK) $< $(HDRCHK_TAIL)

216 DOT_X_CHECK=         \
217     @$(ECHO) "checking $<"; $(RPCGEN) -C -h $< | $(CSTYLE) $(CSTYLE_TAIL); \
218     $(RPCGEN) -C -h $< | $(HDRCHK) $< $(HDRCHK_TAIL)

220 DOT_C_CHECK=         \
221     @$(ECHO) "checking $<"; $(CSTYLE) $< $(CSTYLE_TAIL)

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

230 #
231 # IMPORTANT:: If you change any of INS.file, INS.dir, INS.rename,
232 # INS.link or INS.symlink here, then you must also change the
233 # corresponding override definitions in $CLOSED/Makefile.tonic.
234 # If you do not do this, then the closedbins build for the OpenSolaris
235 # community will break. PS, the gatekeepers will be upset too.
236 INS.file=             $(RM) $@; $(INS) -s -m $(FILEMODE) -f $(@D) $<
237 INS.dir=              $(INS) -s -d -m $(DIRMODE) $@
238 # installs and renames at once
239 #
240 INS.rename=           $(INS.file); $(MV) $(@D)/$(<F) $@

242 # install a link
243 INSLINKTARGET=       $<
244 INS.link=             $(RM) $@; $(LN) $(INSLINKTARGET) $@
245 INS.symlink=         $(RM) $@; $(SYMLINK) $(INSLINKTARGET) $@

247 #
248 # Python bakes the mtime of the .py file into the compiled .pyc and
249 # rebuilds if the baked-in mtime != the mtime of the source file
250 # (rather than only if it's less than), thus when installing python

```

4

```

251 # files we must make certain to not adjust the mtime of the source
252 # (.py) file.
253 #
254 INS.pyfile=      $(INS.file); $(TOUCH) -r $< $@

256 # MACH must be set in the shell environment per uname -p on the build host
257 # More specific architecture variables should be set in lower makefiles.
258 #
259 # MACH64 is derived from MACH, and BUILD64 is set to '#' for
260 # architectures on which we do not build 64-bit versions.
261 # (There are no such architectures at the moment.)
262 #
263 # Set BUILD64=# in the environment to disable 64-bit amd64
264 # builds on i386 machines.

266 MACH64_1=        $(MACH:sparc=sparcv9)
267 MACH64=          $(MACH64_1:i386=amd64)

269 MACH32_1=        $(MACH:sparc=sparcv7)
270 MACH32=          $(MACH32_1:i386=i86)

272 sparc_BUILD64=
273 i386_BUILD64=
274 BUILD64=        $($ (MACH)_BUILD64)

276 #
277 # C compiler mode. Future compilers may change the default on us,
278 # so force extended ANSI mode globally. Lower level makefiles can
279 # override this by setting CCMODE.
280 #
281 CCMODE=          -Xa
282 CCMODE64=        -Xa

284 #
285 # C compiler verbose mode. This is so we can enable it globally,
286 # but turn it off in the lower level makefiles of things we cannot
287 # (or aren't going to) fix.
288 #
289 CCVERBOSE=      -v

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

295 # set this to the secret flag "-Wc,-Qiselect-regsym=0" to disable register
296 # symbols (used to detect conflicts between objects that use global registers)
297 # we disable this now for safety, and because genunix doesn't link with
298 # this feature (the v9 default) enabled.
299 #
300 # REGSYM is separate since the C++ driver syntax is different.
301 CCREGSYM=        -Wc,-Qiselect-regsym=0
302 CCCREGSYM=       -Ooption cg -Qiselect-regsym=0

304 # Prevent the removal of static symbols by the SPARC code generator (cg).
305 # The x86 code generator (ube) does not remove such symbols and as such
306 # using this workaround is not applicable for x86.
307 #
308 CCSTATICSYM=     -Wc,-Qassembler-ounrefsym=0
309 #
310 # generate 32-bit addresses in the v9 kernel. Saves memory.
311 CCABS32=         -Wc,-xcode=abs32
312 #
313 # generate v9 code which tolerates callers using the v7 ABI, for the sake of
314 # system calls.
315 CC32BITCALLERS=  _gcc=-massume-32bit-callers

```

```

317 # GCC, especially, is increasingly beginning to auto-inline functions and
318 # sadly does so separately not under the general -fno-inline-functions
319 # Additionally, we wish to prevent optimisations which cause GCC to clone
320 # functions -- in particular, these may cause unhelpful symbols to be
321 # emitted instead of function names
322 CCNOAUTOINLINE= _gcc=-fno-inline-small-functions \
323                _gcc=-fno-inline-functions-called-once \
324                _gcc=-fno-ipa-cp

326 # One optimization the compiler might perform is to turn this:
327 #   #pragma weak foo
328 #   extern int foo;
329 #   if (&foo)
330 #       foo = 5;
331 # into
332 #   foo = 5;
333 # Since we do some of this (foo might be referenced in common kernel code
334 # but provided only for some cpu modules or platforms), we disable this
335 # optimization.
336 #
337 sparc_CCUNBOUND = -Wd,-xsafe=unboundsym
338 i386_CCUNBOUND =
339 CCUNBOUND =      $($ (MACH)_CCUNBOUND)

341 #
342 # compiler '-xarch' flag. This is here to centralize it and make it
343 # overridable for testing.
344 sparc_XARCH=     -m32
345 sparcv9_XARCH=  -m64
346 i386_XARCH=
347 amd64_XARCH=     -m64 -Ui386 -U_i386

349 # assembler '-xarch' flag. Different from compiler '-xarch' flag.
350 sparc_AS_XARCH=  -xarch=v8plus
351 sparcv9_AS_XARCH= -xarch=v9
352 i386_AS_XARCH=
353 amd64_AS_XARCH=  -xarch=amd64 -P -Ui386 -U_i386

355 #
356 # These flags define what we need to be 'standalone' i.e. -not- part
357 # of the rather more cosy userland environment. This basically means
358 # the kernel.
359 #
360 # XX64 future versions of gcc will make -mmodel=kernel imply -mno-red-zone
361 #
362 sparc_STAND_FLAGS=  _gcc=-ffreestanding
363 sparcv9_STAND_FLAGS= _gcc=-ffreestanding
364 # Disabling MMX also disables 3DNow, disabling SSE also disables all later
365 # additions to SSE (SSE2, AVX ,etc.)
366 NO_SIMD=           _gcc=-mno-mmx _gcc=-mno-sse
367 i386_STAND_FLAGS=  _gcc=-ffreestanding $(NO_SIMD)
368 amd64_STAND_FLAGS= -xmodel=kernel $(NO_SIMD)

370 SAVEARGS=          -Wu,-save_args
371 amd64_STAND_FLAGS += $(SAVEARGS)

373 STAND_FLAGS_32 =   $($ (MACH)_STAND_FLAGS)
374 STAND_FLAGS_64 =   $($ (MACH64)_STAND_FLAGS)

376 #
377 # disable the incremental linker
378 ILDOFF=            -xildoff
379 #
380 XDEPEND=           -xdepend
381 XFFLAG=            -xF=%all
382 XESS=              -xs

```

```

383 XSTRCONST=          -xstrconst

385 #
386 # turn warnings into errors (C)
387 CERRWARN = -errtags=yes -errwarn=all
388 CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
389 CERRWARN += -erroff=E_STATEMENT_NOT_REACHED

391 CERRWARN += -_gcc=-Wno-missing-braces
392 CERRWARN += -_gcc=-Wno-sign-compare
393 CERRWARN += -_gcc=-Wno-unknown-pragmas
394 CERRWARN += -_gcc=-Wno-unused-parameter
395 CERRWARN += -_gcc=-Wno-missing-field-initializers

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

400 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
401 # -nd builds
402 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
403 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

405 #
406 # turn warnings into errors (C++)
407 CCERRWARN=          -xwe

409 # C99 mode
410 C99_ENABLE=         -xc99=%all
411 C99_DISABLE=        -xc99=%none
412 C99MODE=            $(C99_DISABLE)
413 C99LMODE=           $(C99MODE:-xc99%=-Xc99%)

415 # In most places, assignments to these macros should be appended with +=
416 # (CPPFLAGS.master allows values to be prepended to CPPFLAGS).
417 sparc_CFLAGS=       $(sparc_XARCH) $(CCSTATICSYM)
418 sparcv9_CFLAGS=     $(sparcv9_XARCH) -dalign $(CCVERBOSE) $(V9ABIWARN) $(CCREGSYM) \
419                     $(CCSTATICSYM)
420 i386_CFLAGS=        $(i386_XARCH)
421 amd64_CFLAGS=       $(amd64_XARCH)

423 sparc_ASFLAGS=      $(sparc_AS_XARCH)
424 sparcv9_ASFLAGS=    $(sparcv9_AS_XARCH)
425 i386_ASFLAGS=       $(i386_AS_XARCH)
426 amd64_ASFLAGS=      $(amd64_AS_XARCH)

428 #
429 sparc_COPTFLAG=     -xO3
430 sparcv9_COPTFLAG=   -xO3
431 i386_COPTFLAG=      -O
432 amd64_COPTFLAG=     -xO3

434 COPTFLAG=          $(($(MACH)_COPTFLAG)
435 COPTFLAG64=        $(($(MACH64)_COPTFLAG)

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

441 # Direct the Sun Studio compiler to use a static globalization prefix based on t
442 # name of the module rather than something unique. Otherwise, objects
443 # will not build deterministically, as subsequent compilations of identical
444 # source will yield objects that always look different.
445 #
446 # In the same spirit, this will also remove the date from the N_OPT stab.
447 CGLOBALSTATIC= -W0,-xglobalstatic

```

```

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

453 #
454 # Default debug format for Sun Studio 11 is dwarf, so force it to
455 # generate stabs.
456 #
457 DEBUGFORMAT=       -xdebugformat=stabs

459 #
460 # Flags used to build in debug mode for ctf generation. Bugs in the Devpro
461 # compilers currently prevent us from building with cc-emitted DWARF.
462 #
463 CTF_FLAGS_sparc = -g -Wc,-Qiselect-T1 $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)
464 CTF_FLAGS_i386 = -g $(C99MODE) $(CNOGLOBAL) $(CDWARFSTR)

466 CTF_FLAGS_sparcv9 = $(CTF_FLAGS_sparc)
467 CTF_FLAGS_amd64 = $(CTF_FLAGS_i386)

469 # Sun Studio produces broken userland code when saving arguments.
470 $(__GNUCC)CTF_FLAGS_amd64 += $(SAVEARGS)

472 CTF_FLAGS_32 = $(CTF_FLAGS_$(MACH)) $(DEBUGFORMAT)
473 CTF_FLAGS_64 = $(CTF_FLAGS_$(MACH64)) $(DEBUGFORMAT)
474 CTF_FLAGS = $(CTF_FLAGS_32)

476 #
477 # Flags used with genoffsets
478 #
479 GOFLAGS = -_noecho \
480           $(CALLSYMS) \
481           $(CDWARFSTR)

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

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

489 #
490 # tradeoff time for space (smaller is better)
491 #
492 sparc_SPACEFLAG = -xspace -W0,-Lt
493 sparcv9_SPACEFLAG = -xspace -W0,-Lt
494 i386_SPACEFLAG = -xspace
495 amd64_SPACEFLAG =

497 SPACEFLAG = $(($(MACH)_SPACEFLAG)
498 SPACEFLAG64 = $(($(MACH64)_SPACEFLAG)

500 #
501 # The Sun Studio 11 compiler has changed the behaviour of integer
502 # wrap arounds and so a flag is needed to use the legacy behaviour
503 # (without this flag panics/hangs could be exposed within the source).
504 #
505 sparc_IROPTFLAG = -W2,-xwrap_int
506 sparcv9_IROPTFLAG = -W2,-xwrap_int
507 i386_IROPTFLAG =
508 amd64_IROPTFLAG =

510 IROPTFLAG = $(($(MACH)_IROPTFLAG)
511 IROPTFLAG64 = $(($(MACH64)_IROPTFLAG)

513 sparc_XREGSFLAG = -xregs=no%appl
514 sparcv9_XREGSFLAG = -xregs=no%appl

```

```

515 i386_XREGSFLAG      =
516 amd64_XREGSFLAG    =

518 XREGSFLAG          = ${$(MACH)_XREGSFLAG}
519 XREGSFLAG64        = ${$(MACH64)_XREGSFLAG}

521 # dmake SOURCEDEBUG=yes ... enables source-level debugging information, and
522 # avoids stripping it.
523 SOURCEDEBUG         = $(POUND_SIGN)
524 SRCDBGBLD           = $(SOURCEDEBUG:yes=)

526 #
527 # These variables are intended ONLY for use by developers to safely pass extra
528 # flags to the compilers without unintentionally overriding Makefile-set
529 # flags. They should NEVER be set to any value in a Makefile.
530 #
531 # They come last in the associated FLAGS variable such that they can
532 # explicitly override things if necessary, there are gaps in this, but it's
533 # the best we can manage.
534 #
535 CUSERFLAGS          =
536 CUSERFLAGS64        = $(CUSERFLAGS)
537 CCUSERFLAGS         =
538 CCUSERFLAGS64       = $(CCUSERFLAGS)

540 CSOURCEDEBUGFLAGS  =
541 CCSOURCEDEBUGFLAGS =
542 $(SRCDBGBLD)CSOURCEDEBUGFLAGS = -g -xs
543 $(SRCDBGBLD)CCSOURCEDEBUGFLAGS = -g -xs

545 CFLAGS=              $(COPTFLAG) ${$(MACH)_CFLAGS} $(SPACEFLAG) $(CCMODE) \
546                      $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG) \
547                      $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
548                      $(CUSERFLAGS)
549 CFLAGS64=            $(COPTFLAG64) ${$(MACH64)_CFLAGS} $(SPACEFLAG64) $(CCMODE64) \
550                      $(ILDOFF) $(CERRWARN) $(C99MODE) $(CCUNBOUND) $(IROPTFLAG64) \
551                      $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CSOURCEDEBUGFLAGS) \
552                      $(CUSERFLAGS64)
553 #
554 # Flags that are used to build parts of the code that are subsequently
555 # run on the build machine (also known as the NATIVE_BUILD).
556 #
557 NATIVE_CFLAGS=       $(COPTFLAG) ${$(NATIVE_MACH)_CFLAGS} $(CCMODE) \
558                      $(ILDOFF) $(CERRWARN) $(C99MODE) ${$(NATIVE_MACH)_CCUNBOUND} \
559                      $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
560                      $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

562 DTEXTDOM=-DTEXT_DOMAIN="\${TEXT_DOMAIN}"      # For messaging.
563 DTS_ERRNO=-D_TS_ERRNO
564 CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
565                 $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
566 CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4)
567 CPPFLAGS=         $(CPPFLAGS.master)
568 AS_CPPFLAGS=      $(CPPFLAGS.master)
569 JAVAFLAGS=        -deprecation

571 #
572 # For source message catalogue
573 #
574 .SUFFIXES: $(SUFFIXES) .i .po
575 MSGROOT= $(ROOT)/catalog
576 MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
577 MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
578 DCMMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
579 DCMMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)

```

```

581 CLOBBERFILES += $(POFILE) $(POFILES)
582 COMPILE.cpp= $(CC) -E -C $(CFLAGS) $(CPPFLAGS)
583 XGETTEXT= /usr/bin/xgettext
584 XGETFLAGS= -c TRANSLATION_NOTE
585 GNUXGETTEXT= /usr/gnu/bin/xgettext
586 GNUXGETFLAGS= --add-comments=TRANSLATION_NOTE --keyword=_ \
587               --strict --no-location --omit-header
588 BUILD.po= $(XGETTEXT) $(XGETFLAGS) -d $(<F) $<.i ;\
589           $(RM) $@ ;\
590           $(SED) "/^domain/d" < $(<F).po > $@ ;\
591           $(RM) $(<F).po $<.i

593 #
594 # This is overwritten by local Makefile when PROG is a list.
595 #
596 POFILE= $(PROG).po

598 sparc_CCFLAGS=      -cg92 -compat=4 \
599                   -Qoption ccfe -messages=no%anachronism \
600                   $(CCERRWARN)
601 sparcv9_CCFLAGS=    $(sparcv9_XARCH) -dalign -compat=5 \
602                   -Qoption ccfe -messages=no%anachronism \
603                   -Qoption ccfe -features=no%conststrings \
604                   $(CCCREGSYM) \
605                   $(CCERRWARN)
606 i386_CCFLAGS=       -compat=4 \
607                   -Qoption ccfe -messages=no%anachronism \
608                   -Qoption ccfe -features=no%conststrings \
609                   $(CCERRWARN)
610 amd64_CCFLAGS=      $(amd64_XARCH) -compat=5 \
611                   -Qoption ccfe -messages=no%anachronism \
612                   -Qoption ccfe -features=no%conststrings \
613                   $(CCERRWARN)

615 sparc_CCOPTFLAG=    -O
616 sparcv9_CCOPTFLAG=  -O
617 i386_CCOPTFLAG=     -O
618 amd64_CCOPTFLAG=    -O

620 CCOPTFLAG=          ${$(MACH)_CCOPTFLAG}
621 CCOPTFLAG64=        ${$(MACH64)_CCOPTFLAG}
622 CCFLAGS=             $(COPTFLAG) ${$(MACH)_CCFLAGS} $(CCSOURCEDEBUGFLAGS) \
623                     $(CCUSERFLAGS)
624 CCFLAGS64=           $(COPTFLAG64) ${$(MACH64)_CCFLAGS} $(CCSOURCEDEBUGFLAGS) \
625                     $(CCUSERFLAGS64)

627 #
628 #
629 #
630 ELFWRAP_FLAGS =
631 ELFWRAP_FLAGS64 = -64

633 #
634 # Various mapfiles that are used throughout the build, and delivered to
635 # /usr/lib/ld.
636 #
637 MAPFILE.NED_i386 = $(SRC)/common/mapfiles/common/map.noexdata
638 MAPFILE.NED_sparc =
639 MAPFILE.NED = $(MAPFILE.NED_$(MACH))
640 MAPFILE.PGA = $(SRC)/common/mapfiles/common/map.pagealign
641 MAPFILE.NES = $(SRC)/common/mapfiles/common/map.noexstk
642 MAPFILE.FLT = $(SRC)/common/mapfiles/common/map.filter
643 MAPFILE.LEX = $(SRC)/common/mapfiles/common/map.lex.yy

645 #
646 # Generated mapfiles that are compiler specific, and used throughout the

```



```

647 # build.  These mapfiles are not delivered in /usr/lib/ld.
648 #
649 MAPFILE.NGB_sparc= $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
650 $(__GNUC64)MAPFILE.NGB_sparc= \
651 $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
652 MAPFILE.NGB_sparcv9= $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
653 $(__GNUC64)MAPFILE.NGB_sparcv9= \
654 $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
655 MAPFILE.NGB_i386= $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
656 $(__GNUC64)MAPFILE.NGB_i386= \
657 $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
658 MAPFILE.NGB_amd64= $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
659 $(__GNUC64)MAPFILE.NGB_amd64= \
660 $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
661 MAPFILE.NGB = $(MAPFILE.NGB_$(MACH))

663 #
664 # A generic interface mapfile name, used by various dynamic objects to define
665 # the interfaces and interposers the object must export.
666 #
667 MAPFILE.INT = mapfile-intf

669 #
670 # LDLIBS32 can be set in the environment to override the following assignment.
671 # LDLIBS64 can be set to override the assignment made in Makefile.master.64.
672 # These environment settings make sure that no libraries are searched outside
673 # of the local workspace proto area:
674 # LDLIBS32=-YP,$ROOT/lib:$ROOT/usr/lib
675 # LDLIBS64=-YP,$ROOT/lib:$MACH64:$ROOT/usr/lib:$MACH64
676 #
677 LDLIBS32 = $(ENVLDLIBS1) $(ENVLDLIBS2) $(ENVLDLIBS3)
678 LDLIBS.cmd = $(LDLIBS32)
679 LDLIBS.lib = $(LDLIBS32)
680 #
681 # Define compilation macros.
682 #
683 COMPILER.c= $(CC) $(CFLAGS) $(CPPFLAGS) -c
684 COMPILER64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) -c
685 COMPILER.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
686 COMPILER64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
687 COMPILER.s= $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
688 COMPILER64.s= $(AS) $(ASFLAGS) $(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
689 COMPILER.d= $(DTRACE) -G -32
690 COMPILER64.d= $(DTRACE) -G -64
691 COMPILER.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
692 COMPILER64.b= $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

694 CLASSPATH=
695 COMPILER.java= $(JAVAC) $(JAVAFLAGS) -classpath $(CLASSPATH)

697 #
698 # Link time macros
699 #
700 CCNEEDED = -lc
701 CCEXTNEEDED = -lCrun -lCstd
702 $(__GNUC)CCNEEDED = -L$(GCCLIBDIR) -R$(GCCLIBDIR) -lstc++ -lgcc_s
703 $(__GNUC)CCEXTNEEDED = $(CCNEEDED)

705 LINK.c= $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
706 LINK64.c= $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
707 NORUNPATH= -norunpath -nolib
708 LINK.cc= $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
709 $(LDFLAGS) $(CCNEEDED)
710 LINK64.cc= $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
711 $(LDFLAGS) $(CCNEEDED)

```

```

713 #
714 # lint macros
715 #
716 # Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
717 # ON is built with a version of lint that has the fix for 4484186.
718 #
719 ALWAYS_LINT_DEFS = -errtags=yes -s
720 ALWAYS_LINT_DEFS += -erroff=E_PTRDIFF_OVERFLOW
721 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_NARROW_CONV
722 ALWAYS_LINT_DEFS += -U__PRAGMA_REDEFINE_EXTNAME
723 ALWAYS_LINT_DEFS += $(C99LMODE)
724 ALWAYS_LINT_DEFS += -errsecurity=$(SECLEVEL)
725 ALWAYS_LINT_DEFS += -erroff=E_SEC_CREAT_WITHOUT_EXCL
726 ALWAYS_LINT_DEFS += -erroff=E_SEC_FORBIDDEN_WARN_CREAT
727 # XX64 -- really only needed for amd64 lint
728 ALWAYS_LINT_DEFS += -erroff=E_ASSIGN_INT_TO_SMALL_INT
729 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_CONST_TO_SMALL_INT
730 ALWAYS_LINT_DEFS += -erroff=E_CAST_INT_TO_SMALL_INT
731 ALWAYS_LINT_DEFS += -erroff=E_CAST_PTR_FROM_INT
732 ALWAYS_LINT_DEFS += -erroff=E_COMP_INT_WITH_LARGE_INT
733 ALWAYS_LINT_DEFS += -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
734 ALWAYS_LINT_DEFS += -erroff=E_PASS_INT_TO_SMALL_INT
735 ALWAYS_LINT_DEFS += -erroff=E_PTR_CONV_LOSES_BITS

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

741 SECLEVEL= core
742 LINT.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
743 $(ALWAYS_LINT_DEFS)
744 LINT64.c= $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
745 $(ALWAYS_LINT_DEFS)
746 LINT.s= $(LINT.c)

748 # For some future builds, NATIVE_MACH and MACH might be different.
749 # Therefore, NATIVE_MACH needs to be redefined in the
750 # environment as 'uname -p' to override this macro.
751 #
752 # For now at least, we cross-compile amd64 on i386 machines.
753 NATIVE_MACH= $(MACH:amd64=i386)

755 # Define native compilation macros
756 #

758 # Base directory where compilers are loaded.
759 # Defined here so it can be overridden by developer.
760 #
761 SPRO_ROOT= $(BUILD_TOOLS)/SUNWspro
762 SPRO_VROOT= $(SPRO_ROOT)/SS12
763 GNU_ROOT= $(SFW_ROOT)

765 # Till SS12ul formally becomes the NV CBE, LINT is hard
766 # coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
767 # location. Impacted variables are sparc_LINT, sparcv9_LINT,
768 # i386_LINT, amd64_LINT.
769 # Reset them when SS12ul is rolled out.
770 #

772 # Specify platform compiler versions for languages
773 # that we use (currently only c and c++).
774 #
775 sparc_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
776 $(__GNUC)sparc_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -gcc
777 sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
778 $(__GNUC)sparc_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++

```

```

779 sparc_CPP=          /usr/ccs/lib/cpp
780 sparc_AS=           /usr/ccs/bin/as -xregsym=no
781 sparc_LD=           /usr/ccs/bin/ld
782 sparc_LINT=         $(SPRO_ROOT)/sunstudio12.1/bin/lint

784 sparcv9_CC=         $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
785 $(__GNUC64)sparcv9_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
786 sparcv9_CCC=        $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
787 $(__GNUC64)sparcv9_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
788 sparcv9_CPP=        /usr/ccs/lib/cpp
789 sparcv9_AS=         /usr/ccs/bin/as -xregsym=no
790 sparcv9_LD=         /usr/ccs/bin/ld
791 sparcv9_LINT=       $(SPRO_ROOT)/sunstudio12.1/bin/lint

793 i386_CC=            $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
794 $(__GNUC)i386_CC=   $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
795 i386_CCC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
796 $(__GNUC)i386_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
797 i386_CPP=           /usr/ccs/lib/cpp
798 i386_AS=            /usr/ccs/bin/as
799 $(__GNUC)i386_AS=   $(ONBLD_TOOLS)/bin/$(MACH)/aw
800 i386_LD=            /usr/ccs/bin/ld
801 i386_LINT=          $(SPRO_ROOT)/sunstudio12.1/bin/lint

803 amd64_CC=           $(ONBLD_TOOLS)/bin/$(MACH)/cw -_cc
804 $(__GNUC64)amd64_CC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_gcc
805 amd64_CCC=          $(ONBLD_TOOLS)/bin/$(MACH)/cw -_CC
806 $(__GNUC64)amd64_CCC= $(ONBLD_TOOLS)/bin/$(MACH)/cw -_g++
807 amd64_CPP=          /usr/ccs/lib/cpp
808 amd64_AS=           $(ONBLD_TOOLS)/bin/$(MACH)/aw
809 amd64_LD=          /usr/ccs/bin/ld
810 amd64_LINT=         $(SPRO_ROOT)/sunstudio12.1/bin/lint

812 NATIVECC=          $( $(NATIVE_MACH)_CC )
813 NATIVECCC=         $( $(NATIVE_MACH)_CCC )
814 NATIVECPP=         $( $(NATIVE_MACH)_CPP )
815 NATIVEAS=          $( $(NATIVE_MACH)_AS )
816 NATVELD=           $( $(NATIVE_MACH)_LD )
817 NATVELINT=         $( $(NATIVE_MACH)_LINT )

819 #
820 # Makefile.master.64 overrides these settings
821 #
822 CC=                 $(NATIVECC)
823 CCC=                $(NATIVECCC)
824 CPP=                $(NATIVECPP)
825 AS=                 $(NATIVEAS)
826 LD=                 $(NATVELD)
827 LINT=               $(NATVELINT)

829 # The real compilers used for this build
830 CW_CC_CMD=          $(CC) -_compiler
831 CW_CCC_CMD=         $(CCC) -_compiler
832 REAL_CC=            $(CW_CC_CMD:sh)
833 REAL_CCC=          $(CW_CCC_CMD:sh)

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

838 BDIRECT=           -Bdirect
839 BDYNAMIC=          -Bdynamic
840 BLOCAL=            -Blocal
841 BNODIRECT=         -Bnodirect
842 BREDUCE=           -Breduce
843 BSTATIC=           -Bstatic

```

```

845 ZDEFS=             -zdefs
846 ZDIRECT=           -zdirect
847 ZIGNORE=           -zignore
848 ZINITFIRST=        -zinitfirst
849 ZINTERPOSE=        -zinterpose
850 ZLAZYLOAD=         -zlazyload
851 ZLOADFLTR=         -zloadfltr
852 ZMULDEFS=          -zmuldefs
853 ZNODEFAULTLIB=     -znodefautlib
854 ZNODEFS=           -znodelibs
855 ZNODELETE=         -znodelite
856 ZNODLOPEN=        -znodlopen
857 ZNODUMP=           -znodump
858 ZNOLAZYLOAD=       -znolazyload
859 ZNOLDYNAMSYM=      -znoldynsym
860 ZNORELOC=          -znoreloc
861 ZNOVERSION=        -znoversion
862 ZRECORD=           -zrecord
863 ZREDLOCSYM=        -zredlocsymb
864 ZTEXT=             -ztext
865 ZVERBOSE=          -zverbose

867 GSHARED=           -G
868 CCMT=              -mt

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

873 sparc_C_PICFLAGS =   -K pic
874 sparcv9_C_PICFLAGS = -K pic
875 i386_C_PICFLAGS =    -K pic
876 amd64_C_PICFLAGS =   -K pic
877 C_PICFLAGS =         $( $(MACH)_C_PICFLAGS )
878 C_PICFLAGS64 =       $( $(MACH64)_C_PICFLAGS )

880 sparc_C_BIGPICFLAGS = -K PIC
881 sparcv9_C_BIGPICFLAGS = -K PIC
882 i386_C_BIGPICFLAGS = -K PIC
883 amd64_C_BIGPICFLAGS = -K PIC
884 C_BIGPICFLAGS =      $( $(MACH)_C_BIGPICFLAGS )
885 C_BIGPICFLAGS64 =    $( $(MACH64)_C_BIGPICFLAGS )

887 # CC requires there to be no space between '-K' and 'pic' or 'PIC'.
888 sparc_CC_PICFLAGS =  -Kpic
889 sparcv9_CC_PICFLAGS = -Kpic
890 i386_CC_PICFLAGS =   -Kpic
891 amd64_CC_PICFLAGS =  -Kpic
892 CC_PICFLAGS =        $( $(MACH)_CC_PICFLAGS )
893 CC_PICFLAGS64 =      $( $(MACH64)_CC_PICFLAGS )

895 AS_PICFLAGS=         $(C_PICFLAGS)
896 AS_BIGPICFLAGS=      $(C_BIGPICFLAGS)

898 #
899 # Default label for CTF sections
900 #
901 CTFCVTFLAGS=         -i -L VERSION
902 $(SRCDBGBLD)CTFCVTFLAGS += -g

904 #
905 # Override to pass module-specific flags to ctmerge. Currently used only by
906 # krtld to turn on fuzzy matching, and source-level debugging to inhibit
907 # stripping.
908 #
909 CTFMRGFLAGS=
910 $(SRCDBGBLD)CTFMRGFLAGS += -g

```

```

913 CTFCONVERT_O          = $(CTFCONVERT) $(CTFCVTFMFLAGS) $@

915 ELFSIGN_O=            $(TRUE)
916 ELFSIGN_CRYPTO=      $(ELFSIGN_O)
917 ELFSIGN_OBJECT=      $(ELFSIGN_O)

919 # Rules (normally from make.rules) and macros which are used for post
920 # processing files. Normally, these do stripping of the comment section
921 # automatically.
922 #   RELEASE_CM:        Should be edited to reflect the release.
923 #   POST_PROCESS_O:   Post-processing for '.o' files.
924 #   POST_PROCESS_A:   Post-processing for '.a' files (currently null).
925 #   POST_PROCESS_SO:  Post-processing for '.so' files.
926 #   POST_PROCESS:     Post-processing for executable files (no suffix).
927 # Note that these macros are not completely generalized as they are to be
928 # used with the file name to be processed following.
929 #
930 # It is left as an exercise to Release Engineering to embellish the generation
931 # of the release comment string.
932 #
933 #   If this is a standard development build:
934 #       compress the comment section (mcs -c)
935 #       add the standard comment (mcs -a $(RELEASE_CM))
936 #       add the development specific comment (mcs -a $(DEV_CM))
937 #
938 #   If this is an installation build:
939 #       delete the comment section (mcs -d)
940 #       add the standard comment (mcs -a $(RELEASE_CM))
941 #       add the development specific comment (mcs -a $(DEV_CM))
942 #
943 #   If this is an release build:
944 #       delete the comment section (mcs -d)
945 #       add the standard comment (mcs -a $(RELEASE_CM))
946 #
947 # The following list of macros are used in the definition of RELEASE_CM
948 # which is used to label all binaries in the build:
949 #
950 #   RELEASE           Specific release of the build, eg: 5.2
951 #   RELEASE_MAJOR    Major version number part of $(RELEASE)
952 #   RELEASE_MINOR    Minor version number part of $(RELEASE)
953 #   VERSION           Version of the build (alpha, beta, Generic)
954 #   PATCHID          If this is a patch this value should contain
955 #                   the patchid value (eg: "Generic 100832-01"), otherwise
956 #                   it will be set to $(VERSION)
957 #   RELEASE_DATE     Date of the Release Build
958 #   PATCH_DATE       Date the patch was created, if this is blank it
959 #                   will default to the RELEASE_DATE
960 #
961 RELEASE_MAJOR= 5
962 RELEASE_MINOR= 11
963 RELEASE=      $(RELEASE_MAJOR).$(RELEASE_MINOR)
964 VERSION=     SunOS Development
965 PATCHID=     $(VERSION)
966 RELEASE_DATE= release date not set
967 PATCH_DATE=  $(RELEASE_DATE)
968 RELEASE_CM=  "@$(POUND_SIGN)SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
969 DEV_CM=      "@$(POUND_SIGN)SunOS Internal Development: non-nightly build"

971 PROCESS_COMMENT=    @?${MCS} -c -a $(RELEASE_CM) -a $(DEV_CM)
972 $(STRIP_COMMENTS)PROCESS_COMMENT= @?${MCS} -d -a $(RELEASE_CM) -a $(DEV_CM)
973 $(RELEASE_BUILD)PROCESS_COMMENT=  @?${MCS} -d -a $(RELEASE_CM)

975 STRIP_STABS=       :
976 $(RELEASE_BUILD)STRIP_STABS=    $(STRIP) -x $@

```

```

977 $(SRCDBGBLD)STRIP_STABS=       :

979 POST_PROCESS_O=      $(PROCESS_COMMENT) $@
980 POST_PROCESS_A=      $(PROCESS_COMMENT) $@
981 POST_PROCESS_SO=     $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
982                     $(ELFSIGN_OBJECT)
983 POST_PROCESS=        $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
984                     $(ELFSIGN_OBJECT)

986 #
987 # chk4ubin is a tool that inspects a module for a symbol table
988 # ELF section size which can trigger an OBP bug on older platforms.
989 # This problem affects only specific sun4u bootable modules.
990 #
991 CHK4UBIN=            $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
992 CHK4UBINFLAGS=      $(ONBLD_TOOLS)/bin/$(MACH)/chk4ubin
993 CHK4UBINARY=        $(CHK4UBIN) $(CHK4UBINFLAGS) $@

995 #
996 # PKGARCHIVE specifies the default location where packages should be
997 # placed if built.
998 #
999 $(RELEASE_BUILD)PKGARCHIVESUFFIX= -nd
1000 PKGARCHIVE=$(SRC)/../../packages/$(MACH)/nightly$(PKGARCHIVESUFFIX)

1002 #
1003 # The repositories will be created with these publisher settings. To
1004 # update an image to the resulting repositories, this must match the
1005 # publisher name provided to "pkg set-publisher."
1006 #
1007 PKGPUBLISHER_REDIST= on-nightly
1008 PKGPUBLISHER_NONREDIST= on-extra

1010 #   Default build rules which perform comment section post-processing.
1011 #
1012 .c:
1013     $(LINK.c) -o $@ $< $(LDLIBS)
1014     $(POST_PROCESS)
1015 .c.o:
1016     $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
1017     $(POST_PROCESS_O)
1018 .c.a:
1019     $(COMPILE.c) -o $% $<
1020     $(PROCESS_COMMENT) $%
1021     $(AR) $(ARFLAGS) $@ $%
1022     $(RM) $%
1023 .s.o:
1024     $(COMPILE.s) -o $@ $<
1025     $(POST_PROCESS_O)
1026 .s.a:
1027     $(COMPILE.s) -o $% $<
1028     $(PROCESS_COMMENT) $%
1029     $(AR) $(ARFLAGS) $@ $%
1030     $(RM) $%
1031 .cc:
1032     $(LINK.cc) -o $@ $< $(LDLIBS)
1033     $(POST_PROCESS)
1034 .cc.o:
1035     $(COMPILE.cc) $(OUTPUT_OPTION) $<
1036     $(POST_PROCESS_O)
1037 .cc.a:
1038     $(COMPILE.cc) -o $% $<
1039     $(AR) $(ARFLAGS) $@ $%
1040     $(PROCESS_COMMENT) $%
1041     $(RM) $%
1042 .y:

```

```

1043      $(YACC.y) $<
1044      $(LINK.c) -o $@ y.tab.c $(LDLIBS)
1045      $(POST_PROCESS)
1046      $(RM) y.tab.c
1047 .y.o:
1048      $(YACC.y) $<
1049      $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
1050      $(POST_PROCESS_O)
1051      $(RM) y.tab.c
1052 .l:
1053      $(RM) $*.c
1054      $(LEX.l) $< > $*.c
1055      $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
1056      $(POST_PROCESS)
1057      $(RM) $*.c
1058 .l.o:
1059      $(RM) $*.c
1060      $(LEX.l) $< > $*.c
1061      $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
1062      $(POST_PROCESS_O)
1063      $(RM) $*.c

1065 .bin.o:
1066      $(COMPILE.b) -o $@ $<
1067      $(POST_PROCESS_O)

1069 .java.class:
1070      $(COMPILE.java) $<

1072 # Bourne and Korn shell script message catalog build rules.
1073 # We extract all gettext strings with sed(1) (being careful to permit
1074 # multiple gettext strings on the same line), weed out the dups, and
1075 # build the catalogue with awk(1).

1077 .sh.po .ksh.po:
1078      $(SED) -n -e ":a" \
1079      -e "h" \
1080      -e "s/.*gettext *\([^\"]*\).*\|/p" \
1081      -e "x" \
1082      -e "s/\(.*\)gettext *\([^\"]*\).*\|/1\2/" \
1083      -e "t a" \
1084      $< | sort -u | awk '{ print "msgid\t" $$0 "\nmsgstr" }' > $@

1086 #
1087 # Python and Perl executable and message catalog build rules.
1088 #
1089 .SUFFIXES: .pl .pm .py .pyc

1091 .pl:
1092      $(RM) $@;
1093      $(SED) -e "s@TEXT_DOMAIN@"$(TEXT_DOMAIN)@" $< > $@;
1094      $(CHMOD) +x $@

1096 .py:
1097      $(RM) $@; $(CAT) $< > $@; $(CHMOD) +x $@

1099 .py.pyc:
1100      $(RM) $@
1101      $(PYTHON) -mpy_compile $<
1102      @[ $(<)c = $@ ] || $(MV) $(<)c $@

1104 .py.po:
1105      $(GNUXGETTEXT) $(GNUXGETFLAGS) -d $(<F:%.py=%) $< ;

1107 .pl.po .pm.po:
1108      $(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;

```

```

1109      $(RM) $@ ;
1110      $(SED) "/^domain/d" < $(<F).po > $@ ;
1111      $(RM) $(<F).po

1113 #
1114 # When using xgettext, we want messages to go to the default domain,
1115 # rather than the specified one. This special version of the
1116 # COMPILE.cpp macro effectively prevents expansion of TEXT_DOMAIN,
1117 # causing xgettext to put all messages into the default domain.
1118 #
1119 CPPFORPO=$(COMPILE.cpp:"$(TEXT_DOMAIN)"=TEXT_DOMAIN)

1121 .c.i:
1122      $(CPPFORPO) $< > $@

1124 .h.i:
1125      $(CPPFORPO) $< > $@

1127 .y.i:
1128      $(YACC) -d $<
1129      $(CPPFORPO) y.tab.c > $@
1130      $(RM) y.tab.c

1132 .l.i:
1133      $(LEX) $<
1134      $(CPPFORPO) lex.yy.c > $@
1135      $(RM) lex.yy.c

1137 .c.po:
1138      $(CPPFORPO) $< > $<.i
1139      $(BUILD.po)

1141 .y.po:
1142      $(YACC) -d $<
1143      $(CPPFORPO) y.tab.c > $<.i
1144      $(BUILD.po)
1145      $(RM) y.tab.c

1147 .l.po:
1148      $(LEX) $<
1149      $(CPPFORPO) lex.yy.c > $<.i
1150      $(BUILD.po)
1151      $(RM) lex.yy.c

1153 #
1154 # Rules to perform stylistic checks
1155 #
1156 .SUFFIXES: .x .xml .check .xmlchk

1158 .h.check:
1159      $(DOT_H_CHECK)

1161 .x.check:
1162      $(DOT_X_CHECK)

1164 .xml.xmlchk:
1165      $(MANIFEST_CHECK)

1167 #
1168 # Include rules to render automated sccs get rules "safe".
1169 #
1170 include $(SRC)/Makefile.noget

```

```

*****
10856 Thu Aug 15 11:59:20 2013
new/usr/src/cmd/Makefile
4027 remove CLOSED_BUILD
*****
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) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
23 # Copyright 2010 Nexenta Systems, Inc. All rights reserved.
24 # Copyright 2011 Joyent, Inc. All rights reserved.
25 # Copyright (c) 2012 by Delphix. All rights reserved.
26 # Copyright (c) 2013 DEY Storage Systems, Inc. All rights reserved.

28 include ../Makefile.master

30 #
31 # Note that the commands 'agents', 'lp', 'perl', and 'man' are first in
32 # the list, violating alphabetical order. This is because they are very
33 # long-running and should be given the most wall-clock time for a
34 # parallel build.
35 #
36 # Commands in the FIRST_SUBDIRS list are built before starting the build
37 # of other commands. Currently this includes only 'isaexec' and
38 # 'platexec'. This is necessary because $(ROOT)/usr/lib/isaexec or
39 # $(ROOT)/usr/lib/platexec must exist when some other commands are built
40 # because their 'make install' creates a hard link to one of them.
41 #
42 # Commands are listed one per line so that TeamWare can auto-merge most
43 # changes.
44 #

46 FIRST_SUBDIRS= \
47 isaexec \
48 platexec

50 COMMON_SUBDIRS= \
51 allocate \
52 availdevs \
53 lp \
54 perl \
55 man \
56 Adm \
57 abi \
58 adbgen \
59 acct \
60 acctadm \
61 arch

```

```

62 asa \
63 ast \
64 audio \
65 auths \
66 autopush \
67 avs \
68 awk \
69 awk_xpg4 \
70 backup \
71 banner \
72 bart \
73 basename \
74 bc \
75 bdiff \
76 beadm \
77 bfs \
78 bnu \
79 boot \
80 busstat \
81 cal \
82 calendar \
83 captinfo \
84 cat \
85 cdrw \
86 cfgadm \
87 checkeq \
88 checknr \
89 chgrp \
90 chmod \
91 chown \
92 chroot \
93 clear \
94 clinfo \
95 cmd-crypto \
96 cmd-inet \
97 col \
98 compress \
99 consadm \
100 coreadm \
101 cpio \
102 cpc \
103 cron \
104 crypt \
105 csh \
106 csplit \
107 ctrun \
108 ctstat \
109 ctwatch \
110 datadm \
111 date \
112 dc \
113 dd \
114 deroff \
115 devfsadm \
116 syseventd \
117 devctl \
118 devinfo \
119 devmgmt \
120 devprop \
121 dfs.cmds \
122 diff \
123 diff3 \
124 diffmk \
125 dircmp \
126 dirname \
127 dis \

```

new/usr/src/cmd/Makefile

```
128      diskmgtd  //
129      dispadmin //
130      dladm     //
131      dlstat    //
132      dmesg     //
133      dodatadm  //
134      dtrace    //
135      du        //
136      dumpadm   //
137      dumpcs    //
138      echo      //
139      ed        //
140      eeprom    //
141      egrep     //
142      eject     //
143      emul64ioct1 //
144      enhance   //
145      env       //
146      eqn       //
147      expand    //
148      expr      //
149      exstr     //
150      factor    //
151      false     //
152      fcinfo    //
153      fcoesvc   //
154      fdetach   //
155      fdformat  //
156      fdisk     //
157      filesync  //
158      fgrep     //
159      file      //
160      filebench //
161      find      //
162      flowadm   //
163      flowstat  //
164      fm        //
165      fmt       //
166      fmthard   //
167      fmsg      //
168      fold      //
169      format    //
170      fs.d      //
171      fstyp     //
172      fuser     //
173      fwflash   //
174      gcore     //
175      gencat    //
176      geniconvtbl //
177      genmsg    //
178      getconf   //
179      getdevpolicy //
180      getent    //
181      getfacl  //
182      getmajor  //
183      getopt    //
184      gettext   //
185      gettxt    //
186      grep      //
187      grep_xpg4 //
188      groups    //
189      grpck     //
190      gss       //
191      hal       //
192      halt      //
193      head      //
```

new/usr/src/cmd/Makefile

```
194      hostid   //
195      hostname //
196      hotplug   //
197      hotplugd //
198      hwdata    //
199      ibd_upgrade //
200      id        //
201      idmap     //
202      infocmp   //
203      init      //
204      initpkg   //
205      install.d //
206      intrd     //
207      intrstat  //
208      ipcrm     //
209      ipcs      //
210      ipf       //
211      isainfo   //
212      isalist   //
213      itutools  //
214      iscsiadm  //
215      iscsid    //
216      iscsitsvc //
217      isns      //
218      itadm     //
219      java      //
220      kbd       //
221      keyserve  //
222      killall   //
223      krb5      //
224      ksh       //
225      kvmstat   //
226      last      //
227      lastcomm  //
228      latencytop //
229      ldap      //
230      ldapcachemgr //
231      lgrpinfo  //
232      line      //
233      link      //
234      dlmgmt    //
235      listen    //
236      loadkeys  //
237      locale    //
238      localedef //
239      lockstat  //
240      locator   //
241      lofiadm   //
242      logadm    //
243      logger    //
244      login     //
245      logins    //
246      look      //
247      ls        //
248      luxadm    //
249      lvm       //
250      mach      //
251      machid    //
252      mail      //
253      mailx     //
254      makekey   //
255      mdb       //
256      msg       //
257      mkdir     //
258      mkfifo    //
259      mkfile    //
```

new/usr/src/cmd/Makefile

```

260      mkmsgs      //
261      mknod       //
262      mkpwdict    //
263      mktemp      //
264      modload     //
265      more        //
266      mpathadm    //
267      msgfmt      //
268      msgid       //
269      mt          //
270      mv          //
271      mvdir       //
272      ndmpadm     //
273      ndmpd       //
274      ndmpstat    //
275      netadm      //
276      netfiles    //
277      newform     //
278      newgrp      //
279      news        //
280      newtask     //
281      nice        //
282      nl          //
283      nlsadmin    //
284      nohup       //
285      nsadmin     //
286      nscd        //
287      oamuser     //
288      oawk        //
289      od          //
290      pack        //
291      pagesize    //
292      passmgmt    //
293      passwd      //
294      pathchk     //
295      pbind       //
296      pcidr       //
297      pcitool     //
298      pfexec      //
299      pfexecd     //
300      pginfo      //
301      pgstat      //
302      pgrep       //
303      picl        //
304      plimit      //
305      policykit   //
306      pools       //
307      power       //
308      powertop    //
309      ppgsz       //
310      pg          //
311      plockstat   //
312      pr          //
313      prctl       //
314      print       //
315      printf      //
316      priocntl    //
317      profiles    //
318      projadd     //
319      projects    //
320      prstat      //
321      prtconf     //
322      prtdiag     //
323      prtvtoc     //
324      ps          //
325      psradm     //

```

5

new/usr/src/cmd/Makefile

```

326      psrinfo    //
327      psrset     //
328      ptools     //
329      pwck       //
330      pwconv     //
331      pwd        //
332      pyzfs      //
333      raidctl    //
334      ramdiskadm //
335      rcap       //
336      rcm_daemon //
337      rctladm    //
338      refer      //
339      regcmp     //
340      renice     //
341      rexd       //
342      rm         //
343      rmdir      //
344      rmformat   //
345      rmmount    //
346      rmt        //
347      rmvolmgr   //
348      roles      //
349      rpcbind    //
350      rpcgen     //
351      rpcinfo    //
352      rpcsvc     //
353      runat      //
354      sa         //
355      saf        //
356      sasinfo    //
357      savecore   //
358      sbdadm     //
359      script     //
360      scsi       //
361      sdiff      //
362      sdpadm     //
363      sed        //
364      sendmail   //
365      setfacl    //
366      setmnt     //
367      setpgrp    //
368      setuname   //
369      sgs        //
370      sh         //
371      shcomp     //
372      sbios      //
373      smbserver  //
374      smserverd  //
375      soelim     //
376      sort       //
377      spell      //
378      split      //
379      sqlite     //
380      srchtxt    //
381      srptadm    //
382      srptsvc    //
383      ssh        //
384      stat       //
385      stmfadm    //
386      stmfproxy  //
387      stmfsvc    //
388      stmsboot   //
389      streams    //
390      strings    //
391      su         //

```

6

new/usr/src/cmd/Makefile

```

392     sulogin      \
393     sunpc       \
394     svc         \
395     svr4pkg    \
396     swap       \
397     sync       \
398     sysdef     \
399     syseventadm \
400     syslogd    \
401     tabs       \
402     tail       \
403     tar        \
404     tbl        \
405     tcopy      \
406     tcpd       \
407     terminfo   \
408     th_tools   \
409     tic        \
410     time       \
411     tip        \
412     tnf        \
413     touch      \
414     tput       \
415     tr         \
416     trapstat  \
417     troff      \
418     true       \
419     truss      \
420     tsol       \
421     tty        \
422     ttymon     \
423     tzreload   \
424     uadmin     \
425     ul         \
426     uname      \
427     units      \
428     unlink     \
429     unpack     \
430     userattr   \
431     users      \
432     utmp_update \
433     utmpd      \
434     valtools   \
435     vgrind     \
436     vi         \
437     volcheck   \
438     volrmmount \
439     vrrpadm    \
440     vscan      \
441     vt         \
442     w          \
443     wall       \
444     which      \
445     who        \
446     whodo     \
447     wracct    \
448     write      \
449     wusbadm    \
450     xargs      \
451     xstr       \
452     yes        \
453     ypcmd     \
454     yppasswd   \
455     zdb        \
456     zdump     \
457     zfs        \

```

7

new/usr/src/cmd/Makefile

```

458     zhack      \
459     zic        \
460     zinject    \
461     zlogin     \
462     zoneadm    \
463     zoneadmd   \
464     zonecfg    \
465     zonename   \
466     zpool      \
467     zlook      \
468     zonestat   \
469     zstreamdump \
470     ztest      \

472 $(CLOSED_BUILD)COMMON_SUBDIRS += \
473     $(CLOSED)/cmd/iconv \
474     $(CLOSED)/cmd/ksh \
475     $(CLOSED)/cmd/localedef \
476     $(CLOSED)/cmd/more_xpg4 \
477     $(CLOSED)/cmd/mtst \
478     $(CLOSED)/cmd/od \
479     $(CLOSED)/cmd/patch \
480     $(CLOSED)/cmd/pax \
481     $(CLOSED)/cmd/printf \
482     $(CLOSED)/cmd/sed \
483     $(CLOSED)/cmd/sed_xpg4 \

472 i386_SUBDIRS= \
473     acpihpd \
474     addbadsec \
475     biosdev \
476     diskscan \
477     lms \
478     ntfsprogs \
479     parted \
480     rtc \
481     ucodeadm \
482     xvm \

484 sparc_SUBDIRS= \
485     cvcd \
486     dcs \
487     device_remap \
488     drd \
489     fruadm \
490     ldmad \
491     oplhpd \
492     prtdscp \
493     prtfru \
494     scadm \
495     sckmd \
496     sf880drd \
497     virtinfo \
498     vntsd \

500 #
501 # Commands that are messaged. Note that 'lp' and 'man' come first
502 # (see previous comment about 'lp' and 'man').
503 #
504 MSGSUBDIRS= \
505     lp \
506     man \
507     abi \
508     acctadm \
509     allocate \
510     asa \

```

8

new/usr/src/cmd/Makefile

```

511 audio //
512 audit //
513 auditconfig //
514 auditd //
515 auditrecord //
516 auditset //
517 auths //
518 autopush //
519 avs //
520 awk //
521 awk_xpg4 //
522 backup //
523 banner //
524 bart //
525 basename //
526 beadm //
527 bnu //
528 busstat //
529 cal //
530 cat //
531 cdrw //
532 cfgadm //
533 checkeq //
534 checknr //
535 chgrp //
536 chmod //
537 chown //
538 cmd-crypto //
539 cmd-inet //
540 col //
541 compress //
542 consadm //
543 coreadm //
544 cpio //
545 cpc //
546 cron //
547 csh //
548 csplit //
549 ctrun //
550 ctstat //
551 ctwatch //
552 datadm //
553 date //
554 dc //
555 dcs //
556 dd //
557 deroff //
558 devfsadm //
559 dfs.cmds //
560 diff //
561 diffmk //
562 dladm //
563 dlstat //
564 du //
565 dumphcs //
566 ed //
567 eject //
568 env //
569 eqn //
570 expand //
571 expr //
572 fcinfo //
573 fgrep //
574 file //
575 filesync //
576 find //

```

9

new/usr/src/cmd/Makefile

```

577 flowadm //
578 flowstat //
579 fm //
580 fold //
581 fs.d //
582 fwflash //
583 geniconvtbl //
584 genmsg //
585 getconf //
586 getent //
587 gettext //
588 gettxt //
589 grep //
590 grep_xpg4 //
591 grpck //
592 gss //
593 halt //
594 head //
595 hostname //
596 hotplug //
597 id //
598 idmap //
599 isaexec //
600 iscsiadm //
601 iscsid //
602 isns //
603 itadm //
604 kbd //
605 krb5 //
606 ksh //
607 last //
608 ldap //
609 ldapcachemgr //
610 lgrpinfo //
611 locale //
612 lofiadm //
613 logadm //
614 logger //
615 logins //
616 ls //
617 luxadm //
618 lvm //
619 mailx //
620 mesg //
621 mkdir //
622 mkpdict //
623 mktemp //
624 more //
625 mpathadm //
626 msgfmt //
627 mv //
628 ndmpadm //
629 ndmpstat //
630 newgrp //
631 newtask //
632 nice //
633 nohup //
634 oawk //
635 pack //
636 passwd //
637 passmgmt //
638 pathchk //
639 pexec //
640 pg //
641 pgrep //
642 picl //

```

10

```

643     pools           \
644     power           \
645     pr              \
646     praudit        \
647     print           \
648     profiles        \
649     projadd         \
650     projects        \
651     prstat          \
652     prtdiag         \
653     ps              \
654     psrinfo         \
655     ptools          \
656     pwconv          \
657     pwd             \
658     pyzfs           \
659     raidctl         \
660     ramdiskadm     \
661     rcap            \
662     rcm_daemon     \
663     refer           \
664     regcmp          \
665     renice          \
666     roles           \
667     rm              \
668     rmdir           \
669     rmformat        \
670     rmmount         \
671     rmvolmgr       \
672     sasinfo         \
673     sbdadm          \
674     scadm           \
675     script          \
676     scsi            \
677     sdiff           \
678     sdpadm          \
679     sgs             \
680     sh              \
681     shcomp          \
682     smbshr         \
683     sort            \
684     split           \
685     srptadm         \
686     ssh             \
687     stat            \
688     stmfadm         \
689     stmsboot        \
690     strings         \
691     su              \
692     svc             \
693     svr4pkg         \
694     swap            \
695     syseventadm    \
696     syseventd      \
697     tabs            \
698     tar             \
699     tbl             \
700     time            \
701     tnf             \
702     touch           \
703     tput            \
704     troff           \
705     tsol            \
706     tty             \
707     ttymon          \
708     tzreload        \

```

```

709     ul              \
710     uname           \
711     units           \
712     unlink          \
713     unpack          \
714     userattr        \
715     valtools        \
716     vgrind          \
717     vi              \
718     volcheck        \
719     volrmmount     \
720     vrrpadm         \
721     vscan           \
722     w               \
723     who             \
724     whodo           \
725     wracct          \
726     write           \
727     wusbadm         \
728     xargs           \
729     yppasswd        \
730     zdump           \
731     zfs             \
732     zic             \
733     zlogin          \
734     zoneadm         \
735     zoneadm         \
736     zonecfg         \
737     zonename        \
738     zpool           \
739     zonestat        \

```

```

754 $(CLOSED_BUILD)MSGSUBDIRS += \
755     $(CLOSED)/cmd/iconv \
756     $(CLOSED)/cmd/ksh \
757     $(CLOSED)/cmd/localedef \
758     $(CLOSED)/cmd/more_xpg4 \
759     $(CLOSED)/cmd/od \
760     $(CLOSED)/cmd/patch \
761     $(CLOSED)/cmd/pax \
762     $(CLOSED)/cmd/printf \
763     $(CLOSED)/cmd/sed \
764     $(CLOSED)/cmd/sed_xpg4 \

```

```

741 sparc_MSGSUBDIRS= \
742     fruadm         \
743     prtdscp        \
744     prtfru         \
745     virtinfo       \
746     vntsd          \

```

```

748 i386_MSGSUBDIRS= \
749     ucodeadm       \

```

```

751 #
752 # commands that use dcgettext for localized time, LC_TIME
753 #
754 DCSUBDIRS= \
755     cal             \
756     cfgadm          \
757     diff            \
758     ls              \
759     pr              \
760     ps              \
761     tar             \
762     w               \

```

```

763     who           \
764     whodo         \
765     write

792 $(CLOSED_BUILD)DCSUBDIRS += \
793     $(CLOSED)/cmd/pax

767 #
768 # commands that belong only to audit.
769 #
770 AUDITSUBDIRS=
771     amt           \
772     audit         \
773     audit_warn    \
774     auditconfig   \
775     auditd        \
776     auditrecord   \
777     auditreduce   \
778     auditset      \
779     auditstat     \
780     praudit

782 #
783 # commands not owned by the systems group
784 #
785 BWOSDIRS=

788 all :=           TARGET = all
789 install :=       TARGET = install
790 clean :=         TARGET = clean
791 clobber :=       TARGET = clobber
792 lint :=          TARGET = lint
793 _msg :=          TARGET = _msg
794 _dc :=           TARGET = _dc

796 .KEEP_STATE:

798 SUBDIRS = $(COMMON_SUBDIRS) $($MACH)_SUBDIRS

800 .PARALLEL:      $(BWOSDIRS) $(SUBDIRS) $(MSGSUBDIRS) $(AUDITSUBDIRS)

802 all install clean clobber lint: $(FIRST_SUBDIRS) .WAIT $(SUBDIRS) \
803     $(AUDITSUBDIRS)

805 #
806 # Manifests cannot be checked in parallel, because we are using
807 # the global repository that is in $(SRC)/cmd/svc/seed/global.db.
808 # For this reason, to avoid .PARALLEL and .NO_PARALLEL conflicts,
809 # we spawn off a sub-make to perform the non-parallel 'make check'
810 #
811 check:
812     $(MAKE) -f Makefile.check check

814 #
815 # The .WAIT directive works around an apparent bug in parallel make.
816 # Evidently make was getting the target _msg vs. _dc confused under
817 # some level of parallelization, causing some of the _dc objects
818 # not to be built.
819 #
820 _msg: $(MSGSUBDIRS) $($MACH)_MSGSUBDIRS) .WAIT _dc

822 _dc: $(DCSUBDIRS)

824 #
825 # Dependencies

```

```

826 #
827 fs.d: fstyp
828 ksh:  shcomp isaexec
829 mdb:  terminfo
830 print: lp

832 $(FIRST_SUBDIRS) $(BWOSDIRS) $(SUBDIRS) $(AUDITSUBDIRS): FRC
833     @if [ -f $@/Makefile ]; then \
834         cd $@; pwd; $(MAKE) $(TARGET); \
835     else \
836         true; \
837     fi

839 FRC:

```

new/usr/src/cmd/cmd-crypto/Makefile

1

1429 Thu Aug 15 11:59:20 2013

new/usr/src/cmd/cmd-crypto/Makefile

4027 remove CLOSED_BUILD

```
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 # Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
22 #
```

```
24 include ../Makefile.cmd
```

```
26 SUBDIRS1 = \
27     cryptoadm \
28     decrypt \
29     digest \
30     elfsign \
31     kmfcfg \
32     pktool \
33     tpmadm
```

```
35 $(CLOSED_BUILD)SUBDIRS1 += \
36     $(CLOSED)/cmd/cmd-crypto/kcfd
```

```
35 SUBDIRS2 = \
36     etc \
37     scripts
```

```
42 $(CLOSED_BUILD)SUBDIRS2 += \
43     $(CLOSED)/cmd/cmd-crypto/etc
```

```
39 all:=          TARGET= all
40 install:=      TARGET= install
41 clean:=        TARGET= clean
42 clobber:=      TARGET= clobber
43 lint:=         TARGET= lint
44 _msg:=         TARGET= _msg
```

```
46 .KEEP_STATE:
```

```
48 all clean clobber lint _msg: $(SUBDIRS1) $($(_MACH)_SUBDIRS)
```

```
50 install: $(SUBDIRS1) $(SUBDIRS2) $($(_MACH)_SUBDIRS)
```

```
52 $(SUBDIRS1) $(SUBDIRS2) $($(_MACH)_SUBDIRS) : FRC
53     @cd $@; pwd; $(MAKE) $(MFLAGS) $(TARGET)
```

```
55 FRC:
```

```

*****
1929 Thu Aug 15 11:59:21 2013
new/usr/src/cmd/cmd-inet/usr.lib/Makefile
4027 remove CLOSED_BUILD
*****
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) 1996, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 SUBDIRS=      bridged dhcp dsvclockd ilbd in.chargend in.daytimed \
27              in.discardd in.echod in.dhcpd in.mpathd in.ndpd \
28              in.ripngd in.timed inetd mdnsd ncaconfd pppoe \
29              slpd vrrpd wanboot wpaad
30 #
31 MSGSUBDIRS=   dsvclockd ilbd in.dhcpd inetd ncaconfd vrrpd wanboot
32 #
33 include ../Makefile.cmd
34 include ./Makefile.lib
35 #
36 $(CLOSED_BUILD)SUBDIRS += \
37   $(CLOSED)/cmd/cmd-inet/usr.lib/ike-certutils \
38   $(CLOSED)/cmd/cmd-inet/usr.lib/in.iked
39 #
40 #
41 POFILES=      dsvclockd/dsvclockd.po in.dhcpd/in.dhcpd.po \
42              inetd/inetd.po ncaconfd/ncaconfd.po vrrpd/vrrpd.po \
43              wanboot/wanboot.po
44 POFILE=       usr.lib.po
45 #
46 all:=         TARGET= all
47 install:=     TARGET= install
48 clean:=       TARGET= clean
49 clobber:=     TARGET= clobber
50 lint:=        TARGET= lint
51 _msg:=        TARGET= _msg
52 #
53 .KEEP_STATE:
54 #
55 all clean clobber lint: $(SUBDIRS)
56 #
57 install: $(SUBDIRS)
58     -$(RM) $(ROOTLIBINET)/in.iked
59     -$(LN) $(ISAEEXEC) $(ROOTLIBINET)/in.iked
60 #
61 _msg: $(MSGSUBDIRS)

```

```

59 #
60 # The reason this rule checks for the existence of the
61 # Makefile is that some of the directories do not exist
62 # in our exportable source builds or in OpenSolaris.
63 #
64 $(SUBDIRS): FRC
65     @if [ -f $@/Makefile ]; then \
66         cd $@; pwd; $(MAKE) $(TARGET); \
67     else \
68         true; \
69     fi
70 #
71 FRC:

```

new/usr/src/cmd/fs.d/nfs/Makefile

1

1768 Thu Aug 15 11:59:21 2013

new/usr/src/cmd/fs.d/nfs/Makefile

4027 remove CLOSED_BUILD

```
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 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 # cmd/fs.d/nfs/Makefile
27 #
28 # cmd/fs.d/nfs is the directory of all nfs specific commands
29 # whose executable reside in $(INSDIR1) and $(INSDIR2).
30 #
```

```
32 include $(SRC)/Makefile.master
```

```
34 SUBDIR1=      exportfs nfsd rquotad \
35               statd nfsstat mountd dfshares \
36               nfsfind nfs4cbd share
37 SUBDIR2=      clear_locks umount showmount \
38               mount dmounts nfslog nfsmapid \
39               nfsref rp_basic
```

```
41 SUBDIR3=      etc svc
42 SUBDIRS=      $(SUBDIR1) $(SUBDIR2) $(SUBDIR3)
```

```
44 # for messaging catalog files
```

```
45 #
46 POFILES=      $(SUBDIR2:%=%/.po)
47 POFILE=       nfs.po
```

```
49 LOCKD=        $(CLOSED)/cmd/fs.d/nfs/lockd
50 $(CLOSED_BUILD)CLOSED_SUBDIR2= $(LOCKD)
51 $(CLOSED_BUILD)POFILES +=      $(LOCKD)/lockd.po
52 $(CLOSED_BUILD)SUBDIRS +=      $(CLOSED_SUBDIR2)
```

```
49 all:=         TARGET= all
50 install:=     TARGET= install
51 clean:=       TARGET= clean
52 clobber:=     TARGET= clobber
53 lint:=        TARGET= lint
54 catalog:=     TARGET= catalog
```

```
56 .KEEP_STATE:
```

new/usr/src/cmd/fs.d/nfs/Makefile

2

```
58 .PARALLEL:   $(SUBDIRS)
```

```
60 all install clean clobber lint: $(SUBDIRS)
```

```
62 catalog: $(SUBDIR2)
63 catalog: $(SUBDIR2) $(CLOSED_SUBDIR2)
64          $(RM) $(POFILE)
65          cat $(POFILES) > $(POFILE)
```

```
66 $(SUBDIRS): FRC
67          @cd $@; pwd; $(MAKE) $(TARGET)
```

```
69 FRC:
```

new/usr/src/cmd/fwflash/plugins/Makefile

1

1352 Thu Aug 15 11:59:21 2013

new/usr/src/cmd/fwflash/plugins/Makefile

4027 remove CLOSED_BUILD

```
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 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # cmd/fwflash/plugins
26 #
27 include $(SRC)/Makefile.master

29 COMMON_SUBDIRS= $(MACH)
30 $(CLOSED_BUILD)COMMON_SUBDIRS += $(CLOSED)/cmd/fwflash/plugins

31 SUBDIRS=      $(COMMON_SUBDIRS)

33 all :=          TARGET= all
34 install :=     TARGET= install
35 clean :=       TARGET= clean
36 clobber :=     TARGET= clobber
37 lint :=        TARGET= lint
38 _msg :=        TARGET= _msg
39 msg :=         TARGET= msg

42 .KEEP_STATE:

44 all clean clobber install lint msg _msg:          $(SUBDIRS)

47 $(SUBDIRS): FRC
48     @if [ -f $@/Makefile ]; then \
49         cd $@; pwd; $(MAKE) $(TARGET); \
50     else \
51         true; \
52     fi

54 FRC:
```

new/usr/src/cmd/mdb/intel/amd64/Makefile

1

1175 Thu Aug 15 11:59:21 2013

new/usr/src/cmd/mdb/intel/amd64/Makefile

4027 remove CLOSED_BUILD

```
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 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include ../../Makefile.common

28 MODULES = $(COMMON_MODULES_PROC) $(COMMON_MODULES_KVM) uhci

30 $(CLOSED_BUILD)MODULES += \
31     $(CLOSED_COMMON_MODULES_KVM:%=$(CLOSED)/cmd/mdb/intel/amd64/%)

30 SUBDIRS = mdb mdb_ks kmdb libstandctf libstand .WAIT $(MODULES)

32 include ../../Makefile.subdirs

34 .PARALLEL: $(SUBDIRS)

36 # inter-module dependencies
37 kmdb: libstand libstandctf mdb_ks
```


new/usr/src/cmd/mdb/intel/ia32/Makefile

1

1210 Thu Aug 15 11:59:21 2013

new/usr/src/cmd/mdb/intel/ia32/Makefile

4027 remove CLOSED_BUILD

```
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 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include ../../Makefile.common

28 MODULES = $(COMMON_MODULES_PROC) $(COMMON_MODULES_PROC_32BIT) \
29     $(COMMON_MODULES_KVM) uhci

31 $(CLOSED_BUILD)MODULES += \
32     $(CLOSED_COMMON_MODULES_KVM:%=$(CLOSED)/cmd/mdb/intel/ia32/%)

31 SUBDIRS = mdb mdb_ks kmdb libstandctf libstand .WAIT $(MODULES)

33 include ../../Makefile.subdirs

35 .PARALLEL: $(SUBDIRS)

37 # inter-module dependencies
38 kmdb: libstand libstandctf mdb_ks
```

new/usr/src/cmd/mdb/sparc/v9/Makefile

1

```
*****
1237 Thu Aug 15 11:59:21 2013
new/usr/src/cmd/mdb/sparc/v9/Makefile
4027 remove CLOSED_BUILD
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 #ident "%Z%M% %I% %E% SMI"

26 include ../../Makefile.common

28 MODULES = \
29     $(COMMON_MODULES_PROC) \
30     $(COMMON_MODULES_KVM) \
31     intr \
32     ssd

35 $(CLOSED_BUILD)MODULES += \
36     $(CLOSED_COMMON_MODULES_KVM:%=$(CLOSED)/cmd/mdb/sparc/v9/%)
37 $(CLOSED_BUILD)MODULES += \
38     $(CLOSED)/cmd/mdb/sparc/v9/isp

34 #
35 # a "$(MODULES): kmdb" rule would seem to do the trick but, for some reason,
36 # it serializes $(MODULES).
37 #
38 SUBDIRS = mdb mdb_ks kmdb libstandctf libstand .WAIT $(MODULES)

40 include ../../Makefile.subdirs

42 .PARALLEL: $(SUBDIRS)
```

new/usr/src/cmd/tsol/Makefile

1

1735 Thu Aug 15 11:59:21 2013

new/usr/src/cmd/tsol/Makefile

4027 remove CLOSED_BUILD

```
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 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 #ident "%Z%M% %I% %E% SMI"
26 #
25 # cmd/tsol/Makefile
26 #
```

28 include ../Makefile.cmd

```
30 SUBDIRS = \
31     atohexlabel \
32     getlabel \
33     demo \
34     misc \
35     zones \
36     labeld \
37     updatehome \
38     plabel \
39     setlabel \
40     tnchkdb \
41     tninfo \
42     tnctl \
43     tnd \
44     tsol-zones \
45     getzonepath \
46     hextoalabel \
47     lslabels
```

```
50 MSGSUBDIRS = \
51     getlabel \
52     setlabel \
53     tnchkdb \
54     tninfo \
55     tnctl \
56     tnd \
57     getzonepath
```

61 \$(CLOSED_BUILD)SUBDIRS += \$(CLOSED)/cmd/tsol/chk_encodings \

new/usr/src/cmd/tsol/Makefile

2

62 \$(CLOSED)/cmd/tsol/labeld

64 \$(CLOSED_BUILD)CLOSED_MSGSUBDIRS = \$(CLOSED)/cmd/tsol/chk_encodings \

65 \$(CLOSED)/cmd/tsol/labeld

```
59 #
60 # for messaging catalog
61 #
62 POFILE= tsol-cmd.po
63 POFILES= $(MSGSUBDIRS:%=%/.po)
```

```
65 all := TARGET = all
66 install := TARGET = install
67 clean := TARGET = clean
68 clobber := TARGET = clobber
69 lint := TARGET = lint
70 _msg := TARGET = _msg
```

72 .KEEP_STATE:

74 .PARALLEL: \$(SUBDIRS)

76 all install clean clobber lint: \$(SUBDIRS)

```
78 $(POFILE): $(MSGSUBDIRS) $(CLOSED_MSGSUBDIRS)
79     $(RM) $(POFILE)
80     $(CAT) $(POFILES) > $(POFILE)
```

```
82 $(SUBDIRS): FRC
83     @cd $@; pwd; $(MAKE) $(TARGET)
```

85 FRC:

87 include Makefile.targ

```

*****
13386 Thu Aug 15 11:59:22 2013
new/usr/src/lib/Makefile
4027 remove CLOSED_BUILD
*****
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 2011 Nexenta Systems, Inc. All rights reserved.
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright (c) 2012, Joyent, Inc. All rights reserved.

27 include ../Makefile.master

29 # Note that libcurls installs commands along with its library.
30 # This is a minor bug which probably should be fixed.
31 # Note also that a few extra libraries are kept in cmd source.
32 #
33 # Certain libraries are linked with, hence depend on, other libraries.
34 #
35 # Although we have historically used .WAIT to express dependencies, it
36 # reduces the amount of parallelism and thus lengthens the time it
37 # takes to build the libraries. Thus, we now require that any new
38 # libraries explicitly call out their dependencies. Eventually, all
39 # the library dependencies will be called out explicitly. See
40 # "Library interdependencies" near the end of this file.
41 #
42 # Aside from explicit dependencies (and legacy .WAITs), all libraries
43 # are built in parallel.
44 #
45 .PARALLEL:

47 #
48 # The $(CLOSED_BUILD) additions to SUBDIRS & MSGSUBDIRS are unfortunate,
49 # but required due to the "dependencies" of using .WAIT to barrier the
50 # parallel dmake builds. once 4631488 has been fixed, they can be
51 # consolidated into one $(CLOSED_BUILD)SUBDIRS += (all closed libs) as
52 # shown in HDRSUBDIRS
53 #
47 SUBDIRS= \
48     common                .WAIT \
49     ../cmd/sgs/libconv    .WAIT \
50     ../cmd/sgs/libdl      .WAIT

52 SUBDIRS += \
53     libc                .WAIT \
54     ../cmd/sgs/libelf    .WAIT

```

```

55     c_synonyms           \
56     libmd                 \
57     libmd5                \
58     librms                \
59     libmp                 .WAIT \
60     libnsl                \
61     libsecdb              .WAIT \
62     librpcsvc             \
63     libsocket             .WAIT \
64     libsctp               \
65     libsip                \
66     libcommputil         \
67     libresolv             \
68     libresolv2           .WAIT \
69     libw                  .WAIT \
70     libintl               .WAIT \
71     ../cmd/sgs/librtld_db \
72     libaio                \
73     libast                \
74     libdll                \
75     libcmd               \
76     libshell             \
77     libsum                \
78     librt                \
79     libadm               \
80     libctf                \
81     libdtrace            \
82     libdtrace_jni        \
83     libcurls             \
84     libtermcap           \
85     libgen                \
86     libgss                \
87     libpam               \
88     libuuid              \
89     libthread            \
90     libpthread           .WAIT \
91     libslp               \
92     libbsdmalloc         \
93     libdoor              \
94     libdevinfo           \
95     libdladm             \
96     libdlpi              \
97     libeti               \
98     libcrypt             \
99     libdns_sd            \
100    libefi                \
101    libfstyp              \
102    libwanboot            \
103    libwanbootutil       \
104    libcryptoutil        \
105    libinetutil          \
106    libipadm             \
107    libipmp              \
108    libiscsit            \
109    libkmf               \
110    libkstat              \
111    libkvm               \
112    liblm                 \
113    libmalloc            \
114    libmapmalloc         \
115    libmtmalloc          \
116    libnls               \
117    libnwam              \
118    libsbios             \
119    libtecla             \
120    libumem              \

```

new/usr/src/lib/Makefile

```

121      libnvpair      .WAIT  \
122      libexacct     \
123      libsasl       \
124      libldap5      \
125      libslldap     .WAIT  \
126      libbsm        \
127      libsys        \
128      libsysevent   \
129      libnisdb      \
130      libpool       \
131      libpp         \
132      libproc       \
133      libproject    \
134      libsendfile   \
135      nametoaddr    \
136      ncad_addr     \
137      hbaapi       \
138      smhba        \
139      sun_fc        \
140      sun_sas       \
141      gss_mechs/mech_krb5 .WAIT  \
142      libkrb5      .WAIT  \
143      krb5         .WAIT  \
144      libsmvfs      \
145      libfcoe       \
146      libsrpt       \
147      libstmf       \
148      libstmfproxy  \
149      libnsctl      \
150      libunistat    \
151      libdscfg      \
152      librdc        \
153      libinstzones \
154      libpkg        \
155      libpcidb     \
157 SUBDIRS += \
158      passwdutil   \
159      pam_modules  \
160      crypt_modules \
161      libadt_jni   \
162      abi          \
163      auditd_plugins \
164      libvolmgt    \
165      libdevice    \
166      libdevice    \
167      libdhcpsvc   \
168      libc_db      \
169      libndmp      \
170      libsec       \
171      libtnfprobe  \
172      libtnf       \
173      libtnfctl    \
174      libdhcpageant \
175      libdhcpcdu   \
176      libdhcputil  \
177      libxnet      \
178      libipsecutil \
185      libipsecutil \
186      $(CLOSED_BUILD)SUBDIRS += \
187      $(CLOSED)/lib/libike
188 SUBDIRS += \
179      nsswitch      \
180      print         \
181      libuutil      \
182      libscf        \

```

3

new/usr/src/lib/Makefile

```

183      libinetsvc   \
184      librestart   \
185      libsched     \
186      libelfsign   \
187      pkcs11       .WAIT  \
188      libpctx      .WAIT  \
189      libcpc       \
190      watchmalloc  \
191      extendedFILE \
192      madv         \
193      mpss         \
194      libdisasm    \
195      libwrap      \
196      libxcurses   \
197      libxcurses2  \
198      libbrand     .WAIT  \
199      libzonecfg   \
200      libzoneinfo  \
201      libzonestat  \
202      libtsnet     \
203      libtsol      \
204      gss_mechs/mech_spnego \
205      gss_mechs/mech_dummy \
206      gss_mechs/mech_dh    \
207      rpcsec_gss         \
208      libraidcfg        .WAIT  \
209      librcm            .WAIT  \
210      libcfgadm         .WAIT  \
211      libpicl           .WAIT  \
212      libpicltree       .WAIT  \
213      raidcfg_plugins   \
214      cfgadm_plugins    \
215      libmail           \
216      lvm               \
217      libsmmedia       \
218      libipp           \
219      libdiskmgt       \
220      liblgrp          \
221      libfsmgt         \
222      fm               \
223      libavl           \
224      libcmdutils      \
225      libcontract      \
226      ../cmd/sendmail/libmilter \
227      sasl_plugins     \
228      udapl            \
229      libzpool         \
230      libzfs_core      \
231      libzfs           \
232      libbe            \
233      pylibbe          \
234      libzfs_jni       \
235      pyzfs            \
236      pysolaris        \
237      libmapid         \
238      brand            \
239      policykit        \
240      hal              \
241      libshare         \
242      libsqlite        \
243      libidmap         \
244      libadutils       \
245      libipmi          \
246      libexacct/demo   \
247      libvrrpadm      \
248      libvscan         \

```

4

new/usr/src/lib/Makefile

```

249     libgrubmgmt  \
250     smbstrv     \
251     libilb      \
252     scsi        \
253     libima      \
254     libsun_ima  \
255     mpapi       \
256     librstp     \
257     librepase   \
258     libhotplug  \
259     libfruutils .WAIT \
260     libfru      \
261     $(MACH)_SUBDIRS)

263 i386_SUBDIRS= \
264     libntfs     \
265     libparted   \
266     libfdisk    \
267     libsaveargs

269 sparc_SUBDIRS= .WAIT \
270     efcodes     \
271     libds       \
272     libdscp     \
273     libprtdiag .WAIT \
274     libprtdiag_psr \
275     libpri      \
276     librscl     \
277     storage     \
278     libpcp     \
279     libtsalarm  \
280     libv12n

282 FM_sparc_DEPLIBS= libpri

284 fm: \
285     libexacct  \
286     libipmi    \
287     libzfs     \
288     scsi       \
289     $(FM_$(MACH)_DEPLIBS)

291 #
292 # Create a special version of $(SUBDIRS) with no .WAIT's, for use with the
293 # clean and clobber targets (for more information, see those targets, below).
294 #
295 NOWAIT_SUBDIRS= $(SUBDIRS:.WAIT=)

297 DCSSUBDIRS = \
298     lvm

300 MSGSUBDIRS= \
301     abidtd_plugins \
302     auditd_plugins \
303     brand           \
304     cfgadm_plugins \
305     gss_mechs/mech_dh \
306     gss_mechs/mech_krb5 \
307     krb5           \
308     libast         \
309     libbsm        \
310     libcb         \
311     libcbfgadm    \
312     libcmd        \
313     libcontract   \
314     libcurses     \

```

5

new/usr/src/lib/Makefile

```

315     libdhcpsvc  \
316     libdhcputil \
317     libipsecutil \
318     libdiskmgmt \
319     libdladm    \
320     libdll      \
321     libgrubmgmt \
322     libgss      \
323     libidmap    \
324     libipmp     \
325     libilb      \
326     libinetutil \
327     libinstzones \
328     libipadm    \
329     libnsd      \
330     libnwam     \
331     libpam      \
332     libpicl     \
333     libpool     \
334     libpkg      \
335     libpp       \
336     libscf      \
337     libsas1     \
338     libldap5    \
339     libsecdb    \
340     libshare    \
341     libshell    \
342     libslldap   \
343     libslp      \
344     libsbmfs    \
345     libsmmedia  \
346     libsum      \
347     libtsol     \
348     libuutil    \
349     libvrrpadm  \
350     libvscan    \
351     libwanboot  \
352     libwanbootutil \
353     libzfs      \
354     libzonecfg  \
355     lvm         \
356     madv        \
357     mpss        \
358     pam_modules \
359     pyzfs       \
360     pysolaris   \
361     rpcsec_gss  \
362     librepase   \
363     MSGSUBDIRS += \
364     $(MACH)_MSGSUBDIRS)

366 sparc_MSGSUBDIRS= \
367     libprtdiag \
368     libprtdiag_psr

370 i386_MSGSUBDIRS= libfdisk

372 HDRSUBDIRS= \
373     auditd_plugins \
374     libast         \
375     libbrand       \
376     libbsm        \
377     libcb         \
378     libcmd        \
379     libcmdutils   \
380     libcommputil \

```

6

new/usr/src/lib/Makefile

```

381 libcontract \
382 libcpc \
383 libctf \
384 libcurses \
385 libtermcap \
386 libcryptoutil \
387 libdevice \
388 libdevvid \
389 libdevinfo \
390 libdiskmgt \
391 libdladm \
392 libdll \
393 libdlpi \
394 libdhcpagent \
395 libdhcpsvc \
396 libdhcputil \
397 libdisasm \
398 libdns_sd \
399 libdscfg \
400 libdtrace \
401 libdtrace_jni \
402 libelfsign \
403 libeti \
404 libfru \
405 libfstyp \
406 libgen \
407 libipadm \
408 libipsecutil \
409 libinetsvc \
410 libinetutil \
411 libinstzones \
412 libipmi \
413 libipmp \
414 libipp \
415 libiscsit \
416 libkstat \
417 libkvm \
418 libmail \
419 libmd \
420 libmtmalloc \
421 libndmp \
422 libnvpair \
423 libnsctl \
424 libnsl \
425 libnwam \
426 libpam \
427 libpcidb \
428 libpctx \
429 libpicl \
430 libpicltree \
431 libpool \
432 libpp \
433 libproc \
434 libraidcfg \
435 librcm \
436 librdc \
437 libscf \
438 libsip \
439 libsbios \
440 librestart \
441 librpcsvc \
442 librsn \
443 librstp \
444 libsas1 \
445 libsec \
446 libshell \

```

7

new/usr/src/lib/Makefile

```

447 libslp \
448 libsmmedia \
449 libsocket \
450 libsqlite \
451 libfcoe \
452 libsrpt \
453 libstmf \
454 libstmfproxy \
455 libsum \
456 libsysevent \
457 libtecla \
458 libtnf \
459 libtnfctl \
460 libtnfprobe \
461 libtsnet \
462 libtsol \
463 libvrrpadm \
464 libvolmgt \
465 libumem \
466 libunistat \
467 libuutil \
468 libwanboot \
469 libwanbootutil \
470 libwrap \
471 libxcurses2 \
472 libzfs \
473 libzfs_core \
474 libzfs_jni \
475 libzoneinfo \
476 libzonestat \
477 hal \
478 policykit \
479 lvm \
480 pkcs11 \
481 passwdutil \
482 ../cmd/sendmail/libmilter \
483 fm \
484 udapl \
485 libmapid \
486 libkrb5 \
487 libmbfs \
488 libshare \
489 libidmap \
490 libvscan \
491 libgrubmgt \
492 smbsrv \
493 libilb \
494 scsi \
495 hbaapi \
496 smhba \
497 libima \
498 libsun_ima \
499 mpapi \
500 libreparse \
501 ${MACH}_HDRSUBDIRS)

513 ${CLOSED_BUILD}HDRSUBDIRS += \
514 ${CLOSED}/lib/libike

503 i386_HDRSUBDIRS= \
504 libparted \
505 libfdisk \
506 libsaveargs

508 sparc_HDRSUBDIRS= \
509 libds \

```

8

new/usr/src/lib/Makefile

```

510 libdscp \
511 libpri \
512 libv12n \
513 storage

515 all := TARGET= all
516 check := TARGET= check
517 clean := TARGET= clean
518 clobber := TARGET= clobber
519 install := TARGET= install
520 install_h := TARGET= install_h
521 lint := TARGET= lint
522 _dc := TARGET= _dc
523 _msg := TARGET= _msg

525 .KEEP_STATE:

527 #
528 # For the all and install targets, we clearly must respect library
529 # dependencies so that the libraries link correctly. However, for
530 # the remaining targets (check, clean, clobber, install_h, lint, _dc
531 # and _msg), libraries do not have any dependencies on one another
532 # and thus respecting dependencies just slows down the build.
533 # As such, for these rules, we use pattern replacement to explicitly
534 # avoid triggering the dependency information. Note that for clean,
535 # clobber and lint, we must use $(NOWAIT_SUBDIRS) rather than
536 # $(SUBDIRS), to prevent '.WAIT' from expanding to '.WAIT-nodepend'.
537 #

539 all: $(SUBDIRS)

541 install: $(SUBDIRS) .WAIT install_extra

543 # extra libraries kept in other source areas
544 install_extra:
545     @cd ../cmd/sgs; pwd; $(MAKE) install_lib
546     @pwd

548 clean clobber lint: $(NOWAIT_SUBDIRS:%=%-nodepend)

550 install_h check: $(HDRSUBDIRS:%=%-nodepend)

552 _msg: $(MSGSUBDIRS:%=%-nodepend) .WAIT _dc

554 _dc: $(DCSUBDIRS:%=%-nodepend)

556 #
557 # Library interdependencies are called out explicitly here
558 #
559 auditd_plugins: libbsm libnsl libsecdb
560 gss_mechs/mech_krb5: libgss libnsl libsocket libresolv pkcs11
561 libadt_jni: libbsm
562 libast: libsocket
563 libadutils: libldap5 libresolv libsocket libnsl
564 nsswitch: libadutils libidmap
565 libbe: libzfs
566 libbsm: libtsol
567 libcmd: libsum libast libsocket libnsl
568 libcmdutils: libavl
569 libcontract: libnvpair
570 libdevinfo: libdevinfo
571 libdevinfo: libnvpair libsec
572 libdhcpgagent: libsocket libdhcputil libuuid libdlpi libcontract
573 libdhcpsvc: libinetutil
574 libdhcputil: libnsl libgen libinetutil libdlpi
575 libdladm: libdevinfo libinetutil libsocket libscf librcm libnvpair \

```

9

new/usr/src/lib/Makefile

```

576 libexacct libnsl libkstat libcurses
577 libdll: libast
578 libdlpi: libinetutil libdladm
579 libds: libsysevent
580 libdscfg: libnsctl libunistat libsocket libnsl
581 libdtrace: libproc libgen libctf
582 libdtrace_jni: libuutil libdtrace
583 libefi: libuuid
584 libfstyp: libnvpair
585 libelfsign: libcryptoutil libkmf
586 libidmap: libadutils libldap5 libavl libslldap libuutil
587 libipadm: libnsl libinetutil libsocket libdlpi libnvpair libdhcpgagent \
588     libdladm libsecdb
589 libiscsit: libc libnvpair libstmf libuuid libnsl
590 libkmf: libcryptoutil pkcs11
591 libnsl: libmd5
592 libmapid: libresolv
593 librdc: libsocket libnsl libnsctl libunistat libdscfg
594 libuuid: libdlpi
608 $(CLOSED_BUILD)libike: libipseutil libxnet libcryptoutil
595 libinetutil: libsocket
596 libipseutil: libtecla libsocket
597 libinstzones: libzonecfg libcontract
598 libpkg: libwanboot libscf libadm
599 libnwm: libscf
600 libsecdb: libnsl
601 libsas1: libgss libsocket pkcs11 libmd
602 sas_plugins: pkcs11 libgss libsocket libsas1
603 libstcp: libsocket
604 libshell: libast libcmd libdll libsocket libsecdb
605 libsip: libmd5
606 libsmbs: libcmdutils libsocket libnsl libkrb5
607 libsocket: libnsl
608 libstmpoxy: libstmf libsocket libnsl libpthread
609 libsum: libast
610 libsysevent: libsecdb
611 libldap5: libsas1 libsocket libnsl libmd
612 libslldap: libldap5 libtsol libnsl libc libscf libresolv
613 libpool: libnvpair libexacct
614 libpp: libast
615 libzonecfg: libc libsocket libnsl libuuid libnvpair libsysevent libsec \
616     libbrand libpool libscf
617 libproc: ../cmd/sgs/librtld_db ../cmd/sgs/libelf libctf libsaveargs
618 libproject: libpool libproc libsecdb
619 libtermcap: libcurses
620 libtsnet: libnsl libtsol libsecdb
621 libwrap: libnsl libsocket
622 libwanboot: libnvpair libresolv libnsl libsocket libdevinfo libinetutil \
623     libdhcputil
624 libwanbootutil: libnsl
625 pam_modules: libproject passwdutil smbsrv
626 libscf: libuutil libmd libgen libsmbios libnsl
627 libnetsvc: libscf
628 librestart: libuutil libscf
629 libsaveargs: libdisasm
630 ../cmd/sgs/libdl: ../cmd/sgs/libconv
631 ../cmd/sgs/libelf: ../cmd/sgs/libconv
632 pkcs11: libcryptoutil
633 print: libldap5
634 udapl/udapl_tavor: udapl/libdat
635 libzfs: libdevinfo libgen libnvpair libuutil \
636     libadm libavl libefi libidmap libmd libzfs_core
637 libzfs_core: libnvpair
638 libzfs_jni: libdiskmgt libnvpair libzfs
639 libzpool: libavl libumem libnvpair libcmdutils
640 libsec: libavl libidmap

```

10


```
641 brand:      libc libsocket
642 libshare:    libscf libzfs libuuid libfsmgmt libsecdb libumem libsmfbs
643 libxacct/demo: libxacct libproject libsocket libnsl
644 libtsalarm:  libpcp
645 smbsrv:      libsocket libnsl libmd libxnet libpthread librt \
646              libshare libidmap pkcs11 libsqlite libcryptoutil \
647              librepase libcmdutils
648 libv12n:     libds libuuid
649 libvrrpadm:  libsocket libdladm libscf
650 libvscan:    libscf
651 libfru:      libfruutils
652 scsi:        libnvpair libfru
653 mpapi:       libpthread libdevinfo libsysevent libnvpair
654 sun_fc:      libdevinfo libsysevent libnvpair
655 libsun_ima:  libdevinfo libsysevent libnsl
656 sun_sas:     libdevinfo libsysevent libnvpair libkstat libdevid
657 libgrubmgmt: libdevinfo libzfs libfstyp
658 pylibbe:     libbe libzfs
659 pyzfs:       libnvpair libzfs
660 pysolaris:  libsec libidmap
661 libreparse: libnvpair
662 libhotplug: libnvpair
663 cfgadm_plugins: libhotplug
664 libilb:     libsocket
665 $(INTEL_BUILD)libdiskmgt:libfdisk

667 #
668 # The reason this rule checks for the existence of the
669 # Makefile is that some of the directories do not exist
670 # in certain situations (e.g., exportable source builds,
671 # OpenSolaris).
672 #
673 $(SUBDIRS): FRC
674     @if [ -f $@/Makefile ]; then \
675         cd $@; pwd; $(MAKE) $(TARGET); \
676     else \
677         true; \
678     fi

680 $(SUBDIRS:%=%-nodepend):
681     @if [ -f $@:%-nodepend=)/Makefile ]; then \
682         cd $@:%-nodepend=); pwd; $(MAKE) $(TARGET); \
683     else \
684         true; \
685     fi

687 FRC:
```

new/usr/src/lib/raidcfg_plugins/Makefile

1

1268 Thu Aug 15 11:59:22 2013

new/usr/src/lib/raidcfg_plugins/Makefile

4027 remove CLOSED_BUILD

```
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 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # ident "%Z%M% %I% %E% SMI"
26 #
25 # lib/raidcfg_plugins/Makefile
26 #

28 include $(SRC)/Makefile.master

32 CLOSED_PLUGIN = $(CLOSED)/lib/raidcfg_plugins/

34 $(CLOSED_BUILD)COMMON_SUBDIRS += $(CLOSED_PLUGIN)/mpt

30 SUBDIRS= $(COMMON_SUBDIRS) $( $(MACH)_SUBDIRS)

32 all:=          TARGET= all
33 install:=     TARGET= install
34 clean:=       TARGET= clean
35 clobber:=     TARGET= clobber
36 lint:=       TARGET= lint
37 _msg:=       TARGET= _msg

39 .KEEP_STATE:

41 all clean clobber lint install _msg: $(SUBDIRS)

43 $(SUBDIRS): FRC
44     @cd $@; pwd; $(MAKE) $(TARGET)

46 FRC:
```

new/usr/src/pkg/Makefile.lic

1

```
*****
1614 Thu Aug 15 11:59:22 2013
new/usr/src/pkg/Makefile.lic
4027 remove CLOSED_BUILD
*****
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) 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 #
27 # PKGDEBUG and LICROOT should be defined on the command line or in
28 # the environment when using this Makefile, as they generally come
29 # from usr/src/pkg/Makefile. The remaining env vars should already
30 # be part of a working build environment.
31 #
32 #
33 include $(SRC)/Makefile.master
34 #
35 #
36 # For license files in the open source tree, always copy them from src
37 # into the license staging directory in the proto area
38 #
39 $(LICROOT)/usr/src/%: $(CODEMGR_WS)/usr/src/%
40     $(PKGDEBUG)if [ ! -d $(@D) ]; then mkdir -p $(@D); fi
41     $(PKGDEBUG)$(INS.file)
42 #
43 #
44 # A build without closed source relies on closed-bins extraction to stage
45 # the license file and does nothing here.
46 # For license files in the closed source tree, the desired action
47 # depends on the type of build.
48 #
49 #
50 # $(LICROOT)/usr/closed/%:
51 # 1. For a normal build, with closed source present, simply copy the
52 #    file into the license staging directory.
53 #
54 # 2. For a Tonic build, copy the file into both open and closed
55 #    license staging directories.
56 #
57 # 3. For a build without closed source, rely on closed-bins extraction
58 #    to stage the license file, and do nothing here.
59 #
60 #
61 $(TONICBUILD)INS= install -O
62 $(LICROOT)/usr/closed/%: $(CLOSED_BUILD) $(CODEMGR_WS)/usr/closed/%
63     $(CLOSED_BUILD)$(PKGDEBUG) \
```

new/usr/src/pkg/Makefile.lic

2

```
48     if [ ! -d $(@D) ]; then \
49         mkdir -p $(@D); \
50     fi; \
62     $(TONICBUILD) $(RM) ${@:${ROOT}/%=${CLOSEDROOT}/%}
63     $(CLOSED_BUILD)$(PKGDEBUG)$(INS.file)
```

new/usr/src/psm/stand/boot/Makefile

1

```
*****
1957 Thu Aug 15 11:59:22 2013
new/usr/src/psm/stand/boot/Makefile
4027 remove CLOSED_BUILD
*****
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 2010 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include      ../../../../Makefile.master

28 sparcv9_ARCHITECTURES = sparcv9
29 sparc_ARCHITECTURES = $(sparcv9_ARCHITECTURES)

31 SUBDIRS = $(MACH)_ARCHITECTURES

33 all           :=      TARGET= all
34 install      :=      TARGET= install
35 clean        :=      TARGET= clean
36 clobber      :=      TARGET= clobber
37 lint         :=      TARGET= lint

39 .KEEP_STATE:

41 all install lint clean: $(SUBDIRS)

43 clobber: $(SUBDIRS)
44      $(RM) make.out lint.out

46 $(SUBDIRS): FRC
47      @cd $@; pwd; $(MAKE) $(TARGET)

49 #
50 # Cross-reference customization: include all boot-related source files.
51 #
52 UTSDIR =      ../../../../uts
53 UTSCLOSED =  ../../../../closed/uts
54 STANDLIBDIR = ../../../../stand/lib
55 STANDSYSDIRS = ../../../../stand/sys
56 PROMDIRS =   ../../promif
57 NAMESDIRS =  ../../lib/names
58 XRDIRS +=    $(STANDLIBDIR) $(STANDSYSDIRS) $(PROMDIRS) $(NAMESDIRS)

60 #
61 # Components beginning with B! are in the open and closed trees; those
```

new/usr/src/psm/stand/boot/Makefile

2

```
62 # beginning with O! are just in the open tree.
63 #
64 XRINCCOMP = B!sun4u O!sfmmu O!sparc/v7 O!sparc/v9 B!sparc B!sun B!common
65 XRINC_TMP = $(XRINCCOMP:B!%=$(UTSDIR)/%)
66 XRINCDIRS = $(XRINC_TMP:O!%=$(UTSDIR)/%)
67 $(CLOSED_BUILD)XRINC_TMP = $(XRINCCOMP:B!%=$(UTSDIR)/% $(UTSCLOSED)/%)
68 $(CLOSED_BUILD)XRINCDIRS = $(XRINC_TMP:O!%=$(UTSDIR)/%)

68 cscope.out tags: FRC
69      $(XREF) -x $@

71 FRC:
```

new/usr/src/tools/findunref/Makefile

1

1475 Thu Aug 15 11:59:22 2013

new/usr/src/tools/findunref/Makefile

4027 remove CLOSED_BUILD

```
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 2009 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #
27 PROG = findunref
28 MANFILES = findunref.1
29 CFLAGS += $(CCVERBOSE)
30 LINTFLAGS += -ux
31 #
32 include ../Makefile.tools
33 #
34 CERRWARN += _gcc=-Wno-unused
35 CERRWARN += _gcc=-Wno-parentheses
36 #
37 $(ROOTONBLDMANFILES) := FILEMODE= 644
38 #
39 EXCEPTION_SRC= common open
40 $(CLOSED_BUILD)EXCEPTION_SRC += closed
41 EXCEPTION_LISTS= $(EXCEPTION_SRC:%=exception_list.%)
42 #
43 CLOBBERFILES += exception_list
44 .KEEP_STATE:
45 #
46 all: $(PROG) exception_list
47 #
48 install: all .WAIT $(ROOTONBLDMACHPROG) $(ROOTONBLDMANFILES)
49 #
50 lint: lint_PROG
51 #
52 exception_list: $(EXCEPTION_LISTS)
53 -$(RM) $@
54 $(CAT) $(EXCEPTION_LISTS) > $@
55 #
56 clean:
57 #
58 include ../Makefile.targ
```

new/usr/src/tools/scripts/nightly.sh

1

85358 Thu Aug 15 11:59:22 2013

new/usr/src/tools/scripts/nightly.sh

4027 remove CLOSED_BUILD

unchanged_portion_omitted_

```
894 #
895 # Verify that the closed tree is present if it needs to be.
896 # Sets CLOSED_IS_PRESENT for future use.
896 #
897 function check_closed_tree {
898     if [[ ! -d "$ON_CLOSED_BINS" ]]; then
899         if [ -z "$CLOSED_IS_PRESENT" ]; then
900             if [ -d $CODEMGR_WS/usr/closed ]; then
901                 CLOSED_IS_PRESENT="yes"
902             else
903                 CLOSED_IS_PRESENT="no"
904             fi
905             export CLOSED_IS_PRESENT
906         fi
907         if [[ "$CLOSED_IS_PRESENT" = no && ! -d "$ON_CLOSED_BINS" ]]; then
908             #
909             # If it's an old (pre-split) tree or an empty
910             # workspace, don't complain.
911             #
912             if grep -s CLOSED_BUILD $SRC/Makefile.master > /dev/null; then
899             echo "If the closed sources are not present," \
900                 "ON_CLOSED_BINS"
901             echo "must point to the closed binaries tree."
902             build_ok=n
903             exit 1
904         fi
919     fi
905 }
```

unchanged_portion_omitted_

```

*****
6341 Thu Aug 15 11:59:23 2013
new/usr/src/uts/Makefile
4027 remove CLOSED_BUILD
*****
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 #
24 # include global definitions
25 include ../Makefile.master

27 #
28 # List of architectures to build as part of the standard build.
29 #
30 # Some of these architectures are built in parallel (see i386_PARALLEL and
31 # sparc_PARALLEL). This requires building some parts first before parallel build
32 # can start. Platform make files know what should be built as a prerequisite for
33 # the parallel build to work. The i386_PREREQ and sparc_PREREQ variables tell
34 # which platform directory to enter to start making prerequisite dependencies.
35 #
36 sparc_ARCHITECTURES = sun4v sun4u sparc

38 i386_ARCHITECTURES = i86pc i86xpv intel

40 #
41 # For i386 all architectures can be compiled in parallel.
42 #
43 # intel/Makefile knows how to build prerequisites needed for parallel build.
44 #
45 i386_PREREQ = intel
46 i386_PARALLEL = $(i386_ARCHITECTURES)

48 #
49 # For sparc all architectures can be compiled in parallel.
50 #
51 # sun4/Makefile knows how to build prerequisites needed for parallel build.
52 # can start.
53 #
54 sparc_PREREQ = sun4
55 sparc_PARALLEL = $(sparc_ARCHITECTURES)

57 #
58 # Platforms defined in $(MACH)_PARALLEL are built in parallel. DUMMY is placed
59 # at the end in case $(MACH)_PARALLEL is empty to prevent everything going in
60 # parallel.
61 #

```

```

62 .PARALLEL: $(MACH)_PARALLEL) DUMMY

64 #
65 # For build prerequisites we use a special target which is constructed by adding
66 # '.prereq' suffix to the $(MACH)_PREREQ.
67 #
68 PREREQ_TARGET = $(MACH)_PREREQ:%=%prereq

71 def := TARGET= def
72 all := TARGET= all
73 install := TARGET= install
74 install_h := TARGET= install_h
75 clean := TARGET= clean
76 clobber := TARGET= clobber
77 lint := TARGET= lint
78 clean.lint := TARGET= clean.lint
79 check := TARGET= check
80 modlist := TARGET= modlist
81 modlist := NO_STATE= -K $$MODSTATE$$$

83 .KEEP_STATE:

85 def all lint: all_h $(PMTMO_FILE) $(MACH)_ARCHITECTURES)

87 install: all_h install_dirs $(PMTMO_FILE) $(MACH)_ARCHITECTURES)

89 install_dirs:
90 @cd ..; pwd; $(MAKE) rootdirs
91 @pwd

93 #
94 # Rule to build prerequisites. The left part of the pattern will match
95 # PREREQ_TARGET.
96 #
97 # The location of the Makefile is determined by stripping '.prereq' suffix from
98 # the target name. We add '.prereq' suffix to the target passed to the child
99 # Makefile so that it can distinguish prerequisite build from the regular one.
100 #
101 #
102 %.prereq:
103 @cd $(@:%prereq=); pwd; $(MAKE) $(NO_STATE) $(TARGET).prereq

105 #
106 # Rule to build architecture files. Build all required prerequisites and then
107 # build the rest (potentially in parallel).
108 #
109 $(MACH)_ARCHITECTURES: $(PREREQ_TARGET) FRC
110 @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

112 $(PMTMO_FILE) pmtmo_file: $(PATCH_MAKEUP_TABLE)
113 @if [ -z "$(PATCH_MAKEUP_TABLE)" ] ; then \
114 echo 'ERROR: $(PATCH_MAKEUP_TABLE) not set' \
115 'in environment' >&2 ; \
116 exit 1 ; \
117 fi
118 RELEASE="$(RELEASE)" MACH="$(MACH)" \
119 $(CTFCVTP_TBL) -o $(PMTMO_FILE) $(PATCH_MAKEUP_TABLE)

121 #
122 # The following is the list of directories which contain Makefiles with
123 # targets to install header file. The machine independent headers are
124 # installed by invoking the Makefile in the directory containing the
125 # header files. Machine and architecture dependent headers are installed
126 # by invoking the main makefile for that architecture/machine which,
127 # in turn, is responsible for invoking the Makefiles which install headers.

```

new/usr/src/uts/Makefile

3

```

128 # It is done this way so as not to assume that all of the header files in
129 # the architecture/machine dependent subdirectories are in completely
130 # isomorphic locations.
131 #
132 COMMON_HDRDIRS= common/avs \
133                 common/c2 \
134                 common/des \
135                 common/fs \
136                 common/gssapi \
137                 common/idmap \
138                 common/inet \
139                 common/inet/ipf/netinet \
140                 common/inet/kssl \
141                 common/inet/nca \
142                 common/inet/sockmods/netpacket \
143                 common/io/bpf/net \
144                 common/ipp \
145                 common/net \
146                 common/netinet \
147                 common/nfs \
148                 common/pcmcia/sys \
149                 common/rpc \
150                 common/rpcsvc \
151                 common/sharefs \
152                 common/smb \
153                 common/smbsrv \
154                 common/sys \
155                 common/vm

158 # These aren't the only headers in closed. But the other directories
159 # are simple enough that they can be driven from the src tree.
160 $(CLOSED_BUILD)COMMON_HDRDIRS += $(CLOSED)/uts/common/sys

158 #
159 # Subset of COMMON_HDRDIRS in which at least one header is generated
160 # at runtime (e.g., rpcgen). (This is a partial list; there are
161 # other directories that should be included and do not yet have the
162 # necessary Makefile support. See 6414855.)
163 #
164 DYNHDRDIRS = common/rpcsvc common/idmap common/sys

166 sparc_HDRDIRS= sun/sys
167 i386_HDRDIRS= i86pc/vm i86xpv/vm

169 HDRDIRS= $(COMMON_HDRDIRS) $(MACH)_HDRDIRS
170 install_h check: $(HDRDIRS) $(MACH)_ARCHITECTURES

172 $(HDRDIRS): FRC
173     @cd $@; pwd; $(MAKE) $(TARGET)

175 # ensures that headers made by rpcgen and others are available in uts source
176 # for kernel builds to reference without building install_h
177 #
178 all_h: FRC
179     @cd common/sys; pwd; $(MAKE) $@
180     @cd common/rpc; pwd; $(MAKE) $@
181     @cd common/rpcsvc; pwd; $(MAKE) $@
182     @cd common/gssapi; pwd; $(MAKE) $@
183     @cd common/idmap; pwd; $(MAKE) $@

185 clean clobber: $(MACH)_ARCHITECTURES $(DYNHDRDIRS)
186     @if [ '$(PATCH_BUILD)' != '#' ] ; then \
187         echo $(RM) $(PMTMO_FILE) ; \
188         $(RM) $(PMTMO_FILE) ; \
189     fi

```

new/usr/src/uts/Makefile

4

```

191 EXTRA_CLOBBER_TARGETS= common/avs/ns/rdc
192 clobber: $(EXTRA_CLOBBER_TARGETS)

195 clean.lint modlist: $(MACH)_ARCHITECTURES

197 #
198 # Cross-reference customization: build a cross-reference over all of
199 # the supported architectures. Although there's no correct way to set
200 # the include path (since we don't know what architecture is the one
201 # the user will be interested in), it's historically been set to
202 # mirror the $(XRDIRS) list, and that works kinda sorta okay.
203 #
204 XRDIRS = $(sparc_ARCHITECTURES) $(i386_ARCHITECTURES) sun4 sfmmu \
205 # We need to manually prune usr/closed/uts/{i86xpv/sfmmu/i86pc} since
206 # none of them exist.
207 #
208 SHARED_XRDIRS = $(sparc_ARCHITECTURES) $(i386_ARCHITECTURES) sun4 sfmmu \
209                 sun common
210 CLOSED_XRDIRS = $(SHARED_XRDIRS:%=% ../../closed/uts/%)
211 XRDIRS = $(SHARED_XRDIRS)
212 CLOSED_XRDIRS_XEN = $(CLOSED_XRDIRS:../../closed/uts/i86xpv=)
213 CLOSED_XRDIRS_1 = $(CLOSED_XRDIRS_XEN:../../closed/uts/i86pc=)
214 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS_1:../../closed/uts/sfmmu=)

207 XRINCDIRS = $(XRDIRS)

209 cscope.out tags: FRC
210     $(XREF) -x $@

212 FRC:

```


new/usr/src/uts/Makefile.uts

1

```
*****
22668 Thu Aug 15 11:59:23 2013
new/usr/src/uts/Makefile.uts
4027 remove CLOSED_BUILD
*****
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) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
25 # Copyright (c) 2011 by Delphix. All rights reserved.
26 #
27 #
28 #
29 # This Makefile contains the common targets and definitions for
30 # all kernels. It is to be included in the Makefiles for specific
31 # implementation architectures and processor architecture dependent
32 # modules: i.e.: all driving kernel Makefiles.
33 #
34 # Include global definitions:
35 #
36 include $(SRC)/Makefile.master
37 #
38 #
39 # No text domain in the kernel.
40 #
41 DTEXTDOM =
42 #
43 #
44 # Keep references to $(SRC)/common relative.
45 COMMONBASE= $(UTSBASE)/../common
46 #
47 #
48 # Setup build-specific vars
49 # To add a build type:
50 # add name to ALL_BUILDS32 & ALL_BUILDS64
51 # set CLASS_name and OBJ_DIR_name
52 # add targets to Makefile.targ
53 #
54 #
55 #
56 # DEF_BUILDS is for def, lint, sischek, and install
57 # ALL_BUILDS is for everything else (all, clean, ...)
58 #
59 # The NOT_RELEASE_BUILD noise is to maintain compatibility with the
60 # gatekeeper's nightly build script.
61 #
```

new/usr/src/uts/Makefile.uts

2

```
62 DEF_BUILDS32 = obj32
63 DEF_BUILDS64 = obj64
64 DEF_BUILDSONLY64 = obj64
65 $(NOT_RELEASE_BUILD)DEF_BUILDS32 = debug32
66 $(NOT_RELEASE_BUILD)DEF_BUILDS64 = debug64
67 $(NOT_RELEASE_BUILD)DEF_BUILDSONLY64 = debug64
68 ALL_BUILDS32 = obj32 debug32
69 ALL_BUILDS64 = obj64 debug64
70 ALL_BUILDSONLY64 = obj64 debug64
71 #
72 #
73 # For modules in 64b dirs that aren't built 64b
74 # or modules in 64b dirs that aren't built 32b we
75 # need to create empty modlintlib files so global lint works
76 #
77 LINT32_BUILDS = debug32
78 LINT64_BUILDS = debug64
79 #
80 #
81 # Build class (32b or 64b)
82 #
83 CLASS_OBJ32 = 32
84 CLASS_DBG32 = 32
85 CLASS_OBJ64 = 64
86 CLASS_DBG64 = 64
87 CLASS = $(CLASS_$(BUILD_TYPE))
88 #
89 #
90 # Build subdirectory
91 #
92 OBJ32_DIR_OBJ32 = obj32
93 OBJ32_DIR_DBG32 = debug32
94 OBJ64_DIR_OBJ64 = obj64
95 OBJ64_DIR_DBG64 = debug64
96 OBJ32_DIR = $(OBJ32_DIR_$(BUILD_TYPE))
97 #
98 #
99 # Create defaults so empty rules don't
100 # confuse make
101 #
102 CLASS_32 = 32
103 OBJ32_DIR_32 = debug32
104 #
105 #
106 # Build tools
107 #
108 CC_sparc_32 = $(sparc_CC)
109 CC_sparc_64 = $(sparcv9_CC)
110 #
111 CC_i386_32 = $(i386_CC)
112 CC_i386_64 = $(amd64_CC)
113 CC_amd64_64 = $(amd64_CC)
114 #
115 CC = $(CC_$(MACH)_$(CLASS))
116 #
117 AS_sparc_32 = $(sparc_AS)
118 AS_sparc_64 = $(sparcv9_AS)
119 #
120 AS_i386_32 = $(i386_AS)
121 AS_i386_64 = $(amd64_AS)
122 AS_amd64_64 = $(amd64_AS)
123 #
124 AS = $(AS_$(MACH)_$(CLASS))
125 #
126 LD_sparc_32 = $(sparc_LD)
127 LD_sparc_64 = $(sparcv9_LD)
```

```

129 LD_i386_32      = $(i386_LD)
130 LD_i386_64     = $(amd64_LD)
131 LD_amd64_64    = $(amd64_LD)

133 LD              = $(LD_$(MACH)_$(CLASS))

135 LINT_sparc_32   = $(sparc_LINT)
136 LINT_sparc_64  = $(sparcv9_LINT)

138 LINT_i386_32   = $(i386_LINT)
139 LINT_i386_64   = $(amd64_LINT)
140 LINT_amd64_64  = $(amd64_LINT)

142 LINT            = $(LINT_$(MACH)_$(CLASS))

144 MODEL_32       = ilp32
145 MODEL_64       = lp64
146 MODEL           = $(MODEL_$(CLASS))

148 #
149 #   Build rules for linting the kernel.
150 #
151 LHEAD = $(ECHO) "\n$@";

153 # Note: egrep returns "failure" if there are no matches, which is
154 # exactly the opposite of what we need.
155 LGREP.2 =      if egrep -v ' (_init|fini|_info) '; then false; else true; fi

157 LTAIL =

159 LINT.c =        $(LINT) -c -dirout=$(LINTS_DIR) $(LINTFLAGS) $(LINT_DEFS) $(CPPF

161 # Please do not add new erroff directives here.  If you need to disable
162 # lint warnings in your module for things that cannot be fixed in any
163 # reasonable manner, please augment LINTTAGS in your module Makefile
164 # instead.
165 LINTTAGS        = -erroff=E_INCONS_ARG_DECL2
166 LINTTAGS        += -erroff=E_INCONS_VAL_TYPE_DECL2

168 LINTFLAGS_sparc_32 = $(LINTCCMODE) -nsxmuF -errtags=yes
169 LINTFLAGS_sparc_64 = $(LINTFLAGS_sparc_32) -m64
170 LINTFLAGS_i386_32 = $(LINTCCMODE) -nsxmuF -errtags=yes
171 LINTFLAGS_i386_64 = $(LINTFLAGS_i386_32) -m64

173 LINTFLAGS        = $(LINTFLAGS_$(MACH)_$(CLASS)) $(LINTTAGS)
174 LINTFLAGS        += $(C99LMODE)

176 #
177 #   Override this variable to modify the name of the lint target.
178 #
179 LINT_MODULE=     $(MODULE)

181 #
182 #   Build the compile/assemble lines:
183 #
184 EXTRA_OPTIONS    =
185 AS_DEFS          = -D_ASM -D__STDC__=0

187 ALWAYS_DEFS_32  = -D_KERNEL -D_SYSCALL32 -D_DDI_STRICT
188 ALWAYS_DEFS_64  = -D_KERNEL -D_SYSCALL32 -D_SYSCALL32_IMPL -D_ELF64 \
189                 -D_DDI_STRICT
190 #
191 # XX64 This should be defined by the compiler!
192 #
193 ALWAYS_DEFS_64  += -Dsun -D__sun -D__SVR4

```

```

194 ALWAYS_DEFS      = $(ALWAYS_DEFS_$(CLASS))

196 #
197 #   CPPFLAGS is deliberately set with a "=" and not a "+=".  For the kernel
198 #   the header include path should not look for header files outside of
199 #   the kernel code.  This "=" removes the search path built in
200 #   Makefile.master inside CPPFLAGS.  Ditto for AS_CPPFLAGS.
201 #
202 CPPFLAGS          = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) \
203                   $(INCLUDE_PATH) $(EXTRA_OPTIONS)
204 ASFLAGS           += -P
205 AS_CPPFLAGS       = $(ALWAYS_DEFS) $(ALL_DEFS) $(CONFIG_DEFS) $(AS_DEFS) \
206                   $(AS_INC_PATH) $(EXTRA_OPTIONS)

208 #
209 #   Make it (relatively) easy to share compilation options between
210 #   all kernel implementations.
211 #

213 # Override the default, the kernel is squeaky clean
214 CERRWARN = -errtags=yes -errwarn=%all

216 CERRWARN += -_gcc=-Wno-missing-braces
217 CERRWARN += -_gcc=-Wno-sign-compare
218 CERRWARN += -_gcc=-Wno-unknown-pragmas
219 CERRWARN += -_gcc=-Wno-unused-parameter
220 CERRWARN += -_gcc=-Wno-missing-field-initializers

222 # DEBUG v. -nd make for frequent unused variables, empty conditions, etc. in
223 # -nd builds
224 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-unused
225 $(RELEASE_BUILD)CERRWARN += -_gcc=-Wno-empty-body

227 C99MODE = $(C99_ENABLE)

229 CFLAGS_uts        =
230 CFLAGS_uts        += $(STAND_FLAGS_$(CLASS))
231 CFLAGS_uts        += $(CCVERBOSE)
232 CFLAGS_uts        += $(ILD OFF)
233 CFLAGS_uts        += $(XAOPT)
234 CFLAGS_uts        += $(CTF_FLAGS_$(CLASS))
235 CFLAGS_uts        += $(CERRWARN)
236 CFLAGS_uts        += $(CCNOAUTOINLINE)
237 CFLAGS_uts        += $(CGLOBALSTATIC)
238 CFLAGS_uts        += $(EXTRA_CFLAGS)
239 CFLAGS_uts        += $(CSOURCEDEBUGFLAGS)
240 CFLAGS_uts        += $(CUSERFLAGS)

242 #
243 #   Declare that $(OBJECTS) and $(LINTS) can be compiled in parallel.
244 #   The DUMMY target is for those instances where OBJECTS and LINTS
245 #   are empty (to avoid an unconditional .PARALLEL).
246 .PARALLEL:       $(OBJECTS) $(LINTS) DUMMY

248 #
249 #   Expanded dependencies
250 #
251 DEF_DEPS          = $(DEF_BUILDS:%=def.%)
252 ALL_DEPS          = $(ALL_BUILDS:%=all.%)
253 CLEAN_DEPS        = $(ALL_BUILDS:%=clean.%)
254 CLOBBER_DEPS      = $(ALL_BUILDS:%=clobber.%)
255 LINT_DEPS         = $(DEF_BUILDS:%=lint.%)
256 MODLINTLIB_DEPS  = $(DEF_BUILDS:%=modlintlib.%)
257 MODLIST_DEPS      = $(DEF_BUILDS:%=modlist.%)
258 CLEAN_LINT_DEPS  = $(ALL_BUILDS:%=clean.lint.%)
259 INSTALL_DEPS      = $(DEF_BUILDS:%=install.%)

```

new/usr/src/uts/Makefile.uts

```

260 SYM_DEPS      = $(SYM_BUILDS:%=symcheck.%)
261 SISCHECK_DEPS = $(DEF_BUILDS:%=sischeck.%)
262 SISCLEAN_DEPS = $(ALL_BUILDS:%=sisclean.%)

264 #
265 #      Default module name
266 #
267 BINARY        = $(OBJS_DIR)/$(MODULE)

269 #
270 #      Default cleanup definitions
271 #
272 CLEANLINTFILES = $(LINTS) $(MOD_LINT_LIB)
273 CLEANFILES     = $(OBJECTS) $(CLEANLINTFILES)
274 CLOBBERFILES   = $(BINARY) $(CLEANFILES)

276 #
277 #      Installation constants:
278 #
279 #      FILEMODE is the mode given to the kernel modules
280 #      CFILEMODE is the mode given to the '.conf' files
281 #
282 FILEMODE      = 755
283 DIRMODE       = 755
284 CFILEMODE     = 644

286 #
287 #      Special Installation Macros for the installation of '.conf' files.
288 #
289 #      These are unique because they are not installed from the current
290 #      working directory.
291 #
292 #      Sigh. Apparently at some time in the past there was a confusion on
293 #      whether the name is SRC_CONFFILE or SRC_CONFFILE. Consistency with the
294 #      other names would indicate SRC_CONFFILE, but the voting is >180 Makefiles
295 #      with SRC_CONFFILE and about 11 with SRC_CONFFILE. Software development
296 #      isn't a popularity contest, though, and so my inclination is to define
297 #      both names for now and incrementally convert to SRC_CONFFILE to be consistent
298 #      with the other names.
299 #
300 CONFFILE      = $(MODULE).conf
301 SRC_CONFFILE  = $(CONF_SRCDIR)/$(CONFFILE)
302 SRC_CONFFILE  = $(SRC_CONFFILE)
303 ROOT_CONFFILE_32 = $(ROOTMODULE).conf
304 ROOT_CONFFILE_64 = $(ROOTMODULE:%/(SUBDIR64)/$(MODULE)=%/$(MODULE)).conf
305 ROOT_CONFFILE  = $(ROOT_CONFFILE_$(CLASS))

308 INS.conf= \
309   $(RM) $@; $(INS) -s -m $(CFILEMODE) -f $(@D) $(SRC_CONFFILE)

311 #
312 # The CTF merge of child kernel modules is performed against one of the genunix
313 # modules. For Intel builds, all modules will be used with a single genunix:
314 # the one built in intel/genunix. For SPARC builds, a given
315 # module may be
316 # used with one of a number of genunix files, depending on what platform the
317 # module is deployed on. We merge against the sun4u genunix to optimize for
318 # the common case. We also merge against the ip driver since networking is
319 # typically loaded and types defined therein are shared between many modules.
320 #
321 CTFMERGE_GUDIR_sparc = sun4u
322 CTFMERGE_GUDIR_i386 = intel
323 CTFMERGE_GUDIR      = $(CTFMERGE_GUDIR_$(MACH))

325 CTFMERGE_GENUNIX    = \

```

5

new/usr/src/uts/Makefile.uts

```

326   $(UTSBASE)/$(CTFMERGE_GUDIR)/genunix/$(OBJS_DIR)/genunix

328 #
329 # Used to uniquify a non-genunix module against genunix. If used in patch
330 # mode (PATCH_BUILD != "#"), the patch ID corresponding to the module being
331 # built will be used as the label. If no ID is available, or if patch mode
332 # is not being used, the value of $VERSION will be used.
333 #
334 # For the ease of developers dropping modules onto possibly unrelated systems,
335 # you can set NO_GENUNIX_UNIQUIFY= in the environment to skip uniquifying
336 # against genunix.
337 #
338 NO_GENUNIX_UNIQUIFY=$(POUND_SIGN)
339 SKIP_GENUNIX_UNIQUIFY=no
340 $(NO_GENUNIX_UNIQUIFY)SKIP_GENUNIX_UNIQUIFY=yes

342 CTFMERGE_UNIQUIFY_AGAINST_GENUNIX = \
343   @label="-L VERSION" ; \
344   uniq= ; \
345   if [ -z "$(PATCH_BUILD)" ] ; then \
346     uniq="-D BASE" ; \
347     set -- `$(CTFFINDMOD) -n -r -t $(PMTMO_FILE) $@` ; \
348     if [ "X$$1" != "X-" ] ; then \
349       label="-l $$1" ; \
350       if [ "$$2" != "fcs" ] ; then \
351         uniq="-D $$2" ; \
352         fi ; \
353     fi ; \
354   fi ; \
355   if [ "$(SKIP_GENUNIX_UNIQUIFY)" = "yes" ] ; then \
356     uniq= ; \
357   else \
358     uniq="-d $(CTFMERGE_GENUNIX) $$uniq" ; \
359   fi ; \
360   cmd="$(CTFMERGE) $(CTFMERGE_FLAGS) $$label $$uniq" ; \
361   cmd="$cmd -o $@ $(OBJECTS) $(CTFEXTRAOBJS)" ; \
362   echo $$cmd ; \
363   $$cmd

365 #
366 # Used to merge the genunix module. genunix has special requirements in
367 # patch mode. In particular, it needs to be able to find the genunix used
368 # in the previous version of the KU patch (or the FCS version of genunix in
369 # the case of KU 1).
370 #
371 CTFMERGE_GENUNIX_MERGE = \
372   @if [ -z "$(PATCH_BUILD)" ] ; then \
373     set -- `$(CTFFINDMOD) -b $(OBJS_DIR) -o patch,lastgu -n -r \
374       -t $(PMTMO_FILE) $(GENUNIX) || true` ; \
375     msg= ; \
376     if [ $$$$(POUND_SIGN) -eq 1 ] ; \
377     then msg="Error in $(CTFFINDMOD)" ; \
378     elif [ "X$$1" = "X-" ] ; then msg="Did not get label" ; \
379     elif [ "X$$2" = "X-" ] ; then msg="Did not get withfile" ; \
380     fi ; \
381     if [ -n "$$msg" ] ; then \
382       echo "make ctf: $$msg - removing $(GENUNIX)" ; \
383       $(RM) $(GENUNIX) ; \
384       exit 1 ; \
385     fi ; \
386     label="-l $$1" ; \
387     with="-w $$2" ; \
388   else \
389     label="-L VERSION" ; \
390   fi ; \
391   cmd="$(CTFMERGE) $(CTFMERGE_FLAGS) $$label $$with -o $@ ; \

```

6

```

392     echo $$cmd "$(OBJECTS) $(CTFEXTRAOBS) $(IPCTF_TARGET)"; \
393     $$cmd $(OBJECTS) $(CTFEXTRAOBS) $(IPCTF_TARGET)

395 #
396 # We ctfmerge the ip objects into genunix to maximize the number of common types
397 # found there, thus maximizing the effectiveness of uniquification. We don't
398 # want the genunix build to have to know about the individual ip objects, so we
399 # put them in an archive. The genunix ctfmerge then includes this archive.
400 #
401 IPCTF             = $(IPDRV_DIR)/$(OBSJ_DIR)/ipctf.a

403 #
404 # Rule for building fake shared libraries used for symbol resolution
405 # when building other modules. -znoreloc is needed here to avoid
406 # tripping over code that isn't really suitable for shared libraries.
407 #
408 BUILD.SO         = \
409     $(LD) -o @$ $(GSHARED) $(ZNORELOC) -h $(SONAME)

411 #
412 # SONAME defaults for common fake shared libraries.
413 #
414 $(LIBGEN)        := SONAME = $(MODULE)
415 $(PLATLIB)       := SONAME = misc/platmod
416 $(CPULIB)        := SONAME = 'cpu/$$CPU'
417 $(DTRACESTUBS)  := SONAME = dtracestubs

419 #
420 #     Installation directories
421 #

423 #
424 #     For now, 64b modules install into a subdirectory
425 #     of their 32b brethren.
426 #
427 SUBDIR64_sparc   = sparcv9
428 SUBDIR64_i386    = amd64
429 SUBDIR64         = $(SUBDIR64_$(MACH))

431 ROOT_MOD_DIR     = $(ROOT)/kernel

433 ROOT_KERN_DIR_32 = $(ROOT_MOD_DIR)
434 ROOT_BRAND_DIR_32 = $(ROOT_MOD_DIR)/brand
435 ROOT_DRV_DIR_32   = $(ROOT_MOD_DIR)/drv
436 ROOT_DTRACE_DIR_32 = $(ROOT_MOD_DIR)/dtrace
437 ROOT_EXEC_DIR_32  = $(ROOT_MOD_DIR)/exec
438 ROOT_FS_DIR_32    = $(ROOT_MOD_DIR)/fs
439 ROOT_SCHED_DIR_32 = $(ROOT_MOD_DIR)/sched
440 ROOT_SOCKET_DIR_32 = $(ROOT_MOD_DIR)/socketmod
441 ROOT_STRMOD_DIR_32 = $(ROOT_MOD_DIR)/strmod
442 ROOT_IPP_DIR_32   = $(ROOT_MOD_DIR)/ipp
443 ROOT_SYS_DIR_32   = $(ROOT_MOD_DIR)/sys
444 ROOT_MISC_DIR_32  = $(ROOT_MOD_DIR)/misc
445 ROOT_KGSS_DIR_32  = $(ROOT_MOD_DIR)/misc/kgss
446 ROOT_SCSI_VHCI_DIR_32 = $(ROOT_MOD_DIR)/misc/scsi_vhci
447 ROOT_PMCS_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/pmcs
448 ROOT_QLC_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/qlc
449 ROOT_EMLXS_FW_DIR_32 = $(ROOT_MOD_DIR)/misc/emlxs
450 ROOT_NLMISC_DIR_32 = $(ROOT_MOD_DIR)/misc
451 ROOT_MACH_DIR_32  = $(ROOT_MOD_DIR)/mach
452 ROOT_CPU_DIR_32   = $(ROOT_MOD_DIR)/cpu
453 ROOT_TOD_DIR_32   = $(ROOT_MOD_DIR)/tod
454 ROOT_FONT_DIR_32  = $(ROOT_MOD_DIR)/fonts
455 ROOT_DACF_DIR_32  = $(ROOT_MOD_DIR)/dacf
456 ROOT_CRYPTODIR_32 = $(ROOT_MOD_DIR)/crypto
457 ROOT_MAC_DIR_32   = $(ROOT_MOD_DIR)/mac

```

```

458 ROOT_KICONV_DIR_32 = $(ROOT_MOD_DIR)/kiconv

460 ROOT_KERN_DIR_64   = $(ROOT_MOD_DIR)/$(SUBDIR64)
461 ROOT_BRAND_DIR_64  = $(ROOT_MOD_DIR)/brand/$(SUBDIR64)
462 ROOT_DRV_DIR_64    = $(ROOT_MOD_DIR)/drv/$(SUBDIR64)
463 ROOT_DTRACE_DIR_64 = $(ROOT_MOD_DIR)/dtrace/$(SUBDIR64)
464 ROOT_EXEC_DIR_64   = $(ROOT_MOD_DIR)/exec/$(SUBDIR64)
465 ROOT_FS_DIR_64     = $(ROOT_MOD_DIR)/fs/$(SUBDIR64)
466 ROOT_SCHED_DIR_64  = $(ROOT_MOD_DIR)/sched/$(SUBDIR64)
467 ROOT_SOCKET_DIR_64 = $(ROOT_MOD_DIR)/socketmod/$(SUBDIR64)
468 ROOT_STRMOD_DIR_64 = $(ROOT_MOD_DIR)/strmod/$(SUBDIR64)
469 ROOT_IPP_DIR_64    = $(ROOT_MOD_DIR)/ipp/$(SUBDIR64)
470 ROOT_SYS_DIR_64    = $(ROOT_MOD_DIR)/sys/$(SUBDIR64)
471 ROOT_MISC_DIR_64   = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
472 ROOT_KGSS_DIR_64   = $(ROOT_MOD_DIR)/misc/kgss/$(SUBDIR64)
473 ROOT_SCSI_VHCI_DIR_64 = $(ROOT_MOD_DIR)/misc/scsi_vhci/$(SUBDIR64)
474 ROOT_PMCS_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/pmcs/$(SUBDIR64)
475 ROOT_QLC_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/qlc/$(SUBDIR64)
476 ROOT_EMLXS_FW_DIR_64 = $(ROOT_MOD_DIR)/misc/emlxs/$(SUBDIR64)
477 ROOT_NLMISC_DIR_64 = $(ROOT_MOD_DIR)/misc/$(SUBDIR64)
478 ROOT_MACH_DIR_64   = $(ROOT_MOD_DIR)/mach/$(SUBDIR64)
479 ROOT_CPU_DIR_64    = $(ROOT_MOD_DIR)/cpu/$(SUBDIR64)
480 ROOT_TOD_DIR_64    = $(ROOT_MOD_DIR)/tod/$(SUBDIR64)
481 ROOT_FONT_DIR_64   = $(ROOT_MOD_DIR)/fonts/$(SUBDIR64)
482 ROOT_DACF_DIR_64   = $(ROOT_MOD_DIR)/dacf/$(SUBDIR64)
483 ROOT_CRYPTODIR_64 = $(ROOT_MOD_DIR)/crypto/$(SUBDIR64)
484 ROOT_MAC_DIR_64    = $(ROOT_MOD_DIR)/mac/$(SUBDIR64)
485 ROOT_KICONV_DIR_64 = $(ROOT_MOD_DIR)/kiconv/$(SUBDIR64)

487 ROOT_KERN_DIR     = $(ROOT_KERN_DIR_$(CLASS))
488 ROOT_BRAND_DIR     = $(ROOT_BRAND_DIR_$(CLASS))
489 ROOT_DRV_DIR       = $(ROOT_DRV_DIR_$(CLASS))
490 ROOT_DTRACE_DIR    = $(ROOT_DTRACE_DIR_$(CLASS))
491 ROOT_EXEC_DIR      = $(ROOT_EXEC_DIR_$(CLASS))
492 ROOT_FS_DIR        = $(ROOT_FS_DIR_$(CLASS))
493 ROOT_SCHED_DIR     = $(ROOT_SCHED_DIR_$(CLASS))
494 ROOT_SOCKET_DIR    = $(ROOT_SOCKET_DIR_$(CLASS))
495 ROOT_STRMOD_DIR    = $(ROOT_STRMOD_DIR_$(CLASS))
496 ROOT_IPP_DIR       = $(ROOT_IPP_DIR_$(CLASS))
497 ROOT_SYS_DIR       = $(ROOT_SYS_DIR_$(CLASS))
498 ROOT_MISC_DIR      = $(ROOT_MISC_DIR_$(CLASS))
499 ROOT_KGSS_DIR      = $(ROOT_KGSS_DIR_$(CLASS))
500 ROOT_SCSI_VHCI_DIR = $(ROOT_SCSI_VHCI_DIR_$(CLASS))
501 ROOT_PMCS_FW_DIR   = $(ROOT_PMCS_FW_DIR_$(CLASS))
502 ROOT_QLC_FW_DIR    = $(ROOT_QLC_FW_DIR_$(CLASS))
503 ROOT_EMLXS_FW_DIR  = $(ROOT_EMLXS_FW_DIR_$(CLASS))
504 ROOT_NLMISC_DIR    = $(ROOT_NLMISC_DIR_$(CLASS))
505 ROOT_MACH_DIR      = $(ROOT_MACH_DIR_$(CLASS))
506 ROOT_CPU_DIR       = $(ROOT_CPU_DIR_$(CLASS))
507 ROOT_TOD_DIR       = $(ROOT_TOD_DIR_$(CLASS))
508 ROOT_FONT_DIR      = $(ROOT_FONT_DIR_$(CLASS))
509 ROOT_DACF_DIR      = $(ROOT_DACF_DIR_$(CLASS))
510 ROOT_CRYPTODIR     = $(ROOT_CRYPTODIR_$(CLASS))
511 ROOT_MAC_DIR       = $(ROOT_MAC_DIR_$(CLASS))
512 ROOT_KICONV_DIR    = $(ROOT_KICONV_DIR_$(CLASS))

514 ROOT_MOD_DIRS_32  = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
515 ROOT_MOD_DIRS_32  = $(ROOT_BRAND_DIR_32) $(ROOT_DRV_DIR_32)
516 ROOT_MOD_DIRS_32 += $(ROOT_EXEC_DIR_32) $(ROOT_DTRACE_DIR_32)
517 ROOT_MOD_DIRS_32 += $(ROOT_FS_DIR_32) $(ROOT_SCHED_DIR_32)
518 ROOT_MOD_DIRS_32 += $(ROOT_STRMOD_DIR_32) $(ROOT_SYS_DIR_32)
519 ROOT_MOD_DIRS_32 += $(ROOT_IPP_DIR_32) $(ROOT_SOCKET_DIR_32)
520 ROOT_MOD_DIRS_32 += $(ROOT_MISC_DIR_32) $(ROOT_MACH_DIR_32)
521 ROOT_MOD_DIRS_32 += $(ROOT_KGSS_DIR_32)
522 ROOT_MOD_DIRS_32 += $(ROOT_SCSI_VHCI_DIR_32)
523 ROOT_MOD_DIRS_32 += $(ROOT_PMCS_FW_DIR_32)

```

```

524 ROOT_MOD_DIRS_32 += $(ROOT_QLC_FW_DIR_32)
525 ROOT_MOD_DIRS_32 += $(ROOT_EMLXS_FW_DIR_32)
526 ROOT_MOD_DIRS_32 += $(ROOT_CPU_DIR_32) $(ROOT_FONT_DIR_32)
527 ROOT_MOD_DIRS_32 += $(ROOT_TOD_DIR_32) $(ROOT_DACF_DIR_32)
528 ROOT_MOD_DIRS_32 += $(ROOT_CRYPTODIR_32) $(ROOT_MAC_DIR_32)
529 ROOT_MOD_DIRS_32 += $(ROOT_KICONV_DIR_32)

531 USR_MOD_DIR = $(ROOT)/usr/kernel

533 USR_DRV_DIR_32 = $(USR_MOD_DIR)/drv
534 USR_EXEC_DIR_32 = $(USR_MOD_DIR)/exec
535 USR_FS_DIR_32 = $(USR_MOD_DIR)/fs
536 USR_SCHED_DIR_32 = $(USR_MOD_DIR)/sched
537 USR_SOCKET_DIR_32 = $(USR_MOD_DIR)/socketmod
538 USR_STRMOD_DIR_32 = $(USR_MOD_DIR)/strmod
539 USR_SYS_DIR_32 = $(USR_MOD_DIR)/sys
540 USR_MISC_DIR_32 = $(USR_MOD_DIR)/misc
541 USR_DACF_DIR_32 = $(USR_MOD_DIR)/dacf
542 USR_PCBE_DIR_32 = $(USR_MOD_DIR)/pcbe
543 USR_DTRACE_DIR_32 = $(USR_MOD_DIR)/dtrace
544 USR_BRAND_DIR_32 = $(USR_MOD_DIR)/brand

546 USR_DRV_DIR_64 = $(USR_MOD_DIR)/drv/$(SUBDIR64)
547 USR_EXEC_DIR_64 = $(USR_MOD_DIR)/exec/$(SUBDIR64)
548 USR_FS_DIR_64 = $(USR_MOD_DIR)/fs/$(SUBDIR64)
549 USR_SCHED_DIR_64 = $(USR_MOD_DIR)/sched/$(SUBDIR64)
550 USR_SOCKET_DIR_64 = $(USR_MOD_DIR)/socketmod/$(SUBDIR64)
551 USR_STRMOD_DIR_64 = $(USR_MOD_DIR)/strmod/$(SUBDIR64)
552 USR_SYS_DIR_64 = $(USR_MOD_DIR)/sys/$(SUBDIR64)
553 USR_MISC_DIR_64 = $(USR_MOD_DIR)/misc/$(SUBDIR64)
554 USR_DACF_DIR_64 = $(USR_MOD_DIR)/dacf/$(SUBDIR64)
555 USR_PCBE_DIR_64 = $(USR_MOD_DIR)/pcbe/$(SUBDIR64)
556 USR_DTRACE_DIR_64 = $(USR_MOD_DIR)/dtrace/$(SUBDIR64)
557 USR_BRAND_DIR_64 = $(USR_MOD_DIR)/brand/$(SUBDIR64)

559 USR_DRV_DIR = $(USR_DRV_DIR_$(CLASS))
560 USR_EXEC_DIR = $(USR_EXEC_DIR_$(CLASS))
561 USR_FS_DIR = $(USR_FS_DIR_$(CLASS))
562 USR_SCHED_DIR = $(USR_SCHED_DIR_$(CLASS))
563 USR_SOCKET_DIR = $(USR_SOCKET_DIR_$(CLASS))
564 USR_STRMOD_DIR = $(USR_STRMOD_DIR_$(CLASS))
565 USR_SYS_DIR = $(USR_SYS_DIR_$(CLASS))
566 USR_MISC_DIR = $(USR_MISC_DIR_$(CLASS))
567 USR_DACF_DIR = $(USR_DACF_DIR_$(CLASS))
568 USR_PCBE_DIR = $(USR_PCBE_DIR_$(CLASS))
569 USR_DTRACE_DIR = $(USR_DTRACE_DIR_$(CLASS))
570 USR_BRAND_DIR = $(USR_BRAND_DIR_$(CLASS))

572 USR_MOD_DIRS_32 = $(USR_DRV_DIR_32) $(USR_EXEC_DIR_32)
573 USR_MOD_DIRS_32 += $(USR_FS_DIR_32) $(USR_SCHED_DIR_32)
574 USR_MOD_DIRS_32 += $(USR_STRMOD_DIR_32) $(USR_SYS_DIR_32)
575 USR_MOD_DIRS_32 += $(USR_MISC_DIR_32) $(USR_DACF_DIR_32)
576 USR_MOD_DIRS_32 += $(USR_PCBE_DIR_32)
577 USR_MOD_DIRS_32 += $(USR_DTRACE_DIR_32) $(USR_BRAND_DIR_32)
578 USR_MOD_DIRS_32 += $(USR_SOCKET_DIR_32)

580 #
581 #
582 #
583 include $(SRC)/Makefile.psm

585 #
586 # The "-r" on the remove may be considered temporary, but is required
587 # while the replacement of the SUNW,SPARCstation-10,SX directory by
588 # a symbolic link is being propagated.
589 #

```

```

590 # IMPORTANT:: if you change any of these INS.mumble rules, then you MUST also
591 # change the corresponding override definitions in $CLOSED/Makefile.tonic.
592 # If you do not do this, then the closedbins build for the OpenSolaris
593 # community will break. PS, the gatekeepers will be upset too.
594 #
595 INS.slink1= $(RM) -r $@; $(SYMLINK) $(PLATFORM) $@
596 INS.slink2= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/$(@F) $@
597 INS.slink3= $(RM) -r $@; $(SYMLINK) $(IMPLEMENTED_PLATFORM) $@
598 INS.slink4= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/include $@
599 INS.slink5= $(RM) -r $@; $(SYMLINK) ../$(PLATFORM)/sbin $@
600 INS.slink6= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/lib/$(MODULE) $@
601 INS.slink7= $(RM) -r $@; $(SYMLINK) ../../$(PLATFORM)/sbin/$(@F) $@

603 #
604 # Collection of all relevant, delivered kernel modules.
605 #
606 # Note that we insist on building genunix first, because everything else
607 # unifies against it. When doing a 'make' from usr/src/uts/, we'll enter
608 # the platform directories first. These will cd into the corresponding genunix
609 # directory and build it. So genunix /shouldn't/ get rebuilt when we get to
610 # building all the kernel modules. However, due to an as-yet-unexplained
611 # problem with dependencies, sometimes it does get rebuilt, which then messes
612 # up the other modules. So we always force the issue here rather than try to
613 # build genunix in parallel with everything else.
614 #
615 PARALLEL_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
616 $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
617 $(NLMISC_KMODS) $(MACH_KMODS) $(CPU_KMODS) $(GSS_KMODS) \
618 $(MMU_KMODS) $(DACF_KMODS) $(EXPORT_KMODS) $(IPP_KMODS) \
619 $(CRYPTO_KMODS) $(PCBE_KMODS) \
620 $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
621 $(BRAND_KMODS) $(KICONV_KMODS) \
622 $(SOCKET_KMODS)

624 KMODS = $(GENUNIX_KMODS) $(PARALLEL_KMODS)

626 $(PARALLEL_KMODS): $(GENUNIX_KMODS)

633 $(CLOSED_BUILD)CLOSED_KMODS = $(CLOSED_DRV_KMODS) $(CLOSED_TOD_KMODS) \
634 $(CLOSED_MISC_KMODS) $(CLOSED_CPU_KMODS) \
635 $(CLOSED_NLMISC_KMODS) $(CLOSED_DRV_KMODS_$(CLASS))

637 LINT_KMODS = $(DRV_KMODS) $(EXEC_KMODS) $(FS_KMODS) $(SCHED_KMODS) \
638 $(TOD_KMODS) $(STRMOD_KMODS) $(SYS_KMODS) $(MISC_KMODS) \
639 $(MACH_KMODS) $(GSS_KMODS) $(DACF_KMODS) $(IPP_KMODS) \
640 $(CRYPTO_KMODS) $(PCBE_KMODS) \
641 $(DRV_KMODS_$(CLASS)) $(MISC_KMODS_$(CLASS)) $(MAC_KMODS) \
642 $(BRAND_KMODS) $(KICONV_KMODS) $(SOCKET_KMODS)

644 $(CLOSED_BUILD)CLOSED_LINT_KMODS = $(CLOSED_DRV_KMODS) $(CLOSED_TOD_KMODS) \
645 $(CLOSED_MISC_KMODS) $(CLOSED_DRV_KMODS_$(CLASS))

635 THIS_YEAR:sh= /bin/date +%Y
636 $(OBJDIR)/logsubr.o := CPPFLAGS += -DTHIS_YEAR=$(THIS_YEAR)
637 $(OBJDIR)/logsubr.ln := CPPFLAGS += -DTHIS_YEAR=$(THIS_YEAR)

639 #
640 # Files to be compiled with -xa, to generate basic block execution
641 # count data.
642 #
643 # There are several ways to compile parts of the kernel for kcov:

```

```
644 #           1) Add targets to BB_FILES here or in other Makefiles
645 #             (they must in the form of $(OBJS_DIR)/target.o)
646 #           2) setenv BB_FILES '$(XXX_OBJS:%=$(OBJS_DIR)/%)'
647 #           3) setenv BB_FILES '$(OBJECTS)'
648 #
649 #           Do NOT setenv CFLAGS -xa, as that will cause infinite recursion
650 #           in unix_bb.o
651 #
652 BB_FILES =
653 $(BB_FILES) := XAOPT = -xa
654
655 #
656 #           The idea here is for unix_bb.o to be in all kernels except the
657 #           kernel which actually gets shipped to customers. In practice,
658 #           $(RELEASE_BUILD) is on for a number of the late beta and fcs builds.
659 #
660 CODE_COVERAGE=
661 $(RELEASE_BUILD)CODE_COVERAGE:sh=      echo \\043
662 $(CODE_COVERAGE)$(OBJS_DIR)/unix_bb.o := CPPFLAGS      += -DKCOV
663 $(CODE_COVERAGE)$(OBJS_DIR)/unix_bb.ln := CPPFLAGS      += -DKCOV
664
665 #
666 #           Do not let unix_bb.o get compiled with -xa!
667 #
668 $(OBJS_DIR)/unix_bb.o := XAOPT =
669
670 #
671 # Privilege files
672 #
673 PRIVS_AWK = $(SRC)/uts/common/os/privs.awk
674 PRIVS_DEF = $(SRC)/uts/common/os/priv_defs
675
676 #
677 # USB device data
678 #
679 USBDEVS_AWK = $(SRC)/uts/common/io/usb/usbdevs2h.awk
680 USBDEVS_DATA = $(SRC)/uts/common/io/usb/usbdevs
```

```

*****
1482 Thu Aug 15 11:59:23 2013
new/usr/src/uts/common/nfs/Makefile
4027 remove CLOSED_BUILD
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # uts/common/nfs/Makefile
26 #
27 # include global definitions
28 #
29 include ../../../../Makefile.master

31 HDRS=   export.h      lm.h                \
32         mount.h      nfs.h                nfssys.h          nfs_acl.h        \
33         nfs_clnt.h    nfs_log.h          nfs_sec.h         nfs4.h           \
34         nfs4_attr.h  nfs4_clnt.h     rnode.h          rnode4.h        \
35         nfs4_kprot.h  nfs4_db_impl.h  nfs4_idmap_impl.h \
36         nfsid_map.h   auth.h          nfs_cmd.h

38 $(CLOSED_BUILD)CLOSEDHDRS=   lm_impl.h      lm_nlm.h      lm_server.h

40 ALLHDRS= $(HDRS) $(CLOSEDHDRS)

38 ROOTDIRS= $(ROOT)/usr/include/nfs

40 ROOTHDRS= $(HDRS:%=$(ROOTDIRS)/%)
44 ROOTHDRS= $(ALLHDRS:%=$(ROOTDIRS)/%)

42 CHECKHDRS= $(HDRS:%.h=%.check)
46 CHECKHDRS= $(HDRS:%.h=%.check) \
47             $(CLOSEDHDRS:%.h=$(CLOSED)/uts/common/nfs/%.check)

44 $(ROOTDIRS)/%: %
45             $(INS.file)

52 $(ROOTDIRS)/%: $(CLOSED)/uts/common/nfs/%
53             $(INS.file)

47 .KEEP_STATE:

49 .PARALLEL: $(CHECKHDRS)

51 install_h: $(ROOTDIRS) $(ROOTHDRS)

```

```

53 $(ROOTDIRS):
54     $(INS.dir)

56 check: $(CHECKHDRS)

```

new/usr/src/uts/i86pc/Makefile

1

```
*****
4077 Thu Aug 15 11:59:23 2013
new/usr/src/uts/i86pc/Makefile
4027 remove CLOSED_BUILD
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # This makefile drives the production of all implementation architecture
26 # dependent modules for the i86pc architecture.
27 #
28 #
29 UTSBASE = ..
30 #
31 include Makefile.i86pc
32 #
33 #
34 # The following are x86 specific (rather than i86pc) specific modules
35 # which are required for the i86pc kernel to completely lint. They are
36 # not involved in the build in any other way. In order to minimize
37 # build time, it is assumed that they are up to date.
38 #
39 INTEL_LIB_DIR = $(UTSBASE)/intel/lint-libs/$(OBJSDIR)
40 #
41 INTEL_LINTS = genunix
42 #
43 LINT_LIBS = $(LINT_LIB) \
44 $(GENUNIX_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
45 $(PARALLEL_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
46 $(CLOSED_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
47 $(INTEL_LINTS:%=$(INTEL_LIB_DIR)/llib-1%.ln)
48 #
49 #
50 #
51 def := TARGET= def
52 all := TARGET= all
53 install := TARGET= install
54 install_h := TARGET= install_h
55 clean := TARGET= clean
56 clobber := TARGET= clobber
57 lint := TARGET= lint
58 lintlib := TARGET= lintlib
59 machmodlintlib := TARGET= modlintlib
60 modlist := TARGET= modlist
```

new/usr/src/uts/i86pc/Makefile

2

```
61 modlist modlist.intel := NO_STATE=-K $$MODSTATE$$$$
62 clean.lint := TARGET= clean.lint
63 check := TARGET= check
64 #
65 .KEEP_STATE:
66 #
67 .PARALLEL: $(PARALLEL_KMODS) $(XMODS) modlist modlist.intel
68 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
69 modlist modlist.intel
70 #
71 INITIAL_TARGETS = \
72 genassym \
73 unix \
74 cpu/scripts
75 #
76 def all clean clobber clean.lint: setup genassym unix .WAIT \
77 $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
78 $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
79 $(CLOSED_KMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)
80 #
81 install: install_platforms setup genassym unix .WAIT \
82 $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
83 $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
84 $(CLOSED_KMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)
85 #
86 # list the modules under i86pc.
87 modlist: unix $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
88 $(KMODS) $(XMODS) $(IMPLEMENTATIONS)
89 $(CLOSED_KMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)
90 #
91 # list the modules for Install -k i86pc.
92 modlist.karch: modlist modlist.intel
93 #
94 modlist.intel:
95 @cd $(SRC)/uts/intel; pwd; $(MAKE) $(NO_STATE) modlist
96 #
97 lintlib: unix
98 #
99 modlintlib: $(KMODS)
100 modlintlib: $(KMODS) $(CLOSED_KMODS)
101 #
102 genassym unix $(KMODS): FRC
103 @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
104 #
105 setup: FRC
106 @cd cpu/scripts; pwd; $(MAKE) $(TARGET)
107 #
108 $(IMPLEMENTATIONS): FRC
109 @cd $@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)
110 #
111 $(XMODS): FRC
112 @if [ -f $@/Makefile ]; then \
113 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
114 else \
115 true; \
116 fi
117 #
118 $(CLOSED_KMODS): FRC
119 cd $(CLOSED)/uts/i86pc/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
120 #
121 $(CLOSED_XMODS): FRC
122 @if [ -f $(CLOSED)/uts/i86pc/$@/Makefile ]; then \
123 cd $(CLOSED)/uts/i86pc/$@; pwd; \
124 $(MAKE) $(NO_STATE) $(TARGET); \
125 else \
126 true; \
127 fi
```


new/usr/src/uts/i86pc/Makefile

3

```
109 install_h check:      $(IMPLEMENTATIONS) FRC
110     @cd sys; pwd; $(MAKE) $(TARGET)

112 #
113 # Definitions for the /platform directory aliases.
114 # Currently none for i86pc.
115 #
116 PLAT_LINKS      =

118 #
119 # Make the /platform directories.  This is hardwired here because
120 # the first stage of the project (KBI) only implements the userland
121 # changes, but the only reasonable place to record the aliases is
122 # here in kernel land.
123 #
124 install_platforms:     $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
125                        $(ROOT_PLAT_LINKS) $(USR_PLAT_LINKS) \
126                        $(OEM_USR_PLAT_LINKS)

128 #
129 #     Full kernel lint target.
130 #
131 LINT_TARGET      = globallint

133 # workaround for multiply defined errors
134 globallint := LINTFLAGS += -erroff=E_NAME_MULTIPLY_DEF2

136 globallint:
137     @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
138     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

140 lint:    lintlib .WAIT modlintlib .WAIT $(INTEL_LINTS) $(LINT_DEPS) \
141         $(IMPLEMENTATIONS)

143 $(INTEL_LINTS): FRC
144     @cd $(UTSBASE)/intel/$@; pwd; $(MAKE) modlintlib

146 include ../Makefile.targ

148 #
149 # Cross-reference customization: build a cross-reference over all of the
150 # i86pc-related directories.
151 #
152 XRDIRS = ../i86pc ../intel ../common
153 $(CLOSED_BUILD)XRDIRS += ../../../../closed/uts/intel ../../../../closed/uts/common

154 XRPRUNE = sun4u sun4

156 cscope.out tags: FRC
157     $(XREF) -x $@
```

new/usr/src/uts/i86pc/Makefile.i86pc.shared

1

```
*****
9027 Thu Aug 15 11:59:23 2013
new/usr/src/uts/i86pc/Makefile.i86pc.shared
4027 remove CLOSED_BUILD
*****
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 # uts/i86pc/Makefile.i86pc
24 #
25 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
26 #
27 #
28 # This makefile contains the common definitions for the i86pc unix
29 # and all i86pc implementation architecture dependent modules.
30 #
31 #
32 #
33 # Machine type (implementation architecture):
34 #
35 PLATFORM = i86pc
36 #
37 #
38 # uname -m value
39 #
40 UNAME_M = $(PLATFORM)
41 #
42 #
43 # Definitions for the platform-specific /platform directories.
44 #
45 # IMPLEMENTATIONS is used to designate i86pc machines which have
46 # platform specific modules. All code specific to a given implementation
47 # resides in the appropriately named subdirectory. This requires
48 # these platforms to have their own Makefiles to define ROOT_PLAT_DIRS,
49 # USR_PLAT_DIRS, etc.
50 #
51 IMPLEMENTATIONS = i86hvm
52 #
53 #
54 # Everybody needs to know how to build modstubs.o and to locate unix.o
55 #
56 UNIX_DIR = $(UTSBASE)/$(PLATFORM)/unix
57 GENLIB_DIR = $(UTSBASE)/intel/genunix
58 MODSTUBS_DIR = $(UNIX_DIR)
59 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
60 LINTS_DIR = $(OBJS_DIR)
61 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJS_DIR)
```

new/usr/src/uts/i86pc/Makefile.i86pc.shared

2

```
62 GEN_LINT_LIB_DIR = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)
63 #
64 LINT32_DIRS = $(LINT32_BUILDS:%=$(UTSBASE)/$(PLATFORM)/lint-libs/%)
65 LINT32_FILES = $(LINT32_DIRS:%=/llib-1$(MODULE).ln)
66 #
67 DTRACESTUBS_O = $(OBJS_DIR)/dtracestubs.o
68 DTRACESTUBS = $(OBJS_DIR)/libdtracestubs.so
69 #
70 SYM_MOD = $(OBJS_DIR)/unix.sym
71 #
72 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
73 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
74 GENLIB = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
75 LINT_LIB = $(LINT_LIB_DIR)/llib-lunix.ln
76 DBOOT_LINT_LIB = $(LINT_LIB_DIR)/llib-lboot.ln
77 GEN_LINT_LIB = $(GEN_LINT_LIB_DIR)/llib-lgenunix.ln
78 #
79 #
80 # Include the makefiles which define build rule templates, the
81 # collection of files per module, and a few specific flags. Note
82 # that order is significant, just as with an include path. The
83 # first build rule template which matches the files name will be
84 # used. By including these in order from most machine dependent
85 # to most machine independent, we allow a machine dependent file
86 # to be used in preference over a machine independent version
87 # (Such as a machine specific optimization, which preserves the
88 # interfaces.)
89 #
90 include $(UTSTREE)/$(PLATFORM)/Makefile.files
91 include $(UTSTREE)/intel/Makefile.files
92 include $(UTSTREE)/common/Makefile.files
93 #
94 #
95 # Include machine independent rules. Note that this does not imply
96 # that the resulting module from rules in Makefile.uts is machine
97 # independent. Only that the build rules are machine independent.
98 #
99 include $(UTSBASE)/Makefile.uts
100 #
101 #
102 # Define supported builds
103 #
104 DEF_BUILDS = $(DEF_BUILDS64) $(DEF_BUILDS32)
105 ALL_BUILDS = $(ALL_BUILDS64) $(ALL_BUILDS32)
106 #
107 #
108 # x86 or amd64 inline templates
109 #
110 INLINES_32 = $(UTSBASE)/intel/ia32/ml/ia32.il \
111 $(UTSBASE)/$(PLATFORM)/ml/ia32.il
112 INLINES_64 = $(UTSBASE)/intel/amd64/ml/amd64.il \
113 $(UTSBASE)/$(PLATFORM)/ml/amd64.il
114 INLINES += $(INLINES_$(CLASS))
115 #
116 #
117 # kernel-specific optimizations; override default in Makefile.master
118 #
119 #
120 CFLAGS_XARCH_32 = $(i386_CFLAGS)
121 CFLAGS_XARCH_64 = $(amd64_CFLAGS)
122 CFLAGS_XARCH = $(CFLAGS_XARCH_$(CLASS))
123 #
124 COPTFLAG_32 = $(COPTFLAG)
125 COPTFLAG_64 = $(COPTFLAG64)
126 COPTIMIZE = $(COPTFLAG_$(CLASS))
```

```

128 CFLAGS          = $(CFLAGS_XARCH)
129 CFLAGS          += $(COPTIMIZE)
130 CFLAGS          += $(INLINES) -D_ASM_INLINES
131 CFLAGS          += $(CCMODE)
132 CFLAGS          += $(SPACEFLAG)
133 CFLAGS          += $(CCUNBOUND)
134 CFLAGS          += $(CFLAGS_uts)
135 CFLAGS          += -xstrconst

137 ASFLAGS_XARCH_32 = $(i386_ASFLAGS)
138 ASFLAGS_XARCH_64 = $(amd64_ASFLAGS)
139 ASFLAGS_XARCH    = $(ASFLAGS_XARCH_$(CLASS))

141 ASFLAGS          += $(ASFLAGS_XARCH)

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

145 #
146 #   The following must be defined for all implementations:
147 #
148 #   MAPFILE:      ld mapfile for the build of kernel/unix.
149 #   MODSTUBS:    Module stubs source file.
150 #   GENASSYM_SRC: genassym.c
151 #
152 MAPFILE          = $(UTSBASE)/$(PLATFORM)/conf/Mapfile
153 MODSTUBS         = $(UTSBASE)/intel/ia32/ml/modstubs.s
154 GENASSYM_SRC     = $(UTSBASE)/$(PLATFORM)/ml/genassym.c
155 OFFSETS_SRC     = $(UTSBASE)/$(PLATFORM)/ml/offsets.in
156 PLATFORM_OFFSETS_32 = $(UTSBASE)/$(PLATFORM)/ml/mach_offsets.in
157 PLATFORM_OFFSETS_64 = $(UTSBASE)/intel/amd64/ml/mach_offsets.in
158 PLATFORM_OFFSETS_SRC = $(PLATFORM_OFFSETS_$(CLASS))
159 KDI_OFFSETS_SRC  = $(UTSBASE)/intel/kdi/kdi_offsets.in

161 #
162 #   Define the actual specific platforms
163 #
164 MACHINE_DEFS     = -D$(PLATFORM) -D_MACHDEP

166 #
167 #   Software workarounds for hardware "features"
168 #

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

172 #
173 #   Debugging level
174 #
175 #   Special knowledge of which special debugging options effect which
176 #   file is used to optimize the build if these flags are changed.
177 #
178 #   XXX: The above could possibly be done for more flags and files, but
179 #   is left as an experiment to the interested reader. Be forewarned,
180 #   that excessive use could lead to maintenance difficulties.
181 #
182 DEBUG_DEFS_OBJ32 =
183 DEBUG_DEFS_DBG32 = -DDEBUG
184 DEBUG_DEFS_OBJ64 =
185 DEBUG_DEFS_DBG64 = -DDEBUG
186 DEBUG_DEFS       = $(DEBUG_DEFS_$(BUILD_TYPE))

188 DEBUG_COND_OBJ32 = :sh = echo \\043
189 DEBUG_COND_DBG32 =
190 DEBUG_COND_OBJ64 = :sh = echo \\043
191 DEBUG_COND_DBG64 =
192 IF_DEBUG_OBJ     = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

```

```

194 $(IF_DEBUG_OBJ)trap.o          := DEBUG_DEFS += -DTRAPDEBUG -DTRAPTRACE
195 $(IF_DEBUG_OBJ)syscall_asm.o  := DEBUG_DEFS += -DSYSCALLTRACE -DTRAPTRACE
196 $(IF_DEBUG_OBJ)syscall_asm_amd64.o := DEBUG_DEFS += -DSYSCALLTRACE -DTRAPTRACE
197 $(IF_DEBUG_OBJ)fast_trap_asm.o := DEBUG_DEFS += -DTRAPTRACE
198 $(IF_DEBUG_OBJ)interrupt.o    := DEBUG_DEFS += -DTRAPTRACE
199 $(IF_DEBUG_OBJ)intr.o         := DEBUG_DEFS += -DTRAPTRACE
200 $(IF_DEBUG_OBJ)locore.o       := DEBUG_DEFS += -DTRAPTRACE
201 $(IF_DEBUG_OBJ)mp_startup.o   := DEBUG_DEFS += -DTRAPTRACE
202 $(IF_DEBUG_OBJ)machdep.o      := DEBUG_DEFS += -DTRAPTRACE
203 $(IF_DEBUG_OBJ)exception.o    := DEBUG_DEFS += -DTRAPTRACE
204 $(IF_DEBUG_OBJ)x_call.o       := DEBUG_DEFS += -DTRAPTRACE
205 $(IF_DEBUG_OBJ)mp_call.o      := DEBUG_DEFS += -DTRAPTRACE
206 $(IF_DEBUG_OBJ)cbe.o          := DEBUG_DEFS += -DTRAPTRACE

208 #
209 #   Collect the preprocessor definitions to be associated with *all*
210 #   files.
211 #
212 ALL_DEFS          = $(MACHINE_DEFS) $(WORKAROUND_DEFS) $(DEBUG_DEFS) \
213                   $(OPTION_DEFS)
214 GENASSYM_DEFS     = $(MACHINE_DEFS) $(OPTION_DEFS) \
215                   -_gcc=-fno-eliminate-unused-debug-symbols \
216                   -_gcc=-fno-eliminate-unused-debug-types

218 #
219 # ----- TRANSITIONAL SECTION -----
220 #
221 #
222 #
223 #   Not everything which *should* be a module is a module yet. The
224 #   following is a list of such objects which are currently part of
225 #   the base kernel but should soon become kmods.
226 #
227 #   XXX: $(KMACCT_OBJS) is neither in the MT kernel nor was it ever
228 #   made into a module. If it is made MT safe before being made
229 #   into a module, it should be added to this list. It was in
230 #   this list pre ON-4.0.
231 #
232 #
233 MACH_NOT_YET_KMODS = $(AUTOCONF_OBJS)

235 #
236 # ----- END OF TRANSITIONAL SECTION -----
237 #
238 #
239 #
240 #   The kernels modules which are "implementation architecture"
241 #   specific for this machine are enumerated below. Note that most
242 #   of these modules must exist (in one form or another) for each
243 #   architecture.
244 #
245 #   Machine Specific Driver Modules (/kernel/drv)
246 #   DRV_KMODS are built both 32-bit and 64-bit
247 #   DRV_KMODS_32 are built only 32-bit
248 #   DRV_KMODS_64 are built only 64-bit
249 #
250 DRV_KMODS        += rootnex
251 DRV_KMODS        += isa
252 DRV_KMODS        += pcplusmp
253 DRV_KMODS        += apix
254 DRV_KMODS        += cpc
255 DRV_KMODS        += pci
256 DRV_KMODS        += npe
257 DRV_KMODS        += pci-ide
258 DRV_KMODS        += xsvc
259 DRV_KMODS        += tzmon

```

new/usr/src/uts/i86pc/Makefile.i86pc.shared

5

```
260 DRV_KMODS      += acpi_drv
261 DRV_KMODS      += acpinex
262 DRV_KMODS      += amd_iommu
263 DRV_KMODS      += dr
264 DRV_KMODS      += ioat
265 DRV_KMODS      += fipec
267 DRV_KMODS      += cpudrv

270 #
271 # Platform Power Modules
272 #
273 DRV_KMODS      += ppm acpippm

275 #
276 #      CPU Modules
277 #
278 CPU_KMODS      += amd_opteron
279 CPU_KMODS      += generic_cpu
280 CPU_KMODS      += authenticamd
281 CPU_KMODS      += genuineintel

283 #
284 # Don't build some of these for OpenSolaris, since they will be
285 # replaced by binaries that are signed by Sun Release Engineering.
286 #
287 $(CLOSED_BUILD)CLOSED_CPU_KMODS += intel_nhmex

289 #
290 #      Exec Class Modules (/kernel/exec):
291 #
292 EXEC_KMODS     +=

294 #
295 #      Scheduling Class Modules (/kernel/sched):
296 #
297 SCHED_KMODS    +=

299 #
300 #      File System Modules (/kernel/fs):
301 #
302 FS_KMODS       +=

304 #
305 #      Streams Modules (/kernel/strmod):
306 #
307 STRMOD_KMODS   +=

309 #
310 #      'System' Modules (/kernel/sys):
311 #
312 SYS_KMODS      +=

314 #
315 #      'Misc' Modules (/kernel/misc):
316 #
317 MISC_KMODS     += gfx_private pcie
318 MISC_KMODS     += acpidev
319 MISC_KMODS     += drmach_acpi

321 #
322 #      'Dacf' modules (/kernel/dacf)
323 #
324 DACF_KMODS     += consconfig_dacf
```

new/usr/src/uts/i86pc/Makefile.i86pc.shared

6

```
320 #
321 #      'Mach' Modules (/kernel/mach):
322 #
323 MACH_KMODS     += uppc

325 #
326 #      CPR Misc Module.
327 #
328 MISC_KMODS     += cpr
```

new/usr/src/uts/i86xpv/Makefile

1

```
*****
3964 Thu Aug 15 11:59:23 2013
new/usr/src/uts/i86xpv/Makefile
4027 remove CLOSED_BUILD
*****
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 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Copyright 2013 Nexenta Systems, Inc. All rights reserved.
26 #
27 # This makefile drives the production of all implementation architecture
28 # dependent modules for the i86xpv architecture.
29 #
31 UTSBASE = ..
33 include Makefile.i86xpv
35 #
36 # The following are x86 specific (rather than i86pc) specific modules
37 # which are required for the i86pc kernel to completely lint. They are
38 # not involved in the build in any other way. In order to minimize
39 # build time, it is assumed that they are up to date.
40 #
41 INTEL_LIB_DIR = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)
43 INTEL_LINTS = genunix
45 LINT_LIBS = $(LINT_LIB) \
46 $(GENUNIX_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
47 $(PARALLEL_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
48 $(CLOSED_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
49 $(INTEL_LINTS:%=$(INTEL_LIB_DIR)/llib-1%.ln)
50 #
51 #
52 #
53 def := TARGET= def
54 all := TARGET= all
55 install := TARGET= install
56 clean := TARGET= clean
57 clobber := TARGET= clobber
58 lint := TARGET= lint
59 lintlib := TARGET= lintlib
60 machmodlintlib := TARGET= modlintlib
```

new/usr/src/uts/i86xpv/Makefile

2

```
61 modlist := TARGET= modlist
62 modlist modlist.intel := NO_STATE= -K $$MODSTATE$$$$
63 clean.lint := TARGET= clean.lint
64 check := TARGET= check
66 .KEEP_STATE:
68 .PARALLEL: $(PARALLEL_KMODS) $(XMODS) modlist modlist.intel
69 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
70 modlist modlist.intel
71 INITIAL_TARGETS = \
72 genassym \
73 unix \
74 cpu/scripts
75 def all clean clobber clean.lint: setup genassym unix .WAIT \
76 $(KMODS) $(XMODS)
77 $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS)
78 install: install_platforms setup genassym unix .WAIT \
79 $(KMODS) $(XMODS)
80 $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS)
81 # list the modules under i86xpv.
82 modlist: unix $(KMODS) $(XMODS)
83 modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS)
84 # list the modules for Install -k i86xpv.
85 modlist.karch: modlist modlist.intel
87 modlist.intel:
88 @cd $(SRC)/uts/intel; pwd; $(MAKE) $(NO_STATE) modlist
90 lintlib: unix
92 modlintlib: $(KMODS)
93 modlintlib: $(KMODS) $(CLOSED_KMODS)
94 genassym unix $(KMODS): FRC
95 @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
97 setup: FRC
98 @cd cpu/scripts; pwd; $(MAKE) $(TARGET)
100 $(XMODS): FRC
101 @if [ -f $@/Makefile ]; then \
102 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
103 else \
104 true; \
105 fi
109 $(CLOSED_KMODS): FRC
110 cd $(CLOSED)/uts/i86xpv/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
112 $(CLOSED_XMODS): FRC
113 @if [ -f $(CLOSED)/uts/i86xpv/$@/Makefile ]; then \
114 cd $(CLOSED)/uts/i86xpv/$@; pwd; \
115 $(MAKE) $(NO_STATE) $(TARGET); \
116 else \
117 true; \
118 fi
107 install_h check: FRC
108 @cd sys; pwd; $(MAKE) $(TARGET)
```

```
110 #
111 # Definitions for the /platform directory aliases.
112 # Currently none for i86xpv.
113 #
114 PLAT_LINKS      =

116 #
117 # Make the /platform directories. This is hardwired here because
118 # the first stage of the project (KBI) only implements the userland
119 # changes, but the only reasonable place to record the aliases is
120 # here in kernel land.
121 #
122 install_platforms: $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
123                   $(ROOT_PLAT_LINKS) $(USR_PLAT_LINKS) \
124                   $(OEM_USR_PLAT_LINKS)

126 #
127 # Full kernel lint target.
128 #
129 LINT_TARGET      = globallint

131 # workaround for multiply defined errors
132 globallint := LINTFLAGS += -erroff=E_NAME_MULTIPLY_DEF2

134 globallint:
135     @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
136     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

138 lint: lintlib .WAIT modlintlib .WAIT $(INTEL_LINTS) $(LINT_DEPS)

140 $(INTEL_LINTS): FRC
141     @cd $(UTSBASE)/intel/$@; pwd; $(MAKE) modlintlib

143 FRC:

145 include ../Makefile.targ

147 #
148 # Cross-reference customization: build a cross-reference over all of the
149 # i86pc-related directories.
150 #
151 XRDIRS = ../i86xpv ../i86pc ../intel ../common
152 SHARED_XRDIRS = ../i86xpv ../i86pc ../intel ../common
153 XRDIRS = $(SHARED_XRDIRS)
154 CLOSED_XRDIRS1 = $(SHARED_XRDIRS:../%=../% ../../../closed/uts/%)
155 CLOSED_XRDIRS2 = $(CLOSED_XRDIRS1:../../closed/uts/i86pc=)
156 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS2:../../closed/uts/i86xpv=)
157 XRPRUNE = sun4v sun4u sun4 sfmmu sparc

154 cscope.out tags: FRC
155     $(XREF) -x $@
```

new/usr/src/uts/intel/Makefile

1

```
*****
4453 Thu Aug 15 11:59:23 2013
new/usr/src/uts/intel/Makefile
4027 remove CLOSED_BUILD
*****
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 # uts/intel/Makefile
22 #
23 # Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 # This makefile drives the production of all implementation architecture
26 # independent modules for Intel processors.

28 UTSBASE = ..

30 include Makefile.intel

32 LINT_KMODS_X1 = $(LINT_KMODS:nsmb=)
33 LINT_KMODS_X2 = $(LINT_KMODS_X1:smbfs=)
34 LINT_KMODLIBS = $(LINT_KMODS_X2:e1000g=)
35 LINT_LIBS = $(LINT_LIB) $(GEN_LINT_LIB) \
36 $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln)
37 $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
38 $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)

39 $(CLOSED_BUILD)LINT_LIBS += $(SVVS_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)
40 $(CLOSED_BUILD)LINT_CLOSED_XMOD4 = $(CLOSED_XMODS:bnx=)
41 $(CLOSED_BUILD)LINT_CLOSED_XMOD3 = $(LINT_CLOSED_XMOD4:bnxe=)
42 $(CLOSED_BUILD)LINT_CLOSED_XMOD2 = $(LINT_CLOSED_XMOD3:lsimega=)
43 $(CLOSED_BUILD)LINT_CLOSED_XMOD1 = $(LINT_CLOSED_XMOD2:adpu320=)
44 $(CLOSED_BUILD)LINT_LIBS += $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln)

38 #
39 # dprov is delivered in the SUNWcryptpoint package.
40 #
41 DRV_KMODS += dprov

43 #
44 #
45 def := TARGET= def
46 def.prereq := TARGET= def
47 all := TARGET= all
48 all.prereq := TARGET= all
49 install := TARGET= install
50 install.prereq := TARGET= all
51 clean := TARGET= clean
52 clobber := TARGET= clobber
```

new/usr/src/uts/intel/Makefile

2

```
53 lint := TARGET= lint
54 lint.prereq := TARGET= lint
55 modlintlib := TARGET= modlintlib
56 modlist := TARGET= modlist
57 modlist := NO_STATE= -K $$MODSTATE$$$
58 clean.lint := TARGET= clean.lint
59 check := TARGET= check
60 install_h := TARGET= install_h
61 install_h.prereq := TARGET= install_h

63 .KEEP_STATE:

65 .PARALLEL: $(PARALLEL_KMODS) $(XMODS) config $(LINT_DEPS)
73 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(SVVS) $(XMODS) \
74 $(CLOSED_XMODS) config $(LINT_DEPS)

67 def all install clean clobber modlist: $(KMODS) $(XMODS) config
76 def all install clean clobber modlist: $(KMODS) $(CLOSED_KMODS) \
77 $(SVVS) $(XMODS) $(CLOSED_XMODS) config

70 #
71 # Privilege constants
72 #
73 # NOTE: The rules for generating priv_const.c file are shared between all
74 # processor architectures and should be kept in sync. If they are changed in
75 # this file make sure that sparc rules are updated as well.
76 #
77 PRIVS_C = $(SRC)/uts/common/os/priv_const.c

79 $(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
80 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=$@

82 #
83 # Prerequisites
84 #
85 # The uts/Makefile defines build parallelism for x86 platforms such that i86pc,
86 # i86xpv and intel are all built in parallel. This requires building certain
87 # parts before the parallel build can start. The uts/Makefile appends the
88 # 'prereq' string to the original target and executes this Makefile to build
89 # any prerequisites needed before the full parallel build can start. After that
90 # make continues with normal targets.
91 #
92 # Any build prerequisites for x86 builds should be described here.
93 #
94 # genassym is used to build intel/dtrace and genunix, so it should be built
95 # first.
96 #
97 # priv_const.c is required to build genunix.
98 #
99 # genunix is used by everyone to ctf-merge with. Genunix is CTF-merged with
100 # intel/ip so as a side effect this dependency builds intel/ip as part of the
101 # prerequisites.
102 #
103 # intel/dtrace depends on i86pc/genassym, so we need to build both
104 # i86pc/genassym and intel/genassym.
105 #
106 all.prereq install.prereq def.prereq: genunix FRC
107 @cd ../i86pc/genassym; pwd; $(MAKE) $(@:%.prereq=%)

109 #
110 # i86pc lint libraries should be built first
111 #
112 lint.prereq: FRC
113 @cd ../i86pc; pwd; $(MAKE) $(NO_STATE) lint
```

new/usr/src/uts/intel/Makefile

3

```
115 #
116 # Nothing to do for any other prerequisite targets.
117 #
118 %.prereq:

120 genunix: $(PRIVS_C)

122 modlintlib clean.lint: $(LINT_KMODS) $(XMODS)
132 modlintlib clean.lint: $(LINT_KMODS) $(CLOSED_LINT_KMODS) $(SVVS) \
133     $(XMODS) $(CLOSED_XMODS)

124 $(KMODS) $(SUBDIRS) config:      FRC
125     @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

138 $(CLOSED_KMODS):      FRC
139     cd $(CLOSED)/uts/intel/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

127 $(XMODS):      FRC
128     @if [ -f $@/Makefile ]; then \
129         cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
130     else \
131         true; \
132     fi

148 $(SVVS) $(CLOSED_XMODS):      FRC
149     @if [ -f $(CLOSED)/uts/intel/$@/Makefile ]; then \
150         cd $(CLOSED)/uts/intel/$@; pwd; \
151         $(MAKE) $(NO_STATE) $(TARGET); \
152     else \
153         true; \
154     fi

134 install_h check:      FRC
135     @cd sys; pwd; $(MAKE) $(TARGET)
136     @cd asm; pwd; $(MAKE) $(TARGET)
137     @cd ia32/sys; pwd; $(MAKE) $(TARGET)
138     @cd amd64/sys; pwd; $(MAKE) $(TARGET)

140 #
141 # Work-around to disable acpica global crosscheck lint warnings
142 #
143 LGREP.intel = grep -v 'intel/io/acpica'

145 #
146 # Full kernel lint target.
147 #
148 LINT_TARGET = globalint

150 # workaround for multiply defined errors
151 globalint := LINTFLAGS += -erroff=E_NAME_MULTIPLY_DEF2

153 globalint:
154     @pwd
155     @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
156     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.intel) | $(LGREP.2)

158 lint:  modlintlib .WAIT $(LINT_DEPS)

160 include ../Makefile.targ
```



```

*****
6816 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/Makefile.files
4027 remove CLOSED_BUILD
*****
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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012, Joyent, Inc. All rights reserved.
25 #
26 #
27 #
28 # This Makefile defines all file modules and build rules for the
29 # directory uts/intel and its children. These are the source files which
30 # are specific to x86 processor architectures.
31 #
32 #
33 #
34 # Core (unix) objects
35 #
36 CORE_OBJS += \
37 arch_kdi.o \
38 copy.o \
39 copy_subr.o \
40 cpc_subr.o \
41 ddi_arch.o \
42 ddi_i86.o \
43 ddi_i86_asm.o \
44 desctbls.o \
45 desctbls_asm.o \
46 exception.o \
47 float.o \
48 fmsmb.o \
49 fpu.o \
50 i86_subr.o \
51 lock_prim.o \
52 ovbcopy.o \
53 polled_io.o \
54 sseblk.o \
55 sundep.o \
56 swtch.o \
57 sysi86.o
58 #
59 #
60 # 64-bit multiply/divide compiler helper routines
61 # used only for ia32

```

```

62 #
63 #
64 SPECIAL_OBJS_32 += \
65 muldiv.o
66 #
67 #
68 # Generic-unix Module
69 #
70 GENUNIX_OBJS += \
71 archdep.o \
72 getcontext.o \
73 install_utrap.o \
74 lwp_private.o \
75 prom_enter.o \
76 prom_exit.o \
77 prom_panic.o \
78 sendsig.o \
79 syscall.o
80 #
81 #
82 #
83 # PROM Routines
84 #
85 GENUNIX_OBJS += \
86 prom_env.o \
87 prom_emul.o \
88 prom_getchar.o \
89 prom_init.o \
90 prom_node.o \
91 prom_printf.o \
92 prom_prop.o \
93 prom_putchar.o \
94 prom_reboot.o \
95 prom_version.o
96 #
97 #
98 # file system modules
99 #
100 CORE_OBJS += \
101 prmachdep.o
102 #
103 #
104 # ZFS file system module
105 #
106 ZFS_OBJS += \
107 spa_boot.o
108 #
109 #
110 # Decompression code
111 #
112 CORE_OBJS += decompress.o
113 #
114 #
115 # Microcode utilities
116 #
117 CORE_OBJS += ucode_utils.o
118 #
119 #
120 # Driver modules
121 #
122 AGPGART_OBJS += agpgart.o agp_kstat.o
123 AGPTARGET_OBJS += agptarget.o
124 AMD64GART_OBJS += amd64_gart.o
125 ARCMSR_OBJS += arcmsr.o
126 ATA_OBJS += $(GHD_OBJS) ata_blacklist.o ata_common.o ata_disk.o \
127 ata_dma.o atapi.o atapi_fsm.o ata_debug.o \

```

```

128         sil3xxx.o
129 BSCBUS_OBJS += bscbus.o
130 BSCV_OBJS += bscv.o
131 CMDK_OBJS += cmdk.o
132 CMLB_OBJS += cmlb.o
133 CPUNEX_OBJS += cpunex.o
134 DADK_OBJS += dadk.o
135 DCOPY_OBJS += dcopy.o
136 DNET_OBJS += dnet.o dnet_mii.o
137 FD_OBJS += fd.o
138 GDA_OBJS += gda.o
139 GHD_OBJS += ghd.o ghd_debug.o ghd_dma.o ghd_queue.o ghd_scscsa.o \
140         ghd_scscsi.o ghd_timer.o ghd_waitq.o ghd_gcscmd.o
141 I915_OBJS += i915_dma.o i915_drv.o i915_irq.o i915_mem.o \
142         i915_gem.o i915_gem_debug.o i915_gem_tiling.o
143 NSKERN_OBJS += nsc_asm.o
144 PCICFG_OBJS += pcicfg.o
145 PCI_PCINEXUS_OBJS += pci_pci.o
146 PCIEB_OBJS += pcieb_x86.o
147 PIT_BEEP_OBJS += pit_beep.o
148 POWER_OBJS += power.o
149 PCI_AUTOCONFIG_OBJS += pci_autoconfig.o pci_boot.o pcie_nvidia.o \
150         pci_memlist.o pci_resource.o
151 RADEON_OBJS += r300_cmdbuf.o radeon_cp.o radeon_drv.o \
152         radeon_state.o radeon_irq.o radeon_mem.o
153 SD_OBJS += sd.o sd_xbuf.o

155 HECI_OBJS += \
156         heci_init.o \
157         heci_intr.o \
158         heci_interface.o \
159         io_heci.o \
160         heci_main.o

162 STRATEGY_OBJS += strategy.o
163 UCODE_OBJS += ucode_drv.o
164 VGATEXT_OBJS += vgatext.o vgasubr.o

166 #
167 #       Kernel linker
168 #
169 KRTLDD_OBJS += \
170         bootrd.o \
171         ufsops.o \
172         hsf.o \
173         doreloc.o \
174         kobj_boot.o \
175         kobj_convrelstr.o \
176         kobj_crt.o \
177         kobj_isa.o \
178         kobj_reloc.o

180 #
181 #       misc. modules
182 #
183 ACPICA_OBJS += dbcmds.o dbdisply.o \
184         dbexec.o dbfileio.o dbhistory.o dbinput.o dbstats.o \
185         dbutils.o dbxface.o evevent.o evgpe.o evgpeblk.o \
186         evmisc.o evregion.o evrgnini.o evsci.o evxface.o \
187         evxfevnt.o evxfreg.o hwacpi.o hwgpe.o hwregs.o \
188         hwsleep.o hwtimer.o dsfield.o dsinit.o dsmethod.o \
189         dsnthdat.o dsobject.o dsopcode.o dsutils.o dswexec.o \
190         dswload.o dswscope.o dswstate.o exconfig.o exconvrt.o \
191         excreate.o exdump.o exfield.o exfldio.o exmisc.o \
192         exmutex.o exnames.o exoparg1.o exoparg2.o exoparg3.o \
193         exoparg6.o exprep.o exregion.o exresnte.o exresolv.o \

```

```

194         exresop.o exstore.o exstoren.o exstorob.o exsystem.o \
195         exutils.o psargs.o psopcode.o psparse.o psscope.o \
196         pstree.o psutils.o pswalk.o psxface.o nsaccess.o \
197         nsalloc.o nsdump.o nsdumpdv.o nseval.o nsinit.o \
198         nsload.o nsnames.o nsobject.o nsparse.o nssearch.o \
199         nsutils.o nswalk.o nsxfeval.o nsxfname.o nsxfobj.o \
200         rsaddr.o rscal.o rscrc.o rsdump.o \
201         rsinfo.o rsio.o rsirq.o rslst.o rsmemory.o rsmisc.o \
202         rsutils.o rsxface.o tbfadt.o tbfnd.o tbinstal.o \
203         tbutils.o tbxface.o tbxfroot.o \
204         utalloc.o utclib.o utcopy.o utdebug.o utdelete.o \
205         uteval.o utglobal.o utinit.o utmath.o utmisc.o \
206         utobject.o utresrc.o utxface.o acpica.o acpi_enum.o \
207         master_ops.o osl.o osl_ml.o acpica_ec.o utcache.o \
208         utmutex.o utstate.o dmbuffer.o dmnames.o dmbobject.o \
209         dmopcode.o dmresrc.o dmresrcl.o dmresrcs.o dmutils.o \
210         dmwalk.o psloop.o nspredef.o hwxface.o hwvalid.o \
211         utlock.o utids.o nsrepair.o nsrepair2.o \
212         dbmethod.o dbnames.o dsargs.o dscontrol.o dswload2.o \
213         evglock.o evgpeinit.o evgpeutil.o evxfge.o exdebug.o \
214         hwpci.o utdecode.o utosi.o utxferror.o

217 AGP_OBJS += agpmaster.o
218 FBT_OBJS += fbt.o
219 SDT_OBJS += sdt.o

221 #
222 #       AMD8111 NIC driver module
223 #
224 AMD8111S_OBJS += amd8111s_main.o amd8111s_hw.o

226 #
227 #       Pentium Performance Counter BackEnd module
228 #
229 P123_PCBE_OBJS = p123_pcbe.o

231 #
232 #       Pentium 4 Performance Counter BackEnd module
233 #
234 P4_PCBE_OBJS = p4_pcbe.o

236 #
237 #       AMD Opteron/Athlon64 Performance Counter BackEnd module
238 #
239 OPTERON_PCBE_OBJS = opteron_pcbe.o

241 #
242 #       Intel Core Architecture Performance Counter BackEnd module
243 #
244 CORE_PCBE_OBJS = core_pcbe.o

246 #
247 #       AMR module
248 #
249 AMR_OBJS = amr.o

251 #
252 #       IPMI module
253 IPMI_OBJS += ipmi_main.o ipmi.o ipmi_kcs.o

255 #
256 #       IOMMULIB module
257 #
258 IOMMULIB_OBJS = iommulib.o

```

```
260 #
261 #     Brand modules
262 #
263 SN1_BRAND_OBJS =      sn1_brand.o sn1_brand_asm.o
264 S10_BRAND_OBJS =      s10_brand.o s10_brand_asm.o

266 #
267 #     special files
268 #
269 MODSTUB_OBJ +=      \
270     modstubs.o

272 BOOTDEV_OBJS +=      \
273     bootdev.o

275 INC_PATH      += -I$(UTSBASE)/intel

278 CPR_INTEL_OBJS +=      cpr_intel.o

280 #
281 # AMD family 0xf memory controller module
282 #
283 include $(SRC)/common/mc/mc-amd/Makefile.mcamd
284 MCAMD_OBJS      += \
285     $(MCAMD_CMN_OBJS) \
286     mcamd_drv.o \
287     mcamd_dimmcfg.o \
288     mcamd_subr.o \
289     mcamd_pciecfg.o

291 #
292 # Intel Nehalem memory controller module
293 #
294 INTEL_NHM_OBJS += \
295     nhm_init.o \
296     mem_addr.o \
297     intel_nhmdrv.o \
298     nhm_pcie_cfg.o \
299     dimm_topo.o \
300     intel_nhm.o

302 #
303 # Intel 5000/5100/5400/7300 chipset memory controller hub (MCH) module
304 #
305 INTEL_NB5000_OBJS += \
306     intel_nb5000.o \
307     intel_nbdrv.o \
308     dimm_addr.o \
309     nb_pcie_cfg.o \
310     nb5000_init.o
311 #endif /* ! codereview */
```

```

*****
15739 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/Makefile.intel.shared
4027 remove CLOSED_BUILD
*****
1 # CDDL HEADER START
2 #
3 # The contents of this file are subject to the terms of the
4 # Common Development and Distribution License (the "License").
5 # You may not use this file except in compliance with the License.
6 #
7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
8 # or http://www.opensolaris.org/os/licensing.
9 # See the License for the specific language governing permissions
10 # and limitations under the License.
11 #
12 # When distributing Covered Code, include this CDDL HEADER in each
13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
14 # If applicable, add the following below this CDDL HEADER, with the
15 # fields enclosed by brackets "[]" replaced with your own identifying
16 # information: Portions Copyright [yyyy] [name of copyright owner]
17 #
18 # CDDL HEADER END
19 #
21 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
22 # Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
23 #
24 #
25 # This makefile contains the common definitions for all intel
26 # implementation architecture independent modules.
27 #
28 #
29 #
30 # Machine type (implementation architecture):
31 #
32 PLATFORM = i86pc
33 #
34 #
35 # Everybody needs to know how to build modstubs.o and to locate unix.o.
36 # Note that unix.o must currently be selected from among the possible
37 # "implementation architectures". Note further, that unix.o is only
38 # used as an optional error check for undefines so (theoretically)
39 # any "implementation architectures" could be used. We choose i86pc
40 # because it is the reference port.
41 #
42 UNIX_DIR = $(UTSBASE)/i86pc/unix
43 GENLIB_DIR = $(UTSBASE)/intel/genunix
44 IPDRV_DIR = $(UTSBASE)/intel/ip
45 MODSTUBS_DIR = $(UNIX_DIR)
46 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
47 LINTS_DIR = $(OBJS_DIR)
48 LINT_LIB_DIR = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)
49 #
50 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
51 GENLIB = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
52 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
53 LINT_LIB = $(UTSBASE)/i86pc/lint-libs/$(OBJS_DIR)/llib-lunix.ln
54 GEN_LINT_LIB = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln
55 #
56 #
57 # Include the makefiles which define build rule templates, the
58 # collection of files per module, and a few specific flags. Note
59 # that order is significant, just as with an include path. The
60 # first build rule template which matches the files name will be
61 # used. By including these in order from most machine dependent

```

```

62 # to most machine independent, we allow a machine dependent file
63 # to be used in preference over a machine independent version
64 # (Such as a machine specific optimization, which preserves the
65 # interfaces.)
66 #
67 include $(UTSTREE)/intel/Makefile.files
68 include $(UTSTREE)/common/Makefile.files
69 #
70 #
71 # ----- TRANSITIONAL SECTION -----
72 #
73 #
74 #
75 # Not everything which *should* be a module is a module yet. The
76 # following is a list of such objects which are currently part of
77 # genunix but which might someday become kmods. This must be
78 # defined before we include Makefile.uts, or else genunix's build
79 # won't be as parallel as we might like.
80 #
81 NOT_YET_KMODS = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)
82 #
83 #
84 # ----- END OF TRANSITIONAL SECTION -----
85 #
86 # Include machine independent rules. Note that this does not imply
87 # that the resulting module from rules in Makefile.uts is machine
88 # independent. Only that the build rules are machine independent.
89 #
90 include $(UTSBASE)/Makefile.uts
91 #
92 #
93 # The following must be defined for all implementations:
94 #
95 MODSTUBS = $(UTSBASE)/intel/ia32/ml/modstubs.s
96 #
97 #
98 # Define supported builds
99 #
100 DEF_BUILDS = $(DEF_BUILDS64) $(DEF_BUILDS32)
101 ALL_BUILDS = $(ALL_BUILDS64) $(ALL_BUILDS32)
102 #
103 #
104 # x86 or amd64 inline templates
105 #
106 INLINES_32 = $(UTSBASE)/intel/ia32/ml/ia32.il
107 INLINES_64 = $(UTSBASE)/intel/amd64/ml/amd64.il
108 INLINES += $(INLINES_$(CLASS))
109 #
110 #
111 # kernel-specific optimizations; override default in Makefile.master
112 #
113 #
114 CFLAGS_XARCH_32 = $(i386_CFLAGS)
115 CFLAGS_XARCH_64 = $(amd64_CFLAGS)
116 CFLAGS_XARCH = $(CFLAGS_XARCH_$(CLASS))
117 #
118 COPTFLAG_32 = $(COPTFLAG)
119 COPTFLAG_64 = $(COPTFLAG64)
120 COPTIMIZE = $(COPTFLAG_$(CLASS))
121 #
122 CFLAGS = $(CFLAGS_XARCH)
123 CFLAGS += $(COPTIMIZE)
124 CFLAGS += $(INLINES) -D_ASM_INLINES
125 CFLAGS += $(CCMODE)
126 CFLAGS += $(SPACEFLAG)
127 CFLAGS += $(CCUNBOUND)

```

new/usr/src/uts/intel/Makefile.intel.shared

3

```

128 CFLAGS          += $(CFLAGS_uts)
129 CFLAGS          += -xstrconst

131 ASFLAGS_XARCH_32 = $(i386_ASFLAGS)
132 ASFLAGS_XARCH_64 = $(amd64_ASFLAGS)
133 ASFLAGS_XARCH    = $(ASFLAGS_XARCH_$(CLASS))

135 ASFLAGS          += $(ASFLAGS_XARCH)

137 #
138 #       Define the base directory for installation.
139 #
140 BASE_INS_DIR     = $(ROOT)

142 #
143 #       Debugging level
144 #
145 #       Special knowledge of which special debugging options affect which
146 #       file is used to optimize the build if these flags are changed.
147 #
148 DEBUG_DEFS_OBJ32 =
149 DEBUG_DEFS_DBG32 = -DDEBUG
150 DEBUG_DEFS_OBJ64 =
151 DEBUG_DEFS_DBG64 = -DDEBUG
152 DEBUG_DEFS       = $(DEBUG_DEFS_$(BUILD_TYPE))

154 DEBUG_COND_OBJ32 = :sh = echo \\043
155 DEBUG_COND_DBG32 =
156 DEBUG_COND_OBJ64 = :sh = echo \\043
157 DEBUG_COND_DBG64 =
158 IF_DEBUG_OBJ     = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

160 $(IF_DEBUG_OBJ)syscall.o      :=      DEBUG_DEFS      += -DSYSCALLTRACE
161 $(IF_DEBUG_OBJ)clock.o       :=      DEBUG_DEFS      += -DKSLICE=1

163 #
164 #       Collect the preprocessor definitions to be associated with *all*
165 #       files.
166 #
167 ALL_DEFS          = $(DEBUG_DEFS) $(OPTION_DEFS)

169 #
170 #       The kernels modules which are "implementation architecture"
171 #       specific for this machine are enumerated below. Note that most
172 #       of these modules must exist (in one form or another) for each
173 #       architecture.
174 #
175 #       Common Drivers (usually pseudo drivers) (/kernel/drv)
176 #       DRV_KMODS are built both 32-bit and 64-bit
177 #       DRV_KMODS_32 are built only 32-bit
178 #       DRV_KMODS_64 are built only 64-bit
179 #
180 DRV_KMODS        += aac
181 DRV_KMODS        += aggr
182 DRV_KMODS        += ahci
183 DRV_KMODS        += amd64_gart
184 DRV_KMODS        += amr
185 DRV_KMODS        += apgart
186 DRV_KMODS        += srn
187 DRV_KMODS        += agptarget
188 DRV_KMODS        += arn
189 DRV_KMODS        += arp
190 DRV_KMODS        += asy
191 DRV_KMODS        += ata
192 DRV_KMODS        += ath
193 DRV_KMODS        += atu

```

new/usr/src/uts/intel/Makefile.intel.shared

4

```

194 DRV_KMODS        += audio
195 DRV_KMODS        += audio1575
196 DRV_KMODS        += audio810
197 DRV_KMODS        += audiocmi
198 DRV_KMODS        += audiocmihd
199 DRV_KMODS        += audioemul0k
200 DRV_KMODS        += audioens
201 DRV_KMODS        += audiohd
202 DRV_KMODS        += audioixp
203 DRV_KMODS        += audiols
204 DRV_KMODS        += audiopl6x
205 DRV_KMODS        += audiopci
206 DRV_KMODS        += audiosolo
207 DRV_KMODS        += audiots
208 DRV_KMODS        += audiovial823x
209 DRV_KMODS_32    += audiovial97
210 DRV_KMODS        += bl
211 DRV_KMODS        += blkdev
212 DRV_KMODS        += bge
213 DRV_KMODS        += bofi
214 DRV_KMODS        += bpf
215 DRV_KMODS        += bridge
216 DRV_KMODS        += bscbus
217 DRV_KMODS        += bscv
218 DRV_KMODS        += chxge
219 DRV_KMODS        += cxgbe
220 DRV_KMODS        += ntxn
221 DRV_KMODS        += myril0ge
222 DRV_KMODS        += clone
223 DRV_KMODS        += cmdk
224 DRV_KMODS        += cn
225 DRV_KMODS        += conskbd
226 DRV_KMODS        += consms
227 DRV_KMODS        += cpuid
228 DRV_KMODS        += cpunex
229 DRV_KMODS        += crypto
230 DRV_KMODS        += cryptoadm
231 DRV_KMODS        += dca
232 DRV_KMODS        += devinfo
233 DRV_KMODS        += dld
234 DRV_KMODS        += dlpiustub
235 DRV_KMODS_32    += dnet
236 DRV_KMODS        += dump
237 DRV_KMODS        += ecpp
238 DRV_KMODS        += emlxs
239 DRV_KMODS        += fd
240 DRV_KMODS        += fdc
241 DRV_KMODS        += fm
242 DRV_KMODS        += fssnap
243 DRV_KMODS        += hxge
244 DRV_KMODS        += i8042
245 DRV_KMODS        += i915
246 DRV_KMODS        += icmp
247 DRV_KMODS        += icmp6
248 DRV_KMODS        += intel_nb5000
249 DRV_KMODS        += intel_nhm
250 DRV_KMODS        += ip
251 DRV_KMODS        += ip6
252 DRV_KMODS        += ipf
253 DRV_KMODS        += ipnet
254 DRV_KMODS        += ippctl
255 DRV_KMODS        += ipsecah
256 DRV_KMODS        += ipsecesp
257 DRV_KMODS        += ipw
258 DRV_KMODS        += iwh
259 DRV_KMODS        += iwi

```

```

260 DRV_KMODS += iwk
261 DRV_KMODS += iwp
262 DRV_KMODS += iwscn
263 DRV_KMODS += kb8042
264 DRV_KMODS += keysock
265 DRV_KMODS += kssl
266 DRV_KMODS += kstat
267 DRV_KMODS += ksyms
268 DRV_KMODS += kmdb
269 DRV_KMODS += llcl
270 DRV_KMODS += lofi
271 DRV_KMODS += log
272 DRV_KMODS += logindmux
273 DRV_KMODS += mega_sas
274 DRV_KMODS += mc-amd
275 DRV_KMODS += mm
276 DRV_KMODS += mouse8042
277 DRV_KMODS += mpt_sas
278 DRV_KMODS += mr_sas
279 DRV_KMODS += mwl
280 DRV_KMODS += nca
281 DRV_KMODS += nsmb
282 DRV_KMODS += nulldriver
283 DRV_KMODS += nv_sata
284 DRV_KMODS += nxge
285 DRV_KMODS += oce
286 DRV_KMODS += openeep
287 DRV_KMODS += pci_pci
288 DRV_KMODS += pcic
289 DRV_KMODS += pcieb
290 DRV_KMODS += physmem
291 DRV_KMODS += pcan
292 DRV_KMODS += pcwl
293 DRV_KMODS += pit_beeper
294 DRV_KMODS += pm
295 DRV_KMODS += poll
296 DRV_KMODS += pool
297 DRV_KMODS += power
298 DRV_KMODS += pseudo
299 DRV_KMODS += ptc
300 DRV_KMODS += ptm
301 DRV_KMODS += pts
302 DRV_KMODS += ptsl
303 DRV_KMODS += qlge
304 DRV_KMODS += radeon
305 DRV_KMODS += ral
306 DRV_KMODS += ramdisk
307 DRV_KMODS += random
308 DRV_KMODS += rds
309 DRV_KMODS += rds3
310 DRV_KMODS += rpcib
311 DRV_KMODS += rsm
312 DRV_KMODS += rts
313 DRV_KMODS += rtw
314 DRV_KMODS += rum
315 DRV_KMODS += rwd
316 DRV_KMODS += rwn
317 DRV_KMODS += sad
318 DRV_KMODS += sd
319 DRV_KMODS += sdhost
320 DRV_KMODS += sgen
321 DRV_KMODS += si3124
322 DRV_KMODS += smbios
323 DRV_KMODS += softmac
324 DRV_KMODS += spdsoc
325 DRV_KMODS += smbsrv

```

```

326 DRV_KMODS += smp
327 DRV_KMODS += spps
328 DRV_KMODS += sppptun
329 DRV_KMODS += srpt
330 DRV_KMODS += st
331 DRV_KMODS += sy
332 DRV_KMODS += sysevent
333 DRV_KMODS += sysmsg
334 DRV_KMODS += tcp
335 DRV_KMODS += tcp6
336 DRV_KMODS += tl
337 DRV_KMODS += tnf
338 DRV_KMODS += tpm
339 DRV_KMODS += trill
340 DRV_KMODS += udp
341 DRV_KMODS += udp6
342 DRV_KMODS += ucode
343 DRV_KMODS += ural
344 DRV_KMODS += uath
345 DRV_KMODS += urtw
346 DRV_KMODS += vgatext
347 DRV_KMODS += heci
348 DRV_KMODS += vnic
349 DRV_KMODS += vscan
350 DRV_KMODS += wc
351 DRV_KMODS += winlock
352 DRV_KMODS += wpi
353 DRV_KMODS += xge
354 DRV_KMODS += yge
355 DRV_KMODS += zcons
356 DRV_KMODS += zyd
357 DRV_KMODS += simnet
358 DRV_KMODS += stmf
359 DRV_KMODS += stmf_sbd
360 DRV_KMODS += fct
361 DRV_KMODS += fcoe
362 DRV_KMODS += fcoet
363 DRV_KMODS += fcoei
364 DRV_KMODS += qlt
365 DRV_KMODS += iscsit
366 DRV_KMODS += pppt
367 DRV_KMODS += ncall nsctl sdbc nskern sv
368 DRV_KMODS += ii rdc rdcsrv rdcstub
369 DRV_KMODS += iptun

371 $(CLOSED_BUILD)CLOSED_DRV_KMODS += bmc
372 $(CLOSED_BUILD)CLOSED_DRV_KMODS += glm
373 $(CLOSED_BUILD)CLOSED_DRV_KMODS += intel_nhmex
374 $(CLOSED_BUILD)CLOSED_DRV_KMODS += cpgary3
375 $(CLOSED_BUILD)CLOSED_DRV_KMODS += marvell188sx
376 $(CLOSED_BUILD)CLOSED_DRV_KMODS += bcm_sata
377 $(CLOSED_BUILD)CLOSED_DRV_KMODS += memtest
378 $(CLOSED_BUILD)CLOSED_DRV_KMODS += mpt
379 $(CLOSED_BUILD)CLOSED_DRV_KMODS += atiatom
380 $(CLOSED_BUILD)CLOSED_DRV_KMODS += acpi_toshiba

371 #
372 # Common code drivers
373 #

375 DRV_KMODS += afe
376 DRV_KMODS += atge
377 DRV_KMODS += bfe
378 DRV_KMODS += dmfe
379 DRV_KMODS += e1000g
380 DRV_KMODS += efe

```

```

381 DRV_KMODS      += elxl
382 DRV_KMODS      += hme
383 DRV_KMODS      += mxfe
384 DRV_KMODS      += nge
385 DRV_KMODS      += pcn
386 DRV_KMODS      += rge
387 DRV_KMODS      += rtls
388 DRV_KMODS      += sfe
389 DRV_KMODS      += amd8111s
390 DRV_KMODS      += igb
391 DRV_KMODS      += ipmi
392 DRV_KMODS      += iprb
393 DRV_KMODS      += ixgbe
394 DRV_KMODS      += vr
406 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ixgb

```

```

396 #
397 # Virtio drivers
398 #

```

```

400 # Virtio core
401 DRV_KMODS      += virtio

```

```

403 # Virtio block driver
404 DRV_KMODS      += vioblk

```

```

406 #
407 #      DTrace and DTrace Providers
408 #
409 DRV_KMODS      += dtrace
410 DRV_KMODS      += fbt
411 DRV_KMODS      += lockstat
412 DRV_KMODS      += profile
413 DRV_KMODS      += sdt
414 DRV_KMODS      += systrace
415 DRV_KMODS      += fasttrap
416 DRV_KMODS      += dcpc

```

```

418 #
419 #      I/O framework test drivers
420 #
421 DRV_KMODS      += pshot
422 DRV_KMODS      += gen_drv
423 DRV_KMODS      += tvhci tphci tclient
424 DRV_KMODS      += emul64

```

```

426 #
427 #      Machine Specific Driver Modules (/kernel/drv):
428 #
429 DRV_KMODS      += options
430 DRV_KMODS      += scsi_vhci
431 DRV_KMODS      += pmcs
432 DRV_KMODS      += pmcs8001fw
433 DRV_KMODS      += arcmsr
434 DRV_KMODS      += fcp
435 DRV_KMODS      += fcip
436 DRV_KMODS      += fcsn
437 DRV_KMODS      += fp
438 DRV_KMODS      += qlc
439 DRV_KMODS      += iscsi

```

```

441 #
442 #      PCMCIA specific module(s)
443 #
444 DRV_KMODS      += pcs
445 DRV_KMODS      += pcata

```

```

446 MISC_KMODS     += cardbus
459 $(CLOSED_BUILD)CLOSED_DRV_KMODS += pcser

```

```

448 #
449 #      SCSI Enclosure Services driver
450 #
451 DRV_KMODS      += ses

```

```

453 #
454 #      USB specific modules
455 #
456 DRV_KMODS      += hid
457 DRV_KMODS      += hwarc hwahc
458 DRV_KMODS      += hubd
459 DRV_KMODS      += uhci
460 DRV_KMODS      += ehci
461 DRV_KMODS      += ohci
462 DRV_KMODS      += usb_mid
463 DRV_KMODS      += usb_ia
464 DRV_KMODS      += scsa2usb
465 DRV_KMODS      += usbprn
466 DRV_KMODS      += ugen
467 DRV_KMODS      += usbser
468 DRV_KMODS      += usbsacm
469 DRV_KMODS      += usbsksp
470 DRV_KMODS      += usbsprl
471 DRV_KMODS      += usb_ac
472 DRV_KMODS      += usb_as
473 DRV_KMODS      += usbskel
474 DRV_KMODS      += usbvc
475 DRV_KMODS      += usbftdi
476 DRV_KMODS      += wusb_df
477 DRV_KMODS      += wusb_ca
478 DRV_KMODS      += usbecm

```

```

493 $(CLOSED_BUILD)CLOSED_DRV_KMODS += usbser_edge

```

```

480 #
481 #      1394 modules
482 #
483 MISC_KMODS     += s1394 sbp2
484 DRV_KMODS      += hcil394 scsa1394
485 DRV_KMODS      += av1394
486 DRV_KMODS      += dcaml394

```

```

488 #
489 #      InfiniBand pseudo drivers
490 #
491 DRV_KMODS      += ib ibp eibnx eoib rdsib sdp iser daplt hermon tavor sol_ucma
492 DRV_KMODS      += sol_umad

```

```

494 #
495 #      LVM modules
496 #
497 DRV_KMODS      += md
498 MISC_KMODS     += md_stripe md_hotspares md_mirror md_raid md_trans md_notify
499 MISC_KMODS     += md_sp

```

```

501 #
502 #      Brand modules
503 #
504 BRAND_KMODS    += snl_brand s10_brand

```

```

506 #
507 #      Exec Class Modules (/kernel/exec):
508 #

```

```

509 EXEC_KMODS      += elfexec intpexec shbinexec javaexec

511 #
512 #       Scheduling Class Modules (/kernel/sched):
513 #
514 SCHED_KMODS      += IA RT TS RT_DPTBL TS_DPTBL FSS FX FX_DPTBL SDC

516 #
517 #       File System Modules (/kernel/fs):
518 #
519 FS_KMODS         += autofs cachefs ctfs dcfs dev devfs fdfs fifofs hsfs lofs
520 FS_KMODS         += mntfs namefs nfs objfs zfs zut
521 FS_KMODS         += pcfs procfs sockfs specfs tmpfs udfs ufs sharefs
522 FS_KMODS         += smbfs

524 #
525 #       Streams Modules (/kernel/strmod):
526 #
527 STRMOD_KMODS     += bufmod connld dedump ldterm pckt pfmod pipemod
528 STRMOD_KMODS     += ptem redirmod rpcmod rlmod telmod timod
529 STRMOD_KMODS     += sppsasyn sppscomp
530 STRMOD_KMODS     += tirdwr ttcompat
531 STRMOD_KMODS     += usbkbm
532 STRMOD_KMODS     += usbms
533 STRMOD_KMODS     += usbwcm
534 STRMOD_KMODS     += usb_ah
535 STRMOD_KMODS     += drcompat
536 STRMOD_KMODS     += cryptmod
537 STRMOD_KMODS     += vuid2ps2
538 STRMOD_KMODS     += vuid3ps2
539 STRMOD_KMODS     += vuidm3p
540 STRMOD_KMODS     += vuidm4p
541 STRMOD_KMODS     += vuidm5p

543 #
544 #       'System' Modules (/kernel/sys):
545 #
546 SYS_KMODS        += c2audit
547 SYS_KMODS        += doorfs
548 SYS_KMODS        += exacctsys
549 SYS_KMODS        += inst_sync
550 SYS_KMODS        += kaio
551 SYS_KMODS        += msgsys
552 SYS_KMODS        += pipe
553 SYS_KMODS        += portfs
554 SYS_KMODS        += pset
555 SYS_KMODS        += semsys
556 SYS_KMODS        += shmsys
557 SYS_KMODS        += sysacct
558 SYS_KMODS        += acctctl

560 #
561 #       'Misc' Modules (/kernel/misc)
562 #       MISC_KMODS are built both 32-bit and 64-bit
563 #       MISC_KMODS_32 are built only 32-bit
564 #       MISC_KMODS_64 are built only 64-bit
565 #
566 MISC_KMODS       += ac97
567 MISC_KMODS       += acpica
568 MISC_KMODS       += agpmaster
569 MISC_KMODS       += bignum
570 MISC_KMODS       += bootdev
571 MISC_KMODS       += busra
572 MISC_KMODS       += cmlb
573 MISC_KMODS       += consconfig
574 MISC_KMODS       += ctf

```

```

575 MISC_KMODS      += dadk
576 MISC_KMODS      += dcopy
577 MISC_KMODS      += dls
578 MISC_KMODS      += drm
579 MISC_KMODS      += fssnap_if
580 MISC_KMODS      += gda
581 MISC_KMODS      += gld
582 MISC_KMODS      += hidparser
583 MISC_KMODS      += hook
584 MISC_KMODS      += hpcsvc
585 MISC_KMODS      += ibcm
586 MISC_KMODS      += ibdm
587 MISC_KMODS      += ibdma
588 MISC_KMODS      += ibmf
589 MISC_KMODS      += ibtl
590 MISC_KMODS      += idm
591 MISC_KMODS      += idmap
592 MISC_KMODS      += iomulib
593 MISC_KMODS      += ipc
594 MISC_KMODS      += kbtrans
595 MISC_KMODS      += kcf
596 MISC_KMODS      += kgssapi
597 MISC_KMODS      += kmech_dummy
598 MISC_KMODS      += kmech_krb5
599 MISC_KMODS      += ksocket
600 MISC_KMODS      += mac
601 MISC_KMODS      += mii
602 MISC_KMODS      += mwlfw
603 MISC_KMODS      += net80211
604 MISC_KMODS      += nfs_dlboot
605 MISC_KMODS      += nfssrv
606 MISC_KMODS      += neti
607 MISC_KMODS      += pci_autoconfig
608 MISC_KMODS      += pcicfg
609 MISC_KMODS      += pcihp
610 MISC_KMODS      += pcmcia
611 MISC_KMODS      += rpcsec
612 MISC_KMODS      += rpcsec_gss
613 MISC_KMODS      += rsmops
614 MISC_KMODS      += sata
615 MISC_KMODS      += scsi
616 MISC_KMODS      += sda
617 MISC_KMODS      += sol_ofs
618 MISC_KMODS      += spuni
619 MISC_KMODS      += strategy
620 MISC_KMODS      += strplumb
621 MISC_KMODS      += tem
622 MISC_KMODS      += tlimod
623 MISC_KMODS      += usba usba10 usbs49_fw
624 MISC_KMODS      += scsi_vhci_f_sym_hds
625 MISC_KMODS      += scsi_vhci_f_sym
626 MISC_KMODS      += scsi_vhci_f_tpgs
627 MISC_KMODS      += scsi_vhci_f_asym_sun
628 MISC_KMODS      += scsi_vhci_f_tape
629 MISC_KMODS      += scsi_vhci_f_tpgs_tape
630 MISC_KMODS      += fctl
631 MISC_KMODS      += emlxs_fw
632 MISC_KMODS      += qlc_fw_2200
633 MISC_KMODS      += qlc_fw_2300
634 MISC_KMODS      += qlc_fw_2400
635 MISC_KMODS      += qlc_fw_2500
636 MISC_KMODS      += qlc_fw_6322
637 MISC_KMODS      += qlc_fw_8100
638 MISC_KMODS      += hwa1480_fw
639 MISC_KMODS      += uathfw
640 MISC_KMODS      += uwba

```



```

657 $(CLOSED_BUILD)CLOSED_MISC_KMODS += klmmod klmops
658 $(CLOSED_BUILD)CLOSED_MISC_KMODS += scsi_vhci_f_asym_lsi
659 $(CLOSED_BUILD)CLOSED_MISC_KMODS += scsi_vhci_f_asym_emc
660 $(CLOSED_BUILD)CLOSED_MISC_KMODS += scsi_vhci_f_sym_emc

```

```

642 #
643 #     Software Cryptographic Providers (/kernel/crypto):
644 #

```

```

645 CRYPTO_KMODS += aes
646 CRYPTO_KMODS += arcfour
647 CRYPTO_KMODS += blowfish
648 CRYPTO_KMODS += des
649 CRYPTO_KMODS += ecc
650 CRYPTO_KMODS += md4
651 CRYPTO_KMODS += md5
652 CRYPTO_KMODS += rsa
653 CRYPTO_KMODS += sha1
654 CRYPTO_KMODS += sha2
655 CRYPTO_KMODS += swrand

```

```

657 #
658 #     IP Policy Modules (/kernel/ipp)
659 #

```

```

660 IPP_KMODS += dlcosmk
661 IPP_KMODS += flowacct
662 IPP_KMODS += ipgpc
663 IPP_KMODS += dscpmk
664 IPP_KMODS += tokenmt
665 IPP_KMODS += tswtclmt

```

```

667 #
668 #     generic-unix module (/kernel/genunix):
669 #
670 GENUNIX_KMODS += genunix

```

```

672 #
673 #     SVVS Testing Modules (/kernel/stvm):
674 #

```

```

695 #     These are streams and driver modules which are not to be
696 #     delivered with a released system. However, during development
697 #     it is convenient to build and install the SVVS kernel modules.
698 #

```

```

699 SVVS_KMODS += lmodb lmode lmodr lmodt svvslo tidg tivc tmux

```

```

701 $(CLOSED_BUILD)SVVS += svvs

```

```

703 #
704 #     Modules eXcluded from the product:
705 #

```

```

706 $(CLOSED_BUILD)CLOSED_XMODS = \
707     adpu320 \
708     bnx \
709     bnxe \
710     lsimega \
711     sdpiib

```

```

676 #
677 #     'Dacf' Modules (/kernel/dacf):
678 #

```

```

680 #
681 #     Performance Counter BackEnd modules (/usr/kernel/pcbe)
682 #
683 PCBE_KMODS += p123_pcbe p4_pcbe opteron_pcbe core_pcbe

```

```

685 #
686 #     MAC-Type Plugin Modules (/kernel/mac)
687 #
688 MAC_KMODS += mac_6to4
689 MAC_KMODS += mac_ether
690 MAC_KMODS += mac_ipv4
691 MAC_KMODS += mac_ipv6
692 MAC_KMODS += mac_wifi
693 MAC_KMODS += mac_ib

```

```

695 #
696 #     socketmod (kernel/socketmod)
697 #
698 SOCKET_KMODS += sockpfp
699 SOCKET_KMODS += socksctp
700 SOCKET_KMODS += socksdp
701 SOCKET_KMODS += sockrds
702 SOCKET_KMODS += ksslf

```

```

704 #
705 #     kiconv modules (/kernel/kiconv):
706 #
707 KICONV_KMODS += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc

```

```

709 #
710 #     'Dacf' Modules (/kernel/dacf):
711 #
712 DACF_KMODS += net_dacf

```

new/usr/src/uts/intel/intel_nb5000/Makefile

1

```
*****
2185 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/intel_nb5000/Makefile
4027 remove CLOSED_BUILD
*****
1 # CDDL HEADER START
2 #
3 # The contents of this file are subject to the terms of the
4 # Common Development and Distribution License (the "License").
5 # You may not use this file except in compliance with the License.
6 #
7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
8 # or http://www.opensolaris.org/os/licensing.
9 # See the License for the specific language governing permissions
10 # and limitations under the License.
11 #
12 # When distributing Covered Code, include this CDDL HEADER in each
13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
14 # If applicable, add the following below this CDDL HEADER, with the
15 # fields enclosed by brackets "[]" replaced with your own identifying
16 # information: Portions Copyright [yyyy] [name of copyright owner]
17 #
18 # CDDL HEADER END
19 #
20 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
21 # Use is subject to license terms.
22 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
23 #
24 #
25 #
26 # Intel 5000/5100/5400/7300 chipset memory controller hub (MCH) module
27 #
28 #
29 #
30 # Path to the base of the uts directory tree (usually /usr/src/uts).
31 #
32 UTSBASE = ../../
33 UTSCLOSED = ../../../../closed/uts
34 #
35 # Define the module and object file sets.
36 #
37 MODULE = intel_nb5000
38 #
39 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
40 SRCDIR = $(UTSBASE)/intel/io/intel_nb5000
41 CONF_SRCDIR = $(UTSBASE)/intel/io/intel_nb5000
42 #
43 #
44 # Include common rules.
45 #
46 include ../Makefile.intel
47 #
48 #
49 #
50 # The list of object files is defined here, rather than in Makefile.files,
51 # because the "$(CLOSED_BUILD)" macro has not been defined at the time
52 # Makefile.files is processed.
53 #
54 INTEL_NB5000_OBJS += \
55     intel_nb5000.o \
56     intel_nbdrv.o \
57     dimm_addr.o \
58     nb_pci_cfg.o \
59     nb5000_init.o
60 #
61 $(CLOSED_BUILD)INTEL_NB5000_OBJS += memtrans.o
```

new/usr/src/uts/intel/intel_nb5000/Makefile

2

```
48 OBJECTS = $(INTEL_NB5000_OBJS:%=$(OBJS_DIR)/%)
49 LINTS = $(INTEL_NB5000_OBJS:%.o=$(LINTS_DIR)/%.ln)
50 #
51 #
52 # Define targets
53 #
54 ALL_TARGET = $(BINARY)
55 LINT_TARGET = $(LINT_MODULE).lint
56 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFILE)
57 #
58 #
59 # Depends on smbios
60 #
61 LDFLAGS += -dy -N drv/smbios
62 #
63 CERRWARN += -_gcc=-Wno-parentheses
64 CERRWARN += -_gcc=-Wno-uninitialized
65 #
66 #
67 # Default build targets.
68 #
69 .KEEP_STATE:
70 #
71 def: $(DEF_DEPS)
72 #
73 all: $(ALL_DEPS)
74 #
75 clean: $(CLEAN_DEPS)
76 #
77 clobber: $(CLOBBER_DEPS)
78 #
79 lint: $(LINT_DEPS)
80 #
81 modlintlib: $(MODLINTLIB_DEPS)
82 #
83 clean.lint: $(CLEAN_LINT_DEPS)
84 #
85 install: $(INSTALL_DEPS)
86 #
87 $(OBJECTS): $(OBJS_DIR) $(MCAMD_OFF_H)
88 #
89 #
90 # Include common targets.
91 #
92 include ../Makefile.targ
108 $(CLOSED_BUILD)include $(UTSCLOSED)/intel/Makefile.rules
```

new/usr/src/uts/intel/intel_nhm/Makefile

1

```
*****
2012 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/intel_nhm/Makefile
4027 remove CLOSED_BUILD
*****
1 # CDDL HEADER START
2 #
3 # The contents of this file are subject to the terms of the
4 # Common Development and Distribution License (the "License").
5 # You may not use this file except in compliance with the License.
6 #
7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
8 # or http://www.opensolaris.org/os/licensing.
9 # See the License for the specific language governing permissions
10 # and limitations under the License.
11 #
12 # When distributing Covered Code, include this CDDL HEADER in each
13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
14 # If applicable, add the following below this CDDL HEADER, with the
15 # fields enclosed by brackets "[]" replaced with your own identifying
16 # information: Portions Copyright [yyyy] [name of copyright owner]
17 #
18 # CDDL HEADER END
19 #
20 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
21 # Use is subject to license terms.
22 #
23 #
24 #
25 # Intel Nehalem memory controller module
26 #
27 #
28 #
29 # Path to the base of the uts directory tree (usually /usr/src/uts).
30 #
31 UTSBASE = ../../
32 UTSCLOSED = ../../../../closed/uts
33 #
34 # Define the module and object file sets.
35 #
36 MODULE = intel_nhm
37 #
38 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
39 CONF_SRCDIR = $(UTSBASE)/intel/io/intel_nhm
40 #
41 #
42 # Include common rules.
43 #
44 include $(UTSBASE)/intel/Makefile.intel
45 #
46 #
47 #
48 # The list of object files is defined here, rather than in Makefile.files,
49 # because the "$(CLOSED_BUILD)" macro has not been defined at the time
50 # Makefile.files is processed.
51 #
52 INTEL_NHM_OBJS += \
53     nhm_init.o \
54     mem_addr.o \
55     intel_nhmdrv.o \
56     nhm_pci_cfg.o \
57     dimm_topo.o \
58     intel_nhm.o
59 #
60 $(CLOSED_BUILD)INTEL_NHM_OBJS += \
61     nhm_dimm_addr.o
```

new/usr/src/uts/intel/intel_nhm/Makefile

2

```
46 OBJECTS = $(INTEL_NHM_OBJS:%=$(OBJSDIR)/%)
47 LINTS = $(INTEL_NHM_OBJS:%.o=$(LINTSDIR)/%.ln)
48 #
49 # Define targets
50 #
51 #
52 ALL_TARGET = $(BINARY) $(SRC_CONFFILE)
53 LINT_TARGET = $(LINT_MODULE).lint
54 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFILE)
55 #
56 CPPFLAGS += -I$(UTSBASE)/i86pc
57 LDFLAGS += -dy -N drv/smbios
58 #
59 CERRWARN += -_gcc=-Wno-uninitialized
60 #
61 # Default build targets.
62 #
63 #
64 .KEEP_STATE:
65 #
66 def: $(DEF_DEPS)
67 #
68 all: $(ALL_DEPS)
69 #
70 clean: $(CLEAN_DEPS)
71 #
72 clobber: $(CLOBBER_DEPS)
73 #
74 lint: $(LINT_DEPS)
75 #
76 modlintlib: $(MODLINTLIB_DEPS)
77 #
78 clean.lint: $(CLEAN_LINT_DEPS)
79 #
80 install: $(INSTALL_DEPS)
81 #
82 # Include common targets.
83 #
84 #
85 include ../Makefile.targ
103 $(CLOSED_BUILD)include $(UTSCLOSED)/intel/Makefile.rules
```

```

*****
3889 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/sd/Makefile
4027 remove CLOSED_BUILD
*****
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 # uts/intel/io/sd/Makefile
22 #
23 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
26 #
27 # This makefile drives the production of the sd
28 # kernel module.
29 #
30 #
31 #
32 #
33 # Path to the base of the uts directory tree (usually /usr/src/uts).
34 #
35 UTSBASE = ../../
36 #
37 #
38 # Define the module and object file sets.
39 #
40 MODULE = sd
41 OBJECTS = $(SD_OBJS:%=$(OBJS_DIR)/%)
42 LINTS = $(SD_OBJS:%.o=$(LINTS_DIR)/%.ln)
43 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
44 CONF_SRCDIR = $(UTSBASE)/intel/io/scsi/targets
45 WARLOCK_OUT = $(SD_OBJS:%.o=%.ll)
46 WARLOCK_OK = $(MODULE).ok
47 WLCMD_DIR = $(UTSBASE)/common/io/warlock
48 #
49 #
50 # Include common rules.
51 #
52 include $(UTSBASE)/intel/Makefile.intel
53 #
54 #
55 # Define targets
56 #
57 ALL_TARGET = $(BINARY) $(CONFMOD)
58 LINT_TARGET = $(MODULE).lint
59 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFILE)
60 #
61 #

```

```

62 # Overrides.
63 #
64 DEBUG_FLGS =
65 DEBUG_DEFS += $(DEBUG_FLGS)
66 #
67 INC_PATH += -I$(UTSBASE)/intel/io/scsi/targets
68 #
69 #
70 # For now, disable these lint checks; maintainers should endeavor
71 # to investigate and remove these for maximum lint coverage.
72 # Please do not carry these forward to new Makefiles.
73 #
74 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
75 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
76 LINTTAGS += -erroff=E_STATIC_UNUSED
77 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
78 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
79 #
80 CERRWARN += -_gcc=-Wno-unused-variable
81 CERRWARN += -_gcc=-Wno-unused-function
82 CERRWARN += -_gcc=-Wno-unused-label
83 CERRWARN += -_gcc=-Wno-parentheses
84 CERRWARN += -_gcc=-Wno-type-limits
85 CERRWARN += -_gcc=-Wno-uninitialized
86 #
87 #
88 # Depends on scsi and cmlb
89 #
90 LDFLAGS += -dy -N misc/scsi -N misc/cmlb
91 #
92 #
93 # Default build targets.
94 #
95 .KEEP_STATE:
96 #
97 def: $(DEF_DEPS)
98 #
99 all: $(ALL_DEPS)
100 #
101 clean: $(CLEAN_DEPS)
102 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)
103 #
104 clobber: $(CLOBBER_DEPS)
105 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)
106 #
107 lint: $(LINT_DEPS)
108 #
109 modlintlib: $(MODLINTLIB_DEPS)
110 #
111 clean.lint: $(CLEAN_LINT_DEPS)
112 #
113 install: $(INSTALL_DEPS)
114 #
115 #
116 # Include common targets.
117 #
118 include $(UTSBASE)/intel/Makefile.targ
119 #
120 #
121 #
122 # Defines for local commands.
123 #
124 WARLOCK = warlock
125 WLCC = wlcc
126 TOUCH = touch
127 TEST = test

```

```
129 #
130 # Warlock targets
131 #
132 # Note that in warlock_with_{esp,isp} it is important to load sd.ll
133 # before {isp,esp}.ll; the reason is that both have _init/_info/_fini
134 # and warlock can only handle one extern function by a given name;
135 # any loaded after the first are ignored.

137 SCSI_FILES = $(SCSI_OBJS:%.o=-l ../scsi/%.ll)
138 CMLB_FILES = $(CMLB_OBJS:%.o=-l ../cmlb/%.ll)

140 WARLOCK_TARGETS = warlock_alone
141 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_mpt warlock_with_glm

142 warlock: $(WARLOCK_TARGETS)

144 warlock_alone: $(WARLOCK_OK)

146 $(WARLOCK_OK): $(WLCMD_DIR)/sd.wlcmd $(WARLOCK_OUT) scsi_files \
147     warlock_ddi.files cmlb_files
148     $(WARLOCK) -c $(WLCMD_DIR)/sd.wlcmd $(WARLOCK_OUT) $(SCSI_FILES) \
149     $(CMLB_FILES) \
150     -l ../warlock/ddi_dki_impl.ll
151     $(TOUCH) $@

153 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
154     $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

157 warlock_with_glm: $(WLCMD_DIR)/sd_with_glm.wlcmd $(WARLOCK_OUT) scsi_files \
158     glm_files warlock_ddi.files cmlb_files
159     $(WARLOCK) -c $(WLCMD_DIR)/sd_with_glm.wlcmd \
160     $(WARLOCK_OUT) \
161     $(CLOSED)/uts/intel/glm/*.ll \
162     $(SCSI_FILES) $(CMLB_FILES) \
163     -l ../warlock/ddi_dki_impl.ll

165 warlock_with_mpt: $(WLCMD_DIR)/sd_with_mpt.wlcmd $(WARLOCK_OUT) scsi_files \
166     mpt_files warlock_ddi.files cmlb_files
167     $(WARLOCK) -c $(WLCMD_DIR)/sd_with_mpt.wlcmd \
168     $(WARLOCK_OUT) \
169     $(CLOSED)/uts/intel/mpt/*.ll \
170     $(SCSI_FILES) $(CMLB_FILES) \
171     -l ../warlock/ddi_dki_impl.ll

173 glm_files:
174     @cd $(CLOSED)/uts/intel/glm; pwd; $(MAKE) warlock

176 mpt_files:
177     @cd $(CLOSED)/uts/intel/mpt; pwd; $(MAKE) warlock

156 cmlb_files:
157     @cd ../cmlb; pwd; $(MAKE) warlock

159 scsi_files:
160     @cd ../scsi; pwd; $(MAKE) warlock

162 warlock_ddi.files:
163     @cd ../warlock; pwd; $(MAKE) warlock
```

new/usr/src/uts/intel/ses/Makefile

1

```
*****
3198 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/ses/Makefile
4027 remove CLOSED_BUILD
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # uts/intel/ses/Makefile
23 #
24 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
27 #
28 # This makefile drives the production of the ses (SCSI enclosure
29 # services) driver kernel module.
30 #
31 # intel implementation architecture dependent
32 #
33 #
34 #
35 # Path to the base of the uts directory tree (usually /usr/src/uts).
36 #
37 UTSBASE = ../../

39 #
40 # Define the module and object file sets.
41 #
42 MODULE = ses
43 OBJECTS = $(SES_OBJS:%=$(OBJDIR)/%)
44 LINTS = $(SES_OBJS:%.o=$(LINTSDIR)/%.ln)
45 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
46 CONF_SRCDIR = $(UTSBASE)/intel/io/scsi/targets
47 WARLOCK_OUT = $(SES_OBJS:%.o=%11)
48 WARLOCK_OK = $(MODULE).ok
49 WLCMD_DIR = $(UTSBASE)/common/io/warlock

51 #
52 # Include common rules.
53 #
54 include $(UTSBASE)/intel/Makefile.intel

56 #
57 # Define targets
58 #
59 ALL_TARGET = $(BINARY) $(SRC_CONFIL)
60 LINT_TARGET = $(MODULE).lint
61 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFIL)
```

new/usr/src/uts/intel/ses/Makefile

2

```
62 CLEANFILES += $(WARLOCK_TARGETS)

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

72 #
73 # Depends on scsi
74 #
75 LDPLAGS += -dy -N misc/scsi

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

82 def: $(DEF_DEPS)

84 all: $(ALL_DEPS)

86 clean: $(CLEAN_DEPS)
87 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

89 clobber: $(CLOBBER_DEPS)
90 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

92 lint: $(LINT_DEPS)

94 modlintlib: $(MODLINTLIB_DEPS)

96 clean.lint: $(CLEAN_LINT_DEPS)

98 install: $(INSTALL_DEPS)

100 #
101 # Include common targets.
102 #
103 include $(UTSBASE)/intel/Makefile.targ

106 #
107 # Defines for local commands.
108 #
109 WARLOCK = warlock
110 WLCC = wlcc
111 TOUCH = touch
112 TEST = test

114 #
115 # Warlock targets
116 #
117 SCSI_FILES = $(SCSI_OBJS:%.o=-1 ../scsi/%11)

119 WARLOCK_TARGETS = $(WARLOCK_OK)
120 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_glm

121 warlock: $(WARLOCK_TARGETS)

123 # XXX FIX ME: why only ses.ll?

125 $(WARLOCK_OK): $(WLCMD_DIR)/$(MODULE).wlcmd $(WARLOCK_OUT)
126 @cd ../warlock; $(MAKE) warlock
```

```
127     @cd ../scsi; $(MAKE) warlock
128     $(WARLOCK) -c $(WLCMD_DIR)/$(MODULE).wlcmd ses.ll \
129         ../warlock/scsi.ll \
130         -l ../warlock/ddi_dki_impl.ll \
131         $(SCSI_FILES)
132     @ $(TOUCH) $@

134 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
135     $(WLCC) $(CPPFLAGS) -o $@ $<

138 warlock_with_glm: $(WLCMD_DIR)/ses_with_glm.wlcmd $(WARLOCK_OUT) glm_files
139     @cd ../warlock; $(MAKE) warlock
140     @cd ../scsi; $(MAKE) warlock
141     @cd $(CLOSED)/uts/intel/glm; $(MAKE) warlock;
142     $(WARLOCK) -c $(WLCMD_DIR)/ses_with_glm.wlcmd ses.ll \
143         $(CLOSED)/uts/intel/glm/*.ll \
144         $(SCSI_FILES) \
145         ../warlock/scsi.ll \
146         -l ../warlock/ddi_dki_impl.ll

148 glm_files:
149     @cd $(CLOSED)/uts/intel/glm; pwd; $(MAKE) warlock

137 warlock_ddi.files:
138     @cd ../warlock; pwd; $(MAKE) warlock
```

```

*****
3358 Thu Aug 15 11:59:24 2013
new/usr/src/uts/intel/st/Makefile
4027 remove CLOSED_BUILD
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # uts/intel/st/Makefile
23 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
26 #
27 # This makefile drives the production of the st driver
28 # kernel module.
29 #
30 # intel implementation architecture independent
31 #
32 #
33 #
34 # Path to the base of the uts directory tree (usually /usr/src/uts).
35 #
36 UTSBASE = ../../

38 #
39 # Define the module and object file sets.
40 #
41 MODULE = st
42 OBJECTS = $(ST_OBJS:%=$(OBJDIR)/%)
43 LINTS = $(ST_OBJS:%.o=$(LINTS_DIR)/%.ln)
44 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
45 CONF_SRCDIR = $(UTSBASE)/intel/io/scsi/targets
46 WARLOCK_OUT = $(ST_OBJS:%.o=%.ll)
47 WARLOCK_OK = $(MODULE).ok
48 WLCMD_DIR = $(UTSBASE)/common/io/warlock

50 #
51 # Include common rules.
52 #
53 include $(UTSBASE)/intel/Makefile.intel

55 #
56 # Define targets
57 #
58 ALL_TARGET = $(BINARY) $(SRC_CONFIL)
59 LINT_TARGET = $(MODULE).lint
60 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFIL)

```

```

62 #
63 # Overrides.
64 #
65 DEBUG_FLGS =
66 DEBUG_DEFS += $(DEBUG_FLGS)

68 #
69 # For now, disable these lint checks; maintainers should endeavor
70 # to investigate and remove these for maximum lint coverage.
71 # Please do not carry these forward to new Makefiles.
72 #
73 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
74 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
75 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV

77 CERRWARN += -_gcc=-Wno-parentheses
78 CERRWARN += -_gcc=-Wno-switch
79 CERRWARN += -_gcc=-Wno-uninitialized

81 #
82 # Depends on scsi
83 #
84 LDFLAGS += -dy -N misc/scsi

86 #
87 # Default build targets.
88 #
89 .KEEP_STATE:

91 def: $(DEF_DEPS)

93 all: $(ALL_DEPS)

95 clean: $(CLEAN_DEPS)
96 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

98 clobber: $(CLOBBER_DEPS)
99 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

101 lint: $(LINT_DEPS)

103 modlintlib: $(MODLINTLIB_DEPS)

105 clean.lint: $(CLEAN_LINT_DEPS)

107 install: $(INSTALL_DEPS)

109 #
110 # Include common targets.
111 #
112 include $(UTSBASE)/intel/Makefile.targ

114 #
115 # Defines for local commands.
116 #
117 WARLOCK = warlock
118 WLCC = wlcc
119 TOUCH = touch
120 TEST = test

122 #
123 # Warlock targets
124 #

126 SCSI_FILES = $(SCSI_OBJS:%.o=-l ../scsi/%.ll)

```



```
128 WARLOCK_TARGETS = warlock_alone
129 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_glm

130 warlock: $(WARLOCK_TARGETS)

132 warlock_alone: $(WARLOCK_OK)

134 scsi_files:
135     @cd ../scsi; pwd; $(MAKE) warlock

137 $(WARLOCK_OK): $(WLCMD_DIR)/st.wlcmd $(WARLOCK_OUT) scsi_files \
138     warlock_ddi.files
139     $(WARLOCK) -c $(WLCMD_DIR)/st.wlcmd $(WARLOCK_OUT) $(SCSI_FILES) \
140     -l ../warlock/ddi_dki_impl.ll
141     $(TOUCH) $@

143 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
144     $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

147 warlock_with_glm: $(WLCMD_DIR)/st_with_glm.wlcmd $(WARLOCK_OUT) scsi_files \
148     glm_files warlock_ddi.files
149     $(WARLOCK) -c $(WLCMD_DIR)/st_with_glm.wlcmd \
150     $(WARLOCK_OUT) $(CLOSED)/uts/intel/glm/glm $(SCSI_FILES) \
151     -l ../warlock/ddi_dki_impl.ll
152 glm_files:
153     @cd $(CLOSED)/uts/intel/glm; pwd; $(MAKE) warlock

146 warlock_ddi.files:
147     @cd ../warlock; pwd; $(MAKE) warlock

149 scsi.files:
150     @cd ../scsi; pwd; $(MAKE) warlock
```

new/usr/src/uts/intel/sys/Makefile

1

```
*****
2533 Thu Aug 15 11:59:25 2013
new/usr/src/uts/intel/sys/Makefile
4027 remove CLOSED_BUILD
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #

26 include ../../../../Makefile.master

28 # NOTE: hrtcntl.h and hrtsys.h are present in this directory so that the
29 # hrtsys system call can be built to facilitate transportability of
30 # stock SVr4 programs. Every effort is to be made to prevent objects
31 # from being built, so these headers are not exported (installed).

33 HDRS = \
34 archsystem.h \
35 asm_linkage.h \
36 bootconf.h \
37 bootregs.h \
38 bootsvcs.h \
39 controlregs.h \
40 cpu.h \
41 ddi_isa.h \
42 debugreg.h \
43 fasttrap_isa.h \
44 fp.h \
45 frame.h \
46 inline.h \
47 iomulib.h \
48 hypervisor.h \
49 kdi_machimpl.h \
50 kdi_regs.h \
51 machlock.h \
52 machsig.h \
53 machtypes.h \
54 mc.h \
55 mc_amd.h \
56 mc_intel.h \
57 mca_amd.h \
58 mca_x86.h \
59 mutex_impl.h \
60 obpdefs.h \
61 old_procfs.h \
```

new/usr/src/uts/intel/sys/Makefile

2

```
62 pcb.h \
63 pmem.h \
64 privmregs.h \
65 privregs.h \
66 procfs_isa.h \
67 prom_emul.h \
68 prom_isa.h \
69 prom_plat.h \
70 promif.h \
71 promimpl.h \
72 psw.h \
73 pte.h \
74 reg.h \
75 regset.h \
76 segment.h \
77 segments.h \
78 spl.h \
79 stack.h \
80 stat_impl.h \
81 synch32.h \
82 sysconfig_impl.h \
83 sysi86.h \
84 trap.h \
85 traptrace.h \
86 tss.h \
87 ucontext.h \
88 utrap.h \
89 vmparam.h \
90 x86_archext.h \
91 xen_errno.h

93 CLOSEDHDRS = \
94 memtest.h

93 ROOTDIR= $(ROOT)/usr/include/sys
94 SCSIDIR= $(ROOTDIR)/scsi
95 SCSIDIRS= $(SCSIDIR) $(SCSIDIR)/conf $(SCSIDIR)/generic \
96 $(SCSIDIR)/impl $(SCSIDIR)/targets
97 ROOTFSDIR= $(ROOTDIR)/fs
98 ROOTDIRS= $(ROOTDIR) $(ROOTFSDIR)

100 ROOTHDRS= $(HDRS:%=$(ROOTDIR)/%)
104 $(CLOSED_BUILD)ROOTHDRS += $(CLOSEDHDRS:%=$(ROOTDIR)/%)

102 CHECKHDRS= \
103 $(HDRS:%.h=%.check)

109 $(CLOSED_BUILD)CHECKHDRS += $(CLOSEDHDRS:%.h=$(CLOSED)/uts/intel/sys/%.check)

105 # install rules
106 $(ROOTDIR)/%: %
107 $(INS.file)

115 $(ROOTDIR)/%: $(CLOSED)/uts/intel/sys/%
116 $(INS.file)

109 .KEEP_STATE:

111 .PARALLEL: $(CHECKHDRS) $(ROOTHDRS)

113 install_h: $(ROOTDIRS) .WAIT $(ROOTHDRS)

115 $(ROOTDIRS):
116 $(INS.dir)

118 check: $(CHECKHDRS)
```

```

*****
3432 Thu Aug 15 11:59:25 2013
new/usr/src/uts/intel/warlock/Makefile
4027 remove CLOSED_BUILD
*****
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) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 # uts/intel/warlock/Makefile
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 = warlock
35 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)

37 .KEEP_STATE:

39 CFLAGS += -I../common/sys -I../sun/sys/scsi -D_KERNEL
40 #
41 # Defines for local commands.
42 #
43 WARLOCK = warlock
44 WLCC = wlcc
45 TOUCH = touch
46 TEST = test

48 include $(UTSBASE)/intel/Makefile.intel

50 #
51 # lockLint rules
52 #
53 all: warlock warlock.1394 warlock.ecpp warlock.scsi \
54 warlock.usb warlock.ib warlock.sata warlock.wc \
55 warlock.audiohd

57 warlock: $(MODULE).ok

59 warlock.ok: ddi_dki_impl.ll scsi.ll
60 $(TOUCH) $@

```

```

62 %.ll: $(UTSBASE)/common/io/warlock/%.c
63 $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

65 warlock.usb:
66 @cd ../usba; $(MAKE) clean; $(MAKE) warlock
67 @cd ../ohci; $(MAKE) clean; $(MAKE) warlock
68 @cd ../uhci; $(MAKE) clean; $(MAKE) warlock
69 @cd ../ehci; $(MAKE) clean; $(MAKE) warlock
70 @cd ../hid; $(MAKE) clean; $(MAKE) warlock
71 @cd ../scsa2usb; $(MAKE) clean; $(MAKE) warlock
72 @cd ../usb_ac; $(MAKE) clean; $(MAKE) warlock
73 @cd ../usb_as; $(MAKE) clean; $(MAKE) warlock
74 @cd ../usb_ah; $(MAKE) clean; $(MAKE) warlock
75 @cd ../ugen; $(MAKE) clean; $(MAKE) warlock
76 @cd ../usb_mid; $(MAKE) clean; $(MAKE) warlock
77 @cd ../usbprn; $(MAKE) clean; $(MAKE) warlock
78 @cd ../usbser; $(MAKE) clean; $(MAKE) warlock
79 @cd ../usbsksp; $(MAKE) clean; $(MAKE) warlock
80 @cd ../usbspri; $(MAKE) clean; $(MAKE) warlock
81 @cd ../usbsacm; $(MAKE) clean; $(MAKE) warlock
82 @cd ../usbecm; $(MAKE) clean; $(MAKE) warlock
83 @cd ../usbskel; $(MAKE) clean; $(MAKE) warlock
84 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/usbser_edge; \
85 $(MAKE) clean; $(MAKE) warlock

85 warlock.scsi:
86 @cd ../sd; $(MAKE) clean; $(MAKE) warlock
87 @cd ../ses; $(MAKE) clean; $(MAKE) warlock
88 @cd ../st; $(MAKE) clean; $(MAKE) warlock
91 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/glm; $(MAKE) clean; $(MAKE) warlock
92 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/mpt; $(MAKE) clean; $(MAKE) warlock

90 warlock.1394:
91 @cd ../sl394; $(MAKE) clean; $(MAKE) warlock
92 @cd ../hcil394; $(MAKE) clean; $(MAKE) warlock
93 @cd ../scsal394; $(MAKE) clean; $(MAKE) warlock
94 @cd ../avl394; $(MAKE) clean; $(MAKE) warlock

96 warlock.ecpp:
97 @cd ../ecpp; $(MAKE) clean; $(MAKE) warlock

99 warlock.ib:
100 @cd ../ibmf; $(MAKE) clean; $(MAKE) warlock
101 @cd ../ib; $(MAKE) clean; $(MAKE) warlock
102 @cd ../ibtl; $(MAKE) clean; $(MAKE) warlock
103 @cd ../ibcm; $(MAKE) clean; $(MAKE) warlock
104 @cd ../ibd; $(MAKE) clean; $(MAKE) warlock
110 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/tavor; $(MAKE) clean; $(MAKE) warlock
111 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/hermon; $(MAKE) clean; $(MAKE) warlock
112 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/daplt; $(MAKE) clean; $(MAKE) warlock

106 warlock.sata:
107 @cd ../sata; $(MAKE) clean; $(MAKE) warlock
108 @cd ../si3124; $(MAKE) clean; $(MAKE) warlock
109 @cd ../nv_sata; $(MAKE) clean; $(MAKE) warlock
110 @cd ../ahci; $(MAKE) clean; $(MAKE) warlock
119 $(CLOSED_BUILD) @cd $(CLOSED)/uts/intel/marvell88sx; \
120 $(MAKE) clean; $(MAKE) warlock

112 warlock.wc:
113 @cd ../wc; $(MAKE) clean; $(MAKE) warlock

115 warlock.audiohd:
116 @cd ../audiohd; $(MAKE) clean; $(MAKE) warlock

```

new/usr/src/uts/sparc/Makefile

1

```
*****
2567 Thu Aug 15 11:59:25 2013
new/usr/src/uts/sparc/Makefile
4027 remove CLOSED_BUILD
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # uts/sparc/Makefile
26 #
27 # This makefile drives the production of all implementation architecture
28 # independent modules for the SPARC processor. (For those unsure, this
29 # means the module will run on all SPARC processor based machines
30 # running SunOS.)
31 #
32 UTSBASE = ..
33 #
34 include Makefile.sparc
35 #
36 LINT_KMODS_X1 = $(LINT_KMODS:nsmb=)
37 LINT_KMODS_X2 = $(LINT_KMODS_X1:smbfs=)
38 LINT_KMODLIBS = $(LINT_KMODS_X2:e1000g=)
39 LINT_LIBS = $(LINT_LIB) $(GEN_LINT_LIB) \
40 $(LINT_KMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
41 $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln)
42 $(CLOSED_BUILD)LINT_LIBS += $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)
43 #
44 $(CLOSED_BUILD)LINT_LIBS += $(SVVS_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)
45 LINT_LIBS += $(LINT_XMODLIBS:%=$(LINT_LIB_DIR)/llib-1%.ln)
46 $(CLOSED_BUILD)LINT_LIBS += $(CLOSED_XMODS:%=$(LINT_LIB_DIR)/llib-1%.ln)
47 #
48 DRV_KMODS += dprov
49 #
50 def := TARGET= def
51 all := TARGET= all
52 install := TARGET= install
53 clean := TARGET= clean
54 clobber := TARGET= clobber
55 lint := TARGET= lint
56 modlintlib := TARGET= modlintlib
57 modlist := TARGET= modlist
58 modlist := NO_STATE= -K $$MODSTATE$$$
59 clean.lint := TARGET= clean.lint
```

new/usr/src/uts/sparc/Makefile

2

```
55 check := TARGET= check
56 install_h := TARGET= install_h
57 #
58 .KEEP_STATE:
59 #
60 .PARALLEL: $(PARALLEL_KMODS) $(XMODS) config $(LINT_DEPS)
61 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(SVVS) $(XMODS) \
62 $(CLOSED_XMODS) config $(LINT_DEPS)
63 #
64 def all install clean clobber modlist: $(KMODS) $(XMODS) config
65 def all install clean clobber modlist: $(KMODS) $(CLOSED_KMODS) $(SVVS) \
66 $(XMODS) $(CLOSED_XMODS) config
67 #
68 modlintlib clean.lint: $(LINT_KMODS) $(XMODS)
69 modlintlib clean.lint: $(LINT_KMODS) $(CLOSED_LINT_KMODS) $(SVVS) \
70 $(XMODS) $(CLOSED_XMODS)
71 #
72 $(KMODS) config: FRC
73 @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
74 #
75 $(CLOSED_KMODS): FRC
76 cd $(CLOSED)/uts/sparc/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
77 #
78 $(XMODS): FRC
79 @if [ -f $@/Makefile ]; then \
80 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
81 else \
82 true; \
83 fi
84 #
85 $(SVVS) $(CLOSED_XMODS): FRC
86 @if [ -f $(CLOSED)/uts/sparc/$@/Makefile ]; then \
87 cd $(CLOSED)/uts/sparc/$@; pwd; \
88 $(MAKE) $(NO_STATE) $(TARGET); \
89 else \
90 true; \
91 fi
92 #
93 install_h check: FRC
94 @cd asm; pwd; $(MAKE) $(TARGET)
95 @cd sys; pwd; $(MAKE) $(TARGET)
96 @cd v7/sys; pwd; $(MAKE) $(TARGET)
97 @cd v9/sys; pwd; $(MAKE) $(TARGET)
98 #
99 # Full kernel lint target.
100 LINT_TARGET = globallint
101 #
102 globallint:
103 @-$(ECHO) "\nFULL KERNEL: global crosschecks:"
104 @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)
105 #
106 lint: modlintlib .WAIT $(LINT_DEPS)
107 #
108 include ../Makefile.targ
```

```

*****
13025 Thu Aug 15 11:59:25 2013
new/usr/src/uts/sparc/Makefile.sparc.shared
4027 remove CLOSED_BUILD
*****
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.

25 #
26 # This makefile contains the common definitions for all sparc
27 # implementation architecture independent modules.
28 #

30 #
31 # Define supported builds
32 #
33 DEF_BUILDS = $(DEF_BUILDS64)
34 ALL_BUILDS = $(ALL_BUILDS64)

36 #
37 # Everybody needs to know how to build modstubs.o and to locate unix.o.
38 # Note that unix.o must currently be selected from among the possible
39 # "implementation architectures". Note further, that unix.o is only
40 # used as an optional error check for undefines so (theoretically)
41 # any "implementation architectures" could be used. We choose sun4u
42 # because it is the reference port.
43 #
44 UNIX_DIR = $(UTSBASE)/sun4u/unix
45 GENLIB_DIR = $(UTSBASE)/sun4u/genunix
46 IPDRV_DIR = $(UTSBASE)/sparc/ip
47 MODSTUBS_DIR = $(UNIX_DIR)
48 DSF_DIR = $(UNIX_DIR)
49 LINTS_DIR = $(OBJS_DIR)
50 LINT_LIB_DIR = $(UTSBASE)/sparc/lint-libs/$(OBJS_DIR)

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

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

59 LINT_LIB_64 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lunix.ln
60 GEN_LINT_LIB_64 = $(UTSBASE)/sun4u/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln

```

```

62 LINT_LIB = $(LINT_LIB_$(CLASS))
63 GEN_LINT_LIB = $(GEN_LINT_LIB_$(CLASS))

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

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

71 #
72 # Include the makefiles which define build rule templates, the
73 # collection of files per module, and a few specific flags. Note
74 # that order is significant, just as with an include path. The
75 # first build rule template which matches the files name will be
76 # used. By including these in order from most machine dependent
77 # to most machine independent, we allow a machine dependent file
78 # to be used in preference over a machine independent version
79 # (Such as a machine specific optimization, which preserves the
80 # interfaces.)
81 #
82 include $(UTSBASE)/sparc/Makefile.files
83 include $(UTSBASE)/sparc/v9/Makefile.files
84 include $(UTSTREE)/sun/Makefile.files
85 include $(UTSTREE)/common/Makefile.files

87 #
88 # ----- TRANSITIONAL SECTION -----
89 #

91 #
92 # Not everything which *should* be a module is a module yet. The
93 # following is a list of such objects which are currently part of
94 # genunix but which might someday become kmods. This must be
95 # defined before we include Makefile.uts, or else genunix's build
96 # won't be as parallel as we might like.
97 #
98 NOT_YET_KMODS = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)

100 #
101 # ----- END OF TRANSITIONAL SECTION -----
102 #
103 # Include machine independent rules. Note that this does not imply
104 # that the resulting module from rules in Makefile.uts is machine
105 # independent. Only that the build rules are machine independent.
106 #
107 include $(UTSBASE)/Makefile.uts

109 #
110 # machine specific optimization, override default in Makefile.master
111 #
112 XARCH_32 = -xarch=v8
113 XARCH_64 = -m64
114 XARCH = $(XARCH_$(CLASS))

116 COPTIMIZE_32 = -xO3
117 COPTIMIZE_64 = -xO3
118 COPTIMIZE = $(COPTIMIZE_$(CLASS))

120 CCMODE = -Xa

122 CFLAGS_32 = -xcg92
123 CFLAGS_64 = -xchip=ultra $(CCABS32) $(CCREGSYM)
124 CFLAGS = $(CFLAGS_$(CLASS))

126 CFLAGS += $(XARCH)
127 CFLAGS += $(COPTIMIZE)

```

```

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

147 ASFLAGS     += $(XARCH)

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

153 #
154 #   The following must be defined for all implementations:
155 #
156 #   MODSTUBS:      Module stubs source file.
157 #
158 MODSTUBS      = $(UTSBASE)/sparc/ml/modstubs.s

160 #
161 #   Define the actual specific platforms - obviously none.
162 #
163 MACHINE_DEFS   =

165 #
166 #   Debugging level
167 #
168 #   Special knowledge of which special debugging options effect which
169 #   file is used to optimize the build if these flags are changed.
170 #
171 #   XXX: The above could possibly be done for more flags and files, but
172 #   is left as an experiment to the interested reader. Be forewarned,
173 #   that excessive use could lead to maintenance difficulties.
174 #
175 DEBUG_DEFS_OBJ32 =
176 DEBUG_DEFS_DBG32 = -DDEBUG
177 DEBUG_DEFS_OBJ64 =
178 DEBUG_DEFS_DBG64 = -DDEBUG
179 DEBUG_DEFS        = $(DEBUG_DEFS_$(BUILD_TYPE))

181 DEBUG_COND_OBJ32 :sh = echo \\043
182 DEBUG_COND_DBG32 =
183 DEBUG_COND_OBJ64 :sh = echo \\043
184 DEBUG_COND_DBG64 =
185 IF_DEBUG_OBJ     = $(DEBUG_COND_$(BUILD_TYPE))$(OBJDIR)/

187 $(IF_DEBUG_OBJ)syscall.o := DEBUG_DEFS += -DSYSCALLTRACE
188 $(IF_DEBUG_OBJ)clock.o  := DEBUG_DEFS += -DKSLICE=1

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

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

```

```

194 #
195 #   Collect the preprocessor definitions to be associated with *all*
196 #   files.
197 #
198 ALL_DEFS      = $(MACHINE_DEFS) $(DEBUG_DEFS) $(OPTION_DEFS)
199 #
200 #
201 #   The kernels modules which are "implementation architecture"
202 #   specific for this machine are enumerated below. Note that most
203 #   of these modules must exist (in one form or another) for each
204 #   architecture.
205 #
206 #   Common Drivers (usually pseudo drivers) (/kernel/drv):
207 #
208 DRV_KMODS     += aggr arp audio bl blkdev bofi clone cn conskbd consms cpuid
209 DRV_KMODS     += crypto cryptoadm devinfo dump
210 DRV_KMODS     += dtrace fasttrap fbt lockstat profile sdt systrace dcpd
211 DRV_KMODS     += fssnap icmp icmp6 ip ip6 ipnet ipsecsh
212 DRV_KMODS     += ipsecesp iptun iwscn keysock kmdb kstat ksyms llcl
213 DRV_KMODS     += lofi
214 DRV_KMODS     += log logindmux kssl mm nca physmem pm poll pool
215 DRV_KMODS     += pseudo ptc ptm pts ptsl ramdisk random rsm rts sad
216 DRV_KMODS     += simnet softmac sPPP sPpPtun sy sysevent sysmsg
217 DRV_KMODS     += spdsock
218 DRV_KMODS     += tcp tcp6 tl tnf ttymux udp udp6 wc winlock zcons
219 DRV_KMODS     += ippctl
220 DRV_KMODS     += dld
221 DRV_KMODS     += ipf
222 DRV_KMODS     += rpcib
223 DRV_KMODS     += dlpistub
224 DRV_KMODS     += vnic
225 DRV_KMODS     += xge
226 DRV_KMODS     += rds
227 DRV_KMODS     += rdsv3
228 DRV_KMODS     += chxge
229 DRV_KMODS     += smbsrv
230 DRV_KMODS     += vscan
231 DRV_KMODS     += nsmb
232 DRV_KMODS     += fm
233 DRV_KMODS     += nulldriver
234 DRV_KMODS     += bridge trill
235 DRV_KMODS     += bpf
236 DRV_KMODS     += dca

238 $(CLOSED_BUILD)CLOSED_DRV_KMODS += glm
239 $(CLOSED_BUILD)CLOSED_DRV_KMODS += isp
240 $(CLOSED_BUILD)CLOSED_DRV_KMODS += mpt
241 $(CLOSED_BUILD)CLOSED_DRV_KMODS += qus
242 $(CLOSED_BUILD)CLOSED_DRV_KMODS += se

238 #
239 #   Hardware Drivers in common space
240 #
242 DRV_KMODS     += afe
243 DRV_KMODS     += audio1575
244 DRV_KMODS     += audioens
245 DRV_KMODS     += audiols
246 DRV_KMODS     += audiopl6x
247 DRV_KMODS     += audiopci
248 DRV_KMODS     += audiots
249 DRV_KMODS     += e1000g
250 DRV_KMODS     += efe
251 DRV_KMODS     += hxge
252 DRV_KMODS     += mxfe
253 DRV_KMODS     += pcan

```

```

254 DRV_KMODS      += pcwl
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
269 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ixgb
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      += openeep 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      += socal
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      += fcsn
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 nsctl 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

```

```

326 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ifp
327 $(CLOSED_BUILD)CLOSED_DRV_KMODS += uata
328 $(CLOSED_BUILD)CLOSED_DRV_KMODS += usbser_edge

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      += pcata
333 DRV_KMODS      += pcic

346 $(CLOSED_BUILD)CLOSED_DRV_KMODS += pcser

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

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

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

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

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

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

378 #

```

```

379 #      'User' Modules (/kernel/misc):
380 #
381 MISC_KMODS      += ac97
382 MISC_KMODS      += bignum
383 MISC_KMODS      += consconfig gld ipc nfs_dlboot nfssrv scsi
384 MISC_KMODS      += strplumb swapgeneric tlimod
385 MISC_KMODS      += rpcsec rpcsec_gss kgssapi kmecch_dummy
386 MISC_KMODS      += kmecch_krb5
387 MISC_KMODS      += fssnap_if
388 MISC_KMODS      += hidparser kbtrans usba usbal0 usbs49_fw
389 MISC_KMODS      += sl394
390 MISC_KMODS      += hpcsvc pcihp
391 MISC_KMODS      += rsmops
392 MISC_KMODS      += kcf
393 MISC_KMODS      += ksocket
394 MISC_KMODS      += ibcm
395 MISC_KMODS      += ibdm
396 MISC_KMODS      += ibdma
397 MISC_KMODS      += ibmf
398 MISC_KMODS      += ibt1
399 MISC_KMODS      += sol_ofs
400 MISC_KMODS      += idm
401 MISC_KMODS      += idmap
402 MISC_KMODS      += hook
403 MISC_KMODS      += neti
404 MISC_KMODS      += ctf
405 MISC_KMODS      += mac dls
406 MISC_KMODS      += cmlb
407 MISC_KMODS      += tem
408 MISC_KMODS      += pcicfg fcodem fcpci
409 MISC_KMODS      += scsi_vhci_f_sym scsi_vhci_f_tpgs scsi_vhci_f_asym_sun
410 MISC_KMODS      += scsi_vhci_f_sym_hds
411 MISC_KMODS      += scsi_vhci_f_tape scsi_vhci_f_tpgs_tape
412 MISC_KMODS      += fctl
413 MISC_KMODS      += emlxs_fw
414 MISC_KMODS      += qlc_fw_2200
415 MISC_KMODS      += qlc_fw_2300
416 MISC_KMODS      += qlc_fw_2400
417 MISC_KMODS      += qlc_fw_2500
418 MISC_KMODS      += qlc_fw_6322
419 MISC_KMODS      += qlc_fw_8100
420 MISC_KMODS      += spuni
421 MISC_KMODS      += hwa1480_fw uwba
422 MISC_KMODS      += mii

437 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += klmmod klmops
438 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_asym_lsi
439 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_asym_emc
440 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_sym_emc

```

```

424 #
425 #      Software Cryptographic Providers (/kernel/crypto):
426 #
427 CRYPTO_KMODS    += aes
428 CRYPTO_KMODS    += arcfour
429 CRYPTO_KMODS    += blowfish
430 CRYPTO_KMODS    += des
431 CRYPTO_KMODS    += md4
432 CRYPTO_KMODS    += md5
433 CRYPTO_KMODS    += ecc
434 CRYPTO_KMODS    += rsa
435 CRYPTO_KMODS    += sha1
436 CRYPTO_KMODS    += sha2
437 CRYPTO_KMODS    += swrand

```

```

439 #

```

```

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

449 #
450 # 'Dacf' modules (/kernel/dacf)
451 DACF_KMODS      += consconfig_dacf

453 #
454 #      SVVS Testing Modules (/kernel/strmod):
455 #
456 #      These are streams and driver modules which are not to be
457 #      delivered with a released system. However, during development
458 #      it is convenient to build and install the SVVS kernel modules.
459 #
460 SVVS_KMODS      += lmodb lmode lmodr lmodt svvslo tidg tivc tmax

480 $(CLOSED_BUILD)SVVS      += svvs

462 #
463 #      Modules eXcluded from the product:
464 #
465 XMODS           +=
466 $(CLOSED_BUILD)CLOSED_XMODS = \
467     sdpiib      \
468     wsdrv

467 #
468 #      'Dacf' Modules (/kernel/dacf):
469 #
470 DACF_KMODS      += net_dacf

472 #
473 #      MAC-Type Plugin Modules (/kernel/mac)
474 #
475 MAC_KMODS       += mac_6to4
476 MAC_KMODS       += mac_ether
477 MAC_KMODS       += mac_ipv4
478 MAC_KMODS       += mac_ipv6
479 MAC_KMODS       += mac_wifi
480 MAC_KMODS       += mac_ib

482 #
483 #      socketmod (kernel/socketmod)
484 #
485 SOCKET_KMODS    += sockpfp
486 SOCKET_KMODS    += socksctp
487 SOCKET_KMODS    += socksdp
488 SOCKET_KMODS    += sockrds
489 SOCKET_KMODS    += ksslif

491 #
492 #      kiconv modules (/kernel/kiconv):
493 #
494 KICONV_KMODS    += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc

```


new/usr/src/uts/sparc/sd/Makefile

1

```
*****
4517 Thu Aug 15 11:59:25 2013
new/usr/src/uts/sparc/sd/Makefile
4027 remove CLOSED_BUILD
*****
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/sparc/sd/Makefile
23 #
24 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
27 #
28 # This makefile drives the production of the sd driver kernel module.
29 #
30 # sparc architecture dependent
31 #
32 #
33 #
34 # Path to the base of the uts directory tree (usually /usr/src/uts).
35 #
36 UTSBASE = ../../

38 #
39 # Define the module and object file sets.
40 #
41 MODULE = sd
42 OBJECTS = ${SD_OBJS:%=${OBJDIR}/%}
43 LINTS = ${SD_OBJS:%.o=${LINTS_DIR}/%.lint}
44 ROOTMODULE = ${ROOT_DRV_DIR}/${MODULE}
45 CONF_SRCDIR = ${UTSBASE}/sun/io/scsi/targets
46 WARLOCK_OUT = ${SD_OBJS:%.o=%.ll}
47 WARLOCK_OK = ${MODULE}.ok
48 WLCMD_DIR = ${UTSBASE}/common/io/warlock

50 #
51 # Include common rules.
52 #
53 include ${UTSBASE}/sparc/Makefile.sparc

55 #
56 # Define targets
57 #
58 ALL_TARGET = ${BINARY} ${SRC_CONF}
59 LINT_TARGET = ${MODULE}.lint
60 INSTALL_TARGET = ${BINARY} ${ROOTMODULE} ${ROOT_CONF}
```

new/usr/src/uts/sparc/sd/Makefile

2

```
62 #
63 # lint pass one enforcement
64 #
65 CFLAGS += $(CCVERBOSE)

67 #
68 # Define dependencies on scsi and cmlb
69 #
70 LDFLAGS += -dy -N misc/scsi -N misc/cmlb

72 #
73 # For now, disable these lint checks; maintainers should endeavor
74 # to investigate and remove these for maximum lint coverage.
75 # Please do not carry these forward to new Makefiles.
76 #
77 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
78 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
79 LINTTAGS += -erroff=E_STATIC_UNUSED
80 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON

82 CERRWARN += _gcc=-Wno-unused-variable
83 CERRWARN += _gcc=-Wno-unused-function
84 CERRWARN += _gcc=-Wno-unused-label
85 CERRWARN += _gcc=-Wno-parentheses
86 CERRWARN += _gcc=-Wno-type-limits
87 CERRWARN += _gcc=-Wno-uninitialized

89 #
90 # Default build targets.
91 #
92 .KEEP_STATE:

94 all: $(ALL_DEPS)

96 def: $(DEF_DEPS)

98 clean: $(CLEAN_DEPS)
99 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

101 clobber: $(CLOBBER_DEPS)
102 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

104 lint: $(LINT_DEPS)

106 modlintlib: $(MODLINTLIB_DEPS)

108 clean.lint: $(CLEAN_LINT_DEPS)

110 install: $(INSTALL_DEPS)

112 #
113 # Include common targets.
114 #
115 include ${UTSBASE}/sparc/Makefile.targ

118 #
119 # Defines for local commands.
120 #
121 WARLOCK = warlock
122 WLCC = wlcc
123 TOUCH = touch
124 TEST = test

126 #
127 # Warlock targets
```

```

128 #
129 # Note that in warlock_with_{esp,isp} it is important to load sd.ll
130 # before {isp,esp}.ll; the reason is that both have _init/_info/_fini
131 # and warlock can only handle one extern function by a given name;
132 # any loaded after the first are ignored.

134 SCSI_FILES = $(SCSI_OBJS:%.o=-l ../scsi/%.ll)
135 CMLB_FILES = $(CMLB_OBJS:%.o=-l ../cmlb/%.ll)

137 WARLOCK_TARGETS = warlock_alone warlock_with_esp warlock_with_fas
138 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_esp warlock_with_glm \
139     warlock_with_uata warlock_with_mpt

139 warlock: $(WARLOCK_TARGETS)

141 warlock_alone: $(WARLOCK_OK)

143 $(WARLOCK_OK): $(WLCMD_DIR)/sd.wlcmd $(WARLOCK_OUT) scsi_files \
144     warlock_ddi.files cmlb_files
145 $(WARLOCK) -c $(WLCMD_DIR)/sd.wlcmd $(WARLOCK_OUT) $(SCSI_FILES) \
146     $(CMLB_FILES) \
147     -l ../warlock/ddi_dki_impl.ll
148 $(TOUCH) $@

150 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
151 $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

153 warlock_with_esp: $(WLCMD_DIR)/sd_with_esp.wlcmd $(WARLOCK_OUT) scsi_files \
154     esp_files warlock_ddi.files cmlb_files
155 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_esp.wlcmd \
156     $(WARLOCK_OUT) ../esp/esp $(SCSI_FILES) $(CMLB_FILES) \
157     -l ../warlock/ddi_dki_impl.ll

159 warlock_with_fas: $(WLCMD_DIR)/sd_with_fas.wlcmd $(WARLOCK_OUT) scsi_files \
160     fas_files warlock_ddi.files cmlb_files
161 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_fas.wlcmd \
162     $(WARLOCK_OUT) ../fas/fas \
163     ../fas/fas_callbacks \
164     $(SCSI_FILES) $(CMLB_FILES) \
165     -l ../warlock/ddi_dki_impl.ll

169 warlock_with_esp: $(WLCMD_DIR)/sd_with_esp.wlcmd $(WARLOCK_OUT) scsi_files \
170     isp_files warlock_ddi.files cmlb_files
171 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_esp.wlcmd \
172     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/isp/isp $(SCSI_FILES) \
173     $(CMLB_FILES) \
174     -l ../warlock/ddi_dki_impl.ll

176 warlock_with_glm: $(WLCMD_DIR)/sd_with_glm.wlcmd $(WARLOCK_OUT) scsi_files \
177     glm_files warlock_ddi.files cmlb_files
178 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_glm.wlcmd \
179     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/glm/glm $(SCSI_FILES) \
180     $(CMLB_FILES) \
181     -l ../warlock/ddi_dki_impl.ll

183 warlock_with_uata: $(WLCMD_DIR)/sd_with_uata.wlcmd $(WARLOCK_OUT) scsi_files \
184     uata_files warlock_ddi.files cmlb_files
185 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_uata.wlcmd \
186     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/uata/*.ll $(SCSI_FILES) \
187     $(CMLB_FILES) \
188     -l ../warlock/ddi_dki_impl.ll

190 warlock_with_mpt: $(WLCMD_DIR)/sd_with_mpt.wlcmd $(WARLOCK_OUT) scsi_files \
191     mpt_files warlock_ddi.files cmlb_files
192 $(WARLOCK) -c $(WLCMD_DIR)/sd_with_mpt.wlcmd \
193     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/mpt/*.ll $(SCSI_FILES) \

```

```

194     $(CMLB_FILES) \
195     -l ../warlock/ddi_dki_impl.ll

167 cmlb_files:
168     @cd ../cmlb; pwd; $(MAKE) warlock

171 scsi_files:
172     @cd ../scsi; pwd; $(MAKE) warlock

174 esp_files:
175     @cd ../esp; pwd; $(MAKE) warlock

177 fas_files:
178     @cd ../fas; pwd; $(MAKE) warlock

210 isp_files:
211     @cd $(CLOSED)/uts/sparc/isp; pwd; $(MAKE) warlock

213 glm_files:
214     @cd $(CLOSED)/uts/sparc/glm; pwd; $(MAKE) warlock

216 uata_files:
217     @cd $(CLOSED)/uts/sparc/uata; pwd; $(MAKE) warlock

219 mpt_files:
220     @cd $(CLOSED)/uts/sparc/mpt; pwd; $(MAKE) warlock

180 warlock_ddi.files:
181     @cd ../warlock; pwd; $(MAKE) warlock

```

```

*****
3926 Thu Aug 15 11:59:25 2013
new/usr/src/uts/sparc/ses/Makefile
4027 remove CLOSED_BUILD
*****
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/sparc/ses/Makefile
23 #
24 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
27 #
28 # This makefile drives the production of the ses driver kernel module.
29 #
30 # sparc architecture dependent
31 #
32 #
33 #
34 # Path to the base of the uts directory tree (usually /usr/src/uts).
35 #
36 UTSBASE = ../../

38 #
39 # Define the module and object file sets.
40 #
41 MODULE      = ses
42 OBJECTS     = $(SES_OBJS:%=$(OBJS_DIR)/%)
43 LINTS       = $(SES_OBJS:%.o=$(LINTS_DIR)/%.ln)
44 ROOTMODULE  = $(ROOT_DRV_DIR)/$(MODULE)
45 CONF_SRCDIR = $(UTSBASE)/sun/io/scsi/targets
46 WARLOCK_OUT = $(SES_OBJS:%.o=%11)
47 WARLOCK_OK  = $(MODULE).ok
48 WLCMD_DIR   = $(UTSBASE)/common/io/warlock

50 #
51 # Include common rules.
52 #
53 include $(UTSBASE)/sparc/Makefile.sparc

55 #
56 # Define targets
57 #
58 ALL_TARGET = $(BINARY) $(SRC_CONFIL)
59 LINT_TARGET = $(MODULE).lint
60 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFIL)
61 CLEANFILES += $(WARLOCK_TARGETS)

```

```

63 #
64 # Define dependency on scsi
65 #
66 LDFLAGS += -dy -N misc/scsi

68 #
69 # For now, disable these lint checks; maintainers should endeavor
70 # to investigate and remove these for maximum lint coverage.
71 # Please do not carry these forward to new Makefiles.
72 #
73 LINTTAGS      += -erroff=E_BAD_PTR_CAST_ALIGN
74 LINTTAGS      += -erroff=E_ASSIGN_NARROW_CONV
75 LINTTAGS      += -erroff=E_SUSPICIOUS_COMPARISON

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

82 def:          $(DEF_DEPS)

84 all:          $(ALL_DEPS)

86 clean:        $(CLEAN_DEPS)
87               $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

89 clobber:      $(CLOBBER_DEPS)
90               $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

92 modlintlib:   $(MODLINTLIB_DEPS) lint64

94 clean.lint:   $(CLEAN_LINT_DEPS)

96 install:      $(INSTALL_DEPS)

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

104 #
105 # Defines for local commands.
106 #
107 WARLOCK      = warlock
108 WLCC         = wlcc
109 TOUCH        = touch
110 TEST         = test

112 #
113 # Warlock targets
114 #
115 # XXX FIX ME.  this can be improved quite a bit

117 SES_FILES    = $(MODULE).11
118 SCSI_FILES   = $(SCSI_OBJS:%.o=-1 ../scsi/%.11)

120 WARLOCK_TARGETS = $(WARLOCK_OK) warlock_with_esp.ok warlock_with_fas.ok
121 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_esp.ok warlock_with_glm.ok

122 warlock:     $(WARLOCK_TARGETS)

124 $(WARLOCK_OK): $(WLCMD_DIR)/$(MODULE).wlcmd $(SES_FILES)
125                 @cd ../warlock; $(MAKE) warlock
126                 @cd ../scsi; $(MAKE) warlock

```

```
127 $(WARLOCK) -c $(WLCMD_DIR)/$(MODULE).wlcmd $(SES_FILES) ../warlock/scsi.
128 -l ../warlock/ddi_dki_impl.ll \
129 $(SCSI_FILES)
130 @ $(TOUCH) $@

132 warlock_with_esp.ok: $(WLCMD_DIR)/ses_with_esp.wlcmd $(SES_FILES)
133 @cd ../warlock; $(MAKE) warlock
134 @cd ../scsi; $(MAKE) warlock
135 @cd ../esp; $(MAKE) warlock;
136 $(WARLOCK) -c $(WLCMD_DIR)/ses_with_esp.wlcmd $(SES_FILES) ../esp/esp \
137 ../warlock/scsi.ll \
138 -l ../warlock/ddi_dki_impl.ll \
139 $(SCSI_FILES)
140 @ $(TOUCH) $@

142 warlock_with_fas.ok: $(WLCMD_DIR)/ses_with_fas.wlcmd $(SES_FILES)
143 @cd ../warlock; $(MAKE) warlock
144 @cd ../scsi; $(MAKE) warlock
145 @cd ../fas; $(MAKE) warlock;
146 $(WARLOCK) -c $(WLCMD_DIR)/ses_with_fas.wlcmd $(SES_FILES) \
147 ../fas/fas ../fas/fas_callbacks \
148 ../warlock/scsi.ll \
149 -l ../warlock/ddi_dki_impl.ll \
150 $(SCSI_FILES)
151 @ $(TOUCH) $@

154 warlock_with_isp.ok: $(WLCMD_DIR)/ses_with_isp.wlcmd $(SES_FILES)
155 @cd ../warlock; $(MAKE) warlock
156 @cd ../scsi; $(MAKE) warlock
157 @cd $(CLOSED)/uts/sparc/isp; $(MAKE) warlock;
158 $(WARLOCK) -c $(WLCMD_DIR)/ses_with_isp.wlcmd $(SES_FILES) \
159 $(CLOSED)/uts/sparc/isp/isp \
160 ../warlock/scsi.ll \
161 -l ../warlock/ddi_dki_impl.ll \
162 $(SCSI_FILES)
163 @ $(TOUCH) $@

165 warlock_with_glm.ok: $(WLCMD_DIR)/ses_with_glm.wlcmd $(SES_FILES)
166 @cd ../warlock; $(MAKE) warlock
167 @cd ../scsi; $(MAKE) warlock
168 @cd $(CLOSED)/uts/sparc/glm; $(MAKE) warlock;
169 $(WARLOCK) -c $(WLCMD_DIR)/ses_with_glm.wlcmd $(SES_FILES) \
170 $(CLOSED)/uts/sparc/glm/glm \
171 ../warlock/scsi.ll \
172 -l ../warlock/ddi_dki_impl.ll \
173 $(SCSI_FILES)
174 @ $(TOUCH) $@

153 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
154 $(WLCC) $(CPPFLAGS) -o $@ $<
```

```

*****
4201 Thu Aug 15 11:59:25 2013
new/usr/src/uts/sparc/st/Makefile
4027 remove CLOSED_BUILD
*****
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 # uts/sparc/st/Makefile
22 #
23 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 # Copyright (c) 2011 Bayard G. Bell. All rights reserved.
26 #
27 # This makefile drives the production of the st driver kernel module.
28 #
29 # sparc architecture dependent
30 #
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 = st
41 OBJECTS = $(ST_OBJS:%=$(OBJS_DIR)/%)
42 LINTS = $(ST_OBJS:%.o=$(LINTS_DIR)/%.ln)
43 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
44 CONF_SRCDIR = $(UTSBASE)/sun/io/scsi/targets
45 WARLOCK_OUT = $(ST_OBJS:%.o=%%.ll)
46 WARLOCK_OK = $(MODULE).ok
47 WLCMD_DIR = $(UTSBASE)/common/io/warlock

49 #
50 # Include common rules.
51 #
52 include $(UTSBASE)/sparc/Makefile.sparc

54 #
55 # Define targets
56 #
57 ALL_TARGET = $(BINARY) $(SRC_CONFFILE)
58 LINT_TARGET = $(MODULE).lint
59 INSTALL_TARGET = $(BINARY) $(ROOTMODULE) $(ROOT_CONFFILE)

61 #

```

```

62 # lint pass one enforcement
63 #
64 CFLAGS += $(CCVERBOSE)

66 #
67 # Define dependency on scsi
68 #
69 LDFLAGS += -dy -N misc/scsi

71 #
72 # For now, disable these lint checks; maintainers should endeavor
73 # to investigate and remove these for maximum lint coverage.
74 # Please do not carry these forward to new Makefiles.
75 #
76 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
77 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
78 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON

80 CERRWARN += -_gcc=-Wno-parentheses
81 CERRWARN += -_gcc=-Wno-switch
82 CERRWARN += -_gcc=-Wno-uninitialized

84 #
85 # Default build targets.
86 #
87 .KEEP_STATE:

89 all: $(ALL_DEPS)

91 def: $(DEF_DEPS)

93 clean: $(CLEAN_DEPS)
94 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

96 clobber: $(CLOBBER_DEPS)
97 $(RM) $(WARLOCK_OUT) $(WARLOCK_OK)

99 lint: $(LINT_DEPS)

101 modlintlib: $(MODLINTLIB_DEPS)

103 clean.lint: $(CLEAN_LINT_DEPS)

105 install: $(INSTALL_DEPS)

107 # Include common targets.
108 #
109 include $(UTSBASE)/sparc/Makefile.targ

111 #
112 # Defines for local commands.
113 #
114 WARLOCK = warlock
115 WLCC = wlcc
116 TOUCH = touch
117 TEST = test

119 #
120 # Warlock targets
121 #
122 # Note that in warlock_with_{esp,isp} it is important to load st.ll
123 # before {isp,esp}.ll; the reason is that both have _init/_info/_fini
124 # and warlock can only handle one extern function by a given name;
125 # any loaded after the first are ignored.

127 SCSI_FILES = $(SCSI_OBJS:%.o=-l ../scsi/%.ll)

```

```
129 WARLOCK_TARGETS = warlock_alone warlock_with_esp warlock_with_fas
130 $(CLOSED_BUILD)WARLOCK_TARGETS += warlock_with_esp warlock_with_glm

131 warlock: $(WARLOCK_TARGETS)

133 warlock_alone: $(WARLOCK_OK)

135 warlock_with_esp: $(WLCMD_DIR)/st_with_esp.wlcmd $(WARLOCK_OUT) scsi_files \
136     esp_files warlock_ddi.files
137     $(WARLOCK) -c $(WLCMD_DIR)/st_with_esp.wlcmd \
138     $(WARLOCK_OUT) ../esp/esp $(SCSI_FILES) \
139     -l ../warlock/ddi_dki_impl.ll

141 warlock_with_fas: $(WLCMD_DIR)/st_with_fas.wlcmd $(WARLOCK_OUT) scsi_files \
142     fas_files warlock_ddi.files
143     $(WARLOCK) -c $(WLCMD_DIR)/st_with_fas.wlcmd \
144     $(WARLOCK_OUT) \
145     ../fas/fas ../fas/fas_callbacks \
146     $(SCSI_FILES) \
147     -l ../warlock/ddi_dki_impl.ll

150 warlock_with_esp: $(WLCMD_DIR)/st_with_esp.wlcmd $(WARLOCK_OUT) scsi_files \
151     isp_files warlock_ddi.files
152     $(WARLOCK) -c $(WLCMD_DIR)/st_with_esp.wlcmd \
153     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/isp/isp $(SCSI_FILES) \
154     -l ../warlock/ddi_dki_impl.ll

156 warlock_with_glm: $(WLCMD_DIR)/st_with_glm.wlcmd $(WARLOCK_OUT) scsi_files \
157     glm_files warlock_ddi.files
158     $(WARLOCK) -c $(WLCMD_DIR)/st_with_glm.wlcmd \
159     $(WARLOCK_OUT) $(CLOSED)/uts/sparc/glm/glm $(SCSI_FILES) \
160     -l ../warlock/ddi_dki_impl.ll

149 scsi_files:
150     @cd ../scsi; pwd; $(MAKE) warlock

152 esp_files:
153     @cd ../esp; pwd; $(MAKE) warlock

155 fas_files:
156     @cd ../fas; pwd; $(MAKE) warlock

171 isp_files:
172     @cd $(CLOSED)/uts/sparc/isp; pwd; $(MAKE) warlock

174 glm_files:
175     @cd $(CLOSED)/uts/sparc/glm; pwd; $(MAKE) warlock

158 st.ok: $(WLCMD_DIR)/st.wlcmd st.ll st_conf.ll scsi_files warlock_ddi.files
159     $(WARLOCK) -c $(WLCMD_DIR)/st.wlcmd $(WARLOCK_OUT) $(SCSI_FILES) \
160     -l ../warlock/ddi_dki_impl.ll
161     $(TOUCH) $@

163 %.ll: $(UTSBASE)/common/io/scsi/targets/%.c
164     $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

166 warlock_ddi.files:
167     @cd ../warlock; pwd; $(MAKE) warlock

169 scsi.files:
170     @cd ../scsi; pwd; $(MAKE) warlock
```

```

*****
3645 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sparc/warlock/Makefile
4027 remove CLOSED_BUILD
*****
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 # Copyright (c) 1998, 2010, Oracle and/or its affiliates. All rights reserved.
22 #
23 #     sparc architecture dependent
24 #
25 # uts/sparc/warlock/Makefile
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         = warlock
35 ROOTMODULE     = $(ROOT_DRV_DIR)/$(MODULE)

37 .KEEP_STATE:

39 CFLAGS += -I../common/sys -I../sun/sys/scsi -D_KERNEL
40 #
41 #     Defines for local commands.
42 #
43 WARLOCK        = warlock
44 WLCC           = wlcc
45 TOUCH         = touch
46 TEST          = test

48 include $(UTSBASE)/sparc/Makefile.sparc

50 #
51 #     lock_lint rules
52 #
53 all:          warlock warlock.1394 warlock.ecpp warlock.scsi \
54             warlock.usb warlock.ib warlock.sata warlock.wc

56 warlock: $(MODULE).ok

58 warlock.ok: ddi_dki_impl.ll scsi.ll
59             $(TOUCH) $@

61 %.ll: $(UTSBASE)/common/io/warlock/%.c

```

```

62             $(WLCC) $(CPPFLAGS) -DDEBUG -o $@ $<

64 warlock.usb:
65     @cd ../usba; $(MAKE) clean; $(MAKE) warlock
66     @cd ../ohci; $(MAKE) clean; $(MAKE) warlock
67     @cd ../uhci; $(MAKE) clean; $(MAKE) warlock
68     @cd ../ehci; $(MAKE) clean; $(MAKE) warlock
69     @cd ../hid; $(MAKE) clean; $(MAKE) warlock
70     @cd ../scsa2usb; $(MAKE) clean; $(MAKE) warlock
71     @cd ../usb_ac; $(MAKE) clean; $(MAKE) warlock
72     @cd ../usb_as; $(MAKE) clean; $(MAKE) warlock
73     @cd ../usb_ah; $(MAKE) clean; $(MAKE) warlock
74     @cd ../ugen; $(MAKE) clean; $(MAKE) warlock
75     @cd ../usb_mid; $(MAKE) clean; $(MAKE) warlock
76     @cd ../usb_ia; $(MAKE) clean; $(MAKE) warlock
77     @cd ../usbprn; $(MAKE) clean; $(MAKE) warlock
78     @cd ../usbser; $(MAKE) clean; $(MAKE) warlock
79     @cd ../usbsksp; $(MAKE) clean; $(MAKE) warlock
80     @cd ../usbspri; $(MAKE) clean; $(MAKE) warlock
81     @cd ../usbsacm; $(MAKE) clean; $(MAKE) warlock
82     @cd ../usbecm; $(MAKE) clean; $(MAKE) warlock
83     @cd ../usbskel; $(MAKE) clean; $(MAKE) warlock
84 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/usbser_edge; \
85                 $(MAKE) clean; $(MAKE) warlock

85 warlock.scsi:
86     @cd ../dad; $(MAKE) clean; $(MAKE) warlock
87     @cd ../dada; $(MAKE) clean; $(MAKE) warlock
88     @cd ../esp; $(MAKE) clean; $(MAKE) warlock
89     @cd ../fas; $(MAKE) clean; $(MAKE) warlock
90     @cd ../sd; $(MAKE) clean; $(MAKE) warlock
91     @cd ../ses; $(MAKE) clean; $(MAKE) warlock
92     @cd ../st; $(MAKE) clean; $(MAKE) warlock
93     @cd ../ssd; $(MAKE) clean; $(MAKE) warlock
94 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/uata; $(MAKE) clean; $(MAKE) warlock
95 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/glm; $(MAKE) clean; $(MAKE) warlock
96 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/mpt; $(MAKE) clean; $(MAKE) warlock
97 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/ifp; $(MAKE) clean; $(MAKE) warlock
98 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/isp; $(MAKE) clean; $(MAKE) warlock

95 warlock.1394:
96     @cd ../s1394; $(MAKE) clean; $(MAKE) warlock
97     @cd ../hci1394; $(MAKE) clean; $(MAKE) warlock
98     @cd ../scsa1394; $(MAKE) clean; $(MAKE) warlock
99     @cd ../avl1394; $(MAKE) clean; $(MAKE) warlock

101 warlock.ecpp:
102     @cd ../ecpp; $(MAKE) clean; $(MAKE) warlock

104 warlock.ib:
105     @cd ../ibmf; $(MAKE) clean; $(MAKE) warlock
106     @cd ../ib; $(MAKE) clean; $(MAKE) warlock
107     @cd ../ibtl; $(MAKE) clean; $(MAKE) warlock
108     @cd ../ibcm; $(MAKE) clean; $(MAKE) warlock
109     @cd ../ibd; $(MAKE) clean; $(MAKE) warlock
110 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/tavor; $(MAKE) clean; $(MAKE) warlock
111 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/hermon; $(MAKE) clean; $(MAKE) warlock
112 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/dapit; $(MAKE) clean; $(MAKE) warlock

111 warlock.sata:
112     @cd ../sata; $(MAKE) clean; $(MAKE) warlock
113     @cd ../si3124; $(MAKE) clean; $(MAKE) warlock
114     @cd ../ahci; $(MAKE) clean; $(MAKE) warlock
115     @cd ../nv_sata; $(MAKE) clean; $(MAKE) warlock
116 $(CLOSED_BUILD) @cd $(CLOSED)/uts/sparc/marvell88sx; \
117                 $(MAKE) clean; $(MAKE) warlock

```

new/usr/src/uts/sparc/warlock/Makefile

3

```
117 warlock.wc:  
118     @cd ../wc; $(MAKE) clean; $(MAKE) warlock
```



```
*****
```

```
2989 Thu Aug 15 11:59:26 2013
```

```
new/usr/src/uts/sun/sys/Makefile
```

```
4027 remove CLOSED_BUILD
```

```
*****
```

```
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 #
26 # include global definitions
27 include ../../../../Makefile.master
```

```
29 HDRS= \
30 avintr.h \
31 bootconf.h \
32 dkmpio.h
33 fdreg.h          \
34 obpdefs.h       \
35 promif.h        \
36 ser_async.h     \
37 socialio.h      \
38 social_cq_defs.h \
39 ttymux.h        \
40 zsdev.h          \
                    fdvar.h \
                    \
                    promimpl.h \
                    ser_zscc.h \
                    socialreg.h \
                    socialmap.h \
                    socialvar.h \
                    ttymuxuser.h \
```

```
42 SCSCIADHDRS= \
43 fascmd.h      fasdma.h \
44 fasreg.h      fasvar.h      ifpio.h      sfvar.h
```

```
46 CLOSED_SCSCIAD = $(CLOSED)/uts/sun/sys/scsi/adapters
```

```
48 CLOSED_SCSCIADHDRS= \
49 ifpcmd.h      ifpmail.h      ifpreg.h      ifpvar.h \
50 ispcmd.h      ispmail.h      ispreg.h      ispvar.h
```

```
46 SCSTITARGHDRS= \
```

```
48 ROOTDIR=      $(ROOT)/usr/include/sys
49 ROOTDIRS=     $(ROOTDIR) \
50               $(ROOTDIR)/audio \
51               $(ROOTDIR)/scsi/adapters \
52               $(ROOTDIR)/scsi/targets
```

```
54 Roothdrs= $(HDRS:%=$(ROOTDIR)/%)
61 $(CLOSED_BUILD)Roothdrs += $(CLOSED_HDRS:%=$(ROOTDIR)/%)
```

```
55 ROOTFCHDRS= $(FCHDRS:%=$(ROOTDIR)/fc4/%)
56 ROOTSCSIADHDRS= $(SCSIADHDRS:%=$(ROOTDIR)/scsi/adapters/%)
64 $(CLOSED_BUILD)ROOTSCSIADHDRS += \
65     $(CLOSED_SCSCIADHDRS:%=$(ROOTDIR)/scsi/adapters/%)
57 ROOTSCSITARGHDRS= $(SCSITARGHDRS:%=$(ROOTDIR)/scsi/targets/%)
```

```
60 # install rules
61 $(ROOTDIR)/%: %
62     $(INS.file)
```

```
73 $(ROOTDIR)/%: $(CLOSED)/uts/sun/sys/%
74     $(INS.file)
```

```
64 $(ROOTDIR)/audio/%: audio/%
65     $(INS.file)
```

```
67 $(ROOTDIR)/scsi/adapters/%: scsi/adapters/%
68     $(INS.file)
```

```
82 $(ROOTDIR)/scsi/adapters/%: $(CLOSED_SCSCIAD)/%
83     $(INS.file)
```

```
70 $(ROOTDIR)/scsi/targets/%: scsi/targets/%
71     $(INS.file)
```

```
73 # check files really don't exist
```

```
75 audio/%.check:      audio/%.h
76     $(DOT_H_CHECK)
```

```
78 scsi/adapters/%.check: scsi/adapters/%.h
79     $(DOT_H_CHECK)
```

```
81 scsi/targets/%.check: scsi/targets/%.h
82     $(DOT_H_CHECK)
```

```
84 CHECKHDRS= \
85     $(HDRS:%.h=%.check) \
86     $(FCHDRS:%.h=fc4/%.check) \
87     $(SCSIADHDRS:%.h=scsi/adapters/%.check) \
88     $(SCSITARGHDRS:%.h=scsi/targets/%.check)
```

```
105 $(CLOSED_BUILD)CHECKHDRS += \
106     $(CLOSED_HDRS:%.h=$(CLOSED)/uts/sun/sys/%.check) \
107     $(CLOSED_SCSCIADHDRS:%.h=$(CLOSED_SCSCIAD)/%.check)
```

```
90 # headers which won't quite meet the standards...
```

```
91 #
92 # devops.h has a macro where the formal parameters to the macro are greater
93 # than 80 characters. cpp (or the equivalent built into acomp) does not allow
94 # continuation line breaks in the formal parameter list. This could be fixed
95 # by giving shorter names to the formal parameters, but the right fix is to
96 # fix cpp. (Also, /* CSTYLE */ doesn't seem to fix this.
97 #
98 devops.check := CSTYLE_TAIL = | grep -v "line > 80 characters" | true
```

```
100 .KEEP_STATE:
```

```
102 .PARALLEL:      $(CHECKHDRS) $(ROOTHDRS) $(ROOTAUDHDRS) $(ROOTAUDIMPLHDRS) \
103                 $(ROOTSCSIADHDRS) $(ROOTSCSITARGHDRS) $(ROOTXHDRS) \
104                 $(ROOTFCHDRS)
```

```
106 install_h:      $(ROOTDIRS) .WAIT $(ROOTHDRS) \
107                 $(ROOTAUDHDRS) $(ROOTAUDIMPLHDRS) \
108                 $(ROOTSCSIADHDRS) $(ROOTSCSITARGHDRS) $(ROOTFCHDRS)
```

new/usr/src/uts/sun/sys/Makefile

3

```
110 $(ROOTDIRS):  
111     $(INS.dir)  
113 check:  $(CHECKHDRS)
```

new/usr/src/uts/sun4u/Makefile

1

```
*****
7855 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sun4u/Makefile
4027 remove CLOSED_BUILD
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # This makefile drives the production of all implementation architecture
26 # dependent modules for the sun4u architecture.
27 #
28 #
29 UTSBASE = ..
30 #
31 include Makefile.sun4u
32 #
33 #
34 # The following are SPARC specific (rather than sun4u) specific modules
35 # which are required for the sun4u kernel to completely lint. They are
36 # not involved in the build in any other way. In order to minimize
37 # build time, it is assumed that they are up to date. But since sun4u
38 # is really a separate architecture we cannot use the v7 sparc modules.
39 #
40 SPARC_LIB_DIR = $(UTSBASE)/sparc/lint-libs/$(OBSJ_DIR)
41 #
42 SPARC_LINTS =
43 #
44 #
45 #
46 #
47 LINT_LIBS = $(LINT_LIB) \
48             $(LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
49             $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
50             $(SPARC_LINTS:%=$(SPARC_LIB_DIR)/llib-1%.ln)
51 #
52 def := TARGET= def
53 def.prereq := TARGET= def
54 all := TARGET= all
55 all.prereq := TARGET= all
56 install := TARGET= install
57 install.prereq := TARGET= all
58 install_h := TARGET= install_h
59 install_h.prereq := TARGET= install_h
60 clean := TARGET= clean
```

new/usr/src/uts/sun4u/Makefile

2

```
61 clobber := TARGET= clobber
62 lint := TARGET= lint
63 lint.prereq := TARGET= lint
64 lintlib := TARGET= lintlib
65 modlintlib := TARGET= modlintlib
66 modlist := TARGET= modlist
67 modlist.modlist.sparc := NO_STATE= -K $$MODSTATE$$$$
68 clean.lint := TARGET= clean.lint
69 check := TARGET= check
70 #
71 .KEEP_STATE:
72 #
73 .PARALLEL: $(PARALLEL_KMODS) $(XMODS) \
74            $(IMPLEMENTATIONS) \
75            $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
76            $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) \
77            modlist.modlist.sparc
78 #
79 # Override for CPU_KMODS... they cannot be built
80 # in parallel
81 .NO_PARALLEL: $(CPU_KMODS)
82 #
83 def all clean clobber clean.lint: genassym unix .WAIT \
84     $(KMODS) $(XMODS) \
85     $(IMPLEMENTATIONS) \
86     $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
87     $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)
88 #
89 # list the modules under sun4u.
90 modlist: unix $(KMODS) $(XMODS) \
91            $(IMPLEMENTATIONS:.WAIT=)
92 modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
93            $(IMPLEMENTATIONS:.WAIT=) $(CLOSED_IMPLEMENTATIONS)
94 #
95 # list the modules for Install -k sun4u.
96 modlist.karch: modlist.modlist.sparc
97 #
98 modlist.sparc:
99     @cd $(SRC)/uts/sparc; pwd; $(MAKE) $(NO_STATE) modlist
100 #
101 install: install_platforms genassym unix .WAIT $(KMODS) \
102         $(XMODS) $(IMPLEMENTATIONS)
103 install: install_platforms genassym unix .WAIT $(KMODS) $(CLOSED_KMODS) \
104         $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS)
105 #
106 lintlib: unix
107 #
108 modlintlib: $(LINT_KMODS)
109 modlintlib: $(LINT_KMODS) $(CLOSED_LINT_KMODS)
110 #
111 genassym unix $(KMODS): FRC
112 @cd %@; pwd; $(MAKE) $(NO_STATE) $(TARGET)
113 #
114 PRIVS_C = $(UTSBASE)/common/os/priv-const.c
115 $(PRIVS_C): $(PRIVS_AWK) $(PRIVS_DEF)
116 $(NAWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) cfile=%@
```

```

118 #
119 # Prerequisites
120 #
121 # The uts/Makefile defines build parallelism for sun4 platforms such that sparc,
122 # sun4u and sun4v are all built in parallel. Also this Makefile specifies that
123 # all IMPLEMENTATIONS sun4u sub-platforms are built in parallel. This requires
124 # building certain parts before the parallel build can start. The uts/Makefile
125 # appends the '.prereq' string to the original target and executes this Makefile
126 # to build any prerequisites needed before the full parallel build can start.
127 # After that make continues with normal targets.
128 #
129 # Any build prerequisites for sun4 and IMPLEMENTATIONS builds should be
130 # described here.
131 #
132 # genassym is used to build dtrace and genunix, so it should be built first.
133 #
134 # priv_const.c is required to build genunix.
135 #
136 # genunix is used by everyone to ctfmerge with. Genunix is merged with sparc/ip
137 # so as a side effect this dependency builds sparc/ip as part of the
138 # prerequisites.
139 #
140 # unix is not required by itself but several sun4u platforms require
141 # sun4u/platmod to be present. The easiest way to achieve this is to build
142 # sun4u/unix first since sun4u/unix Makefile builds sun4u/platform correctly.
143 # This causes full sun4u/unix to be built before all sun4u platforms and
144 # before uts/sun4v and uts/sparc, but it acceptable since it is not spending
145 # too much time building sun4u/unix.
146 #
147 all.prereq def.prereq install.prereq: genassym genunix unix

149 #
150 # Various sun4u platforms expect proto/root_sparc/platform/sun4u/include to be
151 # present. This is handled by running make install_h in sun4u/unix directory
152 # first.
153 #
154 install_h.prereq: FRC
155     @cd sys; pwd; $(MAKE) $(TARGET)

157 #
158 # sun4u/unix and sun4u/genunix should be linted first since sparc does global
159 # cross-check with these lint libraries. The sun4u/unix and sun4u/genunix can be
160 # linted in parallel.
161 #
162 LINT_PREREQ = unix.lint genunix.lint
163 lint.prereq: $(LINT_PREREQ)

165 .PARALLEL: $(LINT_PREREQ)

167 $(LINT_PREREQ):
168     @cd $(@:%.lint=%); pwd; $(MAKE) $(TARGET)

170 #
171 # Nothing to do with any other prerequisites
172 #
173 %.prereq:

175 #
176 # Platform inter-dependencies
177 #
178 lw8: serengeti

180 quasar: darwin

182 #
183 # The genunix requires priv_const.c file to be generated first.

```

```

184 #
185 genunix: $(PRIVS_C)

187 #
188 # Rules
189 #

191 $(IMPLEMENTATIONS): FRC
192     @cd $@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)

195 $(CLOSED_IMPLEMENTATIONS): FRC
196     cd $(CLOSED)/uts/sun4u/$@; pwd; \
197     THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET); \

194 $(XMODS): FRC
195     @if [ -f $@/Makefile ]; then \
196     cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
197     else \
198     true; \
199     fi

206 $(CLOSED_XMODS): FRC
207     @if [ -f $(CLOSED)/uts/sun4u/$@/Makefile ]; then \
208     cd $(CLOSED)/uts/sun4u/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
209     else \
210     true; \
211     fi

213 $(CLOSED_KMODS): FRC
214     cd $(CLOSED)/uts/sun4u/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

201 install_h check: install_platforms $(IMPLEMENTATIONS) \
217     $(CLOSED_IMPLEMENTATIONS) FRC
202     @cd sys; pwd; $(MAKE) $(TARGET)
203     @cd vm; pwd; $(MAKE) $(TARGET)

205 #
206 # Rules for the /platforms directories. This is hardwired here because
207 # the first stage of the project (KBI) only implements the userland
208 # changes, but the only reasonable place to record the aliases is
209 # here in kernel land.
210 #
211 $(ROOT_PLAT_DIRS): $(ROOT_PLAT_DIR)
212     -$(INS.dir)

214 #
215 # create directories in /usr/platform/ for the implementations that are
216 # defined in $(IMPLEMENTED_PLATFORM)
217 # (eg. SUNW,Ultra-1)
218 #
219 # Foreach $(IMPLEMENTED_PLATFORM) there can be a list of $(LINKED_PLATFORMS)
220 # that are linked to it.
221 #
222 $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM): $(USR_PLAT_DIR)
223     -$(INS.dir)

225 #
226 # create the links in /usr/platform/ foreach $(LINKED_PLATFORMS)
227 # to it's corresponding $(IMPLEMENTED_PLATFORM).
228 #
229 PLATFORMS = $(LINKED_PLATFORMS)

231 $(USR_PLAT_DIRS): $(USR_PLAT_DIR)
232     $(INS.slink3)

234 PLATFORMS += $(IMPLEMENTED_PLATFORM)

```

```
236 #
237 # Make the /platforms directories. This is hardwired here because
238 # the first stage of the project (KBI) only implements the userland
239 # changes, but the only reasonable place to record the aliases is
240 # here in kernel land.
241 #
242 install_platforms:      $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
243                        $(ROOT_PLAT_DIRS) $(USR_PLAT_DIRS) \
244                        $(USR_DESKTOP_DIR) $(USR_DESKTOP_INC_DIR) \
245                        $(USR_DESKTOP_SBIN_DIR) $(USR_DESKTOP_LIB_DIR)
246
247 #
248 # rules for making include, sbin, lib dirs/links in
249 # /usr/platform/$(PLATFORM)/ for desktop platforms
250 #
251 $(USR_DESKTOP_INC_DIR):      $(USR_DESKTOP_DIR)
252     $(INS.slink4)
253
254 $(USR_DESKTOP_SBIN_DIR):     $(USR_DESKTOP_DIR)
255     $(INS.slink5)
256
257 $(USR_DESKTOP_LIB_DIR):      $(USR_DESKTOP_DIR)
258     -$(INS.dir)
259
260 #
261 # Full kernel lint target.
262 #
263 LINT_TARGET      = globalint
264
265 globalint:
266     @pwd
267     @-$(ECHO) "\nSUN4U KERNEL: global crosschecks:"
268     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)
269
270 lint: lintlib .WAIT modlintlib .WAIT $(SPARC_LINTS) $(LINT_DEPS) \
271     $(IMPLEMENTATIONS) $(CPU_KMODS)
272     $(IMPLEMENTATIONS) $(CLOSED_IMPLEMENTATIONS) $(CPU_KMODS)
273
274 include ../Makefile.targ
275 #
276 # Cross-reference customization: build a cross-reference over all of the
277 # sun4u-related directories.
278 #
279 XRDIRS = ../sun4u ../sun4 ../sfmmu ../sparc ../sun ../common
280 SHARED_XRDIRS = ../sun4u ../sun4 ../sfmmu ../sparc ../sun ../common
281 CLOSED_XRDIRS = $(SHARED_XRDIRS:../%=../% ../../../../../closed/uts/%)
282 XRDIRS = $(SHARED_XRDIRS)
283 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS:../../../../closed/uts/sfmmu=)
284
285 XRPRUNE = i86pc
286
287 cscope.out tags: FRC
288     $(XREF) -x $@
```

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

1

```
*****
13345 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sun4u/Makefile.sun4u.shared
4027 remove CLOSED_BUILD
*****
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 #
26 # This makefile contains the common definitions for the sun4u unix
27 # and all sun4u implementation architecture dependent modules.
28 #
29 #
30 #
31 # Machine type (implementation architecture):
32 #
33 PLATFORM = sun4u
34 PROMIF = ieeel275
35 PSMBASE = $(UTSBASE)/../psm
36 #
37 #
38 # uname -m value
39 #
40 UNAME_M = $(PLATFORM)
41 #
42 #
43 # Definitions for the platform-specific /platform directories.
44 #
45 # PLATFORMS designates those sun4u machines which have no platform
46 # specific code.
47 #
48 # IMPLEMENTATIONS is used to designate sun4u machines which do have
49 # platform specific modules (perhaps including their own unix). All
50 # code specific to a given implementation resides in the appropriately
51 # named subdirectory. This requires these platforms to have their
52 # own Makefiles to define ROOT_PLAT_DIRS, USR_PLAT_DIRS, etc.
53 #
54 # So if we had an implementation named 'foo', we would need the following
55 # Makefiles in the foo subdirectory:
56 #
57 # sun4u/foo/Makefile
58 # sun4u/foo/Makefile.foo
59 # sun4u/foo/Makefile.targ
60 #
```

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

2

```
62 #
63 # /usr/platform/$(IMPLEMENTED_PLATFORM) is created as a directory that
64 # all the $(LINKED_PLATFORMS) link to.
65 #
66 IMPLEMENTED_PLATFORM = SUNW,Ultra-2
67 #
68 LINKED_PLATFORMS += SUNW,Ultra-30
69 LINKED_PLATFORMS += SUNW,Ultra-60
70 #
71 #
72 # all PLATFORMS that do not belong in the $(IMPLEMENTATIONS) list
73 # ie. all desktop platforms
74 #
75 PLATFORMS = $(IMPLEMENTED_PLATFORM)
76 PLATFORMS += $(LINKED_PLATFORMS)
77 #
78 ROOT_PLAT_DIRS = $(PLATFORMS:%=$(ROOT_PLAT_DIR)/%)
79 USR_PLAT_DIRS = $(PLATFORMS:%=$(USR_PLAT_DIR)/%)
80 #
81 USR_DESKTOP_DIR = $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM)
82 USR_DESKTOP_INC_DIR = $(USR_DESKTOP_DIR)/include
83 USR_DESKTOP_SBIN_DIR = $(USR_DESKTOP_DIR)/sbin
84 USR_DESKTOP_LIB_DIR = $(USR_DESKTOP_DIR)/lib
85 #
86 #
87 # Welcome to SPARC V9.
88 #
89 #
90 #
91 # Define supported builds
92 #
93 DEF_BUILDS = $(DEF_BUILDS64)
94 ALL_BUILDS = $(ALL_BUILDS64)
95 #
96 #
97 # Everybody needs to know how to build modstubs.o and to locate unix.o
98 #
99 UNIX_DIR = $(UTSBASE)/$(PLATFORM)/unix
100 GENLIB_DIR = $(UTSBASE)/$(PLATFORM)/genunix
101 MODSTUBS_DIR = $(UNIX_DIR)
102 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
103 LINTS_DIR = $(OBJS_DIR)
104 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJS_DIR)
105 #
106 DTRACESTUBS_O = $(OBJS_DIR)/dtracestubs.o
107 DTRACESTUBS = $(OBJS_DIR)/libdtracestubs.so
108 #
109 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
110 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
111 GENLIB = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
112 #
113 LINT_LIB = $(LINT_LIB_DIR)/llib-lunix.ln
114 GEN_LINT_LIB = $(LINT_LIB_DIR)/llib-lgenunix.ln
115 #
116 LINT64_DIRS = $(LINT64_BUILDS:%=$(UTSBASE)/$(PLATFORM)/lint-libs/%)
117 LINT64_FILES = $(LINT64_DIRS:%=%/llib-l$(MODULE).ln)
118 #
119 #
120 # cpu and platform modules need to know how to build their own symcheck mo
121 #
122 PLATMOD = platmod
123 PLATLIB = $(PLAT_DIR)/$(OBJS_DIR)/libplatmod.so
124 #
125 CPUNAME = cpu
126 CPULIB = $(CPU_DIR)/$(OBJS_DIR)/libcpu.so
```

```

128 SYM_MOD          = $(OBJS_DIR)/unix.sym

130 #
131 #   Include the makefiles which define build rule templates, the
132 #   collection of files per module, and a few specific flags. Note
133 #   that order is significant, just as with an include path. The
134 #   first build rule template which matches the files name will be
135 #   used. By including these in order from most machine dependent
136 #   to most machine independent, we allow a machine dependent file
137 #   to be used in preference over a machine independent version
138 #   (Such as a machine specific optimization, which preserves the
139 #   interfaces.)
140 #
141 include $(UTSBASE)/sun4/Makefile.files
142 include $(UTSTREE)/$(PLATFORM)/Makefile.files
143 include $(UTSBASE)/sfmmu/Makefile.files
144 include $(UTSBASE)/sparc/v9/Makefile.files
145 include $(UTSBASE)/sparc/Makefile.files
146 include $(UTSTREE)/sun/Makefile.files
147 include $(SRC)/psm/promif/$(PROMIF)/common/Makefile.files
148 include $(SRC)/psm/promif/$(PROMIF)/$(PLATFORM)/Makefile.files
149 include $(UTSTREE)/common/Makefile.files

151 #
152 #   Include machine independent rules. Note that this does not imply
153 #   that the resulting module from rules in Makefile.uts is machine
154 #   independent. Only that the build rules are machine independent.
155 #
156 include $(UTSBASE)/Makefile.uts

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

188 $(CLOSED_BUILD)CLOSED_IMPLEMENTATIONS = chalupa
189 $(CLOSED_BUILD)CLOSED_IMPLEMENTATIONS += ents

188 #
189 #   machine specific optimization, override default in Makefile.master

```

```

190 #
191 CC_XARCH          = -m64 -xarch=sparcvis
192 AS_XARCH          = -xarch=v9a
193 COPTIMIZE         = -xO3
194 CCMODE            = -Xa

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

218 ASFLAGS          += $(AS_XARCH)

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

222 LINT_KMODS       += $(GENUNIX_KMODS)

224 LINT_DEFS        = -m64

226 #
227 #   The following must be defined for all implementations:
228 #
229 #   MAPFILE:          ld mapfile for the build of kernel/unix.
230 #   MODSTUBS:        Module stubs source file.
231 #   GENCONST_SRC:    genconst.c
232 #   OFFSETS:         offsets.in
233 #   PLATFORM_OFFSETS: Platform specific mach_offsets.in
234 #   FDOFFSETS:       fd_offsets.in
235 #
236 MAPFILE          = $(UTSBASE)/sun4/conf/Mapfile
237 MODSTUBS         = $(UTSBASE)/sparc/ml/modstubs.s
238 GENCONST_SRC     = $(UTSBASE)/sun4/ml/genconst.c
239 OFFSETS          = $(UTSBASE)/sun4/ml/offsets.in
240 PLATFORM_OFFSETS = $(UTSBASE)/sun4u/ml/mach_offsets.in
241 FDOFFSETS        = $(UTSBASE)/sun4u/ml/fd_offsets.in

243 #
244 #   Define the actual specific platforms
245 #

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

249 #
250 #   Software workarounds for hardware "features"
251 #

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

255 #

```

```

256 #      Debugging level
257 #
258 #      Special knowledge of which special debugging options effect which
259 #      file is used to optimize the build if these flags are changed.
260 #
261 #      XXX: The above could possibly be done for more flags and files, but
262 #      is left as an experiment to the interested reader. Be forewarned,
263 #      that excessive use could lead to maintenance difficulties.
264 #
265 #      Note: kslice can be enabled for the sun4u, but is disabled by default
266 #      in all cases.
267 #

269 DEBUG_DEFS_OBJ64      =
270 DEBUG_DEFS_DBG64      = -DDEBUG
271 DEBUG_DEFS             = $(DEBUG_DEFS_$(BUILD_TYPE))

273 DEBUG_COND_OBJ64      :sh = echo \\043
274 DEBUG_COND_DBG64      =
275 IF_DEBUG_OBJ           = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

277 $(IF_DEBUG_OBJ)trap.o      :=      DEBUG_DEFS      += -DTRAPDEBUG
278 $(IF_DEBUG_OBJ)mach_trap.o :=      DEBUG_DEFS      += -DTRAPDEBUG
279 $(IF_DEBUG_OBJ)syscall_trap.o :=    DEBUG_DEFS      += -DSYSCALLTRACE
280 $(IF_DEBUG_OBJ)clock.o     :=      DEBUG_DEFS      += -DKSLICE=0

282 IF_TRAPTRACE_OBJ = $(IF_DEBUG_OBJ)
283 # comment this out for a non-debug kernel with TRAPTRACE
284 #IF_TRAPTRACE_OBJ = $(OBJS_DIR)/

286 $(IF_TRAPTRACE_OBJ)mach_locore.o      :=      DEBUG_DEFS      += -DTRAPTRACE
287 $(IF_TRAPTRACE_OBJ)m1setup.o         :=      DEBUG_DEFS      += -DTRAPTRACE
288 $(IF_TRAPTRACE_OBJ)syscall_trap.o    :=      DEBUG_DEFS      += -DTRAPTRACE
289 $(IF_TRAPTRACE_OBJ)startup.o         :=      DEBUG_DEFS      += -DTRAPTRACE
290 $(IF_TRAPTRACE_OBJ)mach_startup.o    :=      DEBUG_DEFS      += -DTRAPTRACE
291 $(IF_TRAPTRACE_OBJ)mp_startup.o      :=      DEBUG_DEFS      += -DTRAPTRACE
292 $(IF_TRAPTRACE_OBJ)cpu_states.o     :=      DEBUG_DEFS      += -DTRAPTRACE
293 $(IF_TRAPTRACE_OBJ)mach_cpu_states.o :=      DEBUG_DEFS      += -DTRAPTRACE
294 $(IF_TRAPTRACE_OBJ)interrupt.o      :=      DEBUG_DEFS      += -DTRAPTRACE
295 $(IF_TRAPTRACE_OBJ)mach_interrupt.o  :=      DEBUG_DEFS      += -DTRAPTRACE
296 $(IF_TRAPTRACE_OBJ)sfmmu_asm.o      :=      DEBUG_DEFS      += -DTRAPTRACE
297 $(IF_TRAPTRACE_OBJ)trap_table.o     :=      DEBUG_DEFS      += -DTRAPTRACE
298 $(IF_TRAPTRACE_OBJ)xc.o             :=      DEBUG_DEFS      += -DTRAPTRACE
299 $(IF_TRAPTRACE_OBJ)mach_xc.o         :=      DEBUG_DEFS      += -DTRAPTRACE
300 $(IF_TRAPTRACE_OBJ)wbuf.o           :=      DEBUG_DEFS      += -DTRAPTRACE
301 $(IF_TRAPTRACE_OBJ)trap.o           :=      DEBUG_DEFS      += -DTRAPTRACE
302 $(IF_TRAPTRACE_OBJ)mach_trap.o      :=      DEBUG_DEFS      += -DTRAPTRACE
303 $(IF_TRAPTRACE_OBJ)x_call.o         :=      DEBUG_DEFS      += -DTRAPTRACE
304 $(IF_TRAPTRACE_OBJ)spitfire_asm.o   :=      DEBUG_DEFS      += -DTRAPTRACE
305 $(IF_TRAPTRACE_OBJ)us3_common_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
306 $(IF_TRAPTRACE_OBJ)us3_cheetah_asm.o :=      DEBUG_DEFS      += -DTRAPTRACE
307 $(IF_TRAPTRACE_OBJ)us3_cheetahplus_asm.o :=    DEBUG_DEFS      += -DTRAPTRACE
308 $(IF_TRAPTRACE_OBJ)us3_jalapeno_asm.o :=    DEBUG_DEFS      += -DTRAPTRACE
309 $(IF_TRAPTRACE_OBJ)opl_olympus_asm.o :=    DEBUG_DEFS      += -DTRAPTRACE

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

313 #$(IF_DEBUG_OBJ)lock_prim.o      := DEBUG_DEFS      += -DDISP_LOCK_STATS
314 #$(IF_DEBUG_OBJ)disp.o          := DEBUG_DEFS      += -DDISP_LOCK_STATS

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

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

320 #
321 #      Collect the preprocessor definitions to be associated with *all*

```

```

322 #      files.
323 #
324 ALL_DEFS              = $(MACHINE_DEFS) $(WORKAROUND_DEFS) $(DEBUG_DEFS) \
325                        $(OPTION_DEFS)
326 GENCONST_DEFS        = $(MACHINE_DEFS) $(OPTION_DEFS)

328 #
329 # ----- TRANSITIONAL SECTION -----
330 #

332 #
333 #      Not everything which *should* be a module is a module yet. The
334 #      following is a list of such objects which are currently part of
335 #      the base kernel but should soon become kmods.
336 #
337 MACH_NOT_YET_KMODS    = $(AUTOCONF_OBJS)

339 #
340 # ----- END OF TRANSITIONAL SECTION -----
341 #

343 #
344 #      The kernels modules which are "implementation architecture"
345 #      specific for this machine are enumerated below. Note that most
346 #      of these modules must exist (in one form or another) for each
347 #      architecture.
348 #
349 #      Common Drivers (usually pseudo drivers) (/kernel/drv):
350 #

352 #
353 #      Machine Specific Driver Modules (/kernel/drv):
354 #
355 #      XXX: How many of these are really machine specific?
356 #
357 DRV_KMODS             += bbc_beep
358 DRV_KMODS             += cpc
359 DRV_KMODS             += fd
360 DRV_KMODS             += rootnex sbusmem upa64s zs zsh
361 DRV_KMODS             += sbus
362 DRV_KMODS             += pcisch pcipsy simba
363 DRV_KMODS             += px
364 DRV_KMODS             += ebus
365 DRV_KMODS             += su
366 DRV_KMODS             += tod
367 DRV_KMODS             += power
368 DRV_KMODS             += epic
369 DRV_KMODS             += grbeep
370 DRV_KMODS             += pcf8584 max1617 seeprom tda8444 pca9556
371 DRV_KMODS             += ics951601 adm1031
372 DRV_KMODS             += lm75 ltc1427 pcf8591 pcf8574 ssc050 ssc100
373 DRV_KMODS             += pic16f819
374 DRV_KMODS             += pic16f747
375 DRV_KMODS             += adm1026
376 DRV_KMODS             += us
377 DRV_KMODS             += ppm schppm jbusppm
378 DRV_KMODS             += mc-us3
379 DRV_KMODS             += mc-us3i
380 DRV_KMODS             += sbus
381 DRV_KMODS             += db21554
382 DRV_KMODS             += gpio_87317
383 DRV_KMODS             += isadma
384 DRV_KMODS             += sbbc
385 DRV_KMODS             += pmubus
386 DRV_KMODS             += pmugpio
387 DRV_KMODS             += pmc

```



```

388 DRV_KMODS      += trapstat
389 DRV_KMODS      += rmc_comm
390 DRV_KMODS      += rmcadm
391 DRV_KMODS      += rmclovm
392 DRV_KMODS      += sf
393 DRV_KMODS      += nxge
394 DRV_KMODS      += i2bsc
395 DRV_KMODS      += mem_cache

400 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ctsmc
401 $(CLOSED_BUILD)CLOSED_DRV_KMODS += m1535ppm
402 $(CLOSED_BUILD)CLOSED_DRV_KMODS += memtest
403 $(CLOSED_BUILD)CLOSED_DRV_KMODS += mi2cv
404 $(CLOSED_BUILD)CLOSED_DRV_KMODS += smbusr_ara

397 #
398 #      Exec Class Modules (/kernel/exec):
399 #
400 EXEC_KMODS      +=

402 #
403 #      Scheduling Class Modules (/kernel/sched):
404 #
405 SCHED_KMODS    +=

407 #
408 #      File System Modules (/kernel/fs):
409 #
410 FS_KMODS        +=

412 #
413 #      Streams Modules (/kernel/strmod):
414 #
415 STRMOD_KMODS   += kb

417 #
418 #      'System' Modules (/kernel/sys):
419 #
420 SYS_KMODS       +=

422 #
423 #      'User' Modules (/kernel/misc):
424 #
425 MISC_KMODS     += bignum
426 MISC_KMODS     += obpsym bootdev vis cpr platmod md5 sha1 i2c_svc
427 MISC_KMODS     += sbd

429 MISC_KMODS     += opl_cfg
430 MISC_KMODS     += zuluvm
431 MISC_KMODS     += gptwo_cpu gptwo_cfg
432 MISC_KMODS     += pcie

434 #
435 #      Brand modules
436 #
437 BRAND_KMODS    += snl_brand s10_brand

439 #
440 #      Software Cryptographic Providers (/kernel/crypto):
441 #
442 CRYPTO_KMODS   += aes
443 CRYPTO_KMODS   += arcfour
444 CRYPTO_KMODS   += des

446 #
447 #      generic-unix module (/kernel/genunix):

```

```

448 #
449 GENUNIX_KMODS  += genunix

451 #      'User' "Modules" excluded from the Full Kernel lint target:
452 #
453 $(CLOSED_BUILD)CLOSED_NLMISC_KMODS += forthdebug

454 #
455 #      Modules eXcluded from the product:
456 #
457 XMODS          +=

459 #
460 #      cpu modules
461 #
462 CPU_KMODS      += cheetah cheetahplus jalapeno serrano spitfire hummingbird

464 #
465 #      sun4u 'TOD' Modules (/platform/.../kernel/tod):
466 #
467 TOD_KMODS      += todods1287 todods1337 todmostek todstarfire
468 TOD_KMODS      += todm5819 todblade todhq4802 todsg todopl
469 TOD_KMODS      += todm5819p_rmc todstarcat

481 $(CLOSED_BUILD)CLOSED_TOD_KMODS += todm5823

471 #
472 #      Performance Counter BackEnd Modules (/usr/kernel/pcbe):
473 #
474 PCBE_KMODS     += us234_pcbe
475 PCBE_KMODS     += opl_pcbe

```

```

*****
3824 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sun4u/blade/Makefile.blade.shared
4027 remove CLOSED_BUILD
*****
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 #ident "%Z%M% %I% %E% SMI"
23 #
22 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Global definitions for sun4u jbos-blade implementation specific modules.
26 #
27 #
28 #
29 # Define directories.
30 #
31 ROOT_BLADE_DIR = $(ROOT_PLAT_DIR)/SUNW,Serverblade1
32 ROOT_BLADE_MOD_DIR = $(ROOT_BLADE_DIR)/kernel
33 ROOT_BLADE_MISC_DIR_32 = $(ROOT_BLADE_DIR)/kernel/misc
34 ROOT_BLADE_MISC_DIR_64 = $(ROOT_BLADE_MISC_DIR_32)/$(SUBDIR64)
35 ROOT_BLADE_KERN_DIR_32 = $(ROOT_BLADE_MOD_DIR)
36 ROOT_BLADE_KERN_DIR_64 = $(ROOT_BLADE_MOD_DIR)/$(SUBDIR64)
37 ROOT_BLADE_DRV_DIR_32 = $(ROOT_BLADE_MOD_DIR)/drv
38 ROOT_BLADE_DRV_DIR_64 = $(ROOT_BLADE_MOD_DIR)/drv/$(SUBDIR64)
39 #
40 ROOT_BLADE_KERN_DIR = $(ROOT_BLADE_KERN_DIR_$(CLASS))
41 ROOT_BLADE_DRV_DIR = $(ROOT_BLADE_DRV_DIR_$(CLASS))
42 ROOT_BLADE_MISC_DIR = $(ROOT_BLADE_MISC_DIR_$(CLASS))
43 #
44 ROOT_PLAT_MOD_DIRS += $(ROOT_BLADE_MOD_DIR)
45 ROOT_PLAT_MISC_DIRS += $(ROOT_BLADE_MISC_DIR)
46 #
47 #
48 USR_SUN4U_PLAT_DIR = $(USR_PLAT_DIR)/sun4u
49 USR_BLADE_DIR = $(USR_PLAT_DIR)/SUNW,Serverblade1
50 USR_BLADE_INC_DIR = $(USR_BLADE_DIR)/include
51 USR_BLADE_ISYS_DIR = $(USR_BLADE_INC_DIR)/sys
52 USR_BLADE_SBIN_DIR = $(USR_BLADE_DIR)/sbin
53 USR_BLADE_LIB_DIR = $(USR_BLADE_DIR)/lib
54 #
55 #
56 BLADE_LINT_LIB_DIR= $(UTSBASE)/$(PLATFORM)/blade/lint-libs/$(OBJJS_DIR)
57 #
58 #
59 # Define Objects

```

```

60 #
61 BLADE_OBJS = blade.o
62 #
63 #
64 # Option conf file
65 BLADE_OPTION = options
66 #
67 #
68 # Include the makefiles which define build rule templates, the
69 # collection of files per module, and a few specific flags. Note
70 # that order is significant, just as with an include path. The
71 # first build rule template which matches the files name will be
72 # used. By including these in order from most machine dependent
73 # to most machine independent, we allow a machine dependent file
74 # to be used in preference over a machine independent version
75 # (Such as a machine specific optimization, which preserves the
76 # interfaces.)
77 #
78 #
79 include $(UTSTREE)/sun4u/blade/Makefile.files
80 #
81 #
82 # Include common rules.
83 #
84 include $(UTSTREE)/sun4u/Makefile.sun4u
85 #
86 #
87 # Define modules (must come after Makefile.sun4u).
88 # Define modules (must come after Makefile.sun4u, for CLOSED_BUILD).
89 #
90 #
91 BLADE_KMODS = platmod
92 BLADE_KMODS += bscbus
93 BLADE_KMODS += bscv
94 #
95 #
96 LINTS_DIR = $(OBJJS_DIR)
97 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/blade/lint-libs/$(OBJJS_DIR)
98 LINT_LIB= $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJJS_DIR)/llib-lunix.ln
99 GEN_LINT_LIB= $(UTSBASE)/$(PLATFORM)/lint-libs/$(OBJJS_DIR)/llib-lgenunix.ln
100 #
101 #
102 #
103 #
104 #
105 # Define the actual specific platforms
106 #
107 #
108 MACHINE_DEFS += -D$(PLATFORM) -D_MACHDEP -DSFMMU
109 #
110 #
111 #
112 #
113 # Define for inline pre-processing since
114 # cpp not smart about v9 yet.
115 #
116 #
117 #CPP_DEFS_32 =
118 #CPP_DEFS_64 = -D_sparcv9
119 #CPP_DEFS = $(CPP_DEFS_$(CLASS))
120 #
121 #
122 #
123 # For now, disable these lint checks; maintainers should endeavor
124 # to investigate and remove these for maximum lint coverage.
125 # Please do not carry these forward to new Makefiles.
126 #
127 #
128 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
129 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
130 LINTTAGS += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
131 LINTTAGS += -erroff=E_STATIC_UNUSED
132 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
133 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV

```

new/usr/src/uts/sun4u/grover/Makefile.grover.shared

1

```
*****
2644 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sun4u/grover/Makefile.grover.shared
4027 remove CLOSED_BUILD
*****
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/sun4u/grover/Makefile.grover
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #
28 # Global definitions for sun4u implementation specific modules.
29 #
30 # Define directories.
31 #
32 ROOT_GROVER_DIR = $(ROOT_PLAT_DIR)/SUNW,Sun-Blade-100
33 ROOT_GROVER_MOD_DIR = $(ROOT_GROVER_DIR)/kernel
34 #
35 ROOT_GROVER_MISC_DIR_32 = $(ROOT_GROVER_MOD_DIR)/misc
36 ROOT_GROVER_MISC_DIR_64 = $(ROOT_GROVER_MISC_DIR_32)/$(SUBDIR64)
37 ROOT_GROVER_DRV_DIR_32 = $(ROOT_GROVER_MOD_DIR)/drv
38 ROOT_GROVER_DRV_DIR_64 = $(ROOT_GROVER_DRV_DIR_32)/$(SUBDIR64)
39 #
40 ROOT_GROVER_MISC_DIR = $(ROOT_GROVER_MISC_DIR_$(CLASS))
41 ROOT_GROVER_DRV_DIR = $(ROOT_GROVER_DRV_DIR_$(CLASS))
42 #
43 ROOT_PLAT_MOD_DIRS += $(ROOT_GROVER_MOD_DIR)
44 ROOT_PLAT_MISC_DIRS += $(ROOT_GROVER_MISC_DIR)
45 #
46 USR_GROVER_DIR = $(USR_PLAT_DIR)/SUNW,Sun-Blade-100
47 USR_GROVER_LINKED_DIR = $(USR_PLAT_DIR)/$(LINKED_PLATFORM)
48 USR_GROVER_INC_DIR = $(USR_GROVER_DIR)/include
49 USR_GROVER_ISYS_DIR = $(USR_GROVER_INC_DIR)/sys
50 USR_GROVER_SBIN_DIR = $(USR_GROVER_DIR)/sbin
51 USR_GROVER_LIB_DIR = $(USR_GROVER_DIR)/lib
52 #
53 #
54 GROVER_LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/grover/lint-libs/$(OBJS_DIR)
55 #
56 #
57 # Define objects.
58 #
59 GROVER_OBJS = grover.o
```

new/usr/src/uts/sun4u/grover/Makefile.grover.shared

2

```
61 include $(UTSTREE)/sun4u/grover/Makefile.files
62 #
63 #
64 # Include common rules.
65 #
66 include $(UTSTREE)/sun4u/Makefile.sun4u
67 #
68 #
69 # Define modules (must come after Makefile.sun4u).
70 #
71 GROVER_KMODS = platmod
72 GROVER_KMODS += grfans
73 #
74 #
75 # $(CLOSED_BUILD)CLOSED_GROVER_KMODS += grpmm
76 #
77 # For now, disable these lint checks; maintainers should endeavor
78 # to investigate and remove these for maximum lint coverage.
79 # Please do not carry these forward to new Makefiles.
80 #
81 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
82 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
83 LINTTAGS += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
84 LINTTAGS += -erroff=E_STATIC_UNUSED
85 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
86 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
```

new/usr/src/uts/sun4u/javelin/Makefile.javelin.shared

1

```
*****
2446 Thu Aug 15 11:59:26 2013
new/usr/src/uts/sun4u/javelin/Makefile.javelin.shared
4027 remove CLOSED_BUILD
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #
28 # This makefile contains the common definitions for the
29 # sun4u Javelin system dependent modules.
30 #
31 # Define directories
32 #
33 ROOT_JAVELIN_DIR = $(ROOT_PLAT_DIR)/SUNW,Ultra-250
34 ROOT_JAVELIN_MOD_DIR = $(ROOT_JAVELIN_DIR)/kernel
35 ROOT_JAVELIN_KERN_DIR_32 = $(ROOT_JAVELIN_MOD_DIR)
36 ROOT_JAVELIN_KERN_DIR_64 = $(ROOT_JAVELIN_MOD_DIR)/$(SUBDIR64)
37 ROOT_JAVELIN_DRV_DIR_32 = $(ROOT_JAVELIN_MOD_DIR)/drv
38 ROOT_JAVELIN_DRV_DIR_64 = $(ROOT_JAVELIN_MOD_DIR)/drv/$(SUBDIR64)
39 ROOT_JAVELIN_MISC_DIR_32 = $(ROOT_JAVELIN_MOD_DIR)/misc
40 ROOT_JAVELIN_MISC_DIR_64 = $(ROOT_JAVELIN_MOD_DIR)/misc/$(SUBDIR64)
41 #
42 ROOT_JAVELIN_KERN_DIR = $(ROOT_JAVELIN_KERN_DIR_$(CLASS))
43 ROOT_JAVELIN_MISC_DIR = $(ROOT_JAVELIN_MISC_DIR_$(CLASS))
44 ROOT_JAVELIN_DRV_DIR = $(ROOT_JAVELIN_DRV_DIR_$(CLASS))
45 #
46 ROOT_PLAT_MOD_DIRS += $(ROOT_JAVELIN_MOD_DIR)
47 ROOT_PLAT_MISC_DIRS += $(ROOT_JAVELIN_MISC_DIR)
48 ROOT_PLAT_MISC_DIRS_32 += $(ROOT_JAVELIN_MISC_DIR_32)
49 ROOT_PLAT_DRV_DIRS = $(ROOT_JAVELIN_DRV_DIR)
50 #
51 USR_JAVELIN_DIR = $(USR_PLAT_DIR)/SUNW,Ultra-250
52 USR_JAVELIN_INC_DIR = $(USR_JAVELIN_DIR)/include
53 USR_JAVELIN_SBIN_DIR = $(USR_JAVELIN_DIR)/sbin
54 USR_JAVELIN_LIB_DIR = $(USR_JAVELIN_DIR)/lib
55 USR_JAVELIN_ISYS_DIR = $(USR_JAVELIN_INC_DIR)/sys
56 #
57 JAVELIN_LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/javelin/lint-libs/$(OBSJ_DIR)
58 #
59 #
```

new/usr/src/uts/sun4u/javelin/Makefile.javelin.shared

2

```
60 # Define objects.
61 #
62 JAVELIN_OBJS = javelin.o
63 #
64 include $(UTSTREE)/sun4u/javelin/Makefile.files
65 #
66 #
67 # Include common rules.
68 #
69 include $(UTSTREE)/sun4u/Makefile.sun4u
70 #
71 #
72 # Define modules (must come after Makefile.sun4u)
73 #
74 # Define modules (must come after Makefile.sun4u, for CLOSED_BUILD)
75 #
76 JAVELIN_KMODS = platmod
77 JAVELIN_KMODS += envctrltwo
```

```

*****
3908 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4u/mpxu/Makefile.mpxu.shared
4027 remove CLOSED_BUILD
*****
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) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
23 #

25 #
26 #     Global definitions for sun4u implementation specific modules.
27 #

29 #
30 #     Define the name of this implementation.
31 #

33 #
34 #     Define directories.
35 #
36 ROOT_MPXU_DIR    = $(ROOT_PLAT_DIR)/SUNW,Sun-Fire-V240
37 ROOT_MPXU_MOD_DIR = $(ROOT_MPXU_DIR)/kernel

39 ROOT_MPXU_DRV_DIR_32 = $(ROOT_MPXU_MOD_DIR)/drv
40 ROOT_MPXU_DRV_DIR_64 = $(ROOT_MPXU_MOD_DIR)/drv/$(SUBDIR64)
41 ROOT_MPXU_DRV_DIR    = $(ROOT_MPXU_DRV_DIR_$(CLASS))

43 ROOT_MPXU_MISC_DIR_32 = $(ROOT_MPXU_MOD_DIR)/misc
44 ROOT_MPXU_MISC_DIR_64 = $(ROOT_MPXU_MOD_DIR)/misc/$(SUBDIR64)
45 ROOT_MPXU_MISC_DIR    = $(ROOT_MPXU_MISC_DIR_$(CLASS))

47 ROOT_MPXU_CRYPT0_DIR_32 = $(ROOT_MPXU_MOD_DIR)/crypto
48 ROOT_MPXU_CRYPT0_DIR_64 = $(ROOT_MPXU_MOD_DIR)/crypto/$(SUBDIR64)
49 ROOT_MPXU_CRYPT0_DIR    = $(ROOT_MPXU_CRYPT0_DIR_$(CLASS))

51 USR_MPXU_DIR      = $(USR_PLAT_DIR)/SUNW,Sun-Fire-V240
52 USR_MPXU_INC_DIR  = $(USR_MPXU_DIR)/include
53 USR_MPXU_ISYS_DIR = $(USR_MPXU_INC_DIR)/sys
54 USR_MPXU_SBIN_DIR = $(USR_MPXU_DIR)/sbin
55 USR_MPXU_SBIN_PRTDIAG = $(USR_MPXU_SBIN_DIR)/prtdiag
56 USR_MPXU_SBIN_FRUADM = $(USR_MPXU_SBIN_DIR)/fruadm
57 USR_MPXU_LIB_DIR  = $(USR_MPXU_DIR)/lib

59 MPXU_LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/mpxu/lint-libs/$(OBJS_DIR)

61 #

```

```

62 # Links to UltraSparc III crypto modules
63 #
64 MPXU_CRYPT0_LINKS    = aes

66 #
67 #     Include the makefiles which define build rule templates, the
68 #     collection of files per module, and a few specific flags. Note
69 #     that order is significant, just as with an include path. The
70 #     first build rule template which matches the files name will be
71 #     used. By including these in order from most machine dependent
72 #     to most machine independent, we allow a machine dependent file
73 #     to be used in preference over a machine independent version
74 #     (Such as a machine specific optimization, which preserves the
75 #     interfaces.)
76 #

78 include $(UTSTREE)/sun4u/mpxu/Makefile.files
79 #
80 #     Include common rules.
81 #

83 include $(UTSTREE)/sun4u/Makefile.sun4u

85 #
86 #     Define modules (must come after Makefile.sun4u).
87 #
88 #     Define modules (must come after Makefile.sun4u, for CLOSED_BUILD).
89 #
90 MODSTUBS_DIR = $(UNIX_DIR)
91 LINTS_DIR    = $(OBJS_DIR)
92 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/mpxu/lint-libs/$(OBJS_DIR)

94 #
95 #     Define the actual specific platforms
96 #
97 MACHINE_DEFS = -D$(PLATFORM) -D_MACHDEP -DSFMMU -DMP

99 #
100 #     Define platform specific values
101 #
102 #MACHINE_DEFS += -DNCPU=554
103 #MACHINE_DEFS += -DMAX_UPA=1024
104 #MACHINE_DEFS += -DIGN_SIZE=10
105 # Max IOSRAM TOC major version number supported
106 #MACHINE_DEFS += -DMAX_IOSRAM_TOC_VER=0x1

108 #
109 #     Define for inline pre-processing since
110 #     cpp not smart about v9 yet.
111 #
111 CPP_DEFS_32 =
112 CPP_DEFS_64 = -D__sparcv9
113 CPP_DEFS    = $(CPP_DEFS_$(CLASS))

115 #
116 # For now, disable these lint checks; maintainers should endeavor
117 # to investigate and remove these for maximum lint coverage.
118 # Please do not carry these forward to new Makefiles.
119 #
120 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
121 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
122 LINTTAGS += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
123 LINTTAGS += -erroff=E_STATIC_UNUSED
124 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
125 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV

```

new/usr/src/uts/sun4u/opl/Makefile.opl.shared

1

6094 Thu Aug 15 11:59:27 2013

new/usr/src/uts/sun4u/opl/Makefile.opl.shared

4027 remove CLOSED_BUILD

```
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
23 #
24 # Global definitions for sun4u opl implementation specific modules.
25 #
26 # uts/sun4u/opl/Makefile.opl
27 #
28 #
29 #
30 # Define directories.
31 #
32 #
33 #
34 ROOT_OPL_DIR = $(ROOT_PLAT_DIR)/SUNW,SPARC-Enterprise
35 ROOT_OPL_MOD_DIR = $(ROOT_OPL_DIR)/kernel
36 ROOT_OPL_KERN_DIR_32 = $(ROOT_OPL_MOD_DIR)
37 ROOT_OPL_KERN_DIR_64 = $(ROOT_OPL_MOD_DIR)/$(SUBDIR64)
38 ROOT_OPL_MISC_DIR_32 = $(ROOT_OPL_MOD_DIR)/misc
39 ROOT_OPL_MISC_DIR_64 = $(ROOT_OPL_MOD_DIR)/misc/$(SUBDIR64)
40 ROOT_OPL_DRV_DIR_32 = $(ROOT_OPL_MOD_DIR)/drv
41 ROOT_OPL_DRV_DIR_64 = $(ROOT_OPL_MOD_DIR)/drv/$(SUBDIR64)
42 ROOT_OPL_CPU_DIR_32 = $(ROOT_OPL_MOD_DIR)/cpu
43 ROOT_OPL_CPU_DIR_64 = $(ROOT_OPL_MOD_DIR)/cpu/$(SUBDIR64)
44 ROOT_OPL_CRYPTODIR_32 = $(ROOT_OPL_MOD_DIR)/crypto
45 ROOT_OPL_CRYPTODIR_64 = $(ROOT_OPL_MOD_DIR)/crypto/$(SUBDIR64)
46 #
47 ROOT_OPL_KERN_DIR = $(ROOT_OPL_KERN_DIR_$(CLASS))
48 ROOT_OPL_MISC_DIR = $(ROOT_OPL_MISC_DIR_$(CLASS))
49 ROOT_OPL_DRV_DIR = $(ROOT_OPL_DRV_DIR_$(CLASS))
50 ROOT_OPL_CPU_DIR = $(ROOT_OPL_CPU_DIR_$(CLASS))
51 ROOT_OPL_CRYPTODIR = $(ROOT_OPL_CRYPTODIR_$(CLASS))
52 #
53 ROOT_PLAT_MOD_DIRS += $(ROOT_OPL_MOD_DIR)
54 ROOT_PLAT_MISC_DIRS_32 += $(ROOT_OPL_MISC_DIR_32)
55 #
56 USR_OPL_DIR = $(USR_PLAT_DIR)/SUNW,SPARC-Enterprise
57 USR_OPL_LIB_DIR = $(USR_OPL_DIR)/lib
58 USR_OPL_SBIN_DIR = $(USR_OPL_DIR)/sbin
59 USR_OPL_SBIN_PRTDIAG = $(USR_OPL_SBIN_DIR)/prtdiag
60 USR_OPL_SBIN_FRUADM = $(USR_OPL_SBIN_DIR)/fruadm
61 USR_OPL_INC_DIR = $(USR_OPL_DIR)/include
```

new/usr/src/uts/sun4u/opl/Makefile.opl.shared

2

```
62 USR_OPL_ISYS_DIR = $(USR_OPL_INC_DIR)/sys
63 #
64 OPL_LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/opl/lint-libs/$(OBJS_DIR)
65 OPLMSU_OPTION = options
66 #
67 #
68 # Define modules.
69 #
70 OPL_KMODS = platmod
71 OPL_KMODS += dm2s
72 OPL_KMODS += oplkmdrv
73 OPL_KMODS += pcicmu
74 OPL_KMODS += oplpanel
75 OPL_KMODS += dir.WAIT drmach
76 OPL_KMODS += oplmsu
77 OPL_KMODS += mc-opl
78 #
79 #
80 # CPU modules.
81 #
82 OPL_CPU_KMODS += olympus_c
83 #
84 # Links to OPL crypto modules
85 #
86 OPL_CRYPTOLINKS = aes
87 #
88 #
89 # Include the makefiles which define build rule templates, the
90 # collection of files per module, and a few specific flags. Note
91 # that order is significant, just as with an include path. The
92 # first build rule template which matches the files name will be
93 # used. By including these in order from most machine dependent
94 # to most machine independent, we allow a machine dependent file
95 # to be used in preference over a machine independent version
96 # (Such as a machine specific optimization, which preserves the
97 # interfaces.)
98 #
99 include $(UTSBASE)/sun4u/ngdr/Makefile.files
100 include $(UTSTREE)/sun4u/opl/Makefile.files
101 #
102 #
103 # Include common rules.
104 #
105 include $(UTSTREE)/sun4u/Makefile.sun4u
106 #
107 #
108 # Define closed modules (must come after Makefile.sun4u for CLOSED_BUILD).
109 #
110 $(CLOSED_BUILD)CLOSED_OPL_KMODS = scfd
111 #
112 #
113 # Everybody needs to know how to build modstubs.o and to locate unix.o
114 #
115 #
116 UNIX_DIR = $(UTSBASE)/$(PLATFORM)/opl/unix
117 MODSTUBS_DIR = $(UNIX_DIR)
118 DSF_DIR = $(UTSBASE)/$(PLATFORM)/opl/genassym
119 LINTS_DIR = $(OBJS_DIR)
120 LINT_LIB_DIR = $(UTSBASE)/$(PLATFORM)/opl/lint-libs/$(OBJS_DIR)
121 #
122 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
123 #
124 LINT_LIB = $(LINT_LIB_DIR)/llib-lunix.ln
125 #
126 #
127 # Define the actual specific platforms
128 #
```

```
123 MACHINE_DEFS      = -D$(PLATFORM) -D_MACHDEP -DSFMMU -DMP
124 MACHINE_DEFS      += -D_CPU_SIGNATURE

126 #
127 # Maximum CUID = 01111 11 01 1 = 0x1FB (507)
128 # Maximum CHIPID = 1 01111 11 00 0 = 0x5F8 (1528)
129 #
130 MACHINE_DEFS      += -DNPCU=512
131 MACHINE_DEFS      += -DMAX_CPU_CHIPID=1529
132 MACHINE_DEFS      += -DMAX_UPA=256
133 MACHINE_DEFS      += -DIGN_SIZE=8
134 MACHINE_DEFS      += -DMAX_MEM_NODES=16
135 #
136 # UTSB_PHYS will enable user TSB physical access for TL>0
137 #
138 MACHINE_DEFS      += -DUTSB_PHYS
139 MACHINE_DEFS      += -D_OPL
140 MACHINE_DEFS      += -DOLYMPUS_SHARED_FTLB
141 MACHINE_DEFS      += -D_CMP_NO_ERROR_STEERING -D_HW_MEMSCRUB_SUPPORT
142 MACHINE_DEFS      += -DDO_CORELEVEL_LOADBAL
143 MACHINE_DEFS      += -DITLB_32M_256M_SUPPORT
144 #
145 # OLYMPUS C cross-call erratas.
146 # - revision A can only deliver one xcall at a time.
147 # - revision B can dispatch xcalls to 31 (IDSR_BN_SETS) CPUs at a time,
148 #   but it can not send more xcalls until all the pending xcalls are
149 #   dispatched. In other words, all previous 31 xcall slots must be
150 #   in non-busy state before further xcalls can be issued.
151 MACHINE_DEFS      += -DOLYMPUS_C_REV_A_ERRATA_XCALL
152 MACHINE_DEFS      += -DOLYMPUS_C_REV_B_ERRATA_XCALL
153 #
154 # OLYMPUS C Spurious interrupts
155 # - When an UE is detected in a interrupt packet,
156 #   Olympus-C takes an interrupt_vector_trap (TT=0x60) while
157 #   ASI_INTR_RECIEVE.BUSY is set to zero to indicate the existence
158 #   of the error. Software will see this as a spurious interrupt since
159 #   the interrupt busy bit is set to zero. SW will still need to
160 #   explicitly clear the interrupt busy bit to reset the HW state.
161 #   Failure to do so will result in the processor continuously taking
162 #   an interrupt vector trap when PSTATE.IE is reset to one.
163 #   Note that UE in interrupt packet is reported to the SP and handled
164 #   accordingly. For the domain, the system should panic as it is not
165 #   recoverable.
166 MACHINE_DEFS      += -DCLEAR_INTR_BUSYBIT_ON_SPURIOUS

168 #
169 # For now, disable these lint checks; maintainers should endeavor
170 # to investigate and remove these for maximum lint coverage.
171 # Please do not carry these forward to new Makefiles.
172 #
173 LINTTAGS           += -erroff=E_SUSPICIOUS_COMPARISON
174 LINTTAGS           += -erroff=E_BAD_PTR_CAST_ALIGN
175 LINTTAGS           += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
176 LINTTAGS           += -erroff=E_STATIC_UNUSED
177 LINTTAGS           += -erroff=E_PTRDIFF_OVERFLOW
178 LINTTAGS           += -erroff=E_ASSIGN_NARROW_CONV

180 .KEEP_STATE:
```

new/usr/src/uts/sun4u/sunfire/Makefile.sunfire.shared

1

```
*****
2602 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4u/sunfire/Makefile.sunfire.shared
4027 remove CLOSED_BUILD
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #
28 # Global definitions for sun4u implementation specific modules.
29 #
30 # Define the name of this implementation.
31 #
32 #
33 # Define directories.
34 #
35 #
36 ROOT_SUNFIRE_DIR = $(ROOT_PLAT_DIR)/SUNW,Ultra-Enterprise
37 ROOT_SUNFIRE_MOD_DIR = $(ROOT_SUNFIRE_DIR)/kernel
38 #
39 ROOT_SUNFIRE_DRV_DIR_32 = $(ROOT_SUNFIRE_MOD_DIR)/drv
40 ROOT_SUNFIRE_DRV_DIR_64 = $(ROOT_SUNFIRE_MOD_DIR)/drv/$(SUBDIR64)
41 ROOT_SUNFIRE_DRV_DIR = $(ROOT_SUNFIRE_DRV_DIR_$(CLASS))
42 ROOT_PLAT_DRV_DIRS = $(ROOT_SUNFIRE_DRV_DIR)
43 #
44 ROOT_SUNFIRE_MISC_DIR_32 = $(ROOT_SUNFIRE_MOD_DIR)/misc
45 ROOT_SUNFIRE_MISC_DIR_64 = $(ROOT_SUNFIRE_MOD_DIR)/misc/$(SUBDIR64)
46 ROOT_SUNFIRE_MISC_DIR = $(ROOT_SUNFIRE_MISC_DIR_$(CLASS))
47 #
48 USR_SUNFIRE_DIR = $(USR_PLAT_DIR)/SUNW,Ultra-Enterprise
49 USR_SUNFIRE_INC_DIR = $(USR_SUNFIRE_DIR)/include
50 USR_SUNFIRE_ISYS_DIR = $(USR_SUNFIRE_INC_DIR)/sys
51 USR_SUNFIRE_SBIN_DIR = $(USR_SUNFIRE_DIR)/sbin
52 USR_SUNFIRE_LIB_DIR = $(USR_SUNFIRE_DIR)/lib
53 #
54 SUNFIRE_LINT_LIB_DIR= $(UTSBASE)/$(PLATFORM)/sunfire/lint-libs/$(OBSJ_DIR)
55 #
56 #
57 # Define objects.
58 #
59 SUNFIRE_OBJS = sunfire.o
```

new/usr/src/uts/sun4u/sunfire/Makefile.sunfire.shared

2

```
61 include $(UTSTREE)/sun4u/sunfire/Makefile.files
62 #
63 # Include common rules.
64 #
65 #
66 include $(UTSTREE)/sun4u/Makefile.sun4u
67 #
68 #
69 # Define modules (must come after Makefile.sun4u).
70 #
71 # Define modules (must come after Makefile.sun4u for CLOSED_BUILD).
72 #
73 #
74 SUNFIRE_KMODS = ac central environ fhc simmstat sysctrl sram
75 $(CLOSED_BUILD)CLOSED_SUNFIRE_KMODS = platmod
76 #
77 #
78 # For now, disable these lint checks; maintainers should endeavor
79 # to investigate and remove these for maximum lint coverage.
80 # Please do not carry these forward to new Makefiles.
81 #
82 #
83 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
84 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
85 LINTTAGS += -erroff=E_SUPPRESSION_DIRECTIVE_UNUSED
86 LINTTAGS += -erroff=E_STATIC_UNUSED
87 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
88 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
```


new/usr/src/uts/sun4u/sys/Makefile

1

```
*****
3705 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4u/sys/Makefile
4027 remove CLOSED_BUILD
*****
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 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
22 # Use is subject to license terms.
23 #
24 # uts/sun4u/sys/Makefile
25 #
26 UTBASE = ../..

28 #
29 # include global definitions
30 #
31 include ../Makefile.sun4u

33 #
34 # Override defaults.
35 #
36 FILEMODE = 644

38 SUN4_HDRS= \
39     async.h \
40     clock.h \
41     cmp.h \
42     cpc_ultra.h \
43     cpu_sgnblk_defs.h \
44     ddi_subrdefs.h \
45     dvma.h \
46     eeprom.h \
47     errclassify.h \
48     fcode.h \
49     fc_plat.h \
50     idprom.h \
51     intr.h \
52     intreg.h \
53     ivintr.h \
54     memlist_plat.h \
55     memnode.h \
56     nexusdebug.h \
57     prom_debug.h \
58     scb.h \
59     sun4asi.h \
60     tod.h \
61     trapstat.h \
```

new/usr/src/uts/sun4u/sys/Makefile

2

```
62     vis.h \
63     vm_machparam.h \
64     x_call.h \
65     xc_impl.h \
66     zsmach.h

68 $(CLOSED_BUILD)CLOSED_SUN4_HDRS= \
69     memtestio.h

68 HDRS= \
69     cheetahregs.h \
70     cpr_impl.h \
71     cpu_impl.h \
72     ecc_kstat.h \
73     envctrl.h \
74     envctrl_gen.h \
75     envctrl_ue250.h \
76     envctrl_ue450.h \
77     gpio_87317.h \
78     iocache.h \
79     iommu.h \
80     machasi.h \
81     machclock.h \
82     machcpuvar.h \
83     machparam.h \
84     machsystem.h \
85     machthread.h \
86     mem_cache.h \
87     mmu.h \
88     opl_module.h \
89     prom_plat.h \
90     pte.h \
91     sbd_ioctl.h \
92     spitregs.h \
93     starfire.h \
94     sysioerr.h \
95     sysiosbus.h \
96     todmostek.h \
97     traptrace.h

102 $(CLOSED_BUILD)CLOSED_HDRS= \
103     memtestio_ch.h \
104     memtestio_chp.h \
105     memtestio_ja.h \
106     memtestio_jg.h \
107     memtestio_oc.h \
108     memtestio_pn.h \
109     memtestio_sf.h \
110     memtestio_sr.h \
111     memtestio_u.h

99 I2CHDRS = \
100     clients/max1617.h misc/i2c_svc.h clients/i2c_client.h \
101     clients/hpc3130.h clients/lm75.h \
102     clients/pcf8591.h clients/ssc050.h
115 clients/pcf8591.h clients/ssc050.h $(CLOSED_I2CHDRS)

103 I2C_DIRS= \
104     clients misc
105 USR_PSM_ISYS_I2C_ROOT= $(USR_PSM_ISYS_DIR)/i2c
106 USR_PSM_ISYS_I2C_DIRS= $(USR_PSM_ISYS_I2C_ROOT) \
107     $(I2C_DIRS:%=$(USR_PSM_ISYS_I2C_ROOT)/%)

108 ROOTI2CHDRS= \
109     $(I2CHDRS:%=$(USR_PSM_ISYS_I2C_ROOT)/%)

110 MONHDRS= \
111 #MONHDRS= \
112     eeprom.h idprom.h keyboard.h password.h
```

new/usr/src/uts/sun4u/sys/Makefile

3

```

113 USR_PSM_MON_DIR=      $(USR_PSM_ISYS_DIR)/mon
115 ROOTHDRS=             $(HDRS:%=$(USR_PSM_ISYS_DIR)/%)
130 $(CLOSED_BUILD)ROOTHDRS += $(CLOSED_HDRS:%=$(USR_PSM_ISYS_DIR)/%)
117 SUN4_ROOTHDRS=       $(SUN4_HDRS:%=$(USR_PSM_ISYS_DIR)/%)
133 $(CLOSED_BUILD)SUN4_ROOTHDRS += $(CLOSED_SUN4_HDRS:%=$(USR_PSM_ISYS_DIR)/%)

119 ROOTMONHDRS=         $(MONHDRS:%=$(USR_PSM_MON_DIR)/%)

121 ROOTDIR=              $(ROOT)/usr/share/src
122 ROOTDIRS=            $(ROOTDIR)/uts $(ROOTDIR)/uts/$(PLATFORM)

124 ROOTLINK=            $(ROOTDIR)/uts/$(PLATFORM)/sys
125 LINKDEST=            ../../../../platform/$(PLATFORM)/include/sys

127 CHECKHDRS=           $(HDRS:%.h=%.check) \
128 $(MONHDRS:%.h=mon/%.check) \
129 $(SUN4_HDRS:%.h=%.cmncheck)

147 $(CLOSED_BUILD)CHECKHDRS += $(CLOSED_HDRS:%.h=$(CLOSED)/uts/sun4u/sys/%.check)
148 $(CLOSED_BUILD)CHECKHDRS += \
149 $(CLOSED_I2CHDRS:%.h=$(CLOSED)/uts/sun4u/sys/i2c/%.check)

131 .KEEP_STATE:

133 .PARALLEL: $(CHECKHDRS) $(ROOTHDRS) $(ROOTMONHDRS) $(SUN4_ROOTHDRS)

135 install_h: $(ROOTDIRS) $(USR_PSM_ISYS_I2C_DIRS) .WAIT \
136 $(ROOTHDRS) $(ROOTI2CHDRS) \
137 $(ROOTMONHDRS) \
138 $(SUN4_ROOTHDRS) $(ROOTLINK)

140 check: $(CHECKHDRS)

142 #
143 # install rules
144 #
145 $(USR_PSM_MON_DIR):      $(USR_PSM_ISYS_DIR)
146 $(INS.dir)

148 $(USR_PSM_ISYS_I2C_DIRS):
149 $(INS.dir)

171 $(USR_PSM_ISYS_DIR)/%:  $(CLOSED)/uts/sun4u/sys/% $(USR_PSM_ISYS_DIR)
172 $(INS.file)

174 $(USR_PSM_ISYS_DIR)/%:  $(CLOSED)/uts/sun4/sys/% $(USR_PSM_ISYS_DIR)
175 $(INS.file)

151 $(USR_PSM_ISYS_DIR)/%:  ../../sfmmu/sys/% $(USR_PSM_ISYS_DIR)
152 $(INS.file)

154 $(USR_PSM_ISYS_DIR)/%:  ../../sun4/sys/% $(USR_PSM_ISYS_DIR)
155 $(INS.file)

157 $(USR_PSM_MON_DIR)/%:  mon/% $(USR_PSM_MON_DIR)
158 $(INS.file)

160 $(ROOTDIRS):
161 $(INS.dir)

163 # -r because this used to be a directory and is now a link.
164 $(ROOTLINK): $(ROOTDIRS)
165 -$(RM) -r $@; $(SYMLINK) $(LINKDEST) $@

```

new/usr/src/uts/sun4u/sys/Makefile

4

```

167 mon/%.check:      mon/%.h
168 $(DOT_H_CHECK)

170 %.check:          ../../sfmmu/sys/%.h
171 $(DOT_H_CHECK)
172 %.cmncheck:       ../../sun4/sys/%.h
173 $(DOT_H_CHECK)

175 FRC:

177 include ../../Makefile.targ

```

new/usr/src/uts/sun4u/tazmo/Makefile.tazmo.shared

1

```
*****
2766 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4u/tazmo/Makefile.tazmo.shared
4027 remove CLOSED_BUILD
*****
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/sun4u/tazmo/Makefile.impl
23 # Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24 # Use is subject to license terms.
25 #
26 #ident "%Z%M% %I% %E% SMI"
27 #
28 # Global definitions for sun4u implementation specific modules.
29 #
30 # Define directories.
31 #
32 ROOT_TAZMO_DIR = $(ROOT_PLAT_DIR)/SUNW,Ultra-4
33 ROOT_TAZMO_MOD_DIR = $(ROOT_TAZMO_DIR)/kernel
34 #
35 ROOT_TAZMO_KERN_DIR_32 = $(ROOT_TAZMO_MOD_DIR)
36 ROOT_TAZMO_KERN_DIR_64 = $(ROOT_TAZMO_MOD_DIR)/$(SUBDIR64)
37 ROOT_TAZMO_DRV_DIR_32 = $(ROOT_TAZMO_MOD_DIR)/drv
38 ROOT_TAZMO_DRV_DIR_64 = $(ROOT_TAZMO_MOD_DIR)/drv/$(SUBDIR64)
39 ROOT_TAZMO_MISC_DIR_32 = $(ROOT_TAZMO_MOD_DIR)/misc
40 ROOT_TAZMO_MISC_DIR_64 = $(ROOT_TAZMO_MOD_DIR)/misc/$(SUBDIR64)
41 #
42 ROOT_TAZMO_KERN_DIR = $(ROOT_TAZMO_KERN_DIR_$(CLASS))
43 ROOT_TAZMO_MISC_DIR = $(ROOT_TAZMO_MISC_DIR_$(CLASS))
44 ROOT_TAZMO_DRV_DIR = $(ROOT_TAZMO_DRV_DIR_$(CLASS))
45 #
46 ROOT_PLAT_MOD_DIRS += $(ROOT_TAZMO_MOD_DIR)
47 ROOT_PLAT_MISC_DIRS += $(ROOT_TAZMO_MISC_DIR)
48 ROOT_PLAT_MISC_DIRS_32 += $(ROOT_TAZMO_MISC_DIR_32)
49 ROOT_PLAT_DRV_DIRS = $(ROOT_TAZMO_DRV_DIR)
50 #
51 USR_TAZMO_DIR = $(USR_PLAT_DIR)/SUNW,Ultra-4
52 USR_TAZMO_LINKED_DIR = $(USR_PLAT_DIR)/$(LINKED_PLATFORM)
53 USR_TAZMO_INC_DIR = $(USR_TAZMO_DIR)/include
54 USR_TAZMO_ISYS_DIR = $(USR_TAZMO_INC_DIR)/sys
55 USR_TAZMO_SBIN_DIR = $(USR_TAZMO_DIR)/sbin
56 USR_TAZMO_LIB_DIR = $(USR_TAZMO_DIR)/lib
57 #
58 TAZMO_LINT_LIB_DIR= $(UTSBASE)/$(PLATFORM)/tazmo/lint-libs/$(OBJDIR)
```

new/usr/src/uts/sun4u/tazmo/Makefile.tazmo.shared

2

```
60 #
61 # Define objects.
62 #
63 TAZMO_OBJS = tazmo.o
64 #
65 include $(UTSBASE)/sun4u/tazmo/Makefile.files
66 #
67 #
68 # Include common rules.
69 #
70 include $(UTSTREE)/sun4u/Makefile.sun4u
71 #
72 #
73 # Define modules (must come after Makefile.sun4u).
74 # Define modules (must come after Makefile.sun4u, for CLOSED_BUILD).
75 #
76 TAZMO_KMODS = platmod
77 TAZMO_KMODS += envctrl
78 #
79 # For now, disable these lint checks; maintainers should endeavor
80 # to investigate and remove these for maximum lint coverage.
81 # Please do not carry these forward to new Makefiles.
82 #
83 LINTTAGS += -erroff=E_PTRDIFF_OVERFLOW
84 LINTTAGS += -erroff=E_BAD_PTR_CAST_ALIGN
85 LINTTAGS += -erroff=E_ASSIGN_NARROW_CONV
86 LINTTAGS += -erroff=E_SUSPICIOUS_COMPARISON
```

new/usr/src/uts/sun4v/Makefile

1

```
*****
7946 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4v/Makefile
4027 remove CLOSED_BUILD
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # This makefile drives the production of all implementation architecture
26 # dependent modules for the sun4v architecture.
27 #

29 UTSBASE = ..

31 include Makefile.sun4v
32 include Makefile.stpaul
33 include Makefile.huron
34 include Makefile.maramba
35 include Makefile.thunder
36 include Makefile.turgo
37 include Makefile.congo
38 include Makefile.monza

40 USR_GLENDALE_DIR = $(USR_PLAT_DIR)/SUNW,Sun-Blade-T6320
41 USR_GLENDALE_SBIN_DIR = $(USR_GLENDALE_DIR)/sbin
42 USR_GLENDALE_LIB_DIR = $(USR_GLENDALE_DIR)/lib

45 #
46 # The following are SPARC specific (rather than sun4v) specific modules
47 # which are required for the sun4v kernel to completely lint. They are
48 # not involved in the build in any other way. In order to minimize
49 # build time, it is assumed that they are up to date. But since sun4v
50 # is really a separate architecture we cannot use the v7 sparc modules.
51 #
52 SPARC_LIB_DIR = $(UTSBASE)/sparc/lint-libs/$(OBSJ_DIR)

54 SPARC_LINTS =

56 #
57 #
58 #
59 LINT_LIBS = $(LINT_LIB) \
60 $(LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
61 $(CLOSED_LINT_KMODS:%=$(LINT_LIB_DIR)/llib-1%.ln) \
```

new/usr/src/uts/sun4v/Makefile

2

```
62 $(SPARC_LINTS:%=$(SPARC_LIB_DIR)/llib-1%.ln)

64 def := TARGET= def
65 all := TARGET= all
66 install := TARGET= install
67 install_h := TARGET= install_h
68 clean := TARGET= clean
69 clobber := TARGET= clobber
70 lint := TARGET= lint
71 lintlib := TARGET= lintlib
72 modlintlib := TARGET= modlintlib
73 modlist := TARGET= modlist
74 modlist modlist.sparc := NO_STATE= -K $$MODSTATE$$$$
75 clean.lint := TARGET= clean.lint
76 check := TARGET= check

78 .KEEP_STATE:

80 .PARALLEL: $(PARALLEL_KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
81 modlist modlist.sparc

83 # Override for CPU_KMODS... they cannot be built
84 # in parallel
85 .NO_PARALLEL: $(CPU_KMODS)

87 def all clean clobber clean.lint: genassym unix .WAIT \
88 $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)

90 # list the modules under sun4v.
91 modlist: unix $(KMODS) $(CLOSED_KMODS) $(XMODS) $(CLOSED_XMODS) \
92 $(IMPLEMENTATIONS)

94 # list the modules for Install -k sun4v.
95 modlist.karch: modlist modlist.sparc

97 modlist.sparc:
98 @cd $(SRC)/uts/sparc; pwd; $(MAKE) $(NO_STATE) modlist

100 install: install_platforms genassym unix .WAIT $(KMODS) $(CLOSED_KMODS) \
101 $(XMODS) $(CLOSED_XMODS) $(IMPLEMENTATIONS)

103 lintlib: unix

105 modlintlib: $(LINT_KMODS) $(CLOSED_LINT_KMODS)

107 genassym unix $(KMODS): FRC
108 @cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

110 $(IMPLEMENTATIONS): FRC
111 @cd $@; pwd; THISIMPL=$@ $(MAKE) $(NO_STATE) $(TARGET)

113 $(XMODS): FRC
114 @if [ -f $@/Makefile ]; then \
115 cd $@; pwd; $(MAKE) $(NO_STATE) $(TARGET); \
116 else \
117 true; \
118 fi

120 $(CLOSED_XMODS): FRC
121 @if [ -f $(CLOSED)/uts/sun4v/$@/Makefile ]; then \
122 cd $(CLOSED)/uts/sun4v/$@; pwd; \
123 $(MAKE) $(NO_STATE) $(TARGET); \
124 else \
125 true; \
126 fi
```

new/usr/src/uts/sun4v/Makefile

3

```

128 $(CLOSED_KMODS):          FRC
129     cd $(CLOSED)/uts/sun4v/$@; pwd; $(MAKE) $(NO_STATE) $(TARGET)

131 install_h check:          install_platforms $(IMPLEMENTATIONS) FRC
132     @cd sys; pwd; $(MAKE) $(TARGET)
133     @cd vm; pwd; $(MAKE) $(TARGET)

135 #
136 # Rules for the /platforms directories. This is hardwired here because
137 # the first stage of the project (KBI) only implements the userland
138 # changes, but the only reasonable place to record the aliases is
139 # here in kernel land.
140 #
141 $(ROOT_PLAT_DIRS): $(ROOT_PLAT_DIR)
142     -$(INS.dir)

144 $(LINKED_PLATFORMS:%=$(ROOT_PLAT_DIR)/%): $(ROOT_PLAT_DIR)
145     $(INS.slink1)

147 #
148 # create directories in /usr/platform/ for the implementations that are
149 # defined in $(IMPLEMENTED_PLATFORM)
150 #

152 # Foreach $(IMPLEMENTED_PLATFORM) there can be a list of $(LINKED_PLATFORMS)
153 # that are linked to it.
154 #
155 $(USR_PLAT_DIR)/$(IMPLEMENTED_PLATFORM): $(USR_PLAT_DIR)
156     -$(INS.dir)

158 #
159 # create the links in /usr/platform/ foreach $(LINKED_PLATFORMS)
160 # to it's corresponding $(IMPLEMENTED_PLATFORM).
161 #
162 PLATFORMS          = $(LINKED_PLATFORMS)

164 $(USR_PLAT_DIRS): $(USR_PLAT_DIR)
165     $(INS.slink3)

167 PLATFORMS          += $(IMPLEMENTED_PLATFORM)

170 #
171 # Make the /platforms directories. This is hardwired here because
172 # the first stage of the project (KBI) only implements the userland
173 # changes, but the only reasonable place to record the aliases is
174 # here in kernel land.
175 #

177 install_platforms:      $(ROOT_PSM_DIR) $(USR_PSM_DIR) \
178                        $(ROOT_PLAT_DIRS) $(USR_PLAT_DIRS) \
179                        $(LINKED_PLATFORMS:%=$(ROOT_PLAT_DIR)/%) \
180                        $(USR_DESKTOP_DIR) $(USR_DESKTOP_INC_DIR) \
181                        $(USR_DESKTOP_SBIN_DIR) $(USR_DESKTOP_LIB_DIR) \
182                        $(USR_STPAUL_DIR) $(USR_STPAUL_SBIN_DIR) \
183                        $(USR_STPAUL_LIB_DIR) \
184                        $(USR_GLENDALE_DIR) $(USR_GLENDALE_SBIN_DIR) \
185                        $(USR_GLENDALE_LIB_DIR) \
186                        $(USR_HURON_DIR) \
187                        $(USR_HURON_SBIN_DIR) $(USR_HURON_LIB_DIR) \
188                        $(USR_MARAMBA_DIR) $(USR_MARAMBA_SBIN_DIR) \
189                        $(USR_MARAMBA_LIB_DIR) \
190                        $(USR_THUNDER_DIR) $(USR_THUNDER_SBIN_DIR) \
191                        $(USR_THUNDER_LIB_DIR) \
192                        $(USR_TURGO_DIR) $(USR_TURGO_SBIN_DIR) \
193                        $(USR_TURGO_LIB_DIR) \

```

new/usr/src/uts/sun4v/Makefile

4

```

194                        $(USR_CONGO_DIR) $(USR_CONGO_SBIN_DIR) \
195                        $(USR_CONGO_LIB_DIR) \
196                        $(USR_MONZA_DIR) \
197                        $(USR_MONZA_SBIN_DIR) $(USR_MONZA_SBIN_LINKS)

200 #
201 # rules for making include, sbin, lib dirs/links in
202 # /usr/platform/$(PLATFORM)/ for desktop platforms
203 #
204 $(USR_DESKTOP_INC_DIR): $(USR_DESKTOP_DIR)
205     $(INS.slink4)

207 $(USR_DESKTOP_SBIN_DIR): $(USR_DESKTOP_DIR)
208     $(INS.slink5)

210 $(USR_DESKTOP_LIB_DIR): $(USR_DESKTOP_DIR)
211     -$(INS.dir)

213 $(USR_STPAUL_DIR): $(USR_SUN4V_PLAT_DIR)
214     -$(INS.dir)

216 $(USR_STPAUL_SBIN_DIR): $(USR_STPAUL_DIR)
217     $(INS.slink5)

219 $(USR_STPAUL_LIB_DIR): $(USR_STPAUL_DIR)
220     -$(INS.dir)

222 $(USR_HURON_DIR): $(USR_SUN4V_PLAT_DIR)
223     -$(INS.dir)

225 $(USR_HURON_SBIN_DIR): $(USR_HURON_DIR)
226     $(INS.slink5)

228 $(USR_HURON_LIB_DIR): $(USR_HURON_DIR)
229     -$(INS.dir)

231 $(USR_GLENDALE_DIR): $(USR_SUN4V_PLAT_DIR)
232     -$(INS.dir)

234 $(USR_GLENDALE_SBIN_DIR): $(USR_GLENDALE_DIR)
235     $(INS.slink5)

237 $(USR_GLENDALE_LIB_DIR): $(USR_GLENDALE_DIR)
238     -$(INS.dir)

240 $(USR_MARAMBA_DIR): $(USR_SUN4V_PLAT_DIR)
241     -$(INS.dir)

243 $(USR_MARAMBA_SBIN_DIR): $(USR_MARAMBA_DIR)
244     $(INS.slink5)

246 $(USR_MARAMBA_LIB_DIR): $(USR_MARAMBA_DIR)
247     -$(INS.dir)

249 $(USR_THUNDER_DIR): $(USR_SUN4V_PLAT_DIR)
250     -$(INS.dir)

252 $(USR_THUNDER_SBIN_DIR): $(USR_THUNDER_DIR)
253     $(INS.slink5)

255 $(USR_THUNDER_LIB_DIR): $(USR_THUNDER_DIR)
256     -$(INS.dir)

258 $(USR_TURGO_DIR): $(USR_SUN4V_PLAT_DIR)
259     -$(INS.dir)

```

```
261 $(USR_TURGO_SBIN_DIR):      $(USR_TURGO_DIR)
262     $(INS.slink5)

264 $(USR_TURGO_LIB_DIR):      $(USR_TURGO_DIR)
265     -$(INS.dir)

267 $(USR_CONGO_DIR):          $(USR_SUN4V_PLAT_DIR)
268     -$(INS.dir)

270 $(USR_CONGO_SBIN_DIR):      $(USR_CONGO_DIR)
271     $(INS.slink5)

273 $(USR_CONGO_LIB_DIR):      $(USR_CONGO_DIR)
274     -$(INS.dir)

276 $(USR_MONZA_DIR):          $(USR_SUN4V_PLAT_DIR)
277     -$(INS.dir)

279 $(USR_MONZA_SBIN_DIR):      $(USR_MONZA_DIR)
280     -$(INS.dir)

282 $(USR_MONZA_SBIN_LINKS):    $(USR_MONZA_SBIN_DIR)
283     $(INS.slink7)

285 #
286 #     Full kernel lint target.
287 #
288 LINT_TARGET      = globalint

290 globalint:
291     @-$(ECHO) "\nSUN4V KERNEL: global crosschecks:"
292     @-$(LINT) $(LINTFLAGS) $(LINT_LIBS) 2>&1 | $(LGREP.2)

294 lint:    lintlib .WAIT modlintlib .WAIT $(SPARC_LINTS) $(LINT_DEPS) \
295     $(IMPLEMENTATIONS) $(LINT_CPU_KMODS)

297 include ../Makefile.targ

299 #
300 # Cross-reference customization: build a cross-reference over all of the
301 # sun4v-related directories.
302 #
303 XRDIRS = ../sun4v ../sun4 ../sfmmu ../sparc ../sun ../common
304 SHARED_XRDIRS = ../sun4v ../sun4 ../sfmmu ../sparc ../sun ../common
305 CLOSED_XRDIRS = $(SHARED_XRDIRS:../%=../% ../../../closed/uts/%)
306 XRDIRS = $(SHARED_XRDIRS)
307 $(CLOSED_BUILD)XRDIRS = $(CLOSED_XRDIRS:../../../closed/uts/sfmmu=)

304 XRPRUNE = i86pc sun4u intel

306 cscope.out tags: FRC
307     $(XREF) -x $@
```

new/usr/src/uts/sun4v/Makefile.sun4v.shared

1

```
*****
12062 Thu Aug 15 11:59:27 2013
new/usr/src/uts/sun4v/Makefile.sun4v.shared
4027 remove CLOSED_BUILD
*****
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 #
26 # This makefile contains the common definitions for the sun4v unix
27 # and all sun4v implementation architecture dependent modules.
28 #
29 #
30 #
31 # Machine type (implementation architecture):
32 #
33 PLATFORM = sun4v
34 LINKED_PLATFORMS += SUNW,Sun-Fire-T1000
35 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T5120
36 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T5220
37 LINKED_PLATFORMS += SUNW,T5140
38 LINKED_PLATFORMS += SUNW,T5240
39 LINKED_PLATFORMS += SUNW,T5440
40 LINKED_PLATFORMS += SUNW,SPARC-Enterprise-T1000
41 LINKED_PLATFORMS += SUNW,Sun-Blade-T6300
42 LINKED_PLATFORMS += SUNW,Sun-Blade-T6320
43 LINKED_PLATFORMS += SUNW,Netra-CP3260
44 LINKED_PLATFORMS += SUNW,Netra-T5220
45 LINKED_PLATFORMS += SUNW,USBRDT-5240
46 LINKED_PLATFORMS += SUNW,Netra-T5440
47 LINKED_PLATFORMS += SUNW,Sun-Blade-T6340
48 PROMIF = ieee1275
49 PSMBASE = $(UTSBASE)/../psm
50 #
51 #
52 # uname -m value
53 #
54 UNAME_M = $(PLATFORM)
55 #
56 #
57 # Definitions for the platform-specific /platform directories.
58 #
59 # PLATFORMS designates those sun4v machines which have no platform
60 # specific code.
61 #
```

new/usr/src/uts/sun4v/Makefile.sun4v.shared

2

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

```

128 PLATLIB          = $(PLAT_DIR)/$(OBJS_DIR)/libplatmod.so
130 CPUNAME          = cpu
131 CPULIB           = $(CPU_DIR)/$(OBJS_DIR)/libcpu.so
133 SYM_MOD          = $(OBJS_DIR)/unix.sym

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

156 #
157 #   Include machine independent rules. Note that this does not imply
158 #   that the resulting module from rules in Makefile.uts is machine
159 #   independent. Only that the build rules are machine independent.
160 #
161 include $(UTSBASE)/Makefile.uts

163 CTFMERGE_GUDIR   = sun4v

165 #
166 #   machine specific optimization, override default in Makefile.master
167 #
168 CC_XARCH          = -m64 -xarch=sparcvis
169 AS_XARCH          = -xarch=v9v
170 COPTIMIZE         = -xO3
171 CCMODE            = -Xa

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

```

```

195 CPPFLAGS         += -DGLREG
197 ASFLAGS           += $(AS_XARCH) -DGLREG
199 AS_INC_PATH       += -I$(DSF_DIR)/$(OBJS_DIR)
201 LINT_KMODS        += $(GENUNIX_KMODS)
203 LINT_DEFS         = -m64

205 #
206 #   The following must be defined for all implementations:
207 #
208 #   MAPFILE:          ld mapfile for the build of kernel/unix.
209 #   MODSTUBS:        Module stubs source file.
210 #   GENCONST_SRC:    genconst.c
211 #   OFFSETS:         offsets.in
212 #   PLATFORM_OFFSETS: Platform specific mach_offsets.in
213 #   FDOFFSETS:       fd_offsets.in
214 #
215 MAPFILE           = $(UTSBASE)/sun4/conf/Mapfile
216 MODSTUBS          = $(UTSBASE)/sparc/ml/modstubs.s
217 GENCONST_SRC      = $(UTSBASE)/sun4/ml/genconst.c
218 OFFSETS           = $(UTSBASE)/sun4/ml/offsets.in
219 PLATFORM_OFFSETS  = $(UTSBASE)/sun4v/ml/mach_offsets.in
220 FDOFFSETS         = $(UTSBASE)/sun4v/ml/fd_offsets.in

222 #
223 #   Define the actual specific platforms
224 #

226 MACHINE_DEFS     = -D$(PLATFORM) -D_MACHDEP -DSFMMU
227 MACHINE_DEFS     += -DMAX_MEM_NODES=8

229 #
230 #   Software workarounds for hardware "features"
231 #

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

235 #
236 #   Debugging level
237 #
238 #   Special knowledge of which special debugging options effect which
239 #   file is used to optimize the build if these flags are changed.
240 #
241 #   XXX: The above could possibly be done for more flags and files, but
242 #   is left as an experiment to the interested reader. Be forewarned,
243 #   that excessive use could lead to maintenance difficulties.
244 #
245 #   Note: kslice can be enabled for the sun4v, but is disabled by default
246 #   in all cases.
247 #

249 DEBUG_DEFS_OBJ64 =
250 DEBUG_DEFS_DBG64 = -DDEBUG
251 DEBUG_DEFS        = $(DEBUG_DEFS_$(BUILD_TYPE))

253 DEBUG_COND_OBJ64 :sh = echo \\043
254 DEBUG_COND_DBG64 =
255 IF_DEBUG_OBJ      = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

257 $(IF_DEBUG_OBJ)trap.o      :=      DEBUG_DEFS      += -DTRAPDEBUG
258 $(IF_DEBUG_OBJ)mach_trap.o :=      DEBUG_DEFS      += -DTRAPDEBUG
259 $(IF_DEBUG_OBJ)syscall_trap.o :=      DEBUG_DEFS      += -DSYSCALLTRACE

```



```

260 $(IF_DEBUG_OBJ)clock.o      :=      DEBUG_DEFS      += -DKSLICE=0

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

266 $(IF_TRAPTRACE_OBJ)mach_locore.o      :=      DEBUG_DEFS      += -DTRAPTRACE
267 $(IF_TRAPTRACE_OBJ)mlsetup.o         :=      DEBUG_DEFS      += -DTRAPTRACE
268 $(IF_TRAPTRACE_OBJ)syscall_trap.o    :=      DEBUG_DEFS      += -DTRAPTRACE
269 $(IF_TRAPTRACE_OBJ)startup.o         :=      DEBUG_DEFS      += -DTRAPTRACE
270 $(IF_TRAPTRACE_OBJ)mach_startup.o    :=      DEBUG_DEFS      += -DTRAPTRACE
271 $(IF_TRAPTRACE_OBJ)mp_startup.o      :=      DEBUG_DEFS      += -DTRAPTRACE
272 $(IF_TRAPTRACE_OBJ)cpu_states.o      :=      DEBUG_DEFS      += -DTRAPTRACE
273 $(IF_TRAPTRACE_OBJ)mach_cpu_states.o :=      DEBUG_DEFS      += -DTRAPTRACE
274 $(IF_TRAPTRACE_OBJ)interrupt.o       :=      DEBUG_DEFS      += -DTRAPTRACE
275 $(IF_TRAPTRACE_OBJ)mach_interrupt.o  :=      DEBUG_DEFS      += -DTRAPTRACE
276 $(IF_TRAPTRACE_OBJ)sfmmu_asm.o       :=      DEBUG_DEFS      += -DTRAPTRACE
277 $(IF_TRAPTRACE_OBJ)trap_table.o      :=      DEBUG_DEFS      += -DTRAPTRACE
278 $(IF_TRAPTRACE_OBJ)xc.o              :=      DEBUG_DEFS      += -DTRAPTRACE
279 $(IF_TRAPTRACE_OBJ)mach_xc.o          :=      DEBUG_DEFS      += -DTRAPTRACE
280 $(IF_TRAPTRACE_OBJ)wbuf.o            :=      DEBUG_DEFS      += -DTRAPTRACE
281 $(IF_TRAPTRACE_OBJ)trap.o            :=      DEBUG_DEFS      += -DTRAPTRACE
282 $(IF_TRAPTRACE_OBJ)mach_trap.o       :=      DEBUG_DEFS      += -DTRAPTRACE
283 $(IF_TRAPTRACE_OBJ)x_call.o          :=      DEBUG_DEFS      += -DTRAPTRACE

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

287 #$(IF_DEBUG_OBJ)lock_prim.o      := DEBUG_DEFS      += -DDISP_LOCK_STATS
288 #$(IF_DEBUG_OBJ)disp.o          := DEBUG_DEFS      += -DDISP_LOCK_STATS

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

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

294 #
295 #   Collect the preprocessor definitions to be associated with *all*
296 #   files.
297 #
298 ALL_DEFS      = $(MACHINE_DEFS) $(WORKAROUND_DEFS) $(DEBUG_DEFS) \
299               $(OPTION_DEFS)
300 GENCONST_DEFS = $(MACHINE_DEFS) $(OPTION_DEFS)

302 #
303 # ----- TRANSITIONAL SECTION -----
304 #

306 #
307 #   Not everything which *should* be a module is a module yet. The
308 #   following is a list of such objects which are currently part of
309 #   the base kernel but should soon become kmods.
310 #
311 MACH_NOT_YET_KMODS      = $(AUTOCONF_OBJS)

313 #
314 # ----- END OF TRANSITIONAL SECTION -----
315 #

317 #
318 #   The kernels modules which are "implementation architecture"
319 #   specific for this machine are enumerated below. Note that most
320 #   of these modules must exist (in one form or another) for each
321 #   architecture.
322 #
323 #   Common Drivers (usually pseudo drivers) (/kernel/drv):
324 #

```

```

326 #
327 #   Machine Specific Driver Modules (/kernel/drv):
328 #
329 DRV_KMODS      += bge
330 DRV_KMODS      += cnex
331 DRV_KMODS      += cpc
332 DRV_KMODS      += drctl
333 DRV_KMODS      += ds_pri
334 DRV_KMODS      += ds_snmp
335 DRV_KMODS      += ebus
336 DRV_KMODS      += fpc
337 DRV_KMODS      += glvc
338 DRV_KMODS      += mdesc
339 DRV_KMODS      += niumx
340 DRV_KMODS      += ntwdt
341 DRV_KMODS      += nxge
342 DRV_KMODS      += n2piupc
343 DRV_KMODS      += iospc
344 DRV_KMODS      += n2rng
345 DRV_KMODS      += px
346 DRV_KMODS      += qcn
347 DRV_KMODS      += rootnex
348 DRV_KMODS      += su
349 DRV_KMODS      += tpm
350 DRV_KMODS      += trapstat
351 DRV_KMODS      += vcc
352 DRV_KMODS      += vdc
353 DRV_KMODS      += vds
354 DRV_KMODS      += vldc
355 DRV_KMODS      += vlids
356 DRV_KMODS      += vnet
357 DRV_KMODS      += vnex
358 DRV_KMODS      += vsw

360 $(CLOSED_BUILD)CLOSED_DRV_KMODS += bmc
361 $(CLOSED_BUILD)CLOSED_DRV_KMODS += memtest
362 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ncp
363 $(CLOSED_BUILD)CLOSED_DRV_KMODS += n2cp

360 #
361 #   Exec Class Modules (/kernel/exec):
362 #
363 EXEC_KMODS      +=

365 #
366 #   Scheduling Class Modules (/kernel/sched):
367 #
368 SCHED_KMODS     +=

370 #
371 #   File System Modules (/kernel/fs):
372 #
373 FS_KMODS        +=

375 #
376 #   Streams Modules (/kernel/strmod):
377 #
378 STRMOD_KMODS   += kb

380 #
381 #   'System' Modules (/kernel/sys):
382 #
383 SYS_KMODS       +=

385 #
386 #   'User' Modules (/kernel/misc):

```

```
387 #
388 MISC_KMODS      += bootdev
389 MISC_KMODS      += dr_cpu
390 MISC_KMODS      += dr_io
391 MISC_KMODS      += dr_mem
392 MISC_KMODS      += ds
393 MISC_KMODS      += fault_iso
394 MISC_KMODS      += ldc
395 MISC_KMODS      += obpsym
396 MISC_KMODS      += platmod
397 MISC_KMODS      += platsvc
398 MISC_KMODS      += vis
399 MISC_KMODS      += pcie

401 #      md5 optimized for Niagara
402 #
403 MISC_KMODS      += md5

405 #
406 #      Brand modules
407 #
408 BRAND_KMODS     += snl_brand s10_brand

410 #
411 #      Software Cryptographic Providers (/kernel/crypto):
412 #
413 CRYPTO_KMODS    += arcfour

415 #
416 #      generic-unix module (/kernel/genunix):
417 #
418 GENUNIX_KMODS   += genunix

425 #      'User' "Modules" excluded from the Full Kernel lint target:
426 #
427 $(CLOSED_BUILD)CLOSED_NLMISC_KMODS      += forthdebug

420 #
421 #      Modules eXcluded from the product:
422 #
423 XMODS           +=

425 #
426 #      cpu modules
427 #
428 CPU_KMODS       += generic niagara niagara2 vfalls kt

430 LINT_CPU_KMODS += generic

432 #
433 #      Performance Counter BackEnd Modules (/usr/kernel/pcbe):
434 #
435 PCBE_KMODS      += niagara_pcbe
436 PCBE_KMODS      += niagara2_pcbe
437 PCBE_KMODS      += vfalls_pcbe
438 PCBE_KMODS      += kt_pcbe
```

```

*****
2802 Thu Aug 15 11:59:28 2013
new/usr/src/uts/sun4v/sys/Makefile
4027 remove CLOSED_BUILD
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 #
26 # uts/sun4v/sys/Makefile
27 #
28 # include global definitions
29 UTBASE = ../../

31 #
32 # include global definitions
33 #
34 include ../Makefile.sun4v

36 #
37 # Override defaults.
38 #
39 FILEMODE = 644

41 SUN4_HDRS= \
42 clock.h \
43 cmp.h \
44 cpc_ultra.h \
45 cpu_sgnblk_defs.h \
46 ddi_subrdefs.h \
47 dvma.h \
48 eeprom.h \
49 fcode.h \
50 idprom.h \
51 intr.h \
52 intreg.h \
53 ivintr.h \
54 memlist_plat.h \
55 memnode.h \
56 nexusdebug.h \
57 prom_debug.h \
58 scb.h \
59 sun4asi.h \
60 tod.h \
61 trapstat.h \

```

```

62 vis.h \
63 vm_machparam.h \
64 x_call.h \
65 xc_impl.h \
66 zsmach.h

68 CLOSED_SUN4_HDRS= \
69 memtestio.h

68 HDRS= \
69 ds_pri.h \
70 ds_snmp.h \
71 hypervisor_api.h \
72 hsvc.h \
73 machasi.h \
74 machclock.h \
75 machcpuvar.h \
76 mach_descrip.h \
77 machintreg.h \
78 machparam.h \
79 machsystem.h \
80 machthread.h \
81 mmu.h \
82 niagaraasi.h \
83 niagaraargs.h \
84 ntwdt.h \
85 pte.h \
86 prom_plat.h \
87 qcn.h \
88 soft_state.h \
89 traptrace.h \
90 vlds.h

95 CLOSED_HDRS= \
96 memtestio_kt.h \
97 memtestio_ni.h \
98 memtestio_n2.h \
99 memtestio_v.h \
100 memtestio_vf.h

92 Roothdrs= $(HDRS:%=$(USR_PSM_ISYS_DIR)/%)
103 $(CLOSED_BUILD)Roothdrs += $(CLOSED_HDRS:%=$(USR_PSM_ISYS_DIR)/%)

94 SUN4_Roothdrs= $(SUN4_HDRS:%=$(USR_PSM_ISYS_DIR)/%)
106 $(CLOSED_BUILD)SUN4_Roothdrs += $(CLOSED_SUN4_HDRS:%=$(USR_PSM_ISYS_DIR)/%)

96 ROOTDIR= $(ROOT)/usr/share/src
97 ROOTDIRS= $(ROOTDIR)/uts $(ROOTDIR)/uts/$(PLATFORM)

99 ROOTLINK= $(ROOTDIR)/uts/$(PLATFORM)/sys
100 LINKDEST= ../../../../platform/$(PLATFORM)/include/sys

102 CHECKHDRS= $(HDRS:%.h=%.check) \
103 $(SUN4_HDRS:%.h=%.cmncheck)

117 $(CLOSED_BUILD)CHECKHDRS += \
118 $(CLOSED_HDRS:%.h=%.check) \
119 $(CLOSED_SUN4_HDRS:%.h=%.cmncheck)

105 .KEEP_STATE:

107 .PARALLEL: $(CHECKHDRS) $(Roothdrs) $(SUN4_Roothdrs)

109 install_h: $(ROOTDIRS) .WAIT \
110 $(Roothdrs) .WAIT \
111 $(SUN4_Roothdrs) .WAIT $(ROOTLINK)

```

```
113 check: $(CHECKHDRS)

115 #
116 # install rules
117 #
118 $(USR_PSM_ISYS_DIR)/%: ../../sfmmu/sys/% $(USR_PSM_ISYS_DIR)
119     $(INS.file)

121 $(USR_PSM_ISYS_DIR)/%: ../../sun4/sys/% $(USR_PSM_ISYS_DIR)
122     $(INS.file)

140 $(USR_PSM_ISYS_DIR)/%: $(CLOSED)/uts/sun4/sys/% $(USR_PSM_ISYS_DIR)
141     $(INS.file)

143 $(USR_PSM_ISYS_DIR)/%: $(CLOSED)/uts/sun4v/sys/% $(USR_PSM_ISYS_DIR)
144     $(INS.file)

124 $(ROOTDIRS):
125     $(INS.dir)

127 # -r because this used to be a directory and is now a link.
128 $(ROOTLINK): $(ROOTDIRS)
129     -$(RM) -r $@; $(SYMLINK) $(LINKDEST) $@

131 mon/%.check: mon/%.h
132     $(DOT_H_CHECK)

134 %.check: ../../sfmmu/sys/%.h
135     $(DOT_H_CHECK)
158 %.check: $(CLOSED)/uts/sun4v/sys/%.h
159     $(DOT_H_CHECK)
136 %.cmncheck: ../../sun4/sys/%.h
137     $(DOT_H_CHECK)
162 %.cmncheck: $(CLOSED)/uts/sun4/sys/%.h
163     $(DOT_H_CHECK)

139 FRC:

141 include ../../Makefile.targ
```