```
**********************************************************
   38165 Thu Apr 25 02:20:38 2019
new/usr/src/Makefile.master
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright 2014 Garrett D’Amore <garrett@damore.org>
# Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
# Copyright 2015 Gary Mills
# Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
# Copyright 2016 Toomas Soome <tsoome@me.com>
# Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
# Copyright (c) 2019, Joyent, Inc.
#

#
# Makefile.master, global definitions for system source
#
ROOT=           /proto

#
# Adjunct root, containing an additional proto area to be used for headers
# and libraries.
#
ADJUNCT_PROTO=

#
# Adjunct for building things that run on the build machine.
#
NATIVE_ADJUNCT= /usr

#
# RELEASE_BUILD should be cleared for final release builds.
# NOT_RELEASE_BUILD is exactly what the name implies.
#
```

```
# __GNUC toggles the building of ON components using gcc and related tools.
# Normally set to ‘#’, set it to ‘’ to do gcc build.
#
# The declaration POUND_SIGN is always ’#’. This is needed to get around the
# make feature that ’#’ is always a comment delimiter, even when escaped or
# quoted. We use this macro expansion method to get POUND_SIGN rather than
# always breaking out a shell because the general case can cause a noticable
# slowdown in build times when so many Makefiles include Makefile.master.
#
# While the majority of users are expected to override the setting below
# with an env file (via nightly or bldenv), if you aren’t building that way
# (ie, you’re using "ws" or some other bootstrapping method) then you need
# this definition in order to avoid the subshell invocation mentioned above.
#

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

NOT_RELEASE_BUILD=
RELEASE_BUILD=                          $(POUND_SIGN)
$(RELEASE_BUILD)NOT_RELEASE_BUILD=      $(POUND_SIGN)
PATCH_BUILD=                            $(POUND_SIGN)

# SPARC_BLD is ’#’ for an Intel build.
# INTEL_BLD is ’#’ for a Sparc build.
SPARC_BLD_1=    $(MACH:i386=$(POUND_SIGN))
SPARC_BLD=      $(SPARC_BLD_1:sparc=)
INTEL_BLD_1=    $(MACH:sparc=$(POUND_SIGN))
INTEL_BLD=      $(INTEL_BLD_1:i386=)

# The variables below control the compilers used during the build.
# There are a number of permutations.
#
# __GNUC and __SUNC control (and indicate) the primary compiler.  Whichever
# one is not POUND_SIGN is the primary, with the other as the shadow.  They
# may also be used to control entirely compiler-specific Makefile assignments.
# __GNUC and GCC are the default.
#
# __GNUC64 indicates that the 64bit build should use the GNU C compiler.
# There is no Sun C analogue.
#
# The following version-specific options are operative regardless of which
# compiler is primary, and control the versions of the given compilers to be
# used.  They also allow compiler-version specific Makefile fragments.
#

__SUNC=                 $(POUND_SIGN)
$(__SUNC)__GNUC=        $(POUND_SIGN)
__GNUC64=               $(__GNUC)

# Allow build-time "configuration" to enable or disable some things.
# The default is POUND_SIGN, meaning "not enabled". If the environment
# passes in an override like ENABLE_SMB_PRINTING= (empty) that will
# uncomment things in the lower Makefiles to enable the feature.
ENABLE_SMB_PRINTING=    $(POUND_SIGN)

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

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

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

```
# define runtime JAVA_HOME, primarily for cmd/pools/poold
JAVA_HOME=      /usr/java
# define buildtime JAVA_ROOT
JAVA_ROOT=      /usr/java
# Build uses java7 by default.  Pass one the variables below set to empty
# string in the environment to override.
BLD_JAVA_6=     $(POUND_SIGN)
BLD_JAVA_8=     $(POUND_SIGN)

GNUC_ROOT=      /opt/gcc/4.4.4
GCCLIBDIR=      $(GNUC_ROOT)/lib
GCCLIBDIR64=    $(GNUC_ROOT)/lib/$(MACH64)

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

RPCGEN=         /usr/bin/rpcgen
STABS=          $(ONBLD_TOOLS)/bin/$(MACH)/stabs
ELFEXTRACT=     $(ONBLD_TOOLS)/bin/$(MACH)/elfextract
MBH_PATCH=      $(ONBLD_TOOLS)/bin/$(MACH)/mbh_patch
BTXLD=          $(ONBLD_TOOLS)/bin/$(MACH)/btxld
VTFONTCVT=      $(ONBLD_TOOLS)/bin/$(MACH)/vtfontcvt
# echo(1) and true(1) are specified without absolute paths, so that the shell
# spawned by make(1) may use the built-in versions.  This is minimally
# problematic, as the shell spawned by make(1) is known and under control, the
# only risk being if the shell falls back to $PATH.
#
# We specifically want an echo(1) that does interpolation of escape sequences,
# which ksh93, /bin/sh, and bash will all provide.
ECHO=           echo
TRUE=           true
INS=            $(ONBLD_TOOLS)/bin/$(MACH)/install
SYMLINK=        /usr/bin/ln -s
LN=             /usr/bin/ln
MKDIR=          /usr/bin/mkdir
CHMOD=          /usr/bin/chmod
MV=             /usr/bin/mv -f
RM=             /usr/bin/rm -f
CUT=            /usr/bin/cut
NM=             /usr/ccs/bin/nm
DIFF=           /usr/bin/diff
GREP=           /usr/bin/grep
EGREP=          /usr/bin/egrep
ELFWRAP=        /usr/bin/elfwrap
KSH93=          /usr/bin/ksh93
SED=            /usr/bin/sed
AWK=            /usr/bin/nawk
CP=             /usr/bin/cp -f
MCS=            /usr/ccs/bin/mcs
CAT=            /usr/bin/cat
ELFDUMP=        /usr/ccs/bin/elfdump
M4=             /usr/bin/m4
GM4=            /usr/bin/gm4
STRIP=          /usr/ccs/bin/strip
LEX=            /usr/ccs/bin/lex
FLEX=           /usr/bin/flex
YACC=           /usr/ccs/bin/yacc
BISON=          /usr/bin/bison
CPP=            /usr/lib/cpp
ANSI_CPP=       $(GNUC_ROOT)/bin/cpp
JAVAC=          $(JAVA_ROOT)/bin/javac
JAVAH=          $(JAVA_ROOT)/bin/javah
JAVADOC=        $(JAVA_ROOT)/bin/javadoc
RMIC=           $(JAVA_ROOT)/bin/rmic
JAR=            $(JAVA_ROOT)/bin/jar
CTFCONVERT=     $(ONBLD_TOOLS)/bin/$(MACH)/ctfconvert
CTFDIFF=        $(ONBLD_TOOLS)/bin/$(MACH)/ctfdiff
```

```
CTFMERGE=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfmerge
CTFSTABS=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstabs
CTFSTRIP=       $(ONBLD_TOOLS)/bin/$(MACH)/ctfstrip
NDRGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/ndrgen
GENOFFSETS=     $(ONBLD_TOOLS)/bin/genoffsets
XREF=           $(ONBLD_TOOLS)/bin/xref
FIND=           /usr/bin/find
PERL=           /usr/bin/perl
PERL_VERSION=   5.10.0
PERL_PKGVERS=   -510
PERL_ARCH =             i86pc-solaris-64int
$(SPARC_BLD)PERL_ARCH = sun4-solaris-64int
PYTHON_VERSION= 2.7
PYTHON_PKGVERS= -27
PYTHON_SUFFIX=
PYTHON=         /usr/bin/python$(PYTHON_VERSION)
PYTHON3_VERSION=        3.5
PYTHON3_PKGVERS=        -35
PYTHON3_SUFFIX=         m
PYTHON3=                /usr/bin/python$(PYTHON3_VERSION)
$(BUILDPY3TOOLS)TOOLS_PYTHON=           $(PYTHON3)
$(BUILDPY2TOOLS)TOOLS_PYTHON=           $(PYTHON)
SORT=           /usr/bin/sort
TR=             /usr/bin/tr
TOUCH=          /usr/bin/touch
WC=             /usr/bin/wc
XARGS=          /usr/bin/xargs
ELFEDIT=        /usr/bin/elfedit
DTRACE=         /usr/sbin/dtrace -xnolibs
UNIQ=           /usr/bin/uniq
TAR=            /usr/bin/tar
ASTBINDIR=      /usr/ast/bin
MSGCC=          $(ASTBINDIR)/msgcc
MSGFMT=         /usr/bin/msgfmt -s
LCDEF=          $(ONBLD_TOOLS)/bin/$(MACH)/localedef
TIC=            $(ONBLD_TOOLS)/bin/$(MACH)/tic
ZIC=            $(ONBLD_TOOLS)/bin/$(MACH)/zic
OPENSSL=        /usr/bin/openssl
CPCGEN=         $(ONBLD_TOOLS)/bin/$(MACH)/cpcgen

FILEMODE=       644
DIRMODE=        755

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

# For stylistic checks
#
# Note that the X and C checks are not used at this time and may need
# modification when they are actually used.
#
CSTYLE=         $(ONBLD_TOOLS)/bin/cstyle
CSTYLE_TAIL=
HDRCHK=         $(ONBLD_TOOLS)/bin/hdrchk
HDRCHK_TAIL=
JSTYLE=         $(ONBLD_TOOLS)/bin/jstyle

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

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

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

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

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

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

# The path to python that will be used for the shebang line when installing
# python scripts to the proto area. This is overridden by makefiles to
# select to the correct version.
PYSHEBANG=      $(PYTHON)

#
# Python bakes the mtime of the .py file into the compiled .pyc and
# rebuilds if the baked-in mtime != the mtime of the source file
# (rather than only if it’s less than), thus when installing python
# files we must make certain to not adjust the mtime of the source
# (.py) file.
#
INS.pyfile=     $(RM) $@; $(SED) \
                -e "1s:^\#!@PYTHON@:\#!$(PYSHEBANG):" \
                -e "1s:^\#!@TOOLS_PYTHON@:\#!$(TOOLS_PYTHON):" \
                < $< > $@; $(CHMOD) $(FILEMODE) $@; $(TOUCH) -r $< $@

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

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

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

sparc_BUILD64=
i386_BUILD64=
BUILD64=        $($(MACH)_BUILD64)

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

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

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

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

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

# GCC, especially, is increasingly beginning to auto-inline functions and
# sadly does so separately not under the general -fno-inline-functions
# Additionally, we wish to prevent optimisations which cause GCC to clone
# functions -- in particular, these may cause unhelpful symbols to be
# emitted instead of function names
CCNOAUTOINLINE= \
        -_gcc=-fno-inline-small-functions \
        -_gcc=-fno-inline-functions-called-once \
        -_gcc=-fno-ipa-cp \
        -_gcc7=-fno-ipa-icf \
        -_gcc8=-fno-ipa-icf \
        -_gcc7=-fno-clone-functions \
        -_gcc8=-fno-clone-functions

# GCC may put functions in different named sub-sections of .text based on
# their presumed calling frequency.  At least in the kernel, where we actually
# deliver relocatable objects, we don’t want this to happen.
#
# Since at present we don’t benefit from this even in userland, we disable it gl
# but the application of this may move into usr/src/uts/ in future.
CCNOREORDER=    \
        -_gcc7=-fno-reorder-functions \
        -_gcc8=-fno-reorder-functions

#
# gcc has a rather aggressive optimization on by default that infers loop
# bounds based on undefined behavior (!!).  This can lead to some VERY
# surprising optimizations -- ones that may be technically correct in the
# strictest sense but also result in incorrect program behavior.  We turn
# this optimization off, with extreme prejudice.
#
CCNOAGGRESSIVELOOPS= \
        -_gcc7=-fno-aggressive-loop-optimizations \
        -_gcc8=-fno-aggressive-loop-optimizations

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

#
# compiler '-xarch' flag. This is here to centralize it and make it
# overridable for testing.
sparc_XARCH=    -m32
sparcv9_XARCH=  -m64
i386_XARCH=     -m32
amd64_XARCH=    -m64 -Ui386 -U__i386

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

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

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

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

#
# disable the incremental linker
ILDOFF=                 -xildoff
#
XFFLAG=                 -xF=%all
XESS=                   -xs
XSTRCONST=              -xstrconst

#
# turn warnings into errors (C)
CERRWARN = -errtags=yes -errwarn=%all
CERRWARN += -erroff=E_EMPTY_TRANSLATION_UNIT
CERRWARN += -erroff=E_STATEMENT_NOT_REACHED
```

```
CERRWARN += -_gcc=-Wno-missing-braces
CERRWARN += -_gcc=-Wno-sign-compare
CERRWARN += -_gcc=-Wno-unknown-pragmas
CERRWARN += -_gcc=-Wno-unused-parameter
CERRWARN += -_gcc=-Wno-missing-field-initializers

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


CERRWARN += -_smatch=-p=illumos_user
include $(SRC)/Makefile.smatch

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

# C standard.  Keep Studio flags until we get rid of lint.
CSTD_GNU89=     -xc99=%none
CSTD_GNU99=     -xc99=%all
CSTD=           $(CSTD_GNU89)
C99LMODE=       $(CSTD:-xc99%=-Xc99%)

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

sparc_ASFLAGS=  $(sparc_AS_XARCH)
sparcv9_ASFLAGS=$(sparcv9_AS_XARCH)
i386_ASFLAGS=   $(i386_AS_XARCH)
amd64_ASFLAGS=  $(amd64_AS_XARCH)

#
sparc_COPTFLAG=         -xO3
sparcv9_COPTFLAG=       -xO3
i386_COPTFLAG=          -O
amd64_COPTFLAG=         -xO3

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

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

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

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

#
# We force the compilers to generate the debugging information best understood
# by the CTF tools.  With Sun Studio this is stabs due to bugs in the Studio
# compilers.  With GCC this is DWARF v2.
#
```

```
DEBUGFORMAT=    -_cc=-xdebugformat=stabs -_gcc=-gdwarf-2

#
# Ask the compiler to include debugging information
#
CCGDEBUG= -g $(DEBUGFORMAT)

#
# Flags used to build in debug mode for ctf generation.
#
CTF_FLAGS_sparc = $(CCGDEBUG) -Wc,-Qiselect-T1 $(CSTD) $(CNOGLOBAL)
CTF_FLAGS_i386  = $(CCGDEBUG) $(CSTD) $(CNOGLOBAL)

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

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

CTF_FLAGS_32    = $(CTF_FLAGS_$(MACH))
CTF_FLAGS_64    = $(CTF_FLAGS_$(MACH64))
CTF_FLAGS       = $(CTF_FLAGS_32)

#
# Flags used with genoffsets
#
GENOFFSETS_FLAGS = $(CALLSYMS)

OFFSETS_CREATE = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
        $(CW) --noecho $(CW_CC_COMPILERS) -- $(GENOFFSETS_FLAGS) \
        $(CFLAGS) $(CPPFLAGS)

OFFSETS_CREATE64 = $(GENOFFSETS) -s $(CTFSTABS) -r $(CTFCONVERT) \
        $(CW) --noecho $(CW_CC_COMPILERS) -- $(GENOFFSETS_FLAGS) \
        $(CFLAGS64) $(CPPFLAGS)

#
# tradeoff time for space (smaller is better)
#
sparc_SPACEFLAG         = -xspace -W0,-Lt
sparcv9_SPACEFLAG       = -xspace -W0,-Lt
i386_SPACEFLAG          = -xspace
amd64_SPACEFLAG         =

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

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

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

sparc_XREGSFLAG         = -xregs=no%appl
sparcv9_XREGSFLAG       = -xregs=no%appl
i386_XREGSFLAG          =
amd64_XREGSFLAG         =

XREGSFLAG               = $($(MACH)_XREGSFLAG)
```

```
XREGSFLAG64             = $($(MACH64)_XREGSFLAG)

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

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

CSOURCEDEBUGFLAGS       =
CCSOURCEDEBUGFLAGS      =
$(SRCDBGBLD)CSOURCEDEBUGFLAGS   = $(CCGDEBUG) -xs
$(SRCDBGBLD)CCSOURCEDEBUGFLAGS  = $(CCGDEBUG) -xs

CFLAGS=         $(COPTFLAG) $($(MACH)_CFLAGS) $(SPACEFLAG) $(CCMODE) \
                $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG) \
                $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CCNOREORDER) \
                $(CCNOAGGRESSIVELOOPS) \
                $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)
CFLAGS64=       $(COPTFLAG64) $($(MACH64)_CFLAGS) $(SPACEFLAG64) $(CCMODE64) \
                $(ILDOFF) $(CERRWARN) $(CSTD) $(CCUNBOUND) $(IROPTFLAG64) \
                $(CGLOBALSTATIC) $(CCNOAUTOINLINE) $(CCNOREORDER) \
                $(CCNOAGGRESSIVELOOPS) \
                $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS64)
#
# Flags that are used to build parts of the code that are subsequently
# run on the build machine (also known as the NATIVE_BUILD).
#
NATIVE_CFLAGS=  $(COPTFLAG) $($(NATIVE_MACH)_CFLAGS) $(CCMODE) \
                $(ILDOFF) $(CERRWARN) $(CSTD) $($(NATIVE_MACH)_CCUNBOUND) \
                $(IROPTFLAG) $(CGLOBALSTATIC) $(CCNOAUTOINLINE) \
                $(CCNOREORDER) $(CSOURCEDEBUGFLAGS) $(CUSERFLAGS)

DTEXTDOM=-DTEXT_DOMAIN=\"$(TEXT_DOMAIN)\"       # For messaging.
DTS_ERRNO=-D_TS_ERRNO
CPPFLAGS.first= # Please keep empty.  Only lower makefiles should set this.
CPPFLAGS.master=$(DTEXTDOM) $(DTS_ERRNO) \
        $(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) $(ENVCPPFLAGS4) \
        $(ADJUNCT_PROTO:%=-I%/usr/include)
CPPFLAGS.native=$(ENVCPPFLAGS1) $(ENVCPPFLAGS2) $(ENVCPPFLAGS3) \
                $(ENVCPPFLAGS4) -I$(NATIVE_ADJUNCT)/include
CPPFLAGS=       $(CPPFLAGS.first) $(CPPFLAGS.master)
AS_CPPFLAGS=    $(CPPFLAGS.first) $(CPPFLAGS.master)
JAVAFLAGS=      -source 1.6 -target 1.6 -Xlint:deprecation,-options

#
# For source message catalogue
#
.SUFFIXES: $(SUFFIXES) .i .po
MSGROOT= $(ROOT)/catalog
MSGDOMAIN= $(MSGROOT)/$(TEXT_DOMAIN)
MSGDOMAINPOFILE = $(MSGDOMAIN)/$(POFILE)
DCMSGDOMAIN= $(MSGROOT)/LC_TIME/$(TEXT_DOMAIN)
DCMSGDOMAINPOFILE = $(DCMSGDOMAIN)/$(DCFILE:.dc=.po)
```

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

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

sparc_CCFLAGS=          -cg92 -compat=4 \
                        -Qoption ccfe -messages=no%anachronism \
                        $(CCERRWARN)
sparcv9_CCFLAGS=        $(sparcv9_XARCH) -dalign -compat=5 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCCREGSYM) \
                        $(CCERRWARN)
i386_CCFLAGS=           -compat=4 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCERRWARN)
amd64_CCFLAGS=          $(amd64_XARCH) -compat=5 \
                        -Qoption ccfe -messages=no%anachronism \
                        -Qoption ccfe -features=no%conststrings \
                        $(CCERRWARN)

sparc_CCOPTFLAG=        -O
sparcv9_CCOPTFLAG=      -O
i386_CCOPTFLAG=         -O
amd64_CCOPTFLAG=        -O

CCOPTFLAG=      $($(MACH)_CCOPTFLAG)
CCOPTFLAG64=    $($(MACH64)_CCOPTFLAG)
CCFLAGS=        $(CCOPTFLAG) $($(MACH)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
                $(CCUSERFLAGS)
CCFLAGS64=      $(CCOPTFLAG64) $($(MACH64)_CCFLAGS) $(CCSOURCEDEBUGFLAGS) \
                $(CCUSERFLAGS64)

#
#
#
ELFWRAP_FLAGS   =
ELFWRAP_FLAGS64 =       -64

#
# Various mapfiles that are used throughout the build, and delivered to
# /usr/lib/ld.
#
MAPFILE.NED_i386 =      $(SRC)/common/mapfiles/common/map.noexdata
MAPFILE.NED_sparc =
MAPFILE.NED =           $(MAPFILE.NED_$(MACH))
MAPFILE.PGA =           $(SRC)/common/mapfiles/common/map.pagealign
MAPFILE.NES =           $(SRC)/common/mapfiles/common/map.noexstk
MAPFILE.FLT =           $(SRC)/common/mapfiles/common/map.filter
MAPFILE.LEX =           $(SRC)/common/mapfiles/common/map.lex.yy

#
```

```
# Generated mapfiles that are compiler specific, and used throughout the
# build.  These mapfiles are not delivered in /usr/lib/ld.
#
MAPFILE.NGB_sparc=      $(SRC)/common/mapfiles/gen/sparc_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_sparc= \
                        $(SRC)/common/mapfiles/gen/sparc_gcc_map.noexeglobs
MAPFILE.NGB_sparcv9=    $(SRC)/common/mapfiles/gen/sparcv9_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_sparcv9= \
                        $(SRC)/common/mapfiles/gen/sparcv9_gcc_map.noexeglobs
MAPFILE.NGB_i386=       $(SRC)/common/mapfiles/gen/i386_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_i386= \
                        $(SRC)/common/mapfiles/gen/i386_gcc_map.noexeglobs
MAPFILE.NGB_amd64=      $(SRC)/common/mapfiles/gen/amd64_cc_map.noexeglobs
$(__GNUC64)MAPFILE.NGB_amd64= \
                        $(SRC)/common/mapfiles/gen/amd64_gcc_map.noexeglobs
MAPFILE.NGB =           $(MAPFILE.NGB_$(MACH))

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

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

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

#
# Define compilation macros.
#
COMPILE.c=      $(CC) $(CFLAGS) $(CPPFLAGS) -c
COMPILE64.c=    $(CC) $(CFLAGS64) $(CPPFLAGS) -c
COMPILE.cc=     $(CCC) $(CCFLAGS) $(CPPFLAGS) -c
COMPILE64.cc=   $(CCC) $(CCFLAGS64) $(CPPFLAGS) -c
COMPILE.s=      $(AS) $(ASFLAGS) $(AS_CPPFLAGS)
COMPILE64.s=    $(AS) $(ASFLAGS) $($(MACH64)_AS_XARCH) $(AS_CPPFLAGS)
COMPILE.d=      $(DTRACE) -G -32
COMPILE64.d=    $(DTRACE) -G -64
COMPILE.b=      $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))
COMPILE64.b=    $(ELFWRAP) $(ELFWRAP_FLAGS$(CLASS))

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

#
# Link time macros
#
CCNEEDED                = -lC
CCEXTNEEDED             = -lCrun -lCstd
$(__GNUC)CCNEEDED       = -L$(GCCLIBDIR) -lstdc++ -lgcc_s
$(__GNUC)CCEXTNEEDED    = $(CCNEEDED)
```

```
LINK.c=         $(CC) $(CFLAGS) $(CPPFLAGS) $(LDFLAGS)
LINK64.c=       $(CC) $(CFLAGS64) $(CPPFLAGS) $(LDFLAGS)
NORUNPATH=      -norunpath -nolib
LINK.cc=        $(CCC) $(CCFLAGS) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)
LINK64.cc=      $(CCC) $(CCFLAGS64) $(CPPFLAGS) $(NORUNPATH) \
                $(LDFLAGS) $(CCNEEDED)

#
# lint macros
#
# Note that the undefine of __PRAGMA_REDEFINE_EXTNAME can be removed once
# ON is built with a version of lint that has the fix for 4484186.
#
ALWAYS_LINT_DEFS =      -errtags=yes -s
ALWAYS_LINT_DEFS +=     -erroff=E_PTRDIFF_OVERFLOW
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_NARROW_CONV
ALWAYS_LINT_DEFS +=     -U__PRAGMA_REDEFINE_EXTNAME
ALWAYS_LINT_DEFS +=     $(C99LMODE)
ALWAYS_LINT_DEFS +=     -errsecurity=$(SECLEVEL)
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_CREAT_WITHOUT_EXCL
ALWAYS_LINT_DEFS +=     -erroff=E_SEC_FORBIDDEN_WARN_CREAT
# XX64 -- really only needed for amd64 lint
ALWAYS_LINT_DEFS +=     -erroff=E_ASSIGN_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_CONST_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_CAST_TO_PTR_FROM_INT
ALWAYS_LINT_DEFS +=     -erroff=E_COMP_INT_WITH_LARGE_INT
ALWAYS_LINT_DEFS +=     -erroff=E_INTEGRAL_CONST_EXP_EXPECTED
ALWAYS_LINT_DEFS +=     -erroff=E_PASS_INT_TO_SMALL_INT
ALWAYS_LINT_DEFS +=     -erroff=E_PTR_CONV_LOSES_BITS

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

SECLEVEL=       core
LINT.c=         $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT64.c=       $(LINT) $(ONLY_LINT_DEFS) $(LINTFLAGS64) $(CPPFLAGS) \
                $(ALWAYS_LINT_DEFS)
LINT.s=         $(LINT.c)

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

# Define native compilation macros
#

# Base directory where compilers are loaded.
# Defined here so it can be overridden by developer.
#
SPRO_ROOT=              $(BUILD_TOOLS)/SUNWspro
SPRO_VROOT=             $(SPRO_ROOT)/SS12
GNU_ROOT=               /usr

$(__GNUC)PRIMARY_CC=    gcc4,$(GNUC_ROOT)/bin/gcc,gnu
$(__SUNC)PRIMARY_CC=    studio12,$(SPRO_VROOT)/bin/cc,sun
$(__GNUC)PRIMARY_CCC=   gcc4,$(GNUC_ROOT)/bin/g++,gnu
$(__SUNC)PRIMARY_CCC=   studio12,$(SPRO_VROOT)/bin/CC,sun
```

```
CW_CC_COMPILERS=        $(PRIMARY_CC:%=--primary %) $(SHADOW_CCS:%=--shadow %)
CW_CCC_COMPILERS=       $(PRIMARY_CCC:%=--primary %) $(SHADOW_CCCS:%=--shadow %)


# Till SS12u1 formally becomes the NV CBE, LINT is hard
# coded to be picked up from the $SPRO_ROOT/sunstudio12.1/
# location. Impacted variables are sparc_LINT, sparcv9_LINT,
# i386_LINT, amd64_LINT.
# Reset them when SS12u1 is rolled out.
#

# Specify platform compiler versions for languages
# that we use (currently only c and c++).
#
CW=                     $(ONBLD_TOOLS)/bin/$(MACH)/cw

BUILD_CC=               $(CW) $(CW_CC_COMPILERS) --
BUILD_CCC=              $(CW) -C $(CW_CCC_COMPILERS) --
BUILD_CPP=              /usr/ccs/lib/cpp
BUILD_LD=               /usr/ccs/bin/ld
BUILD_LINT=             $(SPRO_ROOT)/sunstudio12.1/bin/lint

$(MACH)_CC=             $(BUILD_CC)
$(MACH)_CCC=            $(BUILD_CCC)
$(MACH)_CPP=            $(BUILD_CPP)
$(MACH)_LD=             $(BUILD_LD)
$(MACH)_LINT=           $(BUILD_LINT)
$(MACH64)_CC=           $(BUILD_CC)
$(MACH64)_CCC=          $(BUILD_CCC)
$(MACH64)_CPP=          $(BUILD_CPP)
$(MACH64)_LD=           $(BUILD_LD)
$(MACH64)_LINT=         $(BUILD_LINT)

sparc_AS=               /usr/ccs/bin/as -xregsym=no
sparcv9_AS=             $($(MACH)_AS)

i386_AS=                /usr/ccs/bin/as
$(__GNUC)i386_AS=       $(ONBLD_TOOLS)/bin/$(MACH)/aw
amd64_AS=               $(ONBLD_TOOLS)/bin/$(MACH)/aw

NATIVECC=               $($(NATIVE_MACH)_CC)
NATIVECCC=              $($(NATIVE_MACH)_CCC)
NATIVECPP=              $($(NATIVE_MACH)_CPP)
NATIVEAS=               $($(NATIVE_MACH)_AS)
NATIVELD=               $($(NATIVE_MACH)_LD)
NATIVELINT=             $($(NATIVE_MACH)_LINT)

#
# Makefile.master.64 overrides these settings
#
CC=                     $(NATIVECC)
CCC=                    $(NATIVECCC)
CPP=                    $(NATIVECPP)
AS=                     $(NATIVEAS)
LD=                     $(NATIVELD)
LINT=                   $(NATIVELINT)

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

BDIRECT=        -Bdirect
BDYNAMIC=       -Bdynamic
BLOCAL=         -Blocal
BNODIRECT=      -Bnodirect
BREDUCE=        -Breduce
BSTATIC=        -Bstatic
```

```
ZDEFS=          -zdefs
ZDIRECT=        -zdirect
ZIGNORE=        -zignore
ZINITFIRST=     -zinitfirst
ZINTERPOSE=     -zinterpose
ZLAZYLOAD=      -zlazyload
ZLOADFLTR=      -zloadfltr
ZMULDEFS=       -zmuldefs
ZNODEFAULTLIB=  -znodefaultlib
ZNODEFS=        -znodefs
ZNODELETE=      -znodelete
ZNODLOPEN=      -znodlopen
ZNODUMP=        -znodump
ZNOLAZYLOAD=    -znolazyload
ZNOLDYNSYM=     -znoldynsym
ZNORELOC=       -znoreloc
ZNOVERSION=     -znoversion
ZRECORD=        -zrecord
ZREDLOCSYM=     -zredlocsym
ZTEXT=          -ztext
ZVERBOSE=       -zverbose

GSHARED=        -G
CCMT=           -mt

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

sparc_C_PICFLAGS =      -fpic
sparcv9_C_PICFLAGS =    -fpic
i386_C_PICFLAGS =       -fpic
amd64_C_PICFLAGS =      -fpic
C_PICFLAGS =            $($(MACH)_C_PICFLAGS)
C_PICFLAGS64 =          $($(MACH64)_C_PICFLAGS)

sparc_C_BIGPICFLAGS =   -fPIC
sparcv9_C_BIGPICFLAGS = -fPIC
i386_C_BIGPICFLAGS =    -fPIC
amd64_C_BIGPICFLAGS =   -fPIC
C_BIGPICFLAGS =         $($(MACH)_C_BIGPICFLAGS)
C_BIGPICFLAGS64 =       $($(MACH64)_C_BIGPICFLAGS)

# CC requires there to be no space between '-K' and 'pic' or 'PIC'.
# and does not support -f
sparc_CC_PICFLAGS =     -_cc=-Kpic -_gcc=-fpic
sparcv9_CC_PICFLAGS =   -_cc=-KPIC -_gcc=-fPIC
i386_CC_PICFLAGS =      -_cc=-Kpic -_gcc=-fpic
amd64_CC_PICFLAGS =     -_cc=-Kpic -_gcc=-fpic
CC_PICFLAGS =           $($(MACH)_CC_PICFLAGS)
CC_PICFLAGS64 =         $($(MACH64)_CC_PICFLAGS)

AS_PICFLAGS=            -K pic
AS_BIGPICFLAGS=         -K PIC

#
# Default label for CTF sections
#
CTFCVTFLAGS=            -L VERSION
CTFCVTFLAGS=            -i -L VERSION

#
# Override to pass module-specific flags to ctfmerge.  Currently used only by
# krtld to turn on fuzzy matching, and source-level debugging to inhibit
# stripping.
#
```

```
CTFMRGFLAGS=

#
# Make the transition between old and new CTF Tools. The new ctf tools
# do not support stabs (eg. Sun Studio). By setting BUILD_OLD_CTF_TOOLS
# here or in the environment file, the old ones will be built.
#
BUILD_NEW_CTF_TOOLS=
BUILD_OLD_CTF_TOOLS=$(POUND_SIGN)
$(BUILD_OLD_CTF_TOOLS)BUILD_NEW_CTF_TOOLS= $(POUND_SIGN)

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

# Rules (normally from make.rules) and macros which are used for post
# processing files. Normally, these do stripping of the comment section
# automatically.
#    RELEASE_CM:         Should be edited to reflect the release.
#    POST_PROCESS_O:     Post-processing for '.o' files (typically C source)
#    POST_PROCESS_S_O:   Post-processing for '.o' files built from asssembly
#    POST_PROCESS_CC_O:  Post-processing for '.o' files built from C++
#    RELEASE_CM:         Should be editted to reflect the release.
#    POST_PROCESS_O:     Post-processing for '.o' files.
#    POST_PROCESS_A:             Post-processing for '.a' files (currently null).
#    POST_PROCESS_SO:    Post-processing for '.so' files.
#    POST_PROCESS:       Post-processing for executable files (no suffix).
#
# Note that these macros are not completely generalized as they are to be
# used with the file name to be processed following.
#
# It is left as an exercise to Release Engineering to embellish the generation
# of the release comment string.
#
#       If this is a standard development build:
#               compress the comment section (mcs -c)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an installation build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#               add the development specific comment (mcs -a $(DEV_CM))
#
#       If this is an release build:
#               delete the comment section (mcs -d)
#               add the standard comment (mcs -a $(RELEASE_CM))
#
# The following list of macros are used in the definition of RELEASE_CM
# which is used to label all binaries in the build:
#
#       RELEASE         Specific release of the build, eg: 5.2
#       RELEASE_MAJOR   Major version number part of $(RELEASE)
#       RELEASE_MINOR   Minor version number part of $(RELEASE)
#       VERSION         Version of the build (alpha, beta, Generic)
#       PATCHID         If this is a patch this value should contain
#                       the patchid value (eg: "Generic 100832-01"), otherwise
#                       it will be set to $(VERSION)
#       RELEASE_DATE    Date of the Release Build
#       PATCH_DATE      Date the patch was created, if this is blank it
#                       will default to the RELEASE_DATE
#
RELEASE_MAJOR=  5
RELEASE_MINOR=  11
RELEASE=        $(RELEASE_MAJOR).$(RELEASE_MINOR)
VERSION=        SunOS Development
PATCHID=        $(VERSION)
RELEASE_DATE=   release date not set
```

```
PATCH_DATE=     $(RELEASE_DATE)
RELEASE_CM=     "@($(POUND_SIGN))SunOS $(RELEASE) $(PATCHID) $(PATCH_DATE)"
DEV_CM=         "@($(POUND_SIGN))SunOS Internal Development: non-nightly build"

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

STRIP_STABS=                       $(STRIP) -x $@
$(SRCDBGBLD)STRIP_STABS=           :

POST_PROCESS_O=
POST_PROCESS_S_O=
POST_PROCESS_CC_O=
POST_PROCESS_A=
POST_PROCESS_SO=        $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)
POST_PROCESS=           $(PROCESS_COMMENT) $@ ; $(STRIP_STABS) ; \
                        $(ELFSIGN_OBJECT)

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

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

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

#       Default build rules which perform comment section post-processing.
#
.c:
        $(LINK.c) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.c.o:
        $(COMPILE.c) $(OUTPUT_OPTION) $< $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
.c.a:
        $(COMPILE.c) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.s.o:
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)
.s.a:
        $(COMPILE.s) -o $% $<
        $(PROCESS_COMMENT) $%
        $(AR) $(ARFLAGS) $@ $%
        $(RM) $%
.cc:
```

```
        $(LINK.cc) -o $@ $< $(LDLIBS)
        $(POST_PROCESS)
.cc.o:
        $(COMPILE.cc) $(OUTPUT_OPTION) $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)
.cc.a:
        $(COMPILE.cc) -o $% $<
        $(AR) $(ARFLAGS) $@ $%
        $(PROCESS_COMMENT) $%
        $(RM) $%
.y:
        $(YACC.y) $<
        $(LINK.c) -o $@ y.tab.c $(LDLIBS)
        $(POST_PROCESS)
        $(RM) y.tab.c
.y.o:
        $(YACC.y) $<
        $(COMPILE.c) -o $@ y.tab.c $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
        $(RM) y.tab.c
.l:
        $(RM) $*.c
        $(LEX.l) $< > $*.c
        $(LINK.c) -o $@ $*.c -ll $(LDLIBS)
        $(POST_PROCESS)
        $(RM) $*.c
.l.o:
        $(RM) $*.c
        $(LEX.l) $< > $*.c
        $(COMPILE.c) -o $@ $*.c $(CTFCONVERT_HOOK)
        $(POST_PROCESS_O)
        $(RM) $*.c

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

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

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

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

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

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

.py:
	$(RM) $@; $(SED) \
		-e "1s:^\#!@PYTHON@:\#!$(PYSHEBANG):" \
		-e "1s:^\#!@TOOLS_PYTHON@:\#!$(TOOLS_PYTHON):" \
		< $< > $@; $(CHMOD) +x $@

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

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

.pl.po .pm.po:
	$(XGETTEXT) $(XGETFLAGS) -d $(<F) $< ;
	$(RM)   $@ ;
	$(SED) "/^domain/d" < $(<F).po > $@ ;
	$(RM) $(<F).po

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

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

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

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

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

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

.cc.po:
	$(CPPFORPO) $< > $<.i
	$(BUILD.po)

.y.po:
	$(YACC) -d $<
	$(CPPFORPO) y.tab.c  > $<.i
	$(BUILD.po)
	$(RM) y.tab.c

.l.po:
	$(LEX) $<
	$(CPPFORPO) lex.yy.c  > $<.i
	$(BUILD.po)
	$(RM) lex.yy.c

#
# Rules to perform stylistic checks
#
```

```
.SUFFIXES: .x .xml .check .xmlchk

.h.check:
	$(DOT_H_CHECK)

.x.check:
	$(DOT_X_CHECK)

.xml.xmlchk:
	$(MANIFEST_CHECK)

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

```
**********************************************************
   10474 Thu Apr 25 02:20:39 2019
new/usr/src/cmd/ctfconvert/ctfconvert.c
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright (c) 2019, Joyent, Inc.
 * Copyright (c) 2015, Joyent, Inc.
 */

/*
 * Create CTF from extant debugging information
 */

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <errno.h>
#include <libelf.h>
#include <libctf.h>
#include <string.h>
#include <libgen.h>
#include <limits.h>
#include <strings.h>
#include <sys/debug.h>

#define CTFCONVERT_OK           0
#define CTFCONVERT_FATAL        1
#define CTFCONVERT_USAGE        2

#define CTFCONVERT_DEFAULT_NTHREADS     4

#define CTFCONVERT_ALTEXEC      "CTFCONVERT_ALTEXEC"

static char *ctfconvert_progname;

static void
ctfconvert_fatal(const char *fmt, ...)
{
        va_list ap;

        (void) fprintf(stderr, "%s: ", ctfconvert_progname);
        va_start(ap, fmt);
```

```
        (void) vfprintf(stderr, fmt, ap);
        va_end(ap);

        exit(CTFCONVERT_FATAL);
}


static void
ctfconvert_usage(const char *fmt, ...)
{
        if (fmt != NULL) {
                va_list ap;

                (void) fprintf(stderr, "%s: ", ctfconvert_progname);
                va_start(ap, fmt);
                (void) vfprintf(stderr, fmt, ap);
                va_end(ap);
        }

        (void) fprintf(stderr, "Usage: %s [-ims] [-j nthrs] [-l label | "
        (void) fprintf(stderr, "Usage: %s [-is] [-j nthrs] [-l label | "
            "-L labelenv] [-o outfile] input\n"
            "\n"
            "\t-i  ignore files not built partially from C sources\n"
            "\t-j  use nthrs threads to perform the merge\n"
            "\t-k  keep around original input file on failure\n"
            "\t-m  allow input to have missing debug info\n"
            "\t-o  copy input to outfile and add CTF\n"
            "\t-l  set output container's label to specified value\n"
            "\t-L  set output container's label to value from environment\n",
            ctfconvert_progname);
}
______unchanged_portion_omitted_

int
main(int argc, char *argv[])
{
        int c, ifd, err;
        boolean_t keep = B_FALSE;
        uint_t flags = 0;
        uint_t nthreads = CTFCONVERT_DEFAULT_NTHREADS;
        const char *outfile = NULL;
        const char *label = NULL;
        const char *infile = NULL;
        char *tmpfile;
        ctf_file_t *ofp;
        long argj;
        char *eptr;
        char buf[4096];
        boolean_t optx = B_FALSE;
        boolean_t ignore_non_c = B_FALSE;

        ctfconvert_progname = basename(argv[0]);

        ctfconvert_altexec(argv);

        while ((c = getopt(argc, argv, ":ij:kl:L:mo:X")) != -1) {
        while ((c = getopt(argc, argv, ":j:kl:L:o:iX")) != -1) {
                switch (c) {
                case 'i':
                        ignore_non_c = B_TRUE;
                case 'k':
                        keep = B_TRUE;
                        break;
                case 'l':
                        label = optarg;
                        break;
                case ’L’:
                        label = getenv(optarg);
                        break;
                case ’j’:
                        errno = 0;
                        argj = strtol(optarg, &eptr, 10);
                        if (errno != 0 || argj == LONG_MAX ||
                            argj > 1024 || *eptr != ’\0’) {
                                ctfconvert_fatal("invalid argument for -j: "
                                    "%s\n", optarg);
                        }
                        nthreads = (uint_t)argj;
                        break;
                case ’k’:
                        keep = B_TRUE;
                        break;
                case ’l’:
                        label = optarg;
                        break;
                case ’L’:
                        label = getenv(optarg);
                        break;
                case ’m’:
                        flags |= CTF_ALLOW_MISSING_DEBUG;
                        break;
                case ’o’:
                        outfile = optarg;
                        break;
                case ’i’:
                        flags |= CTF_CONVERT_F_IGNNONC;
                        break;
                case ’X’:
                        optx = B_TRUE;
                        break;
                case ’:’:
                        ctfconvert_usage("Option -%c requires an operand\n",
                            optopt);
                        return (CTFCONVERT_USAGE);
                case ’?’:
                        ctfconvert_usage("Unknown option: -%c\n", optopt);
                        return (CTFCONVERT_USAGE);
                }
        }

        argv += optind;
        argc -= optind;

        if (argc != 1) {
                ctfconvert_usage("Exactly one input file is required\n");
        if (argc < 1) {
                ctfconvert_usage("Missing required input file\n");
                return (CTFCONVERT_USAGE);
        }
        infile = argv[0];

        if (elf_version(EV_CURRENT) == EV_NONE)
                ctfconvert_fatal("failed to initialize libelf: library is "
                    "out of date\n");

        ifd = open(infile, O_RDONLY);
        if (ifd < 0) {
                ctfconvert_fatal("failed to open input file %s: %s\n", infile,
                    strerror(errno));
        }
```

```
        /*
         * By default we remove the input file on failure unless we’ve been
         * given an output file or -k has been specified.
         */
        if (outfile != NULL && strcmp(infile, outfile) != 0)
                keep = B_TRUE;

        ofp = ctf_fdconvert(ifd, label, nthreads, flags, &err, buf,
            sizeof (buf));
        if (ofp == NULL) {
                /*
                 * Normally, ctfconvert requires that its input file has at
                 * least one C-source compilation unit, and that every C-source
                 * compilation unit has DWARF. This is to avoid accidentally
                 * leaving out useful CTF.
                 *
                 * However, for the benefit of intransigent build environments,
                 * the -i and -m options can be used to relax this.
                 * -i says that we shouldn’t concern ourselves with source files
                 * that weren’t built from C source code in part. Because this
                 * has been traditionally used across all of illumos, we still
                 * honor it.
                 */
                if (err == ECTF_CONVNOCSRC && ignore_non_c) {
                if ((flags & CTF_CONVERT_F_IGNNONC) != 0 &&
                    err == ECTF_CONVNOCSRC) {
                        exit(CTFCONVERT_OK);
                }

                if (err == ECTF_CONVNODEBUG &&
                    (flags & CTF_ALLOW_MISSING_DEBUG) != 0) {
                        exit(CTFCONVERT_OK);
                }

                if (keep == B_FALSE)
                        (void) unlink(infile);

                if (err == ECTF_CONVBKERR || err == ECTF_CONVNODEBUG) {
                        ctfconvert_fatal("%s", buf);
                } else {
                        ctfconvert_fatal("CTF conversion failed: %s\n",
                            ctf_errmsg(err));
                    err == ECTF_CONVBKERR ? buf : ctf_errmsg(err));
                }
        }

        if (optx == B_TRUE)
                ctfconvert_fixup_genunix(ofp);

        tmpfile = NULL;
        if (outfile == NULL || strcmp(infile, outfile) == 0) {
                if (asprintf(&tmpfile, "%s.ctf", infile) == -1) {
                        if (keep == B_FALSE)
                                (void) unlink(infile);
                        ctfconvert_fatal("failed to allocate memory for "
                            "temporary file: %s\n", strerror(errno));
                }
                outfile = tmpfile;
        }
        err = ctf_elfwrite(ofp, infile, outfile, CTF_ELFWRITE_F_COMPRESS);
        if (err == CTF_ERR) {
                (void) unlink(outfile);
                if (keep == B_FALSE)
                        (void) unlink(infile);
                ctfconvert_fatal("failed to write CTF section to output file: "
                    "%s", ctf_errmsg(ctf_errno(ofp)));
        }
        ctf_close(ofp);

        if (tmpfile != NULL) {
                if (rename(tmpfile, infile) != 0) {
                        int e = errno;
                        (void) unlink(outfile);
                        if (keep == B_FALSE)
                                (void) unlink(infile);
                        ctfconvert_fatal("failed to rename temporary file: "
                            "%s\n", strerror(e));
                }
        }
        free(tmpfile);

        return (CTFCONVERT_OK);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
   11748 Thu Apr 25 02:20:39 2019
new/usr/src/cmd/ctfmerge/ctfmerge.c
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright (c) 2019, Joyent, Inc.
 * Copyright (c) 2017, Joyent, Inc.
 */

/*
 * merge CTF containers
 */

#include <stdio.h>
#include <libctf.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>
#include <errno.h>
#include <strings.h>
#include <assert.h>
#include <unistd.h>
#include <sys/fcntl.h>
#include <stdlib.h>
#include <libelf.h>
#include <gelf.h>
#include <sys/mman.h>
#include <libgen.h>
#include <stdarg.h>
#include <limits.h>

static char *g_progname;
static char *g_unique;
static char *g_outfile;
static boolean_t g_req;
static uint_t g_nctf;

#define CTFMERGE_OK     0
#define CTFMERGE_FATAL  1
#define CTFMERGE_USAGE  2

#define CTFMERGE_DEFAULT_NTHREADS       8
#define CTFMERGE_ALTEXEC        "CTFMERGE_ALTEXEC"

static void
ctfmerge_fatal(const char *fmt, ...)
```

```
{
        va_list ap;

        (void) fprintf(stderr, "%s: ", g_progname);
        va_start(ap, fmt);
        (void) vfprintf(stderr, fmt, ap);
        va_end(ap);

        if (g_outfile != NULL)
                (void) unlink(g_outfile);

        exit(CTFMERGE_FATAL);
}

/*
 * We failed to find CTF for this file, check if it’s OK. If we’re not derived
 * from C, or we have the -m option, we let missing CTF pass.
 */
static void
ctfmerge_check_for_c(const char *name, Elf *elf, uint_t flags)
static boolean_t
ctfmerge_expect_ctf(const char *name, Elf *elf)
{
        char errmsg[1024];
        Elf_Scn *scn, *strscn;
        Elf_Data *data, *strdata;
        GElf_Shdr shdr;
        ulong_t i;

        if (flags & CTF_ALLOW_MISSING_DEBUG)
                return;
        if (g_req == B_FALSE)
                return (B_FALSE);

        switch (ctf_has_c_source(elf, errmsg, sizeof (errmsg))) {
        case CHR_ERROR:
                ctfmerge_fatal("failed to open %s: %s\n", name, errmsg);
        scn = NULL;
        while ((scn = elf_nextscn(elf, scn)) != NULL) {
                if (gelf_getshdr(scn, &shdr) == NULL) {
                        ctfmerge_fatal("failed to get section header for file "
                            "%s: %s\n", name, elf_errmsg(elf_errno()));
                }

                if (shdr.sh_type == SHT_SYMTAB)
                break;
        }

        case CHR_NO_C_SOURCE:
                return;
        if (scn == NULL)
                return (B_FALSE);

        default:
                ctfmerge_fatal("failed to open %s: %s\n", name,
                    ctf_errmsg(ECTF_NOCTFDATA));
                break;
        if ((strscn = elf_getscn(elf, shdr.sh_link)) == NULL)
                ctfmerge_fatal("failed to get section header for file %s: %s\n",
                    name, elf_errmsg(elf_errno()));

        if ((data = elf_getdata(scn, NULL)) == NULL)
                ctfmerge_fatal("failed to read symbol table for %s: %s\n",
                    name, elf_errmsg(elf_errno()));

        if ((strdata = elf_getdata(strscn, NULL)) == NULL)
```

```
                ctfmerge_fatal("failed to read string table for %s: %s\n",
                    name, elf_errmsg(elf_errno()));

        for (i = 0; i < shdr.sh_size / shdr.sh_entsize; i++) {
                GElf_Sym sym;
                const char *file;
                size_t len;

                if (gelf_getsym(data, i, &sym) == NULL)
                        ctfmerge_fatal("failed to read symbol table entry %lu "
                            "for %s: %s\n", i, name, elf_errmsg(elf_errno()));

                if (GELF_ST_TYPE(sym.st_info) != STT_FILE)
                        continue;

                file = (const char *)((uintptr_t)strdata->d_buf + sym.st_name);
                len = strlen(file);
                if (len < 2 || name[len - 2] != '.')
                        continue;

                if (name[len - 1] == 'c')
                        return (B_TRUE);
        }

        return (B_FALSE);
}

/*
 * Go through and construct enough information for this Elf Object to try and do
 * a ctf_bufopen().
 */
static int
ctfmerge_elfopen(const char *name, Elf *elf, ctf_merge_t *cmh, uint_t flags)
static void
ctfmerge_elfopen(const char *name, Elf *elf, ctf_merge_t *cmh)
{
        GElf_Ehdr ehdr;
        GElf_Shdr shdr;
        Elf_Scn *scn;
        Elf_Data *ctf_data, *str_data, *sym_data;
        ctf_sect_t ctfsect, symsect, strsect;
        ctf_file_t *fp;
        int err;

        if (gelf_getehdr(elf, &ehdr) == NULL)
                ctfmerge_fatal("failed to get ELF header for %s: %s\n",
                    name, elf_errmsg(elf_errno()));

        bzero(&ctfsect, sizeof (ctf_sect_t));
        bzero(&symsect, sizeof (ctf_sect_t));
        bzero(&strsect, sizeof (ctf_sect_t));

        scn = NULL;
        while ((scn = elf_nextscn(elf, scn)) != NULL) {
                const char *sname;

                if (gelf_getshdr(scn, &shdr) == NULL)
                        ctfmerge_fatal("failed to get section header for "
                            "file %s: %s\n", name, elf_errmsg(elf_errno()));

                sname = elf_strptr(elf, ehdr.e_shstrndx, shdr.sh_name);
                if (shdr.sh_type == SHT_PROGBITS &&
                    strcmp(sname, ".SUNW_ctf") == 0) {
                        ctfsect.cts_name = sname;
                        ctfsect.cts_type = shdr.sh_type;
                        ctfsect.cts_flags = shdr.sh_flags;
```

```
                        ctfsect.cts_size = shdr.sh_size;
                        ctfsect.cts_entsize = shdr.sh_entsize;
                        ctfsect.cts_offset = (off64_t)shdr.sh_offset;

                        ctf_data = elf_getdata(scn, NULL);
                        if (ctf_data == NULL)
                                ctfmerge_fatal("failed to get ELF CTF "
                                    "data section for %s: %s\n", name,
                                    elf_errmsg(elf_errno()));
                        ctfsect.cts_data = ctf_data->d_buf;
                } else if (shdr.sh_type == SHT_SYMTAB) {
                        Elf_Scn *strscn;
                        GElf_Shdr strhdr;

                        symsect.cts_name = sname;
                        symsect.cts_type = shdr.sh_type;
                        symsect.cts_flags = shdr.sh_flags;
                        symsect.cts_size = shdr.sh_size;
                        symsect.cts_entsize = shdr.sh_entsize;
                        symsect.cts_offset = (off64_t)shdr.sh_offset;

                        if ((strscn = elf_getscn(elf, shdr.sh_link)) == NULL ||
                            gelf_getshdr(strscn, &strhdr) == NULL)
                                ctfmerge_fatal("failed to get "
                                    "string table for file %s: %s\n", name,
                                    elf_errmsg(elf_errno()));

                        strsect.cts_name = elf_strptr(elf, ehdr.e_shstrndx,
                            strhdr.sh_name);
                        strsect.cts_type = strhdr.sh_type;
                        strsect.cts_flags = strhdr.sh_flags;
                        strsect.cts_size = strhdr.sh_size;
                        strsect.cts_entsize = strhdr.sh_entsize;
                        strsect.cts_offset = (off64_t)strhdr.sh_offset;

                        sym_data = elf_getdata(scn, NULL);
                        if (sym_data == NULL)
                                ctfmerge_fatal("failed to get ELF CTF "
                                    "data section for %s: %s\n", name,
                                    elf_errmsg(elf_errno()));
                        symsect.cts_data = sym_data->d_buf;

                        str_data = elf_getdata(strscn, NULL);
                        if (str_data == NULL)
                                ctfmerge_fatal("failed to get ELF CTF "
                                    "data section for %s: %s\n", name,
                                    elf_errmsg(elf_errno()));
                        strsect.cts_data = str_data->d_buf;
                }
        }

        if (ctfsect.cts_type == SHT_NULL) {
                ctfmerge_check_for_c(name, elf, flags);
                return (ENOENT);
                if (ctfmerge_expect_ctf(name, elf) == B_FALSE)
                        return;
                ctfmerge_fatal("failed to open %s: %s\n", name,
                    ctf_errmsg(ECTF_NOCTFDATA));
        }

        if (symsect.cts_type != SHT_NULL && strsect.cts_type != SHT_NULL) {
                fp = ctf_bufopen(&ctfsect, &symsect, &strsect, &err);
        } else {
                fp = ctf_bufopen(&ctfsect, NULL, NULL, &err);
        }
        if (fp == NULL) {
                if (ctfmerge_expect_ctf(name, elf) == B_TRUE) {
                ctfmerge_fatal("failed to open file %s: %s\n",
                    name, ctf_errmsg(err));
        }

        } else {
        if ((err = ctf_merge_add(cmh, fp)) != 0) {
                ctfmerge_fatal("failed to add input %s: %s\n",
                    name, ctf_errmsg(err));
        }

        g_nctf++;
        return (0);
        }
}

static void
ctfmerge_read_archive(const char *name, int fd, Elf *elf,
    ctf_merge_t *cmh, uint_t flags)
    ctf_merge_t *cmh)
{
        Elf *aelf;
        Elf_Cmd cmd = ELF_C_READ;
        int cursec = 1;
        Elf *aelf;
        char *nname;

        while ((aelf = elf_begin(fd, cmd, elf)) != NULL) {
                char *nname = NULL;
                Elf_Arhdr *arhdr;
                boolean_t leakelf = B_FALSE;

                if ((arhdr = elf_getarhdr(aelf)) == NULL)
                        ctfmerge_fatal("failed to get archive header %d for "
                            "%s: %s\n", cursec, name, elf_errmsg(elf_errno()));

                cmd = elf_next(aelf);

                if (*(arhdr->ar_name) == '/')
                        goto next;

                if (asprintf(&nname, "%s.%s.%d", name, arhdr->ar_name,
                    cursec) < 0)
                        ctfmerge_fatal("failed to allocate memory for archive "
                            "%d of file %s\n", cursec, name);

                switch (elf_kind(aelf)) {
                case ELF_K_AR:
                        ctfmerge_read_archive(nname, fd, aelf, cmh, flags);
                        ctfmerge_read_archive(nname, fd, aelf, cmh);
                        free(nname);
                        break;
                case ELF_K_ELF:
                        /* ctfmerge_elfopen() takes ownership of aelf. */
                        if (ctfmerge_elfopen(nname, aelf, cmh, flags) == 0)
                                aelf = NULL;
                        ctfmerge_elfopen(nname, aelf, cmh);
                        free(nname);
                        leakelf = B_TRUE;
                        break;
                default:
                        ctfmerge_fatal("unknown elf kind (%d) in archive %d "
                            "for %s\n", elf_kind(aelf), cursec, name);
                        break;
                }
```

```
next:
                cmd = elf_next(aelf);
                if (leakelf == B_FALSE)
                (void) elf_end(aelf);
                free(nname);
                cursec++;
        }
}

static void
ctfmerge_file_add(ctf_merge_t *cmh, const char *file, uint_t flags)
{
        Elf *e;
        int fd;

        if ((fd = open(file, O_RDONLY)) < 0) {
                ctfmerge_fatal("failed to open file %s: %s\n",
                    file, strerror(errno));
        }

        if ((e = elf_begin(fd, ELF_C_READ, NULL)) == NULL) {
                (void) close(fd);
                ctfmerge_fatal("failed to open %s: %s\n",
                    file, elf_errmsg(elf_errno()));
        }

        switch (elf_kind(e)) {
        case ELF_K_AR:
                ctfmerge_read_archive(file, fd, e, cmh, flags);
                break;

        case ELF_K_ELF:
                /* ctfmerge_elfopen() takes ownership of e. */
                if (ctfmerge_elfopen(file, e, cmh, flags) == 0)
                        e = NULL;
                break;

        default:
                ctfmerge_fatal("unknown elf kind (%d) for %s\n",
                    elf_kind(e), file);
        }

        (void) elf_end(e);
        (void) close(fd);
}

static void
ctfmerge_usage(const char *fmt, ...)
{
        if (fmt != NULL) {
                va_list ap;

                (void) fprintf(stderr, "%s: ", g_progname);
                va_start(ap, fmt);
                (void) vfprintf(stderr, fmt, ap);
                va_end(ap);
        }

        (void) fprintf(stderr, "Usage: %s [-m] [-d uniqfile] [-l label] "
        (void) fprintf(stderr, "Usage: %s [-t] [-d uniqfile] [-l label] "
            "[-L labelenv] [-j nthrs] -o outfile file ...\n"
            "\n"
            "\t-d  uniquify merged output against uniqfile\n"
            "\t-j  use nthrs threads to perform the merge\n"
            "\t-l  set output container's label to specified value\n"
            "\t-L  set output container's label to value from environment\n"
            "\t-m  allow C-based input files to not have CTF\n"
            "\t-o  file to add CTF data to\n",
            "\t-o  file to add CTF data to\n"
            "\t-t  require CTF data from all inputs built from C sources\n",
            g_progname);
}
______unchanged_portion_omitted_

int
main(int argc, char *argv[])
{
        int err, i, c, ofd;
        uint_t nthreads = CTFMERGE_DEFAULT_NTHREADS;
        char *tmpfile = NULL, *label = NULL;
        int wflags = CTF_ELFWRITE_F_COMPRESS;
        uint_t flags = 0;
        ctf_merge_t *cmh;
        ctf_file_t *ofp;
        ctf_merge_t *cmh;
        long argj;
        char *eptr;

        g_progname = basename(argv[0]);

        ctfmerge_altexec(argv);

        /*
         * We support a subset of the old CTF merge flags, mostly for
         * compatibility.
         * compatability.
         */
        while ((c = getopt(argc, argv, ":d:fgj:l:L:mo:t")) != -1) {
        while ((c = getopt(argc, argv, ":d:fgj:l:L:o:t")) != -1) {
                switch (c) {
                case 'd':
                        g_unique = optarg;
                        break;
                case 'f':
                        /* Silently ignored for compatibility */
                        break;
                case 'g':
                        /* Silently ignored for compatibility */
                        break;
                case 'j':
                        errno = 0;
                        argj = strtol(optarg, &eptr, 10);
                        if (errno != 0 || argj == LONG_MAX ||
                            argj > 1024 || *eptr != '\0') {
                                ctfmerge_fatal("invalid argument for -j: %s\n",
                                    optarg);
                        }
                        nthreads = (uint_t)argj;
                        break;
                case 'l':
                        label = optarg;
                        break;
                case 'L':
                        label = getenv(optarg);
                        break;
                case 'm':
                        flags |= CTF_ALLOW_MISSING_DEBUG;
                        break;
                case 'o':
                        g_outfile = optarg;
                        break;
```

```
                case 't':
                        /* Silently ignored for compatibility */
                        g_req = B_TRUE;
                        break;
                case ':':
                        ctfmerge_usage("Option -%c requires an operand\n",
                            optopt);
                        return (CTFMERGE_USAGE);
                case '?':
                        ctfmerge_usage("Unknown option: -%c\n", optopt);
                        return (CTFMERGE_USAGE);
                }
        }

        if (g_outfile == NULL) {
                ctfmerge_usage("missing required -o output file\n");
                return (CTFMERGE_USAGE);
        }

        (void) elf_version(EV_CURRENT);

        /*
         * Obviously this isn't atomic, but at least gives us a good starting
         * point.
         */
        if ((ofd = open(g_outfile, O_RDWR)) < 0)
                ctfmerge_fatal("cannot open output file %s: %s\n", g_outfile,
                    strerror(errno));

        argc -= optind;
        argv += optind;

        if (argc < 1) {
                ctfmerge_usage("no input files specified");
                return (CTFMERGE_USAGE);
        }

        cmh = ctf_merge_init(ofd, &err);
        if (cmh == NULL)
                ctfmerge_fatal("failed to create merge handle: %s\n",
                    ctf_errmsg(err));

        if ((err = ctf_merge_set_nthreads(cmh, nthreads)) != 0)
                ctfmerge_fatal("failed to set parallelism to %u: %s\n",
                    nthreads, ctf_errmsg(err));

        for (i = 0; i < argc; i++) {
                ctfmerge_file_add(cmh, argv[i], flags);
                ctf_file_t *ifp;
                int fd;

                if ((fd = open(argv[i], O_RDONLY)) < 0)
                        ctfmerge_fatal("failed to open file %s: %s\n",
                            argv[i], strerror(errno));
                ifp = ctf_fdopen(fd, &err);
                if (ifp == NULL) {
                        Elf *e;

                        if ((e = elf_begin(fd, ELF_C_READ, NULL)) == NULL) {
                                (void) close(fd);
                                ctfmerge_fatal("failed to open %s: %s\n",
                                    argv[i], ctf_errmsg(err));
        }

                        /*
                         * It's an ELF file, check if we have an archive or if
                         * we’re expecting CTF here.
                         */
                        switch (elf_kind(e)) {
                        case ELF_K_AR:
                                break;
                        case ELF_K_ELF:
                                if (ctfmerge_expect_ctf(argv[i], e) == B_TRUE) {
                                        (void) elf_end(e);
                                        (void) close(fd);
                                        ctfmerge_fatal("failed to "
                                            "open %s: file was built from C "
                                            "sources, but missing CTF\n",
                                            argv[i]);
                                }
                                (void) elf_end(e);
                                (void) close(fd);
                                continue;
                        default:
                                (void) elf_end(e);
                                (void) close(fd);
                                ctfmerge_fatal("failed to open %s: "
                                    "unsupported ELF file type", argv[i]);
                        }

                        ctfmerge_read_archive(argv[i], fd, e, cmh);
                        (void) elf_end(e);
                        (void) close(fd);
                        continue;
                }
                (void) close(fd);
                if ((err = ctf_merge_add(cmh, ifp)) != 0)
                        ctfmerge_fatal("failed to add input %s: %s\n",
                            argv[i], ctf_errmsg(err));
                g_nctf++;
        }

        if (g_nctf == 0) {
                ctf_merge_fini(cmh);
                return (0);
        }

        if (g_unique != NULL) {
                ctf_file_t *ufp;
                char *base;

                ufp = ctf_open(g_unique, &err);
                if (ufp == NULL) {
                        ctfmerge_fatal("failed to open uniquify file %s: %s\n",
                            g_unique, ctf_errmsg(err));
                }

                base = basename(g_unique);
                (void) ctf_merge_uniquify(cmh, ufp, base);
        }

        if (label != NULL) {
                if ((err = ctf_merge_label(cmh, label)) != 0)
                        ctfmerge_fatal("failed to add label %s: %s\n", label,
                            ctf_errmsg(err));
        }

        err = ctf_merge_merge(cmh, &ofp);
        if (err != 0)
                ctfmerge_fatal("failed to merge types: %s\n", ctf_errmsg(err));
        ctf_merge_fini(cmh);
```

```
        if (asprintf(&tmpfile, "%s.ctf", g_outfile) == -1)
                ctfmerge_fatal("ran out of memory for temporary file name\n");
        err = ctf_elfwrite(ofp, g_outfile, tmpfile, wflags);
        if (err == CTF_ERR) {
                (void) unlink(tmpfile);
                free(tmpfile);
                ctfmerge_fatal("encountered a libctf error: %s!\n",
                    ctf_errmsg(ctf_errno(ofp)));
        }

        if (rename(tmpfile, g_outfile) != 0) {
                (void) unlink(tmpfile);
                free(tmpfile);
                ctfmerge_fatal("failed to rename temporary file: %s\n",
                    strerror(errno));
        }
        free(tmpfile);

        return (CTFMERGE_OK);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    8770 Thu Apr 25 02:20:39 2019
new/usr/src/cmd/mdb/Makefile.module
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2013 by Delphix. All rights reserved.
# Copyright (c) 2018, Joyent, Inc.
# Copyright (c) 2019 Carlos Neira <cneirabustos@gmail.com>
# Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
# Copyright (c) 2019, Joyent, Inc.
#

.KEEP_STATE:
.SUFFIXES:

include $(SRC)/cmd/mdb/Makefile.tools

#
# Make sure we're getting a consistent execution environment for the
# embedded scripts.
#
SHELL= /usr/bin/ksh93

$(KMOD_SOURCES_DIFFERENT)KMODSRCS = $(MODSRCS)
$(KMOD_SOURCES_DIFFERENT)KMODASMSRCS = $(MODASMSRCS)

MODOBJS = $(MODSRCS:%.c=dmod/%.o) $(MODASMSRCS:%.s=dmod/%.o)
KMODOBJS = $(KMODSRCS:%.c=kmod/%.o) $(KMODASMSRCS:%.s=kmod/%.o)

MODNAME_cmd = if [ -n "$(MODULE_NAME)" ]; then print $(MODULE_NAME); else print
MODNAME = $(MODNAME_cmd:sh)
KMODULE = $(MODNAME)

MODFILE = dmod/$(MODULE)
```

```
KMODFILE = kmod/$(KMODULE)

#
# The mess below is designed to pick the right set of objects to build and/or
# lint.  We have three flavors:
#
#  1. proc and raw modules.  Only $(MODOBJS) are built.
#  2. kvm modules for systems without kmdb.  Only $(MODOBJS) are built.
#  3. kvm modules for systems with kmdb.  $(MODOBJS) and $(KMODOBJS) are built.
#
# Complicating matters, we'd like to make the distinction between 2 and 3 before
# this Makefile is loaded.  By default, we'll assume that all kvm modules should
# be built for kmdb.  If, however, the user sets $(MODULE_BUILD_TYPE) to 'mdb',
# the kmdb variant of the module won't be built.
#

# Which flavors are to be built?
TARGETS_kvm_type_       = both  # Build both if $(MODULE_BUILD_TYPE) is unset
TARGETS_kvm_type_kmdb   = both
TARGETS_kvm_type_mdb    = mdb
TARGETS_kvm_type        = $(TARGETS_kvm_type_$(MODULE_BUILD_TYPE))

# What should we build?
TARGETS_kvm_kmdb        = $(KMODFILE)
TARGETS_kvm_mdb         = $(MODFILE)
TARGETS_kvm_both        = $(TARGETS_kvm_mdb) $(TARGETS_kvm_kmdb)
TARGETS_kvm             = $(TARGETS_kvm_$(TARGETS_kvm_type))
TARGETS_proc            = $(MODFILE)
TARGETS_raw             = $(MODFILE)
TARGETS                 = $(TARGETS_$(MDBTGT))

# Where should we install that which we've built?
ROOTTGTS_kvm_type       = $(TARGETS_kvm_type)   # overridden by mdb_ks
ROOTTGTS_kvm_kmdb       = $(ROOTKMOD)/$(KMODULE)
ROOTTGTS_kvm_mdb        = $(ROOTMOD)/$(MODULE)
ROOTTGTS_kvm_both       = $(ROOTTGTS_kvm_mdb) $(ROOTTGTS_kvm_kmdb)
ROOTTGTS_kvm            = $(ROOTTGTS_kvm_$(ROOTTGTS_kvm_type))
ROOTTGTS_proc           = $(ROOTMOD)/$(MODULE)
ROOTTGTS_raw            = $(ROOTMOD)/$(MODULE)
ROOTTGTS                = $(ROOTTGTS_$(MDBTGT))

# What should we lint?
KLINTOBJS               = $(KMODOBJS:%.o=%.ln)
LINTOBJS                = $(MODOBJS:%.o=%.ln)

LINTFILES_kvm_type      = $(TARGETS_kvm_type)
LINTFILES_kvm_both      = $(KLINTOBJS) $(LINTOBJS)
LINTFILES_kvm_mdb       = $(LINTOBJS)
LINTFILES_kvm           = $(LINTFILES_kvm_$(LINTFILES_kvm_type))
LINTFILES_proc          = $(LINTOBJS)
LINTFILES_raw           = $(LINTOBJS)
LINTFILES               = $(LINTFILES_$(MDBTGT))

#
# Python specific flags. To try and make life easier for folks how are
# building with an LFS python, we attempt to use -isystem when it's
# available.
#
PYCPPFLAGS              = -_gcc=-isystem -_gcc=$(ADJUNCT_PROTO)/usr/include/pyth
PYCPPFLAGS              += -_cc=-I$(ADJUNCT_PROTO)/usr/include/python$(PYTHON_VE
PYLNFLAGS               = -I$(ADJUNCT_PROTO)/usr/include/python$(PYTHON_VERSION)

kvm_TGTFLAGS            = -D_KERNEL
proc_TGTFLAGS           = -D_USER

CSTD                    = $(CSTD_GNU99)
```

```
CFLAGS                  += $(CCVERBOSE)
CFLAGS64                += $(CCVERBOSE)
CPPFLAGS                += $($(MDBTGT)_TGTFLAGS) -I../../../common
LDFLAGS                 += $(ZTEXT)
LDFLAGS64               += $(ZTEXT)
ASFLAGS                 += -P
AS_CPPFLAGS             += -D_ASM

SMOFF += all_func_returns,index_overflow

# Module type-specific compiler flags
$(MODOBJS) :=                   CFLAGS += $(C_BIGPICFLAGS) $(XREGSFLAG)
$(MODOBJS) :=                   CFLAGS64 += $(C_BIGPICFLAGS) $(XREGSFLAG)
$(KMODOBJS) $(KLINTOBJS) :=     CPPFLAGS += -D_KMDB
$(KMODOBJS) :=                  V9CODESIZE = $(CCABS32)
$(KMODOBJS) :=                  DTS_ERRNO =

# Modules aren’t allowed to export symbols
MAPFILE         = $(SRC)/cmd/mdb/common/modules/conf/mapfile

# Modules typically make external references.  To provide for -zdefs use
# and clean ldd(1) processing, explicitly define all external references.
MAPFILE-EXT     = $(SRC)/cmd/mdb/common/modules/conf/mapfile-extern

#
# kmdb is a kernel module, so we’ll use the kernel’s build flags.
$(KMODOBJS) := CFLAGS64 += $(STAND_FLAGS_64)

#
# Override this to pull source files from another directory
#
MODSRCS_DIR = ../../../common/modules/genunix

all: $$(TARGETS)

install: all $$(ROOTTGTS)

dmods: install

clean.lint:
        $(RM) $(LINTFILES) $(MODSRCS:.c=.ln)

clean:
        $(RM) $(MODOBJS) $(KMODOBJS) $(CLEANFILES)

clobber: clean clean.lint
        $(RM) $(MODFILE) $(KMODFILE) $(CLOBBERFILES)

lint: $$(LINTFILES)

.NO_PARALLEL:
.PARALLEL: $(MODOBJS) $(KMODOBJS) mdb_tgt kmdb_tgt dmod kmod \
        $(TARGETS) $(LINTFILES)

$(MODFILE): dmod .WAIT $(MODOBJS) $$(MAPFILE-EXT)
        $(LINK.c) $(ZDEFS) $(ZIGNORE) $(MAPFILE-EXT:%=-M%) $(GSHARED) \
            $(MODOBJS) -o $@ $(LDLIBS) -lc -lproc
        $(CTFMERGE) -L VERSION -o $@ $(MODOBJS)
        $(POST_PROCESS_SO)

#
# kmdb dmods must *not* stray from the module API.  To ensure that they don’t,
# we try to link them, at build time, against an object that exports the symbols
# that they can legally use.  The link test object is, however, only built when
# kmdb itself is built.  Requiring module developers to build kmdb first would
```

```
# be painful, so by default, module-level builds don’t do the link test (the
# $(POUND_SIGN) assignment below takes care of that).  Builds of the entire
# tree can, however, guarantee the construction of kmdb first, and as such can
# override the setting of $(KMDB_LINKTEST_ENABLE).  This override causes the
# link test to be run.
#
# Developers wanting to force a link test for a single module can use the
# ‘linktest’ target from within a module directory.
#
LINKTESTOBJ = $(KMDBDIR)/kmdb_modlinktest.o

KMDB_LINKTEST = \
        $(LD) $(ZDEFS) -dy -r -o $@.linktest $(KMODOBJS) \
        $(STANDOBJS) $(LINKTESTOBJ) && \
        $(RM) $@.linktest

KMDB_LINKTEST_ENABLE=$(POUND_SIGN)
$(KMDB_LINKTEST_ENABLE)KMDB_LINKTEST_CMD = $(KMDB_LINKTEST)

#
# Ensure that dmods don’t use floating point
#
KMDB_FPTEST_CMD = $(KMDB_FPTEST)

$(KMODFILE): kmod .WAIT $(KMODOBJS) $(MAPFILE)
        $(LD) -dy -r $(MAPFILE:%=-M%) -Nmisc/kmdbmod -o $@ $(KMODOBJS) \
        $(STANDOBJS)
        $(KMDB_LINKTEST_CMD)
        $(KMDB_FPTEST_CMD)
        $(CTFMERGE) -f -L VERSION -o $@ $(KMODOBJS)
        $(POST_PROCESS)
        $(SETDYNFLAG) -f DF_1_NOKSYMS $@

linktest: linktest_check .WAIT kmod .WAIT $(KMODOBJS)
        $(KMDB_LINKTEST)

linktest_check:
        @if [ "$(MDBTGT)" != "kvm" ] ; then \
                echo "ERROR: linktest is not supported non-kvm/disasm dmods" \
                    >&2 ; \
                exit 1 ; \
        fi

#
# Dynamic rules for object construction
#
dmod/%.o kmod/%.o: %.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.o kmod%.o: %.s
        $(COMPILE.s) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.o kmod/%.o: ../%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.o kmod%.o: ../%.s
        $(COMPILE.s) -o $@ $<
        $(CTFCONVERT_O)

dmod/%.o kmod/%.o: ../../../common/modules/$(MODNAME)/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
dmod/%.o kmod%.o: ../../../common/modules/$(MODNAME)/%.s
	$(COMPILE.s) -o $@ $<
	$(CTFCONVERT_O)

dmod/%.o kmod/%.o: $$(MODSRCS_DIR)/%.c
	$(COMPILE.c) -o $@ $<
	$(CTFCONVERT_O)

dmod/%.o kmod%.o: $$(MODSRCS_DIR)/%.s
	$(COMPILE.s) -o $@ $<
	$(CTFCONVERT_O)

#
# Lint
#
dmod/%.ln kmod/%.ln: %.c
	$(LINT.c) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: %.s
	$(LINT.s) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: ../%.c
	$(LINT.c) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: ../%.s
	$(LINT.s) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: ../../../common/modules/$(MODNAME)/%.c
	$(LINT.c) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: ../../../common/modules/$(MODNAME)/%.s
	$(LINT.s) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: $$(MODSRCS_DIR)/%.c
	$(LINT.c) -dirout=$(@D) -c $<

dmod/%.ln kmod/%.ln: $$(MODSRCS_DIR)/%.s
	$(LINT.s) -dirout=$(@D) -c $<

#
# Installation targets
#

$(ROOT)/usr/lib/mdb/$(MDBTGT): $(ROOT)/usr/lib/mdb
	$(INS.dir)

$(ROOT)/usr/lib/mdb:
	$(INS.dir)

$(ROOT)/kernel/kmdb:
	$(INS.dir)

$(ROOTMOD)/$(MODULE): $(ROOTMOD)

$(ROOTKMOD)/$(KMODULE): $(ROOTKMOD)

kmod dmod:
	-@mkdir -p $@
```

```
**********************************************************
    1827 Thu Apr 25 02:20:40 2019
new/usr/src/cmd/mdb/common/modules/genunix/Makefile.files
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
# Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2019, Joyent, Inc.
# Copyright 2016 Joyent, Inc.
# Copyright (c) 2013 by Delphix. All rights reserved.
#

#
# This file simply contains the list of sources files compiled together
# to create the genunix mdb module.   Having them in one place saves
# a bunch of unnecessary replication.
#
GENUNIX_SRCS =          \
        avl.c           \
        bio.c           \
        bitset.c        \
        combined.c      \
        contract.c      \
        cpupart.c       \
        cred.c          \
        ctxop.c         \
        cyclic.c        \
        damap.c         \
        ddi_periodic.c  \
        devinfo.c       \
        dist.c          \
        dnlc.c          \
        findstack.c     \
        findstack_subr.c \
        fm.c            \
        gcore.c         \
        gcore_isadep.c  \
```

```
        genunix.c       \
        group.c         \
        hotplug.c       \
        irm.c           \
        kgrep.c         \
        kmem.c          \
        ldi.c           \
        leaky.c         \
        leaky_subr.c    \
        lgrp.c          \
        list.c          \
        log.c           \
        mdi.c           \
        memory.c        \
        mmd.c           \
        modhash.c       \
        ndievents.c     \
        net.c           \
        netstack.c      \
        nvpair.c        \
        pg.c            \
        rctl.c          \
        sobj.c          \
        streams.c       \
        sysevent.c      \
        taskq.c         \
        thread.c        \
        tsd.c           \
        tsol.c          \
        vfs.c           \
        zone.c
```

****
    2330 Thu Apr 25 02:20:40 2019
new/usr/src/cmd/mdb/intel/amd64/genunix/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
****

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


#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2013 by Delphix. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

MODULE = genunix.so
MDBTGT = kvm

include ../../../common/modules/genunix/Makefile.files

COMMONSRCS = \
        $(GENUNIX_SRCS)

KMODSRCS = \
        $(COMMONSRCS)

MODSRCS = \
        $(COMMONSRCS) \
        gcore.c \
        typegraph.c

#
# This signals that $(KMODSRCS) != $(MODSRCS).  Typegraph is not usable under
# kmdb.  As such, we don’t bother compiling it.
KMOD_SOURCES_DIFFERENT=$(POUND_SIGN)

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

```
include ../../../../Makefile.cmd.64
include ../../Makefile.amd64
include ../../../Makefile.module

dmod/$(MODULE) := LDLIBS += -lm -lproc

#
# We are not actually hardwiring some dependency on i86pc, we just need to
# include sys/param.h with _MACHDEP defined, and this forces the inclusion of
# machparam.h, even though we don’t use anything there.  This is a temporary
# kludge until we invent -DDONTINCLUDEMACHPARAM or something.
#
CPPFLAGS += -I$(SRC)/uts/i86pc
CPPFLAGS += -I$(SRC)/uts/i86xpv

# Needed to include c2/audit.h (from cred.h)
CPPFLAGS += -I$(SRC)/uts/common

# Needed to find include file mutex_impl.h
CPPFLAGS += -I$(SRC)/uts/intel

CERRWARN += -_gcc=-Wno-char-subscripts
CERRWARN += -_gcc=-Wno-unused-label
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-parentheses
CERRWARN += -_gcc=-Wno-type-limits

LINTFLAGS64 += -erroff=E_EMPTY_TRANSLATION_UNIT

MODSRCS_DIR = ../../../intel/modules/genunix
```

```
**********************************************************
    2506 Thu Apr 25 02:20:40 2019
new/usr/src/cmd/sgs/rtld/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
#
# Copyright (c) 2019, Joyent, Inc.
#

# Don’t generate separate sections for this module
pics/debugdata.o :=     XFFLAG=

# Since we use RTLD rather than DYNLIB, we must explicitly hook in ctfmerge.
$(RTLD) :=      CTFMERGE_POST = $(CTFMERGE_LIB)

pics/%.o:       %.s
                $(BUILD.s)
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

pics/%.o:       $(CRTSRCS)/%.s
                $(BUILD.s)
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

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

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

```
pics/%.o:       $(VAR_AVLDIR)/%.c
                $(COMPILE.c) -o $@ $<
                $(POST_PROCESS_O)

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

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

pics/%.o:       $(SRCBASE)/uts/$(PLAT)/krtld/%.c
                $(COMPILE.c) -o $@ $<
                $(POST_PROCESS_O)

pics/elfcap.o:  $(ELFCAP)/elfcap.c
                $(COMPILE.c) -o $@ $(ELFCAP)/elfcap.c
                $(POST_PROCESS_O)

$(RTLD):        pics $(PICS) $(CRTS)
                $(LD_USE_PROTO)ld -o $@ -dy -G $(DYNFLAGS) $(CRTI) $(PICS) \
                    $(LDLIBS) $(CRTN)
                $(POST_PROCESS_SO)

delete:
                -$(RM) $(RTLD)

$(ROOTFS_LIBDIR)/%:     %
                $(INS.file)

$(ETCLIBDIR)/%:         %
                $(INS.file)

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

# Derived source and header files (messaging).

catalog:        $(BLTMESG)

chkmsg :=       SGSMSGCHK=../common/rtld.chk.msg

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

$(BLTDEFS) + \
$(BLTDATA) :    $(SGSMSGTARG)
                $(SGSMSG) $(SGSMSGFLAGS2) $(SGSMSGTARG)

$(BLTMESG) :    $(SGSMSGALL)
                $(SGSMSG) $(SGSMSGFLAGS1) $(SGSMSGALL)
```

```
**********************************************************
    4637 Thu Apr 25 02:20:40 2019
new/usr/src/common/ctf/ctf_error.c
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright (c) 2019, Joyent, Inc.
 * Copyright (c) 2015, Joyent, Inc.
 */

#include <ctf_impl.h>

static const char *const _ctf_errlist[] = {
	"File is not in CTF or ELF format",		/* ECTF_FMT */
	"File uses more recent ELF version than libctf", /* ECTF_ELFVERS */
	"File uses more recent CTF version than libctf", /* ECTF_CTFVERS */
	"File is a different endian-ness than libctf",   /* ECTF_ENDIAN */
	"Symbol table uses invalid entry size",		/* ECTF_SYMTAB */
	"Symbol table data buffer is not valid",	/* ECTF_SYMBAD */
	"String table data buffer is not valid",	/* ECTF_STRBAD */
	"File data structure corruption detected",	/* ECTF_CORRUPT */
	"File does not contain CTF data",		/* ECTF_NOCTFDATA */
	"Buffer does not contain CTF data",		/* ECTF_NOCTFBUF */
	"Symbol table information is not available",	/* ECTF_NOSYMTAB */
	"Type information is in parent and unavailable", /* ECTF_NOPARENT */
	"Cannot import types with different data model", /* ECTF_DMODEL */
	"Failed to mmap a needed data section",		/* ECTF_MMAP */
	"Decompression package SUNWzlib not installed",	/* ECTF_ZMISSING */
	"Failed to initialize decompression library",	/* ECTF_ZINIT */
	"Failed to allocate decompression buffer",	/* ECTF_ZALLOC */
	"Failed to decompress CTF data",		/* ECTF_DECOMPRESS */
	"External string table is not available",	/* ECTF_STRTAB */
	"String name offset is corrupt",		/* ECTF_BADNAME */
```

```
	"Invalid type identifier",			/* ECTF_BADID */
	"Type is not a struct or union",		/* ECTF_NOTSOU */
	"Type is not an enum",				/* ECTF_NOTENUM */
	"Type is not a struct, union, or enum",		/* ECTF_NOTSUE */
	"Type is not an integer or float",		/* ECTF_NOTINTFP */
	"Type is not an array",				/* ECTF_NOTARRAY */
	"Type does not reference another type",		/* ECTF_NOTREF */
	"Input buffer is too small for type name",	/* ECTF_NAMELEN */
	"No type information available for that name",	/* ECTF_NOTYPE */
	"Syntax error in type name",			/* ECTF_SYNTAX */
	"Symbol table entry is not a function",		/* ECTF_NOTFUNC */
	"No function information available for symbol",	/* ECTF_NOFUNCDAT */
	"Symbol table entry is not a data object",	/* ECTF_NOTDATA */
	"No type information available for symbol",	/* ECTF_NOTYPEDAT */
	"No label information available for that name",	/* ECTF_NOLABEL */
	"File does not contain any labels",		/* ECTF_NOLABELDATA */
	"Feature not supported",			/* ECTF_NOTSUP */
	"Invalid enum element name",			/* ECTF_NOENUMNAM */
	"Invalid member name",				/* ECTF_NOMEMBNAM */
	"CTF container is read-only",			/* ECTF_RDONLY */
	"Limit on number of dynamic type members reached", /* ECTF_DTFULL */
	"Limit on number of dynamic types reached",	/* ECTF_FULL */
	"Duplicate member name definition",		/* ECTF_DUPMEMBER */
	"Conflicting type is already defined",		/* ECTF_CONFLICT */
	"Type has outstanding references",		/* ECTF_REFERENCED */
	"Type is not a dynamic type",			/* ECTF_NOTDYN */
	"Elf library failure",				/* ECTF_ELF */
	"Cannot merge child container",			/* ECTF_MCHILD */
	"Label already exists",				/* ECTF_LABEL */
	"Merged labels conflict",			/* ECTF_LCONFLICT */
	"Zlib library failure",				/* ECTF_ZLIB */
	"CTF conversion backend error",			/* ECTF_CONVBKERR */
	"No C source to convert from",			/* ECTF_CONVNOCSRC */
	"No debug info found to convert from"		/* ECTF_CONVNODEBUG */
	"No applicable conversion backend"		/* ECTF_NOCONVBKEND */
};
______unchanged_portion_omitted_
```

```
**********************************************************
    8452 Thu Apr 25 02:20:41 2019
new/usr/src/lib/Makefile.lib
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
# Copyright 2015 Gary Mills
# Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2019, Joyent, Inc.
# Copyright (c) 2015, Joyent, Inc.
#
#
# Definitions common to libraries.
#
# include global definitions; SRC should be defined in the shell.
# SRC is needed until RFE 1026993 is implemented.

include         $(SRC)/Makefile.master

LORDER=         lorder
TSORT=          tsort

#
# By default, we define the source directory for libraries to be
# one level up from the ISA-specific directory, where the code is
# actually built.  Many libraries define a 'common' directory to
# contain the source.  These libraries must redefine SRCDIR as:
#       SRCDIR = ../common
# Other variations are possible (../port, ../src, etc).
#
SRCDIR =        ..

#
# We define MAPFILES here for the benefit of most libraries, those that
# follow the convention of having source files and other common files
# in the $(SRCDIR) directory.  Libraries that do not follow this
# convention must define MAPFILES, or MAPFILEDIR for themselves.
```

```
# Libraries that do follow this convention but that need supplemental
# ISA-specific mapfiles can augment MAPFILES like this:
#       MAPFILES += mapfile-vers
#
MAPFILEDIR =    $(SRCDIR)
MAPFILES =      $(MAPFILEDIR)/mapfile-vers

#
# If HDRDIR is left unset, then it's possible for the $(ROOTHDRDIR)/%
# install rule in lib/Makefile.targ to generate false matches if there
# are any common directory names between / and /usr/include ('xfn' is
# one common example).  To prevent this, we set HDRDIR to a directory
# name that will almost surely not exist on the build machine.
#
HDRDIR=         /__nonexistent_directory__

#
# We don't build archive (*.a) libraries by default anymore.
# If a component of the build needs to build an archive library
# for its own internal purposes, it can define LIBS for itself
# after including Makefile.lib, like this:
#       LIBS = $(LIBRARY)
# or:
#       LIBS = $(LIBRARYCCC)
# Archive libraries must not be installed in the proto area.
#
LIBS=
MACHLIBS=       $(LIBS:%=$(MACH)/%)
MACHLIBS64=     $(LIBS:%=$(MACH64)/%)
DYNLIB=         $(LIBRARY:.a=.so$(VERS))
DYNLIBPSR=      $(LIBRARY:.a=_psr.so$(VERS))
DYNLIBCCC=      $(LIBRARYCCC:.a=.so$(VERS))
LIBLINKS=       $(LIBRARY:.a=.so)
LIBLINKSCCC=    $(LIBRARYCCC:.a=.so)
LIBNAME=        $(LIBRARY:lib%.a=%)
LIBLINKPATH=
LIBNULL=        null.a
ROOTHDRDIR=     $(ROOT)/usr/include
ROOTLIBDIR=     $(ROOT)/usr/lib
ROOTLIBDIR64=   $(ROOT)/usr/lib/$(MACH64)
ROOTFS_LIBDIR=  $(ROOT)/lib
ROOTFS_LIBDIR64=        $(ROOT)/lib/$(MACH64)
ROOTHDRS=       $(HDRS:%=$(ROOTHDRDIR)/%)
HDRSRCS=        $(HDRS:%=$(HDRDIR)/%)
CHECKHDRS=      $(HDRSRCS:%.h=%.check)
ROOTLIBS=       $(LIBS:%=$(ROOTLIBDIR)/%)
ROOTLIBS64=     $(LIBS:%=$(ROOTLIBDIR64)/%)
ROOTFS_LIBS=    $(DYNLIB:%=$(ROOTFS_LIBDIR)/%)
ROOTFS_LIBS64=  $(DYNLIB:%=$(ROOTFS_LIBDIR64)/%)
ROOTLINKS=      $(ROOTLIBDIR)/$(LIBLINKS)
ROOTLINKS64=    $(ROOTLIBDIR64)/$(LIBLINKS)
ROOTFS_LINKS=   $(ROOTFS_LIBDIR)/$(LIBLINKS)
ROOTFS_LINKS64= $(ROOTFS_LIBDIR64)/$(LIBLINKS)
ROOTLINKSCCC=   $(ROOTLIBDIR)/$(LIBLINKSCCC)
ROOTLINKSCCC64= $(ROOTLIBDIR64)/$(LIBLINKSCCC)
ROOTFS_LINKSCCC=        $(ROOTFS_LIBDIR)/$(LIBLINKSCCC)
ROOTFS_LINKSCCC64=      $(ROOTFS_LIBDIR64)/$(LIBLINKSCCC)

# Demo rules
DEMOFILES=
DEMOFILESRCDIR=         common
ROOTDEMODIRBASE=        __nonexistent_directory__
ROOTDEMODIRS=
ROOTDEMOFILES=  $(DEMOFILES:%=$(ROOTDEMODIRBASE)/%)
$(ROOTDEMODIRS) :=      DIRMODE =       755
```

```
ARFLAGS=        r
SONAME=         $(DYNLIB)
# For most libraries, we should be able to resolve all symbols at link time,
# either within the library or as dependencies, all text should be pure, and
# combining relocations into one relocation table reduces startup costs.
# All options are tunable to allow overload/omission from lower makefiles.


HSONAME=        -h$(SONAME)
DYNFLAGS=       $(HSONAME) $(ZTEXT) $(ZDEFS) $(BDIRECT) \
                $(MAPFILES:%=-M%) $(MAPFILE.PGA:%=-M%) $(MAPFILE.NED:%=-M%)

LDLIBS=         $(LDLIBS.lib)

OBJS=           $(OBJECTS:%=objs/%)
PICS=           $(OBJECTS:%=pics/%)

# Declare that all library .o's can all be made in parallel.
# The DUMMY target is for those instances where OBJS and PICS
# are empty (to avoid an unconditional .PARALLEL declaration).
.PARALLEL:      $(OBJS) $(PICS) DUMMY

# default value for "portable" source
SRCS=           $(OBJECTS:%.o=$(SRCDIR)/%.c)

# default build of an archive and a shared object,
# overridden locally when extra processing is needed
BUILD.AR=       $(AR) $(ARFLAGS) $@ $(AROBJS)
BUILD.SO=       $(CC) $(CFLAGS) -o $@ $(GSHARED) $(DYNFLAGS) \
                $(PICS) $(EXTPICS) $(LDLIBS)
BUILDCCC.SO=    $(CCC) $(CCFLAGS) -o $@ $(GSHARED) $(DYNFLAGS) \
                $(PICS) $(EXTPICS) $(LDLIBS)

# default dynamic library symlink
INS.liblink=    -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$(LIBLINKS)$(VERS) $@
INS.liblinkccc= -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$(LIBLINKSCCC)$(VERS) $@

# default 64-bit dynamic library symlink
INS.liblink64=  -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$(LIBLINKS)$(VERS) $@
INS.liblinkccc64= -$(RM) $@; $(SYMLINK) $(LIBLINKPATH)$(LIBLINKSCCC)$(VERS) $@

#
# If appropriate, augment POST_PROCESS_O and POST_PROCESS_SO to do CTF
# processing.  We'd like to just conditionally append to POST_PROCESS_O and
# POST_PROCESS_SO, but ParallelMake has a bug which causes the same value to
# sometimes get appended more than once, which will cause ctfconvert to fail.
# So, instead we introduce CTFCONVERT_POST and CTFMERGE_POST, which are always
# appended to POST_PROCESS_O and POST_PROCESS_SO but are no-ops unless CTF
# processing should be done.
#
CTFCONVERT_POST = :
CTFMERGE_POST   = :
POST_PROCESS_O += ; $(CTFCONVERT_POST)
POST_PROCESS_SO += ; $(CTFMERGE_POST)

CTFMERGE_LIB    = $(CTFMERGE) $(CTFMRGFLAGS) -t -f -L VERSION -o $@ $(PICS)

# conditional assignments

$(OBJS)  :=     sparc_CFLAGS += -xregs=no%appl

$(PICS)  :=     sparc_CFLAGS += -xregs=no%appl $(sparc_C_PICFLAGS)
$(PICS)  :=     sparcv9_CFLAGS += -xregs=no%appl $(sparcv9_C_PICFLAGS)
$(PICS)  :=     i386_CFLAGS += $(i386_C_PICFLAGS)
$(PICS)  :=     amd64_CFLAGS += $(amd64_C_PICFLAGS)
$(PICS)  :=     CCFLAGS += $(CC_PICFLAGS)
```

```
$(PICS)  :=     CPPFLAGS += -DPIC -D_REENTRANT
$(PICS)  :=     sparcv9_CCFLAGS += -xregs=no%appl $(sparcv9_CC_PICFLAGS)
$(PICS)  :=     amd64_CCFLAGS += $(amd64_CC_PICFLAGS)
$(PICS)  :=     CFLAGS += $(CTF_FLAGS)
$(PICS)  :=     CFLAGS64 += $(CTF_FLAGS)
$(PICS)  :=     CTFCONVERT_POST = $(CTFCONVERT_O)
$(DYNLIB) :=    CTFMERGE_POST = $(CTFMERGE_LIB)

$(LIBRARY):=    AROBJS = $(OBJS)
$(LIBRARY):=    DIR = objs
$(DYNLIB):=     DIR = pics
$(DYNLIBCCC):=  DIR = pics

SONAMECCC=      $(DYNLIBCCC)
HSONAMECCC=     -h $(SONAMECCC)
#
# Keep in sync with the standard DYNFLAGS
#
$(DYNLIBCCC):=  DYNFLAGS = $(HSONAMECCC) $(ZTEXT) $(ZDEFS) \
                $(MAPFILES:%=-M%) $(MAPFILE.PGA:%=-M%) $(MAPFILE.NED:%=-M%) \
                $(BDIRECT) $(NORUNPATH)


# build rule for "portable" source
objs/%.o pics/%.o: %.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/%.o pics/%.o: %.cc
        $(COMPILE.cc) -o $@ $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)

.PRECIOUS: $(LIBS)

# Define the majority text domain in this directory.
TEXT_DOMAIN= SUNW_OST_OSLIB

#
# Target Architecture
#
TARGETMACH=     $(MACH)

#
# Allow people to define their own clobber rules.  Normal makefiles
# shouldn't override this - they should override $(CLOBBERFILES) instead.
#
CLOBBERTARGFILES= $(LIBS) $(DYNLIB) $(CLOBBERFILES)

#
# Define the default ctfdiff invocation used to check a list of types
# supplied by a user of a library. The goal is to validate that a given
# series of types is the same in both a 32-bit and 64-bit artifact. This
# is only defined if we have a 64-bit build to do.
#
TYPECHECK_LIB32 =               $(TYPECHECK_LIB:%=$(MACH)/%)
TYPECHECK_LIB64 =               $(TYPECHECK_LIB:%=$(MACH64)/%)
TYPECHECK_LIST=                 $(TYPELIST:%=-T %)
$(BUILD64)TYPECHECK.lib =       $(CTFDIFF) -t -I $(TYPECHECK_LIST) $(TYPECHECK_L
TYPECHECK =                     $(TYPECHECK_LIB:%=%.typecheck)
```

```
************************************************************
    2994 Thu Apr 25 02:20:41 2019
new/usr/src/lib/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
************************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright (c) 2019, Joyent, Inc.
#

#
# included to define local library targets
#

# conditional assignment of default permissions for the installed
# DYNLIB/DYNLIBCCC.  Set here and in Makefile.mach, instead of Makefile.lib,
# because ROOTLIBDIR may be changed in the including makefile after it has
# included Makefile.lib.
#
$(ROOTLIBDIR)/$(DYNLIB) :=      FILEMODE= 755
$(ROOTLIBDIR)/$(DYNLIBCCC) :=   FILEMODE= 755

$(ROOTFS_LIBDIR)/$(DYNLIB) :=   FILEMODE= 755
$(ROOTFS_LIBDIR)/$(DYNLIBCCC) :=        FILEMODE= 755

$(ROOTLIBDIR64)/$(DYNLIB) :=    FILEMODE= 755
$(ROOTLIBDIR64)/$(DYNLIBCCC) := FILEMODE= 755

# install rules
$(ROOTHDRDIR)/%: $(HDRDIR)/%
        $(INS.file)

$(ROOTLIBDIR)/%: %
        $(INS.file)
$(ROOTLIBDIR64)/%: %
        $(INS.file)
$(ROOTFS_LIBDIR)/%: %
        $(INS.file)
```

```
$(ROOTFS_LIBDIR64)/%: %
        $(INS.file)

$(ROOTLIBDIR)/$(LIBLINKS): $(ROOTLIBDIR)/$(LIBLINKS)$(VERS)
        $(INS.liblink)
$(ROOTLIBDIR64)/$(LIBLINKS): $(ROOTLIBDIR64)/$(LIBLINKS)$(VERS)
        $(INS.liblink64)

$(ROOTLIBDIR)/$(LIBLINKSCCC): $(ROOTLIBDIR)/$(LIBLINKSCCC)$(VERS)
        $(INS.liblinkccc)
$(ROOTLIBDIR64)/$(LIBLINKSCCC): $(ROOTLIBDIR64)/$(LIBLINKSCCC)$(VERS)
        $(INS.liblinkccc64)

$(ROOTLINTDIR)/%: $(SRCDIR)/%
        $(INS.file)
$(ROOTFS_LINTDIR)/%: $(SRCDIR)/%
        $(INS.file)

$(ROOTDEMODIRS):
        $(INS.dir)
$(ROOTDEMODIRBASE)/%: $(DEMOFILESRCDIR)/%
        $(INS.file)

objs/%.o pics/%.o: $(SRCDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/%.o pics/%.o: $(SRCDIR)/%.cc
        $(COMPILE.cc) -o $@ $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)

objs pics:
        -@mkdir -p $@

$(LIBRARY): objs .WAIT $$(OBJS)
        $(BUILD.AR)
        $(POST_PROCESS_A)

$(DYNLIB): $$(MAPFILES)

$(DYNLIB): pics .WAIT $$(PICS) $$(ALTPICS) $$(EXTPICS)
        $(BUILD.SO)
        $(POST_PROCESS_SO)

$(DYNLIBCCC): pics .WAIT $$(PICS) $$(ALTPICS) $$(EXTPICS)
        $(BUILDCCC.SO)
        $(POST_PROCESS_SO)

$(LINTLIB): $$(SRCS)
        $(LINT.c) -o $(LIBNAME) $(SRCS) > $(LINTOUT) 2>&1

lintcheck: $$(SRCS)
        $(LINT.c) $(LINTCHECKFLAGS) $(SRCS) $(LDLIBS)

$(TYPECHECK): $(TYPECHECK_LIB32) $(TYPECHECK_LIB64)
        $(TYPECHECK.lib)

clobber: clean
        -$(RM) $(CLOBBERTARGFILES)

clean:
        -$(RM) $(OBJS)
        -$(RM) $(PICS) $(DUPLICATE_SRC) $(LINTOUT) $(LINTLIB) $(CLEANFILES)
```

```
**********************************************************
    2530 Thu Apr 25 02:20:41 2019
new/usr/src/lib/brand/shared/brand/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

COBJS =         brand_util.o
ASOBJS =        crt.o handler.o runexe.o
OFFSETS_SRC =   ../common/offsets.in
OFFSETS_H =     assym.h
OBJECTS =       $(COBJS) $(ASOBJS)
CLOBBERFILES += $(OFFSETS_H)

include $(SRC)/lib/Makefile.lib

SRCDIR =        ../common
CSRCS =         $(COBJS:%o=../common/%c)
ASSRCS =        $(ASOBJS:%o=$(ISASRCDIR)/%s)
SRCS =          $(CSRCS) $(ASSRCS)

#
# Ugh, this is a gross hack.  Our assembly routines uses lots of defines
# to simplify variable access.  All these defines work fine for amd64
# compiles because when compiling for amd64 we use the GNU assembler,
# gas.  For 32-bit code we use the Sun assembler, as.  Unfortunately
# as does not handle certian constructs that gas does.  So rather than
# make our code less readable, we'll just use gas to compile our 32-bit
# code as well.
#
i386_AS         = $(amd64_AS)
```

```
CPPFLAGS +=     -D_REENTRANT -U_ASM -I. -I../sys
CFLAGS +=       $(CCVERBOSE)
ASFLAGS =       -P $(ASFLAGS_$(CURTYPE)) -D_ASM -I. -I../sys

.KEEP_STATE:

#
# build the offset header before trying to compile any files.  (it's included
# by brand_misc.h, so it's needed for all objects, not just assembly ones.)
#
# Note we have to build assym.h via its dependency on pics/% so that the
# target dependent assignment of CTF_FLAGS will be there, otherwise make
# will see two different commands to build it (endless rebuilds).
#
all: pics .WAIT $$(PICS)

lint: lintcheck

$(OBJECTS:%=pics/%): $(OFFSETS_H)

$(OFFSETS_H): $(OFFSETS_SRC)
        $(OFFSETS_CREATE) < $(OFFSETS_SRC) >$@

pics/%.o: $(ISASRCDIR)/%.s
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

include $(SRC)/lib/Makefile.targ
```

```
**********************************************************
     807 Thu Apr 25 02:20:42 2019
new/usr/src/lib/commpage/Makefile.shared.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2019, Joyent, Inc.
# Copyright 2016 Joyent, Inc.
#

#
# This Makefile is shared between both libc and other consumers
#

pics/%.o: $(SRC)/lib/commpage/common/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(SRC)/lib/commpage/$(TARGET_ARCH)/%.s $(COMMPAGE_OFFSETS_H)
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

$(COMMPAGE_OFFSETS_H): $(COMMPAGE_OFFSETS_SRC)
        $(OFFSETS_CREATE) <$(COMMPAGE_OFFSETS_SRC) >$@
```

```
**********************************************************
    7457 Thu Apr 25 02:20:42 2019
new/usr/src/lib/libc/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2016 Joyent, Inc.
#
# Copyright (c) 2019, Joyent, Inc.
#

# libc build rules

#
# This first rule picks up optimized sources for libc variants.
#
pics/%.o: $(OPTIMIZED_LIBCBASE)/gen/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/crt/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/crt/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/gen/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/gen/%.s
        $(BUILD.s)
```

```
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/gen/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/_stack_grow.o: $(LIBCBASE)/../$(TARGET_ARCH)/gen/_stack_grow.s
        $(BUILD.s) $(LIBCBASE)/../$(TARGET_ARCH)/gen/_stack_grow.s
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/fp/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(MACH)/fp/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/fp/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/fp/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/fp/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/sys/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../common/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(MACH)/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(MACH)/sys/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/sys/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)
```

```
# rules to build large file aware objects (xxx64.o from xxx.s or xxx.c)

pics/%64.o: $(LIBCBASE)/../common/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/sys/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/gen/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/../port/gen/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/../port/sys/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/../port/print/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/../port/regex/%.c
        $(COMPILE.c) -DM_I18N_MB -DI18N \
        -I$(LIBCBASE)/../port/regex \
        -I$(LIBCBASE)/../port/gen -o $@ $<
        $(POST_PROCESS_O)

pics/%64.o: $(LIBCBASE)/../port/stdio/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_w.o: $(LIBCBASE)/../port/stdio/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_w.o: $(LIBCBASE)/../port/print/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_pos.o: $(LIBCBASE)/../port/i18n/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_sbyte.o: $(LIBCBASE)/../port/i18n/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_possbyte.o: $(LIBCBASE)/../port/i18n/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_longlong.o: $(LIBCBASE)/../port/i18n/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# libc build rules for objects built from "portable" source in ../port

pics/%.o: $(LIBCBASE)/../port/fp/%.c
```

```
        $(COMPILE.c) \
        -I$(LIBCBASE)/../port/fp -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../port/i18n/%.c
        $(COMPILE.c) \
        -I$(LIBCBASE)/../port/i18n -o $@ $<
        $(POST_PROCESS_O)

# gen rules
pics/%.o %.o: $(LIBCBASE)/../port/gen/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# inet rules
pics/%.o %.o: $(LIBCBASE)/../port/inet/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# locale rules
pics/%.o %.o: $(LIBCBASE)/../port/locale/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# print rules
pics/%.o: $(LIBCBASE)/../port/print/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# regex rules
pics/%.o: $(LIBCBASE)/../port/regex/%.c
        $(COMPILE.c) -DM_I18N_MB -DI18N \
        -I$(LIBCBASE)/../port/regex \
        -I$(LIBCBASE)/../port/gen -o $@ $<
        $(POST_PROCESS_O)

# stdio rules
pics/%.o: $(LIBCBASE)/../port/stdio/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# c89 print, stdio rules
pics/%_c89.o: $(LIBCBASE)/../port/print/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%_c89.o: $(LIBCBASE)/../port/stdio/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# aio rules
pics/%.o: $(LIBCBASE)/../port/aio/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# rt rules
pics/%.o: $(LIBCBASE)/../port/rt/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# tpool rules
pics/%.o: $(LIBCBASE)/../port/tpool/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# threads rules
```

```
pics/%.o: $(LIBCBASE)/../port/threads/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/threads/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/threads/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/threads/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/threads/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/unwind/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(TARGET_ARCH)/unwind/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../port/unwind/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../$(MACH)/unwind/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(LIBCBASE)/../port/sys/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

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

pics/%.o: $(LIBCBASE)/$(CRTSRCS)/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

# $(SRC)/common rules
pics/%.o: $(SRC)/common/atomic/$(TARGETMACH)/%.s
        $(BUILD.s)
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

$(COMOBJS:%=pics/%): $(SRC)/common/util/$$(@F:.o=.c)
        $(COMPILE.c) -o $@ $(SRC)/common/util/$(@F:.o=.c)
        $(POST_PROCESS_O)

$(XATTROBJS:%=pics/%): $(SRC)/common/xattr/$$(@F:.o=.c)
        $(COMPILE.c) -o $@ $(SRC)/common/xattr/$(@F:.o=.c)
        $(POST_PROCESS_O)
```

```
$(DTRACEOBJS:%=pics/%): $(SRC)/common/dtrace/$$(@F:.o=.c)
        $(COMPILE.c) -o $@ $(SRC)/common/dtrace/$(@F:.o=.c)
        $(POST_PROCESS_O)

$(SECFLAGSOBJS:%=pics/%): $(SRC)/common/secflags/$$(@F:.o=.c)
        $(COMPILE.c) -o $@ $(SRC)/common/secflags/$(@F:.o=.c)
        $(POST_PROCESS_O)

$(UNICODEOBJS:%=pics/%): $(SRC)/common/unicode/$$(@F:.o=.c)
        $(COMPILE.c) -o $@ $(SRC)/common/unicode/$(@F:.o=.c)
        $(POST_PROCESS_O)

$(CHACHAOBJS:%=pics/%): $(SRC)/common/crypto/chacha/$$(@F:.o=.c)
        $(COMPILE.c) -I$(SRC)/common/crypto/chacha -DKEYSTREAM_ONLY \
            -o $@ $(SRC)/common/crypto/chacha/$(@F:.o=.c)
        $(POST_PROCESS_O)

# DTrace rules
pics/%.o: $(LIBCBASE)/../port/threads/%.d $(THREADSOBJS:%=pics/%)
        $(COMPILE.d) -C -xlazyload -s $< -o $@ $(THREADSOBJS:%=pics/%)
        $(POST_PROCESS_O)

include $(SRC)/lib/Makefile.targ
```

```
**********************************************************
    1613 Thu Apr 25 02:20:42 2019
new/usr/src/lib/libc/capabilities/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

$(SYMCAP):      $(OBJCAP)
$(OBJCAP):      $(PICS)
$(PICS):        pics

# Assemble the original source file.

pics/%.o:       ../../$(TRG_PLATFORM)/common/%.s
                $(COMPILE.s) $< -o $@
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

pics/%.o:       ../../$(ALT_PLATFORM)/common/%.s
                $(COMPILE.s) $< -o $@
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

# Combine all pic objects into one relocatable object.  Assign any capabilities
# to this object, and define the interface.

pics/objcap.o:  $(PICS) $(MAPFILE-CAP)
                $(LD) -r -o $@ $(MAPOPT-CAP) -Breduce $(PICS)
                $(POST_PROCESS_OBJCAP_O)

# Convert the combined object capabilities object into a symbol capabilities
# object.
```

```
pics/symcap.o:  $(OBJCAP)
                $(LD) -r -o $@ -z symbolcap $(OBJCAP)

include         ../../../Makefile.targ
```

```
**********************************************************
    1598 Thu Apr 25 02:20:42 2019
new/usr/src/lib/libc_db/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

LIBRARY = libc_db.a
VERS = .1

CRTI=   crti.o
CRTN=   crtn.o

CMNOBJS = thread_db.o
OBJECTS = $(CRTI) $(CMNOBJS) $(CRTN)

include ../../Makefile.lib
include ../../Makefile.rootfs

LIBS = $(DYNLIB) $(LINTLIB)

SRCDIR =        ../common
SRCS = $(CMNOBJS:%.o=$(SRCDIR)/%.c)
$(LINTLIB) := SRCS = $(SRCDIR)/$(LINTSRC)

ASFLAGS +=      -P -D__STDC__ -D_ASM -DPIC
CPPFLAGS +=     -I../../libc/inc -D_REENTRANT
CFLAGS +=       $(CCVERBOSE)
LDLIBS +=       -lc

CERRWARN +=     -_gcc=-Wno-type-limits
CERRWARN +=     -_gcc=-Wno-uninitialized
```

```
# not linted
SMATCH=off

.KEEP_STATE:

all: $(LIBS)

lint: lintcheck

include ../../Makefile.targ

pics/%.o: $(CRTSRCS)/%.s
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)
```

```
**********************************************************
    1728 Thu Apr 25 02:20:43 2019
new/usr/src/lib/libcpc/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

LIBRARY = libcpc.a
VERS    = .1
COBJS   = libcpc.o subr.o
V1_OBJS = obsoleted.o
OBJECTS = $(ASOBJS) $(MACHCOBJS) $(COBJS) $(V1_OBJS)

include ../../Makefile.lib

SRCS =  $(ASOBJS:%.o=../$(MACH)/%.s)    \
        $(MACHCOBJS:%.o=../$(MACH)/%.c) \
        $(V1_OBJS:%.o=../common/%.c)    \
        $(COBJS:%.o=../common/%.c)

LIBS =          $(DYNLIB) $(LINTLIB)
$(LINTLIB) :=   SRCS = ../common/llib-lcpc
LDLIBS +=       -lpctx -lnvpair -lc

SRCDIR =        ../common

ASFLAGS +=      -P -D_ASM -I../common
CPPFLAGS +=     -D_REENTRANT -I../common
CFLAGS +=       $(CCVERBOSE)

CERRWARN +=     -_gcc=-Wno-switch
CERRWARN +=     -_gcc=-Wno-uninitialized

# not linted
SMATCH=off

.KEEP_STATE:

all: $(LIBS)

lint: lintcheck

include ../../Makefile.targ

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

pics/%.o: ../$(MACH)/%.s
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)
```

```
**********************************************************
    4135 Thu Apr 25 02:20:43 2019
new/usr/src/lib/libctf/common/ctf_convert.c
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2019 Joyent, Inc.
 */

/*
 * Main conversion entry points. This has been designed such that there can be
 * any number of different conversion backends. Currently we only have one that
 * understands DWARFv2 (and bits of DWARFv4). Each backend should be placed in
 * the ctf_converters list and each will be tried in turn.
 */

#include <libctf_impl.h>
#include <assert.h>
#include <gelf.h>

ctf_convert_f ctf_converters[] = {
        ctf_dwarf_convert
};

#define NCONVERTS       (sizeof (ctf_converters) / sizeof (ctf_convert_f))

ctf_hsc_ret_t
ctf_has_c_source(Elf *elf, char *errmsg, size_t errlen)
typedef enum ctf_convert_source {
        CTFCONV_SOURCE_NONE = 0x0,
        CTFCONV_SOURCE_UNKNOWN = 0x01,
        CTFCONV_SOURCE_C = 0x02,
        CTFCONV_SOURCE_S = 0x04
} ctf_convert_source_t;

static void
ctf_convert_ftypes(Elf *elf, ctf_convert_source_t *types)
{
        ctf_hsc_ret_t ret = CHR_NO_C_SOURCE;
        Elf_Scn *scn, *strscn;
        Elf_Data *data, *strdata;
        int i;
        Elf_Scn *scn = NULL, *strscn;
        *types = CTFCONV_SOURCE_NONE;
        GElf_Shdr shdr;
        ulong_t i;
        Elf_Data *data, *strdata;
```

```
        scn = NULL;
        while ((scn = elf_nextscn(elf, scn)) != NULL) {
                if (gelf_getshdr(scn, &shdr) == NULL) {
                        (void) snprintf(errmsg, errlen,
                            "failed to get section header: %s",
                            elf_errmsg(elf_errno()));
                        return (CHR_ERROR);
                }

                if (gelf_getshdr(scn, &shdr) == NULL)
                        return;

                if (shdr.sh_type == SHT_SYMTAB)
                        break;
        }

        if (scn == NULL)
                return (CHR_NO_C_SOURCE);
                return;

        if ((strscn = elf_getscn(elf, shdr.sh_link)) == NULL) {
                (void) snprintf(errmsg, errlen, "failed to get str section: %s",
                    elf_errmsg(elf_errno()));
                return (CHR_ERROR);
        }
        if ((strscn = elf_getscn(elf, shdr.sh_link)) == NULL)
                return;

        if ((data = elf_getdata(scn, NULL)) == NULL) {
                (void) snprintf(errmsg, errlen, "failed to read section: %s",
                    elf_errmsg(elf_errno()));
                return (CHR_ERROR);
        }
        if ((data = elf_getdata(scn, NULL)) == NULL)
                return;

        if ((strdata = elf_getdata(strscn, NULL)) == NULL) {
                (void) snprintf(errmsg, errlen,
                    "failed to read string table: %s", elf_errmsg(elf_errno()));
                return (CHR_ERROR);
        }
        if ((strdata = elf_getdata(strscn, NULL)) == NULL)
                return;

        for (i = 0; i < shdr.sh_size / shdr.sh_entsize; i++) {
                GElf_Sym sym;
                const char *file;
                size_t len;

                if (gelf_getsym(data, i, &sym) == NULL) {
                        (void) snprintf(errmsg, errlen,
                            "failed to read sym %lu: %s",
                            i, elf_errmsg(elf_errno()));
                        return (CHR_ERROR);
                }
                if (gelf_getsym(data, i, &sym) == NULL)
                        return;

                if (GELF_ST_TYPE(sym.st_info) != STT_FILE)
                        continue;

                file = (const char *)((uintptr_t)strdata->d_buf + sym.st_name);
                len = strlen(file);
                if (len >= 2 && strncmp(".c", &file[len - 2], 2) == 0) {
                        ret = CHR_HAS_C_SOURCE;
                if (len < 2 || file[len - 2] != '.') {
                        *types |= CTFCONV_SOURCE_UNKNOWN;
                        continue;
                }

                switch (file[len - 1]) {
                case 'c':
                        *types |= CTFCONV_SOURCE_C;
                        break;
                case 'h':
                        /* We traditionally ignore header files... */
                        break;
                case 's':
                        *types |= CTFCONV_SOURCE_S;
                        break;
                default:
                        *types |= CTFCONV_SOURCE_UNKNOWN;
                        break;
                }
        }

        return (ret);
}

static ctf_file_t *
ctf_elfconvert(int fd, Elf *elf, const char *label, uint_t nthrs, uint_t flags,
    int *errp, char *errbuf, size_t errlen)
{
        int err, i;
        ctf_file_t *fp = NULL;
        boolean_t notsup = B_TRUE;
        ctf_convert_source_t type;

        if (errp == NULL)
                errp = &err;

        if (elf == NULL) {
                *errp = EINVAL;
                return (NULL);
        }

        if (flags & ~CTF_ALLOW_MISSING_DEBUG) {
        if (flags & ~CTF_CONVERT_F_IGNNONC) {
                *errp = EINVAL;
                return (NULL);
        }

        if (elf_kind(elf) != ELF_K_ELF) {
                *errp = ECTF_FMT;
                return (NULL);
        }

        switch (ctf_has_c_source(elf, errbuf, errlen)) {
        case CHR_ERROR:
                *errp = ECTF_ELF;
                return (NULL);

        case CHR_NO_C_SOURCE:
        ctf_convert_ftypes(elf, &type);
        ctf_dprintf("got types: %d\n", type);
        if (flags & CTF_CONVERT_F_IGNNONC) {
                if (type == CTFCONV_SOURCE_NONE ||
                    (type & CTFCONV_SOURCE_UNKNOWN)) {
                *errp = ECTF_CONVNOCSRC;
                return (NULL);
```

```
        default:
                break;
        }
        }

        for (i = 0; i < NCONVERTS; i++) {
                fp = NULL;
                err = ctf_converters[i](fd, elf, nthrs, flags,
                    &fp, errbuf, errlen);
                ctf_conv_status_t cs;

                if (err != ECTF_CONVNODEBUG)
                fp = NULL;
                cs = ctf_converters[i](fd, elf, nthrs, errp, &fp, errbuf,
                    errlen);
                if (cs == CTF_CONV_SUCCESS) {
                        notsup = B_FALSE;
                        break;
        }
                if (cs == CTF_CONV_ERROR) {
                        fp = NULL;
                        notsup = B_FALSE;
                        break;
                }
        }

        if (err != 0) {
                assert(fp == NULL);
                *errp = err;
        if (notsup == B_TRUE) {
                if ((flags & CTF_CONVERT_F_IGNNONC) != 0 &&
                    (type & CTFCONV_SOURCE_C) == 0) {
                        *errp = ECTF_CONVNOCSRC;
                return (NULL);
        }
                *errp = ECTF_NOCONVBKEND;
                return (NULL);
        }

        if (label != NULL) {
        /*
         * Succsesful conversion.
         */
        if (fp != NULL && label != NULL) {
                if (ctf_add_label(fp, label, fp->ctf_typemax, 0) == CTF_ERR) {
                        *errp = ctf_errno(fp);
                        ctf_close(fp);
                        return (NULL);
                }
                if (ctf_update(fp) == CTF_ERR) {
                        *errp = ctf_errno(fp);
                        ctf_close(fp);
                        return (NULL);
                }
        }

        return (fp);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
   84704 Thu Apr 25 02:20:43 2019
new/usr/src/lib/libctf/common/ctf_dwarf.c
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
______unchanged_portion_omitted_
```

```
static int
ctf_dwarf_dwarf_base(ctf_cu_t *cup, Dwarf_Die die, int *kindp,
    ctf_encoding_t *enc)
{
        int ret;
        Dwarf_Signed type;

        if ((ret = ctf_dwarf_signed(cup, die, DW_AT_encoding, &type)) != 0)
                return (ret);

        switch (type) {
        case DW_ATE_unsigned:
        case DW_ATE_address:
                *kindp = CTF_K_INTEGER;
                enc->cte_format = 0;
                break;
        case DW_ATE_unsigned_char:
                *kindp = CTF_K_INTEGER;
                enc->cte_format = CTF_INT_CHAR;
                break;
        case DW_ATE_signed:
                *kindp = CTF_K_INTEGER;
                enc->cte_format = CTF_INT_SIGNED;
                break;
        case DW_ATE_signed_char:
                *kindp = CTF_K_INTEGER;
                enc->cte_format = CTF_INT_SIGNED | CTF_INT_CHAR;
                break;
        case DW_ATE_boolean:
                *kindp = CTF_K_INTEGER;
                enc->cte_format = CTF_INT_SIGNED | CTF_INT_BOOL;
                break;
        case DW_ATE_float:
        case DW_ATE_complex_float:
        case DW_ATE_imaginary_float:
        case DW_ATE_SUN_imaginary_float:
        case DW_ATE_SUN_interval_float:
                *kindp = CTF_K_FLOAT;
                if ((ret = ctf_dwarf_float_base(cup, type, enc)) != 0)
                        return (ret);
                break;
        default:
                (void) snprintf(cup->cu_errbuf, cup->cu_errlen,
                    "encountered unknown DWARF encoding: %d", type);
                    "encountered unkown DWARF encoding: %d", type);
                return (ECTF_CONVBKERR);
        }

        return (0);
}
______unchanged_portion_omitted_
```

```
static int
ctf_dwarf_convert_type(ctf_cu_t *cup, Dwarf_Die die, ctf_id_t *idp,
    int isroot)
{
        int ret;
        Dwarf_Off offset;
        Dwarf_Half tag;
        ctf_dwmap_t lookup, *map;
        ctf_id_t id;

        if (idp == NULL)
                idp = &id;

        if ((ret = ctf_dwarf_offset(cup, die, &offset)) != 0)
                return (ret);

        if (offset > cup->cu_maxoff) {
                (void) snprintf(cup->cu_errbuf, cup->cu_errlen,
                    "die offset %llu beyond maximum for header %llu\n",
                    offset, cup->cu_maxoff);
                return (ECTF_CONVBKERR);
        }

        /*
         * If we've already added an entry for this offset, then we're done.
         */
        lookup.cdm_off = offset;
        if ((map = avl_find(&cup->cu_map, &lookup, NULL)) != NULL) {
                *idp = map->cdm_id;
                return (0);
        }

        if ((ret = ctf_dwarf_tag(cup, die, &tag)) != 0)
                return (ret);

        ret = ENOTSUP;
        switch (tag) {
        case DW_TAG_base_type:
                ctf_dprintf("base\n");
                ret = ctf_dwarf_create_base(cup, die, idp, isroot, offset);
                break;
        case DW_TAG_array_type:
                ctf_dprintf("array\n");
                ret = ctf_dwarf_create_array(cup, die, idp, isroot);
                break;
        case DW_TAG_enumeration_type:
                ctf_dprintf("enum\n");
                ret = ctf_dwarf_create_enum(cup, die, idp, isroot);
                break;
        case DW_TAG_pointer_type:
                ctf_dprintf("pointer\n");
                ret = ctf_dwarf_create_reference(cup, die, idp, CTF_K_POINTER,
                    isroot);
                break;
        case DW_TAG_structure_type:
                ctf_dprintf("struct\n");
                ret = ctf_dwarf_create_sou(cup, die, idp, CTF_K_STRUCT,
                    isroot);
                break;
        case DW_TAG_subroutine_type:
                ctf_dprintf("fptr\n");
                ret = ctf_dwarf_create_fptr(cup, die, idp, isroot);
                break;
        case DW_TAG_typedef:
                ctf_dprintf("typedef\n");
                ret = ctf_dwarf_create_reference(cup, die, idp, CTF_K_TYPEDEF,
                    isroot);
                break;
        case DW_TAG_union_type:
                ctf_dprintf("union\n");
                ret = ctf_dwarf_create_sou(cup, die, idp, CTF_K_UNION,
                    isroot);
                break;
        case DW_TAG_const_type:
                ctf_dprintf("const\n");
                ret = ctf_dwarf_create_reference(cup, die, idp, CTF_K_CONST,
                    isroot);
                break;
        case DW_TAG_volatile_type:
                ctf_dprintf("volatile\n");
                ret = ctf_dwarf_create_reference(cup, die, idp, CTF_K_VOLATILE,
                    isroot);
                break;
        case DW_TAG_restrict_type:
                ctf_dprintf("restrict\n");
                ret = ctf_dwarf_create_reference(cup, die, idp, CTF_K_RESTRICT,
                    isroot);
                break;
        default:
                ctf_dprintf("ignoring tag type %x\n", tag);
                *idp = CTF_ERR;
                ret = 0;
                break;
        }
        ctf_dprintf("ctf_dwarf_convert_type tag specific handler returned %d\n",
            ret);

        return (ret);
}
```

______***unchanged_portion_omitted_***

```
/*
 * Note, we expect that if we're returning a ctf_file_t from one of the dies,
 * say in the single node case, it's been saved and the entry here has been set
 * to NULL, which ctf_close happily ignores.
 */
static void
ctf_dwarf_free_die(ctf_cu_t *cup)
{
        ctf_dwfunc_t *cdf, *ndf;
        ctf_dwvar_t *cdv, *ndv;
        ctf_dwbitf_t *cdb, *ndb;
        ctf_dwmap_t *map;
        void *cookie;
        Dwarf_Error derr;

        ctf_dprintf("Beginning to free die: %p\n", cup);
        cup->cu_elf = NULL;
        ctf_dprintf("Trying to free name: %p\n", cup->cu_name);
        if (cup->cu_name != NULL)
                ctf_free(cup->cu_name, strlen(cup->cu_name) + 1);
        ctf_dprintf("Trying to free merge handle: %p\n", cup->cu_cmh);
        if (cup->cu_cmh != NULL) {
                ctf_merge_fini(cup->cu_cmh);
                cup->cu_cmh = NULL;
        }

        ctf_dprintf("Trying to free functions\n");
        for (cdf = ctf_list_next(&cup->cu_funcs); cdf != NULL; cdf = ndf) {
                ndf = ctf_list_next(cdf);
                ctf_free(cdf->cdf_name, strlen(cdf->cdf_name) + 1);
```

```
                if (cdf->cdf_fip.ctc_argc != 0) {
                        ctf_free(cdf->cdf_argv,
                            sizeof (ctf_id_t) * cdf->cdf_fip.ctc_argc);
                }
                ctf_free(cdf, sizeof (ctf_dwfunc_t));
        }

        ctf_dprintf("Trying to free variables\n");
        for (cdv = ctf_list_next(&cup->cu_vars); cdv != NULL; cdv = ndv) {
                ndv = ctf_list_next(cdv);
                ctf_free(cdv->cdv_name, strlen(cdv->cdv_name) + 1);
                ctf_free(cdv, sizeof (ctf_dwvar_t));
        }

        ctf_dprintf("Trying to free bitfields\n");
        for (cdb = ctf_list_next(&cup->cu_bitfields); cdb != NULL; cdb = ndb) {
                ndb = ctf_list_next(cdb);
                ctf_free(cdb, sizeof (ctf_dwbitf_t));
        }

        ctf_dprintf("Trying to clean up dwarf_t: %p\n", cup->cu_dwarf);
        if (cup->cu_dwarf != NULL)
                (void) dwarf_finish(cup->cu_dwarf, &derr);
        cup->cu_dwarf = NULL;
        ctf_close(cup->cu_ctfp);

        cookie = NULL;
        while ((map = avl_destroy_nodes(&cup->cu_map, &cookie)) != NULL) {
                ctf_free(map, sizeof (ctf_dwmap_t));
        }
        avl_destroy(&cup->cu_map);
        cup->cu_errbuf = NULL;
}
```

______***unchanged_portion_omitted_***

```
static int
ctf_dwarf_count_dies(Dwarf_Debug dw, Dwarf_Error *derr, int *ndies,
    char *errbuf, size_t errlen)
{
        int ret;
        Dwarf_Half vers;
        Dwarf_Unsigned nexthdr;

        while ((ret = dwarf_next_cu_header(dw, NULL, &vers, NULL, NULL,
            &nexthdr, derr)) != DW_DLV_NO_ENTRY) {
                if (ret != DW_DLV_OK) {
                        (void) snprintf(errbuf, errlen,
                            "file does not contain valid DWARF data: %s\n",
                            dwarf_errmsg(*derr));
                        return (ECTF_CONVBKERR);
                }

                if (vers != DWARF_VERSION_TWO) {
                        (void) snprintf(errbuf, errlen,
                            "unsupported DWARF version: %d\n", vers);
                        return (ECTF_CONVBKERR);
                }
                *ndies = *ndies + 1;
        }

        if (*ndies == 0) {
                (void) snprintf(errbuf, errlen,
                    "file does not contain valid DWARF data: %s\n",
                    dwarf_errmsg(*derr));
                return (ECTF_CONVBKERR);
        }

        return (0);
}

static int
ctf_dwarf_init_die(int fd, Elf *elf, ctf_cu_t *cup, int ndie, char *errbuf,
    size_t errlen)
{
        int ret;
        Dwarf_Unsigned hdrlen, abboff, nexthdr;
        Dwarf_Half addrsz;
        Dwarf_Unsigned offset = 0;
        Dwarf_Error derr;

        while ((ret = dwarf_next_cu_header(cup->cu_dwarf, &hdrlen, NULL,
            &abboff, &addrsz, &nexthdr, &derr)) != DW_DLV_NO_ENTRY) {
                char *name;
                Dwarf_Die cu, child;

                /* Based on the counting above, we should be good to go */
                VERIFY(ret == DW_DLV_OK);
                if (ndie > 0) {
                        ndie--;
                        offset = nexthdr;
                        continue;
                }

                /*
                 * Compilers are apparently inconsistent. Some emit no DWARF for
                 * empty files and others emit empty compilation unit.
                 */
                cup->cu_voidtid = CTF_ERR;
                cup->cu_longtid = CTF_ERR;
                cup->cu_elf = elf;
                cup->cu_maxoff = nexthdr - 1;
                cup->cu_ctfp = ctf_fdcreate(fd, &ret);
                if (cup->cu_ctfp == NULL)
                if (cup->cu_ctfp == NULL) {
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ret);

                }
                avl_create(&cup->cu_map, ctf_dwmap_comp, sizeof (ctf_dwmap_t),
                    offsetof(ctf_dwmap_t, cdm_avl));
                cup->cu_errbuf = errbuf;
                cup->cu_errlen = errlen;
                bzero(&cup->cu_vars, sizeof (ctf_list_t));
                bzero(&cup->cu_funcs, sizeof (ctf_list_t));
                bzero(&cup->cu_bitfields, sizeof (ctf_list_t));

                if ((ret = ctf_dwarf_die_elfenc(elf, cup, errbuf,
                    errlen)) != 0)
                    errlen)) != 0) {
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ret);
                }

                if ((ret = ctf_dwarf_sib(cup, NULL, &cu)) != 0)
                if ((ret = ctf_dwarf_sib(cup, NULL, &cu)) != 0) {
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ret);

                }
                if (cu == NULL) {
```

```
                        (void) snprintf(errbuf, errlen,
                            "file does not contain DWARF data");
                        return (ECTF_CONVNODEBUG);
                            "file does not contain DWARF data\n");
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ECTF_CONVBKERR);
                }

                if ((ret = ctf_dwarf_child(cup, cu, &child)) != 0)
                if ((ret = ctf_dwarf_child(cup, cu, &child)) != 0) {
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ret);

                }
                if (child == NULL) {
                        (void) snprintf(errbuf, errlen,
                            "file does not contain DWARF data");
                        return (ECTF_CONVNODEBUG);
                            "file does not contain DWARF data\n");
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ECTF_CONVBKERR);
                }

                cup->cu_cuoff = offset;
                cup->cu_cu = child;

                if ((cup->cu_cmh = ctf_merge_init(fd, &ret)) == NULL)
                if ((cup->cu_cmh = ctf_merge_init(fd, &ret)) == NULL) {
                        avl_destroy(&cup->cu_map);
                        ctf_free(cup, sizeof (ctf_cu_t));
                        return (ret);
                }

                if (ctf_dwarf_string(cup, cu, DW_AT_name, &name) == 0) {
                        size_t len = strlen(name) + 1;
                        char *b = basename(name);
                        cup->cu_name = strdup(b);
                        ctf_free(name, len);
                }
                break;
        }

        return (0);
}

/*
 * This is our only recourse to identify a C source file that is missing debug
 * info: it will be mentioned as an STT_FILE, but not have a compile unit entry.
 * (A traditional ctfmerge works on individual files, so can identify missing
 * DWARF more directly, via ctf_has_c_source() on the .o file.)
 *
 * As we operate on basenames, this can of course miss some cases, but it’s
 * better than not checking at all.
 *
 * We explicitly whitelist some CRT components.  Failing that, there’s always
 * the -m option.
 */
static boolean_t
c_source_has_debug(const char *file, ctf_cu_t *cus, size_t nr_cus)
{
        const char *basename = strrchr(file, ’/’);

        if (basename == NULL)
```

```
                basename = file;
        else
                basename++;

        if (strcmp(basename, "common-crt.c") == 0 ||
            strcmp(basename, "gmon.c") == 0 ||
            strcmp(basename, "dlink_init.c") == 0 ||
            strcmp(basename, "dlink_common.c") == 0 ||
            strncmp(basename, "crt", strlen("crt")) == 0 ||
            strncmp(basename, "values-", strlen("values-")) == 0)
                return (B_TRUE);

        for (size_t i = 0; i < nr_cus; i++) {
                if (strcmp(basename, cus[i].cu_name) == 0)
                        return (B_TRUE);
        }

        return (B_FALSE);
}

static int
ctf_dwarf_check_missing(ctf_cu_t *cus, size_t nr_cus, Elf *elf,
ctf_conv_status_t
ctf_dwarf_convert(int fd, Elf *elf, uint_t nthrs, int *errp, ctf_file_t **fpp,
    char *errmsg, size_t errlen)
{
        Elf_Scn *scn, *strscn;
        Elf_Data *data, *strdata;
        GElf_Shdr shdr;
        ulong_t i;

        scn = NULL;
        while ((scn = elf_nextscn(elf, scn)) != NULL) {
                if (gelf_getshdr(scn, &shdr) == NULL) {
                        (void) snprintf(errmsg, errlen,
                            "failed to get section header: %s\n",
                            elf_errmsg(elf_errno()));
                        return (EINVAL);
                }

                if (shdr.sh_type == SHT_SYMTAB)
                        break;
        }

        if (scn == NULL)
                return (0);

        if ((strscn = elf_getscn(elf, shdr.sh_link)) == NULL) {
                (void) snprintf(errmsg, errlen,
                    "failed to get str section: %s\n",
                    elf_errmsg(elf_errno()));
                return (EINVAL);
        }

        if ((data = elf_getdata(scn, NULL)) == NULL) {
                (void) snprintf(errmsg, errlen, "failed to read section: %s\n",
                    elf_errmsg(elf_errno()));
                return (EINVAL);
        }

        if ((strdata = elf_getdata(strscn, NULL)) == NULL) {
                (void) snprintf(errmsg, errlen,
                    "failed to read string table: %s\n",
                    elf_errmsg(elf_errno()));
                return (EINVAL);
        }
```

```

        for (i = 0; i < shdr.sh_size / shdr.sh_entsize; i++) {
                GElf_Sym sym;
                const char *file;
                size_t len;

                if (gelf_getsym(data, i, &sym) == NULL) {
                        (void) snprintf(errmsg, errlen,
                            "failed to read sym %lu: %s\n",
                            i, elf_errmsg(elf_errno()));
                        return (EINVAL);
                }

                if (GELF_ST_TYPE(sym.st_info) != STT_FILE)
                        continue;

                file = (const char *)((uintptr_t)strdata->d_buf + sym.st_name);
                len = strlen(file);
                if (len < 2 || strncmp(".c", &file[len - 2], 2) != 0)
                        continue;

                if (!c_source_has_debug(file, cus, nr_cus)) {
                        (void) snprintf(errmsg, errlen,
                            "file %s is missing debug info\n", file);
                        return (ECTF_CONVNODEBUG);
                }
        }

        return (0);
}

int
ctf_dwarf_convert(int fd, Elf *elf, uint_t nthrs, uint_t flags,
    ctf_file_t **fpp, char *errbuf, size_t errlen)
{
        int err, ret, ndies, i;
        Dwarf_Debug dw;
        Dwarf_Error derr;
        ctf_cu_t *cdies = NULL, *cup;
        workq_t *wqp = NULL;

        if (errp == NULL)
                errp = &err;
        *errp = 0;
        *fpp = NULL;

        ret = dwarf_elf_init(elf, DW_DLC_READ, NULL, NULL, &dw, &derr);
        if (ret != DW_DLV_OK) {
                /*
                 * We may want to expect DWARF data here and fail conversion if
                 * it’s missing. In this case, if we actually have some amount
                 * of DWARF, but no section, for now, just go ahead and create
                 * an empty CTF file.
                 */
                if (ret == DW_DLV_NO_ENTRY ||
                    dwarf_errno(derr) == DW_DLE_DEBUG_INFO_NULL) {
                        (void) snprintf(errbuf, errlen,
                            "file does not contain DWARF data\n");
                        return (ECTF_CONVNODEBUG);
                        *fpp = ctf_create(errp);
                        return (*fpp != NULL ? CTF_CONV_SUCCESS :
                            CTF_CONV_ERROR);
                }

                (void) snprintf(errbuf, errlen,
                    "dwarf_elf_init() failed: %s\n", dwarf_errmsg(derr));
                return (ECTF_CONVBKERR);
                (void) snprintf(errmsg, errlen,
                    "failed to initialize DWARF: %s\n",
                    dwarf_errmsg(derr));
                *errp = ECTF_CONVBKERR;
                return (CTF_CONV_ERROR);
        }

        /*
         * Iterate over all of the compilation units and create a ctf_cu_t for
         * each of them.  This is used to determine if we have zero, one, or
         * multiple dies to convert. If we have zero, that's an error. If
         * there's only one die, that's the simple case.  No merge needed and
         * only a single Dwarf_Debug as well.
         */
        ndies = 0;
        err = ctf_dwarf_count_dies(dw, &derr, &ndies, errbuf, errlen);

        ctf_dprintf("found %d DWARF CUs\n", ndies);

        if (ndies == 0) {
                (void) snprintf(errbuf, errlen,
                    "file does not contain DWARF data\n");
                return (ECTF_CONVNODEBUG);
        ret = ctf_dwarf_count_dies(dw, &derr, &ndies, errmsg, errlen);
        if (ret != 0) {
                *errp = ret;
                goto out;
        }

        (void) dwarf_finish(dw, &derr);
        cdies = ctf_alloc(sizeof (ctf_cu_t) * ndies);
        if (cdies == NULL) {
                return (ENOMEM);
                *errp = ENOMEM;
                return (CTF_CONV_ERROR);
        }

        bzero(cdies, sizeof (ctf_cu_t) * ndies);

        for (i = 0; i < ndies; i++) {
                cup = &cdies[i];
                ret = dwarf_elf_init(elf, DW_DLC_READ, NULL, NULL,
                    &cup->cu_dwarf, &derr);
                if (ret != 0) {
                        ctf_free(cdies, sizeof (ctf_cu_t) * ndies);
                        (void) snprintf(errbuf, errlen,
                        (void) snprintf(errmsg, errlen,
                            "failed to initialize DWARF: %s\n",
                            dwarf_errmsg(derr));
                        return (ECTF_CONVBKERR);
                        *errp = ECTF_CONVBKERR;
                        return (CTF_CONV_ERROR);
                }

                err = ctf_dwarf_init_die(fd, elf, cup, i, errbuf, errlen);
                if (err != 0)
                ret = ctf_dwarf_init_die(fd, elf, &cdies[i], i, errmsg, errlen);
                if (ret != 0) {
                        *errp = ret;
                        goto out;
                }

                cup->cu_doweaks = ndies > 1 ? B_FALSE : B_TRUE;
        }
```

```
        if (!(flags & CTF_ALLOW_MISSING_DEBUG) &&
            (err = ctf_dwarf_check_missing(cdies, ndies,
            elf, errbuf, errlen)) != 0)
                goto out;
        ctf_dprintf("found %d DWARF CUs\n", ndies);

        /*
         * If we only have one compilation unit, there's no reason to use
         * multiple threads, even if the user requested them. After all, they
         * just gave us an upper bound.
         */
        if (ndies == 1)
                nthrs = 1;

        if (workq_init(&wqp, nthrs) == -1) {
                err = errno;
                *errp = errno;
                goto out;
        }

        for (i = 0; i < ndies; i++) {
                cup = &cdies[i];
                ctf_dprintf("adding cu %s: %p, %x %x\n", cup->cu_name,
                    cup->cu_cu, cup->cu_cuoff, cup->cu_maxoff);
                if (workq_add(wqp, cup) == -1) {
                        err = errno;
                        *errp = errno;
                        goto out;
                }
        }

        ret = workq_work(wqp, ctf_dwarf_convert_one, NULL, &err);
        ret = workq_work(wqp, ctf_dwarf_convert_one, NULL, errp);
        if (ret == WORKQ_ERROR) {
                err = errno;
                *errp = errno;
                goto out;
        } else if (ret == WORKQ_UERROR) {
                ctf_dprintf("internal convert failed: %s\n",
                    ctf_errmsg(err));
                    ctf_errmsg(*errp));
                goto out;
        }

        ctf_dprintf("Determining next phase: have %d CUs\n", ndies);
        if (ndies != 1) {
                ctf_merge_t *cmp;

                cmp = ctf_merge_init(fd, &err);
                if (cmp == NULL)
                cmp = ctf_merge_init(fd, &ret);
                if (cmp == NULL) {
                        *errp = ret;
                        goto out;
                }

                ctf_dprintf("setting threads\n");
                if ((err = ctf_merge_set_nthreads(cmp, nthrs)) != 0) {
                if ((ret = ctf_merge_set_nthreads(cmp, nthrs)) != 0) {
                        ctf_merge_fini(cmp);
                        *errp = ret;
                        goto out;
                }

                for (i = 0; i < ndies; i++) {
                        cup = &cdies[i];
                        if ((err = ctf_merge_add(cmp, cup->cu_ctfp)) != 0) {
                        ctf_dprintf("adding cu %s (%p)\n", cup->cu_name,
                            cup->cu_ctfp);
                        if ((ret = ctf_merge_add(cmp, cup->cu_ctfp)) != 0) {
                                ctf_merge_fini(cmp);
                                *errp = ret;
                                goto out;
                        }
                }

                ctf_dprintf("performing merge\n");
                err = ctf_merge_merge(cmp, fpp);
                if (err != 0) {
                ret = ctf_merge_merge(cmp, fpp);
                if (ret != 0) {
                        ctf_dprintf("failed merge!\n");
                        *fpp = NULL;
                        ctf_merge_fini(cmp);
                        *errp = ret;
                        goto out;
                }
                ctf_merge_fini(cmp);
                err = 0;
                *errp = 0;
                ctf_dprintf("successfully converted!\n");
        } else {
                err = 0;
                *errp = 0;
                *fpp = cdies->cu_ctfp;
                cdies->cu_ctfp = NULL;
                ctf_dprintf("successfully converted!\n");
        }

out:
        workq_fini(wqp);
        ctf_dwarf_free_dies(cdies, ndies);
        return (err);
        return (*fpp != NULL ? CTF_CONV_SUCCESS : CTF_CONV_ERROR);
}
```

______***unchanged_portion_omitted_***

```
**********************************************************
    3843 Thu Apr 25 02:20:44 2019
new/usr/src/lib/libctf/common/libctf.h
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2001-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright (c) 2019, Joyent, Inc.
 * Copyright (c) 2015, Joyent, Inc.
 */

/*
 * This header file defines the interfaces available from the CTF debugger
 * library, libctf.  This library provides functions that a debugger can
 * use to operate on data in the Compact ANSI-C Type Format (CTF).  This
 * is NOT a public interface, although it may eventually become one in
 * the fullness of time after we gain more experience with the interfaces.
 *
 * In the meantime, be aware that any program linked with libctf in this
 * release of illumos is almost guaranteed to break in the next release.
 *
 * In short, do not user this header file or libctf for any purpose.
 */

#ifndef _LIBCTF_H
#define _LIBCTF_H

#include <sys/ctf_api.h>
#include <libelf.h>

#ifdef  __cplusplus
extern "C" {
#endif
```

```
/*
 * This flag can be used to enable debug messages.
 */
extern int _libctf_debug;

typedef enum ctf_diff_flag {
        CTF_DIFF_F_IGNORE_INTNAMES = 0x01
} ctf_diff_flag_t;

typedef struct ctf_diff ctf_diff_t;
typedef void (*ctf_diff_type_f)(ctf_file_t *, ctf_id_t, boolean_t, ctf_file_t *,
    ctf_id_t, void *);
typedef void (*ctf_diff_func_f)(ctf_file_t *, ulong_t, boolean_t, ctf_file_t *,
    ulong_t, void *);
typedef void (*ctf_diff_obj_f)(ctf_file_t *, ulong_t, ctf_id_t, boolean_t,
    ctf_file_t *, ulong_t, ctf_id_t, void *);

extern int ctf_diff_init(ctf_file_t *, ctf_file_t *, ctf_diff_t **);
extern uint_t ctf_diff_getflags(ctf_diff_t *);
extern int ctf_diff_setflags(ctf_diff_t *, uint_t);
extern int ctf_diff_types(ctf_diff_t *, ctf_diff_type_f, void *);
extern int ctf_diff_functions(ctf_diff_t *, ctf_diff_func_f, void *);
extern int ctf_diff_objects(ctf_diff_t *, ctf_diff_obj_f, void *);
extern void ctf_diff_fini(ctf_diff_t *);

/*
 * Normally, we return a failure if we find a C-derived compilation unit that
 * lacks DWARF or CTF (as required).  This flag over-rides this check.
 */
#define CTF_ALLOW_MISSING_DEBUG 0x01

#define CTF_CONVERT_F_IGNNONC   0x01
extern ctf_file_t *ctf_fdconvert(int, const char *, uint_t, uint_t, int *,
    char *, size_t);

typedef enum ctf_hsc_ret {
        CHR_ERROR = -1,
        CHR_NO_C_SOURCE = 0,
        CHR_HAS_C_SOURCE = 1
} ctf_hsc_ret_t;

extern ctf_hsc_ret_t ctf_has_c_source(Elf *, char *, size_t);

typedef struct ctf_merge_handle ctf_merge_t;
extern ctf_merge_t *ctf_merge_init(int, int *);
extern int ctf_merge_add(ctf_merge_t *, ctf_file_t *);
extern int ctf_merge_set_nthreads(ctf_merge_t *, const uint_t);
extern int ctf_merge_label(ctf_merge_t *, const char *);
extern int ctf_merge_uniquify(ctf_merge_t *, ctf_file_t *, const char *);
extern int ctf_merge_merge(ctf_merge_t *, ctf_file_t **);
extern int ctf_merge_dedup(ctf_merge_t *, ctf_file_t **);
extern void ctf_merge_fini(ctf_merge_t *);

#define CTF_ELFWRITE_F_COMPRESS         0x1
extern int ctf_elffdwrite(ctf_file_t *, int, int, int);
extern int ctf_elfwrite(ctf_file_t *, const char *, const char *, int);

#ifdef  __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
    1502 Thu Apr 25 02:20:44 2019
new/usr/src/lib/libctf/common/libctf_impl.h
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2019 Joyent, Inc.
 */

#ifndef _LIBCTF_IMPL_H
#define _LIBCTF_IMPL_H

/*
 * Portions of libctf implementations that are only suitable for CTF’s userland
 * library, eg. converting and merging related routines.
 */

#include <libelf.h>
#include <libctf.h>
#include <ctf_impl.h>

#ifdef __cplusplus
extern "C" {
#endif

typedef int (*ctf_convert_f)(int, Elf *, uint_t, uint_t,
typedef enum ctf_conv_status {
        CTF_CONV_SUCCESS        = 0,
        CTF_CONV_ERROR          = 1,
        CTF_CONV_NOTSUP         = 2
} ctf_conv_status_t;

typedef ctf_conv_status_t (*ctf_convert_f)(int, Elf *, uint_t, int *,
    ctf_file_t **, char *, size_t);
extern int ctf_dwarf_convert(int, Elf *, uint_t, uint_t,
extern ctf_conv_status_t ctf_dwarf_convert(int, Elf *, uint_t, int *,
    ctf_file_t **, char *, size_t);

/*
 * Symbol walking
 */
typedef int (*ctf_symtab_f)(const Elf64_Sym *, ulong_t, const char *,
    const char *, boolean_t, void *);
extern int ctf_symtab_iter(ctf_file_t *, ctf_symtab_f, void *);

/*
 * zlib compression routines
```

```
 */
extern int ctf_compress(ctf_file_t *fp, void **, size_t *, size_t *);

extern int ctf_diff_self(ctf_diff_t *, ctf_diff_type_f, void *);

/*
 * Internal debugging aids
 */
extern void ctf_phase_dump(ctf_file_t *, const char *, const char *);
extern void ctf_phase_bump(void);

#ifdef __cplusplus
}
______unchanged_portion_omitted_
```

```
**********************************************************
    3257 Thu Apr 25 02:20:44 2019
new/usr/src/lib/libctf/common/mapfile-vers
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
#

#
# Copyright 2019 Joyent, Inc.
# Copyright 2018 Joyent, Inc.
#

#
# MAPFILE HEADER START
#
# WARNING:  STOP NOW.  DO NOT MODIFY THIS FILE.
# Object versioning must comply with the rules detailed in
#
#       usr/src/lib/README.mapfiles
#
# You should not be making modifications here until you’ve read the most current
# copy of that file. If you need help, contact a gatekeeper for guidance.
#
# MAPFILE HEADER END
#

$mapfile_version 2

# There really should be only one SUNWprivate version.
# Don’t add any more.  Add new private symbols to SUNWprivate_1.2

SYMBOL_VERSION SUNWprivate_1.2 {
    global:
        ctf_add_array;
        ctf_add_const;
        ctf_add_enum;
```

```
        ctf_add_enumerator;
        ctf_add_float;
        ctf_add_forward;
        ctf_add_funcptr;
        ctf_add_function;
        ctf_add_integer;
        ctf_add_label;
        ctf_add_member;
        ctf_add_object;
        ctf_add_pointer;
        ctf_add_restrict;
        ctf_add_struct;
        ctf_add_type;
        ctf_add_typedef;
        ctf_add_union;
        ctf_add_volatile;
        ctf_create;
        ctf_dataptr;
        ctf_delete_type;
        ctf_diff_fini;
        ctf_diff_functions;
        ctf_diff_getflags;
        ctf_diff_init;
        ctf_diff_objects;
        ctf_diff_setflags;
        ctf_diff_types;
        ctf_discard;
        ctf_dup;
        ctf_elffdwrite;
        ctf_elfwrite;
        ctf_enum_value;
        ctf_fdconvert;
        ctf_flags;
        ctf_func_args_by_id;
        ctf_func_info_by_id;
        ctf_function_iter;
        ctf_has_c_source;
        ctf_kind_name;
        ctf_label_info;
        ctf_label_iter;
        ctf_label_topmost;
        ctf_max_id;
        ctf_member_info;
        ctf_merge_add;
        ctf_merge_dedup;
        ctf_merge_fini;
        ctf_merge_init;
        ctf_merge_label;
        ctf_merge_merge;
        ctf_merge_set_nthreads;
        ctf_merge_uniquify;
        ctf_nr_syms;
        ctf_object_iter;
        ctf_parent_file;
        ctf_parent_label;
        ctf_parent_name;
        ctf_set_array;
        ctf_set_root;
        ctf_set_size;
        ctf_string_iter;
        ctf_symbol_name;
        ctf_type_align;
        ctf_type_cmp;
        ctf_type_cname;
        ctf_type_compat;
        ctf_type_pointer;
        ctf_update;
        ctf_write;
} SUNWprivate_1.1;
```

______***unchanged_portion_omitted_***

```
**********************************************************
    1856 Thu Apr 25 02:20:45 2019
new/usr/src/lib/libfru/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License, Version 1.0 only
# (the "License").  You may not use this file except in compliance
# with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#pragma ident   "%Z%%M% %I%     %E% SMI"
#
# lib/libfru/Makefile.targ
#

include ../../Makefile.targ

# build rules for pics.  (access sub directories)
pics/%.o objs/%.o:      ../libfru/%.cc
        $(COMPILE.cc) -o $@ $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)

pics/%.o objs/%.o:      ../libfrureg/%.cc
        $(COMPILE.cc) -o $@ $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)

pics/%.o objs/%.o:      ../libgenutil/%.cc
        $(COMPILE.cc) -o $@ $<
        $(POST_PROCESS_CC_O)
        $(POST_PROCESS_O)

# some special rules for the yacc and lex parser.  (with sub directories)
pics/y.tab.cc pics/y.tab.h: ../libfru/nameSyntaxYacc.y
        $(YACC.y) -d ../libfru/nameSyntaxYacc.y
        $(SED) -f ../libfru/yy-sed y.tab.c > pics/y.tab.cc
```

```
        $(SED) -f ../libfru/yy-sed y.tab.h > pics/y.tab.h
        $(RM) y.tab.c y.tab.h

pics/nameSyntaxYacc.o: pics/y.tab.cc
        $(COMPILE.cc) -I../libfru -Ipics -o $@ pics/y.tab.cc

pics/nameSyntaxLex.o: ../libfru/nameSyntaxLex.l pics/y.tab.h
        $(LEX.l) ../libfru/nameSyntaxLex.l | \
                $(SED) -f ../libfru/yy-lsed > pics/lex.fru.cc
        $(COMPILE.cc) -I../libfru -Ipics -o $@ pics/lex.fru.cc
```

```
**********************************************************
    1158 Thu Apr 25 02:20:45 2019
new/usr/src/lib/libm/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

pics/%.o: $(LIBMDIR)/$(TARGETMACH)/src/%.$(ASSUFFIX)
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

pics/%.o: $(SRCDIR)/C/%.c $(LM_IL)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(SRCDIR)/$(LDBLDIR)/%.c $(LM_IL)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(SRCDIR)/R/%.c $(LM_IL)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(SRCDIR)/complex/%.c $(LM_IL)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(SRCDIR)/m9x/%.c $(LM_IL) $(m9x_IL)
        $(COMPILE.c) $(m9x_IL) -o $@ $<
        $(POST_PROCESS_O)

$(ROOTLIBDIR): $(ROOTFS_LIBDIR)
        $(INS.dir)

$(ROOTLIBDIR64): $(ROOTFS_LIBDIR64)
        $(INS.dir)

include $(SRC)/lib/Makefile.targ
```

```
**********************************************************
    5598 Thu Apr 25 02:20:45 2019
new/usr/src/lib/libmvec/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright 2011 Nexenta Systems, Inc.  All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

LIBMDIR         = $(SRC)/lib/libm

mvecOBJS        = \
                __vTBL_atan1.o \
                __vTBL_atan2.o \
                __vTBL_rsqrt.o \
                __vTBL_sincos.o \
                __vTBL_sincos2.o \
                __vTBL_sqrtf.o \
                __vatan.o \
                __vatan2.o \
                __vatan2f.o \
                __vatanf.o \
                __vc_abs.o \
                __vc_exp.o \
                __vc_log.o \
                __vc_pow.o \
                __vcos.o \
                __vcosbig.o \
                __vcosbigf.o \
                __vcosf.o \
                __vexp.o \
                __vexpf.o \
                __vhypot.o \
                __vhypotf.o \
                __vlog.o \
                __vlogf.o \
                __vpow.o \
                __vpowf.o \
                __vrem_pio2m.o \
                __vrhypot.o \
                __vrhypotf.o \
                __vrsqrt.o \
                __vrsqrtf.o \
                __vsin.o \
                __vsinbig.o \
```

```
                __vsinbigf.o \
                __vsincos.o \
                __vsincosbig.o \
                __vsincosbigf.o \
                __vsincosf.o \
                __vsinf.o \
                __vsqrt.o \
                __vsqrtf.o \
                __vz_abs.o \
                __vz_exp.o \
                __vz_log.o \
                __vz_pow.o \
                vatan2_.o \
                vatan2f_.o \
                vatan_.o \
                vatanf_.o \
                vc_abs_.o \
                vc_exp_.o \
                vc_log_.o \
                vc_pow_.o \
                vcos_.o \
                vcosf_.o \
                vexp_.o \
                vexpf_.o \
                vhypot_.o \
                vhypotf_.o \
                vlog_.o \
                vlogf_.o \
                vpow_.o \
                vpowf_.o \
                vrhypot_.o \
                vrhypotf_.o \
                vrsqrt_.o \
                vrsqrtf_.o \
                vsin_.o \
                vsincos_.o \
                vsincosf_.o \
                vsinf_.o \
                vsqrt_.o \
                vsqrtf_.o \
                vz_abs_.o \
                vz_exp_.o \
                vz_log_.o \
                vz_pow_.o

mvecvisCOBJS    = \
                __vTBL_atan1.o \
                __vTBL_atan2.o \
                __vTBL_rsqrt.o \
                __vTBL_sincos.o \
                __vTBL_sincos2.o \
                __vTBL_sqrtf.o \
                __vcosbig.o \
                __vcosbigf.o \
                __vrem_pio2m.o \
                __vsinbig.o \
                __vsinbigf.o \
                __vsincosbig.o \
                __vsincosbigf.o

mvecvisSOBJS    = \
                __vatan.o \
                __vatan2.o \
                __vatan2f.o \
                __vatanf.o \
                __vcos.o \
                __vcosf.o \
                __vexp.o \
                __vexpf.o \
                __vhypot.o \
                __vhypotf.o \
                __vlog.o \
                __vlogf.o \
                __vpow.o \
                __vpowf.o \
                __vrhypot.o \
                __vrhypotf.o \
                __vrsqrt.o \
                __vrsqrtf.o \
                __vsin.o \
                __vsincos.o \
                __vsincosf.o \
                __vsinf.o \
                __vsqrt.o \
                __vsqrtf.o

mvecvis2COBJS   = \
                __vTBL_sincos.o \
                __vTBL_sincos2.o \
                __vTBL_sqrtf.o \
                __vcosbig.o \
                __vcosbig_ultra3.o \
                __vrem_pio2m.o \
                __vsinbig.o \
                __vsinbig_ultra3.o

mvecvis2SOBJS   = \
                __vcos_ultra3.o \
                __vlog_ultra3.o \
                __vsin_ultra3.o \
                __vsqrtf_ultra3.o

include         $(SRC)/lib/Makefile.lib
include         $(SRC)/lib/Makefile.rootfs
include         $(LIBMDIR)/Makefile.libm.com

LIBS            = $(DYNLIB)
SRCDIR          = ../common/
DYNFLAGS        += $(ZIGNORE)

LINTERROFF      = -erroff=E_FP_DIVISION_BY_ZERO
LINTERROFF      += -erroff=E_FP_INVALID
LINTERROFF      += -erroff=E_BAD_PTR_CAST_ALIGN
LINTERROFF      += -erroff=E_ASSIGMENT_CAUSE_LOSS_PREC
LINTERROFF      += -erroff=E_FUNC_SET_NOT_USED

LINTFLAGS       += $(LINTERROFF)
LINTFLAGS64     += $(LINTERROFF)
LINTFLAGS64     += -errchk=longptr64

CLAGS           += $(LINTERROFF)
CFLAGS64        += $(LINTERROFF)

FLTRPATH_sparc          = $$ORIGIN/cpu/$$ISALIST/libmvec_isa.so.1
FLTRPATH_sparcv9        = $$ORIGIN/../cpu/$$ISALIST/sparcv9/libmvec_isa.so.1
FLTRPATH_i386           = $$ORIGIN/libmvec/$$HWCAP
FLTRPATH                = $(FLTRPATH_$(TARGET_ARCH))

sparc_CFLAGS += -_cc=-W0,-xintrinsic
sparcv9_CFLAGS += -_cc=-W0,-xintrinsic

SRCS_mvec_i386 = ../common/__vsqrtf.c
```

```
SRCS_mvec_sparc = $(SRCS_mvec_i386)

SRCS_mvec_sparcv9 = $(SRCS_mvec_i386)


SRCS_mvec = \
        $(SRCS_mvec_$(TARGETMACH)) \
        ../common/__vTBL_atan1.c \
        ../common/__vTBL_atan2.c \
        ../common/__vTBL_rsqrt.c \
        ../common/__vTBL_sincos.c \
        ../common/__vTBL_sincos2.c \
        ../common/__vTBL_sqrtf.c \
        ../common/__vatan.c \
        ../common/__vatan2.c \
        ../common/__vatan2f.c \
        ../common/__vatanf.c \
        ../common/__vc_abs.c \
        ../common/__vc_exp.c \
        ../common/__vc_log.c \
        ../common/__vc_pow.c \
        ../common/__vcos.c \
        ../common/__vcosbig.c \
        ../common/__vcosbigf.c \
        ../common/__vcosf.c \
        ../common/__vexp.c \
        ../common/__vexpf.c \
        ../common/__vhypot.c \
        ../common/__vhypotf.c \
        ../common/__vlog.c \
        ../common/__vlogf.c \
        ../common/__vpow.c \
        ../common/__vpowf.c \
        ../common/__vrem_pio2m.c \
        ../common/__vrhypot.c \
        ../common/__vrhypotf.c \
        ../common/__vrsqrt.c \
        ../common/__vrsqrtf.c \
        ../common/__vsin.c \
        ../common/__vsinbig.c \
        ../common/__vsinbigf.c \
        ../common/__vsincos.c \
        ../common/__vsincosbig.c \
        ../common/__vsincosbigf.c \
        ../common/__vsincosf.c \
        ../common/__vsinf.c \
        ../common/__vsqrt.c \
        ../common/__vz_abs.c \
        ../common/__vz_exp.c \
        ../common/__vz_log.c \
        ../common/__vz_pow.c \
        ../common/vatan2_.c \
        ../common/vatan2f_.c \
        ../common/vatan_.c \
        ../common/vatanf_.c \
        ../common/vc_abs_.c \
        ../common/vc_exp_.c \
        ../common/vc_log_.c \
        ../common/vc_pow_.c \
        ../common/vcos_.c \
        ../common/vcosf_.c \
        ../common/vexp_.c \
        ../common/vexpf_.c \
        ../common/vhypot_.c \
        ../common/vhypotf_.c \
	../common/vlog_.c \
	../common/vlogf_.c \
	../common/vpow_.c \
	../common/vpowf_.c \
	../common/vrhypot_.c \
	../common/vrhypotf_.c \
	../common/vrsqrt_.c \
	../common/vrsqrtf_.c \
	../common/vsin_.c \
	../common/vsincos_.c \
	../common/vsincosf_.c \
	../common/vsinf_.c \
	../common/vsqrt_.c \
	../common/vsqrtf_.c \
	../common/vz_abs_.c \
	../common/vz_exp_.c \
	../common/vz_log_.c \
	../common/vz_pow_.c

.KEEP_STATE:

all:	$(LIBS)

lint:	lintcheck

pics/%.o: ../$(TARGET_ARCH)/src/%.S
	$(COMPILE.s) -o $@ $<
	$(POST_PROCESS_S_O)
	$(POST_PROCESS_O)

pics/%.o: ../common/$$(CHIP)/%.S
	$(COMPILE.s) -o $@ $<
	$(POST_PROCESS_S_O)
	$(POST_PROCESS_O)
```

```
**********************************************************
    6566 Thu Apr 25 02:20:45 2019
new/usr/src/lib/libnisdb/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2015 Gary Mills
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

LIBRARY= libnisdb.a
VERS=   .2
PROTOCOL_DIR= $(ROOT)/include/rpcsvc
YPTOL_DIR= ./yptol

SED=    sed

OBJECTS = \
        db_entry.o db_entry_c_xdr.o \
        db_item.o db_item_c_xdr.o \
        db_vers.o db_vers_c_xdr.o \
        db_pickle.o \
        db_table.o db_table_c_xdr.o \
        db_index_entry.o db_index_entry_c.o \
        db_index.o db_index_c_xdr.o \
        db_scheme.o db_scheme_c_xdr.o \
        db_query.o db_query_c_xdr.o \
        db_mindex.o db_mindex2.o db_mindex3.o db_mindex_c_xdr.o \
        db_log_entry.o db_log_entry_c_xdr.o \
        db_log.o \
        db.o \
        xdr_nullptr.o \
        db_dictionary.o db_dictionary_c_xdr.o \
        db_dictxdr.o db_dictlog.o db_dictlog_c_xdr.o \
        nis_db.o \
```

```
        nisdb_mt.o nisdb_rw.o \
        nis_hashitem.o \
        ldap_attr.o ldap_cto.o ldap_glob.o ldap_ldap.o \
        ldap_map.o ldap_nisdbquery.o \
        ldap_op.o ldap_parse.o ldap_print.o ldap_ruleval.o \
        ldap_util.o ldap_val.o ldap_xdr.o ldap_scheme.o \
        nis_ldap.o \
        nis_parse_ldap_attr.o nis_parse_ldap_conf.o \
        nis_parse_ldap_map.o nis_parse_ldap_util.o \
        nis_parse_ldap_yp_util.o        \
        dit_access.o \
        dit_access_utils.o \
        lock_update.o \
        map_conv.o \
        map_ctrl.o \
        yptol_utils.o \
        shim.o \
        shim_ancil.o \
        shim_lockmap.o \
        ttl_utils.o \
        update.o \
        shim_changepasswd.o \
        stubs.o

X_FILES= \
    db_c.x             db_dictionary_c.x db_entry_c.x  db_index_c.x     \
    db_index_entry_c.x db_item_c.x       db_log_c.x    db_log_entry_c.x \
    db_mindex_c.x      db_query_c.x      db_scheme_c.x db_table_c.x     \
    db_vers_c.x        db_dictlog_c.x

DERIVED_HEADERS= $(X_FILES:%_c.x=%_c.h) $(X_FILES:%_c.x=%.h)
DERIVED_SOURCES= $(X_FILES:%_c.x=%_c_xdr.c)
DERIVED_FILES= $(DERIVED_HEADERS) $(DERIVED_SOURCES)

# delete the derived files when cleaning up
CLEANFILES += $(DERIVED_FILES)

# include library definitions
include ../Makefile.lib

MAPFILES =      mapfile-vers

SRCS= db_dictionary_c_xdr.c db_dictlog_c_xdr.c db_dictxdr.c \
db_entry_c_xdr.c db_index_c_xdr.c db_index_entry_c_xdr.c db_item_c_xdr.c \
db_log_c_xdr.c db_log_entry_c_xdr.c db_mindex_c_xdr.c db_query_c_xdr.c \
db_scheme_c_xdr.c db_table_c_xdr.c db_vers_c_xdr.c \
$(YPTOL_DIR)/dit_access.c \
$(YPTOL_DIR)/dit_access_utils.c \
$(YPTOL_DIR)/lock_update.c \
$(YPTOL_DIR)/map_conv.c \
$(YPTOL_DIR)/map_ctrl.c \
$(YPTOL_DIR)/yptol_utils.c \
$(YPTOL_DIR)/shim.c \
$(YPTOL_DIR)/shim_ancil.c \
$(YPTOL_DIR)/shim_lockmap.c \
$(YPTOL_DIR)/ttl_utils.c \
$(YPTOL_DIR)/update.c \
$(YPTOL_DIR)/shim_changepasswd.c \
$(YPTOL_DIR)/stubs.c \
#db_c_xdr.c xdr_nullptr.c

# Libnisdb has grown large enough that there may be too many symbols
# wanting 'small' PIC references. Hence, compile some files with
# 'large' PIC references; the parser files are good candidates, since
# they're only used at startup, and performance isn't too important.
LARGESYMS=      nis_parse_ldap_conf.o \
		nis_parse_ldap_attr.o \
		nis_parse_ldap_yp_util.o \
		nis_parse_ldap_map.o \
		ldap_parse.o \
		dit_access.o \
		dit_access_utilis.o \
		map_conv.o \
		map_ctrl.o \
		shim.o \
		shim_ancil.o \
		shim_changepasswd.o \
		update.o \
		stubs.o \
		yptol_utils.o \
		nis_parse_ldap_util.o
LARGEPICS=	$(LARGESYMS:%=pics/%)
$(LARGEPICS) := sparc_C_PICFLAGS = $(C_BIGPICFLAGS)
$(LARGEPICS) := i386_C_PICFLAGS = $(C_BIGPICFLAGS)

# More and stricter semantic checks and lint-like checks
CFLAGS  +=	$(CCVERBOSE)

CPPFLAGS +=	-D_REENTRANT

# Have an unused variable that can’t be removed
UVAR_PICS = \
pics/db_dictionary_c_xdr.o \
pics/db_dictlog_c_xdr.o \
pics/db_entry_c_xdr.o \
pics/db_index_c_xdr.o \
pics/db_item_c_xdr.o \
pics/db_log_entry_c_xdr.o \
pics/db_mindex_c_xdr.o \
pics/db_query_c_xdr.o \
pics/db_scheme_c_xdr.o \
pics/db_table_c_xdr.o \
pics/db_vers_c_xdr.o \
pics/ldap_xdr.o \
pics/dit_access.o \
pics/update.o \
pics/yptol_utils.o \
pics/map_ctrl.o

$(UVAR_PICS) := CERRWARN += -_gcc=-Wno-unused-variable

# Correcting these might alter logic
CERRWARN +=	-_gcc=-Wno-uninitialized
CERRWARN +=	-_gcc=-Wno-switch

# not linted
SMATCH=off

# Extra includes, from yp, for yptol files.
CPPFLAGS += -I$(SRC)/cmd/ypcmd

# Need the path to nis_clnt.h
CLNT_PICS = pics/db_entry_c_xdr.o pics/ldap_xdr.o pics/db_mindex_c_xdr.o

$(CLNT_PICS) := CPPFLAGS += -I../libnsl/nis/gen

LIBS = $(DYNLIB)
ZDEFS=
LDLIBS += -lnsl -lldap -lc

# A number of interfaces are interposed by numerous applications, therefore
# prevent direct binding to anything in libnisdb.  Disable libnisdb from
```

```
# directly binding to itself, but allow libnisdb to directly bind to its
# dependencies (ie. map -Bdirect -> -zdirect).  Ensure lazy loading is
# established (which is enabled automatically with -Bdirect).  In addition,
# libnisdb interposes on socket(), so tag this library as an interposer.
# dependencies (lazyload).
BDIRECT =
DYNFLAGS +=	$(BNODIRECT) $(ZINTERPOSE) $(ZDIRECT) $(ZLAZYLOAD)

all :=		TARGET= all
clean :=	TARGET= clean
clobber :=	TARGET= clobber
install :=	TARGET= install
lint :=		TARGET= lint

.KEEP_STATE:

all:	$(DERIVED_FILES) $(LIBS)

install: all $(ROOTLIBS) $(ROOTLINKS)

lint: $(DERIVED_FILES) .WAIT lintcheck

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

pics/%.o: %.cc
	$(COMPILE.cc) -o $@ $<
	$(POST_PROCESS_O)

objs/%.o pics/%.o profs/%.o: $(YPTOL_DIR)/%.c
	$(COMPILE.c) -o $@ $<
	$(POST_PROCESS_O)

#
# Pattern matching rules that define how to build the derived files.
#
%_c.h: %_c.x
	$(RM) $@
	$(RPCGEN) -DUSINGC -h -o $@ $<

%.h: %_c.x
	$(RM) $@
	$(RPCGEN) -h -o $@ $<

%_c_xdr.c: %_c.x
	$(RM) $@
	$(RPCGEN) -DUSINGC -c -o $@ $<

# rename the xdr_db_free_entry() in the db_table_c_xdr.c.  So the one
# in db_index_entry_c is used instead.
DB_TABLE=db_table_c

$(DB_TABLE)_xdr.c:	$(DB_TABLE).x
	$(RM) $@
	$(RPCGEN) -DUSINGC -c $(DB_TABLE).x |\
	$(SED) -e ’s/^xdr_db_free_entry(/__OBSOLETED_xdr_db_free_entry(/’ > $@

# include library targets
include ../Makefile.targ
```

```
**********************************************************
    6690 Thu Apr 25 02:20:46 2019
new/usr/src/lib/libsqlite/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
# Copyright 2016 Nexenta Systems, Inc.  All rights reserved.
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.
# Copyright 2019 OmniOS Community Edition (OmniOSce) Association.
#

# Make the SO name unlikely to conflict with any other
# libsqlite that might also be found on the system.
LIBRARY = libsqlite-sys.a

VERS = .2.8.15
OBJECTS = \
        attach.o        \
        auth.o          \
        btree.o         \
        btree_rb.o      \
        build.o         \
        copy.o          \
        date.o          \
        delete.o        \
        encode.o        \
        expr.o          \
        func.o          \
        hash.o          \
        insert.o        \
        main.o          \
        opcodes.o       \
        os.o            \
        pager.o         \
        parse.o         \
        pragma.o        \
        printf.o        \
        random.o        \
        select.o        \
        table.o         \
        tokenize.o      \
        trigger.o       \
        update.o        \
        util.o          \
        vacuum.o        \
        vdbe.o          \
        vdbeaux.o       \
        where.o

include $(SRC)/lib/Makefile.lib

# install this library in the root filesystem
include $(SRC)/lib/Makefile.rootfs
```

```
SRCDIR = ../src
TOOLDIR = ../tool
$(DYNLIB) := LDLIBS += -lc
LIBS = $(DYNLIB) $(LINTLIB) $(NATIVERELOC)

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

# generated sources
GENSRC = opcodes.c parse.c

# all sources
SRCS = \
        $(GENSRC) \
        $(SRCDIR)/attach.c      \
        $(SRCDIR)/auth.c        \
        $(SRCDIR)/btree.c       \
        $(SRCDIR)/btree_rb.c    \
        $(SRCDIR)/build.c       \
        $(SRCDIR)/copy.c        \
        $(SRCDIR)/date.c        \
        $(SRCDIR)/delete.c      \
        $(SRCDIR)/encode.c      \
        $(SRCDIR)/expr.c        \
        $(SRCDIR)/func.c        \
        $(SRCDIR)/hash.c        \
        $(SRCDIR)/insert.c      \
        $(SRCDIR)/main.c        \
        $(SRCDIR)/os.c          \
        $(SRCDIR)/pager.c       \
        $(SRCDIR)/pragma.c      \
        $(SRCDIR)/printf.c      \
        $(SRCDIR)/random.c      \
        $(SRCDIR)/select.c      \
        $(SRCDIR)/table.c       \
        $(SRCDIR)/tokenize.c    \
        $(SRCDIR)/update.c      \
        $(SRCDIR)/util.c        \
        $(SRCDIR)/vacuum.c      \
        $(SRCDIR)/vdbe.c        \
        $(SRCDIR)/vdbeaux.c     \
        $(SRCDIR)/where.c       \
        $(SRCDIR)/trigger.c

MYCPPFLAGS = -D_REENTRANT -DTHREADSAFE=1 -DHAVE_USLEEP=1 -I. -I.. -I$(SRCDIR)
CPPFLAGS += $(MYCPPFLAGS)

CERRWARN += -_gcc=-Wno-implicit-function-declaration
CERRWARN += -_gcc=-Wno-uninitialized
CERRWARN += -_gcc=-Wno-unused-function
CERRWARN += -_gcc=-Wno-unused-label

# not linted
SMATCH=off

MAPFILES = ../mapfile-sqlite

# headers generated here
GENHDR = opcodes.h parse.h

# Header files used by all library source files.
#
HDR = \
        $(GENHDR) \
        $(SRCDIR)/btree.h       \
        $(SRCDIR)/config.h      \
        $(SRCDIR)/hash.h        \
        $(SRCDIR)/os.h          \
        ../sqlite.h             \
        $(SRCDIR)/sqliteInt.h   \
        $(SRCDIR)/vdbe.h        \
        $(SRCDIR)/vdbeInt.h

#
# Sources used for test harness
#
TESTSRC = \
        $(SRCDIR)/tclsqlite.c   \
        $(SRCDIR)/btree.c       \
        $(SRCDIR)/func.c        \
        $(SRCDIR)/os.c          \
        $(SRCDIR)/pager.c       \
        $(SRCDIR)/test1.c       \
        $(SRCDIR)/test2.c       \
        $(SRCDIR)/test3.c       \
        $(SRCDIR)/md5.c

TESTOBJS = $(TESTSRC:$(SRCDIR)/%.c=%.o)

TESTCLEAN = $(TESTOBJS) test.db test.tcl test1.bt test2.db testdb

#
# Native variant (needed by cmd/configd)
#
NATIVERELOC = libsqlite-native.o
NATIVEPROGS = testfixture
NATIVEOBJS = $(OBJS:%.o=%-native.o)

NATIVETARGETS = $(NATIVEPROGS) $(NATIVEOBJS) $(NATIVERELOC)

$(NATIVETARGETS) :=     CC = $(NATIVECC)
$(NATIVETARGETS) :=     LD = $(NATIVELD)
$(NATIVETARGETS) :=     CFLAGS = $(NATIVE_CFLAGS)
$(NATIVETARGETS) :=     CPPFLAGS = $(MYCPPFLAGS)
$(NATIVETARGETS) :=     LDFLAGS =
$(NATIVETARGETS) :=     LDLIBS = -lc

$(OBJS) :=              CFLAGS += $(CTF_FLAGS)
$(OBJS) :=              CTFCONVERT_POST = $(CTFCONVERT_O)
$(NATIVEOBJS) :=        CFLAGS += $(CTF_FLAGS)
$(NATIVEOBJS) :=        CTFCONVERT_POST = $(CTFCONVERT_O)

TCLBASE = /usr/sfw
TCLVERS = tcl8.3

testfixture := MYCPPFLAGS += -I$(TCLBASE)/include -DTCLSH -DSQLITE_TEST=1
#
# work around compiler issues
#
testfixture := CFLAGS += \
        -erroff=E_ARRAY_OF_INCOMPLETE \
        -erroff=E_ARG_INCOMPATIBLE_WITH_ARG

testfixture := LDLIBS += -R$(TCLBASE)/lib -L$(TCLBASE)/lib -l$(TCLVERS) -lm -ldl

CLEANFILES += \
        $(NATIVETARGETS) \
        $(TESTCLEAN)     \
        lemon            \
        lemon.o          \
        lempar.c         \
        opcodes.c        \
        opcodes.h        \
```

```
        parse_tmp.c     \
        parse_tmp.h     \
        parse_tmp.out   \
        parse_tmp.y     \
        parse.c         \
        parse.h

ENCODING  = ISO8859


.PARALLEL: $(OBJS) $(OBJS:%.o=%-native.o)
.KEEP_STATE:

# This is the default Makefile target.  The objects listed here
# are what get build when you type just "make" with no arguments.
#
all:            $(LIBS)
install:        all \
                $(ROOTLIBDIR)/$(DYNLIB) \
                $(ROOTLIBDIR)/$(NATIVERELOC)

lint:

all_h: $(GENHDR)

$(ROOTLIBDIR)/$(NATIVERELOC)    := FILEMODE= 644
$(ROOTLINTDIR)/$(LINTLIB)       := FILEMODE= 644

$(ROOTLINK): $(ROOTLIBDIR) $(ROOTLIBDIR)/$(DYNLIB)
        $(INS.liblink)

$(ROOTLINTDIR)/%: ../%
        $(INS.file)

native: $(NATIVERELOC)

$(NATIVERELOC): objs .WAIT $(OBJS:%.o=%-native.o)
        $(LD) -r -o $(NATIVERELOC) $(OBJS:%.o=%-native.o)

opcodes.h: $(SRCDIR)/vdbe.c
        @echo "Generating $@"; \
         $(RM) -f $@ ; \
         echo ’/* Automatically generated file.  Do not edit */’ > $@ ; \
         grep ’^case OP_’ $(SRCDIR)/vdbe.c | \
            sed -e ’s/://’ | \
            $(AWK) ’{printf "#define %-30s %3d\n", $$2, ++cnt}’ >> $@

opcodes.c: $(SRCDIR)/vdbe.c
        @echo "Generating $@"; \
         $(RM) -f $@ ; \
         echo ’/* Automatically generated file.  Do not edit */’ > $@ ; \
         echo ’char *sqliteOpcodeNames[] = { "???", ’ >> $@ ; \
         grep ’^case OP_’ $(SRCDIR)/vdbe.c | \
            sed -e ’s/^.*OP_/  "/’ -e ’s/:.*$$/", /’ >> $@ ; \
         echo ’};’ >> $@

testfixture: FRC
        @if [ -f $(TCLBASE)/include/tcl.h ]; then \
                unset SUNPRO_DEPENDENCIES; \
                echo $(LINK.c) -o testfixture $(TESTSRC) $(LIBRARY) $(LDLIBS) ;\
                exec $(LINK.c) -o testfixture $(TESTSRC) $(LIBRARY) $(LDLIBS) ;\
        else \
                echo "$(TCLBASE)/include/tcl.h: not found."; \
                exit 1; \
        fi
```

```
# Prevent Makefile.lib $(PICS) := from adding PICFLAGS
# by building lemon in a recursive make invocation.
# Otherwise, this target causes a rebuild every time after
# the PICS target builds this one way, then lint the other.
parse.h parse.c : $(SRCDIR)/parse.y $(TOOLDIR)/lemon.c $(TOOLDIR)/lempar.c
        -$(RM) parse_tmp.y lempar.c
        $(CP) $(SRCDIR)/parse.y parse_tmp.y
        $(CP) $(TOOLDIR)/lempar.c lempar.c
        $(MAKE) lemon
        ./lemon parse_tmp.y
        -$(RM) parse.c parse.h
        $(CP) parse_tmp.h parse.h
        $(CP) parse_tmp.c parse.c

lemon: $(TOOLDIR)/lemon.c
        $(NATIVECC) $(NATIVE_CFLAGS) -o $@ $(TOOLDIR)/lemon.c

objs/%-native.o: $(SRCDIR)/%.c $(GENHDR)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/%-native.o: %.c $(GENHDR)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/parse-native.o: parse.c $(GENHDR)
        $(COMPILE.c) -o $@ parse.c
        $(POST_PROCESS_O)

objs/%.o pics/%.o: $(SRCDIR)/%.c $(GENHDR)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/%.o pics/%.o: %.c $(GENHDR)
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

# need direct rules for generated files
objs/opcodes.o pics/opcodes.o: opcodes.c $(GENHDR)
        $(COMPILE.c) -o $@ opcodes.c
        $(POST_PROCESS_O)

objs/parse.o pics/parse.o: parse.c $(GENHDR)
        $(COMPILE.c) -o $@ parse.c
        $(POST_PROCESS_O)

include $(SRC)/lib/Makefile.targ

FRC:
```

```
**********************************************************
    2431 Thu Apr 25 02:20:46 2019
new/usr/src/lib/libtnfctl/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

LIBRARY=        libtnfctl.a
VERS=           .1
OBJECTS.c=      open.o          \
                prb_child.o     \
                prb_shmem.o     \
                prb_proc.o      \
                prb_lmap.o      \
                prb_rtld.o      \
                prb_findexec.o  \
                prb_status.o    \
                util.o          \
                traverse.o      \
                sym.o           \
                elf.o           \
                continue.o      \
                checklib.o      \
                probes.o        \
                probes_ext.o    \
                close.o         \
                targmem.o       \
                comb.o          \
                kernel_int.o    \
                kernel.o        \
                internal.o      \
                status.o
```

```
OBJECTS.s=      $(MACH)_assm.o

OBJECTS=        $(OBJECTS.c) $(OBJECTS.s)

include ../../Makefile.lib

# We omit $(OBJECTS.s:%.o=%.s) in the next line, because lint no like
SRCS= $(OBJECTS.c:%.o=../%.c)

LIBS=           $(DYNLIB)

HDRS=           tnfctl.h
ROOTHDRDIR=     $(ROOT)/usr/include/tnf
ROOTHDRS=       $(HDRS:%=$(ROOTHDRDIR)/%)
CHECKHDRS=      $(HDRS:%.h=%.check)
$(ROOTHDRS) :=  FILEMODE = 0644
CHECKHDRS =     $(HDRS:%.h=%.check)

LDLIBS +=       -lc -lelf

CPPFLAGS +=     -I$(SRC)/lib/libtnfprobe -D_REENTRANT -I$(SRC)/cmd/sgs/include

LINTFLAGS +=    -y

CERRWARN +=     -_gcc=-Wno-uninitialized
CERRWARN +=     -_gcc=-Wno-empty-body
CERRWARN +=     -_gcc=-Wno-parentheses

# not linted
SMATCH=off

ASFLAGS +=      -P

$(ROOTHDRS) :=  FILEMODE = 644

.KEEP_STATE:

all: $(LIBS)

install_h: $(ROOTHDRDIR) $(ROOTHDRS)

lint:
        $(LINT.c) $(SRCS)

check: $(CHECKHDRS)

$(ROOTLIBDIR) $(ROOTHDRDIR):
        $(INS.dir)

$(ROOTHDRDIR)/% : %
        $(INS.file)

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

objs/%.o pics/%.o: ../%.s
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

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


%.o:            ../%.c
```

```
	$(COMPILE.c) -o $@ $<
	$(POST_PROCESS_O)

include ../../Makefile.targ
```

```
**********************************************************
    2616 Thu Apr 25 02:20:46 2019
new/usr/src/lib/libtnfprobe/Makefile.com
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright (c) 1997, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

LIBRARY=        libtnfprobe.a
VERS=           .1
OBJECTS.c=      tnf_buf.o       \
                trace_init.o    \
                trace_funcs.o   \
                debug_funcs.o   \
                probe_mem.o     \
                tnf_args.o      \
                tnf_trace.o     \
                probe_cntl.o

UFSDIR=         $(SRC)/uts/common/tnf
UFSOBJS=        tnf_writer.o tnf_probe.o

OBJECTS.s=      $(MACH)_locks.o

OBJECTS=        $(OBJECTS.c) $(UFSOBJS) $(OBJECTS.s)

include ../../Makefile.lib

# We omit $(OBJECTS.s:%.o=%.s) in the next line, because lint no like
SRCS= $(OBJECTS.c:%.o=../%.c) $(UFSOBJS:%.o=$(UFSDIR)/%.c)

LIBS=           $(DYNLIB)

DYNFLAGS +=     $(ZINTERPOSE)
```

```
HDRS=           com.h writer.h probe.h
ROOTHDRDIR=     $(ROOT)/usr/include/tnf
ROOTHDRS=       $(HDRS:%=$(ROOTHDRDIR)/%)
CHECKHDRS=      $(HDRS:%.h=%.check)
$(ROOTHDRS) :=  FILEMODE = 0644
CHECKHDRS =     $(HDRS:%.h=%.check)

# Include .. first to pick up tnf_trace.h in current dir, Include UFSDIR to
#       pick up tnf_types.h
CPPFLAGS +=     -I.. -I$(UFSDIR) -D_REENTRANT -D_TNF_LIBRARY

LINTFLAGS +=    -y

CERRWARN +=     -_gcc=-Wno-unused-variable
CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-uninitialized

# not linted
SMATCH=off

$(ROOTHDRS) :=  FILEMODE = 644

LDLIBS += -lc

.KEEP_STATE:

all: $(LIBS)

install_h: $(ROOTHDRDIR) $(ROOTHDRS)

lint:
        $(LINT.c) $(SRCS)

check: $(CHECKHDRS)

$(ROOTLIBDIR) $(ROOTHDRDIR):
        $(INS.dir)

$(ROOTHDRDIR)/% : %
        $(INS.file)

#ASFLAGS=       $(AS_PICFLAGS) -P -D_SYS_SYS_S -D_LOCORE -D_ASM -DPIC -DLOCORE $
ASFLAGS=        -P -D_SYS_SYS_S -D_LOCORE -D_ASM -DPIC -DLOCORE $(CPPFLAGS)
BUILD.s=        $(AS) $(ASFLAGS) $< -o $@

objs/%.o pics/%.o: ../%.s
                $(COMPILE.s) -o $@ $<
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

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

pics/%.o objs/%.o:      $(UFSDIR)/%.c
                $(COMPILE.c) -o $@ $<
                $(POST_PROCESS_O)

include ../../Makefile.targ
```

```
**********************************************************
    2375 Thu Apr 25 02:20:46 2019
new/usr/src/lib/libumem/Makefile.targ
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright (c) 2016, Chris Fraire <cfraire@me.com>.
# Copyright (c) 2019, Joyent, Inc.
#

#
# We build each flavor in a separate make invocation to improve clarity(!) in
# Makefile.com.  The subordinate makes have $(CURTYPE) set to indicate the
# flavor they’re supposed to build.  This causes the correct set of source
# files and compiler and linker flags to be selected.
#

install: $(TYPES:%=install.%)

install.library: all.library $(INSTALL_DEPS_library)
install.standalone: all.standalone $(INSTALL_DEPS_standalone)

all: $(TYPES:%=all.%)

$(TYPES:%=all.%):
        @$(MAKE) $@.targ CURTYPE=$(@:all.%=%)

all.library.targ: $(LIBS)
all.standalone.targ: $(STANDLIBRARY)

lint: $(TYPES:%=lint.%)

$(TYPES:%=lint.%):
        @$(MAKE) lintcheck CURTYPE=$(@:lint.%=%)

$(STANDLIBRARY): $(OBJS) $(LINKTEST_OBJ)
```

```
        $(LD) $(BREDUCE) $(ZDEFS) $(LDFLAGS) -o $@.linktest $(OBJS) $(LINKTEST_O
        rm $@.linktest
        $(LD) $(LDFLAGS) -o $@ $(OBJS)

clobber: $(TYPES:%=clobber.%)

$(TYPES:%=clobber.%):
        @$(MAKE) clobber.targ CURTYPE=$(@:clobber.%=%)

clobber.targ: clean
        -$(RM) $(CLOBBERTARGFILES)

# include library targets
include ../../Makefile.targ

$(PICS): pics
$(OBJS): objs

objs/%.o pics/%.o: $(ISASRCDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

objs/%.o pics/%.o: $(ISASRCDIR)/%.s
        $(COMPILE.s) -o $@ $<
        $(POST_PROCESS_S_O)
        $(POST_PROCESS_O)

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

# install rule for lint library target
$(ROOTLINTDIR)/%: ../common/%
        $(INS.file)
```

```
**********************************************************
    3038 Thu Apr 25 02:20:47 2019
new/usr/src/lib/pkcs11/libsoftcrypto/amd64/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2017 Jason King
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.
#
# Copyright (c) 2018, Joyent, Inc.

LIBRARY =       libsoftcrypto.a
VERS =          .1

include         ../Makefile.com

AES_PSM_OBJS =  aes_amd64.o aes_intel.o aeskey.o
ARCFOUR_PSM_OBJS = arcfour-x86_64.o
BIGNUM_PSM_OBJS = bignum_amd64.o bignum_amd64_asm.o
MODES_PSM_OBJS = gcm_intel.o

include         $(SRC)/lib/Makefile.lib
include         $(SRC)/lib/Makefile.lib.64

CERRWARN +=     -_gcc=-Wno-type-limits
CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-uninitialized

# not linted
SMATCH=off

AES_PSM_SRC =   $(AES_DIR)/$(MACH64)/aes_amd64.s \
                $(AES_DIR)/$(MACH64)/aes_intel.s \
                $(AES_DIR)/$(MACH64)/aeskey.c
ARCFOUR_PSM_SRC = arcfour-x86_64.s
```

```
BIGNUM_PSM_SRC = $(BIGNUM_DIR)/$(MACH64)/bignum_amd64.c \
                $(BIGNUM_DIR)/$(MACH64)/bignum_amd64_asm.s

MODES_PSM_SRC = $(MODES_DIR)/$(MACH64)/gcm_intel.s

# Sources need to be redefined after Makefile.lib inclusion.
SRCS =          $(AES_SRC) $(ARCFOUR_SRC) $(BIGNUM_SRC) $(BLOWFISH_SRC) \
                $(DES_SRC) $(MODES_SRC) $(DH_SRC) $(DSA_SRC) $(RSA_SRC) \
                $(PAD_SRC)

SRCDIR =        $(SRC)/lib/pkcs11/libsoftcrypto/common

LIBS =          $(DYNLIB) $(LINTLIB)
MAPFILEDIR =    ../common

CFLAGS +=       -xO4 -xcrossfile
CPPFLAGS +=     -I$(CRYPTODIR) -I$(UTSDIR) -D_POSIX_PTHREAD_SEMANTICS
ASFLAGS +=      $(AS_PICFLAGS) -P -D__STDC__ -D_ASM
BIGNUM_FLAGS += -DPSR_MUL
CLEANFILES +=   arcfour-x86_64.s
LDLIBS +=       -lcryptoutil -lc
LINTFLAGS64 +=  $(EXTRA_LINT_FLAGS)

.KEEP_STATE:

all:            $(LIBS)

lint:           $(SRCS)

install:        all $(ROOTLIBS64) $(ROOTLINKS64) $(ROOTLINT64)

pics/%.o: $(AES_DIR)/$(MACH64)/%.c
                $(COMPILE.c) $(AES_FLAGS) -o $@ $<
                $(POST_PROCESS_O)

pics/%.o: $(AES_DIR)/$(MACH64)/%.s
                $(COMPILE.s) $(AES_FLAGS) -o $@ $<
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

pics/%.o: $(BIGNUM_DIR)/$(MACH64)/%.c
                $(COMPILE.c) $(BIGNUM_FLAGS) -o $@ $<
                $(POST_PROCESS_O)

pics/%.o: $(BIGNUM_DIR)/$(MACH64)/%.s
                $(COMPILE64.s) $(BIGNUM_FLAGS) -o $@ $<
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

pics/%.o: $(MODES_DIR)/$(MACH64)/%.s
                $(COMPILE64.s) $(MODES_FLAGS) -o $@ $<
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)

include         ../Makefile.targ

arcfour-x86_64.s: $(ARCFOUR_DIR)/amd64/arcfour-x86_64.pl
                $(PERL) $? $@

pics/%.o: arcfour-x86_64.s
                $(COMPILE64.s) $(ARCFOUR_FLAGS) -o $@ $<
                $(POST_PROCESS_S_O)
                $(POST_PROCESS_O)
```

```
**********************************************************
    1772 Thu Apr 25 02:20:47 2019
new/usr/src/lib/pkcs11/libsoftcrypto/capabilities/intel/i386/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#

BIGNUM_PSM_OBJS =	bignum_i386.o bignum_i386_asm.o

include		../../Makefile.com

# Redefine the objects required for this capabilities group.
OBJECTS =	$(BIGNUM_OBJS)

include		$(SRC)/lib/Makefile.lib

AS_CPPFLAGS +=	-D__STDC__ -D_ASM -DPIC -D_REENTRANT -D$(MACH)
ASFLAGS +=	$(AS_PICFLAGS) -P
CFLAGS +=	$(CCVERBOSE)
CPPFLAGS +=	-I$(CRYPTODIR) -I$(UTSDIR) -D_POSIX_PTHREAD_SEMANTICS
BIGNUM_FLAGS += -DMMX_MANAGE -DHWCAP -DPSR_MUL -DNO_BIG_ONE -DNO_BIG_TWO

CERRWARN +=	 -_gcc=-Wno-unused-variable

.KEEP_STATE:

all install:	$(SYMCAP)

include		../../Makefile.targ

pics/bignum_i386.o: $(BIGNUM_DIR)/i386/bignum_i386.c
		$(COMPILE.c) $(BIGNUM_FLAGS) -o $@ \
		    $(BIGNUM_DIR)/i386/bignum_i386.c
```

```
		$(POST_PROCESS_O)

pics/bignum_i386_asm.o: $(BIGNUM_DIR)/i386/bignum_i386_asm.s
		$(COMPILE.s) -o $@ $(BIGNUM_DIR)/i386/bignum_i386_asm.s
		$(POST_PROCESS_S_O)
		$(POST_PROCESS_O)
```

```
**********************************************************
     926 Thu Apr 25 02:20:47 2019
new/usr/src/test/crypto-tests/tests/common/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
# Copyright (c) 2019, Joyent, Inc.
#

include $(SRC)/cmd/Makefile.cmd
include $(SRC)/cmd/Makefile.ctf
include $(SRC)/test/Makefile.com

OBJS = testfuncs.o cryptotest_pkcs.o cryptotest_kcf.o
SRCS = $(OBJS:%.o=%.c)

CPPFLAGS += -D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -I$(SRC)/common/crypto/

all install: $(OBJS)

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

lint:

clobber clean:
        -$(RM) $(OBJS)
```

```
**********************************************************
    1445 Thu Apr 25 02:20:47 2019
new/usr/src/test/util-tests/runfiles/default.run
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright (c) 2012 by Delphix. All rights reserved.
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
# Copyright 2014 Nexenta Systems, Inc. All rights reserved.
# Copyright 2019, Joyent, Inc.
# Copyright 2017 Joyent, Inc.
#

[DEFAULT]
pre =
verbose = False
quiet = False
timeout = 60
post =
outputdir = /var/tmp/test_results

[/opt/util-tests/tests/printf_test]
[/opt/util-tests/tests/allowed-ips]
[/opt/util-tests/tests/set-linkprop]
[/opt/util-tests/tests/libsff/libsff]

[/opt/util-tests/tests/xargs_test]

[/opt/util-tests/tests/mergeq/mqt]
[/opt/util-tests/tests/mergeq/wqt]

[/opt/util-tests/tests/libnvpair_json]
tests = ['json_00_blank', 'json_01_boolean', 'json_02_numbers',
    'json_03_empty_arrays', 'json_04_number_arrays', 'json_05_strings',
    'json_06_nested', 'json_07_nested_arrays']

[/opt/util-tests/tests/grep_test]
[/opt/util-tests/tests/date_test]
[/opt/util-tests/tests/chown_test]

[/opt/util-tests/tests/demangle]
tests = ['afl-fast', 'gcc-libstdc++', 'llvm-stdcxxabi']

[/opt/util-tests/tests/ctf]
pre = precheck
tests = [ 'ctftest' ]
```

```
**********************************************************
    3390 Thu Apr 25 02:20:48 2019
new/usr/src/test/util-tests/tests/ctf/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2019, Joyent, Inc.
#

include $(SRC)/Makefile.master

ROOTOPTPKG = $(ROOT)/opt/util-tests
TESTDIR = $(ROOTOPTPKG)/tests/ctf

SCRIPTS =       precheck.ksh \
                ctftest.ksh \
                ctftest-convert-non-c.ksh \
                ctftest-convert-no-dwarf.ksh \
                ctftest-merge-no-ctf.ksh \
SCRIPTS =       ctftest.ksh

TESTS =         test-float.c \
                test-reference.c \
                test-int.c \
                test-array.c \
                test-enum.c \
                test-forward.c \
                test-sou.c \
                test-function.c \
                test-merge-static/Makefile.ctftest \
                test-merge-static/test-a.c \
                test-merge-static/test-b.c \
                test-merge-static/test-c.c \
                test-merge-static/test-d.c \
                test-merge-static/test-main.c \
                test-merge-forward/Makefile.ctftest \
                test-merge-forward/test-impl.c \
                test-merge-forward/test-merge.c \
                test-merge-dedup/Makefile.ctftest \
                test-merge-dedup/test-merge-1.c \
                test-merge-dedup/test-merge-2.c \
                test-merge-dedup/test-merge-3.c \
                test-merge-dedup/test-merge-dedup.c \
                test-merge-reduction/Makefile.ctftest \
                test-merge-reduction/mapfile-vers \
                test-merge-reduction/test-global.c \
                test-merge-reduction/test-scoped.c \
```

```
                test-merge-weak/Makefile.ctftest \
                test-merge-weak/test-merge-weak.c \
                test-weak.c \
                Makefile.ctftest.com

MAKEDIRS =      test-merge-static \
                test-merge-forward \
                test-merge-dedup \
                test-merge-reduction \
                test-merge-weak

CHECKS =        check-float-32 \
                check-float-64 \
                check-int-32 \
                check-int-64 \
                check-reference \
                check-array \
                check-enum \
                check-sou-32 \
                check-sou-64 \
                check-forward-32 \
                check-forward-64 \
                check-function \
                check-merge-static \
                check-merge-forward-32 \
                check-merge-forward-64 \
                check-merge-dedup \
                check-merge-reduction \
                check-merge-weak \
                check-weak

COMMON_OBJS =   check-common.o
ALL_OBJS =      $(CHECKS:%=%.o) $(CHECKS:%-32=%.32.o) $(CHECKS:%-64=%.64.o) $(CO

ROOTTESTS =     $(TESTS:%=$(TESTDIR)/%)
ROOTMAKEDIRS =  $(MAKEDIRS:%=$(TESTDIR)/%)
ROOTCHECKS =    $(CHECKS:%=$(TESTDIR)/%)
ROOTSCRIPTS =   $(SCRIPTS:%.ksh=$(TESTDIR)/%)

ROOTTESTS       := FILEMODE =   0444
ROOTCHECKS      := FILEMODE =   0555
ROOTSCRIPTS     := FILEMODE =   0555

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

LDLIBS +=       -lctf

check-merge-static :=   LDLIBS += -lelf

all: $(CHECKS)

install: all $(ROOTTESTS) $(ROOTCHECKS) $(ROOTSCRIPTS)

$(CHECKS): $(COMMON_OBJS)

clean:
        $(RM) $(ALL_OBJS)

clobber: clean
        $(RM) $(CHECKS)

$(ROOTTESTS): $(TESTDIR) $(ROOTMAKEDIRS) $(TESTS)
$(ROOTCHECKS): $(TESTDIR) $(CHECKS)
$(ROOTSCRIPTS): $(TESTDIR) $(SCRIPTS)
```

```
$(TESTDIR):
        $(INS.dir)

$(ROOTMAKEDIRS):
        $(INS.dir)

$(TESTDIR)/%: %
        $(INS.file)

$(TESTDIR)/%: %.ksh
        $(INS.rename)

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

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

%.64.o: %.c
        $(COMPILE.c) -DTARGET_LP64 -o $@ $<
        $(POST_PROCESS_O)

%-32: %.32.o
        $(LINK.c) -o $@ $< $(COMMON_OBJS) $(LDLIBS)
        $(POST_PROCESS)

%-64: %.64.o
        $(LINK.c) -o $@ $< $(COMMON_OBJS) $(LDLIBS)
        $(POST_PROCESS)

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

```
**********************************************************
    3034 Thu Apr 25 02:20:48 2019
new/usr/src/test/util-tests/tests/ctf/ctftest-convert-no-dwarf.ksh
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#!/usr/bin/ksh
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#
# Copyright (c) 2019, Joyent, Inc.
#

set -e

result=0

progname=$(basename $0)

fail()
{
        echo "Failed: $*" 2>&1
        result=1
}

fail_no_debug()
{
        cmd="$@"
        set +e
        out=$($ctf_convert $cmd 2>&1)

        if [[ $? -eq 0 ]]; then
                fail "$cmd succeeded but should have failed"
                set -e
                return;
        fi

        set -e

        if echo "$out" | grep "is missing debug info" >/dev/null; then
                return;
        fi

        if echo "$out" | grep "does not contain DWARF data" >/dev/null; then
                return;
        fi
        fail "$cmd: incorrect output $out"
}

has_ctf()
{
        for f in "$@"; do
```

```
                if ! elfdump -c -N .SUNW_ctf "$f" |
                    grep '.SUNW_ctf' >/dev/null; then
                        fail "$f lacks CTF section"
                        return
                fi
        done
}

cat <<EOF >file1.c
#include <stdio.h>
struct foo { int a; };
int main(void) { struct foo foo = { 4 }; printf("%d\n", foo.a); }
EOF

cat <<EOF >file2.c
#include <stdio.h>
char myfunc(int a) { printf("%d\n", a); }
EOF

cat <<EOF >file3.cc
struct bar { char *tar; };
void mycxxfunc(char *c) { c[0] = '9'; };
EOF

cat <<EOF >file4.s
.globl caller
.type caller,@function
caller:
        movl 4(%ebp), %eax
        ret
EOF

echo "$progname: An empty file should fail conversion due to no DWARF"
echo >emptyfile.c

$ctf_cc -c -o emptyfile.o emptyfile.c
fail_no_debug emptyfile.o
$ctf_cc -c -o emptyfile.o emptyfile.c
$ctf_convert -m emptyfile.o

$ctf_cc $ctf_debugflags -c -o emptyfile.o emptyfile.c
fail_no_debug emptyfile.o
$ctf_cc $ctf_debugflags -c -o emptyfile.o emptyfile.c
$ctf_convert -m emptyfile.o

echo "$progname: A file missing DWARF should fail conversion"

$ctf_cc -c -o file1.o file1.c
fail_no_debug file1.o
$ctf_cc -c -o file1.o file1.c
$ctf_convert -m file1.o

echo "$progname: A binary with DWARF but 0 debug dies should fail conversion"

$ctf_cc -o mybin file1.c
fail_no_debug mybin
$ctf_cc -o mybin file1.c
$ctf_convert -m mybin

echo "$progname: One C file missing DWARF should fail ctfconvert"

$ctf_cc -c -o file1.o file1.c
$ctf_cc $ctf_debugflags -c -o file2.o file2.c
ld -r -o files.o file2.o file1.o
fail_no_debug files.o
ld -r -o files.o file2.o file1.o
```

```
$ctf_convert -m files.o
has_ctf files.o

echo "$progname: One .cc file missing DWARF should pass"

$ctf_cc $ctf_debugflags -c -o file1.o file1.c
$ctf_cc $ctf_debugflags -c -o file2.o file2.c
$ctf_cxx -c -o file3.o file3.cc
ld -r -o files.o file1.o file2.o file3.o
$ctf_convert files.o
has_ctf files.o

echo "$progname: One .s file missing DWARF should pass"
$ctf_cc $ctf_debugflags -c -o file1.o file1.c
$ctf_cc $ctf_debugflags -c -o file2.o file2.c
$ctf_as -o file4.o file4.s
$ctf_cc -o mybin file1.o file2.o file4.o
$ctf_convert mybin
has_ctf mybin

echo "result is $result"
exit $result
```

```
**********************************************************
    1861 Thu Apr 25 02:20:48 2019
new/usr/src/test/util-tests/tests/ctf/ctftest-convert-non-c.ksh
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#!/usr/bin/ksh
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#
# Copyright (c) 2019, Joyent, Inc.
#

set -e

result=0

progname=$(basename $0)

fail()
{
        echo "Failed: $*" 2>&1
        result=1
}

fail_non_c()
{
        cmd="$@"
        set +e
        out=$($ctf_convert $cmd 2>&1)

        if [[ $? -eq 0 ]]; then
                fail "$cmd succeeded but should have failed"
                set -e
                return;
        fi

        set -e

        if ! echo "$out" | \
            grep "No C source to convert from" >/dev/null; then
                fail "$cmd: incorrect output $out"
                return;
        fi
}

no_ctf()
{
        for f in "$@"; do
                if elfdump -c -N .SUNW_ctf "$f" |
                    grep '.SUNW_ctf' >/dev/null; then
                        fail "$f has CTF section"
```

```
                        return
                fi
        done
}

cat <<EOF >file1.c
#include <stdio.h>
struct foo { int a; };
int main(void) { struct foo foo = { 4 }; printf("%d\n", foo.a); }
EOF

cat <<EOF >file2.cc
struct bar { char *tar; };
void mycxxfunc(char *c) { c[0] = '9'; };
EOF

cat <<EOF >file3.s
.globl caller
.type caller,@function
caller:
        movl 4(%ebp), %eax
        ret
EOF

echo "$progname: ctfconvert should fail on a .cc-derived object"
$ctf_cxx -c -o file2.o file2.cc
fail_non_c file2.o
$ctf_cxx -c -o file2.o file2.cc
$ctf_convert -i file2.o

echo "$progname: ctfconvert shouldn’t process .cc-derived DWARF"
$ctf_cc $DEBUGFLAGS -c -o file2.o file2.cc
$ctf_convert -i file2.o
no_ctf file2.o

echo "$progname: ctfconvert should fail on a .s-derived object"
$ctf_as -o file3.o file3.s
fail_non_c file3.o
$ctf_as -o file3.o file3.s
$ctf_convert -i file3.o
no_ctf file3.o

echo "result is $result"
exit $result
```

****
2489 Thu Apr 25 02:20:48 2019
new/usr/src/test/util-tests/tests/ctf/ctftest-merge-no-ctf.ksh
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
****

```
#!/usr/bin/ksh
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#
# Copyright (c) 2019, Joyent, Inc.
#

set -e

result=0

progname=$(basename $0)

fail()
{
        echo "Failed: $*" 2>&1
        result=1
}

fail_no_ctf()
{
        cmd="$@"
        set +e
        out=$($cmd 2>&1)

        if [[ $? -eq 0 ]]; then
                fail "$cmd succeeded but should have failed"
                set -e
                return;
        fi

        set -e

        if ! echo "$out" | \
            grep "File does not contain CTF data" >/dev/null; then
                fail "$cmd: incorrect output $out"
                return;
        fi
}

has_ctf()
{
        for f in "$@"; do
                if ! elfdump -c -N .SUNW_ctf "$f" |
                    grep ’.SUNW_ctf’ >/dev/null; then
                        fail "$f lacks CTF section"
```

```
                        return
                fi
        done
}

cat <<EOF >file1.c
#include <stdio.h>
struct foo { int a; };
struct foo foos[400];
int main(void) { struct foo foo = { 4 }; printf("%d\n", foo.a); }
EOF

cat <<EOF >file2.c
#include <stdio.h>
struct foo { char b; float c; };
struct foo stuff[90];
char myfunc(int a) { printf("%d\n", a); }
EOF

cat <<EOF >file3.cc
struct bar { char *tar; };
void mycxxfunc(char *c) { c[0] = ’9’; };
EOF

cat <<EOF >file4.s
.globl caller
.type caller,@function
caller:
        movl 4(%ebp), %eax
        ret
EOF

echo "$progname: ctfmerge should fail if one C-source lacks CTF"

$ctf_cc $ctf_debugflags -c -o file1.o file1.c
$ctf_convert file1.o
$ctf_cc -c -o file2.o file2.c
ld -r -o files.o file2.o file1.o
fail_no_ctf $ctf_merge -o files.o file2.o file1.o
ld -r -o files.o file2.o file1.o
$ctf_merge -m -o files.o file2.o file1.o
has_ctf files.o
$ctf_cc -o mybin file2.o file1.o
fail_no_ctf $ctf_merge -o mybin file2.o file1.o
$ctf_cc -o mybin file2.o file1.o
$ctf_merge -m -o mybin file2.o file1.o


echo "$progname: ctfmerge should allow a .cc file to lack CTF"
$ctf_cxx -c -o file3.o file3.cc
ld -r -o files.o file1.o file3.o
$ctf_merge -o files.o file1.o file3.o
ld -r -o files.o file1.o file3.o
$ctf_merge -m -o files.o file1.o file3.o

echo "$progname: ctfmerge should allow an .s file to lack CTF"
$ctf_as -o file4.o file4.s
ld -r -o files.o file4.o file1.o
$ctf_merge -o files.o file4.o file1.o
ld -r -o files.o file4.o file1.o
$ctf_merge -m -o files.o file4.o file1.o

echo "result is $result"
exit $result
```

```
**********************************************************
    5845 Thu Apr 25 02:20:48 2019
new/usr/src/test/util-tests/tests/ctf/ctftest.ksh
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#!/usr/bin/ksh
#
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet at
# http://www.illumos.org/license/CDDL.
#

#
# Copyright (c) 2019, Joyent, Inc.
#

#
# Run all of the various CTF tests
#

unalias -a
#set -o xtrace

if [[ -z "$TMPDIR" ]]; then
        TMPDIR="/tmp"
fi



ctf_arg0=$(basename $0)
ctf_root=$(cd $(dirname $0) && echo $PWD)
ctf_tests=
ctf_cc="gcc"
ctf_cxx="g++"
ctf_as="as"
ctf_compiler="gcc"
ctf_convert="ctfconvert"
ctf_merge="ctfmerge"
ctf_debugflags="-gdwarf-2 "
ctf_mach32="-m32"
ctf_mach64="-m64"
ctf_32cflags="$ctf_mach32 $ctf_debugflags"
ctf_64cflags="$ctf_mach64 $ctf_debugflags"
ctf_temp="$TMPDIR/ctftest.$$.o"
ctf_makefile="Makefile.ctftest"
ctf_nerrs=0

usage()
{
        typeset msg="$*"
        [[ -z "$msg" ]] || echo "$msg" >&2
        cat <<USAGE >&2
Usage: $ctf_arg0 [-a as] [-c cc] [-C CC] [-g flags] [-m ctfmerge] [-t ctfconvert
```

```
Usage: $ctf_arg0  [-c compiler] [-g flags] [-m ctfmerge] [-t ctfconvert]

        Runs the CTF test suite

        -a assembler            Use the specified assembler, defaults to 'as'
        -c compiler             Use the specified C compiler, defaults to 'gcc'
        -C compiler             Use the specified C++ compiler, defaults to 'g++
        -c compiler             Use the specified compiler, defaults to 'gcc'
                                on path.
        -g flags                Use flags to generate debug info. Defaults to
                                "-gdwarf-2".
        -m ctfmerge             Use the specified ctfmerge, defaults to
                                'ctfmerge'
                                'ctfmerge' on path.
        -t ctfconvert           Use the specified ctfconvert, defaults to
                                'ctfconvert'
                                'ctfconvert' on path.
USAGE
        exit 2
}
______unchanged_portion_omitted_

check_env()
{
        if which "$1" 2>/dev/null >/dev/null; then
                return
        fi

        [[ -f "$1" ]] || fatal "failed to find tool $1"
}

announce()
{
        cat << EOF
Beginning CTF tests with the following settings:
cc:             $(which $ctf_cc)
CC:             $(which $ctf_cxx)
as:             $(which $ctf_as)
ctfconvert:     $(which $ctf_convert)
ctfmerge:       $(which $ctf_merge)
COMPILER:       $(which $ctf_compiler)
CTFCONVERT:     $(which $ctf_convert)
CTFMERGE:       $(which $ctf_merge)
32-bit CFLAGS:  $ctf_32cflags
64-bit CFLAGS:  $ctf_64cflags

EOF
}

run_one()
{
        typeset source=$1 checker=$2 flags=$3

        if ! "$ctf_cc" $flags -o "$ctf_temp" -c "$source"; then
        if ! "$ctf_compiler" $flags -o "$ctf_temp" -c "$source"; then
                test_fail "failed to compile $source with flags: $flags"
                return
        fi

        if ! "$ctf_convert" "$ctf_temp"; then
                test_fail "failed to convert CTF in $source"
                return
        fi

        if ! "$checker" "$ctf_temp"; then
                test_fail "check for $source, $checker, failed"
                return
        fi

        echo "TEST PASSED: $source $flags"
}

#
# Perform a more complex build. The Makefile present will drive the
# building of the artifacts and the running of the tests based on the
# variables that we pass to it.
#
run_dir()
{
        typeset dir outdir check32 check64 flags32 flags64

        dir=$1
        outdir="$TMPDIR/ctftest.$$-$(basename $d)"
        check32=$2
        flags32=$3
        check64=$4
        flags64=$5

        if ! mkdir $outdir; then
                fatal "failed to make temporary directory '$outdir'"
        fi

        if ! make -C $dir -f Makefile.ctftest \
            BUILDDIR="$outdir" \
            CC="$ctf_cc" \
            CC="$ctf_compiler" \
            CFLAGS32="$ctf_mach32" \
            CFLAGS64="$ctf_mach64" \
            DEBUGFLAGS="$ctf_debugflags" \
            CTFCONVERT="$ctf_convert" \
            CTFMERGE="$ctf_merge" \
            build 1>/dev/null; then
                rm -rf $outdir
                test_fail "failed to build $dir"
                return
        fi

        if ! make -C $dir -f Makefile.ctftest \
            BUILDDIR="$outdir" \
            CHECK32="$check32" \
            CHECK64="$check64" \
            run-test 1>/dev/null; then
                rm -rf $outdir
                test_fail "failed to run tests for $dir"
                return
        fi

        rm -rf $outdir
        echo "TEST PASSED: $dir (dir)"
}

#
# Find all of the tests that exist and then try to run them all. Tests
# may either be a single file or a directory.
#
run_tests()
{
        typeset t base check
        ctf_tests=$(ls "$ctf_root"/*.c)
        for t in $ctf_tests; do
                base=$(basename "$t" .c)
                check=$(echo "$base" | sed s/test-/check-/)
```

```
                if [[ -f "$ctf_root/$check" ]]; then
                        run_one $t "$ctf_root/$check" "$ctf_32cflags"
                        run_one $t "$ctf_root/$check" "$ctf_64cflags"
                elif [[ -f "$ctf_root/$check-32" && \
                    -f "$ctf_root/$check-64" ]]; then
                        run_one $t "$ctf_root/$check-32" "$ctf_32cflags"
                        run_one $t "$ctf_root/$check-64" "$ctf_64cflags"
                else
                        test_fail "missing checker for $t"
                fi
        done

        for d in $(find "$ctf_root" -maxdepth 1 -type d -name 'test-*'); do
                [[ ! -f "$d/$ctf_makefile" ]] && continue
                base=$(basename "$d")
                check=$(echo "$base" | sed s/test-/check-/)
                if [[ -f "$ctf_root/$check" ]]; then
                        run_dir $d "$ctf_root/$check" "$ctf_32cflags" \
                            "$ctf_root/$check" "$ctf_64cflags"
                elif [[ -f "$ctf_root/$check-32" && \
                    -f "$ctf_root/$check-64" ]]; then
                        run_dir $d "$ctf_root/$check-32" "$ctf_32cflags" \
                            "$ctf_root/$check-64" "$ctf_64cflags"
                else
                        test_fail "missing checker for $t"
                fi
        done

        outdir="$TMPDIR/ctftest.$$"

        for f in $(find "$ctf_root" -maxdepth 1 -type f -name 'ctftest-*'); do
                if ! mkdir $outdir; then
                        fatal "failed to make temporary directory '$outdir'"
                fi

                echo "Running $f in $outdir"

                (cd $outdir && $f)

                if [[ $? -ne 0 ]]; then
                        test_fail "$f failed"
                else
                        echo "TEST PASSED: $f"
                fi

                rm -rf $outdir
        done
}

while getopts ":a:C:c:g:m:t:" c $@; do
while getopts ":c:g:m:t:" c $@; do
        case "$c" in
        a)
                ctf_as=$OPTARG
                ;;
        C)
                ctf_cxx=$OPTARG
                ;;
        c)
                ctf_cc=$OPTARG
                ctf_compiler=$OPTARG
                ;;
        g)
                ctf_debugflags=$OPTARG
                ;;
        m)
```

```
                ctf_merge=$OPTARG
                ;;
        t)
                ctf_convert=$OPTARG
                ;;
        :)
                usage "option requires an argument -- $OPTARG"
                ;;
        *)
                usage "invalid option -- $OPTARG"
                ;;
        esac
done

ctf_32cflags="$ctf_mach32 $ctf_debugflags"
ctf_64cflags="$ctf_mach64 $ctf_debugflags"

export ctf_as ctf_cc ctf_cxx ctf_debugflags ctf_merge ctf_convert

check_env "$ctf_compiler"
check_env "$ctf_convert"
check_env "$ctf_merge"
announce

run_tests

if [[ $ctf_nerrs -ne 0 ]]; then
        if [[ $ctf_nerrs -eq 1 ]]; then
                printf "\n%s: %u test failed\n" "$ctf_arg0" "$ctf_nerrs"
        else
                printf "\n%s: %u tests failed\n" "$ctf_arg0" "$ctf_nerrs"
        fi
        exit 1
else
        printf "\n%s: All tests passed successfully\n" "$ctf_arg0"
        exit 0
fi
```

```
**********************************************************
     737 Thu Apr 25 02:20:49 2019
new/usr/src/test/util-tests/tests/ctf/precheck.ksh
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

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

#
# Copyright (c) 2019, Joyent, Inc.
#

unalias -a

check_env()
{
        if which "$1" 2>/dev/null >/dev/null; then
                return
        fi

        [[ -f "$1" ]] || {
                echo "failed to find $1" >&2
                exit 1
        }
}

check_env as
check_env ctfconvert
check_env ctfmerge
check_env elfdump
check_env gcc
check_env g++
check_env ld
check_env make
```

```
**********************************************************
   74634 Thu Apr 25 02:20:49 2019
new/usr/src/uts/common/Makefile.rules
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2016 Garrett D'Amore <garrett@damore.org>
# Copyright 2013 Saso Kiselkov. All rights reserved.
# Copyright 2019 Joyent, Inc.
# Copyright 2016 Joyent, Inc.
# Copyright 2018 Nexenta Systems, Inc.
# Copyright (c) 2016 by Delphix. All rights reserved.
#

#
# uts/common/Makefile.rules
#
#       This Makefile defines all the file build rules for the directory
# uts/common and its children. These are the source files which may
# be considered common to all SunOS systems.
#
#       The following two-level ordering must be maintained in this file.
#         Lines are sorted first in order of decreasing specificity based on
#         the first directory component.  That is, sun4u rules come before
#         sparc rules come before common rules.
#
#         Lines whose initial directory components are equal are sorted
#         alphabetically by the remaining components.

#
#       Section 1a: C objects build rules
#
$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/aes/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/arcfour/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/blowfish/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/ecc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/modes/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/padding/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/rng/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/rsa/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/bignum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/bignum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/mpi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/acl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/avl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/ucode/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/brand/sn1/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/brand/solaris10/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/c2/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/contract/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/cpr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ctf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/ctf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/des/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/secflags/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/smbios/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/des/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/api/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/core/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/io/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/crypto/spi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/pci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/devid/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/disp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/dtrace/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(COMMONBASE)/exacct/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/aout/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/elf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/intp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/shbin/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/exec/java/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/$(FONT).c:  $(FONT_DIR)/$(FONT_SRC).bdf
        $(VTFONTCVT) -f source -o $@ $(FONT_DIR)/$(FONT_SRC).bdf

$(OBJS_DIR)/$(FONT).o:          $(OBJS_DIR)/$(FONT).c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/font/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/autofs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/bootfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/dcfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/devfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/ctfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/doorfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/dev/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/fd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/fifofs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/hsfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/lofs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/mntfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/namefs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/nfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/smbsrv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/vga/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbsrv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/objfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/pcfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/portfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/proc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/sharefs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/smbclnt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbclnt/netsmb/%.c
```

```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/sockfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/specfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/swapfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/tmpfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/udfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/ufs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vscan/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/zfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/zfs/lua/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/fs/zut/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/xattr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/zfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:        $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:        $(UTSBASE)/common/io/scsi/adapters/pmcs/%.bin
        $(COMPILE.b) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/fsreparse/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
KMECHKRB5_BASE=$(UTSBASE)/common/gssapi/mechs/krb5

KGSSDFLAGS=-I $(UTSBASE)/common/gssapi/include

# Note, KRB5_DEFS can be assigned various preprocessor flags,
# typically -D defines on the make invocation.  The standard compiler
# flags will not be overwritten.
KGSSDFLAGS += $(KRB5_DEFS)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/gssapi/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/gssapi/mechs/dummy/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/des/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/arcfour/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/dk/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/raw/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/crypto/old/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/krb/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/krb5/os/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/ser_sctx.o := CPPFLAGS += -DPROVIDE_KERNEL_IMPORT=1

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/mech/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(KMECHKRB5_BASE)/profile/%.c
        $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/idmap/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/arp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ip/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipnet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/iptun/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/kssl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sctp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/tcp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ilb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/ipd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/net/patricia/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/udp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/nca/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/sockmods/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/dlpistub/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/adapters/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/av1394/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/1394/targets/scsa1394/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sbp2/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/aac/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/afe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/atge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/arn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:               $(UTSBASE)/common/io/ath/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:               $(UTSBASE)/common/io/atu/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/impl/%.c
        $(COMPILE.c) -o $@ $<
```

```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/ac97/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioens/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio1575/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio810/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiohd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioixp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiols/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiopci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiop16x/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiots/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/blkdev/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnx/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnx/570x/driver/common/lmde
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/common/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/e
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bpf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cardbus/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/stmf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/fct/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/qlt/%.c
        $(COMPILE.c) -o $@ $<
```

```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/srpt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/fcoet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsit/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/iscsit/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/port/pppt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cpqary3/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dld/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dls/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/dmfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/drm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/efe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/elxl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fcoe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hme/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pciex/hotplug/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hotplug/pcihp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/rds/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/iser/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/ibd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/eoib/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/sdp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
        $(COMPILE.c) -o $@ $<
```

```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/ibnex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/ibtl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/tavor/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/hermon/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ib/clients/daplt/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/idm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ipw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwh/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iwp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/kb8042/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/kbtrans/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ksocket/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/aggr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/lp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mac/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mac/plugins/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mega_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mii/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mr_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mxfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mwl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/mwl/mwl_fw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/net80211/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nvme/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/npi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/nxge/%.s
        $(COMPILE.s) -o $@ $<
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pci-ide/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/pcn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/sppp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/spppasyn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ppp/sppptun/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/qede/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/qede/579xx/drivers/ecore/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ral/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rtls/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rsm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rtw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rum/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rwd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/rwn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/ahci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
```

```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/si3124/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sata/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/conf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/targets/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/ulp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/impl/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sfe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/simnet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/softmac/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/uath/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/uath/uath_fw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ural/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/urtw/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbskel/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hid/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/hidparser/%.c
```

```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/printer/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbms/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/ugen/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/clients/usbecm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/openhci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/ehci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/xhci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hcd/uhci/%.c
        $(COMPILE.c) -I../../common -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/hubd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/scsa2usb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usb_mid/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usb_ia/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usba/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usb/usba10/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/usbgem/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/axf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/udf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/udmf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/upf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/urf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vuidmice/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vnic/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/wpi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/zyd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/chxge/com/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/chxge/%.c
```

```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/common/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/shared/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/firmware/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/t4nex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ixgb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/xge/drv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/xge/hal/xgehal/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/e1000api/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/e1000g/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/igb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/iprb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ixgbe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ixgbe/core/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/i40e/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/i40e/core/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ntxn/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/myri10ge/drv/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/ipgpc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/dlcosmk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/flowacct/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/dscpmk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ipp/meters/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_emea/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_ja/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_ko/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_sc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kiconv/kiconv_tc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/klm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/kmdb/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/ktli/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
```

```
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/iscsi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/kifconf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/yge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sfxge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sfxge/common/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/skd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/virtio/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vioblk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/vioif/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
#
# krtld must refer to its own bzero/bcopy until the kernel is fully linked
#
$(OBJS_DIR)/bootrd.o            := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/doreloc.o           := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj.o              := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_boot.o         := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_bootflags.o    := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_convrelstr.o   := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_isa.o          := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_kdi.o          := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_lm.o           := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_reloc.o        := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_stubs.o        := CPPFLAGS += -DKOBJ_OVERRIDES
$(OBJS_DIR)/kobj_subr.o         := CPPFLAGS += -DKOBJ_OVERRIDES

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/krtld/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/list/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(COMMONBASE)/lzma/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/md4/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/md5/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/net/dhcp/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/nvpair/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/os/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/cis/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/cs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/nexus/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/pcmcia/pcs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/sec/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/rpc/sec_gss/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/edonr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/sha1/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/sha2/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/crypto/skein/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/syscall/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/tnf/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/tsol/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/util/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(COMMONBASE)/unicode/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/vm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/zmod/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/zlib_obj.o:         $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
        $(LD) -r -Breduce -M$(UTSBASE)/common/zmod/mapfile -o $@ \
            $(ZLIB_OBJS:%=$(OBJS_DIR)/%)
        $(CTFMERGE) -t -f -L VERSION -o $@ $(ZLIB_OBJS:%=$(OBJS_DIR)/%)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/hxge/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/tpm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/tpm/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/bz2%.o:             $(COMMONBASE)/bzip2/%.c
        $(COMPILE.c) -o $@ -I$(COMMONBASE)/bzip2 $<
        $(CTFCONVERT_O)

BZ2LINT = -erroff=%all -I$(UTSBASE)/common/bzip2

$(LINTS_DIR)/bz2%.ln:           $(COMMONBASE)/bzip2/%.c
         @($(LHEAD) $(LINT.c) -C $(LINTS_DIR)/‘basename $@ .ln‘ $(BZ2LINT) $< $(

#
#       Section 1b:     Lint ‘objects’
#
$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/aes/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/arcfour/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/blowfish/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/ecc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/modes/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/padding/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/rng/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/rsa/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/bignum/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/bignum/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/mpi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/acl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/avl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/ucode/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/brand/sn1/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/brand/solaris10/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/c2/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/conf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/contract/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/cpr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ctf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/ctf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/pci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/devid/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/des/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(COMMONBASE)/secflags/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/smbios/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/des/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/api/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/core/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/io/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/crypto/spi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/disp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/dtrace/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/exacct/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/exec/aout/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/exec/elf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/exec/intp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/exec/shbin/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/exec/java/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/autofs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/bootfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/ctfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/doorfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/dcfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/devfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/dev/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/fd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/fifofs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/hsfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/lofs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/mntfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/namefs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/smbsrv/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/smbsrv/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/nfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/objfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/pcfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/portfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/proc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/sharefs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/smbclnt/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/smbclnt/netsmb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/sockfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/specfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/swapfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/tmpfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/udfs/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/ufs/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/ufs_log/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vscan/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/zfs/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/zfs/lua/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/fs/zut/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/xattr/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/zfs/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/gssapi/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/gssapi/mechs/dummy/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/des/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/dk/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/os/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/arcfour/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/raw/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/crypto/old/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/krb5/krb/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/krb5/os/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(KMECHKRB5_BASE)/mech/%.c
	@($(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/idmap/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/sockmods/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/arp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ip/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ipnet/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/iptun/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ipd/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ipf/%.c
	@($(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/kssl/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/net/patricia/%.c
	@($(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/udp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/sctp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/tcp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/ilb/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/nca/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/dlpistub/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/font/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/adapters/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/av1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/1394/targets/scsa1394/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sbp2/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/aac/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/afe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/atge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/arn/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ath/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/atu/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/impl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/ac97/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audio1575/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audio810/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioens/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiohd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audioixp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiols/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiopci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiop16x/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiots/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bfe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bpf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/blkdev/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/common/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/e
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/bnxe/577xx/drivers/common/l
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cardbus/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/fct/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/qlt/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/srpt/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsit/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/fcoet/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/iscsit/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/port/pppt/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/comstar/stmf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cpqary3/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dld/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dls/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/dmfe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/drm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/efe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/elxl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fcoe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hme/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/pciex/hotplug/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hotplug/pcihp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/rds/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/iser/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/ibd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/eoib/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/sdp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/ibnex/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/ibtl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/adapters/tavor/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/adapters/hermon/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ib/clients/daplt/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/idm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ipw/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwh/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iwk/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/iwp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/kb8042/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/kbtrans/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/ksocket/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/aggr/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/lp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mac/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mac/plugins/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mega_sas/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mii/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mr_sas/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mxfe/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mwl/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/mwl/mwl_fw/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/net80211/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/nge/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/nvme/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/nxge/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/nxge/%.s
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/nxge/npi/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/pci-ide/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/pcn/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/ppp/sppp/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/ppp/spppasyn/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/ppp/sppptun/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/qede/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/ral/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rge/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rtls/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rsm/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rtw/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rum/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rwd/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/rwn/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/sata/adapters/ahci/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/sata/adapters/si3124/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/sata/impl/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
	@($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:		$(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
	@($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/ulp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/impl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/conf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/impl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/targets/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/impl/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sfe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/simnet/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/softmac/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/uath/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/uath/uath_fw/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ural/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/urtw/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbskel/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hid/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/hidparser/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbms/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/ugen/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/printer/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/clients/usbecm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/openhci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/ehci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/xhci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hcd/uhci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/hubd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/scsa2usb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usb_mid/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usb_ia/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usba/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usb/usba10/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/usbgem/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/axf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/udmf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/upf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/urf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vuidmice/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vnic/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/wpi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/zyd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/chxge/com/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/chxge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/common/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/shared/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/firmware/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/t4nex/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/cxgbe/cxgbe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ixgb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/xge/drv/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/xge/hal/xgehal/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/e1000g/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/e1000api/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/igb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/iprb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ixgbe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ixgbe/core/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/i40e/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/ntxn/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/myri10ge/drv/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/ipgpc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/dlcosmk/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/flowacct/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/dscpmk/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ipp/meters/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_emea/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_ja/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_ko/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_sc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kiconv/kiconv_tc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/klm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/kmdb/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/krtld/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/ktli/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/list/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/lzma/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/md4/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/md5/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/net/dhcp/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/nvpair/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/os/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/cs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/cis/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/nexus/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/pcmcia/pcs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/sec/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/rpc/sec_gss/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/edonr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/sha1/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/sha2/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/crypto/skein/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/syscall/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/tnf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/tsol/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/util/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/unicode/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/vm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/iscsi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/inet/kifconf/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/virtio/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vioblk/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vioif/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

ZMODLINTFLAGS = -erroff=E_CONSTANT_CONDITION

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/zmod/%.c
        @($(LHEAD) $(LINT.c) $(ZMODLINTFLAGS) $< $(LTAIL))

$(LINTS_DIR)/zlib_obj.ln:       $(ZLIB_OBJS:%.o=$(LINTS_DIR)/%.ln) \
                                $(UTSBASE)/common/zmod/zlib_lint.c
        @($(LHEAD) $(LINT.c) -C $(LINTS_DIR)/zlib_obj \
            $(UTSBASE)/common/zmod/zlib_lint.c $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/hxge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/tpm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/tpm/%.s
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/vr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/yge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sfxge/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/sfxge/common/%.c
```

```
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/io/skd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(COMMONBASE)/fsreparse/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
************************************************************
   14030 Thu Apr 25 02:20:49 2019
new/usr/src/uts/common/sys/ctf_api.h
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
************************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2005 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */
/*
 * Copyright 2019 Joyent, Inc.
 * Copyright 2018 Joyent, Inc.
 */

/*
 * This header file defines the interfaces available from the CTF debugger
 * library, libctf, and an equivalent kernel module.  This API can be used by
 * a debugger to operate on data in the Compact ANSI-C Type Format (CTF).
 * This is NOT a public interface, although it may eventually become one in
 * the fullness of time after we gain more experience with the interfaces.
 *
 * In the meantime, be aware that any program linked with this API in this
 * release of Solaris is almost guaranteed to break in the next release.
 *
 * In short, do not user this header file or the CTF routines for any purpose.
 */

#ifndef _CTF_API_H
#define _CTF_API_H

#include <sys/types.h>
#include <sys/param.h>
#include <sys/elf.h>
#include <sys/ctf.h>

#ifdef  __cplusplus
extern "C" {
```

```
#endif

/*
 * Clients can open one or more CTF containers and obtain a pointer to an
 * opaque ctf_file_t.  Types are identified by an opaque ctf_id_t token.
 * These opaque definitions allow libctf to evolve without breaking clients.
 */
typedef struct ctf_file ctf_file_t;
typedef long ctf_id_t;

#define ECTF_BASE       1000    /* base value for libctf errnos */

enum {
        ECTF_FMT = ECTF_BASE,   /* file is not in CTF or ELF format */
        ECTF_ELFVERS,           /* ELF version is more recent than libctf */
        ECTF_CTFVERS,           /* CTF version is more recent than libctf */
        ECTF_ENDIAN,            /* data is different endian-ness than lib */
        ECTF_SYMTAB,            /* symbol table uses invalid entry size */
        ECTF_SYMBAD,            /* symbol table data buffer invalid */
        ECTF_STRBAD,            /* string table data buffer invalid */
        ECTF_CORRUPT,           /* file data corruption detected */
        ECTF_NOCTFDATA,         /* ELF file does not contain CTF data */
        ECTF_NOCTFBUF,          /* buffer does not contain CTF data */
        ECTF_NOSYMTAB,          /* symbol table data is not available */
        ECTF_NOPARENT,          /* parent CTF container is not available */
        ECTF_DMODEL,            /* data model mismatch */
        ECTF_MMAP,              /* failed to mmap a data section */
        ECTF_ZMISSING,          /* decompression library not installed */
        ECTF_ZINIT,             /* failed to initialize decompression library */
        ECTF_ZALLOC,            /* failed to allocate decompression buffer */
        ECTF_DECOMPRESS,        /* failed to decompress CTF data */
        ECTF_STRTAB,            /* string table for this string is missing */
        ECTF_BADNAME,           /* string offset is corrupt w.r.t. strtab */
        ECTF_BADID,             /* invalid type ID number */
        ECTF_NOTSOU,            /* type is not a struct or union */
        ECTF_NOTENUM,           /* type is not an enum */
        ECTF_NOTSUE,            /* type is not a struct, union, or enum */
        ECTF_NOTINTFP,          /* type is not an integer or float */
        ECTF_NOTARRAY,          /* type is not an array */
        ECTF_NOTREF,            /* type does not reference another type */
        ECTF_NAMELEN,           /* buffer is too small to hold type name */
        ECTF_NOTYPE,            /* no type found corresponding to name */
        ECTF_SYNTAX,            /* syntax error in type name */
        ECTF_NOTFUNC,           /* symtab entry does not refer to a function */
        ECTF_NOFUNCDAT,         /* no func info available for function */
        ECTF_NOTDATA,           /* symtab entry does not refer to a data obj */
        ECTF_NOTYPEDAT,         /* no type info available for object */
        ECTF_NOLABEL,           /* no label found corresponding to name */
        ECTF_NOLABELDATA,       /* file does not contain any labels */
        ECTF_NOTSUP,            /* feature not supported */
        ECTF_NOENUMNAM,         /* enum element name not found */
        ECTF_NOMEMBNAM,         /* member name not found */
        ECTF_RDONLY,            /* CTF container is read-only */
        ECTF_DTFULL,            /* CTF type is full (no more members allowed) */
        ECTF_FULL,              /* CTF container is full */
        ECTF_DUPMEMBER,         /* duplicate member name definition */
        ECTF_CONFLICT,          /* conflicting type definition present */
        ECTF_REFERENCED,        /* type has outstanding references */
        ECTF_NOTDYN,            /* type is not a dynamic type */
        ECTF_ELF,               /* elf library failure */
        ECTF_MCHILD,            /* cannot merge child container */
        ECTF_LABELEXISTS,       /* label already exists */
        ECTF_LCONFLICT,         /* merged labels conflict */
        ECTF_ZLIB,              /* zlib library failure */
        ECTF_CONVBKERR,         /* CTF conversion backend error */
        ECTF_CONVNOCSRC,        /* No C source to convert from */
        ECTF_CONVNODEBUG        /* No debug info to convert into CTF */
        ECTF_NOCONVBKEND        /* No applicable conversion backend */
};
```

______*unchanged_portion_omitted_*

```
**********************************************************
   13511 Thu Apr 25 02:20:49 2019
new/usr/src/uts/intel/Makefile.rules
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2019 Joyent, Inc.  All rights reserved.
# Copyright 2018 Joyent, Inc.  All rights reserved.
# Copyright 2017 Nexenta Systems, Inc.
#

#       This Makefile defines all file modules and build rules for the
# directory uts/intel and its children. These are the source files which
# are specific to the intel processor.
#
#       The following two-level ordering must be maintained in this file.
#         Lines are sorted first in order of decreasing specificity based on
#         the first directory component.  That is, sun4u rules come before
#         sparc rules come before common rules.
#
#         Lines whose initial directory components are equal are sorted
#         alphabetically by the remaining components.

#
# Need a way to distinguish between the ia32 and amd64 subdirs.
#
SUBARCH_DIR_32 = ia32
SUBARCH_DIR_64 = amd64
SUBARCH_DIR     = $(SUBARCH_DIR_$(CLASS))

#
#       Section 1a: C object build rules
#
$(OBJS_DIR)/%.o:                $(SRC)/common/fs/%.c
        $(COMPILE.c) -o $@ $<
```

```
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/power/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/util/i386/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/brand/sn1/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/brand/solaris10/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/dtrace/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/dtrace/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/fs/proc/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/ia32/ml/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/ia32/os/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/ia32/promif/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/ia32/syscall/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/acpica/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/acpica/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/events/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/hardware/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/dispatcher/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/executer/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/parser/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/namespace/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/resources/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/tables/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/utilities/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/acpica/disassembler/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/agpgart/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/agpmaster/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/amd8111s/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/amr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/drm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/hotplug/pcicfg/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/ipmi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/intel_nb5000/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/intel_nhm/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(SRC)/common/mc/mc-amd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/mc-amd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/pci/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/pciex/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dktp/controller/ata/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dktp/dcdev/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dktp/disk/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dktp/drvobj/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dktp/hba/ghd/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/dnet/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/scsi/adapters/arcmsr/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/scsi/targets/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/vgatext/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/vmxnet3s/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/io/scsi/adapters/pvscsi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/nskern/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/os/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/pcbe/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)
```

```
$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/promif/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/syscall/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/common/os/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/kdi/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/kdi/%.s
        $(COMPILE.s) -o $@ $<

$(OBJS_DIR)/%.o:                $(UTSBASE)/intel/zfs/%.c
        $(COMPILE.c) -o $@ $<
        $(CTFCONVERT_O)

#
# krtld compiled into unix
#

KRTLD_INC_PATH  = -I$(UTSBASE)/common/krtld -I$(UTSBASE)/intel/sys
KRTLD_INC_PATH  += -I$(UTSBASE)/intel/$(SUBARCH_DIR)/krtld

KRTLD_CPPFLAGS_32       = -DELF_TARGET_386
KRTLD_CPPFLAGS_64       = -DELF_TARGET_AMD64 -DMODDIR_SUFFIX=\"amd64\"
KRTLD_CPPFLAGS          = $(KRTLD_CPPFLAGS_$(CLASS)) -D_KRTLD

$(OBJS_DIR)/%.o:        $(UTSBASE)/common/krtld/%.c
        $(COMPILE.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) -o $@ $<
        $(CTFCONVERT_O)

$(OBJS_DIR)/%.o:        $(UTSBASE)/intel/$(SUBARCH_DIR)/krtld/%.c
        $(COMPILE.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) -o $@ $<
        $(CTFCONVERT_O)

#
# _DBOOT indicates that krtld is called from a dboot ELF section
#
$(OBJS_DIR)/kobj.o              := CPPFLAGS += -D_DBOOT

$(OBJS_DIR)/%.o:        $(UTSBASE)/intel/$(SUBARCH_DIR)/krtld/%.s
        $(COMPILE.s) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) -o $@ $<
        $(CTFCONVERT_O)
$(OBJS_DIR)/%.o:        $(SRC)/common/util/$(SUBARCH_DIR)/%.c
        $(COMPILE.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) -o $@ $<
        $(CTFCONVERT_O)


#
#       Section 1b: Lint ‘object’ build rules.
#
$(LINTS_DIR)/%.ln:              $(SRC)/common/fs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/util/i386/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/brand/sn1/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/brand/solaris10/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/dtrace/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/dtrace/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/zfs/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/fs/proc/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/ia32/ml/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/ia32/os/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/ia32/promif/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/ia32/syscall/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/acpica/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/acpica/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/events/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/hardware/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/dispatcher/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/executer/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/parser/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/namespace/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/resources/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/tables/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/utilities/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(SRC)/common/acpica/disassembler/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/agpgart/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/agpmaster/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/amd8111s/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/amr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/scsi/adapters/arcmsr/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/drm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/hotplug/pcicfg/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/hotplug/pci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/intel_nb5000/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/intel_nhm/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/ipmi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(SRC)/common/mc/mc-amd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/mc-amd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/pci/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/pciex/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dktp/controller/ata/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dktp/dcdev/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dktp/disk/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dktp/drvobj/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dktp/hba/ghd/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/dnet/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))
```

```
$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/scsi/targets/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/vgatext/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/vmxnet3s/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/io/scsi/adapters/pvscsi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/os/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/pcbe/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/promif/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/syscall/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/os/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/kdi/%.c
        @($(LHEAD) $(LINT.c) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/kdi/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

$(LINTS_DIR)/%.ln:              $(UTSBASE)/intel/nskern/%.s
        @($(LHEAD) $(LINT.s) $< $(LTAIL))

#
# krtld lints
#
$(LINTS_DIR)/%.ln:              $(UTSBASE)/common/krtld/%.c
        @($(LHEAD) $(LINT.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:      $(UTSBASE)/intel/$(SUBARCH_DIR)/krtld/%.c
        @($(LHEAD) $(LINT.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:      $(UTSBASE)/intel/$(SUBARCH_DIR)/krtld/%.s
        @($(LHEAD) $(LINT.s) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) $< $(LTAIL))

$(LINTS_DIR)/%.ln:      $(SRC)/common/util/$(SUBARCH_DIR)/%.c
        @($(LHEAD) $(LINT.c) $(KRTLD_INC_PATH) $(KRTLD_CPPFLAGS) $< $(LTAIL))

$(OBJS_DIR)/kobj.ln             := CPPFLAGS += -D_DBOOT
```

```
**********************************************************
    2492 Thu Apr 25 02:20:50 2019
new/usr/src/uts/intel/aes/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2014 Garrett D’Amore <garrett@damore.org>
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
#       This makefile drives the production of the AES KEF provider.
#
#       Intel implementation architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COM_DIR = $(COMMONBASE)/crypto/aes

#
#       Define the module and object file sets.
#
MODULE          = aes
LINTS           = $(AESPROV_OBJS:%.o=$(LINTS_DIR)/%.ln)
AESPROV_OBJS_32 =
AESPROV_OBJS_64 = aes_amd64.o aes_intel.o aeskey.o
AESPROV_OBJS    += $(AESPROV_OBJS_$(CLASS))
OBJECTS         = $(AESPROV_OBJS:%=$(OBJS_DIR)/%)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel
```

```
#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

#
#       Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf

CPPFLAGS        += -I$(COM_DIR) -I$(COM_DIR)/..
CPPFLAGS        += -DCRYPTO_PROVIDER_NAME=\"$(MODULE)\"
AS_CPPFLAGS     += -I../../$(PLATFORM)

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

$(OBJS_DIR)/%.ln: $(COM_DIR)/amd64/%.c
        @($(LHEAD) $(LINT.c) $(COM_DIR)/amd64/${@F:.ln=.c} $(LTAIL))

$(OBJS_DIR)/%.o:  $(COM_DIR)/amd64/%.c
        $(COMPILE.c) -o $@ $(COM_DIR)/amd64/${@F:.o=.c}
        $(CTFCONVERT_O)
        $(POST_PROCESS_O)

$(OBJS_DIR)/%.o: $(COM_DIR)/amd64/%.s
        $(COMPILE.s) -o $@ $(COM_DIR)/amd64/${@F:.o=.s}
        $(POST_PROCESS_O)

$(OBJS_DIR)/%.ln: $(COM_DIR)/amd64/%.s
        @($(LHEAD) $(LINT.s) $(COM_DIR)/amd64/${@F:.ln=.s} $(LTAIL))
```

```
**********************************************************
    2274 Thu Apr 25 02:20:50 2019
new/usr/src/uts/intel/arcfour/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
#       This makefile drives the production of the ARCFOUR KEF provider.
#
#       Intel implementation architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COM_DIR = $(COMMONBASE)/crypto/arcfour

#
#       Define the module and object file sets.
#
MODULE          = arcfour
LINTS           = $(ARCFOURPROV_OBJS:%.o=$(LINTS_DIR)/%.ln)
ARCFOURPROV_OBJS_32 =
ARCFOURPROV_OBJS_64 = arcfour-x86_64.o
ARCFOURPROV_OBJS += $(ARCFOURPROV_OBJS_$(CLASS))
OBJECTS         = $(ARCFOURPROV_OBJS:%=$(OBJS_DIR)/%)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel
```

```
#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

#
# Overrides
#
CPPFLAGS        += -I$(COM_DIR)
CFLAGS          += -xO4 -xcrossfile
CLEANFILES      += arcfour-x86_64.s

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

$(OBJS_DIR)/arcfour-x86_64.o: arcfour-x86_64.s
        $(COMPILE.s) -o $@ ${@F:.o=.s}
        $(POST_PROCESS_O)

$(OBJS_DIR)/arcfour-x86_64.ln: arcfour-x86_64.s
        @($(LHEAD) $(LINT.s) ${@F:.ln=.s} $(LTAIL))

arcfour-x86_64.s: $(COM_DIR)/amd64/arcfour-x86_64.pl
        $(PERL) $? $@
```

```
**********************************************************
    3440 Thu Apr 25 02:20:50 2019
new/usr/src/uts/intel/bignum/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
#
# Copyright (c) 2019, Joyent, Inc.
#       This makefile drives the production of the RSA KEF provider.
#
#       intel implementation architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
BIGNUMDIR = $(COMMONBASE)/bignum
CRYPTODIR = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = bignum
OBJECTS         = $(BIGNUM_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(BIGNUM_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_MISC_DIR)/$(MODULE)

BIGNUM_PSR_OBJS = bignum_amd64.o bignum_amd64_asm.o
BIGNUM_OBJS_32 = bignum_i386.o bignum_i386_asm.o
BIGNUM_OBJS_64 = bignum_amd64.o bignum_amd64_asm.o
BIGNUM_PSR_OBJS = $(BIGNUM_OBJS_$(CLASS))

#
#       Include common rules.
```

```
#
include $(UTSBASE)/intel/Makefile.intel

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

#
# Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf

CPPFLAGS        += -I$(BIGNUMDIR) -I$(CRYPTODIR)

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

# Bignum configuration (BIGNUM_CFG):
#   PSR_MUL:
#       There is a processor-specific implementation bignum multiply functions
#   HWCAP:
#       There are multiple implementations of bignum functions, and the
#       appropriate one must be chosen at run time, based on testing
#       hardware capabilites.
#
# -DPSR_MUL:
# For AMD64, there is a processor-specific implementation of
# the bignum multiply functions, which takes advantage of the
# 64x64->128 bit multiply instruction.
#
# -UHWCAP:
# There is only one implementation, because the 128 bit multiply using
# general-purpose registers is faster than any MMX or SSE2 implementation.

BIGNUM_CFG = -DPSR_MUL
CFLAGS  += -xO4 -xcrossfile
$(OBJS_DIR)/bignumimpl.o $(LINTS_DIR)/bignumimpl.ln := \
        CPPFLAGS += $(BIGNUM_CFG)
$(OBJS_DIR)/bignum_amd64.o $(LINTS_DIR)/bignum_amd64.ln := \
        CPPFLAGS += $(BIGNUM_CFG)
```

```
$(OBJS_DIR)/bignum_amd64.o: $(BIGNUMDIR)/amd64/bignum_amd64.c
        $(COMPILE.c) -o $@ $(BIGNUM_CFG) $(BIGNUMDIR)/amd64/bignum_amd64.c
        $(CTFCONVERT_O)

$(OBJS_DIR)/bignum_amd64_asm.o: $(BIGNUMDIR)/amd64/bignum_amd64_asm.s
        $(COMPILE.s) -P -o $@ $(BIGNUM_CFG) \
                $(BIGNUMDIR)/amd64/bignum_amd64_asm.s

$(LINTS_DIR)/bignum_amd64.ln: $(BIGNUMDIR)/amd64/bignum_amd64.c
        @($(LHEAD) $(LINT.c) $(BIGNUMDIR)/amd64/bignum_amd64.c $(LTAIL))

$(LINTS_DIR)/bignum_amd64_asm.ln: $(BIGNUMDIR)/amd64/bignum_amd64_asm.s
        @($(LHEAD) $(LINT.s) $(BIGNUMDIR)/amd64/bignum_amd64_asm.s $(LTAIL))
include Makefile.$(CLASS)
```

```
**********************************************************
    3405 Thu Apr 25 02:20:51 2019
new/usr/src/uts/intel/elfexec/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# uts/intel/elfexec/Makefile
#
# Copyright 2006 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# Copyright 2019, Joyent, Inc.
# Copyright 2016 Joyent, Inc.
#

#
#       This makefile drives the production of the elfexec exec kernel
#       module.
#
#       intel architecture dependent
#

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

#
#       Define the module and object file sets.
#
MODULE          = elfexec
ELFEXEC_OBJS_32 =
ELFEXEC_OBJS_64 = elf32.o elf32_notes.o old32_notes.o
ELFEXEC_OBJS    += $(ELFEXEC_OBJS_$(CLASS))
OBJECTS         = $(ELFEXEC_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(ELFEXEC_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_EXEC_DIR)/$(MODULE)

#
```

```
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

INC_PATH        += -I$(UTSBASE)/intel/ia32

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

CERRWARN        += -_gcc=-Wno-parentheses
CERRWARN        += -_gcc=-Wno-uninitialized

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

.NO_PARALLEL:   $(LINTS)

$(OBJS_DIR)/elf32.o:    $(UTSBASE)/common/exec/elf/elf.c
        $(COMPILE.c) -o $@ -D_ELF32_COMPAT $(UTSBASE)/common/exec/elf/elf.c
        $(CTFCONVERT_O)

$(OBJS_DIR)/elf32_notes.o: $(UTSBASE)/common/exec/elf/elf_notes.c
        $(COMPILE.c) -o $@ -D_ELF32_COMPAT $(UTSBASE)/common/exec/elf/elf_notes.
        $(CTFCONVERT_O)

$(OBJS_DIR)/old32_notes.o: $(UTSBASE)/common/exec/elf/old_notes.c
        $(COMPILE.c) -o $@ -D_ELF32_COMPAT $(UTSBASE)/common/exec/elf/old_notes.
        $(CTFCONVERT_O)

$(LINTS_DIR)/elf32.ln:  $(UTSBASE)/common/exec/elf/elf.c
```

```
	@($(LHEAD) $(LINT.c) -Celf32 -D_ELF32_COMPAT $(UTSBASE)/common/exec/elf/
	@$(MV) $(@F) $@

$(LINTS_DIR)/elf32_notes.ln: $(UTSBASE)/common/exec/elf/elf_notes.c
	@($(LHEAD) $(LINT.c) -Celf32_notes -D_ELF32_COMPAT $(UTSBASE)/common/exe
	@$(MV) $(@F) $@

$(LINTS_DIR)/old32_notes.ln: $(UTSBASE)/common/exec/elf/old_notes.c
	@($(LHEAD) $(LINT.c) -Cold32_notes -D_ELF32_COMPAT $(UTSBASE)/common/exe
	@$(MV) $(@F) $@
```

```
**********************************************************
    2389 Thu Apr 25 02:20:51 2019
new/usr/src/uts/intel/kcf/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn’t build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
# Copyright (c) 2018, Joyent, Inc.

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COM_DIR = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = kcf
LINTS           = $(KCF_OBJS:%.o=$(LINTS_DIR)/%.ln)
KCF_OBJS_32     =
KCF_OBJS_64     = gcm_intel.o
KCF_OBJS        += $(KCF_OBJS_$(CLASS))
OBJECTS         = $(KCF_OBJS:%=$(OBJS_DIR)/%)
ROOTMODULE      = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
```

```
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE)

#
# Linkage dependencies
#

#
# lint pass one enforcement
#
CFLAGS          += $(CCVERBOSE) -I$(COM_DIR)
AS_CPPFLAGS     += -I../../$(PLATFORM)

LINTTAGS        += -I$(COM_DIR)

CERRWARN        += -_gcc=-Wno-switch
CERRWARN        += -_gcc=-Wno-uninitialized
CERRWARN        += -_gcc=-Wno-parentheses
CERRWARN        += -_gcc=-Wno-unused-label

# needs work
SMOFF += all_func_returns,signed_integer_overflow_check

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

$(OBJS_DIR)/%.o: $(COM_DIR)/modes/amd64/%.s
        $(COMPILE.s) -o $@ $(COM_DIR)/modes/amd64/${@F:.o=.s}
        $(POST_PROCESS_O)

$(OBJS_DIR)/%.ln: $(COM_DIR)/modes/amd64/%.s
        @($(LHEAD) $(LINT.s) $(COM_DIR)/modes/amd64/${@F:.ln=.s} $(LTAIL))
```

```
**********************************************************
    2297 Thu Apr 25 02:20:51 2019
new/usr/src/uts/intel/md5/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# uts/intel/md5/Makefile
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
#       This makefile drives the production of the md5 crypto kernel module.
#
#       intel architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COMDIR = $(COMMONBASE)/crypto/md5

#
#       Define the module and object file sets.
#
MODULE          = md5
MD5_OBJS_32     =
MD5_OBJS_64     = md5_amd64.o
MD5_OBJS        += $(MD5_OBJS_$(CLASS))
OBJECTS         = $(MD5_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(MD5_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
#
```

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

#
#       Override defaults
#
CLEANFILES += md5_amd64.s

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

#
# md5 depends on the kcf framework
#
LDFLAGS         += -dy -Nmisc/kcf

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

md5_amd64.s: $(COMDIR)/amd64/md5_amd64.pl
        $(PERL) $? $@

$(OBJS_DIR)/md5_amd64.o: md5_amd64.s
        $(COMPILE.s) -o $@ md5_amd64.s
        $(POST_PROCESS_O)

$(OBJS_DIR)/md5_amd64.ln: md5_amd64.s
        @($(LHEAD) $(LINT.c) md5_amd64.s $(LTAIL))
```

```
************************************************************
    2329 Thu Apr 25 02:20:51 2019
new/usr/src/uts/intel/sha1/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
************************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# uts/intel/sha1/Makefile
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
#
#       This makefile drives the production of the sha1 crypto kernel module.
#
#       intel architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COMDIR = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = sha1
SHA1_OBJS_32    =
SHA1_OBJS_64    = sha1-x86_64.o
SHA1_OBJS       += $(SHA1_OBJS_$(CLASS))
OBJECTS         = $(SHA1_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(SHA1_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
```

```
#
include $(UTSBASE)/intel/Makefile.intel

#
#       Override defaults
#
CLEANFILES += sha1-x86_64.s

#
# Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf

CFLAGS += -I$(COMDIR)
LINTFLAGS       += -I$(COMDIR)


#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

$(OBJS_DIR)/sha1-x86_64.o: sha1-x86_64.s
        $(COMPILE.s) -o $@ ${@F:.o=.s}
        $(POST_PROCESS_O)

$(OBJS_DIR)/sha1-x86_64.ln: sha1-x86_64.s
        @($(LHEAD) $(LINT.c) ${@F:.ln=.s} $(LTAIL))

sha1-x86_64.s: $(COMDIR)/sha1/amd64/sha1-x86_64.pl
        $(PERL) $? $@
```

```
**********************************************************
    2479 Thu Apr 25 02:20:52 2019
new/usr/src/uts/intel/sha2/Makefile
10816 ctf_dwarf_convert_type() relies on un-initialized id
10817 ctfconvert -i option is mis-handled
10818 Improve ctfconvert error messages
10819 ctfconvert should handle empty dies
10820 ctfconvert -i never converts
10821 bad free in ctf_dwarf_init_die
10815 shouldn't build gcore.c as part of kmdb
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Jerry Jelinek <jerry.jelinek@joyent.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2019, Joyent, Inc.
#
#       This makefile drives the production of the sha2 crypto kernel module.
#
#       intel architecture dependent
#

#
#       Path to the base of the uts directory tree (usually /usr/src/uts).
#
UTSBASE = ../..
COMDIR  = $(COMMONBASE)/crypto

#
#       Define the module and object file sets.
#
MODULE          = sha2
SHA2_OBJS_32    =
SHA2_OBJS_64    = sha512-x86_64.o sha256-x86_64.o
SHA2_OBJS       += $(SHA2_OBJS_$(CLASS))
OBJECTS         = $(SHA2_OBJS:%=$(OBJS_DIR)/%)
LINTS           = $(SHA2_OBJS:%.o=$(LINTS_DIR)/%.ln)
ROOTMODULE      = $(ROOT_CRYPTO_DIR)/$(MODULE)
ROOTLINK        = $(ROOT_MISC_DIR)/$(MODULE)

#
#       Include common rules.
#
include $(UTSBASE)/intel/Makefile.intel
```

```
#
#       Override defaults
#
CLEANFILES += sha512-x86_64.s sha256-x86_64.s

#
#       Define targets
#
ALL_TARGET      = $(BINARY)
LINT_TARGET     = $(MODULE).lint
INSTALL_TARGET  = $(BINARY) $(ROOTMODULE) $(ROOTLINK)

#
# Linkage dependencies
#
LDFLAGS += -dy -Nmisc/kcf

CFLAGS += -I$(COMDIR)
LINTFLAGS       += -I$(COMDIR)

CERRWARN        += -_gcc=-Wno-switch
CERRWARN        += -_gcc=-Wno-uninitialized

#
#       Default build targets.
#
.KEEP_STATE:

def:            $(DEF_DEPS)

all:            $(ALL_DEPS)

clean:          $(CLEAN_DEPS)

clobber:        $(CLOBBER_DEPS)

lint:           $(LINT_DEPS)

modlintlib:     $(MODLINTLIB_DEPS)

clean.lint:     $(CLEAN_LINT_DEPS)

install:        $(INSTALL_DEPS)

$(ROOTLINK):    $(ROOT_MISC_DIR) $(ROOTMODULE)
        -$(RM) $@; ln $(ROOTMODULE) $@

#
#       Include common targets.
#
include $(UTSBASE)/intel/Makefile.targ

$(OBJS_DIR)/%.o: %.s
        $(COMPILE.s) -o $@ ${@F:.o=.s}
        $(POST_PROCESS_O)

$(OBJS_DIR)/%.ln: %.s
        @($(LHEAD) $(LINT.c) ${@F:.ln=.s} $(LTAIL))

sha512-x86_64.s: $(COMDIR)/sha2/amd64/sha512-x86_64.pl
        $(PERL) $? $@

sha256-x86_64.s: $(COMDIR)/sha2/amd64/sha512-x86_64.pl
        $(PERL) $? $@
```