**USAGE**

Compilación a secas, sin funciones adicionales:

% make

Para compilar con el compilador de Intel:

% make -f Makefile.init Makefiles CC="/opt/icc/bin/icc"

% make

Para compilar con Tiny CC (tcc):

make -f Makefile.init Makefiles CC="/usr/bin/tcc" AUXLIBS="-lm"

make

Para compilar con assertions:

% make -f Makefile.init Makefiles CCARGS="-DDEBUG"

% make

Para compilar en un entorno x86 de 64 bits:

% make -f Makefile.init Makefiles CCARGS="-m32"

% make

Para apagar las optimizaciones:

% make -f Makefile.init Makefiles OPT="-O0"

% make

Para hacer profiling con gprof:

$ make -f Makefile.init Makefiles CCARGS="-pg" AUXLIBS="-lc\_p"

$ make

**makedefs**

#! /bin/sh

# Ugly function to make our error message more visible among the

# garbage that is output by some versions of make(1).

#

# By now all shells must have functions.

#

error() {

# Alas, tput(1) is not portable so we can't use visual effects.

echo "ATTENTION:" 1>&2;

echo "ATTENTION:" $\* 1>&2;

echo "ATTENTION:" 1>&2;

exit 1

}

SYSTEM=`(uname -s) 2>/dev/null`

RELEASE=`(uname -r) 2>/dev/null`

MACHINE=`(uname -m) 2>/dev/null`

VERSION=`(uname -v) 2>/dev/null`

SYSLIBS=

case "$SYSTEM.$RELEASE" in

NetBSD.3\*)

SYSTYPE=NETBSD3

;;

Linux.2\*)

SYSTYPE=LINUX2

;;

Linux.3\*)

SYSTYPE=LINUX3

;;

SunOS.5\*)

SYSTYPE=SUNOS5

;;

\*)

error "Unknown system type: $SYSTEM $RELEASE"

;;

esac

: ${CC='gcc $(WARN)'} ${OPT='-O'} ${DEBUG='-g'} ${WARN='-Wall'}

case "$CCARGS" in

\*-DUSE\_SMOOTH\_SHADING\*)

SYSLIBS="$SYSLIBS -lm"

;;

esac

export SYSTYPE SYSLIBS CC OPT DEBUG OPTS

sed 's/ / /g' <<EOF

SYSTYPE = $SYSTYPE

SYSLIBS = $AUXLIBS $SYSLIBS

CC = $CC $CCARGS

OPT = $OPT

DEBUG = $DEBUG

EXPORT = AUXLIBS='$AUXLIBS' CCARGS='$CCARGS' OPT='$OPT' DEBUG='$DEBUG'

WARN = $WARN

EOF

**Makefile**

# Usage:

# make makefiles [CC=compiler] [OPT=compiler-flags] [DEBUG=debug-flags]

#

# The defaults are: CC=gcc, OPT=-O, and DEBUG=-g. Examples:

#

# make makefiles

# make makefiles CC="purify cc"

# make makefiles CC=cc OPT=

#

SHELL = /bin/sh

default: update

update depend clean: Makefiles

$(MAKE) MAKELEVEL= $@

makefiles Makefiles:

$(MAKE) -f Makefile.in MAKELEVEL= Makefiles

**Makefile.in**

SHELL = /bin/sh

DEFS = -I. -D$(SYSTYPE)

CFLAGS = $(DEBUG) $(OPT) $(DEFS)

OPTS = 'CC=$(CC)'

HDRS = debug.h defs.h mygetopt.h param.h

SRCS = mips32\_plot.c main.c mygetopt\_long.c

PROG = tp1-2014-2-bin

$(PROG): $(SRCS) $(HDRS)

$(CC) $(CFLAGS) -o $@ $(SRCS)

update: $(PROG)

:

makefiles Makefiles:

(echo "# Do not edit -- this file documents how this program was built for your machine."; $(SHELL) makedefs) >makedefs.tmp

set +e; \

if cmp makedefs.tmp makedefs.out; then \

rm makedefs.tmp; \

else \

mv makedefs.tmp makedefs.out; \

fi >/dev/null 2>/dev/null

rm -f Makefile; (cat makedefs.out Makefile.in) >Makefile

clean:

rm -f Makefile

cp Makefile.init Makefile

rm -f bin/[!CRS]\* lib/[!CRS]\* include/[!CRS]\* libexec/[!CRS]\* \

junk \*core .nfs\* .pure \*.out \*.out.\* \*.tmp \*.a \*~ \*- \*.orig \

\*.t \*.o \*.bak make.err \*.gmon $(PROG)

**Makefile.init**

# Usage:

# make makefiles [CC=compiler] [OPT=compiler-flags] [DEBUG=debug-flags]

#

# The defaults are: CC=gcc, OPT=-O, and DEBUG=-g. Examples:

#

# make makefiles

# make makefiles CC="purify cc"

# make makefiles CC=cc OPT=

#

SHELL = /bin/sh

default: update

update depend clean: Makefiles

$(MAKE) MAKELEVEL= $@

makefiles Makefiles:

$(MAKE) -f Makefile.in MAKELEVEL= Makefiles