

Some source code walks into a bar.

Immediately, the bartender turns red and yells, "GET OUT OF HERE, WE DON'T SERVE YOUR KIND HERE!"

The source code flares up and says menacingly, "Oh, yeah? Well, why don't you MAKE me!"

--From an anonymous 303 student

David Notkin • Autumn 2009 • CSE303 Lecture 21

A few makefiles

- A standard, simple, example (from the GNU make manual) – very much (but not exactly) like the one you'll need for HW5 and 6
- A more complicated example (also from that manual)
- A simple online demo of configuring and making with a basic piece of GNU software (I already downloaded and gunzipped it)

CSE303 Au09

2

```
edit : main.o kbd.o command.o display.o insert.o search.o files.o \
      utils.o
cc -o edit main.o kbd.o command.o display.o insert.o search.o \
    files.o utils.o
main.o : main.c defs.h
cc -c main.c
kbd.o : kbd.c defs.h command.h
cc -c kbd.c
command.o : command.c defs.h command.h
cc -c command.c
display.o : display.c defs.h buffer.h
cc -c display.c
insert.o : insert.c defs.h buffer.h
cc -c insert.c
search.o : search.c defs.h buffer.h
cc -c search.c
files.o : files.c defs.h buffer.h command.h
cc -c files.c
utils.o : utils.c defs.h
cc -c utils.c
clean :
rm edit main.o kbd.o command.o display.o insert.o search.o
    files.o utils.o
```

Example from
gnu make
manual

3

Complex Makefile Example (Gnu manual)

- "Here is the makefile for the GNU tar program. This is a moderately complex makefile.
- "Because it is the first target, the default goal is `all'. An interesting feature of this makefile is that testpad.h is a source file automatically created by the testpad program, itself compiled from testpad.c.
- "If you type `make' or `make all', then make creates the tar executable, the rmt daemon that provides remote tape access, and the tar.info Info file.
- "If you type `make install', then make not only creates tar, rmt, and tar.info, but also installs them.
- "If you type `make clean', then make removes the `.o' files, and the tar, rmt, testpad, testpad.h, and core files.

CSE303 Au09

4

Continued

- "If you type `make distclean', then make not only removes the same files as does `make clean' but also the TAGS, Makefile, and config.status files. (Although it is not evident, this makefile (and config.status) is generated by the user with the configure program, which is provided in the tar distribution, but is not shown here.)
- "If you type `make realclean', then make removes the same files as does `make distclean' and also removes the Info files generated from tar.texinfo.
- "In addition, there are targets shar and dist that create distribution kits."

CSE303 Au09

5

Makefile

```
# Generated automatically from Makefile.in by configure.
# Un*x Makefile for GNU tar program.
# Copyright (C) 1991 Free Software Foundation, Inc.

# This program is free software; you can redistribute
# it and/or modify it under the terms of the GNU
# General Public License ...
...
...

SHELL = /bin/sh

#### Start of system configuration section.####

srcdir = .
```

CSE303 Au09

6

```

# If you use gcc, you should either run the
# fixincludes script that comes with it or else use
# gcc with the -traditional option. Otherwise ioctl
# calls will be compiled incorrectly on some systems.
CC = gcc -O
YACC = bison -y
INSTALL = /usr/local/bin/install -c
INSTALLDATA = /usr/local/bin/install -c -m 644

```

CSE303 Au09

7

```

# Things you might add to DEFS:
# -DSTDC_HEADERS      If you have ANSI C headers and
#                    libraries.
# -DPOSIX             If you have POSIX.1 headers and
#                    libraries.
# -DBSD42            If you have sys/dir.h (unless
#                    you use -DBSD42), sys/file.h,
#                    and ut_blocks in struct stat.
# -DSG              If you have System V/BSD C
#                    string and memory functions
#                    and headers, sys/asmname.h,
#                    fcntl.h, getmnt, etc. values.
#                    and mdix.h (unless
#                    you use -DSG288).
# -DNO_MKDIR_H       If SG or STDC_HEADERS but do not
#                    include memory.h.
# -DSG288            If SG and you have direct.h
#                    instead of mdix.h.
# -DSIGTYPE=long    If your signal headers
#                    return int, not void.
# ...
# -DNODIR_MISSING   If you lack mkdir and
#                    mkdir system calls.
# -DNOBNAME_MISSING If you lack rename system call.
# -DPRINTF_MISSING  If you lack truncate
#                    system call.
# -DVT              On Version 7 Unix (not
#                    tested in a long time).
# ...

DEFS = -DSIGTYPE=int -DDIRENT -DSTRSTR_MISSING \
      -DPRINTF_MISSING -DBSD42

```

CSE303 Au09

8

```

# Set this to rtapelib.o unless you defined NO_REMOTE, in which case make it empty.
RTAPELIB = rtapelib.o
LIBS =
DEF_AR_FILE = /dev/rmt8
DEFBLOCKING = 20

CDEBUG = -g
CFLAGS = $(CDEBUG) -I. -I$(srkdir) $(DEFS) \
        -DDEF_AR_FILE="$(DEF_AR_FILE)" \
        -DDEFBLOCKING=$(DEFBLOCKING)
LDFLAGS = -g

prefix = /usr/local
# Prefix for each installed program,
# normally empty or 'g'.
binprefix =

# The directory to install tar in.
bindir = $(prefix)/bin

# The directory to install the info files in.
infodir = $(prefix)/info

#### End of system configuration section. ####
CSE303 Au09

```

9

```

SRC1 = tar.c create.c extract.c buffer.c \
      getoldopt.c update.c gnu.c mangle.c
SRC2 = version.c list.c names.c diffarch.c \
      port.c wildmat.c getopt.c
SRC3 = getopt1.c regex.c getdate.y
SRCS = $(SRC1) $(SRC2) $(SRC3)
OBJ1 = tar.o create.o extract.o buffer.o \
      getoldopt.o update.o gnu.o mangle.o
OBJ2 = version.o list.o names.o diffarch.o \
      port.o wildmat.o getopt.o
OBJ3 = getopt1.o regex.o getdate.o $(RTAPELIB)
OBJJS = $(OBJ1) $(OBJ2) $(OBJ3)
AUX = README COPYING ChangeLog Makefile.in \
      makefile.pc configure configure.in \
      tar.texinfo tar.info* texinfo.tex \
      tar.h port.h open3.h getopt.h regex.h \
      rmt.h rmt.c rtapelib.c alloca.c \
      msd_dir.h msd_dir.c texparg.c \
      level-0 level-1 backup-specs testpad.c

```

CSE303 Au09

10

```

.PHONY: all
all: tar rmt tar.info

.PHONY: tar
tar: $(OBJJS)
      $(CC) $(LDFLAGS) -o $@ $(OBJJS) $(LIBS)

rmt: rmt.c
      $(CC) $(CFLAGS) $(LDFLAGS) -o $@ rmt.c

tar.info: tar.texinfo
      makeinfo tar.texinfo

.PHONY: install
install: all
      $(INSTALL) tar $(bindir)/$(binprefix)tar
      -test ! -f rmt || $(INSTALL) rmt /etc/rmt
      $(INSTALLDATA) $(srkdir)/tar.info* $(infodir)

$(OBJJS): tar.h port.h testpad.h
      regex.o buffer.o tar.o: regex.h

```

11

```

# getdate.y has 8 shift/reduce conflicts.
testpad.h: testpad
      ./testpad

testpad: testpad.o
      $(CC) -o $@ testpad.o

TAGS: $(SRCS)
      etags $(SRCS)

.PHONY: clean
clean:
      rm -f *.o tar rmt testpad testpad.h core

.PHONY: distclean
distclean: clean
      rm -f TAGS Makefile config.status

.PHONY: realclean
realclean: distclean
      rm -f tar.info*

```

CSE303 Au09

12

```
.PHONY: shar
shar: $(SRCS) $(AUX)
    shar $(SRCS) $(AUX) | compress \
    > tar -sed -e '/version_string/!d' \
    -e 's/[^0-9.]*\([0-9.]*\)*/\1/' \
    -e q
    version.c`.shar.Z

.PHONY: dist
dist: $(SRCS) $(AUX)
    echo tar -sed \
    -e '/version_string/!d' \
    -e 's/[^0-9.]*\([0-9.]*\)*/\1/' \
    -e q
    version.c` > .fname
    -rm -rf `cat .fname`
    mkdir `cat .fname`
    ln $(SRCS) $(AUX) `cat .fname`
    tar chzf `cat .fname`.tar.Z `cat .fname`
    -rm -rf `cat .fname`.fname
```

CSE303 Au09 13

```
tar.zoo: $(SRCS) $(AUX)
    -rm -rf tmp.dir
    -mkdir tmp.dir
    -rm tar.zoo
    for X in $(SRCS) $(AUX) ; do \
        echo $$X ; \
        sed 's/[^0-9.]*\([0-9.]*\)*/\1/' $$X \
        > tmp.dir/$$X ; done
    cd tmp.dir ; zoo aM ../tar.zoo *
    -rm -rf tmp.dir
```

CSE303 Au09 14

Configuring and building “units”

CSE303 Au09 15

Working in teams

- What is different about working in teams vs. working on your own?

CSE303 Au09 16

One dimension: version control

- Ill-defined, but roughly focused on the mechanics of managing files across teams over time
- Where is the “official” copy of the project?
- How can we explore changes without affecting our teammates?
- What happens if we both try to edit the same file?
- What happens if we make a mistake and corrupt an important file?
- How do I know what code each teammate is working on?
- ...

CSE303 Au09 17

Some of the version control lingo

- Baseline
- Branch
- Change
- Checkout
- Commit
- Conflict
- Head
- Import
- Merge
- Repository
- Revision
- Tag
- Trunk
- Update
- Working copy
- ...

This file is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. In short you are free to share and make derivative works of this file under the condition that you appropriately attribute it, and that you distribute it only under a license identical to this one. <http://creativecommons.org/licenses/by-nc-sa/3.0/>

CSE303 Au09 18

Common version control systems

- Well, there are lots
- Older ones include SCCS, RCS and CVS
- Newer ones include SVN, Git, Bazaar, Mercurial, ClearCase, and many more
- We'll look into SVN more on Friday, and recommend you use it for 303

Questions?