

CSC295 2014S, Class 10: Improved Make

Overview

- Preliminaries.
 - Admin.
 - Questions.
- A quick review of Make.
- Automatic variables.
- Generic rules.
- Another example: Document development.
- How do I use other people's code?
- Standard make rules.
- Other useful techniques.
- Miscellaneous tools and techniques.

Preliminaries

Admin

- We'll spend a few minutes looking at your homework assignments.
- Homework: Create a full make suite for some useful C project, including options for cleaning, packaging, and more.
- Tomorrow's CS table is on lambdas in Java 8.
- Cool CS talk today at 4:30 p.m.

Questions

Sample Makefiles

AK

```
CFLAGS = gcc -Wall -ggdb3
```

Traditionally, the CC command goes separately, so

```
CC = gcc CFLAGS = -Wall
```

```
all: upperToLower
```

```
upperToLower : upperToLower.c charUtils.o charUtils.h  
${CFLAGS} -o upperToLower upperToLower.c charUtils.o
```

Usually we put .o files in the target.

```
charUtils.o : charUtils.c charUtils.h
${CFLAGS} -c charUtils.c
```

Another one

```
.PHONY : clean
clean:
    -rm -rf *.o
```

Why the - before rm? We're not sure. Traditionally, make stops with the first consequent that fails. The - says "keep going".

What happens if we drop the .PHONY? Then Make won't run the rule when a file named `clean` exists and is up to date.

Sam discusses chmod

chmod sets permissions on files and directories.

You can use mnemonics

You can use three digit octal numbers

Each digit: 4 - read, 2 - write, 1 - execute

- First digit: User
- Second digit: Group
- Third digit: All users

Execute generally means "run a program". For directories, it means something about "can access elements of the directory."

A quick review of Make

- Tool for automating construction of projects
- Three basic concepts:
 - Targets (files or actions)
 - Prerequisites
 - Actions
- Tied together with rules

```
target: prerequisites
    action
    action
```

Automatic variables

Goal: Be more efficient with your time and with your rules

```
column-utests: column-utests.o column.o
               $(CC) $(CFLAGS) $(LDFLAGS) -o column-utests column-utests.o column.o
```

Goal: Make that target simpler (other than just using the default rule).

```
LINK = $(CC) $(CFLAGS) $(LDFLAGS)
column-utests: column-utests.o column.o
               $(LINK) -o column-utests column-utests.o column.o
```

What happens if I want to change the target?

```
cut: column-utests.o column.o
      $(LINK) -o column-utests column-utests.o column.o
```

Whoops. I forgot to change it in the next line

Make includes default variables for

- target: \$@
- prereqs: \$^
- First prereq \$<
- Prefix of target \$*
- newer prerequisites ???

```
cut: column-utests.o column.o
      $(LINK) -o $@ $^
```

Generic rules

What if we have common rules for building certain things? Why copy/paste/change? With the percent sign wildcard.

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

Another example: Document development

Let's say we start with Markdown. Markdown.pl translates a .md file to something like HTML.

That gives us incomplete HTML, we need to wrap it using wrap-md-html.

So, let's write rules.

Given a markdown file (.md) create a partial HTML file (.htm) using Markdown.pl

```
file.htm: file.md
         Markdown.pl file.md > file.htm
```

Improved

```
file.htm: file.md
         Markdown.pl $< > $@
```

How do I use other people's libraries

- Best answer: autoconf/automake figures it out for you
 - Hard to learn
- Traditional answer: Make user fill out a config file
- Intermediate answer: pkgconfig

Standard make rules

Other useful techniques

Miscellaneous tools and techniques

```
14:00:21 up 1004 days, 3:56, 2 users, load average: 11.95, 4.28, 3.89 USER TTY FROM LOGIN@ IDLE
JCPU PCPU WHAT root tty1 - 17Feb14 9days 1.90s 1.90s -bash root tty3 - 24Feb14 44days 0.02s 0.01s
-bash
```

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.