Bash: SED and make

Glenn Bruns
CSUMB

Lecture Objectives

After this lecture, you should be able to:

- perform automated edits with sed
- automate builds with make

Make

- Make is a “little language” in which you can define how to build software
- Invented by Stuart Feldman in 1977
- Useful for projects big and small

The key concept of make

Question: does msh4.c need to be compiled?

```
$ ls -l
total 16
-rwxr-xr-x 1 brun1992 shell_faculty 8258 Sep 26 14:09 msh4
-rw-r--r-- 1 brun1992 shell_faculty 3231 Sep 26 14:06 msh4.c
```

What about in this example below?

```
$ ls -l
total 16
-rwxr-xr-x 1 brun1992 shell_faculty 8258 Sep 26 14:09 msh4
-rw-r--r-- 1 brun1992 shell_faculty 3231 Sep 27 10:12 msh4.c
```

By comparing the modification dates of msh4 and msh4.c, you can see if msh4.c needs to be compiled.

Structure of a Makefile

```
$ ls
backup    msh   msh.c      msh-hw-3.c          msh-hw4-tests.txt
Makefile  msh3  msh-hw2.c  msh-hw3-tests.txt   README.txt
$ cat Makefile

msh:    msh.c
        gcc -o msh msh.c

clean:
        rm -f *.o msh
$
```

A make **rule** has the form:

```
target: components
        command 1
        command 2
        …
```

This says:

1. the target depends on the components
2. run the commands to build the target from the components

Running make

```
$ cat Makefile

msh:    msh.c
        gcc -o msh msh.c

clean:
        rm -f *.o msh
$ touch msh.c
$ make
gcc -o msh msh.c
$
```

this changes the modification date of msh.c to the current time

- When you run 'make', it looks for a file named ‘Makefile’ in the current directory.
- If you run 'make <target>', it will build the target if any component is newer than the target, or a component needs to be built
- If you run 'make', the target is the first target in the file.

Running make

```
$ cat Makefile

msh:    msh.c
        gcc -o msh msh.c

clean:
        rm -f *.o msh
$ make clean
rm -f *.o msh
```

clean has no components

If a target has no components, make will simply run the commands

Another example

```
$ cat Makefile

tests:  msh
        ./test1.sh; echo $$?
        ./test2.sh; echo $$?

msh:    msh.c
        gcc -o msh msh.c

clean:
        rm -f *.o msh
```

Use $$ with bash variables within a Makefile

sed: a use case

- Some students use the wrong file name when they submit their work on iLearn.
- I don't want to manually edit student filenames.
- With a bash loop and a sed command I can rename all student files.

sed automates editing

```
$ cat temp3.txt
flam
blam
glam
glum
$
$ sed 's/la/oo/' temp3.txt
foom
boom
goom
glum
$
```

Question: was file temp3.txt modified?

How sed processes a file

1. reads the input one line at a time
2. matches and updates the data according to the editor command
3. writes modified data to standard output (stdout)

Basic edits

```
$ cat foo.csv
First name,Surname,Username,Email address,Last downloaded
Sean,Carpenter,carp3173,secarpenter@csumb.edu,1519229800
Alfredo,Cortez,cort2444,alcortez@csumb.edu,1519229800
Will,Czubiak,czub5930,wczubiak@csumb.edu,1519229800

$ sed 's/,/ /' foo.csv
First name Surname,Username,Email address,Last downloaded
Sean Carpenter,carp3173,secarpenter@csumb.edu,1519229800
Alfredo Cortez,cort2444,alcortez@csumb.edu,1519229800
Will Czubiak,czub5930,wczubiak@csumb.edu,1519229800

$ sed 's/,/ /g' foo.csv
First name Surname Username Email address Last downloaded
Sean Carpenter carp3173 secarpenter@csumb.edu 1519229800
Alfredo Cortez cort2444 alcortez@csumb.edu 1519229800
Will Czubiak czub5930 wczubiak@csumb.edu 1519229800
```

Using regular expressions

```
$ cat cands.csv
P60007697,"Graham, Lindsey O."
P60006723,"Rubio, Marco"
P00003392,"Clinton, Hillary Rodham"
P20003281,"Perry, James R. (Rick)“
$
```

Question: what will `sed ‘s/P.*,//’ cands.c` produce?

```
$ sed 's/P.*,//' cands.csv
 Lindsey O."
 Marco"
 Hillary Rodham"
 James R. (Rick)"
$
```

Beware of output redirection to self

```
$ sed 's/P.*,//' cands.csv > cands.csv
$ cat cands.csv
$
```

with the –i flag ("in place"), sed will modify the input file

```
$ sed -i 's/P.*,//' cands.csv
$ cat cands.csv
 Lindsey O."
 Marco"
 Hillary Rodham"
 James R. (Rick)"
$
```

Using extended regular expressions

```
$ cat foo.csv
First name,Surname,Username,Email address,Last downloaded
Sean,Carpenter,carp3173,secarpenter@csumb.edu,1519229800
Alfredo,Cortez,cort2444,alcortez@csumb.edu,1519229800
Will,Czubiak,czub5930,wczubiak@csumb.edu,1519229800
```

What will `sed -r 's/[a-z]{4}[0-9]{4}//' cands.c` produce?

```
$ sed -r 's/[a-z]{4}[0-9]{4}//' foo.csv
First name,Surname,Username,Email address,Last downloaded
Sean,Carpenter,,secarpenter@csumb.edu,1519229800
Alfredo,Cortez,,alcortez@csumb.edu,1519229800
Will,Czubiak,,wczubiak@csumb.edu,1519229800
```

Putting editing commands in a file

```
$ cat edits.txt
s/la/oo/
s/g/d/
$
$ sed -f edits.txt temp3.txt
foom
boom
doom
dlum
$
```

This is handy when sed is being used to automate tasks.

Advanced: capturing text

```
$ cat foo.csv
First name,Surname,Username,Email address,Last downloaded
Sean,Carpenter,carp3173,secarpenter@csumb.edu,1519229800
Alfredo,Cortez,cort2444,alcortez@csumb.edu,1519229800
Will,Czubiak,czub5930,wczubiak@csumb.edu,1519229800
```

What will `sed -r 's/\([a-z]{4}\)[0-9]{4}/\1/' cands.c` produce?

```
$ sed -r 's/([a-z]{4})[0-9]{4}/\1/' foo.csv
First name,Surname,Username,Email address,Last downloaded
Sean,Carpenter,carp,secarpenter@csumb.edu,1519229800
Alfredo,Cortez,cort,alcortez@csumb.edu,1519229800
Will,Czubiak,czub,wczubiak@csumb.edu,1519229800
```

In sed without -r, you must escape the capture symbols (and) !

Advanced: Deleting/quitting on matches

```
$ cat cands.csv
P60007697,"Graham, Lindsey O."
P60006723,"Rubio, Marco"
P00003392,"Clinton, Hillary Rodham"
P20003281,"Perry, James R. (Rick)“
$
$ sed '/6000/d' cands.csv
P00003392,"Clinton, Hillary Rodham"
P20003281,"Perry, James R. (Rick)“
$
$ sed '/Marco/q' cands.csv
P60007697,"Graham, Lindsey O."
P60006723,"Rubio, Marco“
$
```

Summary

- make automates program building
- sed automates editing