

Lecture 7

Introduction to Make

COP 3344 Introduction to UNIX
Fall 2007

Acknowledgment: These slides are modified versions of Prof. Sudhir Aggarwal's slides

1

Make

- It is a tool to update files that depend on other files

`- make [-f makefile][option] target`

```
Makefile
file1.bak: file1
    ./backup file1
# The line above starts with
# a tab, not with blanks
file2.bak: file2
    ./backup file2
    echo Backed up file2
```

```
$ls -l file*
... Oct 28 11:24 file1
... Oct 28 11:25 file2
$make -f Makefile file1.bak
$make file1.bak
./backup file1
$make file2.bak
./backup file2
echo Backed up file2
Backed up file2
$ls -l file*
... Oct 28 11:24 file1
... Oct 28 11:28 file1.bak
... Oct 28 11:25 file2
... Oct 28 11:28 file2.bak
```

2

Target Rules

- General form
Target: [dependencies]
<tab> command
<tab> command
– Target depends on its dependencies
 - The commands are executed, unless all dependencies have an earlier time stamp than Target
 - These commands generally update Target
- Notes
 - The default makefile is *Makefile* or *makefile*
 - By default, the first target in the makefile is made

3

Make Example

```
Makefile2
# By default, make all the
# files given in 'all'
all: file1.bak file2.bak
file1.bak: file1
    ./backup file1
file2.bak: file2
    ./backup file2
clean:
    rm *.bak
```

```
$ls -l file*
... Oct 28 11:30 file1
... Oct 28 11:30 file1.bak
... Oct 28 11:30 file2
... Oct 28 11:28 file2.bak
$make -f Makefile2
./backup file1
./backup file2
$ls -l file*
... Oct 28 11:30 file1
... Oct 28 11:31 file1.bak
... Oct 28 11:30 file2
... Oct 28 11:31 file2.bak
$make -f Makefile2 clean
rm *.bak
$ls -l file*
... Oct 28 11:30 file1
... Oct 28 11:30 file2
```

4

Macros and Multiple Dependencies

```
Makefile
CC=gcc
CFLAGS=-g -Wall
a.out: myprog1.o myprog2.o
    $(CC) $(CFLAGS) myprog1.o myprog2.o
myprog1.o: myprog1.c myprog.h
    $(CC) $(CFLAGS) -c myprog1.c
myprog2.o: myprog2.c myprog.h
    $(CC) $(CFLAGS) -c myprog2.c
```

```
$ls -l
... Oct 28 12:24 Makefile
... Oct 28 12:26 myprog1.c
... Oct 28 12:21 myprog2.c
... Oct 28 12:20 myprog.h
$make
```

```
gcc -g -Wall -c myprog1.c
gcc -g -Wall -c myprog2.c
gcc -g -Wall myprog1.o myprog2.o
$ls
a.out Makefile myprog1.c
myprog1.o myprog2.c myprog2.o
myprog.h
$touch myprog1.c
$make
gcc -g -Wall -c myprog1.c
gcc -g -Wall myprog1.o myprog2.o
$touch myprog.h
$make
gcc -g -Wall -c myprog1.c
gcc -g -Wall -c myprog2.c
gcc -g -Wall myprog1.o myprog2.o
```

5