Compiling in Unix

Makefiles, Common Errors, and Debugging
CS-121

Using make

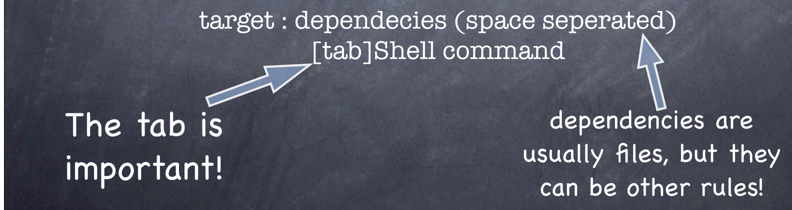
- make is a program which simplifies the task of compiling programs
- Often a program is made up of many C++ source files (.cpp), make keeps track of which have changed and only compiles those
- Make knows about C++ and compiling files. It calls g++ for you (often you just type make)

make rules

- Make compiles programs based on rules which you specify in a file named "Makefile"
- Each rule defines 3 things
  - The program you want to create (target)
  - The source (.cpp) file or files needed to create the target (dependencies)
  - Shell commands for creating the target from the dependencies

make rules

- The basic format of rules in the Makefile is as follows:



```
target : dependecies (space seperated)
        [tab]Shell command
```

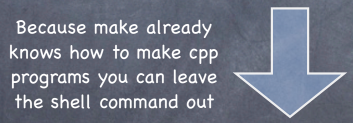
The tab is important!

dependencies are usually files, but they can be other rules!

make rules

- Example:

```
helloworld : helloworld.cpp
[tab] g++ helloworld.cpp -o helloworld
```



Because make already knows how to make cpp programs you can leave the shell command out

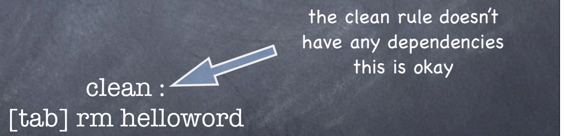
```
helloworld : helloworld.cpp
```

make rules

- You can have any number of rules in a makefile
  - The first rule is the default rule -- it's executed by just typing make
  - Other rules you need to type make [targetname]

make rules

- You can have phony rules
  - phony rules don't make anything but they are useful



the clean rule doesn't have any dependencies this is okay

```
clean :
[tab] rm helloword
```

Makefile example