

EEB 177 Lecture 2

Topics

- ▶ Shell basics

What is Unix?

- ▶ operating system written by AT&T Bell scientists in the 70s
- ▶ many variants today: OpenBSD, Sun Solaris, Apple OS X, Linux
- ▶ multi-user, network-oriented, stores data as plain text files

The Shell

why use it? * easier to log commands * easier to script * writing GUIs takes a long time

What is a shell?

- ▶ program which displays command line
- ▶ interface for communicating with core OS (kernel)
- ▶ several versions. Default on OS X is bash
- ▶ check now. Open terminal and type `echo $SHELL`

Why use Linux?

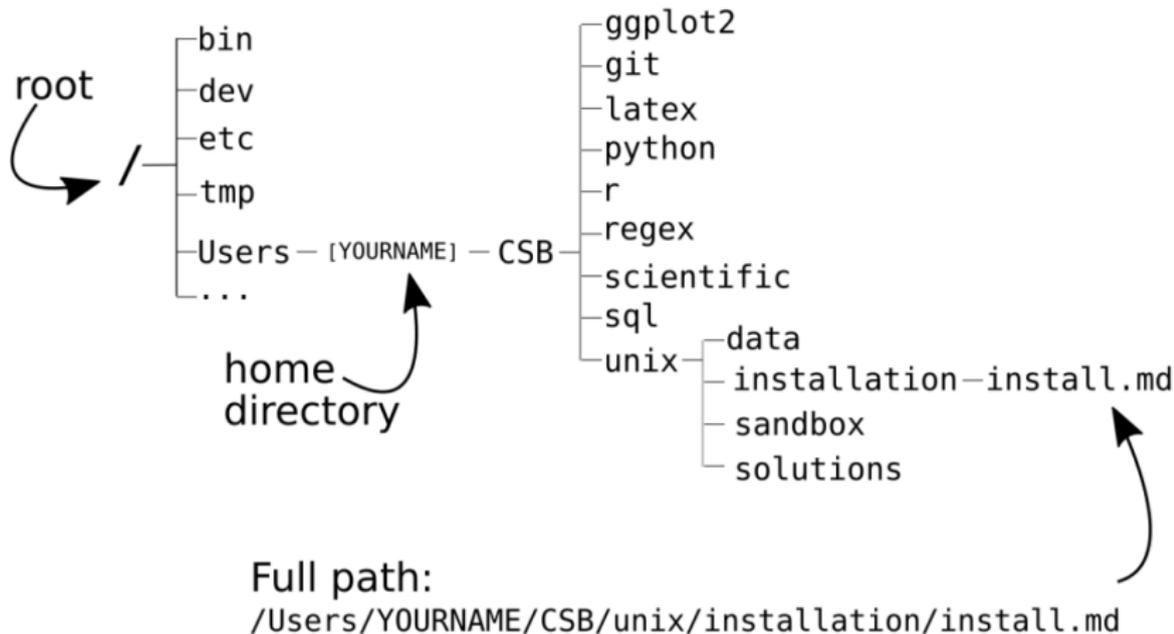
- ▶ environment written by programmers/scientists for programmers/scientists
- ▶ 100s of super efficient small programs available, easily strung together
- ▶ text-based
- ▶ stable, many tutorials
- ▶ can't avoid it!

Directory structures

know these

- ▶ root / the topmost directory in the hierarchy
- ▶ home /Users/YourName where your stuff lives
 - ▶ shortcut to home directory: ~
 - ▶ print your home directory echo \$HOME

Figure 1.1: An exemplary directory structure in the OS X operating system (in Ubuntu, the directory `home` is equivalent to the directory `Users` in OS X).



Using the shell

- ▶ open terminal in your virtual box using `Ctrl + Alt + t`



Ctrl + A	Go to the beginning of the line
Ctrl + E	Go to the end of the line
Ctrl + L	Clear the screen
Ctrl + U	Clear the line before the cursor position
Ctrl + K	Clear the line after the cursor
Ctrl + C	Kill the command that is currently running
Ctrl + D	Exit the current shell
Alt + F	Move cursor forward one word (in OS X, <code>Esc + F</code>)
Alt + B	Move cursor backward one word (in OS X, <code>Esc + B</code>)

Spaces

Challenge 1: Open a terminal and create a file called “badly named file.txt” using the touch command * try copying this file to your home directory using with the cp command * cp filename directory * what just happened?

Naming files

- ▶ file names can contain letters, numbers, and . , _
- ▶ names that start with a period will be hidden `.config`
- ▶ avoid using uppercase letters
- ▶ avoid naming files with same name as a unix command

Overview of navigating files and directories

we will work through the software carpentry lesson on this topic here <http://swcarpentry.github.io/shell-novice/02-filedir/>

Getting help in Unix

- ▶ `man` brings up the manual
- ▶ try `man ls`
- ▶ google and stackoverflow also extremely useful

What if no manual is available?

Sometimes you will see this message when you try to access the manual.

```
$ man cd No manual entry for cd This can happen with  
commands that are part of the shell program itself
```

```
$ type cd cd is a shell builtin To see the manual for these  
commands type man bash
```

Getting around in a manual

Manuals can be long and intimidating. To find a specific character string within a manual, type `/stringname` within the manual. You can press `n` to see the next occurrence of the string.

Use the manual to find the arguments of the `cd` command.

cd

- ▶ `cd ..` move one directory up
- ▶ `cd /` move to the root directory
- ▶ `cd` move to your home directory



try going to the sandbox directory in CSB/unix

`pwd`

- ▶ prints the path of your current directory
- ▶ try it!

ls

- ▶ lists files and subdirectories of current directories
- ▶ takes arguments
 - ▶ `ls -a` lists all files (including hidden)
 - ▶ `ls -l` lists files with details
 - ▶ `ls -h` gives sizes in human readable format

absolute and relative paths

- ▶ the absolute path is the entire path starting from the root, /
 - ▶ `/Users/michael_alfaro/tools`
- ▶ the relative path is defined relative to the current directory
 - ▶ if we were in the `/tools` directory above, we could get to the root using the relative path: `cd ../../..`

we could also go there using the absolute path: `cd /`

Challenge 2

- ▶ Go to your home directory.
- ▶ Navigate to the sandbox within the CSB/unix directory.
- ▶ Use a relative path to go to the data within the python directory.
- ▶ Use an absolute path to go to the sandbox directory within ggplot2.
- ▶ Return to the data directory within the python directory.

cp

- ▶ copies files
- ▶ `cp file_to_copy path_to_destination`

```
# If you specify the full path,  
# your current location does not matter  
$ cp ~/CSB/unix/data/Buzzard2015_about.txt  
    ~/CSB/unix/sandbox/  
# assume your current location is the unix sandbox  
# we can use a relative path  
$ cp ../data/Buzzard2015_about.txt .  
# the dot is shorthand to say "here"  
# rename the file in the copying process  
cp ../data/Buzzard2015_about.txt ~/buzz-2015.txt
```

mv

moves (or renames) files

▶ `mv file_to_move destination`

move the file to the data directory

```
$ mv Buzzard2015_about2.txt ../data/
```

rename a file

```
$ mv ../data/Buzzard2015_about2.txt Buzzard-2015.txt
```

touch

- ▶ creates an empty file (or updates date of access of existing file)
- ▶ `touch my_file_name.txt`