An Introduction to UNIX

Jason Stajich
UC Riverside
@hyphaltip @stajichlab

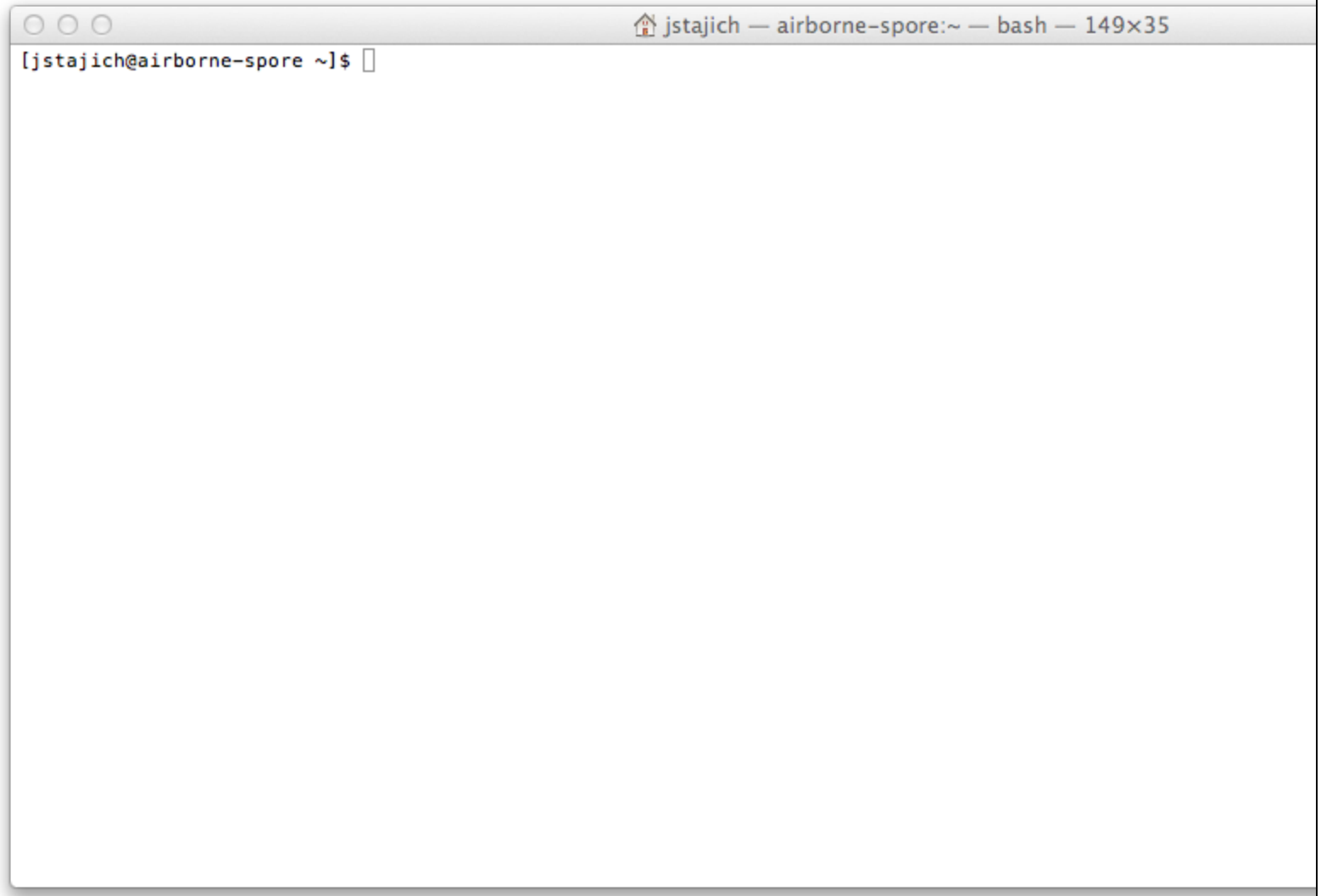
Concepts to cover today

- Using the terminal on your mac laptop
- The command-line interface to UNIX
- Directories & Files
- Logging into biocluster
- Running programs

Using the terminal

- In the Applications / Utilities Folder
- Are other programs useful with add-ons (iTerm)

- Can copy and paste within the terminal or from other programs


jstajich — airborne-spore:~ — bash — 149×35
[jstajich@airborne-spore ~]$

Command prompt

- This is where all the work happens
- The prompt will have a '$' or a '%' usually
- Sometimes the prompt will include the current directory you are in

```
me@biocluster:/data/squid/ $
john@tombstone:/home/john %
```

- It can be customized so that colors, extra notifications can show up
- Running a program on the command line by typing and hitting enter

```
$ ls
```

Directories (folders)

- Listing files in a directory

```
$ ls
$ ls -l
```

- making a directory

```
$ mkdir mynewdir
```

- changing directories

```
$ cd mynewdir # go in
$ cd ..       # go back
$ cd ~        # go HOME
$ cd          # go HOME
$ cd ../mynewdir/text # go to another directory
```

- removing directories

```
$ rmdir mynewdir
```

Listing contents

- Listing files in a directory with `ls`
- Other options `ls -l` for long listing
- The long listint can give you information on how big the files are too
- `ls -lt` shows listing ordered by date
- `man ls` will give you more information on

```
$ cd /srv/projects/db/ncbi/current
$ ls -l
total 96376844
-rw-r--r-- 1 jenkins-agent staff 43320522376 Feb  1  2013 est_others
-rw-r--r-- 1 jenkins-agent staff    17975066 Feb  1  2013 est_others-download.log
-rw-r--r-- 1 jenkins-agent staff  1364382091 Feb  1  2013 est_others.00.nhr
-rw-r--r-- 1 jenkins-agent staff    92999188 Feb  1  2013 est_others.00.nin
-rw-r--r-- 1 jenkins-agent staff    61999408 Feb  1  2013 est_others.00.nnd
-rw-r--r-- 1 jenkins-agent staff      242236 Feb  1  2013 est_others.00.nni
-rw-r--r-- 1 jenkins-agent staff   276087180 Feb  1  2013 est_others.00.nsd
-rw-r--r-- 1 jenkins-agent staff     6009232 Feb  1  2013 est_others.00.nsi
-rw-r--r-- 1 jenkins-agent staff  1052297716 Feb  1  2013 est_others.00.nsq
-rw-r--r-- 1 jenkins-agent staff  1179327368 Feb  1  2013 est_others.01.nhr
```

Files

- Files are the oh of what will work with in UNIX
- Can create them in editors and many other programs
- The command `touch` will create an empty file
- Files are the results of other programs running

Moving and renaming files

- Moving files or directories to new locations
- Renaming files is just moving them to a new name
- Multiple things can be moved into a directory as long as the directory is the last argument

```
$ touch testfil1 # make an empty file called 'testfil1'
$ mv testfil1 testfile1 # rename the filename
$ mkdir newdir # make a new directory
$ mv testfile1 newdir # put the file in a directory
$ touch testfile2 testfile3 # make 2 more files
$ mv testfile2 testfile3 newdir # move them both into the directory
# mv newdir oldir # rename the newdir to 'oldir'
# mkdir another_dir # make a new folder
# mv oldir another_dir # move the 'oldir' folder into this new dir
```

Copying files

- the command `cp` for copying
- If you want to copy a whole directory use `cp -r` for recursive copy
- Tools like `rsync` can copy but are smart enough to only copy what's changed

Shell expansions and using the *

- The shell can expand file or directory names for you
- If you wanted to get all the files that started with 'test' and move them into a folder

```
$ touch test1 test2 test_ing test.out notes
$ mkdir my_tests
$ ls
my_tests notes  test.out  test1  test2  test_ing  testfile1
$ mv test* my_tests
$ ls
my_tests  notes
$ ls my_tests
test.out  test1  test2  test_ing  testfile1
```

  - The * means match anything, 0-multiple times
  - Other options '?' mean match one chacter, 0 or 1 times

Read the manual

- `man` is a built in utility that prints out help and usage for programs
- very useful to find out what a tool does or how to use it
- `man man` tells you how to use 'man'

```
NAME
man - an interface to the on-line reference manuals

SYNOPSIS
man  [-C  file]  [-d]  [-D]  [--warnings[=warnings]]
[-R encoding] [-L locale] [-m system[,...]] [-M path] [-S
list] [-e extension] [-i|-I] [--regex|--wildcard]
[--names-only] [-a]  [-u]  [--no-subpages]  [-P  pager]
[-r prompt] [-7] [-E encoding] [--no-hyphenation]
[--no-justification]

DESCRIPTION
man  is  the system's manual pager. Each page argument given
to man is normally the name of a program, utility or function.
The manual page associated with each of these arguments is
then found and displayed. A section, if  provided,  will
direct man to look only in that section of the manual.  The
default action is to search in all of the available sections,
following a pre-defined order and to show only the first page
found,  even  if page exists in several sections.
```

Some reminders / gotchas

- UNIX is case sensitive
- spaces in filenames and folders are annoying - to use them you need to Escape them with a \ or enclose in quotes

```
$ mkdir "The End"
$ rmdir The End # this will fail
$ rmdir The\ End #or
$ rmdir "The End"
```

Home

- This is your homebase directory, when you login you will end up here
- `cd` with no other arguments will take you here
- You'll have some configuration files and other things stored here
- For this class most of your work will take place in this folder

Reading files: more or less

- seeing contents of files with `more` or `less`
- page at a time view
- search within, use the '/' to search

Creating files and Editors

- Creating text files with editors in UNIX - this is what you have to do to program.
- Typically used tools are `vi` `emacs` `nano`
- On Mac - try TextEdit or TextWrangler
- Practice using the text editor

Command line programs

- anatomy of running a program
- the program name comes first. It can be just the name `ls` or the full path to the program `/bin/ls`

```
$ ls
$ /bin/ls
```

- arguments to the program come next. Some can be named with an argument like `program -i input -o output`
- or the program can just have a set of arguments like `cat file1 file2 file3`
- Instead of having output be printed to the screen it can be captured to a file

```
$ ls > listing
$ more listing
$ cat file1 file2 file3 > all_files
```

First homework

- Not a graded homework, but is to practice UNIX making files, reading them, creating directories, navigating
- read the UNIX To the Rescue chapter on UNIX
- If you didn't buy the book yet Read 'Part 1' on the UNIX and Perl primer http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v3.1.1.html.