

Working With Unix

Scott A. Handley*
September 14, 2015

*Adapted from UNIX introduction material created by Dr. Julian Catchen

What is UNIX?

- An operating system (OS)
- Designed to be multiuser and multitasking
- Lots of flavors, some shared characteristics
 - *Kernal*: core program loaded into memory that controls hardware allocation
 - *Standard utility programs*: *cp* to copy, *mv* to move, etc. Interaction through the shell
 - *System configuration files*: read by *kernal* and *configuration files*. Provide instructions on how to work
- Frequently have a GUI nowadays, but almost everything can be done without the GUI

Why bother?

- The tools and concepts discussed today are absolutely necessary for you to install, use and interpret the results from a significant majority of bioinformatic programs today
- Foundational for all subsequent analysis
- It just makes you a better person

1000% necessary for bioinformatics

To use TopHat, you will need the following programs in your PATH:

- bowtie2 and bowtie2-align (or bowtie)
- bowtie2-inspect (or bowtie-inspect)
- bowtie2-build (or bowtie-build)
- samtools

To install TopHat from source package, unpack the tarball and change directory to the package directory as follows:

```
tar zxvf tophat-2.0.0.tar.gz  
cd tophat-2.0.0/
```

Configure the package, specifying the install path and the library dependencies as needed (see the [Getting started](#) guide for details):

```
./configure --prefix=<install_prefix> --with-boost=<boost_install_prefix> --with-bam=<samtools_install_prefix>
```

Finally, build and install TopHat:

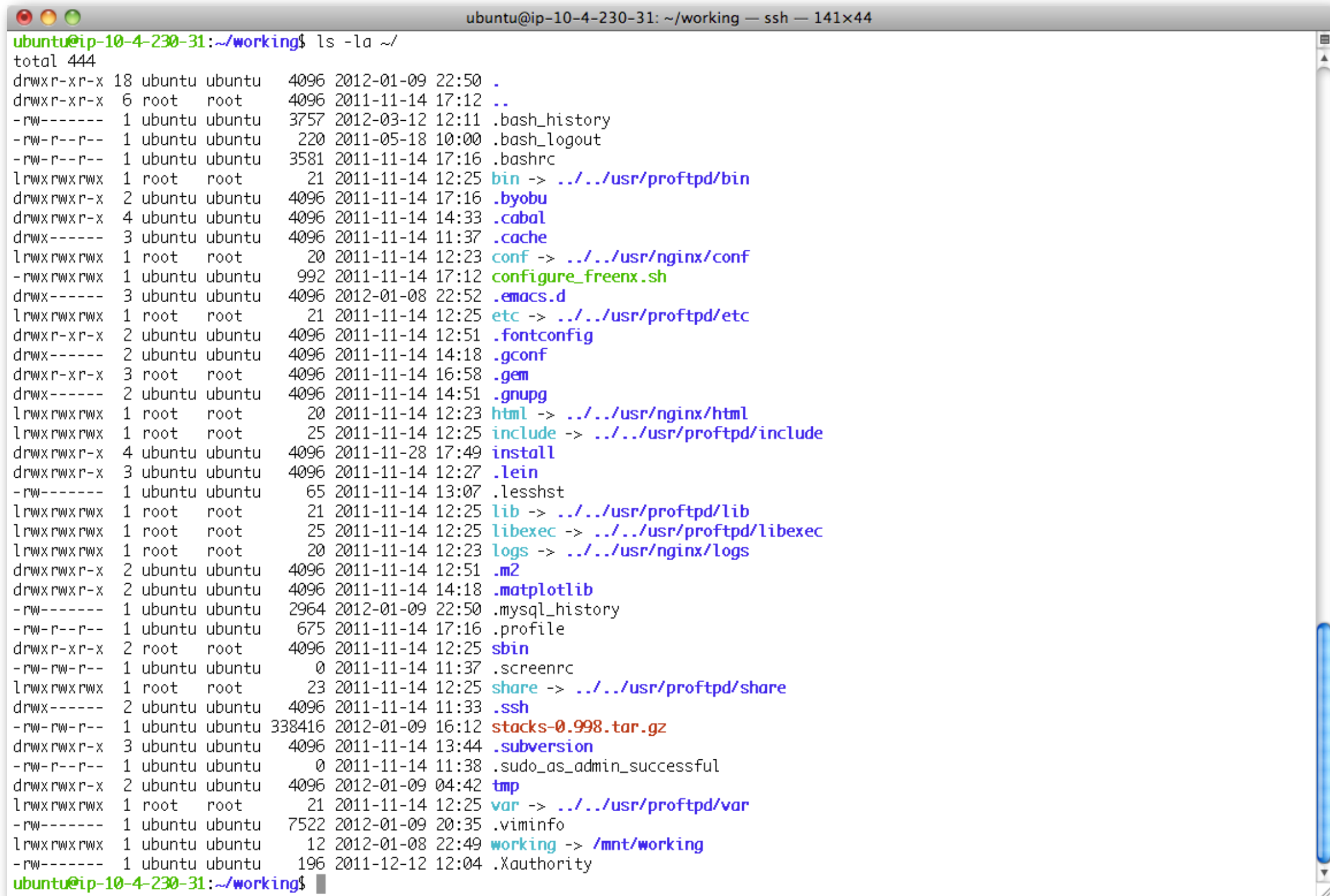
```
make  
make install
```

What computers can run Unix?



- Apple OS X Macs
- Google's Android phones
- Most airplane entertainment systems
- Wireless internet routers

The Terminal Window



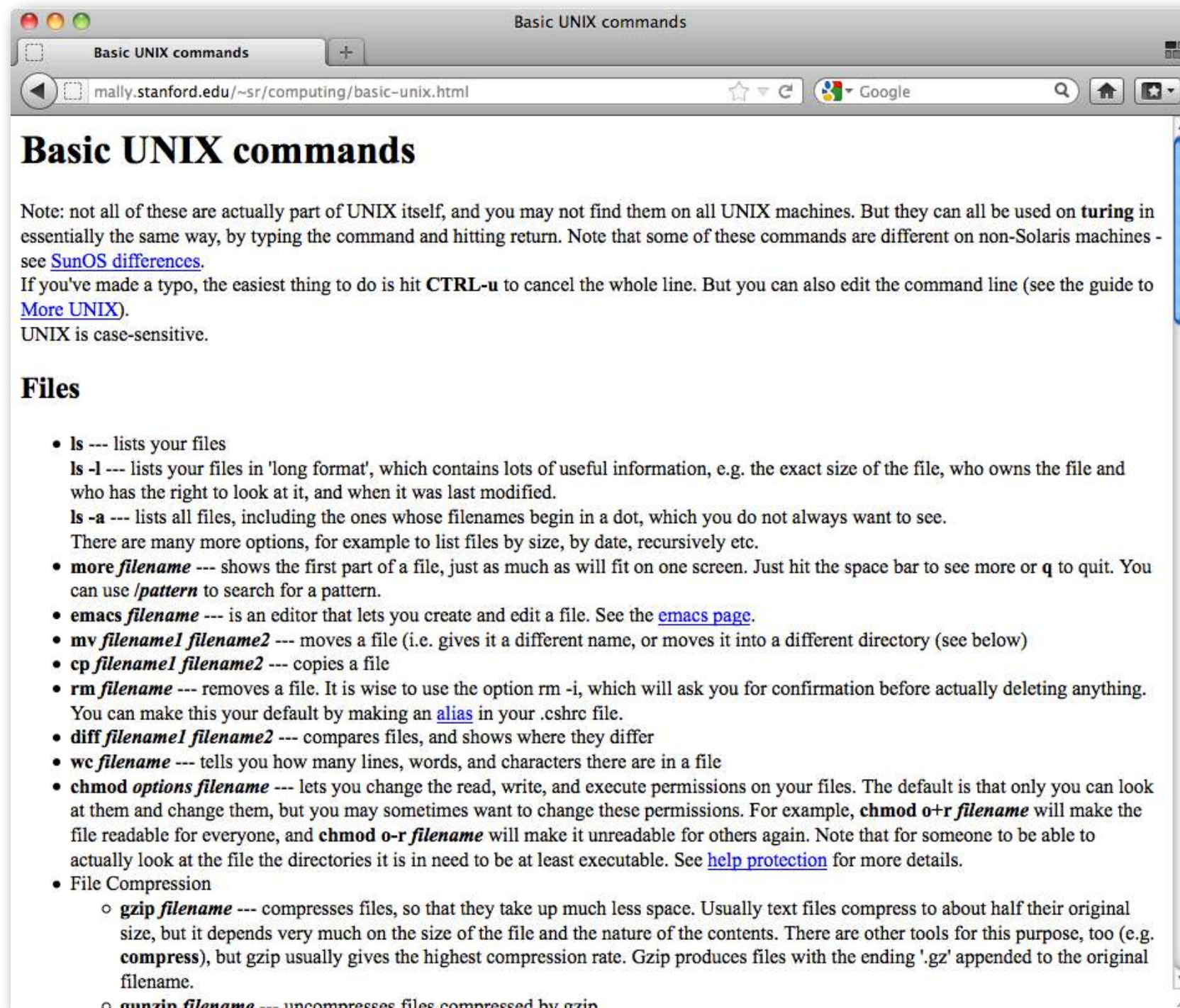
```
ubuntu@ip-10-4-230-31: ~/working — ssh — 141x44
ubuntu@ip-10-4-230-31:~/working$ ls -la ~/
total 444
drwxr-xr-x 18 ubuntu ubuntu 4096 2012-01-09 22:50 .
drwxr-xr-x  6 root    root   4096 2011-11-14 17:12 ..
-rw-r----- 1 ubuntu ubuntu 3757 2012-03-12 12:11 .bash_history
-rw-r--r--  1 ubuntu ubuntu  220 2011-05-18 10:00 .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3581 2011-11-14 17:16 .bashrc
lrwxrwxrwx  1 root    root    21 2011-11-14 12:25 bin -> ../../usr/proftpd/bin
drwxrwxr-x  2 ubuntu ubuntu 4096 2011-11-14 17:16 .byobu
drwxrwxr-x  4 ubuntu ubuntu 4096 2011-11-14 14:33 .cabal
drwx----- 3 ubuntu ubuntu 4096 2011-11-14 11:37 .cache
lrwxrwxrwx  1 root    root    20 2011-11-14 12:23 conf -> ../../usr/nginx/conf
-rwxrwxrwx  1 ubuntu ubuntu  992 2011-11-14 17:12 configure_freenx.sh
drwx----- 3 ubuntu ubuntu 4096 2012-01-08 22:52 .emacs.d
lrwxrwxrwx  1 root    root    21 2011-11-14 12:25 etc -> ../../usr/proftpd/etc
drwxr-xr-x  2 ubuntu ubuntu 4096 2011-11-14 12:51 .fontconfig
drwx----- 2 ubuntu ubuntu 4096 2011-11-14 14:18 .gconf
drwxr-xr-x  3 root    root   4096 2011-11-14 16:58 .gem
drwx----- 2 ubuntu ubuntu 4096 2011-11-14 14:51 .gnupg
lrwxrwxrwx  1 root    root    20 2011-11-14 12:23 html -> ../../usr/nginx/html
lrwxrwxrwx  1 root    root    25 2011-11-14 12:25 include -> ../../usr/proftpd/include
drwxrwxr-x  4 ubuntu ubuntu 4096 2011-11-28 17:49 install
drwxrwxr-x  3 ubuntu ubuntu 4096 2011-11-14 12:27 .lein
-rw-r----- 1 ubuntu ubuntu  65 2011-11-14 13:07 .lessht
lrwxrwxrwx  1 root    root    21 2011-11-14 12:25 lib -> ../../usr/proftpd/lib
lrwxrwxrwx  1 root    root    25 2011-11-14 12:25 libexec -> ../../usr/proftpd/libexec
lrwxrwxrwx  1 root    root    20 2011-11-14 12:23 logs -> ../../usr/nginx/logs
drwxrwxr-x  2 ubuntu ubuntu 4096 2011-11-14 12:51 .m2
drwxrwxr-x  2 ubuntu ubuntu 4096 2011-11-14 14:18 .matplotlib
-rw-r----- 1 ubuntu ubuntu 2964 2012-01-09 22:50 .mysql_history
-rw-r--r--  1 ubuntu ubuntu  675 2011-11-14 17:16 .profile
drwxr-xr-x  2 root    root   4096 2011-11-14 12:25 sbin
-rw-rw-r--  1 ubuntu ubuntu  0 2011-11-14 11:37 .screenrc
lrwxrwxrwx  1 root    root    23 2011-11-14 12:25 share -> ../../usr/proftpd/share
drwx----- 2 ubuntu ubuntu 4096 2011-11-14 11:33 .ssh
-rw-rw-r--  1 ubuntu ubuntu 338416 2012-01-09 16:12 stacks-0.998.tar.gz
drwxrwxr-x  3 ubuntu ubuntu 4096 2011-11-14 13:44 .subversion
-rw-r--r--  1 ubuntu ubuntu  0 2011-11-14 11:38 .sudo_as_admin_successful
drwxrwxr-x  2 ubuntu ubuntu 4096 2012-01-09 04:42 tmp
lrwxrwxrwx  1 root    root    21 2011-11-14 12:25 var -> ../../usr/proftpd/var
-rw-r----- 1 ubuntu ubuntu 7522 2012-01-09 20:35 .viminfo
lrwxrwxrwx  1 ubuntu ubuntu  12 2012-01-08 22:49 working -> /mnt/working
-rw-r----- 1 ubuntu ubuntu  196 2011-12-12 12:04 .Xauthority
ubuntu@ip-10-4-230-31:~/working$
```

Make it comfortable to work in:

- Resize the window
- Change your font size
- Open multiple terminal windows

How to get help

google “unix commands”



The screenshot shows a web browser window with the title 'Basic UNIX commands'. The address bar displays the URL 'mally.stanford.edu/~sr/computing/basic-unix.html'. The page content is titled 'Basic UNIX commands' and includes a note about command availability, a tip on using CTRL-u, and a section on file commands.

Basic UNIX commands

Note: not all of these are actually part of UNIX itself, and you may not find them on all UNIX machines. But they can all be used on **turing** in essentially the same way, by typing the command and hitting return. Note that some of these commands are different on non-Solaris machines - see [SunOS differences](#).
If you've made a typo, the easiest thing to do is hit **CTRL-u** to cancel the whole line. But you can also edit the command line (see the guide to [More UNIX](#)).
UNIX is case-sensitive.

Files

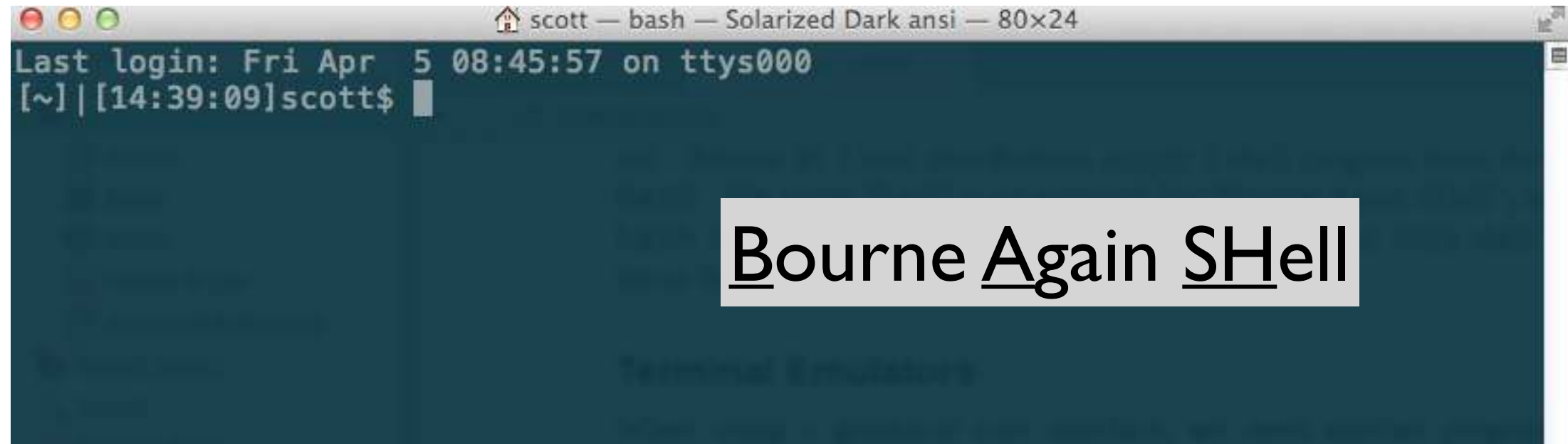
- **ls** --- lists your files
 - ls -l** --- lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.
 - ls -a** --- lists all files, including the ones whose filenames begin in a dot, which you do not always want to see.
- There are many more options, for example to list files by size, by date, recursively etc.
- **more filename** --- shows the first part of a file, just as much as will fit on one screen. Just hit the space bar to see more or **q** to quit. You can use */pattern* to search for a pattern.
- **emacs filename** --- is an editor that lets you create and edit a file. See the [emacs page](#).
- **mv filename1 filename2** --- moves a file (i.e. gives it a different name, or moves it into a different directory (see below))
- **cp filename1 filename2** --- copies a file
- **rm filename** --- removes a file. It is wise to use the option **rm -i**, which will ask you for confirmation before actually deleting anything. You can make this your default by making an [alias](#) in your **.cshrc** file.
- **diff filename1 filename2** --- compares files, and shows where they differ
- **wc filename** --- tells you how many lines, words, and characters there are in a file
- **chmod options filename** --- lets you change the read, write, and execute permissions on your files. The default is that only you can look at them and change them, but you may sometimes want to change these permissions. For example, **chmod o+r filename** will make the file readable for everyone, and **chmod o-r filename** will make it unreadable for others again. Note that for someone to be able to actually look at the file the directories it is in need to be at least executable. See [help protection](#) for more details.
- File Compression
 - **gzip filename** --- compresses files, so that they take up much less space. Usually text files compress to about half their original size, but it depends very much on the size of the file and the nature of the contents. There are other tools for this purpose, too (e.g. **compress**), but gzip usually gives the highest compression rate. Gzip produces files with the ending **'.gz'** appended to the original filename.
 - **gunzip filename** --- uncompresses files compressed by gzip

Textbook

The Linux[®] Command Line

William E. Shotts, Jr.

First commands

A screenshot of a terminal window with a dark blue background and light gray text. The window title bar at the top shows a home icon, the text 'scott — bash — Solarized Dark ansi — 80x24', and window control buttons. The terminal content shows 'Last login: Fri Apr 5 08:45:57 on ttys000' followed by a prompt '[~] | [14:39:09]scott\$' and a cursor. A semi-transparent gray box is overlaid on the right side of the terminal, containing the text 'Bourne Again SHell'.

```
scott — bash — Solarized Dark ansi — 80x24
Last login: Fri Apr 5 08:45:57 on ttys000
[~] | [14:39:09]scott$
```

Bourne Again SHell

NOTE! There are differences between Linux and Mac OS

```
$ cal
```

```
$ cal -jy
```

```
$ man cal
```

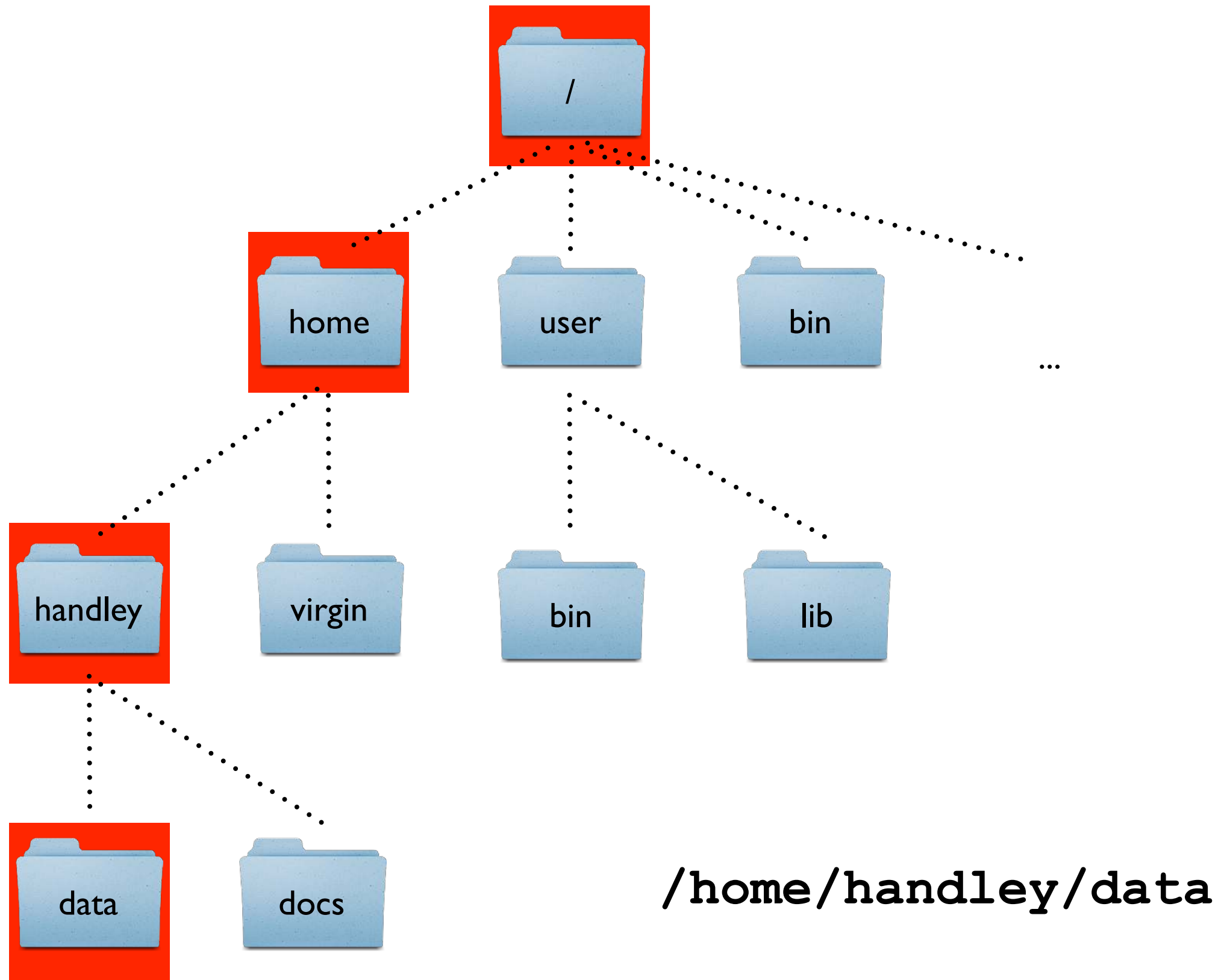
NOTE! CoMmaNdS aRe cAse-SENsiTiVE

Directory Navigation

Hierarchical Folder Structure



Paths



Navigation

```
$ pwd
```

print working directory

```
$ ls
```

list directory contents

```
$ mkdir test1
```

make directory

```
$ cd test1
```

change directory

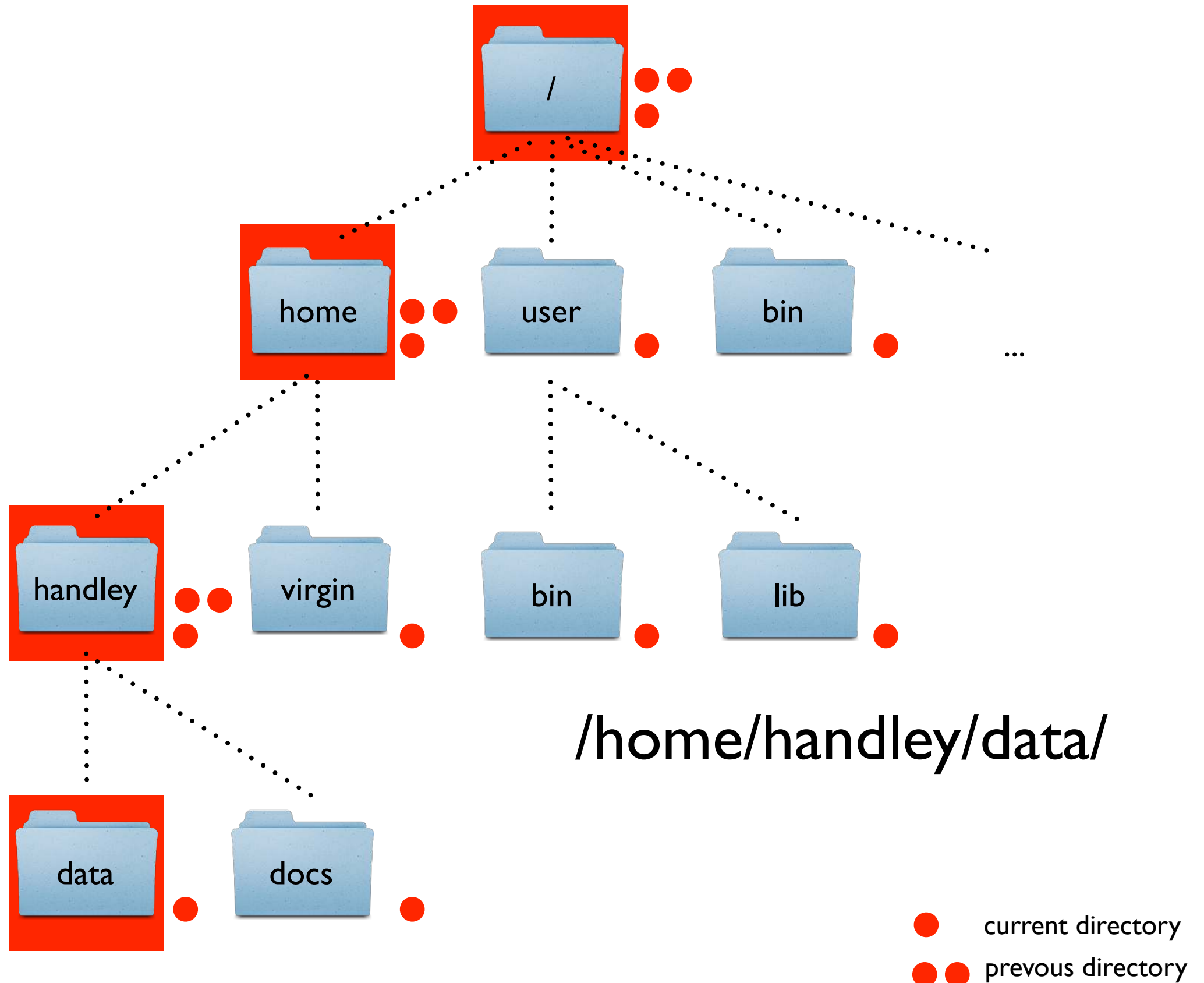
.....

```
$ mkdir test1.1 test 1.2
```

```
$ cd test1.1
```

```
$ pwd
```

Relative Paths



More Navigation

```
$ cd ~
```

take you home

```
$ cd ~/test1/test1.1
```

```
$ cd -
```

previous directory

ls

```
$ cd ~
```

```
$ ls
```

```
$ ls -l
```

ls has lots of options

```
$ man ls
```

manual pages

```
$ ls -lt
```

```
$ ls -lrt
```

```
$ ls -lrth
```

If your typing you are doing it wrong

Tab completion

Tab once to complete uniquely

Tab twice to see all possible options

Up/Down arrows

Cycle through history

```
$ !!
```

```
$ history
```

```
$ !10
```

working with text

<code>more</code>	<code>head</code>	<code>tail</code>	<code>cat</code>
view a text file one screen full at a time	view the top 10 lines of a file	view the last 10 lines of a file	spit the whole file at once
space-bar: scroll q: quit	-n num controls the number of lines	-n num controls the number of lines	

Text Editors

- Wars have been fought over this
- Complex but powerful
- Many benefits for coding, not so many for basic text editing
- Can launch in-line with the terminal. No GUI required
- Emacs, vi, nano, pico ...

```
$ nano filename
```

Explore a FASTA file

Important File Formats

FASTA

```
>HWI-ST0747:162:C03AJACXX:3:1108:19763:106771 1:N:0:  
TTTGTCTGCAGGGGGACACGTCAAAGTCAAACGCAGGCAAGTTTGTGTTTATGTCCAGTGGATCTTTTGATTTT  
ACATACTGCAGGGTCAGGAGGATTATCTCCTCTGCAAGGTAACGCCTGCTGTAACCGTTGTTCTTCATCCTTTT  
CCTAACTGCAGGGCTGTCTTGTCTCAGGTCTGACAAGACATATGCAGGGCTCAATTTGAGATAATTGCTCAATATA
```

FASTQ

```
@HWI-ST0747:162:C03AJACXX:3:1108:19763:106771 1:N:0:  
TTTGTCTGCAGGGGGACACGTCAAAGTCAAACGCAGGCAAGTTTGTGTTTATGTCCAGTGGATCTTTTGATTTT  
+  
<?@DDDDDFHHFBB@GGIACFHGGHBGHHGCDHBEAHACHI=@CH.=7ACAHHADECDBCC66(6>@C>5@CACCA
```

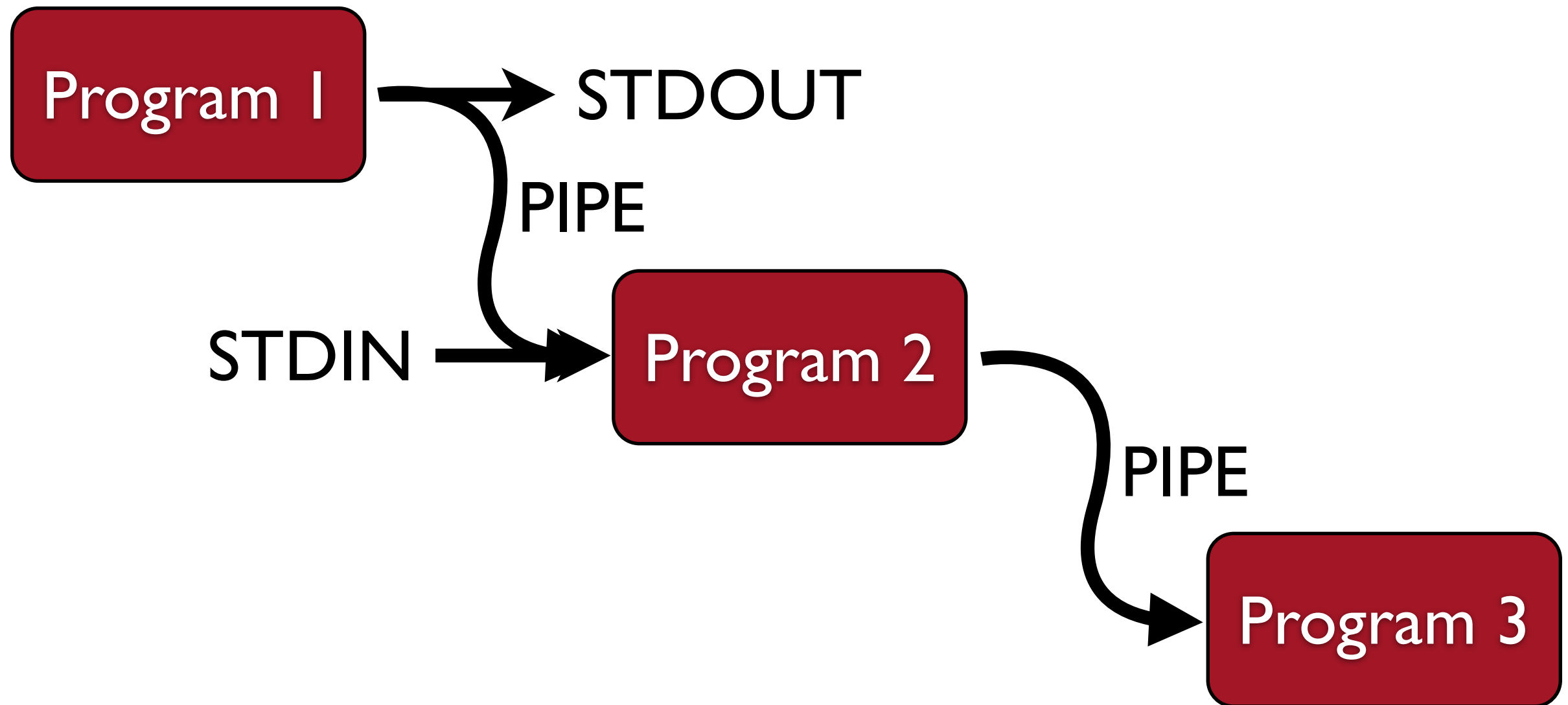
Obtaining Example Files

- wget: retrieves files using HTTP, HTTPS and FTP, the most widely-used Internet protocols.
- Steps
 - Move to your home directory (cd ~) and make a directory called test_data
 - Move into the test_data directory
 - wget http://evomicsorg.wpengine.netdna-cdn.com/wp-content/uploads/2014/09/fasta_example.txt
 - wget http://evomicsorg.wpengine.netdna-cdn.com/wp-content/uploads/2014/09/annot_contigs.txt

Data analysis with pipes



Plumbing: Connecting pipes



Explore a FASTA file with Unix commands and pipes

```
$ more fasta_example.txt
```

```
$ tail fasta_example.txt
```

```
$ mv fasta_example.txt example.fasta
```

```
$ nano example.fasta    ##ctrl-x to exit
```

How many fasta entires are there in this file?

grep

- grep: get regular expressions
 - Very powerful! One of your best friends!
 - For our purposes, we will just use it to "get" simple things
 - For example, what can we get to count in a fasta file to enumerate all of the entries?

```
$ grep ">" example.fasta
```

```
$ grep ">" example.fasta > fasta.headers
```

Counting and Piping

- `wc`: counts words, lines and bytes for stdin

```
$ wc fasta.headers
```

- Piping is accomplished with the `|`

```
$ ls -l ~ | head -n 3
```

- How can you pipe `grep` and `wc` to count the number of fasta entries in `example.fasta`

```
$ grep ">" example.fasta | wc -l
```

- Read the man page for `grep` to find a way that `grep` can do this without piping to `wc`

Slice and Dice a Tab-delimited File