



Linux Boot Camp

Jack Biggs

9 Sep 2014

Connecting

SSH

Windows users: PuTTY

Mac users: Terminal

andrewid@shark.ics.cs.cmu.edu

Files

Windows & Mac users: Filezilla

andrewid@unix.andrew.cmu.edu

Welcome!

```
$ ls
```

```
$ cd private
```

```
$ mkdir 15-213
```

```
$ cd 15-213
```

```
$ mv ~/Downloads/datalab-handout.tar .
```

```
$ tar xf datalab-handout.tar
```

```
$ cd datalab-handout
```

Some Nice Terminal Shortcuts

- `~/` is your **home directory!**
- `.` is an alias for your **present working directory!**
- Pressing *tab* will **autocomplete** file and folder names!
- `Control+C` will **stop** execution of your current program!
- `Control+R` will let you **search** your command history!
- `Control+L` will **clear** your screen!
- `cmd argument1 ... argumentn > file.txt` will put the output of `cmd` into `file.txt`!
- Use the **up** and **down** arrow keys to **scroll through your command history!**

`ls <dir>`

- Lists the files in the present working directory, or, if specified, `dir`.
- `pwd` tells you your present working directory.

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py  Movies      resume.pdf  test.wav
demo.py           foo2.py      Music       school      timer.py
Desktop           foo.txt      Pictures    solutions.py www
display.py       Fravic.pdf  private     src
Documents        Library     public      Templates
Downloads        Minecraft.jar Public       test.py
jbiggs@blueshark ~ $ pwd
/afs/andrew.cmu.edu/usr10/jbiggs
jbiggs@blueshark ~ $
```

`cd <directory>`

- Stands for **change directory**, or **chdir**
 - Try using `chdir` instead, it actually works!
- Changes your present working directory.

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py  Movies      resume.pdf  test.wav
demo.py           foo2.py      Music       school      timer.py
Desktop          foo.txt      Pictures    solutions.py www
display.py       Fravic.pdf  private     src
Documents        Library     public      Templates
Downloads        Minecraft.jar Public       test.py
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $
```

`mkdir <dirname>`

- Makes a directory `dirname` in your present working directory.
- Directories and folders are the **same thing!**

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py  Movies      resume.pdf  test.wav
demo.py           foo2.py      Music       school      timer.py
Desktop           foo.txt      Pictures    solutions.py www
display.py       Fravic.pdf   private     src
Documents        Library      public      Templates
Downloads        Minecraft.jar Public       test.py
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $ mkdir 15-213
jbiggs@blueshark ~/private $ cd 15-213
jbiggs@blueshark ~/private/15-213 $
```

```
mv <src> <dest>
```

- `cp` works in exactly the same way, but copies instead
 - for copying folders, use `cp -r`
- `dest` can be into an existing folder (preserves name), or a file/folder of a different name
- `src` can be either a file or a folder

```
jbiggs@blueshark ~ $ cd private/  
jbiggs@blueshark ~/private $ mkdir 15-213  
jbiggs@blueshark ~/private $ cd 15-213  
jbiggs@blueshark ~/private/15-213 $ mv ~/Downloads/datalab-handout.  
tar .
```



```
tar <options> <filename>
```

- For full list of options, see `man tar`
- `tar` stands for **t**ape **a**rchive. Was used on tapes!
- `x` - extract, `v` - verbose, `f` - file input
- All of our handouts will be in `tar` format.

```
jbiggs@blueshark ~/private/15-213 $ tar xvf datalab-handout.tar
datalab-handout/
datalab-handout/bits.c
datalab-handout/Makefile
datalab-handout/README
datalab-handout/btest.h
datalab-handout/btest.c
datalab-handout/bits.h
datalab-handout/decl.c
datalab-handout/tests.c
datalab-handout/fshow.c
```

Also, `rm <file1> <file2> ... <filen>`

- To remove an (empty) directory, use `rmdir`
- To remove a folder and its contents, use `rm -rf`
 - **Please be careful, don't delete your project.**
 - **There is no "Trash" here. It's gone.**

What's in a file? (using `cat`)

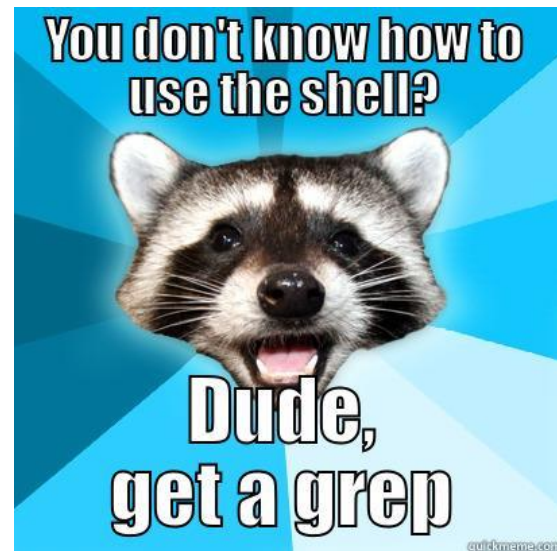
- `cat <file1> <file2> ... <filen>` lets you display the contents of a file in the terminal window.
 - Use `cat -n` to add line numbers!
- You can *combine* multiple files into one!
 - `cat <file1> ... <filen> > file.txt`
- Good for seeing what's in small files.
- Try `cat -n bits.c`. Too big, right?

What's in a file? (using `less`)

- `less <file>` will give you a scrollable interface for viewing large files **without** editing them.
 - To find something, use `/`
 - To view the next occurrence, press `n`
 - To view previous occurrence, press `N`
 - To quit, use `q`
- Try it: Type `"/isPower2"`

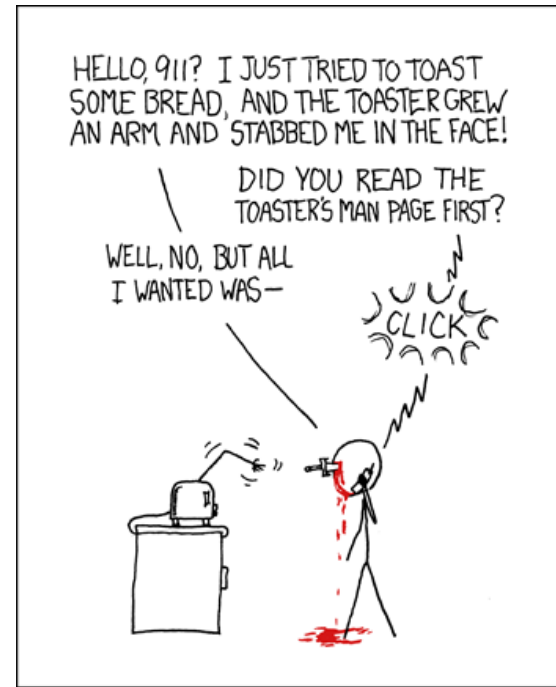
What's in a file? (using `grep`)

- `grep <pattern> <file>` will output any lines of `file` that have `pattern` as a substring
 - `grep -v` will output lines *without* `pattern` as substring
 - `grep -R` will search *recursively*
- Try it: `grep 'isPower2' bits.c`
 - `grep -v '*' bits.c`
 - `grep -R 'unsigned' .`

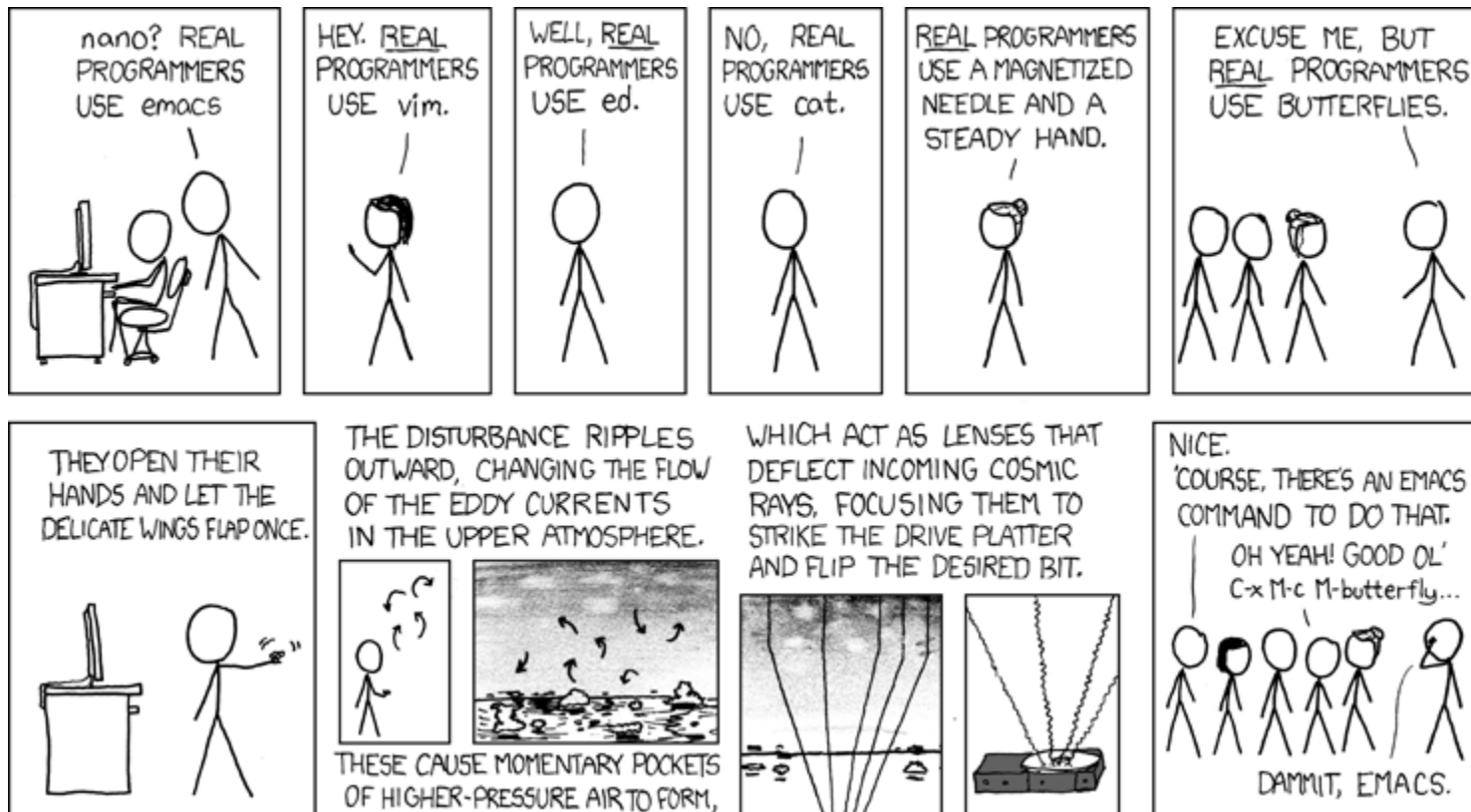


man <thing>

- What is that command? What is this C standard library function? What does this library do? Check to see if it has a man page!
- Pages viewed with `less`
- Try it!
 - `man grep`
 - `man tar`
 - `man printf`
 - `man strlen`



Editors (a touchy subject)



Editors (a touchy subject)

- `vim` is nice, made for very powerful text editing
 - Try running `vimtutor` to get started learning
- `emacs` is nice, made to be more versatile
 - Look online for help!
- `gedit` has a GUI, but requires X Forwarding setup. Too platform-dependent to show here, sadly.
- With enough dedication, the terminal will become your best friend and you will understand each other
 - So just edit on the terminal :)
- **Gist**: Use an editor with auto-indent and line numbers

Editors (if you really really just want a GUI)

- Simple answer: Go to a Linux cluster on-campus, open a terminal, and run:

```
ssh -Y andrewid@shark.ics.cs.cmu.edu
```

- Now you can run `gedit <filename> &`
- *&* *forks* your process into the background so you can use the prompt without waiting for `gedit` to finish

Editors (if you really, **really** just want a GUI)

- Not-so-simple answer: Google “How to install X Forwarding on <platform>”
 - Mac: You need XQuartz
 - Windows: You need Xming and PuTTY
- This allows you to execute GUI applications on the shark machines, but have the GUI appear on your computer.

Commands related to 15-213

- `gdb`, the **GNU Debugger**, will be used for bomb lab.
- `objdump` displays the symbols in an executable.
- `gcc` is the **GNU C Compiler**.
- `make` reads a configuration file to run a series of commands. Often used for compiling your programs.
- We will provide other tools in the handouts as well

Virtualization

- A nice environment for practicing Linux skills outside of the cluster is a virtual machine

- We have created a Red Hat Linux virtual image

`/afs/cs/project/ugradlabs/vm/RHEL6-20140905.img.zip`

- Copy to your computer (6GB), extract, and set up with VirtualBox: <http://virtualbox.org>

- You have full administrative access!
- Break it as much as you want! Experiment!!
- Username is user, password is user

