

Unix on the Mac

Eric Choi & Spencer Bliven

Basics

- Terminal or X11 - friendly vs extra functions
- `man` - help!
- `cd`, `pwd` and `ls` - basic navigation
- `mkdir`, `rm`, `ln -s`

Fink / Fink Commander

- Fink - the mac port tree
- Contains all the standard linux open-source software
- Binaries are fast, but not available for everything

X11

- Used for running graphical unix software
- `.Xdefaults` - customize x11 shell
 - `.bashrc` - run when starting a bash shell
 - `.profile` - only run by Terminal, not X11

Networking

- `Ping IPorDomain` - detect if computers are online
- `ifconfig` - view information about your network status
- `ssh user@dante.u.washington.edu`
 - `ssh -X` forwards X11 from the remote server to your computer
- `sftp user@domain.com`

Scripting

- Output: `echo`, `cat`
- Variables: declared without `$`, used with `$` (weird whitespace issues)
- Quoting: `"$VAR"` evaluates `VAR`, while `'$VAR'` literally returns `$VAR`. ``COMMAND`` evaluates the `COMMAND` inside the backticks
- `$()` evaluates the commands inside the `()` and can be used as a parameter

Scripting - flow control

- `if` statements - are true only if the expression inside returns 0. Use the `[` program to do more advanced conditions
- ```
if [-e file]
then
 echo "File exists."
fi
```

# Scripting - for loops

- `for DIR in `ls -1 /` ; do echo $DIR ; done`

# Regular expressions

- `.` (dot) matches a character
- `[abcd]` matches any of the characters inside the `[]`. Ranges are ok ( `[a-z]` ).
- `[^a-z]` matches anything NOT inside the brackets
- `*`, `+`, `?` matches the previous atom 0+ times, 1+ times, and 0 or 1 times
- `()` Parentheses group together atoms

# Grep

- A filter which takes input and outputs lines which match an expression
  - `grep -options pattern file`
  - `grep 'ft-?mug' index.html`
  - `ls -R | grep 'virus'`

# Grep Options

- `-v` matches lines that do NOT match the expression (inVert)
- `-Cn` prints out n lines before and after the line that was matched (C is for context)
- `-i` case insensitive
- `-n` prints out just the number of times the expression matched

# Sed

- a powerful filter which takes input and uses regexps to modify its output
- `sed -options program infile`
- programs operate on each line.
  - `s` replaces text
    - `'s/spencer/eric/g'` replaces all “spencer” with “eric”
  - `p` prints text, must be used with the `-n` option (no output except where `p` prints)
    - `'/spencer/p'` prints lines which contain “spencer”