CS114: Lecture 2
Permissions, UNIX basics

Have you logged into the CSUG machines?

Any problems?

Questions about HW1?

A note on finding documentation

- Why do I need `man`? Can't I just google things?
- `man` is faster
- `man` is about ***your*** system
- `ls -B`
  - My laptop (OSX; BSD)
    - Force printing of non-printable characters in file names as \xxx
  - CSUG Linux machines
    - Do not list implied entries ending with ~

Funny paths

- Where does `foo/./bar/../baz` point?
- Special characters
  - This list might vary a bit system to system
  - space/tab ! ' “ | & ( ) > < \
  - Need to *quote* them or *escape* them
  - Quote: put between ' ' or “ “
  - Escape: prefix by \

Usage messages

- Two ways to find them: `man`, and “usage” message
- [demo: `perl -h; man -h`]
- You won't always get usage messages.
- To get usage: try `-h –help -help -?`

Passing options

- One option
  - `ls -l`
- Two options
  - `ls -l -d`
  - `ls -ld`
  - `ls -dl`
  - `ls -d -l`
- Not all commands let you collapse options; some options depend on order (e.g. `rm -f -i`)

Other kinds of options

- `man` **`-P`** *`more`* `ls`
- What if my filename begins with -?
  - `--` stops option processing
- “long options”
  - `ls –directory`
    - GNU-style long options begin w/ --; not always

More commands

- `mkdir` *`dirname`* – create directory
- `rmdir` *`dirname`* – remove directory (must be empty)
- `cat` *`filename`* – print contents of file
  - `cat file1 file2 file3` – con*cat*enate files
- `touch` *`filename`* – create empty file
- *`command`* `| less`
  - “page” through output of command
  - `man` does this implicitly
  - h for help, q to exit

Permissions

- `-rwxrwxrwx`
  - Owner
  - Group
  - Everybody else
- First char = d for directories, - for regular files

Permissions: files vs directories

- Files
  - r = read
  - w = write (modify the file)
  - x = execute (run the program)
- Directory
  - r = list files in the directory
  - w = add or delete files from the directory
  - x = cd to the directory
- s = special; for now, think of it like x

Changing permissions: `chmod`

- `chmod` *`who-gets-it`*`(+/-/=)`*`what-permission`*
  - u (user; owner of file)
  - g (group of file)
  - o (other; everybody else)
  - a (all of the above)
- `chmod g+w filename`
- `chmod a-xw file1 file2`
- `chmod -R u=rwx directory`

How commands are run

- `% some-command`
- Find the command
  - Absolute path: problem solved
  - Relative path: check “usual locations” ($PATH)
- Am I allowed to run the command?
  - `........x (o+x)`
  - `.....x... (g+x)` and I'm in the command's group
  - `..x...... (u+x)` and I'm the owner of the command

Some useful keys

- Control-C – stop currently running command

- Control-L – clear screen
- Control-S – pause current printing
- Control-Q – resume