Lecture 8: Working with the shell on AWS

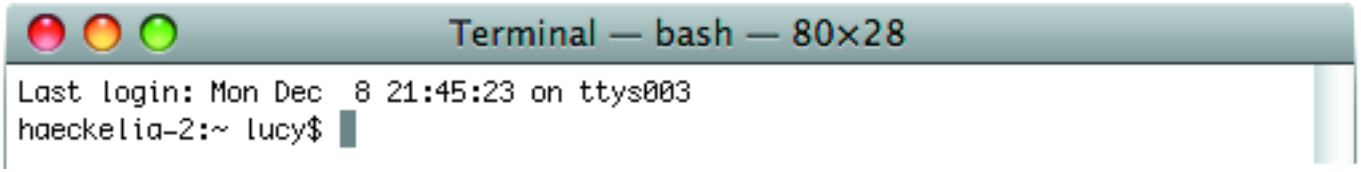
Greg Caporaso

GUI v command line

- Many processes are sequences of operations: these are frustrating in a GUI (they can be scripted from when working on the command line)
- Logging data is generally more difficult when working with a GUI
- GUIs aren't conducive to working on a cluster
- GUI development is labor-intensive and often operating-system-specific

The command line

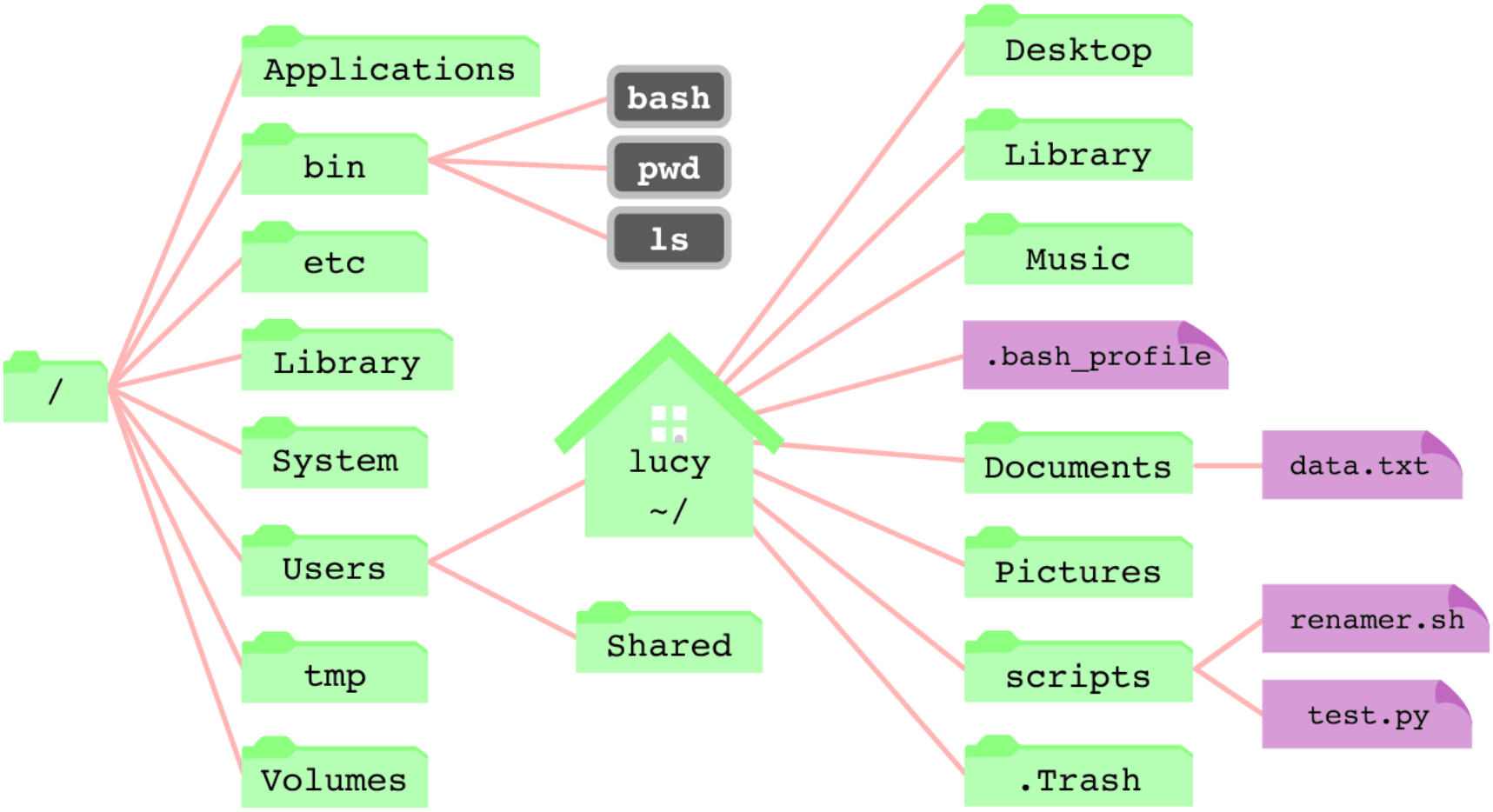
- Your prompt (often $ or >) and cursor
- You're interacting with a program called the *shell* when working on the command line – we'll use the *bash shell.*



The filesystem

- Some important *directories* or locations:
  - The *root* of the filesystem: /
  - Your home directory: ~ or `/home/<username>`
  - Greg's home directory: `~greg` or `/home/<greg>`
  - The current working directory: `.`
  - The directory 'above' this directory: `..`
- Everything on your filesystem lives under the root, including your hard drive, external (e.g., USB) drives, etc.

The Mac OS X filesystem



Absolute versus relative paths

- Absolute path: complete and unambiguous description of a location on the filesystem in relation to the root of the filesystem
  - Hint: these start with a /
- Relative path: location on the filesystem relative to some other location (e.g., the current working directory)
  - Hint: these don't start with a /

Navigating the filesystem

- Change directory: `cd`
- List files and directories: ls
- Print the current directory: pwd
- Create a directory: mkdir
- Remove a directory: rmdir
- Copy files: cp
- Move files: mv

The ever-important shortcuts

- Up and down arrows: navigate through your command history
- Left and right arrows: navigate through your command (and hit Enter at any point to execute it – you don't need to be at the end of the line)
- Tab: complete a file name
  - Works when the text you've entered so far is unique
  - If you only get a partial list, press tab again to see possible completions, and then expand your text to make it unique.

Modifying the behavior of commands with command line arguments

- Most commands take *arguments* (or *parameters* or *options*)
- For example:
  - ls -a
  - ls -l
  - ls /
  - ls –al ~

less: view files in read-only mode

- Navigate with up and down arrows
- Search for hello by typing: / hello
- Exit by typing: q
- Go to beginning with: g
- Got to end with: G

man: print the *manual page* for a command

- man ls
- man uses *less* to display the manual, so you can navigate as you did previous
- (Some command line tools, like QIIME, implement a –h or --help option instead of using man pages.)