Basic Unix Operations on Files and Directories

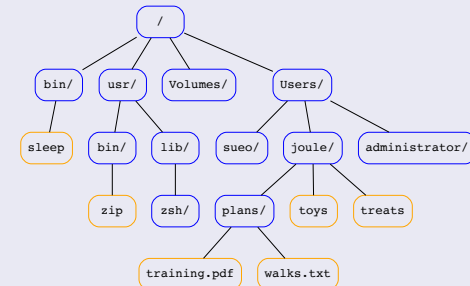Prof. Susan Older

26 August 2019

Unix Beginnings (1969 at Bell Labs)

Design philosophy:

- View **everything** as a file (but allow different types of files).
- Incorporate lots of small commands that can be combined easily to accomplish more complex tasks.
- Employ a hierarchical file system, which forms a tree structure:

Note: / is called the root directory.

Current Working Directory and Pathnames

When you log in (or start a Terminal window), your current working directory is your home directory:

- /Users/joule is my dog's home directory.
- ~ is the abbreviation for one's home directory.
- . is the abbreviation for the current working directory.
- .. is the abbreviation for directory directly above current directory.

Absolute pathnames start with / (i.e., they start at root directory):
/Users/joule/plans/training.pdf or /bin/sleep or /usr

Relative pathnames start relative to current working directory.
If current working directory is /Users/joule, then:

- plans/walks.txt refers to /Users/joule/plans/walks.txt
- ../sueo refers to /Users/sueo,   ../../bin refers to /bin

Moving Around the System

- cd
  Make your home directory (~) the current working directory
- cd *pathname*
  Make *pathname* the current working directory
- pwd
  Print the name of the current working directory

To find out more about specific commands:

∗ man *command*
  Get a manual page on *command*

Viewing Files and Directories

- `ls`
  List the contents of current directory, but don't show hidden files (i.e., files whose names start with a dot/period (`.`))
- `ls pathname`
  List the contents of the directory at *pathname*

Additional options:

- `ls -a`
  List **all files**, including hidden files
- `ls -l`
  Generate a **long listing**, which includes information about permissions, file size, etc.
- Options can be combined, such as:

  `ls -al ~/class_stuff`

Working with Files

Basic commands:

- `cp source target`
  Make a copy of file *source* and name the copy *target*
- `mv source target`
  Move (or rename) the file *source* to *target*
- `rm source`
  Remove the file *source*

Generalizations:

- $*$ `cp source`$_1$ `...` `source`$_k$ `targetDirectory`
  Make a copy of each of the files *source*$_i$ and place them in the directory *targetDirectory*
- $*$ `mv source`$_1$ `...` `source`$_k$ `targetDirectory`
  Move each of the files *source*$_i$ and place them in the directory *targetDirectory*

Some More Commands

Working with directories:

- `mkdir target`
  Make a new directory called *target*
- `rmdir target`
  Remove the directory *target*

Working with text:

- `more name`
  Page through text file *name*, one screenful at a time
- `less name`
  A more versatile (and more efficient) version of `more`
- `file name`
  Find out what type of file *name* is