Intro to Unix/Linux

Emily Josephs & Nancy Chen

March 16, 2016

This is simply a brief introduction to Unix/Linux and a bare bones list of useful commands. If you want to learn more, there are many more thorough/advanced guides online.
Typeset conventions: All code is written in `constant-width font`. Keyboard commands are in **bold**.

The command line

The command line is a user interface where you interact with your computer by typing in commands at a prompt. It is accessed via the program Terminal, often found under Utilities or Accessories.

The command line prompt provides information about the computer, the user, and the current directory. The order in which this information is presented may vary, but `$` always means the shell is ready for input:

- Linux: `username@computername directory$`
  Ex: `nancy@ClarkLabLinux:Desktop$`)
- Unix: `computername:directory username$`
  Ex: `NancysLaptop:Desktop nancychen$`

All commands have this basic structure:

```
command [options] [arguments]
```

- options modify the action of the command and are usually preceded by a dash
- arguments provide additional info
- Ex: `ls -l Desktop`

Some tips for using the command line:

- use **copy/paste**
- use **up/down arrow** keys to cycle through commands
- use **TAB** to complete commands or present a list of choices
- avoid spaces in file names. Use underscore instead. Spaces are special characters and require a backslash: “EEB Lunch Bunch” is denoted “`EEB\ Lunch\ Bunch`”
- use wildcard characters (`*` = match anything). `*.txt` refers to all files ending in `.txt`
- **Control + C** will stop a running command and return to an active prompt

Directory structure

Unix has a hierarchical way of organizing files.

When you log in, you are typically in your home directory (often denoted as `~`).

A single period (`.`) refers to the current directory.

You can refer to files two ways:

1. absolute/full path:
   `/Users/nancychen/Desktop/EEB_LunchBunch/IntroToLinux.txt` or
   `~/Desktop/EEB_LunchBunch/IntroToLinux.txt`
2. relative path (depends on your current directory):
   `EEB_LunchBunch/IntroToLinux.txt` (if current directory is `/Users/nancychen/Desktop`)

A LIST OF USEFUL COMMANDS

`man commandName`: manual page for a command. All commands have a help page.

- Scroll with **space** or **up/down arrows**.
- Press **q** to quit.

`history X`: prints the past `X` commands.

Working with directories

`pwd`: (print working directory) show the current directory

`cd`: change directory (can use absolute or relative locations)

- `cd`: go to home directory
- `cd ..`: move one level up
- `cd ../..`: move two levels up
- `cd /`: go to the root directory
- `cd -`: move back to directory you just left

`mkdir newDir`: make new directory `newDir` in current directory

`rmdir dirToDelete`: remove directory `dirToDelete`. Only works if it is empty.

`ls myDir`: list the contents of directory `myDir`. If no directory is specified, lists contents of the current directory. Some useful options (see `man ls` for the full list):

`-l` : include info on permissions, ownership, size, and last modified date
`-a` : list hidden files & folders as well
`-h` : list in human readable form
`-alt` : list in order of modification time (`-altr` to reverse order)
`-alS` : list in order of size (`-alSr` to reverse order)

Working with files

`cp`: copy files

- `cp oldFile newFile`: copy file `oldFile` to file `newFile` in the same directory
- `cp test.txt /workdir/nancy/test1.txt`: copy `test.txt` from current directory to a file called `test1.txt` in directory `/workdir/nancy`
- `cp /workdir/nancy/*.txt .`: copy all files ending with `.txt` in `/workdir/nancy` to current directory

`mv`: move files

- `mv oldFile newFile`: move file `oldFile` to file `newFile`. Use this to rename files.
- `mv fileName newDir`: move file `fileName` to directory `newDir` without renaming the file.

`rm fileName`: delete file (really delete, not just move it to Trash)

- use option `-i` to ask for confirmation (a good idea)

`less filename`: view a file

- you can search within the file (forwards from your cursor position) by typing / followed by the desired text
- type **q** to exit.

`head -X filename`: view first `X` lines in file (default is to show first 10 lines)

`tail -X filename`: view last `X` lines in file (default is to show first 10 lines)

`cat filename`: print file on screen.

- can be used to combine files: `cat file1 file2` > `file3`.
- > redirects output into `file3`.

`wc filename(s)`: print the number of lines, words, characters in `filename(s)`

- `wc -l filename`: print only the number of lines

`grep "text" fileName(s)`: search for string `"text"` in the specified file(s)

- option `-c` counts the number of lines that contain string `"text"`

Editing files

There are many command line text editors. The most common ones are:

- `nano filename`: open `filename` for editing. If no file is specified, it opens a new file. Already installed on most computers.
  - commands displayed in bottom bar
  - **Control + X**: exit program
  - **Control + O**: save changes
- `vi`: requires special keyboard commands. Installed on all Unix/Linux computers.
- `emacs`: not always included. You might need to install it yourself.

You can always copy and paste from your favorite text editor on your local computer.