A quick guide on becoming comfortable with using Terminal

CS 10A – UNIX TERMINAL CRIB NOTES

Allen Zhao - SRJC CS 10A

Introduction

- Terminal is just another way of navigating through your computer in the most old fashioned way possible.
- Every action is performed with a command.
- Most commands can be tacked on with additional options known as flags. These are preceded with a hyphen.
- Only the most commonly used commands will be needed for this class. Commands will be highlighted in yellow.
- Some shortcuts can make Terminal more convenient and faster even when compared to modern UIs.

Navigation

- cd: Change Directories. Use this to change folders.
  - cd folderName : go to folderName
  - cd .. : go one level up in the folder hierarchy
  - cd - : back button. Goes back to wherever you were previously
  - cd ~ : go home. Goes back to your specified home directory
- ls: List files. Use this to list all files and folders in the current location.
  - ls –a : list all files and folders, including hidden ones
- pwd: Print Working Directory. Use this to show the current file path of your location.
- clear: Clear the console. Use this when you feel that your console log has become too cluttered.

Creating and Deleting

- vim: Opens up the Vim editor.
  - vim fileName : opens up fileName. If fileName doesn’t exist, then a new file with filename will be created
- mv: Move. Basically cut and paste a file elsewhere. Also acts as the rename file command.
  - mv fileName newFile : renames the file (or not). List out the whole path destination in newFile in order to specify where to move the file to
- cp: Copy. Copy and paste a file elsewhere.
  - cp filename newFile : copies the file. You can also specify where to copy it to by listing out the whole path destination in newFile
- mkdir: Make Directory. Use this to create a new folder.
  - mkdir folderName : creates a new folder with folderName
- rm: Remove. Use this to delete files and folders.
  - rm fileName : deletes a file
  - rm –i fileName : deletes a file but asks for confirmation before deleting.
  - rm –r folderName : deletes a folder, but you’ll be asked to confirm the deletion of each item in the folder
  - rm –rf folderName : deletes a folder and automatically deletes everything in it

Programming

- gcc: Use the gcc compiler. For C programs (*.c). Output file by default will be named ‘a’.
  - gcc fileName.c : compile fileName.c
  - gcc fileName.c –o outputName : compiles fileName.c. The compiled output will have outputName instead of ‘a’.
- g++: Use the g++ compiler. For C++ programs (*.cpp).
  - g++ fileName.cpp : compile fileName.c
  - g++ fileName.cpp –o outputName : compiles fileName.c. The compiled output will have outputName instead of the default ‘a’.
- ./ : run an executable file. *.exe for Windows. *.out for MacOS and Linux. There is no space between this command and the file.
  - ./fileName.exe : run the compiled file (for Windows)
  - ./fileName.out : run the compiled file (for MacOS and Linux)

Useful Shortcuts

- The Tab key is an auto-fill feature. You can use this to auto-fill file and folder names. Tapping it twice when there are multiple possible fill options will show a list of all file and folder names that fit what you typed in so far.
- The Up Arrow and Down Arrow keys can be used to cycle through your command history. Up goes further into the past. Down goes toward the present.
- Home and End allows you to automatically go to the front or end of the command, respectively. Useful for very long commands.
- Ctrl + C. This is the abort key. Use this to automatically terminate any program or command. Useful for stopping infinite loops.