CS BOOTCAMP

Introduction to Linux and Code Editing

Some quick things

- If you have used Linux and worked with it remotely (Ubuntu, Debian, RedHat, etc.) this presentation isn't meant for you.
- Please stop me to ask questions.
- Use your resources. This will be recorded, and the slides/materials will be publicly posted for your viewing pleasure.

- Materials - https://csit.kutztown.edu/UNIXbootcamp/index.html
- Student Resources - Maintained by the CSIT Department

Linux

- Linux is another type of operating system, just like Windows or MacOS.
- Command Line Overview
- Linux and Kutztown
  - The CSIT faculty and students have access to multiple servers
  - The two main ones:
    csitrd - (a.k.a. acad, csit, login, unixweb) &

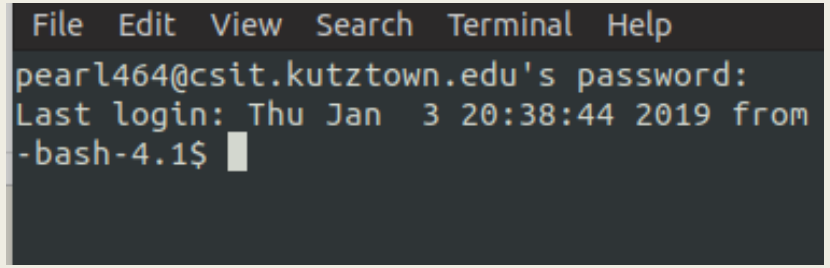
    Mcgonagall (a.k.a. Kupapcsit01)

Connecting to CSIT

- **Windows – Additional Setup Required**
  - Personal Laptop
    - Follow link to install PuTTY in materials
    - Profile Setup Walkthrough
  - Lab Computers
    - Profiles already setup, open PuTTY and double click "acad no x11"
- **MacOS / Linux**
  - Just need to use the terminal
  - ssh *username@host*
    - *I.e. ssh pearl464@csit.kutztown.edu*

After Login

- Scenario One:



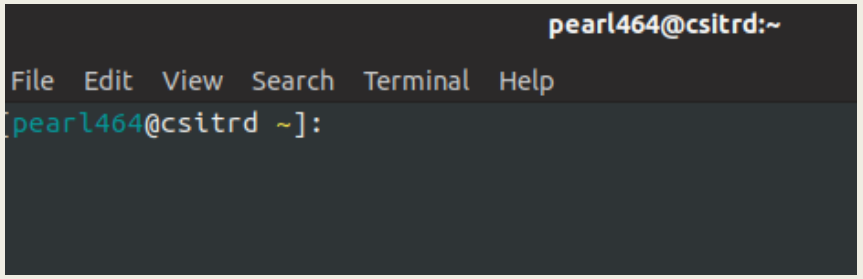
- Enter these commands

  wget http://csit.kutztown.edu/~pearl464/bootcamp/.bashrc

  source .bashrc

  - Your bash should be properly setup now

- Scenario Two:



- If your prompt looks like this, you're golden :-)

Commands Overview

- Some terminology:
  - Directory (a folder) – Just like your folders in Mac's Finder or Window's File Explorer. Stores multiple files
  - "~" - Your home directory

- Structure of command:
  - ls [space] -al [space] ~
  - command [options] [file]

Common Commands

- pwd
  - "Print working directory"
  - Current directory that your bash prompt is at
- ls
  - "List directory contents"
  - Shows files and directories within the given directory. Defaults to the current working directory
  - Common options:
    - -a – Lists all files and directories (Even hidden if you have permission)
    - -l – Formats the output as a list
- mkdir
  - Creates a new directory
  - Example: mkdir ~/csc135
    - Will make a new directory called "csc135" located in your home

Command Commands (cont.)

- rm
  - Remove file
  - Options:
    - -r – Recursive. Used to remove multipe files
    - -f – Force removal of a file
  - NOTE: Be careful using the –f option. Once a file is deleted, there is a slim chance it can be recovered.
  - Example: rm test.txt
- man
  - Provides the manual page for a given command. Very useful if you need to understand how a command works and what the options for it are.

G++

- GNU Compiler
- Basic format: g++ <filename>
  - Results in a.out
- G++ has many options and flags built in, two important ones are shown below. You can find more information about them and the different flags in the *man* page.
- -o allows you to name your compiled program:
  - Example: g++ p1.cpp -o p1 will give you an executable called p1
- -g to enable debugging
  - This will keep function/variable names with the executable allowing you to debug your program (Don't worry if you don't know all these terms yet)

File Transfer – Work with files locally

- FileZilla Client – MacOS and Windows - https://filezilla-project.org/
  - https://wiki.filezilla-project.org/FileZilla_Client_Tutorial_(en) Tutorial (For Later Reference)
- WinSCP – Windows Only - https://winscp.net/eng/download.php
  - https://winscp.net/eng/docs/start - Documentation (For Later Reference)

Picking a Text Editor

- Pick the right tool for your job
- Make things better for yourself:
  - ***Learn keyboard shortcuts***
  - Plugins
  - Themes
- Make your editor part of your workflow
- Shortcuts speed things up
- Command Line vs GUI

Text Editing – Command Line

emacs

- Most of the faculty use it
- Flexible and powerful – An IDE
- Exiting: Cntl-X Cntl-C

vim

- Quick and fast
- Editor
- Exiting: <ESC> :q

More Resources:
Emacs Guide Tutorial:
https://www.gnu.org/software/emacs/tour/
Interactive Vim Tutorial: https://www.openvim.com/

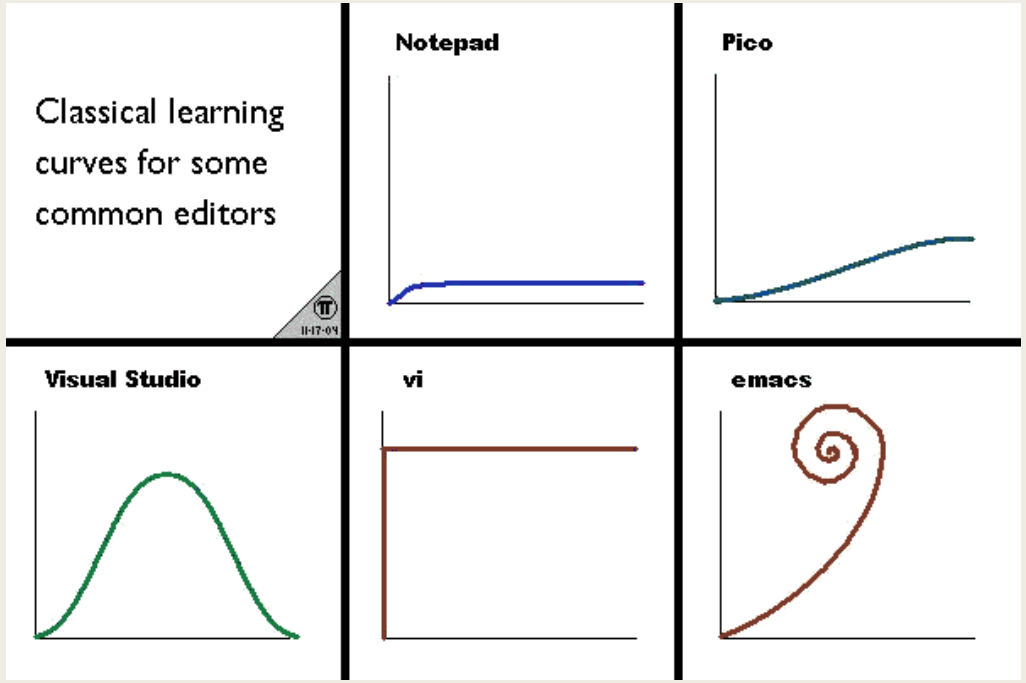


Image Source: https://unix.stackexchange.com/questions/986/what-are-the-pros-and-cons-of-vim-and-emacs/988#988

Handy Keyboard Shortcuts

- Select next instance of word (multi-cursor):
  - Sublime, VSCode: CTRL+D
- Duplicate Line:
  - Sublime: CTRL+Shift+D
  - VSCode: Shift+Alt+Down or Shift+Alt+Up
  - Notepad++: CTRL+D
  - Emacs: `C-a C-SPACE C-n M-w C-y`

Emacs Commands:

- *C (short for CTRL)*
- *M (short for Meta, either Alt or ESC)*

- C-a – move cursor to start of line
- C-Space – sets a mark to begin selection ("start highlighting")
- C-n – move cursor to next line
- M-w - copy highlighted region
- C-y – paste (stands for "yank")
- C-k – cut ("kill") the rest of the line

The point is to learn ways of speeding up repetitive tasks!

Notepad++

Pros

- Free forever
- Fast/lightweight
- FTP Capabilities

Cons

- No Git support
- Windows only
- Bad plugin support
- Dated UI

VSCode

Pros

- Free/Open Source
- Great plugin options
- Built-in terminal
- Git support
- Modern and customizable UI
- Windows, macOS, Linux

Cons

- No default FTP (3rd party plugins available)
- Built on Electron, a RAM hog