COP 3363

SPRING 2020

RECITATION 1

UNIX BASED OPERATING SYSTEMS

- Why are we learning this and why is the OS important?
  - Has been in use since the 1970s
  - Many versions (flavors) available running on a wide variety of machines world wide
    - Raspbian - Linux derivative designed for RaspberryPi
    - ~98% of all publicly accessible servers on the Internet use a Unix or Unix-like OS
    - As of 2017 the 500 fastest supercomputers in the world all run some version of Linux

THE COMMAND LINE INTERFACE (CLI)

- Again, why are we learning this and why is the CLI important?
  - CLIs have existed since the 1960s
  - Widely used in a non personal-computing context
  - Can perform actions on a machine which a GUI cannot
  - Nerd cred

LINPROG

- A collection of servers running Linux
  - Used by faculty and students to compile and run code
  - Use a command line application and the Secure Shell protocol to connect to the server
    - macOS | Linux - Terminal application
    - Windows - Tectia
    - ssh <cs username>@linprog.cs.fsu.edu
      - ex. ssh mcinnest@linprog.cs.fsu.edu

BASIC COMMANDS

- Interact with the machine by issuing it commands
  - Commands are typically accompanied by flags and sometimes with a string or input file as an argument
    - (Very basic) Format: <command> -<flags>
  - Unsure of how to use a command?
    - man - displays the documentation for a command and associated flags
      - ex. man ls displays the documentation for the ls command

LS

- Command which lists the contents of a directory (aka folder)
  - Useful flags
    - a - list the contents of a directory, including hidden files and hidden subdirectories
    - l - display a detailed listing of the directory contents
    - ex. ls -la
      - detailed list of all files and directories in the current directory

CD

- Command which changes your current working directory
  - ex. cd test/
    - Would make the current working directory test, assuming it is a directory within the current directory
  - cd ~
    - Changes your current working directory to your home directory

NANO

- One of multiple text editors which run in a CLI environment
- nano <filename>
  - opens the file <filename> in the nano editor
- nano
  - opens nano and a new temporary file which can be saved later
- Other editors exist (all have benefits and drawbacks)
  - Vi/Vim, Pico, Emacs

OTHER USEFUL UNIX COMMANDS

- touch <filename>
  - creates an empty file with name <filename> within the current directory, if the file does not exist
  - updates the files last modified timestamp if it does exist
- rm <filename>
  - remove the file with name <filename> in the current directory
- rmdir <directory name>
  - remove the directory with name <directory name> in the current directory
- mkdir <directory name>
  - create a directory named <directory name> in the current directory
- cp <filename1> <filename2>
  - copies the file <filename1> and names the copy <filename2>
- mv <filename1> <filename2>
  - renames the file <filename1> to <filename2>
  - if a path is included before <filename2> the new file will be moved to that directory

USEFUL UNIX FEATURES

- Tab completion
- Up/down arrows to view previous commands
- ! character followed by 1 or more characters finds the most recent command starting with those characters
  - ex. touch testfile
    - !to + <tab> or <enter/return> displays/runs touch testfile again

FILE TRANSFER

- How do I move files/folders to and from a server?
  - SFTP - Secure File Transfer Protocol
    - available from the command line
      - sftp <username>@<server address>
        - ex. sftp mcinnest@linprog.cs.fsu.edu
    - Use an SFTP GUI application such as FileZilla

CODING STYLE

- Practice using a clean and readable coding style
    - makes your code easier to debug
    - in the industry, makes the code you write more maintainable
        - you will typically be working in a team of programmers
- Utilize indentation to denote blocks of code
    - useful with loops and control flow statements (we will learn about these concepts later)

COMMENTS

- Learning to document (comment) code in a clean and useful manner is critical to becoming an effective programmer
- Comments help you (and more importantly) others understand what your program is doing at a given point
- The software you will work on after school will often be thousands or even millions of lines long
  - Microsoft Windows code base is comprised of 50+ million lines of code
  - As a professional, your time is valuable. A good comment can save hours of time trying to understand a code block.

VARIABLE NAMES

- Should be descriptive, but not too lengthy
  - try to name variables in a way which would make then understandable at first glance
  - ex. int x; vs int taxRate;
- Good naming styles: taxRate, tax_rate
- Const variables are often all uppercase
  - ex. const int SIZE = 10;
- Variable declaration should be done at the top of main()
  - makes it easier to identify all of the variables used in a program, along with their initial values

G++

- Open source C++ code compiler
  - converts the code you write into a useable program
- Useful flags
  - -std=c++<version> compiles using a specific version of C++
  - -o <filename> specify the name of the executable file the compiler creates
  - -Wall display a detailed list of compile warnings and errors
- ex. g++ hello.cpp -o firstProg

REFERENCES

- https://en.wikipedia.org/wiki/Usage_share_of_operating_systems
- https://en.wikipedia.org/wiki/Supercomputer_operating_systems
- https://en.wikipedia.org/wiki/Command-line_interface

QUESTIONS?