Lab 1: Introduction to Linux

UWYO COSC 2030

1 Introduction to the Labs

Welcome to Computer Science 2030. During this course you will learn fundamentals of linux, git, as well as an overview of algorithms. For most labs you will have a week to do them, they will be due by the following Sunday at 11:59pm. Labs will be turned in via github. Some labs will require use of department raspberry pi's to demonstrate the need for effective coding due to their lower computational power. The week of each mid-semester test and the week before the final we will have challenge labs. These labs will be harder coding challenges, but will be optional and successful completion will award you up to 5 extra credit points, depending on the lab.

2 Expectations:

- All non-challenge labs must be turned in on github
- Repos MUST include a README.md file with your name and lab section
- Repos lacking a readme will have points removed
- Labs will be due the following Sunday, labs will not be accepted after 3 days late unless you have contacted me beforehand
- Lab work must be independently authored. You may work together but you must do your own work for submission
- Challenge labs are timed and can only be completed during lab time. Late submissions will not be accepted and attendance is mandatory if you wish to submit a challenge lab

3 Lab: Introduction to Linux

For this course you will be learning how to use Linux for coding, especially the use of a terminal. Here are some basic Linux commands. If you ever have a question, [man "command"] is your friend (man is short for manual). Google or [insert your favorite search engine here] are great tools as well.

- ssh- secure shell. This is used for connecting over command line to another computer. This will be used whenever when you connect to one of the pi's
- ls- list. This lists the documents and folders in your current directory
- pwd - print working directory. This prints the path to your current directory
- cd - change Directory. Used to navigate the file system e.g cd folder or cd .. to travel back up
- g++ - the gnu C++ compiler. g++ example.cpp
- touch- makes a file with the supplied name use: touch example.cpp
- nano - opens the a text editor for use. nano example.cpp, How you will make and edit documents. You can also use vi rather than nano
- vi- another text editor, similar use to nano
- ./ - running your program. Use g++ example.cpp -o program to make the executable ./program to run. the -o signifies what g++ should name the executable, a.out is default

- mkdir- Make directory. This is how you may make a new folder. Used with mkdir folderName
- rm- remove a file. Used with rm fileName

3.1 Example C++ file creation and execution

- touch example.cpp (This step can be skipped though)
- nano example.cpp
- g++ example.cpp -o exampleOut
- ./exampleOut

4 Lab: Assignment

You will begin to learn the basics of Linux following these steps:

- Step 0. Open the terminal you set up in lab 0
- Step 1. Connect to the department Linux machines using ssh, once you have logged in use passwd to change your password. The initial password will be: changeme
- Step 2. Use ls and pwd commands to see the results
- Step 3. Make a folder called 2030_Labs to make a folder for these labs
- Step 4. Move your current working directory into your new 2030_Labs folder, travel back up to your home folder, and return to the 2030_Labs folder
- Step 5. Make a new .cpp file named lab1
- Step 6. In nano/vi write Hello World
- Step 7. Save and exit your text editor
- Step 8. Compile your program, set the output to be labOne
- Step 9. Run your program, address errors as needed

5 Lab: Submission

There is nothing to turn in for this lab.