

Lab 1: Introduction to Linux

UWYO COSC 2030

1 TA Contact Information

- Email: mwolff3@uwyo.edu
- Office Hours: 11-12 Wednesdays & Fridays, EN 4072

2 Introduction to the Labs

Welcome to Data Structures. During this course, you will learn fundamentals of data structures and algorithms, as well as the practical application thereof. The primary function of the labs is to provide you with an opportunity to learn these practical applications. Most labs will be assigned during Tuesday lab sections and will be due the following Sunday at 23:59 MT. Labs will be submitted via Github Classroom. The week prior to exams, optional challenge labs will be offered. Successful completion will award you up to 5 extra credit points on the exam.

3 Expectations

- Collaboration is allowed and encouraged, but lab work must be authored independently. Please submit your own work and note if you worked together with anyone else.
- Use of generative AI is strongly discouraged. You are expected to learn the fundamentals of data structures and algorithms in this course, a foundation you will need to refer to for the remainder of your time in the field of computer science. Building a strong foundation without using unreliable sources is recommended. If you do use generative AI, please disclose all prompts, responses, models used, and any code generated in your README.md file.
- Repositories must include a README.md file with your name, lab section, and any external sources consulted.
- Late labs will be accepted for full credit only if you contact me at least 24 hours prior to the deadline to arrange a new submission date, or in the case of an emergency. Otherwise, labs will be penalized 10% for each day late, or 5% per day if you contact me less than 24 hours before the deadline. The same policy applies if you need an additional extension beyond the new submission date that you have arranged with me. (I am generally willing to make any reasonable accommodations if you communicate with me in advance.)
- Challenge labs are timed and can only be completed during lab sections. Whatever you accomplish in that time will earn extra credit towards the exam. Late submissions will not be accepted and lab attendance is mandatory if you wish to submit a challenge lab.

4 Lab 1 Instructions

4.1 Introduction to Linux

For this lab, you will learn the basics of using the Linux terminal. Some frequently used commands are listed below. For additional information, `man "command"` provides an overview of a given command (man is short for manual). Additional commands can be searched online.

- `ssh` - secure shell. Used for connecting over command line to another computer. This command will be used whenever when you connect to one of the Raspberry Pis

- `ls` - list. Lists the files and folders in your current directory
- `pwd` - print working directory. 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 the directory structure
- `mv` - move a file from one directory to another. Example: `mv sourceDirectory destinationDirectory`
- `g++` - the gnu C++ compiler. Example: `g++ example.cpp`
- `touch` - makes a file with the supplied name use. Example: `touch example.cpp`
- `nano` - opens the text editor for use, which enables you to create and edit files. Example: `nano example.cpp`
- `vi` - another text editor, similar to nano. Example: `vi example.cpp`
- `./` - runs a program. Use `g++ example.cpp -o program` to make the executable `./program` to run. `-o` signifies what `g++` should name the executable. `a.out` is default if you do not specify the name of the executable
- `mkdir` - make directory. This is how you make a new folder. Example: `mkdir folderName`
- `rm` - remove a file. Example: `rm fileName`

4.2 Example C++ file creation and execution

- `touch example.cpp` //creates a C++ file called `example.cpp` (optional)
- `nano example.cpp` //opens nano as the text editor to add code to `example.cpp`; creates a file if one has not already been created
- `g++ example.cpp -o exampleOut` //compiles `example.cpp` to the executable `exampleOut`
- `./exampleOut` //runs the program `exampleOut`

4.3 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` or `vi` write a Hello World program
- Step 7. Save and exit your text editor
- Step 8. Compile your program, set the output to be `labOne`
- Step 9. Run your program and address errors as needed

4.4 Submission

There are no deliverables to submit for this lab.