CISC 3110

Lab 0: Introduction to Unix/Linux

Objectives:

- To gain experience using the Linux command line (`bash` shell)
- To gain experience using a common Unix editor (`vim` or `emacs`)
- To gain experience compiling and executing a C++ program in the Linux environment

Part I: Command Line Basics

In this section, I'll ask you to do some simple tasks using the command line. You may have to consult today's readings to find appropriate commands.

In Linux, open a Terminal window, if one isn't yet open.https://bc-cisc3110-s17.github.io/lab0/gcd.cc

- What is your current working directory?
- What files are in your current working directory?
- What is the largest file in your current working directory?
- Make your working directory the "root" directory—the one identified by `/`
- What files and subdirectories are in the root directory?

I recommend that you put all your files for this class in the `/sync` directory (so that, for example, you can print them easily). Make a subdirectory `/sync/lab1` that you'll use for this lab.

Part II: Editor Basics

Make `/sync/lab0` your working directory. Use one of the editors `vim` or `emacs` to create a file called `typingtest.txt.` Put the text

```
The quick brown fox jumped over the lazy dog.
```

in this file, make sure it's saved, and exit the editor.

From the command line, view the contents of your newly created file.

Download the file into your `/sync/lab0` directory. Modify the code so that the first line of output is "GCD, modified by <your name>".

Part III: Compiler Basics

Compile the GCD program. When compilation is successful, look at the contents of your working directory. What new files were created, if any?

Run the GCD program. If you get a "command not found" error message, check the readings.

Change the compiler's behavior so that the command to run the gcd program is `gcd`.

Part IV: A Simple Program, Start to Finish

Write a C++ program that reads up to 10 integers from the keyboard, stores them in an array, calculates the average, then outputs the list of numbers and their average. Calculate the average by calling a function called a function you write named `array_average` that returns the average of an array of integers (what signature should this function have?). Make sure your name is in a comment at the beginning of the file.

Once this compiles and runs successfully, email your code to me (subject: CISC 3110 Lab 0). I must receive this by 11:59pm, Tuesday, February 6.