Lab 1 - Description

(Setting up your development environment)

Lab Overview:

For this lab, we will be setting up our Linux development environment for systems programming in C. Make sure to follow all steps outlined in the "tasks" subsection of this lab description. The tasks outlined in this lab description also include tasks from assignment 0. Be sure to ask questions if anything is not immediately clear.

Core Tasks:

1. Set up your virtual machine (VM).
2. Set up a Github Private Repo.
3. Installing an IDE/Text editor.
4. Install development tools (GDB, Valgrind)
5. Hello World in C. (Developing, Compiling and Running code)
6. Debugging and Developmental testing.

Task Details:

1. Set up your virtual machine (VM).
    a. Open VirtualBox and click the "**New**" button.
    b. Enter "cis415" as the name and select "**use an existing virtual hard disk file**", then click "**create**".
    c. Next, click the start button to start your VM.
2. Set up a Github Private Repo.
    a. First, log in or create a GitHub account. (https://github.com/)
    b. Click the "**New**" button under repositories to create a new repo.
    c. Name the repo "CIS415" and select the "**private**" option.
3. Installing an IDE/Text editor.
    a. Choose and install an IDE/text editor that fits you. (We use Sublime and vim)
4. Install development tools (GDB, Valgrind)

    a. Make sure GDB works type: "**gdb**". (You should already have it, if not then speak to one of us (Jared or Monil)
        i. This is a super helpful tutorial for GDB: https://www.geeksforgeeks.org/gdb-step-by-step-introduction/
    b. Make sure Valgrind works, type: "**valgrind**".
        i. This is a helpful tutorial on how to use Valgrind: https://www.cprogramming.com/debugging/valgrind.html
5. Simple programming in C. (Developing, Compiling and Running code)
    a. Write a simple program in C echo's what the user enters into the command line.
    b. Compile and run it with gcc. Demo it when you are finished with the lab.
6. Debugging and Developmental testing. (See the test code provided)
    a. This code does not work. You will need to debug and fix the code using GDB and Valgrind.
    b. Set breakpoints in the test code's main function and step through the code until you find the bugs.
    c. Demo the GDB run and Valgrind call.

Submission Requirements:

In order to receive any credit for a lab, completion of the labs' core tasks must be demonstrated to the TA's. (i.e. just show us a small GDB run through the fixed test code and your Valgrind output showing your code has no memory leaks, along with your echo program)