CS 277 Lab 1
Warm Up: Unix command line, *Vim* & Compilation with *GCC*
Due 11:59pm PST Sunday February 2, 2014

Purpose

The purpose of this lab is to give you some practice with some fundamental tools for software development: the Unix command line, the *Vim* (aka"vi") command line text editor and gcc, the GNU compiler collection which transforms your beautiful C code into the cold, hard binaries that a computer can understand. You are not required to use these tools for this class, however, you are responsible for knowing how to use them and many lab assignments will be facilitated by the basic skills covered in this assignment. Along the way, you will also have the chance to re-familiarize yourself with a little C syntax before we jump into a full programming assignment for Lab 2. At the end of this assignment, you should be comfortable navigating through the Unix file system via the command-line, using vi to create and edit text files and compiling C source files with GCC.

Assignment – Part 1

Linux is a Unix-like operating system and in this part of the assignment we'll review a few basic commands for navigating through the file system via the command line. You will find these same commands on any Unix based system including Mac OS X.

Step 1: First things first: if you are using a windowing environment, open a terminal.

Step 2: Where are we? In the terminal window, type **pwd** and press enter. This command stands for "print working directory" and will print the *path* to the location in the file system where you have navigated your terminal session. The path to your working directory should be something like */home/users/yourusernamehere*. This path location is known as your *home directory*. Unless configured otherwise, every user on a Unix system has his/her own unique home directory.

Step 3: What is here? Type **ls** and press enter. This command stands for "list" and will list the contents of the current working directory.

Step 4: Creation. Lets create a directory. From your home directory type **mkdir cs277** and press enter. Now type **ls** and you should see the new directory. Now type **mkdir cs277/lab1** and press enter. This will create a subdirectory called *lab1* in the *cs*277 directory.

Step 5: This and That. Type **ls -F** and press enter. This command includes a *command-line option* (known in the vernacular as a "flag"). Running the command with this flag will list the contents of the current working directory but will add a / to the end of the name of the subdirectories. You should see a / after the 277 directory you created.

Step 6: Navigation. Type **cd cs277/lab1** and press enter. This command stands for "change directory" and will navigate your terminal session into the directory name you tell it, in this case your lab1 directory. You will save any files you create for this lab in the lab1 directory. Type **pwd** and press enter. You should now see something like this listed as the working directory: */home/users/yourusernamehere/cs277/lab1*. There are two special directory names (.) and (..). (.) means current directory. So typing **cd .** and pressing enter does nothing. (..) means the parent directory. So to navigate back to your home directory from the lab1 directory you could use **cd ..** twice or type **cd ../..**

Step 7: The Root of it All. Every path in the Unix file system begins at the *root path*. No matter where you are in the file system, you can navigate to the root path by typing **cd /** and pressing enter. Do this it now. Now issue the **pwd** and **ls** commands to verify your path and to take a look at all of the directories that exist in the root path.

Step 8: The Way Home. No matter where you are in the file system, you can quickly return to your home directory by typing either **cd** by itself or by typing **cd ~**. Try it out. First type **cd /** and press enter. This takes you to the root directory of the file system. If you type **pwd**, you will notice that your current path is now /. Now, to get back home type **cd** ~ and press enter. Now if you type **pwd** and press enter, you will see you have returned to your home directory.

Step 9: Moving and Shaking. Now we will learn how to move files with the **mv** command. Using the **cd** command, navigate to the directory where you saved/downloaded your lab files. Type **mv cs277-lab1* ~/cs277/lab1/** and press enter. This will move the lab 1 files into the *cs277/lab1* directory we created in Step 4. Notice that this command takes two parameters. The the source file(s) and the destination directory. You will also notice the * at the end of the source. This is a wildcard and tells the move command to move any file that begins cs277-lab1 to the destination directory. You will also notice the ~ on the destination path. Remember ~ this is a shortcut to your home directory. We use it here, so that no matter where we are in the file system, we want to copy relative to your home directory.

Assignment – Part 2

You will now complete the Vim tutorial module to familiarize yourself with the basic features of this powerful command-line based text editor. When you have finished save your work to the file:

~/cs277/lab1/vim.txt

To complete the tutorial, type **vimtutor** and press enter. Follow the directions in each section of the Vim Tutorial module. Be sure to save your work when you are finished!!

Once you have completed the Vim Tutorial, use vi to edit your vim.txt file. At the top of the file put your name on the first line, followed by a blank line and then type a 3 to 4 sentence paragraph about how much experience you have with Unix and C programming. Be sure to save the file when you are finished.

Assignment – Part 3

Now you will try to compile the cs277-lab1.c source file using the GCC compiler.

Step 1: Navigate to *~/cs277/lab1,* type **gcc cs277-lab1.c** and press enter. You will notice some compilation errors printed to the terminal.

Step 2: Open up the source file with vi and fix those errors! Type **vi cs277-lab1.c** and press enter. Try to fix the compilation errors. Give it your best shot. Once you have finished, save your file and run the command from Step 1 again to attempt another compilation. You will know when gcc successfully compiles the source file when gcc returns with no errors and a file named *a.out* is created. Keep opening the source file with vi and correcting the errors until you can get the source to compile without any errors.

Step 3: Once it has compiled, it is time to run the executable produced by GCC. By default GCC will write the executable to a file named *a.out*. Run the executable by typing **./a.out** at the command line and then press enter. If the executable runs correctly, then congratulations you are done with Lab 1!

Submission

You will create a .tar file to submit your lab files.

Step 1: Change to your cs277 directory.

Step 2: Create an archive file by typing **tar -cvf lab1-yourusername.tar lab1/*** and then press enter.
For example: tar -cvf lab1-mharmon.tar lab1/*

Step 3: Email the archive file, lab1-yourusername.tar to me at mharmon@lclark.edu

Hints and Tips

Start early. This assignment is not difficult but does cover a bit of ground, especially if you haven't used these tools in the past. Depending on your experience working with Vim, C source files, the Unix command-line and with GCC, you may want to take a break between each section in order to give yourself time to absorb the material.

Evaluation

To receive full credit:

1) Submit on time cs277-lab.c and vim.txt in the archive format.
2) vim.txt should contain the output produced by completing the Vim tutorial module as well as the other changes required. (50%)
3) cs277-lab.c file should compile and run **--->** ***on a lab machine*** **<---.** (25% if it compiles, 25% if it runs)