CSCI 1730 Breakout Lab 01

Hello World

Learning Outcomes

- Use basic Unix commands: cd, pwd, mkdir, ls, grep, man, apropos, ps, kill, top, less, more, chmod, cp, mv, rm, diff, sort, wc, and find.
- Use scp or sftp to copy files from your odin account to your laptop.
- Use emacs to write programs in a Unix environment.
- Compile, link, and run programs via the command line.

Problem / Exercise

The purpose of this lab is to to practice basic Unix commands, to practice with scp or sftp, to use emacs to write a C program, to use gcc to compile a C program, and to run a compiled C program via the command line.

Requirements

1. Before you begin this lab, you must setup your odin account. A document named `odinSetup.pdf` on eLC has instruction on how to do this.
2. Practice basic Unix commands: cd, pwd, mkdir, ls, grep, man, apropos, ps, kill, top, less, more, chmod, cp, mv, rm, diff, sort, wc, and find. Use the Unix manual (man command) for each of these Unix commands.
3. Use emacs on odin to write the C source code for this assignment in a file named lab01.c. See the Examples section to see what your C program should output. Compile your C program using the following command on odin. If there are any warnings or errors, then you should fix them before you submit this assignment to us for grading.

   ```
   gcc -Wall lab01.c -o lab01.out
   ```
4. Use scp or sftp to copy lab01.c from your odin account to your laptop as discussed in a lecture class. You will need to copy the file(s) for this lab (and future labs and projects) from your odin account to your laptop so you can submit them on eLC.

Examples

After you finish lab01.c, the I/O from running lab01.out should match exactly what is shown below.

```
./lab01.out
Hello World!
I setup my odin account to use gcc and g++ version 11.2.0.
I practiced using basic Unix commands.
```

1 C Program

1.1 Setup

Make sure that all of your files are in a directory called `lab01`.

1.2 Code

For this lab, you should place your code in `lab01.c` and follow the instructions aforementioned.

1.3 Coding Style Requirements

1. All functions must be commented. Comments must include a brief description of what the function does, its input(s), its output, and any assumptions associated with calling it. If your function has a prototype and an implementation separated into different parts of your source code, then you only need to put the comments for that function above its prototype (there is no need to comment both the prototype and implementation; commenting the prototype is sufficient).
2. All global variables and static variables must be commented.
3. All identifiers must be named well to denote their functionality. Badly named identifiers are not allowed. For example, identifiers like a, aaa, b, bbb, bbbb are bad names for identifiers.
4. Every line of source code must be indented properly and consistently.

1.4 README.txt File Requirements

Make sure to include a `README.txt` file (use a .txt file extension) that includes the following information presented in a reasonably formatted way:

- Your First and Last Name (as they appear on eLC) and your 810/811#
- Instructions on how to compile and run your program.

1.5 Other Requirements

1. All written and verbal instructions stated by the teaching staff (lecture instructor, lab instructor(s), etc.) must be followed for this assignment. Failure to follow instructions may result in losing points.

2 Submission

Before the date and time stated on eLC, you need to submit your files on eLC under the Assignment named Lab 01. The submission on eLC must include only the following files.

1. All source files required for this lab (lab01.c)
2. A README.txt file filled out correctly

3 Grading: 10 points

If your program does not compile on odin using the required compiler for this class (`gcc version 11.2.0`), then you'll receive a grade of zero on this assignment. Otherwise, your program will be graded using the criteria below.

Program runs correctly with various test cases on odin	10 points
README.txt file missing or filled out incorrectly	-1 point
Not submitting to the correct Assignment on eLC	-1 point
Late penalty for submitting 0 hours to 24 hours late	-2 points
One or more compiler warnings	-2 points
Not adhering to one or more coding style requirements	-2 points
Not submitting this assignment before its late period expires	-10 points
Penalty for not following instructions (invalid I/O, etc.)	Penalty decided by grader

You must test, test, and retest your code to make sure it compiles and runs correctly on odin..