ECE177: Programming I: From C...

Lab #1 — WSL Basics

1 Introduction

1. Before starting to write code some Linux commands will be introduced to help you stay organized in WSL. Use the following instructions to learn how to create a new "lab2" directory and make a copy of the "hello.c" program from lab1 as a starting point for this lab.

   (a) Open WSL. Make sure the command line looks like this:

   `[Your Username]:~$`

   (b) NOTE: if instead of the tilde (~) you see something like `WINDOWS/system32` reopen wsl by searching your machine for "Ubuntu" and opening that instead. The tilde means you are in your home directory.

   (c) To list the contents of your current directory (home directory) you can use the list command: `ls` by typing the following:

   `ls`

   (d) NOTE: if you completed the WSL setup lab you should see "hello.c" somewhere in there.

   (e) Create a directory using the make directory command: `mkdir`. In order to make a directory called "lab2" use the following:

   `mkdir lab2`

   (f) Now if you retype the list command (ls) you should see the new directory called "lab2". In order to copy `hello.c` into the lab2 directory we can use the copy command: `cp` by typing the following:

   `cp hello.c lab2`

   (g) NOTE: the command means: copy `hello.c` to the `lab2` directory

   (h) Next enter the `lab2` directory with the change directory command: `cd` by typing the following:

   `cd lab1`

   (i) Lastly we will rename hello.c using the move command: `mv`

   `mv hello.c lab1.c`

   (j) NOTE: `hello.c` can be renamed to anything as long as the `.c` is maintained.

   (k) You are now in the `lab2` directory and can use `lab2.c` as a starting point. It is a good idea to use previous labs as starting points for any new labs.

2 Command Summary

1. ls: list contents of current directory.
2. mkdir: make a new directory. The new directory's name must be specified.
3. cd: change to specified directory.
4. cp: copy file to another directory or file.
5. mv: give existing file a new name
6. gcc: Code compilation: gcc yourCode.c
   NOTE: this will compile yourCode.c and produce an a.out file in your current directory
7. Alternatively use: gcc -o yourCode yourCode.c
   NOTE: this will compile yourCode.c and produce file called yourCode in your current directory
8. Running code: ./yourCode