Achtung, Atención, Attenzione, Oppmerksomhet, Attention

You **must** have Xming (Windows) or XQuartz (OSX) installed before attempting this lab.

1 Introduction

The operating system of a computer is the coordinator of all the computer's activities, including memory allocation, process switching, and file access. At Wake Forest University students typically use the Windows operating system on their laptops. In CSC 112 you will use a version of Debian called Ubuntu on a remote server. For this reason, this lab will introduce you to these environments for C++ programming. Specifically this lab covers,

- 1. Login to your bally.cs.wfu.edu account.
- 2. Introduce the Linux environment: file structure and commands.
- 3. Create the directory structure required for this course.
- 4. Use Unix commands to copy/compile/execute an existing C++ program.

2 Logging into Your Account

This semester you will use bally.cs.wfu.edu, a Linux server hosted by the Computer Science Department, to complete your lab assignments. When you access the server you'll likely open at least two windows, one for a command line (Xterm) and the other for an editor (sublime). To do this you may need to install and configure a few applications. See the course web page for details about installing the software and logging into your account.

3 Unix File Structure and Commands

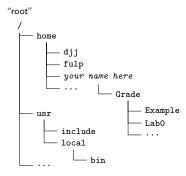


Figure 1: Partial directory structure for your Linux system.

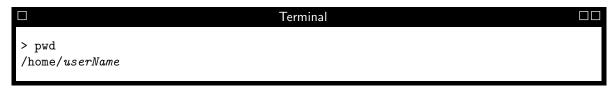
The Linux operating system is organized as a hierarchical file structure. As seen in figure 1, this structure appears like the roots of a tree. At the top is the root directory, which is symbolized by a forward slash /1. Beneath the root directory are all the other directories and files in the Unix system (this includes your files). Therefore, each file and directory has a specific place in the file system, which is called the pathname. Pathnames indicate the way through the file structure.

For example using figure 1, the path from root to your own directory path would be

/home/userName

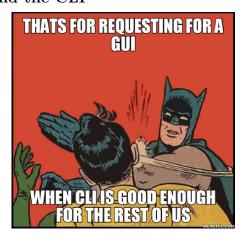
¹As discussed later, the forward slash serves two functions: represents the root directory, and separates names in a path name (unlike DOS). Relax, it will only get more confusing

This path is returned if you issue the *present working directory* command at the terminal prompt. To determine the directory enter pwd at the prompt and hit Enter.



This directory is also called your *home directory*. Another example is the directory that contains the include files necessary for compiling a C++ program. These files are located in /usr/include/. During the semester you will create various files and directories off of your home directory. For this reason you must know a few simple Unix commands, which are introduced in the next section.

3.1 Unix Commands and the CLI



official opening ceremony ribbb

Unix (Linux) commands provide a mechanism to interface with the operating system. These commands allow a user to create files, move directories, compile programs, etc... On the server bally.cs.wfu.edu there are approximately 3,408 commands available (hit the tab key twice and enter yes to see the list). Unix commands (as well as file and directory names) are case sensitive; therefore, entering PWD at the prompt will not return the current directory, you must enter pwd at the prompt. Below is a list of important Unix commands you must know.

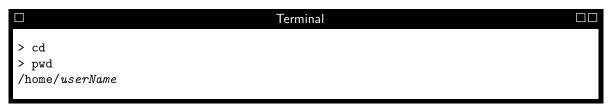
Command	Description
pwd	Returns the path name from root to the current working directory. This is called the <i>full path name</i> .
ls	Lists the current directory. There is a variety of command options for ls. For example, ls -al provides detailed information about each file in the current directory.
mkdir directoryName	Creates a directory called directoryName.
rmdir directoryName	Removes the directory called <i>directoryName</i> . Note the directory must be empty.
cd pathName	Change directory to <i>pathName</i> . Note that entering cd without a <i>pathName</i> will return you to your home directory. Furthermore, cd moves back one directory.
cp fileName1 fileName2	Copy file fileName1 to file fileName2.
rm fileName	Erase file fileName. Once erased you can not get the file back.
more fileName	View the contents of the file called fileName (if it is ASCII).
man commandName	Provides a description of the Unix command <i>commandName</i> including syntax and options.

The path or file name for any command will be relative to the current working directory. For that reason, it is not always necessary to supply the full path name. For example using figure 1, suppose you are in the directory usr and want to move to include. You could enter cd include, instead of supplying the full pathname cd /usr/include

4 Creating Your CSC112 Directory Structure and Compiling an Existing C++ Program

For this portion of the lab you will create two directories, copy a C++ program into your directory, compile the program, then execute the program. For this class (CSC112), all programming assignments must reside in certain directories for grading. This portion of the assignment will cover how these directories are created using the commands in the previous section.

First, move to your home directory. To do this enter the cd command at the prompt. Issue a pwd command to be certain you are in your home directory.



Now create a new directory called CSC112 off your home directory. Issue the following commands to accomplish this.



Now list your home directory. You should see the directory you just created (among other files in your home directory). All programs for this class will reside in this new directory. The program files for each lab/assignment will be contained in a separate directory. For this lab, create a directory called LabO in the CSC112 directory. To accomplish this, first cd to the CSC112 directory. Then create the LabO directory.

```
Terminal

> cd CSC112
> mkdir Lab0
```

4.1 Copy and Compile

A C++ program (source) file called hello.cpp is located at course web page. To obtain the file, you can use firefox for a web browser (type firefox & at the command prompt). OK, that's a lame way of doing it, way too slow. You can actually copy the program from the web using wget command as follows.

```
Terminal

> wget www.cs.wfu.edu/~fulp/CSC112/hello.cpp
```

After you have copied the file, place it in your LabO directory (use the mv or cp command if necessary). Then compile and execute the program. The command to compile a C++ program is

```
g++ -Wall -pedantic -o executableName sourceName
```

Where <code>sourceName</code> is the C++ program (source) file and <code>executableName</code> is the resulting executable file. For the <code>hello.cpp</code> compile the program so the resulting executable is called <code>hello</code>. The g++ compiler has more options, which will be introduced as they are needed. To run the program enter <code>./hello</code> at the prompt.

5 The Grade Directory

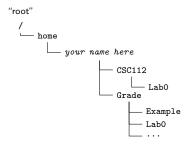


Figure 2: Partial structure for your home directory.

Your home directory contains a **Grade** subdirectory. Inside this directory are other subdirectories for each lab assignment this semester. This is depicted in Figure 2. Once you are done with an assignment, you will copy your completed work to one of these subdirectories for grading. If you work is not in the correct subdirectory by the assignment due date, then it will be considered late.

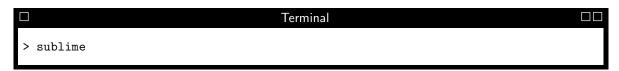
Assume you are currently in your CSC112/Lab0 directory and are ready to copy your work (lab0.cpp) to your Grade/Lab0 directory. To do so you would enter the following command.

```
Terminal

> pwd
/home/yourLoginID/CSC112/Lab0
> cp lab0.cpp ../../Grade/Lab0
```

6 A Simple C++ Program

In this portion of the assignment you will write your own C++ program. To accomplish this you will need to start a text editor. Although our Unix environment has multiple text editors installed, we will start with edit. To start sublime enter sublime at the Unix prompt.



6.1 Program: Temperature Conversion

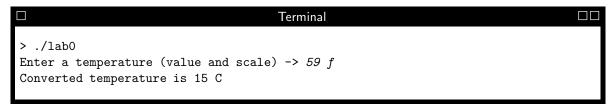
Create a new file called lab0.cpp in your Lab0 subdirectory. Then write a program that solves the following problem. Be certain you follow the programming points at the end of this section.

Write a simple program that will convert temperatures in Fahrenheit to Celsius, or vice versa depending on the user's input. The equations you need are as follows, where c is the temperature in Celsius and f is the temperature in Fahrenheit.

$$c = (f - 32) \times \frac{5}{9}$$
$$f = c \times \frac{9}{5} + 32$$

6.2 Program Description

Prompt the user to enter a temperature, which is a number followed by an upper case or lower case 'c' or 'f' for the scale. You may assume the number and the character will be separated by at least a space. If the scale is 'c' then display the temperature in Fahrenheit. If the scale is 'f' then display the temperature in Celsius. If the scale is neither 'c' or 'f' then display an error message.

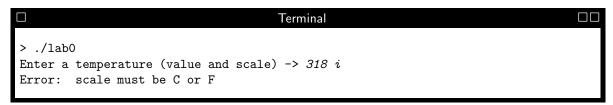


```
Terminal

> ./lab0

Enter a temperature (value and scale) -> 33 C

Converted temperature is 91.4 F
```



6.3 Programming Points

You must turn-in an electronic copy of your programs. This must be done before the assignment due date. Electronic copies will be submitted by placing the source file (lab0.cpp) in your Grade/Lab0 directory. You must adhere to all of the following points to receive credit for this program.

- 1. Name your source (C++ program) file lab0.cpp
- 2. Program must compile without errors or warnings using the -Wall and -pedantic options
- 3. Global variables are not allowed. All variables must be declared within a function.
- 4. All prompts and output must appear as described. Please note the examples carefully.
- 5. All code **must** follow the style guidelines for this course and be documented.

7 Electronically Submitting Your Program

You must turn-in an electronic copy of your program. This must be done before the assignment due date. Electronic copies will be submitted by placing the source file (lab0.cpp) in your ~/Grade/Lab0 directory.