Programming Assignment 1

Hello World!

Goals: This assignment forces you to experiment with several components routinely used in software development, including: GNU compilers, SecureShell, CMake, package managers, external libraries, and Git.

Assignment: compile a Git repository using the Tuxedo server hosted by our department. Take a screenshot of the graphical output and submit it to Canvas as an image. That's it.

Git repository: https://github.com/STIM-Lab/helloworld

Server: tuxedo.egr.e.uh.edu

Running on Tuxedo ECE Server

The following steps provide an overview on how to build and execute the application on the Tuxedo ECE server. This server already has Git, CMake, and all of the necessary compilers installed.

1. Connect to the server. You'll need some sort of SecureShell client that supports X forwarding (for graphics). My recommendation for Windows users is MobaXTerm, which has a user interface that should be intuitive to most users. Mac and Linux users can use OpenSSH:

   ```
   ssh username@tuxedo.egr.e.uh.edu -Y
   ```

   The "-Y" argument sets up graphics forwarding so you'll be able to see graphics and UI elements on your client.

2. Install vcpkg to manage external libraries. It may be helpful to go to the main vcpkg website and learn a little bit about how to use it. In short, you can clone it to your home directory using git:

   ```
   git clone https://github.com/microsoft/vcpkg.git
   ```

   This will create a vcpkg directory in your home folder on the server that will store libraries need to build software.

3. Clone the helloworld repository. The website on GitHub is provided above and has the repository locations. This is the demo code that you'll need to compile for this assignment:

   ```
   git clone https://github.com/STIM-Lab/helloworld.git
   ```

   This will copy the repository to a directory called "helloworld" on the server (similar to vcpkg).

4. Change to the new source directory:

   ```
   cd helloworld
   ```

5. Run CMake using the vcpkg profile (I've created one in the source directory that assumes you installed vcpkg into your home directory using the command in Step 2:

   ```
   cmake --preset=vcpkg
   ```

This creates a subdirectory called “build” in the “helloworld” directory. This is where the application will be compiled to create an executable.

6. Build the repository:

   ```
   cmake --build build
   ```

7. Execute the program:

   ```
   ./build/helloworld
   ```

Recommendations for Developing on Your Own System

Setting up a development environment on your own system will vary based on your preferences and operating system. I can provide some recommendations:

- You’ll probably have to install Git and CMake yourself.
- For Windows systems I recommend installing Visual Studio Community. When you run the installer, make sure to download and install the tools for C/C++ development.
- If you have any specific questions or need recommendations, feel free to contact me.