CS 15 Lab 0: System Setup

Introduction

Welcome to CS 15! This lab is intended to help students who have not taken CS 11 at Tufts acclimate to the course workflow (which is the same as the current COMP 11 workflow). That said, even if you have taken CS 11 at Tufts, it may be useful to refresh your memory and get ready to hit the ground running this semester. Be sure to ask questions if you get stuck.

It is expected that you have these steps completed before Lab 1.

Part 0: System Setup

This section will help you do the following things:

- Access a command line terminal
- Ssh to the Halligan server
- Use basic `UNIX` commands
- Have VSCode and Remote-SSH installed and configured

Working Remotely

Remote access to the server will allow you to, create, edit, and remove files on the Halligan homework server from your local system. It will also allow you to compile and run your work in a course-approved environment. The following steps should be familiar to those of you who have taken CS 11 here at Tufts, but please don't hesitate to ask for help with any of them.

Step 1: Terminal Access

In order to work remotely on your own computer, you'll need a program that lets you interact with the department server. To do this, you first need access to a command-line interface.

- For Windows:
  - Use `MobaXTerm` or `PuTTY`
  - If using `MobaXTerm`, I'd suggest going to
    - Settings/Configuration
    - terminal tab
    - uncheck "Backspace sends ^H"
- For Mac:
  - Use the native Terminal application
  - You many want to enable developer tools. To do this, Open the Terminal, and type: `DevToolsSecurity -enable`
  - For GUI applications, we suggest `https://www.xquartz.org/`
- For GNU / Linux
  - Use the native terminal application

Step 2: SSH

After you have access to a terminal, you'll use `ssh` (Secure SHell) to connect to the Halligan homework server. The `ssh` command establishes a connection between your local machine and (in this case) the homework server, which allows you to run commands on (your personal folder in) the homework server from your machine.

To use `ssh`, open the terminal, and type:

```
ssh utln@homework.cs.tufts.edu
```

Where *utln* is your CS department-provided *utln*. If you do not yet have one, please email `staff@eecs.tufts.edu`. They are usually quick to respond. Once you have a *utln*, please return here to continue.

Note: If you want to open up a GUI (graphical user-interface) running emacs or kate, then you'll need to tell the server that it can open windows on your machine. This is almost the same as above:

```
ssh -X utln@homework.cs.tufts.edu
```

The -X tells homework it can use X-Windows, to communicate with your computer.

Step 2: SCP

It's often useful to know how to copy files between your local system and the homework server. SCP (secure copy protocol) is a tool for just this purpose. The syntax is as follows:

```
scp utln@homework.cs.tufts.edu:/server-path/ local-path
```

So, if you ever need to copy a file from your home directory on the server to the current directory on your local machine, you can do:

```
scp utln@homework.cs.tufts.edu:~/filename .
```

And, if you ever need to copy a file from your local machine to the server, you can do:

```
scp local-path/filename utln@homework.cs.tufts.edu:~/
```

Also, use the -r option to copy any directories (recursive mode).

```
scp -r local-path utln@homework.cs.tufts.edu:~/
```

UNIX Reference

You will probably notice that the command-line interface is quite different from the graphical-user interfaces we commonly use today (no mouse!) Because you will be spending quite a bit of time in this kind of environment, it's a good idea to be familiar with basic Unix commands (Linux is kind of Unix operating system). You will need to create directories, navigate among directories, create, copy and remove files, compile files, etc. One good reference is: `http://www.ee.surrey.ac.uk/Teaching/Unix/`

VSCode and Retmote-SSH

Initial Setup

You will need a text editor installed in order to edit your code! While there are many available options (`emacs`, `vim`, `Atom`, `Sublime`, etc.), we are officially supporting `VSCode` with the `Remote-SSH` package.

Here we will walk you through the process of getting `VSCode` up and running on your personal computer. `VSCode` can be downloaded for use on any operating system here: `https://code.visualstudio.com/`

Remote-SSH Installation

To ssh into the Halligan server with minimal fuss, you'll want to use the Remote-SSH package. There are detailed instructions to install this, written by a former CS-15 TA Kevin Destin: `https://github.com/kedestin/Quick-References/blob/master/Development/vscode-remotessh.md`.

Notes on VSCode

If you're having issues with weird terminal output after ssh-ing with Remote-SSH, try the following fix:

- Go the the Extensions tab
- Click the settings gear for Remote-SSH
- Click "Extension Settings"
- At the bottom there is an option called "Remote.SSH : Use Local Server" - uncheck this box
- Connect to the host again
- This time you are prompted for the Remote server's os, select 'Linux'
- Input authentication
- Done!

Also, one thing to remember about `VSCode` is that it uses space in your home directory, under a folder named `.vscode-server`. Sometimes, this folder can take up lots of space - if you're seeing a "disk quota exceeded" error when creating a file, etc., try removing the `.vscode-server` directory.

Part 1: Compile and run "Hello World!"

Note! The Unix commands below are shown to help you understand what they do. Understanding the commands will save you a lot of time! Therefore, do not copy and paste them. Instead, type the commands by hand. This will help you learn what they are and how they work. Please ask questions about anything that isn't clear.

Great! Let's get started.

```
# First, open a terminal on your local system
[matth.DESKTOP-L0J06D1]:

#  Now, ssh to the Halligan homework server
[matth.DESKTOP-L0J06D1]: ssh your_utln@homework.cs.tufts.edu

#  You'll see something like this
vm-hw09{mrussell}101:

# Make a cs15m1 directory
vm-hw09{mrussell}101: mkdir cs15m1

# Move into your CS 15 directory
vm-hw09{mrussell}101: cd cs15m1
```

Great! In `VSCode` now, under your `cs15m1` directory, make a directory for this walkthrough:

```
mkdir lab0_setup
```

Excellent! Inside that directory, open a file named `hello_world.cpp`. Write out the following text, which is a simple example in C++:

```cpp
#include <iostream>

int main() {
    std::cout << "Hello world!" << std::endl;
    return 0;
}
```

When you save the text, head back to your terminal (which has a `ssh` connection to the homework server). You should be able to run the command:

```
cd lab0_setup
```

If this does not work, make sure that you have saved your file correctly. Please ask questions on piazza or ask a TA for help if stuck here!

Okay, now compile your code in the terminal:

```
clang++ -Wall -Wextra hello_world.cpp
```

By default, an executable file named `a.out` is created by the preceding step. At this point, you can run the file with the command `./a.out` You should see the following text in the terminal:

```
Hello world
```

Note the `./` in the preceding command. This is to indicate that the a.out file is in the current working directory. You can name your program by specifying an output filename using `-o outputfilename` to the compile command:

```
clang++ -std=c++11 -Wall -Wextra -o hello_world hello_world.cpp
```

and then you can run the program like this: `./hello_world`

Note that this is a simple example, and it does not contain any documentation. Please read the `https://www.cs.tufts.edu/comp/15-2021s/reference/StyleGuide2020.html` — all code that you submit for the course will be expected to adhere to this style guide.

Part 2: Read the Course Website!

At this point, it would be a good idea to read carefully through the course website. There is lots of information there designed to help you!