BYOD: Bring Your Own Device

Developing C on Your Own Machine
Computer Science I

Department of Computer Science & Engineering

University of Nebraska–Lincoln

Introduction

Using the lab computers will always be an option. However, you may want to untether yourself from the lab and develop on your own machine, either desktop or laptop. There are many options for doing this, the following are only a few suggestions.

Both of these suggestions have you make a connection to your "Z-drive" on the CSE server. This is a remote file system where you can store all of your files, but interact with them as if they were on your computer. It is similar to plugging in a memory stick: it will appear as yet another drive on your computer that you can drag-drop files from/to and open/save files on.

These suggestions still have you compiling and running on the CSE server from the command line. This saves you from having to install your own C compiler/development software. Also, there are many libraries that your labs and exercises may require. All of these have been installed for you already on the CSE server. If you choose to go your own way and install alternative software, you may have difficulty installing libraries if you don't know what you're doing.

1 Remote/Online Users

For students in the online section and students who are looking to completely untether from the physical CSE labs and the UNL campus, one solution is to use an online IDE (Integrated Development Environment). We recommend using Harvard's CS50 IDE (`https://ide.cs50.io/`) which gives you an entire development environment in your browser with no installations and the full functionality of a virtual machine all for free. For a step-by-step process for getting started, see lab 1.0.

There are many other alternatives for online, browser-based IDEs. These all vary according to cost and functionality.

- REPL.it: https://repl.it/
- Gitpod: https://gitpod.io/
- Codeboard: https://codeboard.io/
- Online GDB: https://www.onlinegdb.com/online_c_compiler

2 On Campus Users

If you plan on doing most of your work on campus, you can utilize the CSE server (its resources, file storage, compiler and other software) from the command line. We recommend that you use an SSH client (in order to execute commands at the command line) in conjunction with a samba connection so you can connect to your "Z-drive" directly over the network. This solution only works if you are on campus since a samba connection would be fire-walled off campus.

Alternatively, you could use a Virtual Private Network (VPN) to make it appear you are on campus. UNL's VPN service (and the required software) can be found at: https://its.unl.edu/services/vpn/.

2.1 Windows

1. Connect to your Z: drive using the following FAQ:
   https://cse.unl.edu/faq-section/windows#node-290
   (hint: the URL contains a ONE, not a lower case "L").
2. Install Atom IO https://atom.io/, a free code/text editor. Use this editor to edit files on your Z drive once connected.

   Note: once installed, you should change the default line ending for atom.io. To do this:

   a) Click `File` → `Settings`

   b) Click `Packages`

   c) "Search" for `line-ending-selector`

   d) Under Settings, select `LF` instead of `CRLF`
3. Download and install Putty, an SSH (Secure Shell) client:
   http://www.putty.org/
   Once you've written your program(s), login to the CSE server (the hostname is `cse.unl.edu`) using Putty and compile and run your programs on the CSE server.

2.2 Mac

1. Connect to your Z: drive using the following FAQ:
   https://cse.unl.edu/faq-section/macintosh#node-295
   (hint: the URL contains a ONE, not a lower case "L").
2. Install Atom IO https://atom.io/, a free code/text editor. Use this editor to edit files on your Z drive once connected.
3. Use Terminal (Apps → Utilities → Terminal) to login to the CSE server and compile and run on the CSE server. To login, type `ssh login@cse.unl.edu` where `login` is replaced with your CSE login.

Alternatives

The following are some alternatives if you want to explore further. All of them are cross-platform solutions. Keep in mind that you will need to install and troubleshoot various required libraries yourself if you choose these.

- Eclipse for C/C++ is a full Integrated Development Environment (IDE): https://www.eclipse.org/downloads/eclipse-packages/
- Code::Blocks is another IDE for for C/C++: http://www.codeblocks.org/
- Sublime Text is an excellent code editor but is commercial: https://www.sublimetext.com/ (this is an alternative to atom.io, it does not include a compiler).
- If you are off campus and need to transfer files between your own machine and the CSE server, you can use an SFTP (Secure File Transfer Protocol) client. We recommend FileZilla: https://filezilla-project.org/ (download the *client*, not the server). For the host, use `sftp://cse.unl.edu` and use your cse login and password.