Preparing your Mac for EECS338

Getting a compiler

The following instructions are adapted from here. If you find any portion of them unclear, please visit the link before asking for clarification. It is highly recommended that you upgrade your operating system to at least Mavericks (10.9), as this makes the installation process easier.

In OSX, the compiler is installed as part of a package called `command line tools`. For users of OSX Mavericks (10.9), Yosemite (10.10), or El Capitan (10.11), The package can be installed easily by opening up the `Terminal` app, and entering the following command:

```
xcode-select --install
```

This command will install the minimum portion of Apple’s Integrated Development Environment, `Xcode`, when you click `Install` on the dialog box that pops up. If you receive the error `Can’t install the software because it is not currently available from the Software Update server`, fear not, for you may already have installed all of `xcode`, which will include the `command line tools`.

Once completed, reboot your computer, and run the following command to verify everything went successfully:

```
gcc --version
```

If you recevied the error `command not found: gcc`, then you need to install `xcode`, and install `command line tools` from `xcode`. If you received any other non-error message, then you successfully installed a C compiler.

If you received an error, your operating system is older than Mavericks, and you must install `xcode` to get the `command line tools`. Again, it is strongly recommended that you upgrade your operating system through the `App Store` and follow the directions above, but it is not required.

Open the `App Store`, and search and install `xcode`. Follow the instructions noted below:

```
Xcode 4.3 and later does not automatically install the command line tools. To install them, open the Xcode application, go to the Preferences window, to the Downloads section, and click the Install button next to Command Line Tools. Be sure to return to this window after every Xcode upgrade to ensure that the command line tools are also upgraded.
```

After completion, reboot, and verify the installation was successful by running the following command:

```
gcc --version
```

Using an editor to write code

Writing code in OSX’s built-in program `TextEdit` is not a very pleasant experience. For a basic text editor with syntax highlighting and bracket completion, `Sublime Text` is recommended.

For minor edits on the command line, `nano` is recommended, but `emacs` and `vim` provide more advanced alternatives.

Connecting to the `eecslinab3.case.edu` server

The common server we will use in this class is `eecslinab3`. Please note that you must be on the campus network or connected through a VPN to access this server.

Connecting to the server is done through `ssh`, short for Secure Socket Shell. The simplest way to connect is by entering the following command in the `Terminal` app:

```
ssh <cwru network id>@eecslinab3.case.edu
```

If my CWRU network ID was `abc123`, I would connect by running the command:

```
ssh abc123@eecslinab3.case.edu
```

If this is your first time connecting to the server, you will receive a message asking you to confirm the authenticity of the host.

```
The authenticity of host 'eecslinab3.eecs.cwru.edu (129.22.149.166)' can't be established.
ECDSA key fingerprint is [...]
Are you sure you want to continue connecting (yes/no)? yes
```

Type in `yes`, and hit enter to contine.

Congratulations, you're now connected to the `eecslinab3` server!

Transferring files to and from the server

Now that you're connected, you'll want to put the code you wrote on the server. This is done through `scp`, short for Secure Copy. `scp` shares much of the same authentication methods as `ssh`, so once `ssh` is working for you, `scp` will also work.

Before you transfer any files, there's a couple things to note about directory structures:

`~`, the tilde character, is shorthand for what's known as a user's `home directory`. When you `ssh` into a server or open the `Terminal` app, your current location is your `home directory`.

`.`, the period or dot character, is shorthand for what’s known as the `current directory`. This is wherever you currently are.

`..`, two dots, is shorthand for what’s known as the `parent directory`. This is the directory that contains your `current directory` (i.e. it’s one folder up from where you are).

To move a file called `bar.c` sitting on the desktop of your Mac to the server, the following command would be used:

```
scp ~/Desktop/bar.c abc123@eecslinab3.case.edu:~
```

The syntax is the following:

```
scp [file] [username]@[server]:[where to put the file on the server]
```

So we can see that we copied the file `bar.c` and placed it in our `home directory` on `eecslinab3`. To actually try this, you can create the file on your Mac with:

```
touch ~/Desktop/bar.c
```

and then `scp` it with the command above. Once it has been `scp`'d over, `ssh` in and verify that the file is there with the `ls` command.

If you have a file on the server that you want to copy back to your local computer, `scp` has the following syntax:

```
scp [username]@[server]:[where the file is] [where to put the file on your local computer]
```

To verify this, we can remove our local copy of `bar.c` with the following command:

```
rm ~/Desktop/bar.c
```

Then we can copy the one left on the server back over:

```
scp abc123@eecslinab3.case.edu:~/bar.c ~/Desktop
```

Additionally, you can copy over entire directories with `scp`. To test this, you can make the directory `foo` on your local computer with the following command:

```
mkdir ~/Desktop/foo
```

and move `bar.c` into it with the following command:

```
mv ~/Desktop/bar.c ~/Desktop/foo/bar.c
```

You can then copy the entire `foo` directory to the server with the command:

```
scp -r ~/Desktop/foo abc123@eecslinab3.case.edu:~
```

in this command, the `-r` `flag` stands for recursive, indicating `scp` should copy the `~/Desktop/foo` directory and everything inside.

You can remove your local copy of the directory with the command:

```
rm -r ~/Desktop/foo
```

and copy it back from the server with the command:

```
scp -r abc123@eecslinab3.case.edu:~/foo ~/Desktop
```

You can verify that `bar.c` is still in the folder with the command:

```
ls ~/Desktop/foo
```

If you’re confused about any of these commands, you can read the `man`, short for manual, pages on the command (or use Google). You can do this with the following command:

```
man [command]
```

So if you’re confused how `scp` works, you can find out with the following command:

```
man scp
```

When reading the `man` pages, you scroll with the arrow keys, and exit with `q`.