

Assignment I

Download example code from this week's lectures and learn how to compile and run the code. In this lab, we will use accounts on the cslinux server at Eastern's computing center. There are some hoops to jump through to make that work.

I. Accessing the vpn

In order to use the cslinux system, you will need to download and run a vpn client to connect to the ewu vpn.

Mac and Windows Instructions:

Follow the instructions under "Installing VPN" on [this webpage](#).

Linux Instructions:

Only install on Linux if you already know how to use linux tools pretty well. If you really want to try it, feel free to ask me how to do this

Next you need to get a terminal program running on the cslinux computer. The terminal is where we will work with C and other tools to run programs.

In order to get the terminal running you will need to run the program **ssh** on your computer.

2. Getting **ssh** on your computer

Mac Instructions:

Open the "terminal" application. you should see a prompt. Type the following into the terminal:

```
`ssh <<your ewu user id >>@cscd-linux01.eastern.ewu.edu`
```

For example, someone's user id might be `ewarren05` so I use the following string to log in through ssh:

```
ssh ewarren05@cscd-linux01.eastern.ewu.edu
```

Windows Instructions:

Windows does not have a native terminal app but there is an add-on called `putty` which can be installed and used for ssh. `putty` can be found [here](#). Install putty and then run it to get a terminal on the ``cscd-linux01.eastern.ewu.edu`` server.

Linux Instructions:

Open the “terminal” application. you should see a prompt. type the following into the terminal:

```
`ssh <<your ewu user id >>@cscd-linux01.eastern.ewu.edu`
```

For example, someone’s user id might be `ewarren05` so I use the following string to log in through ssh:

```
ssh ewarren05@cscd-linux01.eastern.ewu.edu
```

3. Work With the Example Programs From the Lecture

The first week’s lectures include some example programs. The remainder of this assignment will be to learn some basic linux terminal commands for navigation and file editing, and to do download, compile, and run the example programs from the lectures.

Some Basic Terminal Commands We Need.

The linux terminal is where we will do most of our work this quarter. The terminal has a rich variety of commands and capabilities but can be frustrating at time since you have to know how to use it before you use it.

We are going to use just a small subset of commands at first to:

- Find where we are
- Create a directory
- Change from one directory to another
- Download code
- Compile code
- Run our programs

In the following part of the lab, we are assuming that you have already connected to the vpn and run `ssh` to get a terminal on the cslinux computer. If you have built your own local linux system, then these commands will work there as well.

When we successfully attach to the system, the screen looks like this:

```
jdumoulin05@cscd-linux01: ~  
File Edit View Search Terminal Help  
      .MMMMMMMM. M MMMMMMMMMMMMMMM  
      .MMMMMMMMM +MM DMMM$ MMMMMMM  
      .. .. .MMMMM. MMMM MMMM  
      MMMMMMMMMO MMMMMMMMMMMM .M .  
      .~M MMMMMMMMM=MM$.=MMMMMMMMN.  
      .MMMMMMMMMMMMMMMM.MMN =MMMMM  
      NM. NMNMMMMMMMMM.N .MM..  
      . M.MM, = MN . .MMMMMM8 .  
      .M+MMMMZ +MMMM .MMMMMMMMMMMM.  
      .MM M MM MMMMM .MMMMMMMM.  
      .~M,. MMM. ... .MMMMMMMMMM  
      MMM .MI. .MMMMMM .  
      M +. 7 .  
  
Welcome to the cscd-linux01 student shell server.  
  
Please observe the following rules:  
  1.) Any activity that may be perceived as malicious is grounds for termination  
      of shell server privileges and possible disciplinary action.  
  2.) Observe the ACM code of ethics (http://www.acm.org/about/code-of-ethics)  
  3.) Do not write to other user's ttys without agreeing to do so beforehand.  
  4.) Think before you type.  
  
For more information, tutorials on usage, and to disable this banner see:  
http://cslabswiki.compsci.ewu.edu/wiki/index.php/Shell\_Server  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Sat Apr 4 11:19:51 PDT 2020  
  
System load: 0.0 Processes: 126  
Usage of /: 38.1% of 27.20GB Users logged in: 0  
Memory usage: 9% IP address for eth0: 146.187.134.22  
Swap usage: 0%  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
  https://ubuntu.com/livepatch  
Last login: Sat Apr 4 10:59:09 2020 from 172.16.9.220  
jdumoulin05@cscd-linux01:~$
```

The `$` is where we can type by clicking into the window. If you press `Enter` you will see the terminal advance down the screen. Note the the mouse does not work in the terminal window. It can be used to select text though.

For a general overview of linux terminal commands there are many good sources.

[This page](#) has a lot of good advice.

Where Are We? - pwd

When you want to know where you are, you can type the command `pwd`. This command tells us where our terminal is currently pointing in the tree of directories on the computer's file system. the terminal displays the directory information like this:

```
jdumoulin05@cscd-linux01: ~
File Edit View Search Terminal Help
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sat Apr 4 11:19:51 PDT 2020

System load: 0.0          Processes: 126
Usage of /: 38.1% of 27.20GB Users logged in: 0
Memory usage: 9%          IP address for eth0: 146.187.134.22
Swap usage: 0%

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch
Last login: Sat Apr 4 10:59:09 2020 from 172.16.9.220
jdumoulin05@cscd-linux01:~$
jdumoulin05@cscd-linux01:~$
jdumoulin05@cscd-linux01:~$
jdumoulin05@cscd-linux01:~$ pwd
/home/EASTERN/jdumoulin05
jdumoulin05@cscd-linux01:~$
```

This shows that we are logged on to Eastern's system in our home directory.

New Directories and Navigation - mkdir and ls

Next we are going to make a new directory that is inside our home directory. we do this with the command `mkdir`.

At the terminal prompt, type

`mkdir projects` and press `Enter`.

This will create a new directory at the current location called 'projects'.

Now type `ls` and `Enter`.

This will show you the names of all directories and files at the current location.

`mkdir` stands for Make Directory. This creates a new directory at the current location.

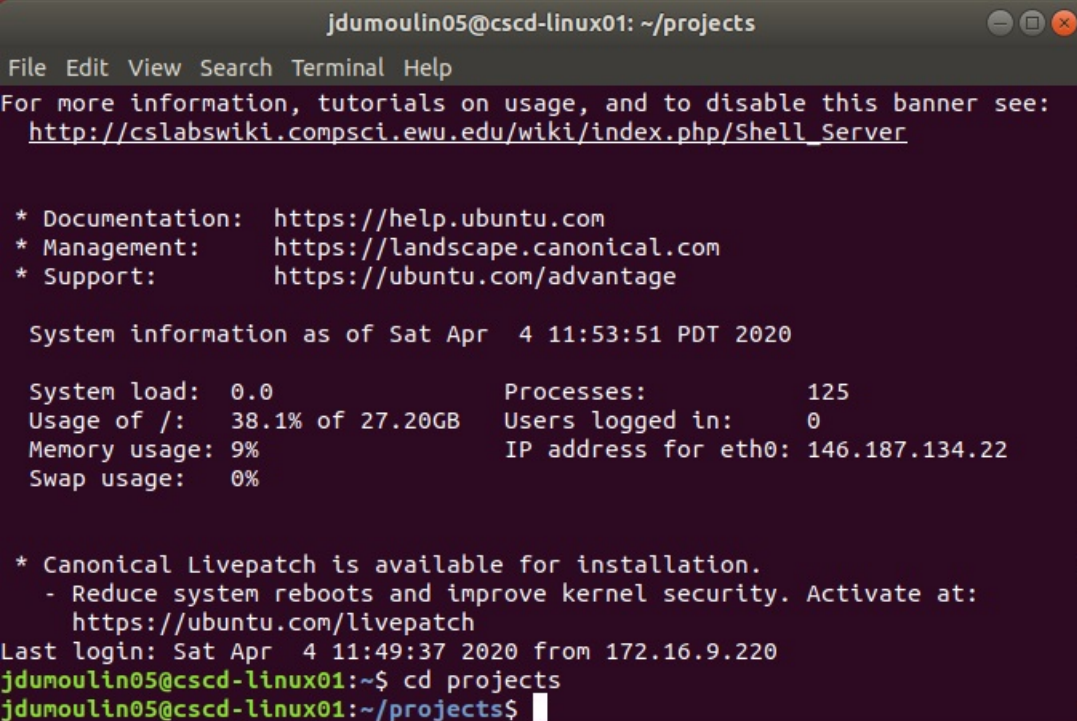
`ls` stands for List Contents. ``ls simply displays all the contents (directories and files) in the working directory.

Changing directories - cd

Next, type

`cd projects` and `Enter`.

Now you have changed the current location to be inside the new directory you created. Your screen should look something like this:

A terminal window titled 'jdumoulin05@cscd-linux01: ~/projects'. The window shows a menu bar with 'File Edit View Search Terminal Help'. Below the menu bar, there is a message about getting more information, followed by system information as of Sat Apr 4 11:53:51 PDT 2020. The system information includes system load, usage of /, memory usage, swap usage, processes, users logged in, and IP address for eth0. There is also a message about Canonical Livepatch. The terminal shows the user's last login and the command 'cd projects' being executed, changing the prompt to 'jdumoulin05@cscd-linux01:~/projects\$'.

You can see that the prompt has changed to show the current directory is “projects”.

Downloading Source Code from github - git clone

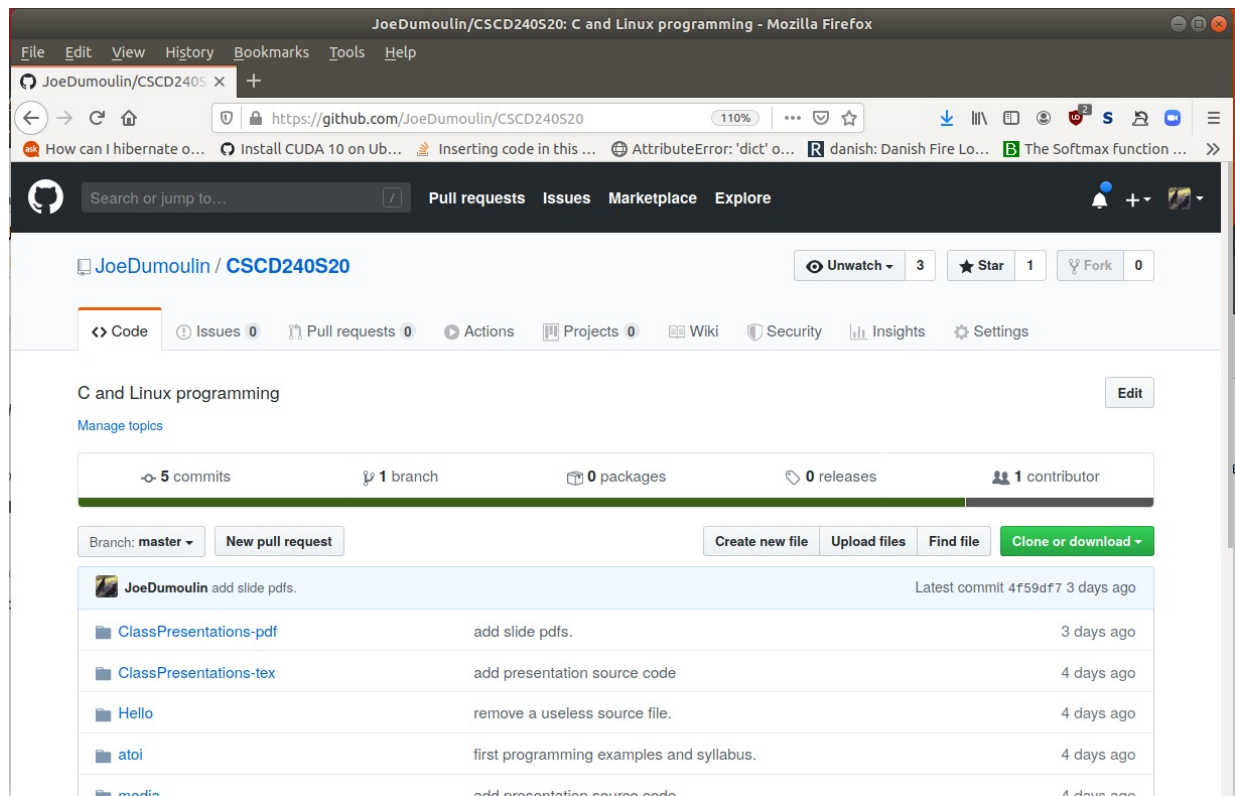
Github is a very popular source code repository for open source and shared source. You may have downloaded and used code from github before. I guarantee you will do this in the future if you stick with Computer Science as a career.

Github is built on a tool called ‘git’, which is written in C and is used by the maintainers of the Linux operating system to manage changes to the system. git is a really important program for CS in industry.

Now we will use the command line version of git to get source code and other things from my github repository (repo).

Make sure you are in the projects folder by looking at the command prompt to see if it ends with `~/projects$`. If you are not in the projects directory, use `cd projects` to go there.

open a window in your browser and navigate to `https://github.com/JoeDumoulin/CSCD240S20`. This is my github page for this class. The browser should display something like the following:



Press the green button on the far right center of the page (Clone or Download) and then observe the text box that appears under “Clone with HTTPS”. This is the name of the repository that we will clone to our linux projects directory.

Next type

```
git clone https://github.com/JoeDumoulin/CSCD240S20.git and press Enter.
```

This should start copying the repository to your projects directory.

When the prompt reappears, type `ls` to see what was created by the clone.

```
jdumoulin05@cscd-linux01: ~/projects
File Edit View Search Terminal Help
System load: 0.16          Processes: 126
Usage of /: 38.1% of 27.20GB Users logged in: 0
Memory usage: 9%          IP address for eth0: 146.187.134.22
Swap usage: 0%

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
Last login: Sat Apr 4 11:53:52 2020 from 172.16.9.220
jdumoulin05@cscd-linux01:~$ cd projects
jdumoulin05@cscd-linux01:~/projects$ git clone https://github.com/JoeDumoulin/CSCD240S20.git
Cloning into 'CSCD240S20'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 33 (delta 4), reused 29 (delta 3), pack-reused 0
Unpacking objects: 100% (33/33), done.
jdumoulin05@cscd-linux01:~/projects$ ls
CSCD240S20
jdumoulin05@cscd-linux01:~/projects$
```

next, type

`cd CSCD240S20/Hello` and then type `ls`.

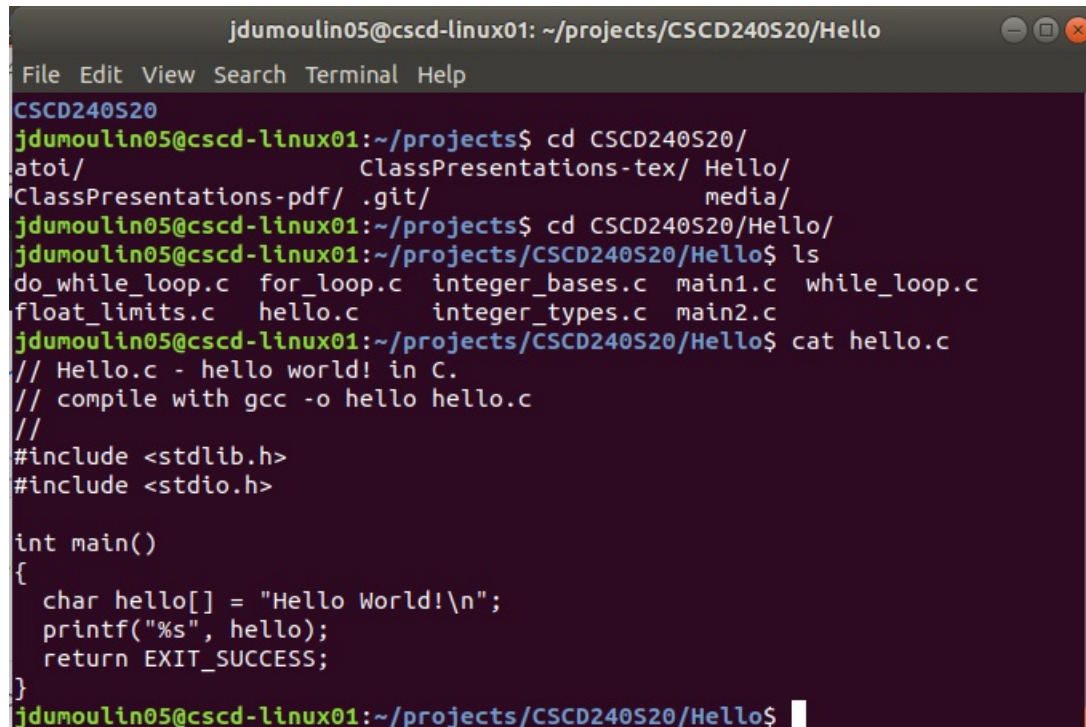
You will see something like the following:

```
jdumoulin05@cscd-linux01: ~/projects/CSCD240S20/Hello
File Edit View Search Terminal Help
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
Last login: Sat Apr 4 11:53:52 2020 from 172.16.9.220
jdumoulin05@cscd-linux01:~$ cd projects
jdumoulin05@cscd-linux01:~/projects$ git clone https://github.com/JoeDumoulin/CSCD240S20.git
Cloning into 'CSCD240S20'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 33 (delta 4), reused 29 (delta 3), pack-reused 0
Unpacking objects: 100% (33/33), done.
jdumoulin05@cscd-linux01:~/projects$ ls
CSCD240S20
jdumoulin05@cscd-linux01:~/projects$ cd CSCD240S20/
atoi/          ClassPresentations-tex/ Hello/
ClassPresentations-pdf/ .git/          media/
jdumoulin05@cscd-linux01:~/projects$ cd CSCD240S20/Hello/
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ ls
do_while_loop.c for_loop.c integer_bases.c main1.c while_loop.c
float_limits.c  hello.c      integer_types.c main2.c
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
```

each for the file ending in “.c” is a C program source file. We can display the contents of one of these files by using the `cat` command.

```
cat hello.c
```

You should see something like:

A terminal window titled 'jdumoulin05@cscd-linux01: ~/projects/CSCD240S20/Hello' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
CSCD240S20
jdumoulin05@cscd-linux01:~/projects$ cd CSCD240S20/
atoi/          ClassPresentations-tex/ Hello/
ClassPresentations-pdf/ .git/          media/
jdumoulin05@cscd-linux01:~/projects$ cd CSCD240S20/Hello/
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ ls
do_while_loop.c  for_loop.c  integer_bases.c  main1.c  while_loop.c
float_limits.c   hello.c     integer_types.c  main2.c
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ cat hello.c
// Hello.c - hello world! in C.
// compile with gcc -o hello hello.c
//
#include <stdlib.h>
#include <stdio.h>

int main()
{
    char hello[] = "Hello World!\n";
    printf("%s", hello);
    return EXIT_SUCCESS;
}
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
```

The screen is showing you the source code in the file. Later on we will learn how to edit the files but for now, we are just going to compile and run `hello.c`.

Compiling the Code

Notice that near the top of the file there is a comment that begins ‘compile with’. This is the command needed to compile this program. type the command (it starts with “gcc”) into the the terminal and press `Enter`.

If there are no errors in the source code, you should get no messages, just the next prompt.

Run Your Program

To run the program in the terminal, type the following

```
./hello followed by Enter
```

You should see something like the following.


```
jdumoulin05@cscd-linux01: ~/projects/CSCD240S20/Hello
File Edit View Search Terminal Help
do_while_loop.c for_loop.c integer_bases.c main1.c while_loop.c
float_limits.c hello.c integer_types.c main2.c
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ cat hello.c
// Hello.c - hello world! in C.
// compile with gcc -o hello hello.c
//
#include <stdlib.h>
#include <stdio.h>

int main()
{
    char hello[] = "Hello World!\n";
    printf("%s", hello);
    return EXIT_SUCCESS;
}
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ gcc -o hello hello.c
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$ ./hello
Hello World!
jdumoulin05@cscd-linux01:~/projects/CSCD240S20/Hello$
```

Other things to do

That's the end of the lab. There is no grade, though we will have a quiz on some of these concepts and commands. You can of course continue and compile other programs in the directory and run them to see what they do.

You can also look up the editor `nano`. search for `nano` in your web browser and look for how-to articles on how to use it. We will use `nano` as an editor to to work with our programs as the class progresses.

Try using `nano` to modify the `hello.c` program to say hello to you. Compile and run the program as above to see if your change worked. If you get an error, what does it say? what do you think it means?