CMSC 201

Lab 00 – Hello World

Assignment: Lab 00 – Hello World
Due Date: Sunday, September 11th, 11:59:59pm
Value: 10 points

At UMBC, the GL system is designed to grant students the privileges of running a website, storing files, and even developing applications. GL itself is a Linux-based system that we will interact with using a command line interface (CLI). All major operating systems (Windows, Mac, and Linux) can log into GL.

In order to log into GL, you will need to have a Secure Shell (SSH) client. SSH is a network protocol that connects a client computer to a server securely, without the risk of exposing your password or any other sensitive data to anybody watching the network traffic. It is a leading standard in remote login. So, what does all of that mean? Simply put, SSH allows you to use your home computer to connect to a server (*i.e.*, **`gl.umbc.edu`**) and issue commands as if you were sitting at that computer. So, once you SSH into GL, you can do everything you would be able to do from a computer on-campus. This client is built into Mac and Linux. For Windows, we recommend downloading the open-source client PuTTY.

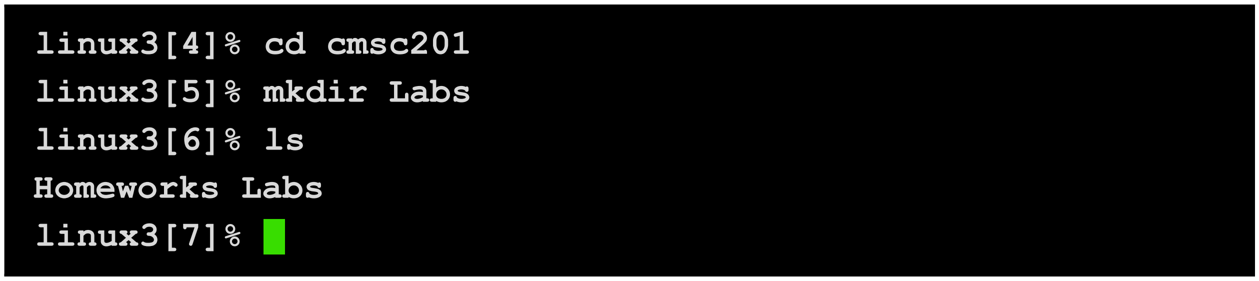
In Lab 0, we will be logging onto GL and setting up folders for cmsc201 in your home directory. We'll also create a simple python program, and turn it in using the **`submit`** command. Finally, you'll send your TA an email with a bit of information about you, so they can get to know you better.

This lab is a way for you to make sure that everything needed to complete the homeworks works for you.

<u>Part 1: Visiting the GL directory once again</u>

<u>Step 1:</u>

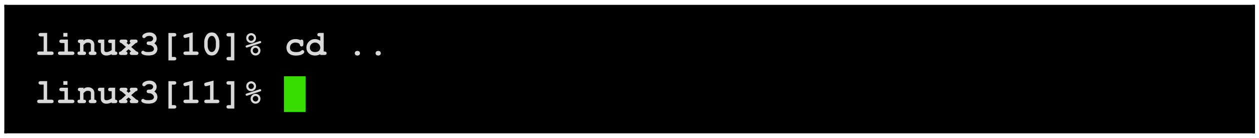
We want to create one directory inside your cmsc201 directory called Labs. For this step, we again use the **`mkdir`** command. Type **`mkdir Labs`**, and hit enter again. Finally, you can type **`ls`** to verify that you now have two directories named “Homeworks” and “Labs” in your cmsc201 folder.



```
linux3[4]% cd cmsc201
linux3[5]% mkdir Labs
linux3[6]% ls
Homeworks Labs
linux3[7]%
```

<u>Step 2:</u>

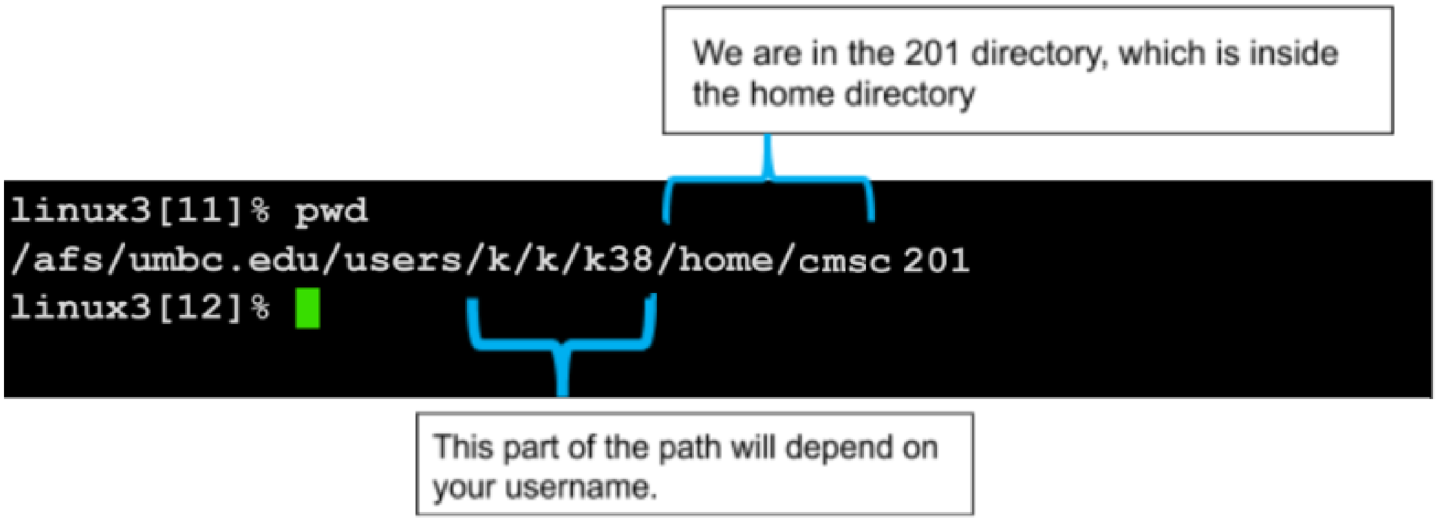
Now we want to create a lab00 subdirectory inside Labs. Before we can do this, we need to move to the Labs directory. First, we go up one “level,” back to the cmsc201 directory. To do this, we type **`cd ..`** (“cd” followed by a space, then two periods). The two periods (`..`) represents the parent directory, or the directory above the current location. The parent directory of Homeworks is cmsc201, so we are now back in the cmsc201 folder.



```
linux3[10]% cd ..
linux3[11]%
```

Step 3:

Let's just double check that we are actually back in the cmsc201 directory. We do this by asking GL to "print working directory," using the **`pwd`** command. Type **`pwd`** and hit enter. It will show you the full path, including "umbc.edu" and your username. If we look at the end, we can see that we are in the cmsc201 directory, just like we wanted to be.



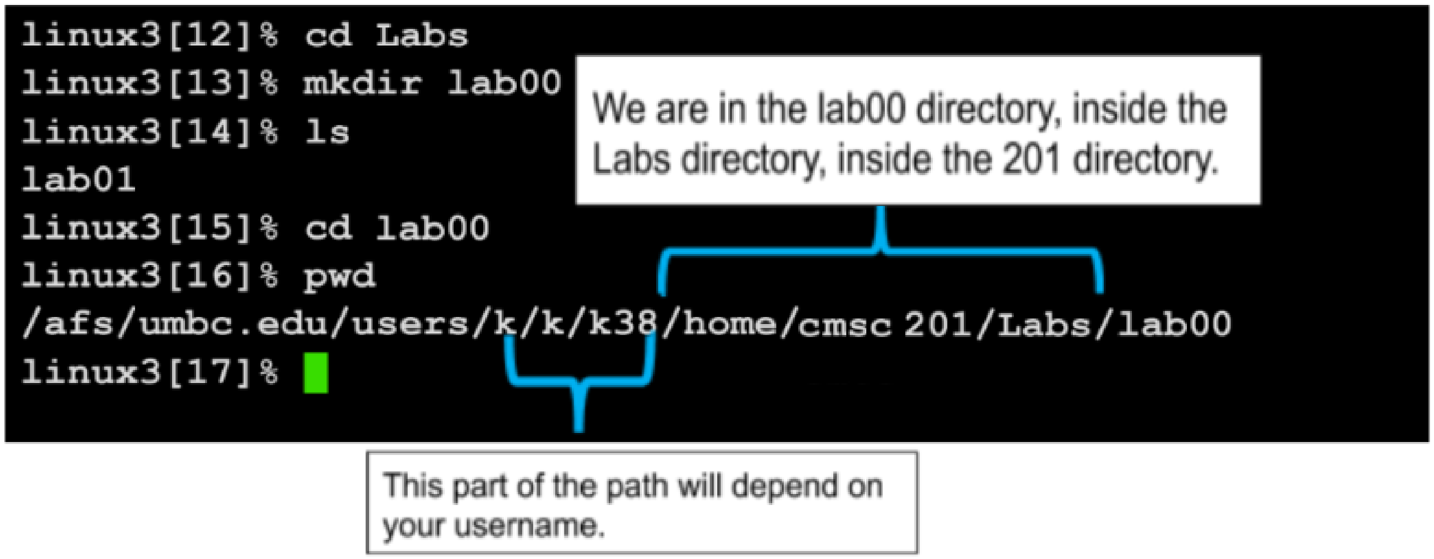
We are in the 201 directory, which is inside the home directory

```
linux3[11]% pwd
/afs/umbc.edu/users/k/k/k38/home/cmsc201
linux3[12]%
```

This part of the path will depend on your username.

Step 4:

Now we can create the lab00 folder inside the Labs directory. First, we go into the Labs directory by typing **`cd Labs`**. Next, we create the lab00 directory by typing **`mkdir lab00`**. We can verify it was created by typing **`ls`** to list the contents of the Labs directory. Once it's created, we can move into the new directory by typing **`cd lab00`**. We can verify that lab00 is now our current directory by typing **`pwd`**.



```
linux3[12]% cd Labs
linux3[13]% mkdir lab00
linux3[14]% ls
lab01
linux3[15]% cd lab00
linux3[16]% pwd
/afs/umbc.edu/users/k/k/k38/home/cmsc201/Labs/lab00
linux3[17]%
```

We are in the lab00 directory, inside the Labs directory, inside the 201 directory.

This part of the path will depend on your username.

Part 2: Creating your first Python program

We are now ready to create our first Python program. Using the emacs editor, we will create a “Hello World” program, the traditional first coding project. (Make sure that when you complete this part of the lab, your “present working directory” is lab00!)

Step 1:

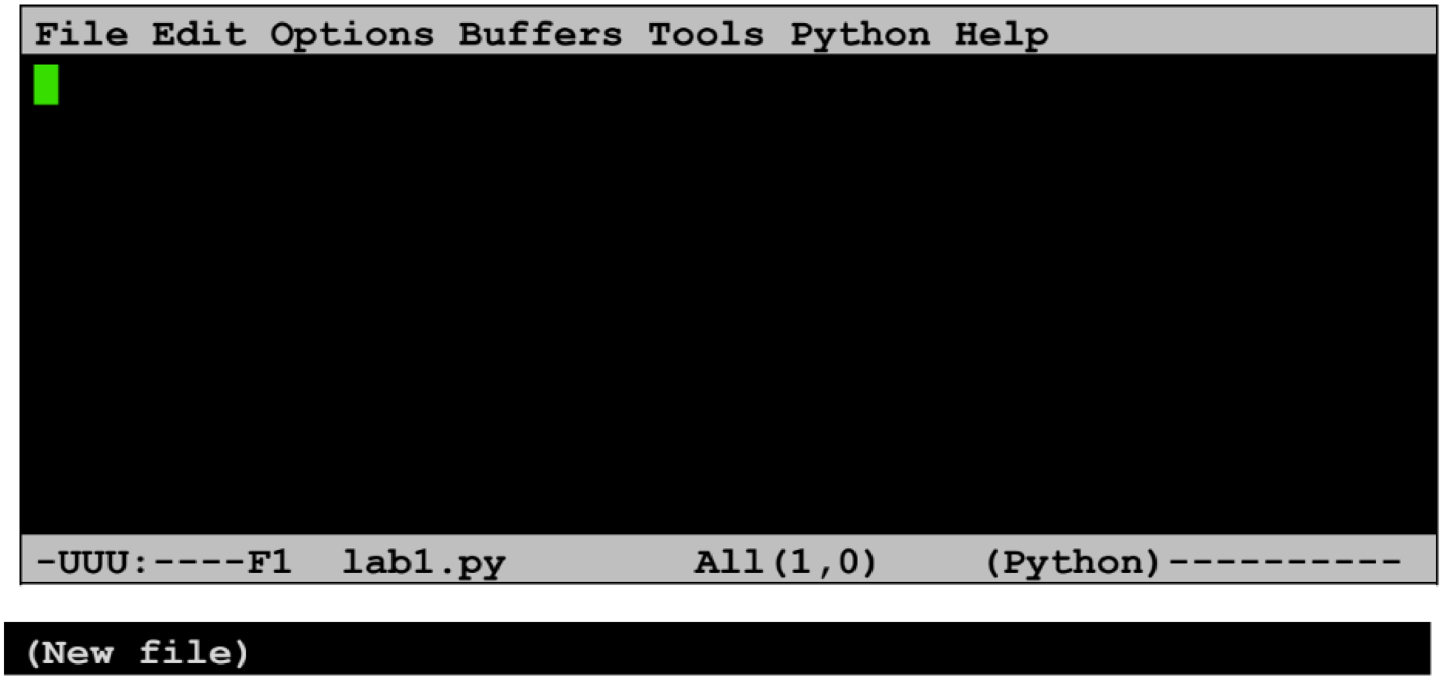
Before we can begin, we need to create the file we will edit. We’ll use the **`touch`** command to create a new (empty) file, by giving it the filename. In this case, we want to create a file called “lab00.py”, so type **`touch lab00.py`** now, then hit enter.

```
linux3[17]% touch lab00.py
```

Now that the file has been created, we can open it for editing, by using the emacs editor. By using the command **`emacs lab00.py`**, we are opening the file lab00.py in the emacs editor. (The emacs editor is similar to Notepad on Windows, or TextEdit on a Mac. It’s nothing fancy – it just lets you type and edit files.)

```
linux3[18]% emacs lab00.py
```

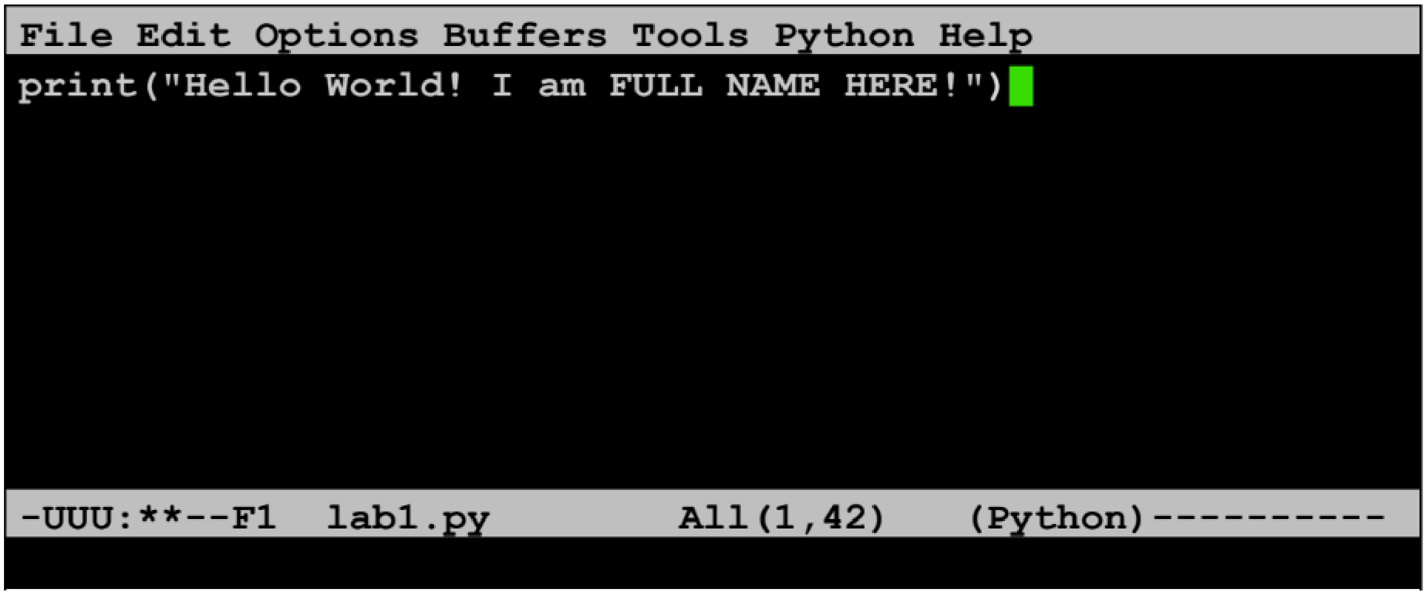
The emacs editor will replace your entire window of text, and should look similar to the example below.



```
-UUU:----F1  lab1.py          All(1,0)     (Python)----------
(New file)
```

Step 2:

Let's code our first program! This program will print out a short greeting when we run it. Type in the line of code shown below, but where the code says **`FULL NAME HERE`**, replace it with your full name.



```
print("Hello World! I am FULL NAME HERE!")
```

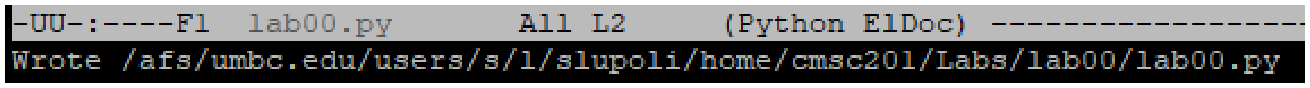
```
-UUU:**--F1   lab1.py          All(1,42)    (Python)----------
```

It's important that you type in the code exactly as shown. Every character is important, and even small changes can keep the program from working. The only change should be replacing **`FULL NAME HERE`** with your full name.

Step 3:

Once you are done typing your code into emacs (and have replaced **`FULL NAME HERE`** with your name), you need to save your work and exit. First, save it by pressing **`CTRL+X`** and **`CTRL+S`** if you have Windows. (Press the CTRL and X keys at the same time, then the CTRL and S keys at the same time. You do not need to hold the CTRL key down for the entire command.) If you have a Mac, use the **`Control`** key instead of the **`CTRL`** key since Macs do not have this key.

If it worked, you should see a confirmation message at the bottom of your screen, similar to the one below.



```
-UU-:----F1  lab00.py        All L2      (Python ElDoc) ------------------
Wrote /afs/umbc.edu/users/s/l/slupoli/home/cmsc201/Labs/lab00/lab00.py
```

Now that you've saved your code, you can exit by pressing **`CTRL+X`** and **`CTRL+C`**. You will exit out of emacs, and be back to the command line.

Part 3: Running your first Python program

Step 1:

To run your code, type in **`python3 lab00.py`** and hit enter. Your program should run as seen below, but with your name instead of **`FULL NAME HERE`**.

```
linux3[0] python3 lab00.py
Hello World! I am FULL NAME HERE!
linux3[0] █
```

If your program does not run, but instead gives you an error message, you have at least one error. Just open the file up in emacs again (type **`emacs lab00.py`** after the **`linux3[0]`** prompt) and take a close look at your code. Make sure it matches the code from Step 11. For example, here's the error if the quotes are outside of the parentheses:

```
linux3[0] python3 lab00.py
  File "lab00.py", line 1
    print"(Hello World! I am FULL NAME HERE!)"
                                            ^

SyntaxError: invalid syntax
linux3[0] █
```

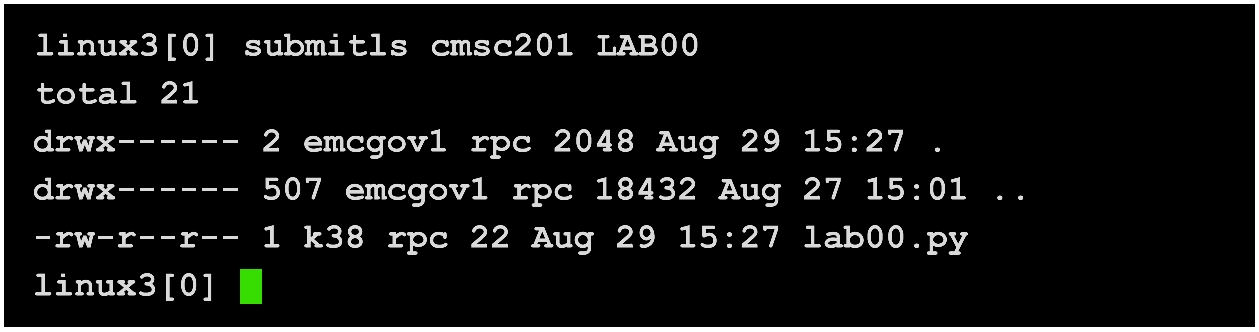
Part 4: Submitting your first Python program

Step 1:

Watch this video (https://www.youtube.com/watch?v=r0otsJZ1ry0) for an explanation of how GL works. Follow the instructions in the video to submit your lab00.py file.

Step 2:

After submitting the file (by following the instructions in the video from the previous step), we can verify that it went through by using the **`submitls`** command. The command needs to know the name of the class and the name of the assignment – these should be the same as for the **`submit`** command you just ran: cs201 for the class, and LAB0 for the assignment.

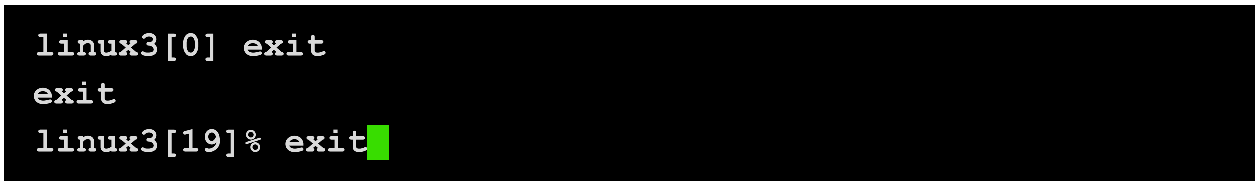


```
linux3[0] submitls cmsc201 LAB00
total 21
drwx------ 2 emcgov1 rpc 2048 Aug 29 15:27 .
drwx------ 507 emcgov1 rpc 18432 Aug 27 15:01 ..
-rw-r--r-- 1 k38 rpc 22 Aug 29 15:27 lab00.py
linux3[0]
```

On the far right of the last line, we can see that the file lab00.py was submitted successfully. The other two shown ("`.`" and "`..`") are not important for now.

Step 3:

To exit out of bash, and back to the linux prompt, type **`exit`** and hit enter. To exit from GL, type the **`exit`** command again and hit enter.



```
linux3[0] exit
exit
linux3[19]% exit
```

This will log you out of GL and close your terminal window.

Part 5: Emailing your TA

Step 1:

Use your @umbc.edu email account to send your TA an email, with a subject line of "CMSC 201 Intro -- Section ## -- YOUR_UMBC_ID". Inside, your email should contain the following information.

My preferred name (or nickname): FILL_OUT_HERE
My pronouns: FILL_OUT_HERE
My major: FILL_OUT _HERE
My programming experience: FILL_OUT _HERE
Fun fact about me: FILL_OUT _HERE

For more information about "my pronouns" and what that means, click here. For your programming experience, it's fine if you simply put down "none."
You can find out who your TA is under the "Sections" page on the course website. Match the section number from your CMSC 201 discussion section (you can find it on your schedule) to the section number in the table. Make sure that the time and place also match the discussion in your schedule!

Part 6: Wrapping Up

Step 1:

Watch this video (https://www.youtube.com/watch?v=ci42iOOtTRI) for an explanation of the "life cycle" of remote programming on GL.

Step 2:

Finally, let's discuss how the labs will be graded in this course.

There will be 13 labs over the course of the semester; your best 10 scores will be used to compute your lab average. To receive credit for attending the lab and completing the assignment, your work must be verified by your TA.

You must attend your assigned lab section and the lab assignment must be completed **during** the assigned lab time (not before).

During the lab, your TA will explain the lab assignment, provide assistance as needed, and record your successful completion of the assignment.

Lab assignments are graded on a scale from 0 to 10 at the discretion of the TA.

Score	Description
8 - 10	Showed up and almost to fully completed the assignment
4 - 7	Showed up and attempted the work
1 - 3	Showed up and barely attempted the work
0	Did not show up

Labs are worth a total of 10 points each.

Special thanks to James Gerity who wrote a GL primer on his website:
http://userpages.umbc.edu/~jg5/logon-howto.html
Mac logon instructions based on a guide from Rackspace:
https://support.rackspace.com/how-to/connecting-to-linux-from-mac-os-x-by-using-terminal/