CLI (Command Line Interface) and First Assignment Tutorial

Command Line Interface and First Assignment Tutorial

Intro: This tutorial is made to assist those who little to no experience using a command line interface (see following paragraph for description of what this is). It will consist of several parts addressing the following topics:

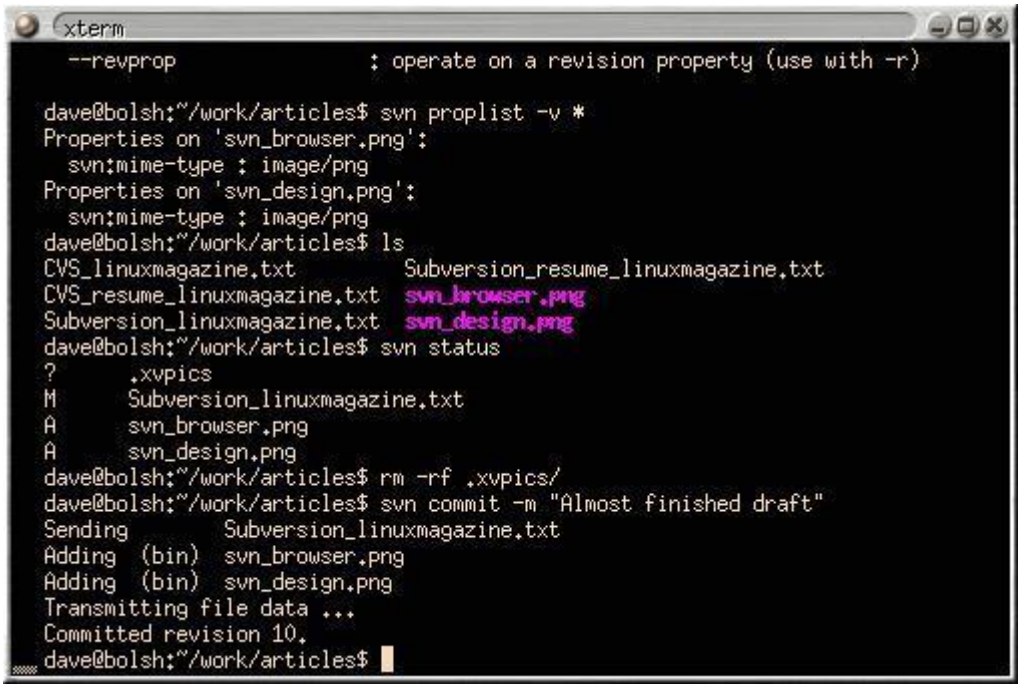
1. **What a command line interface (CLI) is**
2. **How to access the LSU server**
3. **Some basic navigation techniques in a CLI/the LSU server**
4. **How to compile and submit on/to a server (In this case LSU's)**
5. **Answers to some frequently asked questions**

What is a CLI?

To best understand how to navigate and use the servers at LSU, a person has to understand the difference between a GUI (graphical user interface) and a CLI (command line interface). A GUI is an interface in which the user interacts with easy to understand buttons, icons, text boxes, etc. Windows and most applications that run on Windows are GUIs.

GUI examples

In contrast to these pretty GUIs, we have a CLI (command line interface). To use a CLI, the user has to type in lines of text called "command lines" to perform a task. There are several different CLIs but the one we will be dealing with specifically is Linux (the interface used by LSU).



CLI example

Why Use CLI? although CLI may be more difficult to use then a GUI, it takes significantly less memory on the server (improving performance for each user) and can be substantially more powerful in certain instances.

Logging Into and Using the LSU Servers

Step 1:

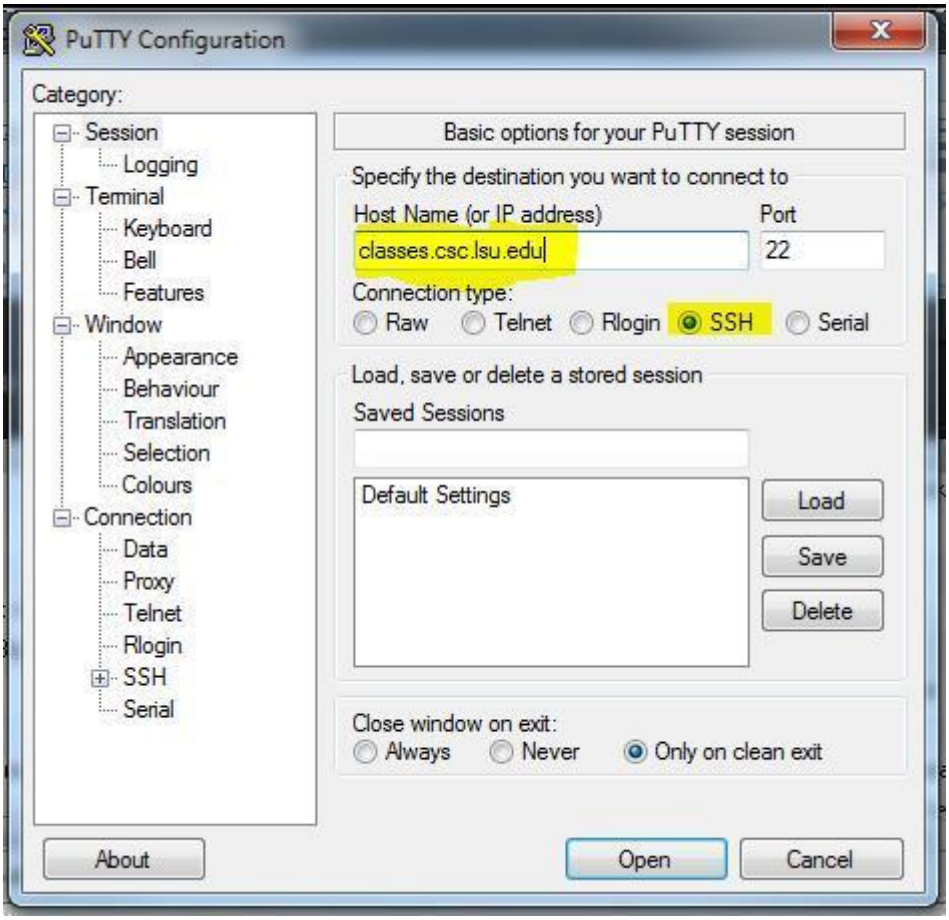
For Windows: If you are running Windows and have not done so already, you will need to download an SSH client to access the servers. A popular SSH client called PuTTy can be downloaded at this link:

http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

For OSX/Linux: If you are using a OSX (the mac operating system) or linux, you already have an SSH client by default. To access it, go to spotlight (the little magnifying glass in the top-right of your screen) and type in "terminal" (without the parentheses) in the search bar. Open terminal and proceed to the OSX/linux portion of step 2 for further instructions.

Step 2:

For Windows: Open up Putty and type in classes.csc.lsu.edu using port 22. make sure that the SSH radio button is checked. Your client should match the one shown below when you finish. Click "Open" when you are done.



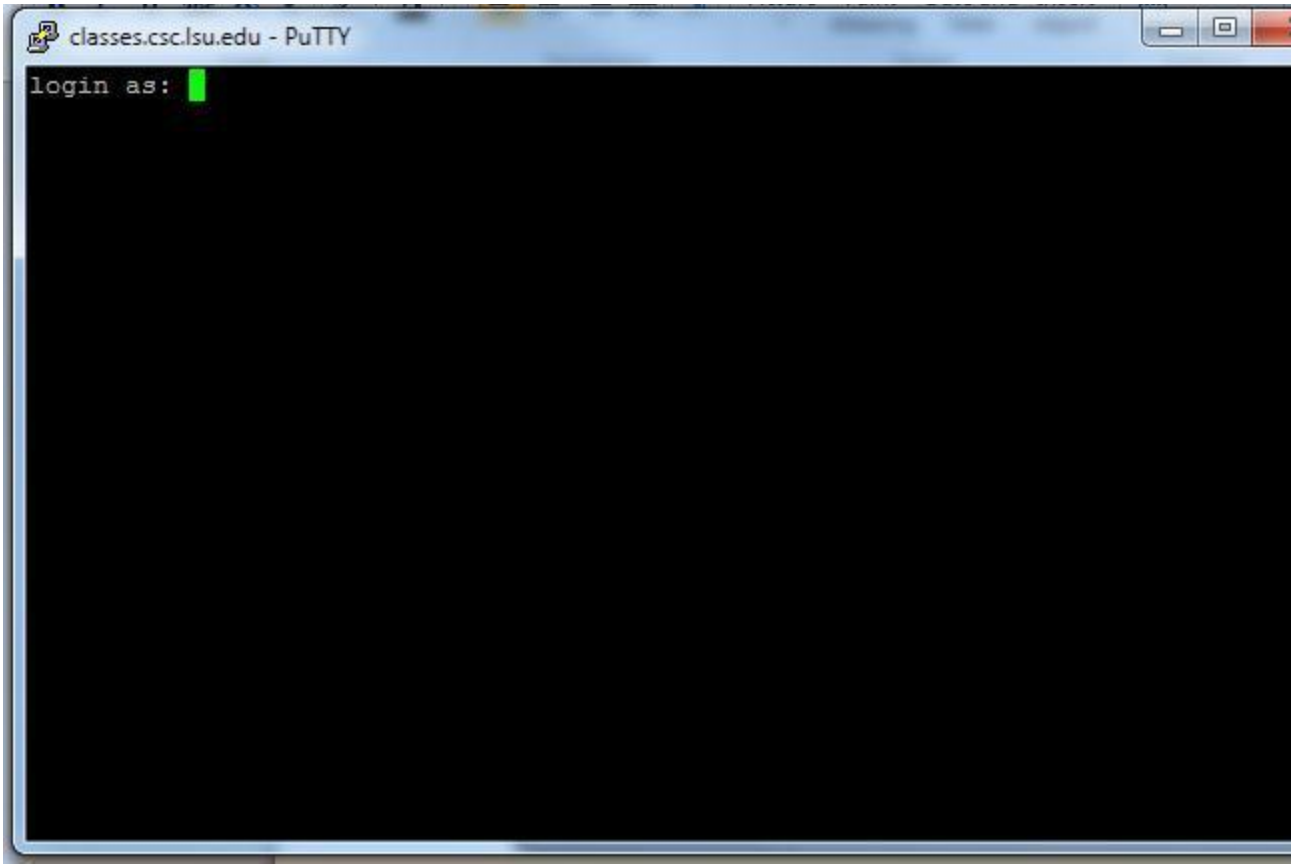
For OSX/Linux: in the terminal type in the following:

```
ssh -l (your username) classes.csc.lsu.edu
```

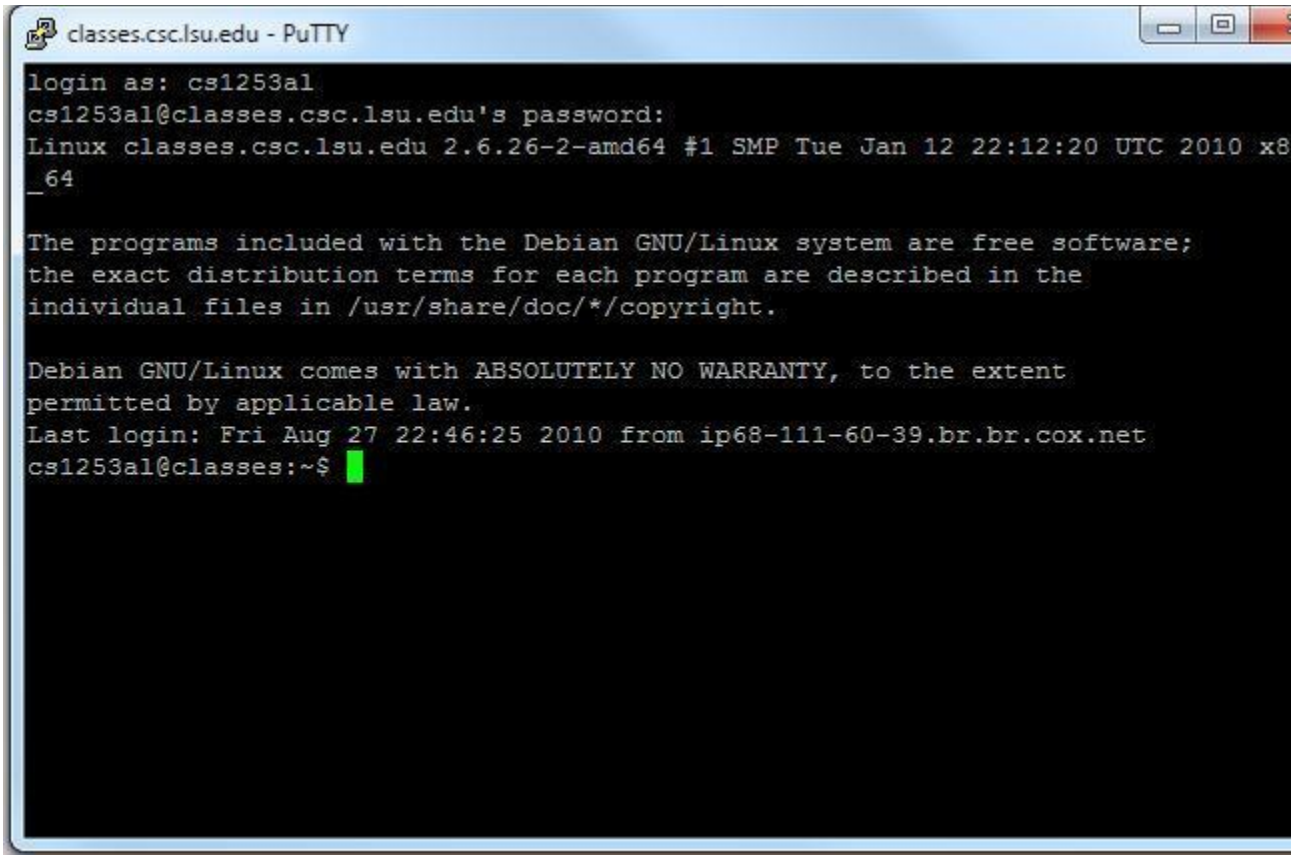
in place of "your username", type the username provided to you earlier this week. hit enter.

Step 3:

For Windows: You should now see a screen resembling the one below



In this window, type in your username provided to you earlier this week. After doing this, the PuTTY client should ask for a password. Once it does, enter the password provided to you earlier this week. If you did everything correctly, you should see a screen resembling the one below.



For OSX/Linux: You should now see a prompt to enter your password. Enter the password provided to you earlier this week. If you have done everything correctly up to this point, you should see a window similar to the one above (except with black text and a white backdrop).

Using/Navigating the Server

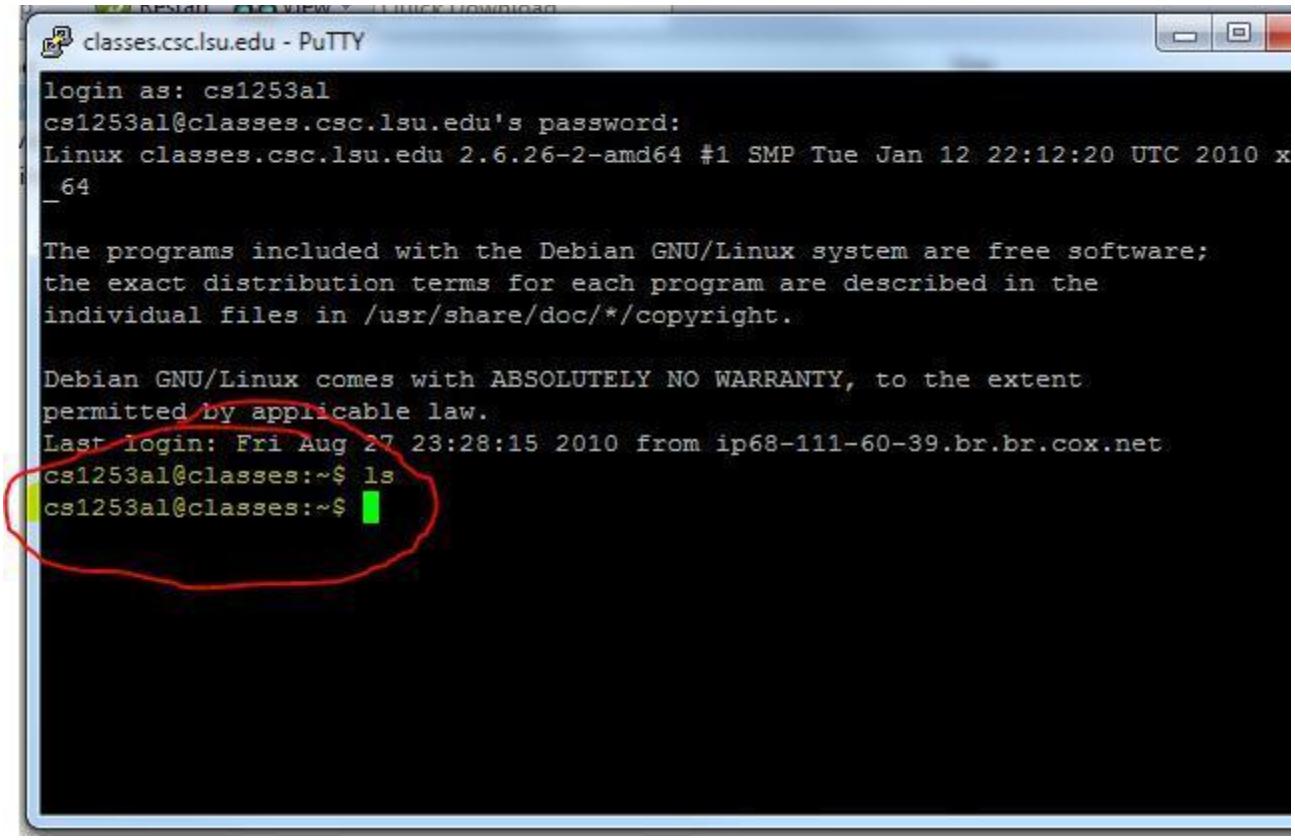
Now that you are logged in to the server, we can begin learning to navigate it properly. The navigation will work the same for Windows/OSX/Linux as we are now using the LSU server and NOT using our own computers (to a certain extent). When you logged in to the server, you were put into your "home" folder. the server is made up of several levels or "tiers" of folders. within each folder is either files, more folders, or both. This can sound confusing at first, but it becomes clearer as you learn to navigate the server. At this point, we will begin learning how to navigate. Let's start with three basic navigation commands

"ls" -This command shows all folders and files that are contained within the folder you are in.

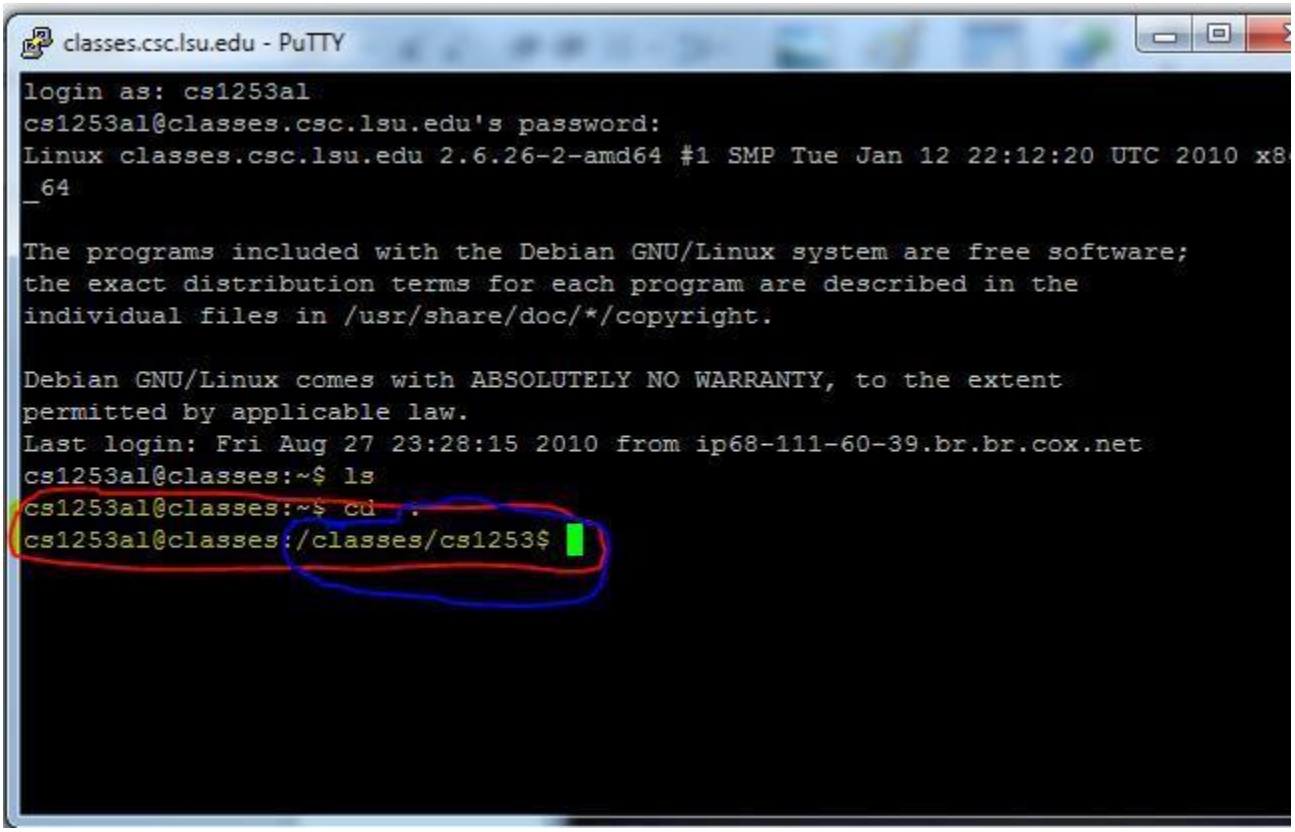
"cd (file/folder name)" -This command enters the file or folder designated. replace "(file/folder name)" with the file or folder you wish to enter.

"cd .." -This command Brings you up a level or "tier" of folders

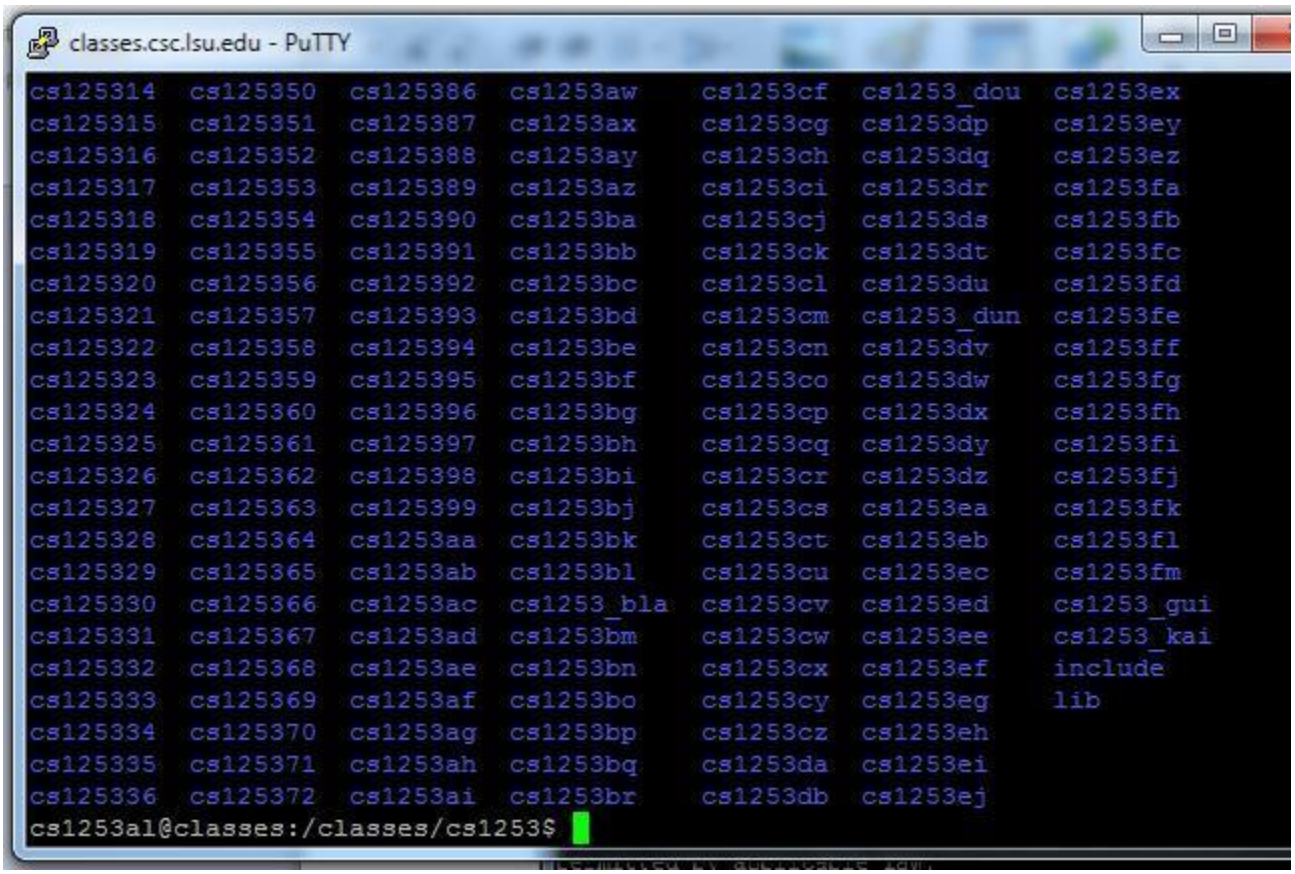
Now that we have these basic commands, lets play around with them a little. Using window from the "Logging Into and Using the LSU Servers" section, try typing in the command "ls" (without the parentheses)



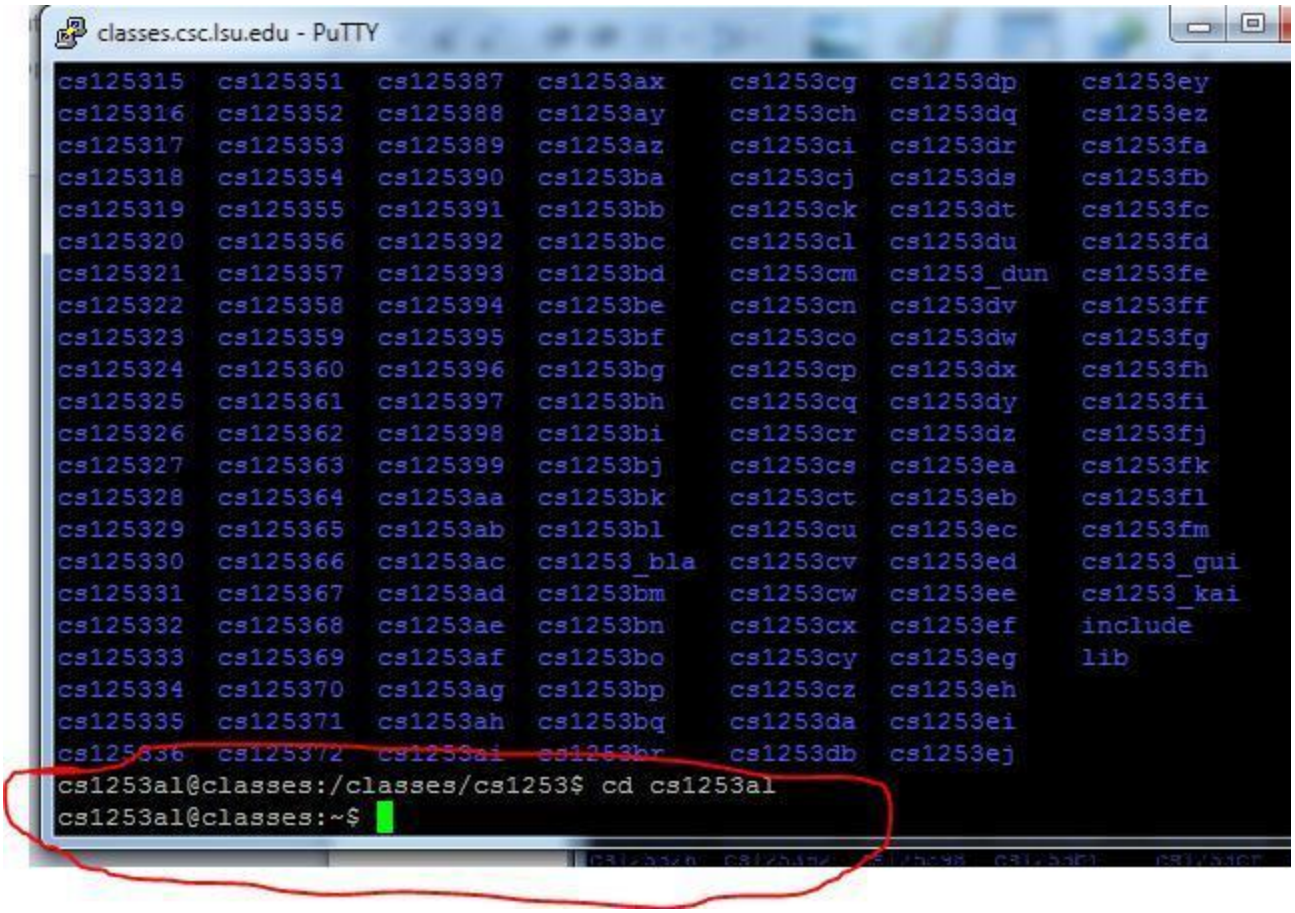
You probably noticed that nothing happened. This is because there are currently no files or folders within your account. Let's see what happens if we type in the "cd .." command. (remember that the "cd .." command brings us up a level of folders)



After you types in "cd ..", a new line should have appeared. This part is circled in blue in the picture above. these are an indication of which folder we are now located in. Now that we have gone up a level of folders, try typing in "ls" again.



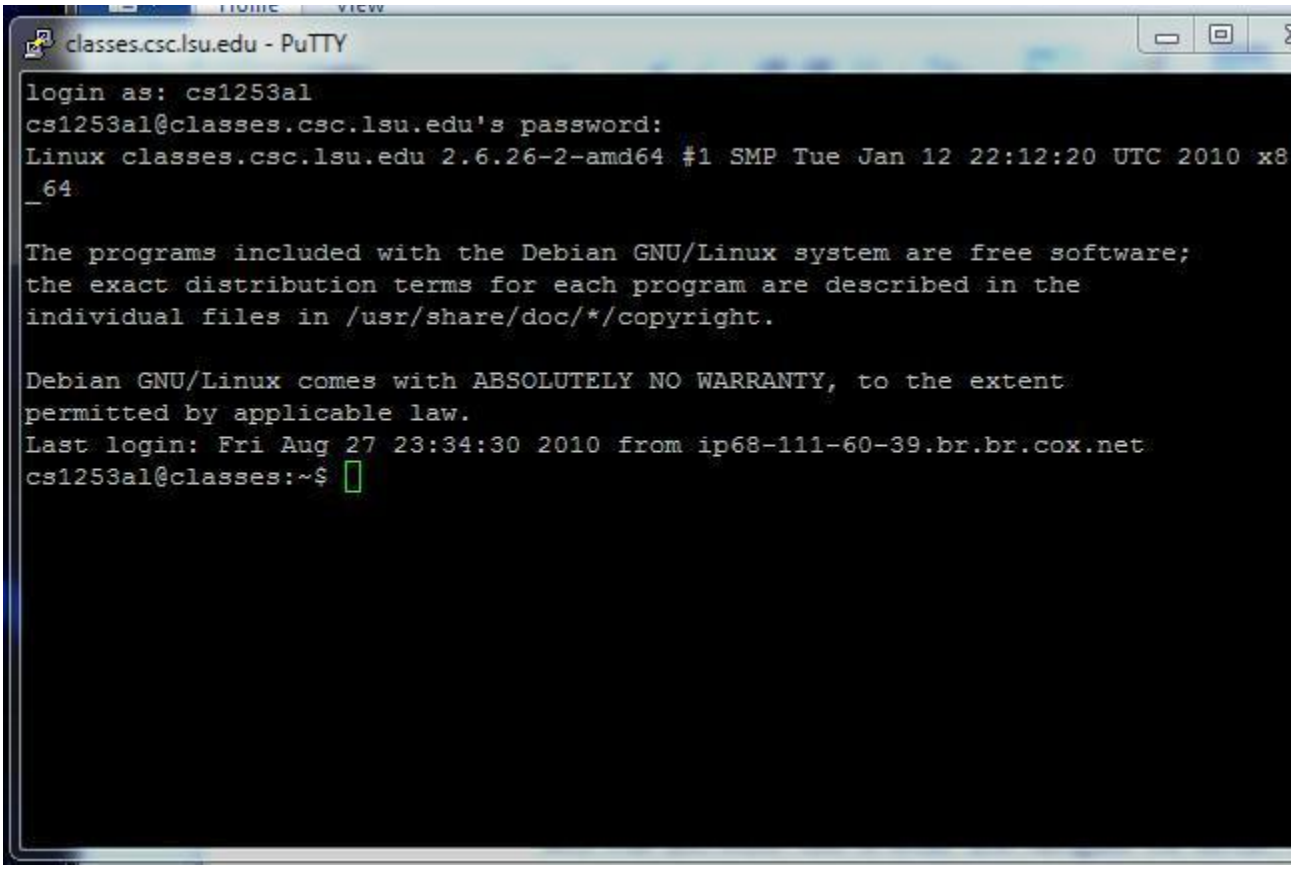
We can now see the folders that are contained within the folder we are now navigating. To enter another folder/file, all we need to do now is type in "cd (file/folder name)". Try entering our home folder again (the folder name is the same as your user name).



to get a list of basic commands at any time, simply type "help"

How to Create/Compile/Save/Modify a Program (The First Assignment)

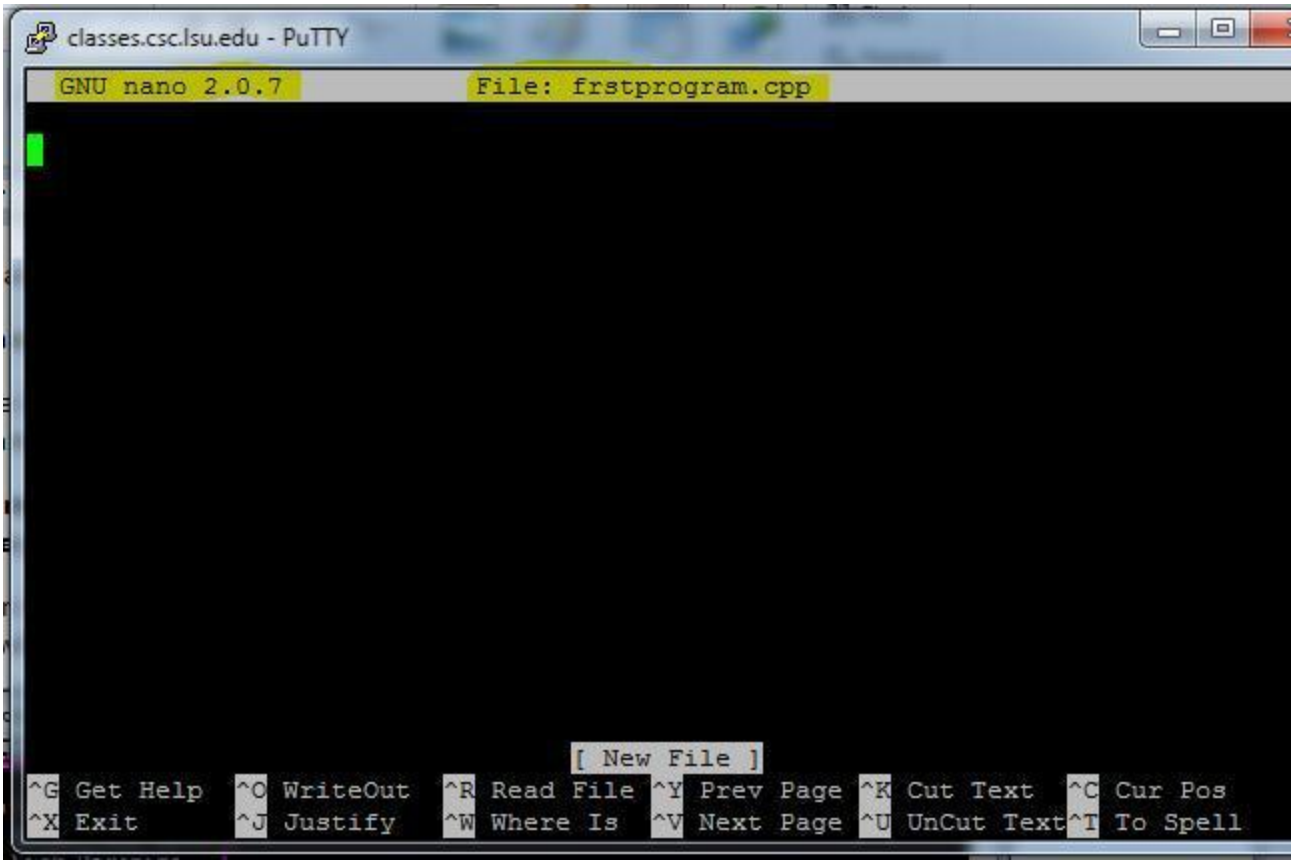
Now that you know how to enter and navigate the server, its time to learn how to create, compile, save, and modify a C++ program. To begin, lets navigate back to our home folders.



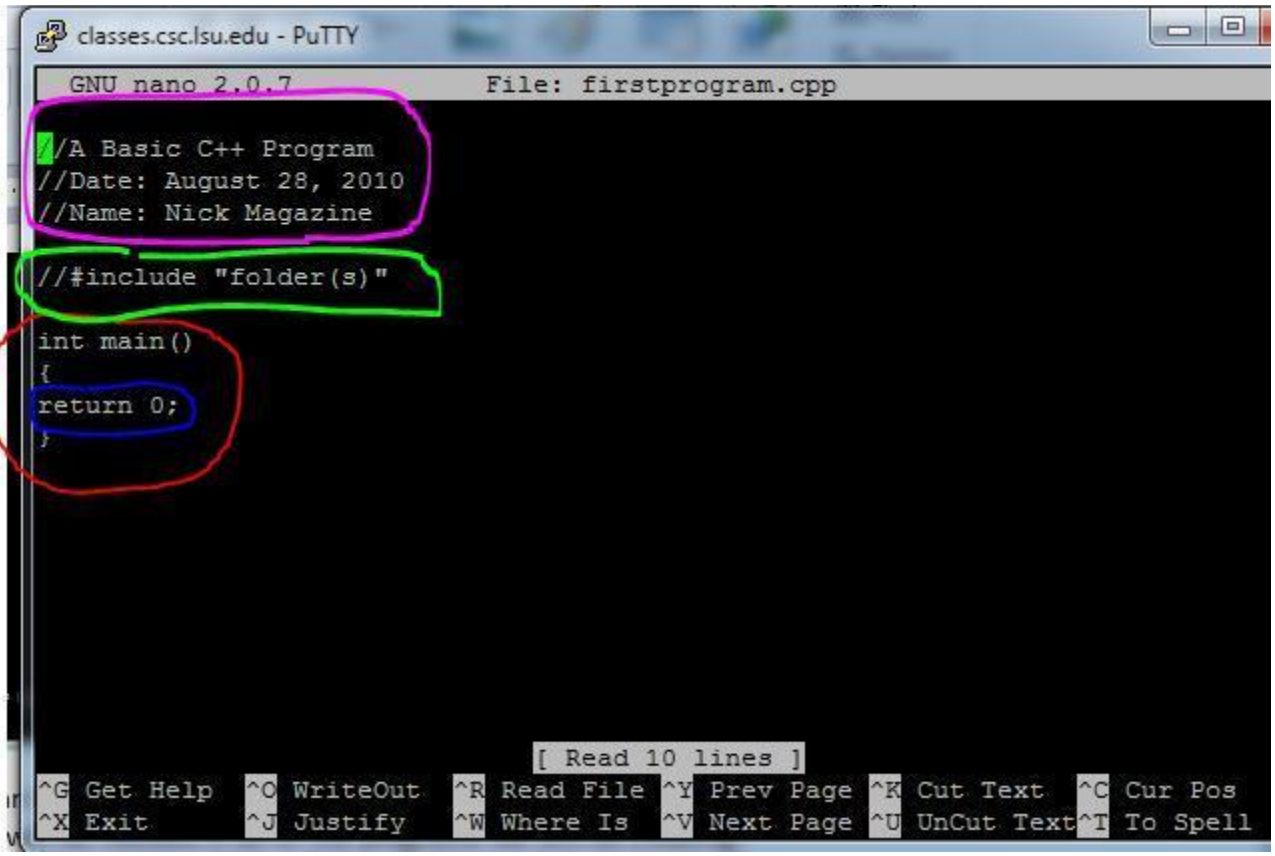
Once you are in your home folder, you will need to open a text editor to begin coding. There should be a basic, easy to use text editor already built in to the server. This editor is called "pico" and can be evoked by typing:

```
pico (file name)
```

Replace the (file name) with what you want to name your program. I called mine "firstprogram.cpp". After hitting enter, you should see a screen like the one below.



Note: you must include the .cpp extension after every program name! If you do not do this, the program will not compile later! Now we can begin coding. Lets begin by typing the basic elements that (for practical purposes) will be needed in all of the programs we will be creating.



In pink: A header. It is a good habit to put a header at the start of every program. Remember that "//" denotes a comment. Comments are lines of text that are not read by the computer and therefore do not effect the code.

In green: The folders that will be imported. By importing certain folders, we gain access to more C++ functions. This can often save us the time of writing many lines of code to accomplish a redundant task. Because the program we will be creating is basic, this line will not be needed (this time anyway).

In red: The body and code container. Within this area is where we will be typing our code.

In blue: The return type. In this field we put a return value. Return values can be used in several different ways. In this case, the return value denotes whether or not the code executed properly. Returning 0 indicates to the computer that the code was executed without any errors. At this point, everything is set up and ready for us to begin typing our body of code. You can enter whatever you would like. Just make sure that your code follows the proper format. For my program, I made a simple currency converter. The converter will change Euros (€) to USD ($). This is the code for the Euro to USD converter. When you make your program, try to integrate a couple of simple commands such as "cin and "cout". Reading and writing actual code is the best and usually easiest way to learn.

```
//A Basic C++ Program
//Date: August 28, 2010
//Name: Nick Magazine

#include "../include/std_lib_facilities.h"

int main()
{
   double var_euros = 0;
   double var_usd = 0;

   cout << "This Program converts Euros to USD.\n";

   do
   {
      cout << "Please enter the number of Euros: ";
      cin >> var_euros;
      var_usd = var_euros * 1.31;
      cout << var_usd << "USD\n";
      var_euros = 0;
   } while (var_euros = 0);
   return 0;
}
```

Now that you have typed your code, lets compile it. To save your work in pico, hit control+x

```
      var_euros = 0;
   } while (var_euros = 0);
   return 0;
}

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

When pico asks if you would like to save, hit y to say yes.

```
cs1253a1@classes:~$ firstprogram
This Program converts Euros to USD.
Please enter the number of Euros: 8
10.48USD
```

At this point, pico may or may not ask you if you want to change the files name. you can if you want, but you don't have to.