# SSH Guidelines

The purpose of this document is to walk through how to remotely log in to the CS Lab machines to test your code. The TA's will run tests on your code on the lab machines. By doing this, you do not need to physically be at a lab machine in the GDC to test your code. You will need to do this in your future classes, like CS 429 and CS439, so feel free to get ahead now! I will talk about how to SSH via Mac and Windows. I'll also go through some essential commands needed to navigate the Linux terminal, and how to compile and test your code remotely. The compiling and testing will work if you are physically at the lab machines and want to run your code that way.

***A short version of these instructions is at the bottom of this document**

## Before starting this:

- You will need a UTCS Account for this to work. To request an account, visit this link.
- You must physically log into a lab machine to set up your account properly.
- Windows machines and Mac's have different instructions in some spots. Make sure you are following the correct set of instructions
- Note that if you put files on one lab machine, they will show up on any lab machine you log into.

## Terminology

Throughout this document, I will use words inside brackets like this <>. If there's any text inside this bracket, you are meant to type in your own personal information. For example <CS_username> means type in your specific CS username.

GUI- graphical user interface. Similar to File Explorer on Windows and Finder on Mac. They make things look prettier than on a terminal

### What's SSH and why would we use it?

SSH stands for Secure Shell. It's a secure way for you to remotely log into the CS machines from your laptop. This is meant for use if you have your code on your laptop/computer, but would like to compile and run it on the lab machines for testing.

When SSHing into a lab machine, you cannot use Eclipse or BlueJay to edit your files!!! If you wish to edit your files via SSH, there are some IDE's that will allow this on Mac, or you can use a text editor like vim or emacs (there are many others).

When we grade, we use the lab machines. If your code compiles on your computer, but not on the lab machines, you will receive a zero for correctness. We're giving you the tools to check your work. Use them.

### Computer Setup and Login

Mac:

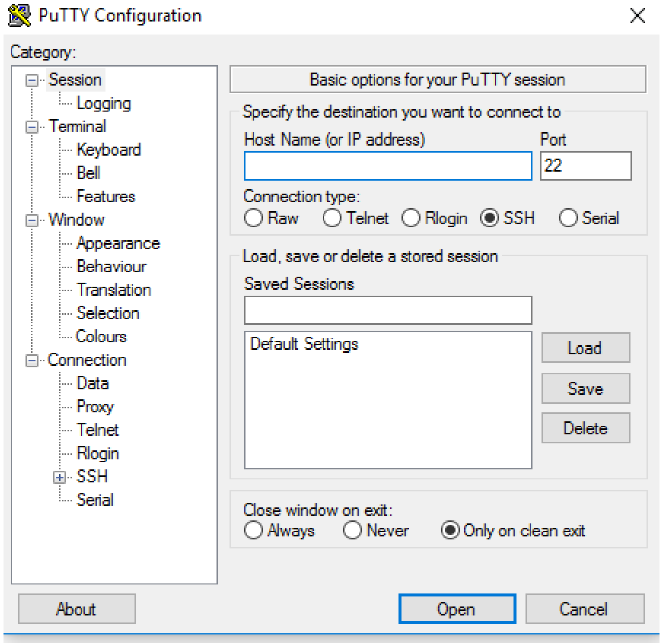
There are two options for Mac—you can either use the terminal, or you can install Putty. To use the terminal, follow these instructions. **To use Putty, look at the Windows instructions**.

1. When you open up the terminal, you'll see a prompt that looks something along the lines of: <myComputerName>:
2. type in "ssh <CS_username>@<cs_lab_machine>.cs.utexas.edu"
    a. A list of all the CS Lab machines and their names can be found here:
    b. We recommend you don't use linux.cs.utexas.edu or skipper.cs.utexas.edu because those tend to have a very high load.
3. Press enter.
4. This should prompt you for your password. Type that in.
5. You're successfully logged into the lab machine!

Windows:

There are some SSH clients out there to help with SSHing into different computers—I recommend Putty. Download it here and install it on your machine

1. When you pull up Putty, it looks something like this:



Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

2. Type the name of a lab machine in the Host Name box. Click "Open".
    a. A list of all the CS Lab machines and their names can be found here:
    b. We recommend you don't use linux.cs.utexas.edu or skipper.cs.utexas.edu because those tend to have a very high load.
3. Type in your CS Username. Press Enter.
6. Another option to login is to type in CSLOGIN@cs_lab_machine.cs.utexas.edu into the hostname at the beginning
4. Type in your CS password. Press Enter.
5. You're SSH'd into your lab machine!

**Navigating Linux**

We're going to go through a few useful commands for you to get around a Linux machine and compile/run your code.

1. First, type "ls"
    a. This lists all of the folders and files that are in your home directory (folders are called directories).
2. To go into one a directory, type "cd <directory name>"
    a. Suppose your directory name is "CS312_Project1". You may not want to type that out each time you want to access anything in the directory. Linux has a great auto complete feature to help with this. If you type in "cd CS" and hit the Tab key, it autocompletes the folder name for you.
    b. Suppose you had another directory in the same workspace called "CS314_Project1". You can still use autocomplete. If you type in "cd CS" and hit the tab key twice, the terminal displays all the directories/files that start with those letters. If you want to access "CS314_Project1", simply hit the up arrow key. This will pull the last instruction you typed (and you can keep going up or down to hit different commands that you've typed). Fill in 314 and autocomplete with a tab.
3. Suppose you want to make a new directory to put all of your java files in. Type "mkdir <directory_name>". This makes the directory.
4. If you want to remove this directory, type "rm <directory_name>". Note that if there are any files in this directory, you'll get an error. You can either go in and remove these files one by one, or you can recursively remove all the files in the directory using "rm -r <directory_name>". Now, if you have other directories inside this directory, google may tell you to "rm -rf <directory_name>". DO NOT DO THIS UNLESS YOU ARE 100% SURE YOU KNOW WHAT YOU ARE DOING. This will completely wipe all the contents of the directory without asking you to confirm. Companies have deleted part of the internet/their entire company because of this (no I'm not kidding…. Here's a link if you're interested in stupid things people do:).
5. Now, we need to put our java files from our computer onto the lab machines
    a. Mac
        i. Option A: GUI (graphical user interface)
            1. Download FileZilla here. You can also use another file transfer client compatible with Macs. Note: this is a tool I have not tested or used.
            2. Install FileZilla.
            3. When you open the application, you'll see several fields at the top
            4. For the hostname, type in <cs_machine>.cs.utexas.edu
            5. For username, type in your cs login id
            6. For password, type in your cs password
            7. For the port, type in 22
            8. The left side is the files on your machine, the right shows files on the Lab machine
            9. Navigate to where your file is on your computer (the left side)
            10. Navigate to the folder you want to put it on the lab machine on the right side (like Desktop)

ii. Option B: Terminal

1. This is a little tricky because you have to actually type commands into the terminal to make this work. Mac is build on top of a version of linux, so when we type in commands into the Mac terminal, they are very similar to what we use on the Linux terminal when we are ssh'd in.
2. Open a 2nd terminal (do not ssh into the lab machine)
3. Using the "ls" and "cd" linux commands, navigate to where your java file is on your computer.
4. Type in "scp <filename> <cs_username>@<cs_lab_machine>.cs.utexas.edu:<path to file>"
   a. If I want to place a Song.java file in my Desktop folder on the lab machine, I would type: "ssh Song.java <cs_login>@<cs_lab_machine>.cs.utexas.edu:~/Desktop"
   b. If you want to copy all java files from this location, type : "ssh *.java <cs_login>@<cs_lab_machine>.cs.utexas.edu:~/Desktop"
5. Go back to the terminal that is ssh'd in to the lab machine. Navigate to the folder you placed your java file in using "cd". If you type in "ls" where your file should be, the terminal will tell you that the file is in the correct spot.
6. To copy a file from the lab machine to your computer. The syntax is similar.

b. Windows:

i. Download WinSCP here

ii. Log in to the lab machine like you did with putty. The hostname is the <cs_lab_machine>, and the username and password correspond to your CS login information. The left GUI (graphical user interface) will show the files on your machine, and the right will show files on the Lab machine.

iii. Drag and drop the files from your computer to the lab machine.

iv. Now your files are on the lab machine!

6. Now, go back to the terminal (Putty if you were on windows). Navigate inside the folder with your java file.

a. Type in "javac <filename>.java"

i. For example, if I'm compiling a file called Song.java, I would type in "javac Song.java". Note that this is case sensitive. "javac song.java" will not work.

ii. This compiles the java file into byte code that the computer knows how to read.

iii. If you have any issues, the terminal will give you back some information regarding your compile error.

iv. If you have 10 files that you need to compile, you can type in "javac *.java". The * is like a wildcard. This tells the computer to compile all the files that end with java (saving you a lot of tedious work).

b. Now to run the program. Type in "java <filename>". You don't need the .java extension here because you're running the executable file generated when you compiled the code.

i. If I'm running my Song file, I would type in "java Song".

**Short Instructions**

Shortened Instructions for Mac

1. Open up a terminal
2. ssh into a lab machine with "ssh <CS_username>@<cs_lab_machine>.cs.utexas.edu"
3. use "cd" to navigate to where you want to put your java file(s).
4. Option 1: GUI
    a. Open up FileZilla or another file transfer client.
    b. Log in with your hostname, cs username and password, and port 22
    c. On the left of FileZilla, use the GUI to navigate to where your java file(s) are located on your machine.
    d. On the right of FileZilla, use the GUI to navigate to where you want to put your java file(s) on the lab machine. (should be the same spot you cd'd into on Putty).
    e. Drag the java file(s) from the left into the location on the right (lab machine)
5. Option 2: Terminal:
    a. Open another terminal that's not ssh'd in
    b. Use cd to navigate to where your java file(s) are.
    c. Type in ""scp <filename> <cs_username>@<cs_lab_machine>.cs.utexas.edu:<path to file>". More options listed above.
6. Go back to the terminal that is ssh'd in.
7. Type in "ls". All of the file(s) you copied over should show up.
8. For a single file, type in "javac <filename>.java". For multiple, type in "javac *.java".
9. Run your code. Type in "java <filename with main method>"

Shortened Instructions for Windows:

1. Open Putty.
2. Log into putty with your username, password, and a cs lab machine
3. Use "cd" to navigate to where you want to put your java file(s).
4. Open WinSCP.
5. Log in to WinSCP with your cs username and password, and a cs lab machine
6. On the left of WinSCP, use the GUI to navigate to where your java file(s) are located on your machine.
7. On the right of WinSCP, use the GUI to navigate to where you want to put your java file(s) on the lab machine. (should be the same spot you cd'd into on Putty).
8. Drag the java file(s) from the left into the location on the right (lab machine)
9. Go back to Putty.
10. Type in "ls". All of the file(s) you copied over should show up.
11. For a single file, type in "javac <filename>.java". For multiple, type in "javac *.java".
12. Run your code. Type in "java <filename with main method>"

These are the basic techniques and setup to get your code up and running on the lab machines. The other option, if you don't want to go through all this, is to physically go to a lab machine and log in. You can either email yourself the java files or put them on a USB stick (or some other form of transfer). Once you have the files on the lab machine, run the "javac <filename>.java" and "javac <filename>" commands.