

# Department of Electrical and Computer Engineering

## The University of Texas at Austin

EE 460N, Fall 2016

Lab Assignment 0

Due: Sunday, August 28 ~~28~~ 30, 2016, 11:59 pm

Yale Patt, Instructor

Siavash Zangeneh, Ali Fakhrzadehgan, Steven Flolid, and Matthew Normyle, TAs

Expected time to complete lab: 10 minutes

## Introduction

The purpose of this lab is to make sure that you have a working ECE LRC account. This lab has two parts. Part A is intended to help students who have no prior linux experience and will not be worth any credit. If you feel comfortable with the linux command line then feel free to skip this part. Part B is intended to make sure you have a functional ECE LRC account and understand how to submit, test, and grade files on our system. Successful completion of part B will earn you 10 extra points on your lowest lab grade. **This is individual work. Only those who submit on their user accounts will receive credit.**

## ECE LRC Accounts

To get an ECE LRC account visit this link:

<https://utdirect.utexas.edu/apps/ece/db/acme/>

Log in using your normal UT EID and password, then click on “Click here to create or manage your account.” If it’s your first time creating an account, a usage agreement may come up, and you need to agree to the agreement before you can continue.

Now check the status of your account and make sure it is “active”. If your status is not active then you can activate your account by resetting the password. If your status is retired then you will need to contact IT ([help@ece.utexas.edu](mailto:help@ece.utexas.edu)) to reactive your account. Once you reset your password it will take about 15 minutes for it to work. **Sometimes this process can take up to a day so please do it early!**

## Programs Required

Now that you have an ECE LRC account, you will need to be able to access the machines available to you. Please download the required software based on your OS. It is all free.

**Windows:**

Download putty: [putty.exe](#)

Download Winscp: [winscp.zip](#)

**OSX and Linux:**

Already come with a terminal installed.

## Commands you should know

The following is a bare minimum list of commands needed for the course. If you do not know some of these commands, please try to learn them. If you need additional help the TA's are always available, but the best resource is Google. These are also the commands that we will be going over in the first discussion section. If you are already familiar with linux and do not feel you need this extra help, please still go to the beginning of the first discussion section as the TA's will have a few important announcements.

1. ssh <username>@mario.ece.utexas.edu
2. pwd
3. cd
4. cp, -r
5. mv
6. mkdir
7. ls, -a, -l
8. rm, -r
9. dos2unix
10. sftp, scp
11. vim
12. history
13. gcc, -ansi
14. man
15. less
16. exit
17. chmod
18. cat

## Part A

1. Log onto mario.ece.utexas.edu (hint: ssh)
2. Make a directory, call it EE460N
3. Print out the permissions of that directory (hint: ls)
4. Ensure that the permissions restrict other students from viewing the contents of that directory. If they do not, change them.
5. cd into the EE460N directory
6. Make a directory, call it lab0

7. cd into that directory
8. Print the name of the current directory to the screen (hint: pwd)
9. Using vim, create a file in the current directory. Call it Lab0Submit.txt
10. Type the following into the file: "The EE460N TAs RULE!"
11. Save and quit the file
12. View the file using less
13. run dos2unix on the file (Note: this will not actually do anything. Why?)
14. Use vi to create another file, call it hello.c
15. Write a simple hello world program in C.
16. Compile this program using gcc -ansi
17. Run this program
18. Cd back to the EE460N directory

## Part B

For part B, Use the [submission instructions](#) to submit the Lab0Submit.txt file from Part A. Try running the -ls option. Now try the -show\_acceptable\_files option. Run the -test option. Now run the -test option again. Now run -grade.

### Here's what you just did:

When you submit the file, you are just uploading it to our server. Testing the file runs the full grading system on the file, but it only outputs the grade. For example, say you are submitting lab 1. If you run with the -test option the script will return something like this:

```
spruett 10
```

Which means that the user named 'spruett' got a 10 on that assignment. When you run -test, that runs ALL of the test cases we have. So if I were to run -grade at this point I would get a 10 still. This is helpful because you get a sneak peak at what your grade is before you actually grade it. Note that it doesn't give you very much information. This is by design. The testing system is not meant to be a debugger, it is only meant to be a sanity check before submission.

**You will only be getting 1 test case per lab.** So use it wisely. Once you are satisfied with your lab you can run the script with the -grade option. This will actually grade the program and return to you a full grading report. It will not tell you what the test cases are and we do not release the test cases at any point in the year. Once you run -grade the score reported will be recorded as your final score on that lab. You will not be given another chance to grade, so again, use it wisely. You must run -grade. If you have not run -grade by the deadline then your score will not be recorded, even if you submitted something.

**Contact a TA if you do not understand the output of any of the submission commands.**