# Pre-Lab for Lab 0

## Lab Overview

In this first lab, you will be introduced to working our Linux Lab. You will be provided your CSCI user account information, be shown how to log in, how to activate your card key, and be guided through creating and running your first C++ program.

For all your pre-labs, your task is to complete and submit the pre-lab and then **bring the completed pre-lab to the lab.**

## Pre-Lab Submission

Complete this Pre-Lab and submit it as a PDF by 23:59 the night before your lab section for Lab 0.

* Complete the following sections of this document before submitting:
  + Lab Prep: Basic Linux Commands.
  + Lab Prep: More Advanced Commands

## Learning Goals

1. learn to access your account
2. learn to navigate the file system on the command line
3. learn to create, edit and save a file on the command line

## 1. Access Your Account

All your labs will be done in our department’s Linux Lab. In order to access the computers in this lab, you need to **log in** with your csci user account. In this lab, your instructor will give you your **csci account information**, which you will use to log in to a computer in the lab. You will likely be prompted to update or change your password. So please carefully follow the instructions as you log in. If you encounter any problems, please notify your instructor.

## 2. Navigate the File System and Using Linux Commands

During lab, you will follow along with the instructor to familiarize yourself with the linux file system. All our work for this class will take place on the command line, and there are some important commands that you’ll want to keep track of. Please fill out the following exercise to prepare yourself for lab. The answers can be found in various places including: the Lab 0 document, the course slides, and in the course readings.

On the command line, we write small phrases that are usually of the following format:

: command\_name option1 option2

In this example, command\_name is actually program, and it has two *command line arguments*. These command line arguments are things that are passed into the program. They could be a word that tells the program how to run, or they could be a file name.

### Converting My Wishes in to Linux Commands

Figure : What I want to do

Create a directory and name it “Lab0”

mkdir Lab0

Figure : How I tell Linux what I want to do

Lab Prep: Basic Linux Commands.

Identify the Linux command for each of the following tasks and **fill in the empty cells in the table**. The first one has been completed for you:

|  |  |
| --- | --- |
| **Task** | **Command** |
| Create a directory | mkdir |
| List all files and directories |  |
| Change directories |  |
| Show my current directory |  |

### 3. Create a first program, and make it go!

Once you’ve set up all your directories for the course, you will create a program, and run it. The process of creating a program on the command line involves (at least) the following steps:

1. Create a file, and give it a meaningful name
2. Write code in the file
3. Save the file
4. Compile the file and generate a program
5. Run the program

Your lab document breaks these steps down for you and goes into more detail. Please review Part 2 of the Lab document to help you answer the following questions:

Lab Prep: More Advanced Commands

Identify the command that will let you do the following tasks, you will need to add options to these commands for them to be complete. Again, I’ve completed the first one for you:

|  |  |
| --- | --- |
| **Task** | **Command** |
| Open a file named Lab0.cpp in emacs in its own thread | emacs Lab0a.cpp & |
| Compile Lab0.cpp to the fill Lab0x with all warnings and errors shown |  |
| Run the program Lab0x |  |