# Lab 1

**CIS3110/Operating Systems**

W25

CODING LABORATORY

For this lab you have a formally assigned task: make sure that you can get to work in our programming environment.

We will use `linux.socs.uoguelph.ca` for this course. This is where everything will be graded, where all the tools are installed, and therefore is where you should plan to do your work.

## 1 Tasks:

1) Log in to `linux.socs.uoguelph.ca` using ssh(1)

   - Note that commands used in this course will be written in this format, which tells you that the command ssh has documentation available to you on the Linux environment in chapter 1 of the manual page system, which is accessed using the command man(1):

     ```
     man ssh
     ```

     - There are additional resources on how to use ssh(1) in our CourseLink site for this course
   - For an overview of the manual page system, use the argument `intro`:

     ```
     man intro
     ```

2) Get the lab source code from the course directory and copy it to some location of your choice and then cd into it and list the files:

   ```
   cp -r /home/courses/cis3110/assignments/CIS3110-W25-Lab1-code .
   cd CIS3110-W25-Lab1-code
   ls
   ```

3) Build it using your C compiler

   - Hint: you should know how to do this from previous courses
   - Bigger hint: notice the presence of a makefile in the current directory – what tool have you used in the past based on a makefile?

4) Run the executable that you get:

   ```
   ./areWeLost
   ```

5) Look at the source code in the lab1_main.c file. Editors installed on the system for you include:

   - joe(1) : "Joe's Own Editor" – this will work for most of your programming purposes. You can get help at any time by typing:

     ```
     [CTRL]-K H
     ```

     (That is press both [CTRL] and K at once, and then press the H key – this will put a help reference at the top of the screen. Sometimes people write [CTRL]-K as '^K' instead.)

     Other common joe(1) commands are:

```
* '[CTRL]—K X' : save the file
* '[CTRL]—K F' : find something
* '[CTRL]—C'   : abort (exit, possibly without saving)
```

- nano(1) and pico(1) : these are really simple editors that came with an old email program called pine(1). They will put what you type into a file, but don't do much else. **Be aware that if you have a long line in your program and you scroll past it using these editors they will insert a line break "to make it look nice" that may break your code**.
- vim(1) and vi (1) : powerful but arcane program editors – if you want to learn how to use vim(1), there are good tutorials such as this one: `https://openvim.com/`
  - If you get into vim by accident and are trapped, type "ZZ" ([SHIFT]—Z—Z) and it will save and exit
- NOT *VS Code* – it isnt' installed here because this editor cannot be run over ssh(1), while the ones mentioned above can.

  *VS Code* has lots of great functions, but it can be confusing because it is so complicated. It does many things for you, and some of these things you actually don't want. We don't have the resources to debug *VS Code* for you in this course – if you want to use it that is up to you (we really don't care what you use to type in your assignment code) – **BUT**: be aware that lots of people in the past have confused themselves into developing on the wrong system when using it.

  That is in fact fundamentally the reason we are doing this lab – so that you can be absolutely sure that you are writing and running programs on the environment where we will mark it.

  It will be *your responsibility* in this course to ensure that your code works as run on linux.socs – don't get foolishly burned by handing in code that only works somewhere else.

  Ensuring that you know when you are correctly logged into our programming environment is the first step to success.

6) Answer the questions on CourseLink based on the lab tasks

   Completing this lab will let both you and the instructional staff know that you can successfully log in to our working environment.