

CSC 101 Lab Week 1

Getting Started in the Computer Science Labs (CSL)

ISSUED: Monday, 26 March 2012
DUE: Monday, 2 April 2012, by the end of lab
POINTS POSSIBLE: 1
WEIGHT: 1% of total class grade

Overview

The purpose of this week's lab is to familiarize you with the basic operation of the CSL computers and to introduce fundamental concepts of the UNIX operating system. "CSL" stands for the "Computer Science Labs".

Exercise 0: Getting Your Computer Account

As noted in the syllabus, you should have a CSL account if you are pre-enrolled in the class. The account has the same user name and password as your Cal Poly campus account. If you're pre-enrolled and don't have an account, tell me immediately during the first lab and we'll get an account request filed.

If you're adding the class, then you should pair up in the lab with someone who does have an account, so you can work on the first day. As noted in the syllabus, you'll get a CSL account in a day or two after you use your e-permit for the class.

Exercise 1: Logging on to Linux in the CSL

Logon to one of the lab machines in 14-301. The machines should be initially configured to start up in the Linux operating system. If your machine happens to be running Microsoft Windows, ask me and I'll restart it to run Linux.

Once you've logged into Linux, start a Terminal program. You can do this by right clicking on the main background of the display and selecting the menu item 'Open in Terminal'. Another way to open a terminal is to click on the 'Applications' menu in the top menubar, selecting the bottommost 'System Tools' submenu, and then selecting the 'Terminal' item. When the terminal starts, it displays a prompt in the upper left corner of the window, and otherwise the window is blank.

At this point, you are running in a "UNIX shell". This is where you type UNIX commands to get work done. If you're used to point-and-click style operating systems like Windows or Apple, it may take you a while to get used to the text-based command shell in UNIX. This lab is a start.

Exercise 2: Creating a File Directory for CSC 101

Type in the following terminal commands:

```
cd ~
ls
mkdir csc101
cd csc101
```

These commands do the following things:

- change directory (`cd`) into your home directory (`~`)
- list the contents (`ls`) of the current directory
- make a new file directory (`mkdir`) named "csc101"
- change directory (`cd`) from your home directory into the `csc101` directory

Exercise 3: Creating and Saving a File with Emacs

In the terminal window, type the command

```
emacs
```

This will start the Emacs text editor that you can use for creating, editing, and saving files.

Emacs is a very large and very powerful system. In the beginning, you should use it as just a basic text editor, similar to how you use a program like Notepad on a PC, or TextEdit on a Mac. Emacs has several features that make it useful for editing programs, not just plain text files. You will become familiar with these features as the quarter progresses.

The best way to get started with Emacs is to run its own tutorial. To do this, click on the "Emacs Tutorial" link in the startup screen. You do not need to go through the entire tutorial, just enough to get you familiar with the basics. You can also play around with the menu commands and toolbar to see what it does.

Once you become familiar with the Emacs basics, create a file with the following contents:

```
This is my lab 1 file.  
This class is CSC 101.
```

Save the this on a file named `lab1.txt`, and exit emacs. Confirm that your `lab1.txt` file exists in your `csc101` directory using the UNIX `ls` command.

Exercise 4: Submitting a File Using handin on Unix1

In the terminal window, execute the following commands:

```
ssh unix1  
cd ~/csc101  
handin gfisher 101_lab1 lab1.txt  
exit
```

These commands do the following things:

- log into the computer named `unix1` using the secure shell program `ssh`
- change into the `csc101` directory on `unix1`
- run the `handin` program to submit your `lab1.txt` file
- log off of `unix1`, going back to the lab machine

This is the way that you will submit all CSC 101 assignments. Note that files and directories you create on any CSL machine will automatically exist on `unix1`. This is because the lab machines and `unix1` all share the same file space. However, you must always explicitly login to `unix1` to run `handin`. That is, DO NOT run `handin` from any lab machine other than `unix1`.

Exercise 5: Exploring more UNIX Commands

Have a look at the CSC 101 "UNIX Basics" handout in the 101 handouts directory. You do not need to understand everything that's there, but you should read through it a play around with some of the UNIX commands. Be careful not to do any major damage with the `rm` command. It removes files permanently, and there is no way to undo it.

There are many basic tutorials available on UNIX if you want to learn more, for example <http://www.calpoly.edu/~www/unixcmds.html> from the Cal Poly help desk folks. If you want to use the Vim editor instead of Emacs, John Seng has a tutorial that is linked from the CSC 101 handouts page.

In addition to using the lab machines, you can login to `unix1` from off campus. There is help for doing this at <https://wiki.csc.calpoly.edu/labs/wiki/FAQLoginHelp>

You do not need to become a UNIX expert in CSC 101, just know enough to write your programs, test them, and hand them in.

Exercise 6: Editing, Compiling, and Handing in a Simple Program

Using the necessary UNIX commands, perform the following tasks:

- a. Copy the example program from `~gfisher/classes/101/labs/1/example1.c` into your `csc101` directory.
- b. Carefully edit the program so that it checks if a number is negative, instead of checking if it's positive. "Carefully" means not to change anything other than a single character that makes the program do what it needs to do. (Hint: it's on line 15.)
- c. Carefully edit the program comment on line 15 so that it describes what your modified program does, i.e., checks if a number is negative. "Carefully" means not to change anything other than what is between the `/*` and `*/` comment characters on line 15.
- d. Compile the program using the following command:

```
gcc -ansi -pedantic -Wall -Werror example1.c
```

which will put the compiled program in a file named `a.out`.
- e. Run `a.out` enough times to convince yourself that it works, i.e., that it checks if a number is negative.
- f. Login to `unix1` and submit your program using the following command:

```
handin gfisher 101_lab1 example1.c
```

Confirm that You're Done

When you've completed all six exercises, raise your hand and I'll come around to check that you're done. You have until the end of lab on Monday 2 April to complete the work. This should be more than enough time. If you get done early with Lab 1, you can use lab time to start working on Program 1.