Sessions and C Programming
Lab D

A tutorial section is provided at the end of this lab.

The labs, for this course, are designed to be completed on your own at home or in the 3rd floor Trottier labs. These labs are not graded. You do not hand in these labs. If you prefer to work on a lab with a TA tutorial group, then check the schedule for a TA's tutorial session. You will find this schedule in our MyCourses page under Content/Course Information/Prof & TA Coordinates. Since the university has limited lab space, the TA might ask you to bring your laptop and work in a classroom instead of a lab.

This lab is about two topics, mimi sessions and programming in C.

Some labs will have a question zero. These questions will not be covered by the TA during the tutorial. It is extra content meant for you to do on your own.

QUESTION ZERO: Optional problem.

Type `man gcc` to see what it tells you. What are the basic command options you will need to compile a C program?

Type `echo $SHELL`, what does it tell you? Use what you saw during class to determine which shell you are using. This will help you identify your login script. Which login script does mimi use?

QUESTION ONE: Edit your login script.

Your login script is probably `.bashrc` or `.bash_profile`, which is found in your home directory. If the file does not exist, then create it using vi or vim (yes, the filename is proceeded by a period). Make sure you identified the correct login script by editing the login script (as described below) and then logging out and then back in again to see if your edits executed when you logged in.

Do the following at the command-line:

`vi .bashrc` or `vi .bash_profile`

If you are using **ssh**, the best one to use is `.bash_profile`.

Then edit the file in the following ways:

- Change your prompt
- Echo "Welcome to Bash"

You will need to logout and then log back in to make sure it is running. You can test it temporarily by typing at the command-line:

`source .bashrc` or `source .bash_profile`

But to make sure it runs when you login, the best test is to logout and then back in.

Important: Filezilla users must delete all echo statements from .bash_profile and .bashrc after this experiment. Filezilla does not like it when the login script has echo statements.

QUESTION TWO: Compiling your first C program.

Using vi or vim type in the following C program and call the file `labD.c`:

```
#include<stdio.h>
int main() {
   puts(“Hello World\n\n”);
   return 0;
}
```

Now compile that file by typing the following command at the command-line:

```
gcc labD.c
```

If you see no errors, then run the program by typing at the command-line:

```
./a.out
```

If there were errors, then look at the errors and fix them and try to compile and run again.

QUESTION THREE: Edit your first C program.

Modify the file `labD.c` in the following way:

- Create an integer variable named count and initialize it to 5.
- Using a for loop, print out the Hello World, count times.

Compile and run the program.

You have completed your lab.

TA TUTORAIL INSTRUCTIONS

Begin the class by helping the students determine which login script they are using. **The safest one to edit is `.bash_profile`. I suggest the students try that one with `ssh`**. If they already have a `.bash_profile` or `.bashrc` file, have them edit each file by adding an `echo` statement that prints `Hello World RC` or `Hello World Profile`. Then they can logout and then back in to see which login script printed `Hello World`. If they do not have the file, then they should create each file and logout/in to discover which script executed. If both scripts executed, then they can pick any one of them.

Ask the students if they want you to do the Lab questions together or they want to do it on their own. If they elect to do it on their own, then hang around to offer help (but tell them about .bash_profile and ssh). If they elect for you to lead them, then connect your screen to the projector and do each question. Do each part of the question waiting for the class to complete the step before going on to the next step. Make sure you do everything from the mimi command line using vi or vim.

If there is time cover these extra topics:

- Show the student how to manage C compile time errors from the command-line
  - Redirect errors to a text file (in Bash use `2>`).
  - Use the vi or vim **<u>colon</u>** command to jump directly to an error line number.

- Show the students how to compile their programs using a name other than a.out.

- Use any remaining time to answer general questions about the material in this lab.