How to Complete Program 1

First Steps

Step 1: Obtain your Unix Account Username and Password from the staff in Olsen Hall, Room 312.

Step 2: For Windows users, download the Bitvise SSH Client onto your computer. Here is the link:

https://www.bitvise.com

This utility will provide you with the ability to create both a terminal window and a drag and drop file transfer program.

Step 3: Using bitvise, create a terminal window and connect to cs.uml.edu. Use your username. And then type in your password. When you see the Unix system, type the Unix ls command. The ls command lists the files in your current directory. Everytime you log onto cs.uml.edu the system will place you in your “home” directory.

Step 4: Immediately after you complete the first 3 steps, send james_Canning@uml.edu an email message saying: “Jim, I am able to log onto the cs.uml.edu.”

Step 5: Using the mkdir create a subdirectory beneath your home directory. Call this new subdirectory computing1. You would type: mkdir computing1 and then hit the return key.

Unix-prompt> mkdir computing1

Step 6: Go to your new computing1 subdirectory by invoking the Unix command cd as follows:

Unix-prompt> cd computing1

Cd stands for change directory.

Step 7: While you are in your computing1 subdirectory, I want you to use the emacs editor and create a file called: template.c. Issue the emacs command as follows:

Unix-Prompt> emacs -nw template.c

Remember, I want each of you to create your programs on the cs.uml.edu machine and I want each of you to use the emacs editor. Learning the emacs editor may look difficult, but it is easy to learn the few necessary and basic commands. Once you understand them, you will be off and flying.

Your template.c file should contain the following text given below. Instead of typing “Your Name”, put your first and last name there. Type the text into emacs. Periodically you may wish to save what you have typed by issuing the “emacs write named file” command. This is a sequence of two keyboard depressions. That is,

Control-x followed by Control-w
You would use the Control key like you use the Shift key. Depress both the Control key and the x key together. Then depress the Control key and the w key together.

This “write” command will save your buffer to your file and keep you inside the emacs environment. Most users use this C-x C-w sequence frequently.

You can exit emacs and get back to the Unix system by issuing the sequence:

\[
\text{Control-x Control-c}
\]

Once you filled template.c with the text above, exit emacs and get back to Unix. Then you should type the \texttt{ls} command again and see that your template.c file is there.

\textbf{Step 8:} Issue the Unix more command to take a look at your template.c file without getting back into emacs. Hit the carriage return key when you finish the typing the command.

\[
\text{Unix-Prompt> more template.c}
\]

\textbf{Step 9:} After you complete Step 5 through Step 8, I want you to send another email to me. It should say:

“Jim, I have used emacs to create my template.c file. Nice.”

\textbf{Step 10:} Now, I want you to use the Unix file copy command. It is the \texttt{cp} command. Use this command to copy your template.c file into a new file called, p1.c.

\[
\text{Unix-Prompt> cp template.c p1.c}
\]

\textbf{Step 11:} Issue the \texttt{ls} command to see that your file is in your computing1 directory.

\textbf{Step 12:} Issue the more command to look at your file without entering emacs.
Step 13: Use emacs to and edit your p1.c file.

Unix-Prompt> emacs -nw p1.c

Change the number of your program to: Program 1.

Change the name of your program to Hello World.

Make sure that all of your asterisks line up.

Add the printf statement as indicated in your Program 1 description assignment sheet.

Save the buffer to p1.c. Exit emacs.

Step 14: Issue the ls command if you wish. Issue the more p1.c command if you wish.

Step 15: Now, I want you transform p1.c into an executable image. How? By issuing the gcc utility as follows:

Unix-Prompt> gcc -Wall -ansi p1.c

Step 16: If there we no errors or warning messages, then run the program. How? By typing this on the Unix Command line:

Unix-Prompt: ./a.out

Step 17: If your program worked, it should print out Hello World onto your screen. Then, I want you to electronically submit your program to me. I DO NOT want you to submit a.out. You need to submit your p1.c file to me using the submit utility as follows:

Unix-Prompt> submit canning p1 p1.c

When you do this, I will automatically be notified that you submitted your p1.c file to me.

Step 18: If you have gotten this far, get up and dance around the room. This is a milestone event for you in this course.

Step 19: Now we need to take about printing your file. Hmmm. I DO NOT want you to select from the screen. No way. Nunca. Nada. Here are two printing options.

Option 1: Using the Campus Printers, but this will cost you money.
Option 2: Using your own local printer that is connected to your laptop/desktop.

Option 1: Issue the Unix lpr command as follows:

lpr -PHP_BW -Usid p1.c

where sid is your student id number.
Option 2: Using the drag and drop feature of the bitvise “other” screen move the p1.c file from cs.uml.edu to your laptop. Once the file is on your lab top open the file in some utility that you can print from. Remember, I want you to use courier font. Make it look nice. Do not have your header comment spill over to the next line. Delete some of the asterisks if need be.

**Step 20:** Once you have printed out p1.c and it looks like a “thing of beauty” send an email to me saying: “Jim, I have printed out p1.c. It is a thing of beauty.”

**Step 21:** After you printed this out, you can slide the paper copy underneath my office door. My office is located in O’Leary 302.