Homework 1 – Xwindows, ggc, make, valgrind and tar

Objective: To familiarize the student with the basic aspects of all homework assignments in this course such as, connecting to the system using a secure shell, editing, Xwindows, compiling with **gcc**, basic Linux operations, creating make files, packing code into a **tar** file and submitting a tar file.

Part A. Login to any CE machines noted in the lecture using a secure shell client (e.g., **putty**). Make a directory called **hw1** (**mkdir hw1**). Then, go to that directory (**cd hw1**) and using your editor of choice – *e.g.*, **vi**, **nano**, – create a text file, **lastName_myinfo.txt**, (lastName is your last name) with the following information (one item per line):

- Full Name
- Major and anticipated graduation date
- List of each programming language that you know and how well you know it.

Part B. You must have Xwindows installed on your PC, Mac or other device. Read the Week 1 class notes to find out how to enable Xwindows.

- Run the Linux command: **module load matlab**
- Run Matlab by typing **matlab** and take a screen capture, or picture, of the Matlab logo splash screen and include it in your tar file as “screen.xxx” (xxx is the file type of your screen shot).
- Note: Matlab off campus can be **very**, **very** slow.

Part C. List all the hidden files in your **root directory** (**e.g. cd ~ ls -la**) and redirect the output using UNIX redirection “**>**” to a file called “**hidden.txt**”. Edit the hidden **“.bashrc**” file using the editor of choice and ***add*** the following to the bottom of the file:

```
export GDFONTPATH=/usr/share/fonts/dejavu
export GNUPLOT_DEFAULT_DDFONT="DejaVuSans.ttf"
```

then copy the .bashrc file into your hw1 directory (**cp .bashrc hw1/bashrc.txt**), calling it “**bashrc.txt**”. Be sure to include your “**hidden.txt**” and **“bashrc.txt**” file in your TAR file.

Part D. Attempt to compile the file “**QuadraticSolver.c**”. It has numerous “typos” in it. Fix each one using your editor of choice. ***Change the program comments*** (top lines) ***to reflect the correct identification***.

When you have fixed all the problems, generate an executable called **qs**. Run the program to solve the following quadratic equations:

1) $2x^2+2392.2x-766267.2=0$

2) $x^2-1x+1=0$

3) $0x^2+x-10=0$.

Capture the output in a text file called **out.txt** using the UNIX redirection “**>**” and append “**>>**” operators.

Did all three runs produce good output? If not, what was wrong and why do you think it is wrong? Document your reasons in the analysis.txt file. Your analysis.txt file must be formatted so that it can be viewed on an78 character wide display. (e.g. Format the final file by adding a hard CR/LF at column 78 or before).

Part E. Compile the **val.c** program and then run it using valgrind to discover programming errors. Use the class notes to determine the exact valgrind syntax. Run the valgrind command before making any changes and redirect the output to **before.txt**. Using valgrind output as a clue, fix each of the "bugs" in val.c.

Part F. Write a quality makefile to implement the following targets: **all, test, mem1, mem2, help, clean.** Important sections like test and mem should use the echo command to notify the user about what is happening.

"all" – should build a binary called **"qs"** and **"val"** from the provided code.

"test" – should run the identified **qs** tests and redirect all the output to a "out.txt" file, then display the out.txt file on the console

"mem1" – should run the standard valgrind memory command:

v**algrind --tool=memcheck --leak-check=yes --track-origins=yes** with a good quadratic solver (**qs)** equation and redirect **ALL** the output (stdout & stderr) to "**mem.txt**" **and** display the contents on the console.

"mem2" – should run the standard valgrind memory command with the **val** and redirect **ALL** the output to "**mem.txt**" **and** display the contents on the console.

"help" – should list all the key makefile targets

"clean" – should remove all temporary and intermediate files.

Part G. To submit your work go to your current homework directory, in this case **hw1**, and create ***a single tar file*** (**lastName_hw1.tar**) containing the required files (where lastName is your last name, e.g. Repka_hw1.tar):

lastname_myinfo.txt,

QuadraticSolver.c

qs

analysis.txt

mem.txt

out.txt

before.txt

bashrc.txt

hidden.txt

Makefile

screen.xxx **(xxx the file type of your screen shot)**

Important ! Always make sure that your tar files have been created correctly.

Verify the content using **tar -tvf lastName_hw1.tar**

Once you are sure that everything is correct, submit your homework **lastName_hw1.tar** to the DropBox in MyCourses.

Grading Rubric

1. (20 points) Valid and correct tar file (with the requested files and filenames)
2. (20 points) Complete **lastname_myinfo.txt, screen.xxx, out.txt, hidden.txt, bashrc.txt, mem.txt, before.txt, after,txt** and **analysis.txt** formatted for a 78 column display.
3. (20 points) Complete make file
4. (20 points) **QuadraticSolver.c** compiles cleanly, with no warnings, no valgrind errors and runs to completion.
5. (20 points) **val.c** compiles cleanly, with no warnings, no valgrind errors and runs to completion.

Additional Notes:

- Compile the programusing the flags **-O1 - Wall -pedantic -std=c99**
  gcc -Wall -pedantic -std=c99 -O1 -o qs QuadraticSolver.c -lm
  Make sure to eliminate all the warnings and errors. (Note the **-lm** directive, it is necessary to link the **math library** to your executable.)

- Run the program to solve the first equation given. Redirect the program output to **out.txt** using
  ./qs *a b c* > out.txt
  where ***a***, ***b*** and ***c*** should be replaced by the actual values of the coefficients of the quadratic ax^2+bx+c.

- Run the program again to solve the other equations. This time you should append the program output to **out.txt** using
  ./qs *a b c* >> out.txt

  Verify with your editor that the **out.txt** file has properly recorded the output of the three runs.