Assignment 1: Familiarization with Linux, System Tools and C Compiler

**Due: Wed, Jan 18, 11:59pm**

Learn and understand the functionality of all practiced commands in the following!  
Start your Ubuntu VM or physical Ubuntu machine and login. Select Accessory/Terminal to open a window running the Bash shell.  Type "**help**" to learn about a list of shell commands defined internally. Put all your answers in the table at the end of this document.

1. Three of the text editors available in Ubuntu are ***nano***, ***gedit*** and ***vi***. Of these ***gedit*** is a GUI program. Type the following program in an editor of your choice, then save in a file named ***hello.c***. If ***ProgramA*** is not installed, type ***sudo apt-get install ProgramA*** to install it. **Note:** You may google "Linux vi tutorial" for the commands to navigate the ***vi*** editor. You can also search online tutorials for ***nano*** and ***gedit***. Get familiar with at least one of these editors. What is your favorite among these three editors? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include <stdio.h>  
int main(int argc, char \*argv[])

{

fprintf(stdout, "hello world\n");

return 0;

}

1. Use ***pwd*** to print the current directory path. What is the output and what is the meaning of this output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Type ***man mkdir***. Type **q** to quit manual and return to shell command prompt. (Note **vi** is used for display. The ***man [1-8]* command** can be used to get manual for any **command.**) What is displayed on the first line of the output? What is the meaning of the whole output?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Type ***clear***. Type ***mkdir***. What is displayed on the first line of the output? What is the functionality of this command? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Type ***mkdir --help***. What is displayed on the first line of the displayed output? What is the meaning of the whole output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (From 3 & 5, notice there might be multiple ways to get help about how to use a command.)
3. Create a directory named ***test*** using the ***mkdir*** command.
   1. Use the ***mv*** command to move ***hello.c*** you just created to directory ***test***. What is the functionality of the ***mv*** command? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. Use the ***cd*** command to change the current directory to ***test***. What is the functionality of the ***cd*** command? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Type the ***ls -l*** command. What is the output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ What is the functionality of the ***ls*** command? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Type the ***set*** command. The output may have several pages. You may use “**set | more**” to see output one page at a time.
   1. What is the first directory in the search path? (**Hint:** The answer is the first directory in the value of the *PATH* environment variable. The directories in the value of *PATH* are separated by “:”. You can type **set | grep -a “PATH=”** to filter the output of **set** and find the output line that starts with “PATH=”. Note that | (the vertical bar) is the pipe operator of the shell and is used to pipe the output of **set** into the input of **grep**. You can also get answer by typing **echo $PATH**. )

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. What are the functionalities of the ***set,*** ***grep*** and ***echo*** commands respectively? (If “**man *command***” does not work, try “***command* --help**” or “**help *command***” or search online.) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use ***cat,* | *(the pipe operator as used in 7a),*** and ***sort*** on the command line to display the lines of ***hello.c*** numbered and in reverse order. (You can type **cat --help** and **sort --help** to get help on how to use these commands.)  
   What command line can achieve this task? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Type ***wc hello.c***.
   1. What is displayed? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. What does each number represent? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. What command line would you use to display only the word count? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Type ***grep -B3 printf \*.c***. In your own words, state what is the purpose of the executed command? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Type ***gcc -c hello.c*** to compile your C program ***hello.c***. If you see compile errors, reopen ***hello.c*** in a text editor, fix the errors and recompile until success. Type the ***ls*** command to list the contents of current directory. What new file did the compiler generate? Does this new file contain machine language, assembly language or high level language? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Type ***gcc hello.c***. What new file did the compiler generate? How can you use this new file? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   **Note:** The new file is the executable for the program. Type its name to execute it. If you get “permission denied” error, you may use “**chmod +x ProgramName**” to add execution permission. If you get “command not found” error, you may type **./ProgramName** if the current directory is not in your *PATH* environment variable.
6. We can use “***>***”to redirect the output of the compiled ***hello*** program to the file ***flip.txt***. Type ***./hello > flip.txt***. Did you see any output message in the Terminal window? Type the ***ls*** command to list the contents of current directory. What new file did you see? Can you guess the functionality of the > output redirection operator? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. Type ***cat flip.txt***. What is the output? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What does the **cat** command do? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use the ***rm*** command to delete all files in your ***test*** directory EXCEPT ***hello.c***. What is the functionality of the ***rm*** command? How do you delete all files in current directory with suffix ***.o*** by invoking the ***rm*** command only once? \_\_\_\_\_\_\_\_\_\_\_\_\_
2. Compile and run the ***squareroot.c*** program attached with this homework.
   1. What does this program produce? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. This exercise tests the input redirection operator “**<”** in the shell. Create a text file **twonumbers.txt** containing two lines, each containing a number. Run your program as **squareroot < twonumbers.txt**. What did you see in the output? What are the purposes of the **>** and **<** redirection operators respectively? (See also 13) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Compile and run the ***TestAssert.c*** program attached with this homework.
   1. What is produced if you enter the integer value 6? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. What is produced if you enter the integer value 11? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. What is the purpose of the **assert** statement in the C code? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Question | Your answer |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |