Indian Institute of Information Technology Pune
Department of Computer Science & Engineering

BT23104: Computer Network
Assignment – 4

Instructions:

- Make sure that you read, understand, and follow these instructions carefully. Your cooperation will help to speed up the grading process.
- Following are generic instructions. Make sure that you also check carefully and follow any specific instructions associated with particular questions.
- In this assignment, you will explore the basics of shell script, fork and inter process communication in operating systems.
- Complete the assignment with the best of your own capabilities. Create a single zipped/compressed file containing all your programs and related files.
- The compressed file must contain:
  - All your program files
  - Makefile to compile your programs
  - Readme file to explain your program flow
  - Proper code indentation and comments

  *** Each of the component mentioned above carries mark.**

.

Ethical Guidelines (lab policy):

- Deadlines: Deadlines should be strictly met. Assignments submitted after their respective deadlines will not be considered for evaluation.
- Cheating: You are expected to complete the assignment by yourself. Cases of unfair means and copying other's solution will not be tolerated, even if you make cosmetic changes to them. If we suspect any form of cheating, we are compelled to award -2x of the total mark alloted to the assignment.
- If you have issues in meeting the deadline, it is suggested that you consult the instructor and not use any unfair means to submit your assignment.

Question-1: Introduction to basic shell scripting:

A. Write a shell script that searches for a particular string "**search_string**" (case insensitive) in a directory containing multiple files (recursively).

a) Print the line number and file name (inside the directory) where the search_string occurs.

b) Also, print the total occurrence of the search_string in each file.

* The **search_string** should be given as a command line parameter.

* Use of inbuilt grep command is restricted.

B. Implement the above question (part A) in C. Using in-built system command is prohibited in this question.

Question-2: Create a Shell of your own.

Write a C program to simulate a bash shell of your own say “**my_shell”** which consists of Linux commands:

1. pwd
2. mv
3. grep
4. split

* The commands are to be implemented explicitly using C and should be executed via **my_shell** process using exec system call. Use of already existing implementation of these commands in bash or any other shell is strictly prohibited.

Please note that:

1. **my_shell** must look exactly like your terminal. The main program should read command line arguments from standard input and it should parse them into A) **Command_Name** (as mentioned above). B) **Argument** (if any).

2. **my_shell** should also show correct usage of any command. Hence implement **help** command for your linux commands.

3. **my_shell** should look exactly like the one in your Linux terminal: user@hostname:path to current dir.
   For example: **dipika@system_lab:~/Assignment1$**

4. Any illegal command, unrecognized arguments should throw a meaningful error to the user and the shell should not crash or be killed.

5. **my_shell** should also support the **exit** command so that the process is not killed or forcefully exited and print a meaningful exit message.

6. **my_shell** should also remember the recently used commands. Hence implement **history** command for your shell. History function need not be explicitly implemented, but when using **UP** and **DOWN** arrows, the recently used commands must be available.

7. **For grep and split commands implement a minimum of 4 arguments.**

***** Hint: Use readline function in C.**