COMP 8567- Advanced Systems Programming
Instructors: Dr. Boufama and Dr. Ranga
Project Work
Title: Distributed processing of grep with client server programming.
Due by: Apr/26/2022 (to be demonstrated to the GAs)

This project can be implemented by one student, or in a group of two students.

Write two programs server.c **(server)** and dgrep.c **(client)** in C to implement a simple command to search **whole words** in two files using distributed processing.

- The server and the client are required to run on two different machines.
- The communication between the client and the server will be through sockets
- The server must start running before the client, and wait for a connection from the client.
- The server must run in an infinite loop.

The client's name is **dgrep** and must accept the following (three) command line arguments

- argv[1]=*pattern* (pattern to be searched in whole words)
- argv[2]= *file1(Relative or absolute path of the first file)*
- argv[3]= *file2 (Relative or absolute path of the second file)*

Ex: **$ dgrep hello sample.txt test.txt**

$ dgrep include t1.txt ~/folder1/t2.txt

- Both *file2* and *file2* are initially stored on the client machine.
- if either *file1* or *file2* cannot be found by the client, the client program is terminated with a suitable message.
- If the **client is unable to connect** to the server for any reason, the client program is aborted and a suitable message is displayed.
- **If the client successfully connects to the server:**
  - The server forks a child process and lets the child process take care of the client through a separate function called **handle_client()**. The parent process then goes back to waiting for a client.
  - The client sends *file2* to the server (*file1* remains on the client machine)

- the client runs grep -w *pattern file1* locally on the client machine
- handle_client() runs grep -w *pattern file2* in the child process on the server and returns the result to the client. (use the system() library function to execute commands)
- The communication between the server and the client must be through sockets
- The client should finally display the combined results (from both the client and the server – in that order) on the standard output.
    - Each line of the display must be prefixed by the name of the respective file followed by ':' Ex: the following should be the output of
      $ dgrep hello text1.txt text2.txt assuming text1.txt has 4 lines of "Hello and Welcome to COMP8567" and text2.txt has three lines of "Hello and Welcome"

      text1.txt:Hello and Welcome to COMP8567
      text1.txt:Hello and Welcome to COMP8567
      text1.txt:Hello and Welcome to COMP8567
      text1.txt:Hello and Welcome to COMP8567
      text2.txt:Hello and Welcome
      text2.txt:Hello and Welcome
      text2.txt:Hello and Welcome
      text2.txt:Hello and Welcome

- The pattern should be highlighted in red
- The client can terminate its connection to the server and also terminate its execution after the displaying the results as mentioned in the previous step.
- Since the server runs continuously and is waiting for the client, the client program can run anytime with the required parameters.

Submission:

- Two files: server.c and dgrep.c
- Note: The successful working of the server and the client on different machines and the overall working of the program must be demonstrated to the GAs. Slots will be communicated in due course.