COMP 8567
Fall 2022
Project
Marks 100

This project is to be implemented either by **one student** or by a **group of two students** only.

Write two C programs a client and a server to implement a remote shell connection using sockets. The server process/s and the client process **will run on two different machines** and the communication between the two processes is achieved using Sockets.

Server:

• Two **copies** of the server (serverA and serverB) must start running before any of the client/s and wait for connections.

- serverA and serverB can run on two different terminals of the same machine or on two different machines.
  - **Please Note**: serverA and serverB are identical copies of the same program running on different terminals/machines

• When either the server receives a request from a client, it forks and lets the child process (of the server) take care of the client request in a separate function called ServiceClient(), while the parent process goes back to wait for the next client.

- A server can receive requests from multiple client programs (running on different systems/ different shell terminals of the same system). Each request must lead to a new child process (of the server) that is created to service the request.

The Server's Child Process:

1. uses " **dup2()**" to read from the client socket instead of standard input
2. gets in an infinite loop:

– reads a shell command from the client's socket,

– if the client sends "quit", then the server's child, closes socket and quits.

– otherwise, it executes the command (You must execute the command **without** using any of the **exec series** of system calls)

3. The results of the command must be transferred to the client.

Client:

The client process connects to the server and gets into an infinite loop

1. reads a command from keyboard,

2. writes the command to the server,

3. if command is "quit", closes socket, and quits

4. otherwise, reads command output from the socket (received from the server) and displays it on the screen

Additional Rules for serverA and serverB (very important):

- The first 5 client connections are to be handled by serverA
- The next 5 client connections are to be handled by serverB
- The remaining client connections are to be handled by serverA and serverB in an alternating fashion. (ex: connection 11 is to be handled by server A, connection 12 by serverB, 13 by serverA and so on)
    - This is mainly to simulate load balancing

Submission and Demo:

- You are required to **submit** two files server.c and client.c
- You will also be required to provide **dem**o of the complete working of the project to the Professor/GAs in the designated time slots which will be announced shortly.
- There would be **no reports** required.

****