Assignment 4
Socket Programming
[Max marks: 16]
Deadline: 19-Feb-2019

Instructions:

- This is a home assignment.
- You are free to use internet.
- Kindly do not copy from any source or other's assignment.
- For **Task 1** you can refer the code from the link given below but do not copy the code.

Task 1: ECHO client-server model with server handling multiple clients. [8 marks]

- Create an echo server to handle 3 client requests simultaneously.
- Connect three clients simultaneously to the server, when server is in listening state .
- Each client will send a message to the server which will be echoed back by the server to that particular client only.

Things to note down:

- Server and Client will accept port number through command line argument. If not implemented in this way then student will be graded 0 this time, for this question.
- Marks will be deducted if message of a client will be echoed to other clients also.
- Code client to accept more than one message. So that client remains connected while other client is communicating with the server.
- Use some phrase like "exit" or "BYE" to terminate the client.

Sample Input:

Client 1	Client 2	Client 3
Hello	Hi!	Hey!

Sample Output:

Client 1	Client 2	Client 3
Hello	Hi!	Hey!
Hello	Hi!	Hey!

Client 1, Client 2 and Client 3 are running on separate terminals.

Task 2 : Program to Build Web Server using sockets [8 marks]

The Client will request a Web page which resides at the Server side .

Task Description:

A Web Page is the page which contains html tags that can be executed in a browser. An application process i.e Client will be requesting access to a web page to the Web Server. On getting such a request, Server will be responding with the page requested. Create 5 HTML pages containing only one heading “Page 1”, “Page 2”and so on. Store these pages in the same directory where the server resides. When client will request one of these pages, display the page content. If page requested by the client is missing then return error “404: Page not found”.

Action at Server side :

- Include appropriate header files.
- Create a TCP Socket.
- Fill in the socket address structure (with server information)
- Bind the address and port using bind() system call.
- Server executes listen() system call to indicate its willingness to receive connections.
- Accept the next completed connection from the client process by using an accept() system call.
- Receive a message from the Client using recv() system call. The message will be actually the name of the web page requested by the client.
- Send the result of the request made by the client using send() system call.

Actions at Client Side :

- Create a TCP Socket.
- Fill in the socket address structure (with server information)
- Establish connection to the Server using connect() system call.
- Make a request to web server with the web page that is residing at the server side and send a message containing this request using send() system call.
- Receive the result of the request made to the server using recv() system call.
- Write the result thus obtained on the standard output.

Sample Input:

Enter the filename from the list:
web1.html
web2.html
web3.html
Web4.html
web5.html

Sample Output:

web1.html
<html>
<title> WEB1</title>
<body> This is WEB1.html </body>
</html>

References :

https://www.geeksforgeeks.org/socket-programming-in-cc-handling-multiple-clients-on-server-without-multi-threading/