Computer Networks Lab Assignment 2

Deadline: 03.03.21

Max Marks:20

Instructions:

- Please submit original work. Plagiarism/copying of any form, will deduct the mark.
- Submit all documents in google classroom.
- Save all submission files into a single folder and submit the compressed folder.
- Explain the working of functions in comments.
- Create a readme file explaining how to execute your code. (Terminal commands etc.)
- **Must follow the following naming convention:**
    - Name the ZIP file submission as, `Yourname_Rollnumber_Assignmentnumber.zip`

Task 1: C Program to reverse the string. [10 marks]

Build connection-oriented (TCP) client server model. Client sends the string to the server and server reverses the string sent by the client and sends it back to the client.

Action at Server :

1. Create a socket using *socket( )* system call..
2. Bind server's address and port using *bind( )* system call.
3. Convert the socket into a listening socket using *listen( )* system call.
4. Wait for client connection to complete using *accept( )* system call.
5. Receive the Client request using *recv()* system call which consist of the name of the command that is to be executed along with data parameters (if any)
6. The command is interpreted and executed.
7. On successful execution the result is passed back to the client by the server

Actions at Client

1. Create a socket.
2. Fill in the internet socket address structure (with server information).
3. Connect to server using connect system call.
4. The client passes the command and data parameters (if any) to the server.
5. Read the result sent by the server, write it to standard output.
6. Close the socket connection.

Sample Input: The Client sends the string "CNLAB"

Sample Output The string will get back as reverse "BALNC"

[10 marks]

Task 2-

Write two separate C programs, one for TCP server (handles requests for multiple users) and another one for clients.
Your server program will be a multi-process server that will "**fork**" a process for every new client it receives. Multiple clients should be able to simultaneously chat with the server.

The protocol between the client and server is as follows:
1. The client connects to the server, and then asks the user for input. The user enters a simple arithmetic expression string in postfix form (e.g., "1 2 +", "5 6 22.3 * +"). The user's input is sent to the server via the connected socket.
2. The server reads the user's input from the client socket, evaluates the postfix expression, and sends the result back to the client as well as writes the following in a file named "server_records.txt". [at the beginning create an empty file]
<client_id> <query> <answer> <time_elapsed>
3. The client should display the server's reply to the user, and prompt the user for the next input, until the user terminates the client program.

Server should be able to handle addition, multiplication, subtraction, and division operations (postfix form is space separated).

Sample test cases are:
User types: 1 2 +, server replies 3
User types: 2 3 *, server replies 6
User types: 4 7 3 + -, server replies -6
User types: 30 1.0 /, server replies 30.0

Below is a sample run of the client.
$ gcc client.c -o client
$ ./client
Connected to server
Please enter the message to the server: 22 44 +
Server replied: 66
Please enter the message to the server: 3 4 *
Server replied: 12