Assignment 3

AMAN KUMAR GUPTA
2018217

Implementation

The code is divided into 2 parts one is the server file and other is a client file which can be run multiple times.

I have used Socket to solve the reader writer problem.

Running the Chat-System(via Makefile

We can run the system by using the make command to run the server

And we can run multiple instances of the client file by using make client command.

Server.c code implementation

First, we create a socket using “server_sock = socket(AF_INET,SOCK_STREAM,0);” command and store it in a socket descriptor

Then we assign port no to the socket as well as the domain ip for the socket to communicate.

Then we bind the socket to the respective IP Address and the port decided above.

Then we put the socket into listen or passive mode so that it waits for the client to approach the server to make a connection.

Implementation of Server.c

Then, we enter an infinite loop which keeps accepting connection requests and adds new clients to the client linked list as well as an array for sending and receiving messages.

Then we create a thread for each client, where we receive requests from the clients.

Based on the request by the client there are 6 functions editspecific, enQueue, deQueue, displayall, displayspecific, isEmpty . editspecific is used to edit a certain element already present in queue.enQueue is used to normally enqueue the data in the queue,and dequeue is for dequeuing an element,displayall is for reading all messages in queue, and display specific is used to display specific data at an index in the queue,isempty is just for help.

Implementation of Server.c

I have used send and recv function calls to send and receive data on the socket from client to server and vice versa.

A string slicing function is also used for slicing the string at appropriate places due to the encoding of the messages.

Implementation of Client.c

We again create a socket and we give the same Port no and same domain IP Address used in the server.

Then I create a thread for parallel receiving of the messages from the server.

Then I show a menu and the client can choose from the listed options to use the chat system.

.

User Input

After running the makefile of both the serve and the client

The client is expected to enter his/her name and then he can start using the chat system as soon as the client file connects to the server.

The client can join as reader or writer.

For reader, the user has options of reading all messages in the queue, the user also has a option of reading a specific index in the queue

The reader can also dequeue messages from the queue

User Input

The writer can directly enqueue data into the queue.

The writer also has an option to edit some previous messages.

Both the reader and writer has an option of quitting and accessing the help menu wherever required.

Expected Output

Assuming there are no errors while running the chat system,

The user being a writer/reader can perform his/her basic operations as mentioned above and the respected output will be displayed on their terminal.

Error values

Perror has been used wherever there is a possibility of a failing of the command.

There is a brief description of the error as well attached with the perror as well to identify the error.

Errors for failing of sending/receiving messages via socket, creation of socket,binding of socket, and some input errors have been handled.

References

https://www.geeksforgeeks.org/socket-programming-cc/

https://www.geeksforgeeks.org/queue-linked-list-implementation/

https://github.com/codophobia/Multi-Client-Server-Chat/blob/master/(just to see the creation of a socket).