



# Assignment 2

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 implement the chat system.



# 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 3 functions `sendtoeveryone` , `sendtospecific` , `sendtoself` . `sendtoeveryone` function is used when the information should be sent to everyone, `sendtospecific` function is used when the information should be sent to a specific person, `sendtoself` function is used when the information is requested by the client for his own use.



# 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 send messages using the SEND command

The client can broadcast the message by typing "SEND \* (message)".

The client can send a message to a specific person by typing "SEND (client no of the expected user) (message)"





# User Input

Also, user can use the “HELP” command to see the Help menu.

User, can also type “USERS” to see active no of users connected to the server.



# Expected Output

Assuming there are no errors while running the chat system,

The user can send messages as mentioned in the above slides, also when a user receives a messages he/she will see that if the message sendd was send to him only or was send to all to differentiate between the different type of messages received on the server



# 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.



# References

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

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

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