

MULTI USER CHAT SYSTEM DOCUMENTATION

This is a multi user chat platform which has a server and multiple clients.

To compile the code, first compile the server by writing " gcc -pthread server.c -o server " . To start the server, write "./server". Similarly now, we can make several clients by compiling client code by " gcc -pthread client.c -o client ". To start a client, run the following command "./client

name_of_client". Here name_of_client refers to the name and can be chosen by us.

The 2 keywords used to send messages are:

- ❑ **SEND client_number my_msg** : This sends a personal message to the specific client number. For instance: SEND 4 hey there
- ❑ **ALL my_msg** : This sends a broadcast message to everyone (i.e every client connected). For instance ALL hello

First connection is established between the server and client through UNIX DOMAIN SOCKET. The message which is sent through the client is appended with the client's name. The message is received by the server and it is checked for the keywords(ALL/SEND)in our case, and different functions are called respectively which in turn sends the message to the respective client.

Personal messages and Global(i.e broadcasted) messages can also be differentiated from the receiving client side.

When the message is received, the name of the sender is also shown explicitly to avoid confusion.

The output message format is "**Global/personal message from name_of_sender: message** "

Errors handled:

Server side:

- Binding error

- Listening error
- Accepting error
- Socket error
- Sending and receiving error

Client side:

- Connection failure
- Sending and receiving error
- Trying to send messages to self
- Trying to send messages to an undefined client
- Input format other than SEND/ALL will throw an error message.
- Missing argument in case of SEND command(eg. If client_number is inputted as a string)