

## CHAT SERVER

A concurrent server which can support upto 100 clients simultaneously. Server is continuously listening to the client requests and asks for the USERNAME as soon as a client makes request to join the Chat Server. A new entry is made into the Server if a new UserName is found, but if an old Username is found then it is recognised and the old connection is setup again.

Commands used by user:

- 1) **exit** => to go offline
- 2) **get\_users** => to know which users are currently online and offline ( o – online ; x - offline )
- 3) **<username>: <message>** => to send message to any of the specified user in the get\_user list either online or offline. (<username> and : are written without spaces)
- 4) **broad: <message>** => to send message to all the users ever registered in the chat system
- 5) **mk\_grp:<group\_name>:<username>:<username>.....**(max 32 number of users) => to make a group of users. Later this group name can be used to send message to the specified users. (all without spaces).
- 6) **<group\_name>: <message>** => to send message to all the users in the group either online or offline. (<groupname> and : are written without spaces).

(a) to join and leave the chat system

run the ./client command on the terminal and type the username to join the chat system.  
Send exit to go offline.

(b) to get the list of online and offline users

use command 2) get\_users

(c) to send message to any users

to send to all users use command 4) eg;- broad: hello

to send to a particular user use command 3) eg;- user1: hello

to send to a group of users first create a group by command 5)

eg;- mk\_grp:grp1:user1:user2:user3 then type grp1: hello and the message is delivered to all the users in the group.

(d) server creating a new process for every client connection and use message queue for communicating among the child processes.

Server has one global queue storing structure **msgform**. All messages sent across the server are stored on the message queue with the sender name and receiver\_id. All the child process are continuously polling for their respective clients messages on the queue. If found it fetched and does a send call to the client.

(e) chat groups can be created

by using command 5) and communication can be done by any client on the chat group by command 6).