TCP Simple Broadcast Chat Server and Client:

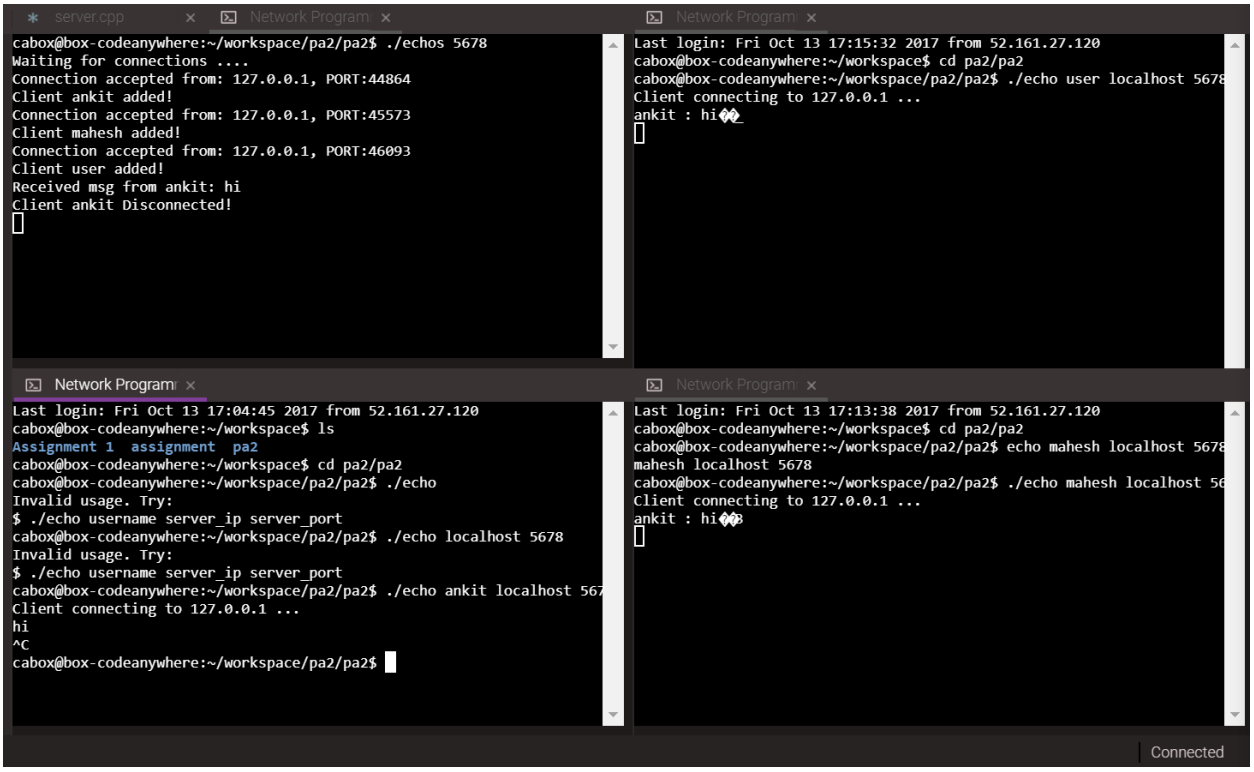
In this programming assignment SBCP(Simple Mail Broadcast Protocol) is used to communicate between server and client. SBCP defines a packet structure which can be used for different purposes: joining chat session, sending message, making a query to the server, like the list of online clients, type of message, username.

Whenever a client has to join the chat session, it sends a join request to the server. Server looks into the details of all the present clients. If a client with the same username is present in the system, the server denies the connection and sends a NACK. If the username is not present already. server accepts the join request and user is allowed to participate in the chat session. Server sends the list of clienst present in the system. Clients have the capability to check if a user is online. When the client see that an another client has gone offline, it notifies to the server. Upon receiving this information from the client server deletes the record of that user, frees up the username. This makes the usernames re-usuable. Client uses I/O multiplexing to handle simultaneous send and receive.

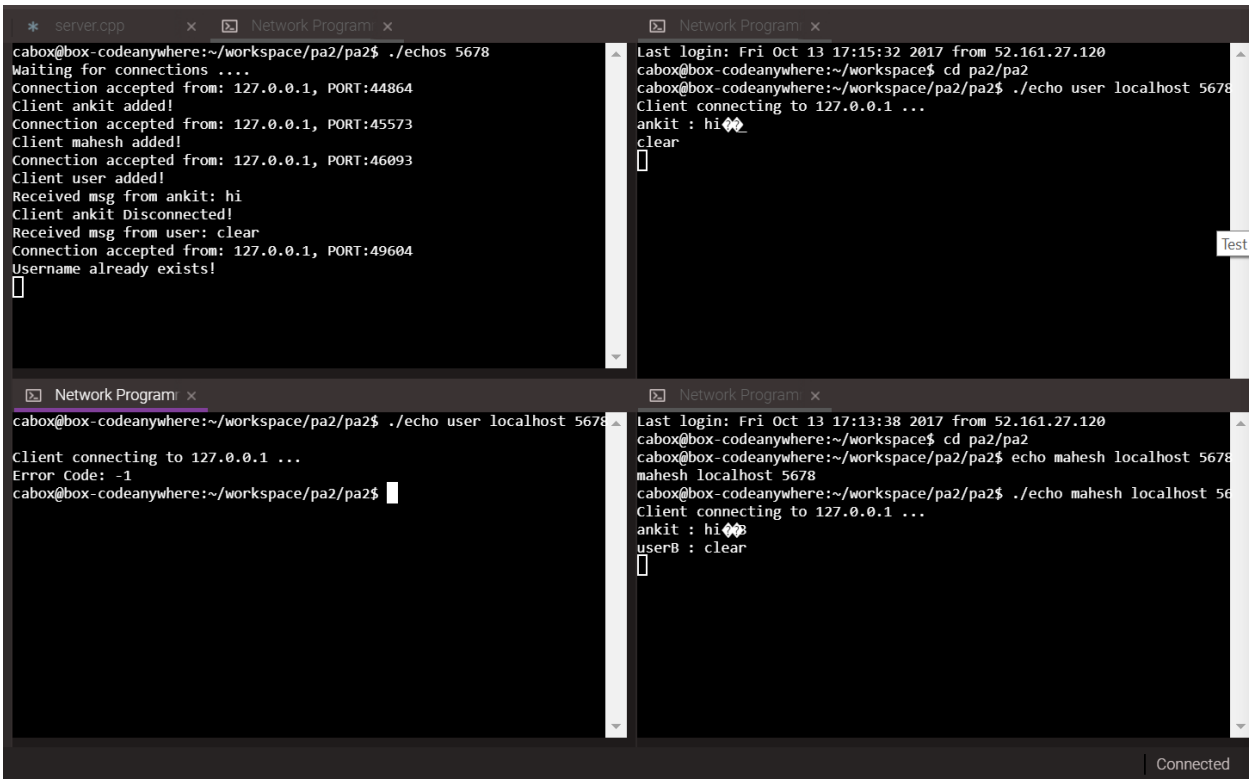
Server works as the central agent, it forwards the message received from a client to everyone except the client who sent the message. Server only accepts the request from the unknown users. When a user makes a request to join, server checks if a client with the same username is already present. If not server adds it to the list of clients and sends it to everyone. If a user exits the chat session unceremoniously, the server detects that and removes that client from the list. Server uses the I/O multiplexing to hear from the listening ports and connected clients simultaneously.

Test Cases:

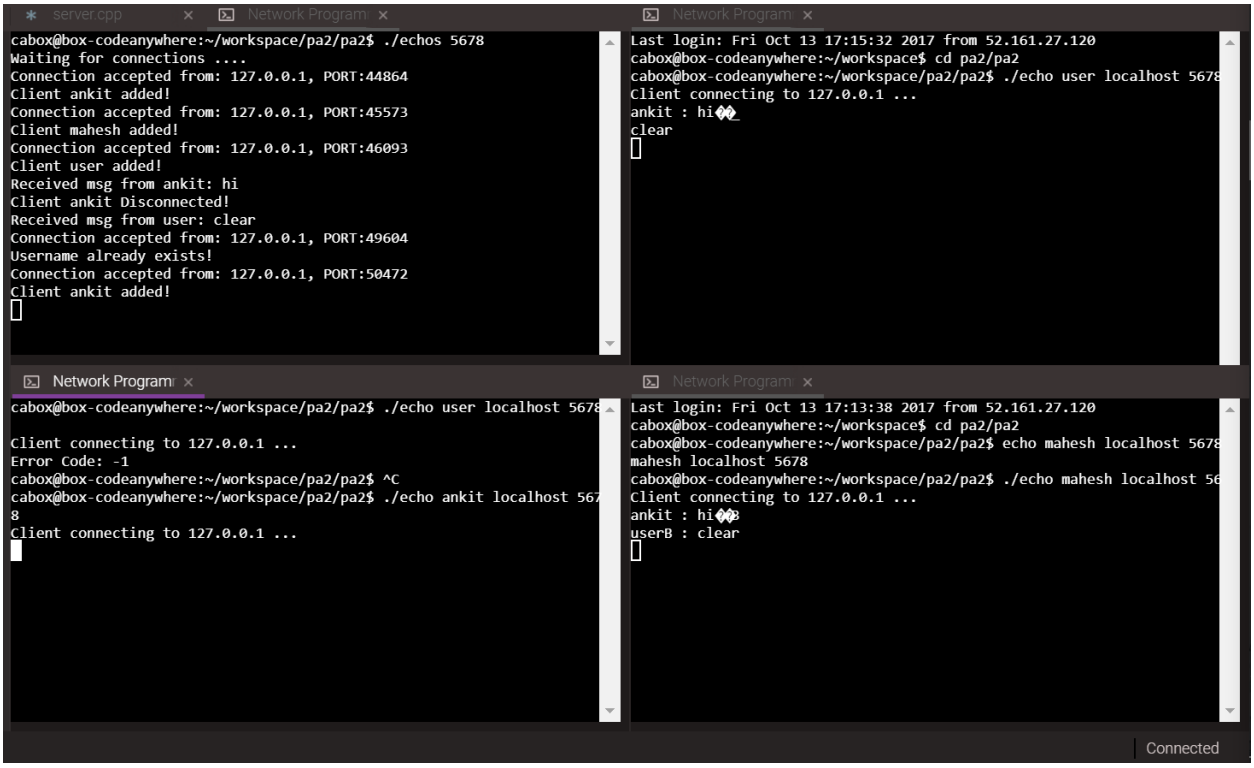
1. Normal operation of the chat client with three clients connected:



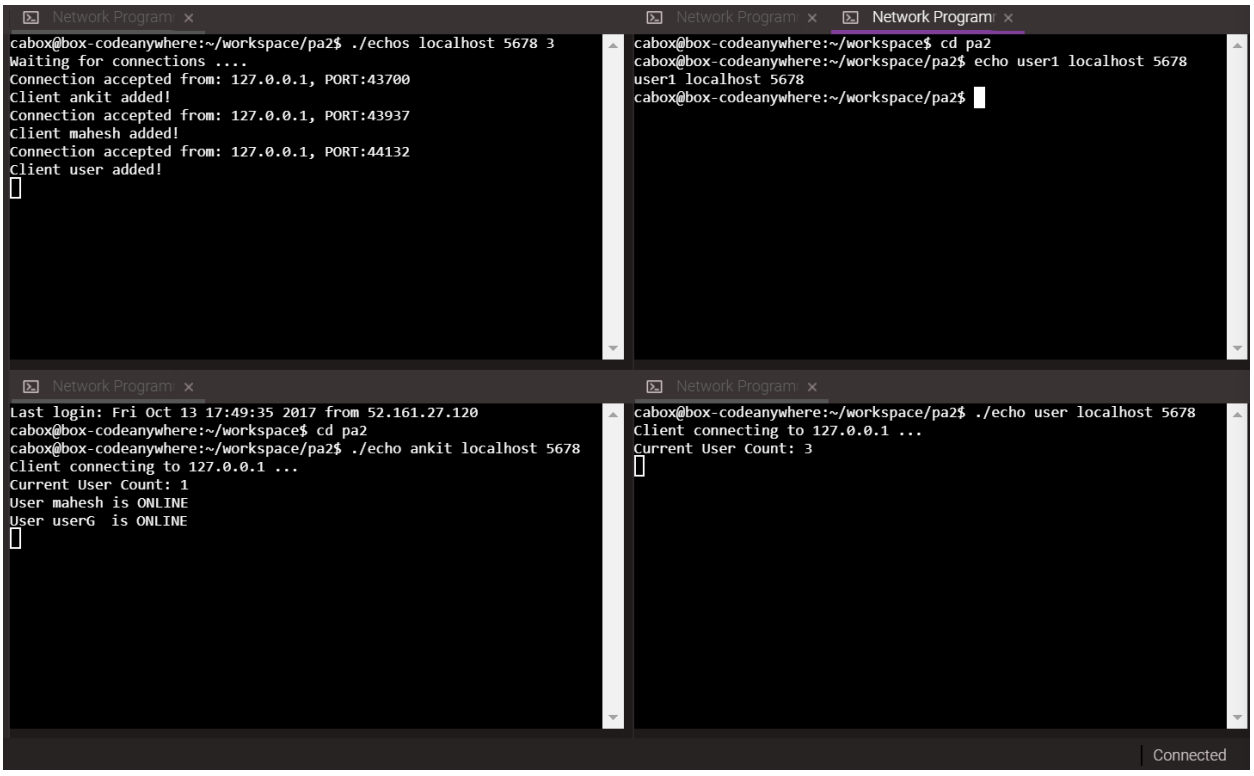
2. Server rejects a client with a duplicate username:



3. Server allows a previously used username to be reused:



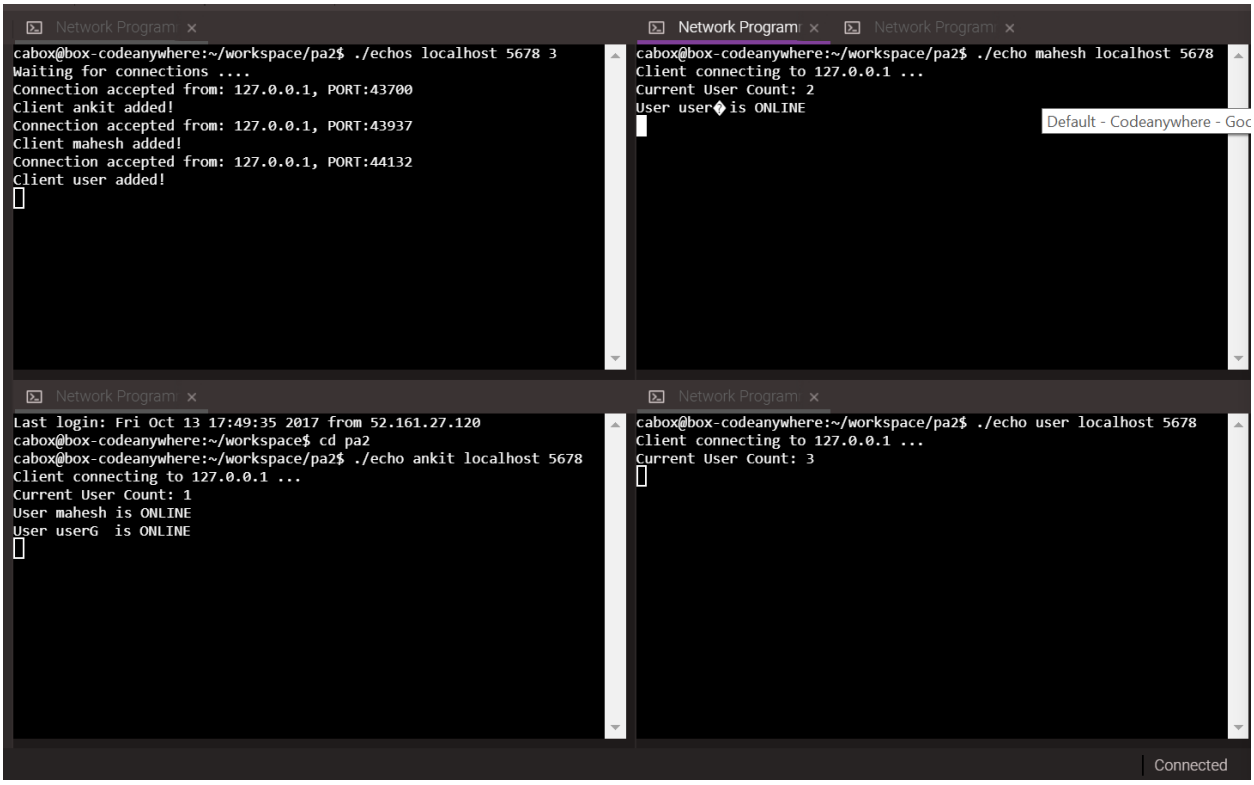
4. Server rejects the client because it exceeds the maximum number of clients allowed:



5. Separate test cases for the bonus features:

(a) IPv6 included

(b)


Network Program
cabox@box-codeanywhere:~/workspace/pa2$ ./echos localhost 5678 3
Waiting for connections ....
Connection accepted from: 127.0.0.1, PORT:43700
Client ankit added!
Connection accepted from: 127.0.0.1, PORT:43937
Client mahesh added!
Connection accepted from: 127.0.0.1, PORT:44132
Client user added!
Network Program
Network Program
cabox@box-codeanywhere:~/workspace/pa2$ ./echo mahesh localhost 5678
Client connecting to 127.0.0.1 ...
Current User Count: 2
User user is ONLINE
Default - Codeanywhere - Goo
Network Program
Last login: Fri Oct 13 17:49:35 2017 from 52.161.27.120
cabox@box-codeanywhere:~/workspace$ cd pa2
cabox@box-codeanywhere:~/workspace/pa2$ ./echo ankit localhost 5678
Client connecting to 127.0.0.1 ...
Current User Count: 1
User mahesh is ONLINE
User userG is ONLINE
Network Program
cabox@box-codeanywhere:~/workspace/pa2$ ./echo user localhost 5678
Client connecting to 127.0.0.1 ...
Current User Count: 3
Connected