CO318 Project I: Multiuser Chartroom Server

Project Report

Group 18

- **E/10/049**
- **E/10/170**

Objective

Design and code a chat server capable of handling multiple clients using multithreading.

Overview

Here we designed and code for chat server capable of handling multiple clients using multithreading. When one client sends a message it going to be broadcast to all clients connected to the server at the same time.

The client sends what the user writes in console and the server send the same message back. We just need the server to send the same message to every thread (client) connected. Generally we set up a server and client completely separate.

The client should connect to the server (ip+port) and the server should be listening on the given port. Once a connection is established we have a socket connecting the two programs.

To design a client and server module using socket programming we categorize our major tasks which performs following:

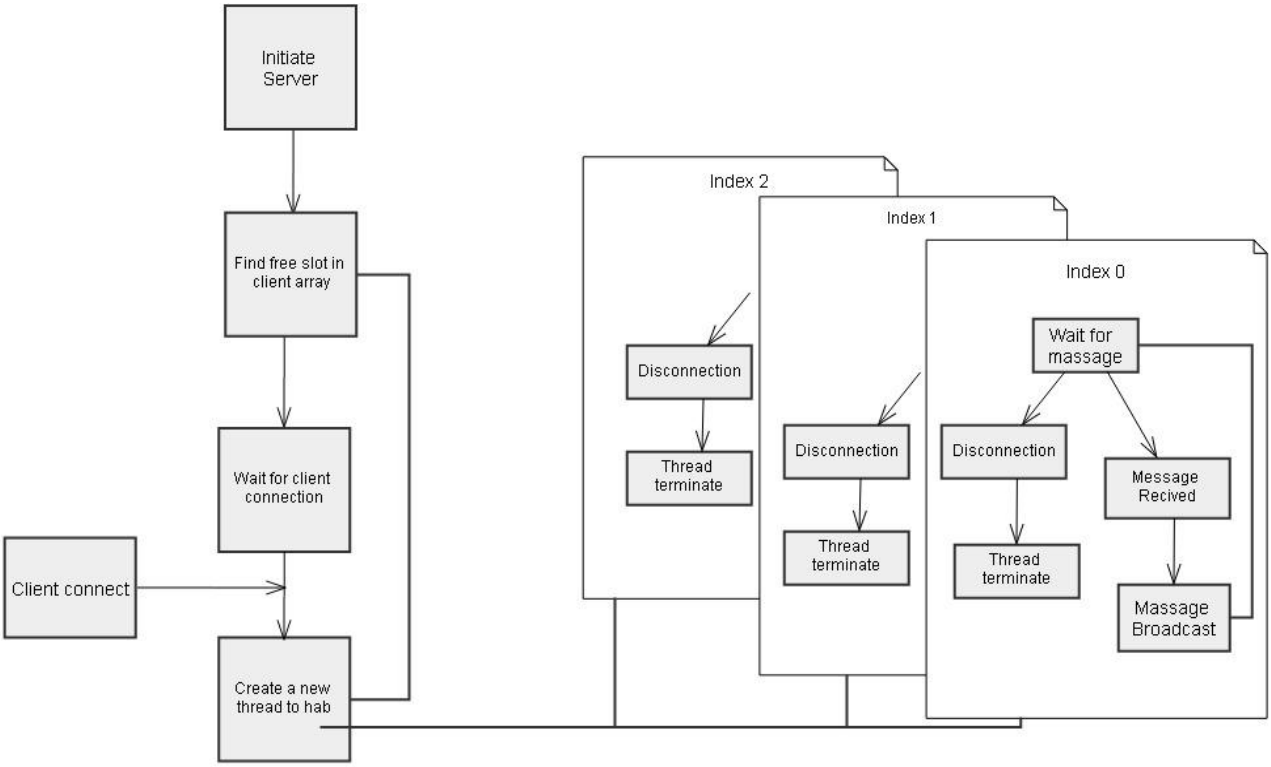

Figure : 1 Flow chart of the algorithm regarding to our program

Client handling

Here we keep client array for count the number of clients connected to the server. As long as it is not fully filled client can make connection with server but if client array already filled client has to wait till server done with already connected client. When new clients connected to the server, server will accept his request, after that server will create new thread and let that thread to deal with newly connected client.

Threading structure

Here we dynamically create and destroy threads to handle client communication. As we explained above we create new thread when client wanted to connect to the server after client disconnected from server we are going to terminate and free the memory related to that thread

Synchronization considerations

To ensure there are no race conditions between threads on our server we used mutexes. When broadcasting the massages we put a lock and release it when broadcast is over. Because of that reason multiple threads cannot broadcast at the same time.

Resource handling

We allocate memory for each client in the heap using malloc function. After client disconnected form the server we free that space.