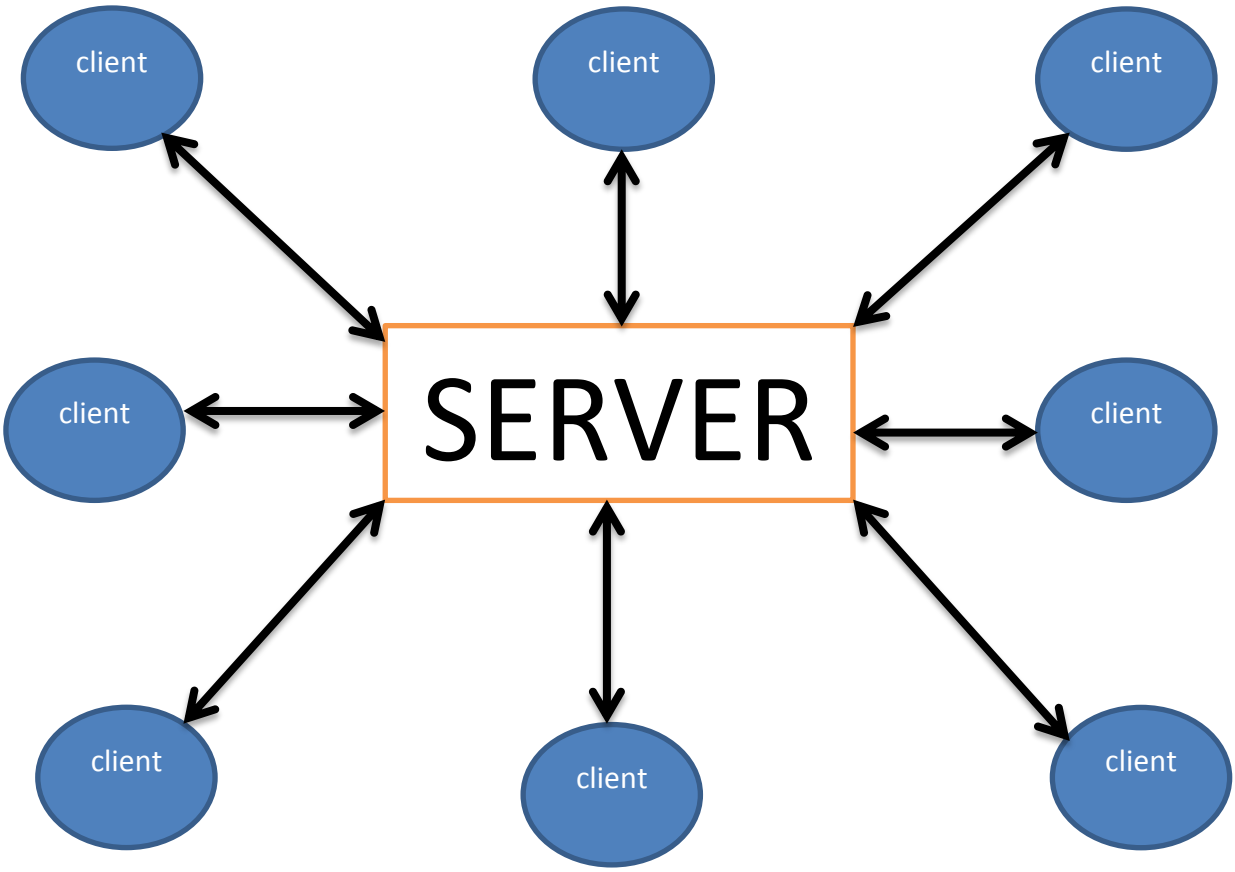
E/10/376

E/10/169

CHATROOM PROJECT



We make this chat room for chat with clients. There are one server and clients. One client send massage server, then server sends this massage other clients.

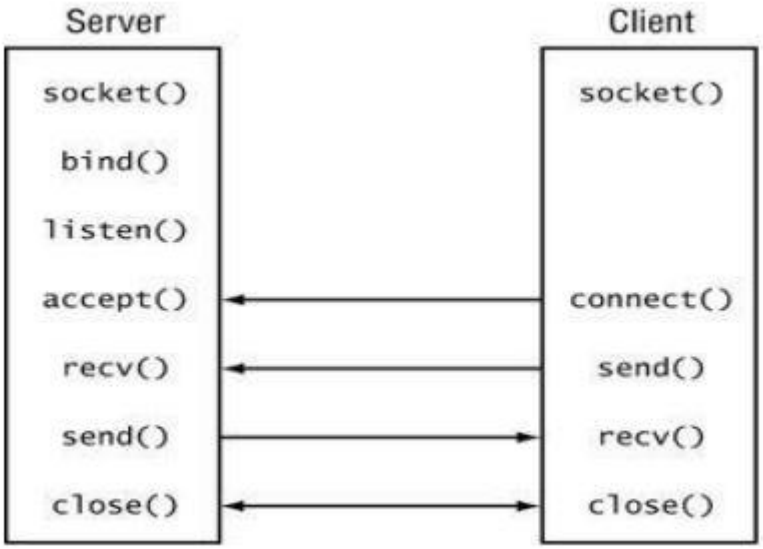
The client handling part implemented to perfectly handle clients by the server, gracefully able to handle client connections and client disconnections at any time. This chat server project has done to do text message passing only.

The server creates threads, after disconnect clients threads close.

Mutex have used to reduce trace conditions between threads. Server is able to handle more clients at approximately same time. In this case have used thread lock and thread unlock.

We used TCP connection between server and clients. TCP protocol is a connection-oriented socket, the TCP protocol is used to establish a session between two IP address endpoints. There is a fair amount of overhead involved with establishing the connection, but once it is established, the data can be reliably transferred between the devices.

Creat TCP connection



Structure of the code

We used array get details of clients. ----→ static client_t *clients[MAXCLIENTS];

After create node then store the details of clients.

```
typedef struct {
        int index;
        int sd;
        pthread_t tid;
        char name[100];
```

} client_t;

index: position in the clients array

sd: socket descriptor for this client

tid: ID of the thread handling this client

name: hostname/IP of the client

- **int next_free()**

  This function is use for fined next free place of the clients array.
  If there is no space for joint client then return -1.

- **void * handle_client (void *arg)**

  Read massage for clients. and sent it broad cast function.
  Check client exit or not. If exit client remove details of that client and null the buffer. Before sending massage to broadcast function use thread lock. After run broadcast function threads unlock.

- **void broadcast_msg(char *message)**

  Send a received message to each client currently connected.

- **int setup_server(void)**

  in this function initialize socket structure and bind the host address using bind() call.

- **void cleanup (int signal)**

  if we press ctrl+c in server, call clean up function.
  Function check coming signal and close the listnfd.

- **int main(int argc, char *argv[])**

  In this function install signal handler for SIGINT → signal(SIGINT,cleanup);
  And allocate client structure in this function. After allocating create p thread for each client.