

IT694 -Computer Network Lab 3

Group ID -11

Rutvi Bhatt - 202112068

Atharva Vaze - 202112015

Saksham Jain - 202112029

Process of designing the code:

1. Design the application protocol between client and server.
2. Decided on using non blocking socket communication
3. Design the required DS
4. Designed the error handling for graceful execution.

Algorithm:

1. Setup the Data Structures
2. Create the non blocking FD pairs
3. setup socket communication between server and client
4. use select to read changes in client
5. use select to understand the client messages [connect when ready to send]
6. process each event as per design protocol.

SERVER CODE

```
/*  
*****  
Group [11]  
Rutvi Bhatt[202112068]  
Atharva Vaze [202112015]  
Saksham Jain[202112029]  
*****/  
#include<stdio.h>  
#include<stdlib.h>  
#include<string.h>  
#include<unistd.h>  
#include<sys/types.h>  
#include<sys/socket.h>  
#include<sys/time.h>  
#include<errno.h>  
#include<netinet/in.h>  
#include<arpa/inet.h>  
#include<time.h>  
#include<ctype.h>  
//storing message text information  
struct message{  
    int from;  
    int to;  
    char msg[1024];  
};  
//data struct that stored the group info  
struct group{  
    int admin;  
    int no_of_clients;  
    int members[10];  
};  
*/
```

```

int group_id;
};
//storing temporary group information
struct create_grp{
int g_id;
int members[10];
int processed[10];
int admin;
int no_of_clients;
int no_of_rplys;
int dec_count;
};
//main function
int main(int argc,char *argv[]){
struct group *grps[10];
struct create_grp *grp_reqs[10];
int group_count=1,req=0;
int serv_fd,port,client_port,sock_activity,client_socket;
int max_sd,total_clients = 5,sd,online_clients=0;
char buf[1024];
int opt = 1;
struct sockaddr_in serv_addr;
//checking required argument provided or not
if(argc < 2){
printf("ERROR: please provide port no..\n");
exit(0);
}
//socket descriptors set
fd_set readfds;
int clients[5],i;
//initialized all clients with 0
for(i=0;i<total_clients;i++){
clients[i] = -1;
}
for(i=0;i<10;i++){
grps[i] = NULL;
grp_reqs[i] = NULL;
}
//creating socket
serv_fd = socket(AF_INET,SOCK_STREAM,0);
if(serv_fd < 0){
perror("ERROR: Creating Server Socket..\n");
exit(0);
}
printf("Succussfully Server Socket is created.. \n");
//set server socket to allow multiple connections
if( setsockopt(serv_fd, SOL_SOCKET, SO_REUSEADDR,(char *)&opt,sizeof(opt)) < 0 ) {
perror("setsockopt");
exit(EXIT_FAILURE);
}
bzero(&serv_addr,sizeof(serv_addr));
//assigning PORT and IP
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
port = atoi(argv[1]);
serv_addr.sin_port = htons(port);
//binding to the given port
int bind_val = bind(serv_fd,(struct sockaddr*)&serv_addr,sizeof(serv_addr));
if(bind_val < 0){
printf("ERROR: While binding to port %d\n",port);
exit(0);
}
printf("Server Socket successfully binded...\n");
//sever listening
if(listen(serv_fd,5) != 0){
printf("ERROR: while listen...\n");
exit(0);
}
printf("Server listening..\n");
socklen_t len = sizeof(serv_addr);
printf("Waiting for connection...\n");
//infinite loop to get multiple clients connected
while(1){
struct message *msg = NULL;
//clearing readfds set
FD_ZERO(&readfds);

```

```

//adding serv_fd to readfds set
FD_SET(serv_fd,&readfds);
max_sd = serv_fd;
for(i=0;i<total_clients;i++){
//socket descriptor
sd = clients[i];
//cheching for socket decriptor valid or not
if(sd > 0){
FD_SET(sd,&readfds);
}
if(sd > max_sd){
max_sd = sd;
}
}
//wait for an activity on one of the sockets
sock_activity = select(max_sd+1,&readfds,NULL,NULL,NULL);
if(sock_activity < 0){
perror("ERROR:in Select..\n");
exit(0);
}
//if something happens on server socket
if(FD_ISSET(serv_fd,&readfds)){
//accepting client request
client_socket = accept(serv_fd,(struct sockaddr*)&serv_addr,&len);
printf("socket desrcipoter %d\n",client_socket);
if(client_socket < 0){
perror("ERROR: While accepting client requests..\n");
exit(0);
}
if(online_clients < total_clients){
printf("Connection accepted from %s:%d\n",inet_ntoa(serv_addr.sin_addr),ntohs(serv_addr.sin_port));
bzero(buf,sizeof(buf));
snprintf(buf, sizeof(buf), "Succussfully connected to server...\nWelcome to our Chat Application...\nYour user id is %05d\n",client_
send(client_socket,buf,sizeof(buf),0);
online_clients++;
printf("online = %d total = %d\n",online_clients,total_clients);
for(i=0;i<total_clients;i++){
if(clients[i] == -1){
clients[i] = client_socket;
break;
}
}
}
else{
send(client_socket,"Connection Limit Exceeded !!\n",strlen("Connection Limit Exceeded !!\n"),0);
}
}
for(i=0;i<total_clients;i++){
sd = clients[i];
if(FD_ISSET(sd,&readfds)){
bzero(buf,sizeof(buf));
//reading from client
int read_bytes = read(sd,buf,1024);
char buf2[1024];
bzero(buf2,sizeof(buf2));
strcpy(buf2,buf);
printf("From client%s\n",buf);
char* start_token = '\0';
if(strlen(buf2) > 0){
start_token = strtok(buf2, " ");
if(start_token[strlen(start_token)-1] == '\n'){
start_token[strlen(start_token)-1] = '\0';
}
}
if(read_bytes <= 0){
getpeername(sd,(struct sockaddr*)&serv_addr,&len);
printf("Host Address { %s:%d } is Disconnected from Server..\n",inet_ntoa(serv_addr.sin_addr),ntohs(serv_addr.sin_port));
for(int j=0;j<total_clients;j++){
if(clients[j] != -1 && clients[j] != sd){
char msg[256];
sprintf(msg,"%05d is going to exit..\n",sd);
//printf("%s",msg);
send(clients[j],msg,sizeof(msg),0);
}
}
printf("%05d is going to exit..\n",sd);
}
}

```

```

for(int j=0;j<10;j++){
if(grps[j] != NULL){
for(int k=0;k<grps[j]->no_of_clients;k++){
if(grps[j]->members[k] == sd){
printf("%05d is quitting from Group %05d.\n",sd,grps[j]->group_id);
grps[j]->members[k] = -1;
//send(sd,rply,sizeof(rply),0);
break;
}
}
}
}
close(sd);
clients[i] = -1;
online_clients--;
}
else if(strcmp(start_token,">MYGROUPS") == 0){
char rply[256];
char temp[256];
int check = 0;
bzero(rply,sizeof(rply));
sprintf(rply,"Server: Groups having %05d has a member\n",sd);
for(int j=0;j<10;j++){
if(grps[j] != NULL){
for(int k=0;k<grps[j]->no_of_clients;k++){
if(grps[j]->members[k] == sd){
bzero(temp,sizeof(temp));
sprintf(temp,"%05d\n",grps[j]->group_id);
printf("k %d temp %s\n",k,temp);
strcat(rply,temp);
check++;
}
}
}
}
if(check == 0){
bzero(rply,sizeof(rply));
sprintf(rply,"Server: No Groups having %05d has a member\n",sd);
send(sd,rply,sizeof(rply),0);
}
else
send(sd,rply,sizeof(rply),0);
printf("%s",rply);
}
else if(strcmp(start_token,">SEND") == 0){
msg = (struct message*)malloc(sizeof(struct message));
msg->from = sd;
char* token = strtok(NULL, " ");
//token = strtok(NULL, " ");
int start_index = strlen(start_token) + strlen(token)+2,check=0;
char rply[1024];
bzero(rply,sizeof(rply));
if(token != NULL){
msg->to = atoi(token);
for(int j=0;j<total_clients;j++){
if(clients[j] == msg->to){
check =1;
break;
}
}
}
else{
sprintf(rply,"Invalid Format...\n");
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
if(check == 0){
sprintf(rply,"toination Not Exists...\n");
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
strcpy(msg->msg,buf+start_index);
bzero(buf,sizeof(buf));
sprintf(buf,"%05d: ",msg->from);
strcat(buf,msg->msg);
printf("sending : %s\n",buf);

```

```

for(int j=0;j<total_clients;j++){
if(cclients[j] == msg->to){
send(cclients[j],buf,sizeof(buf),0);
char reply[256];
sprintf(reply,"Message sent to %05d\n",msg->to);
printf("%s",rply);
send(sd,reply,sizeof(reply),0);
}
}
free(msg);
}
else if(strcmp(start_token,">BROADCAST") == 0){
msg = (struct message*)malloc(sizeof(struct message));
msg->from = sd;
strcpy(msg->msg,buf + 10);
bzero(buf,sizeof(buf));
sprintf(buf,"%05d: ",msg->from);
strcat(buf,msg->msg);
printf("%s\n",buf);
for(int j=0;j<total_clients;j++){
if(cclients[j] != sd){
send(cclients[j],buf,sizeof(buf),0);
}
}
char reply[256];
sprintf(reply,"Message broadcasted to all Active clients..!\n");
//printf("%s",reply);
send(sd,reply,sizeof(reply),0);
free(msg);
}
else if(strcmp(start_token,">MAKEGROUP") == 0){
struct group *new_group = (struct group*)malloc(sizeof(struct group));
new_group->no_of_clients = 0;
new_group->admin = sd;
new_group->group_id = group_count;
char grp_clients[1024];
bzero(grp_clients,sizeof(grp_clients));
strcpy(grp_clients,buf+10);
char* token = strtok(grp_clients, " ");
char reply[256],create = 0;
while(token != NULL){
int id = atoi(token);
int check = 0;
for(int j=0;j<total_clients;j++){
if(id == cclients[j] && id != sd){
new_group->members[new_group->no_of_clients++] = id;
bzero(reply,sizeof(reply));
sprintf(reply,"You were added to group with id %05d\n",new_group->group_id);
printf("%05d added to group %05d\n",id,new_group->group_id);
send(id,reply,sizeof(reply),0);
check = 1;
create++;
break;
}
}
if(check == 0 && id != sd){
bzero(reply,sizeof(reply));
sprintf(reply,"No client existed with id %s..!",token);
printf("%s",reply);
send(sd,reply,sizeof(reply),0);
}
token = strtok(NULL, " ");
}
if(create == 0){
bzero(reply,sizeof(reply));
sprintf(reply,"Group %05d is not created..!\n",new_group->group_id);
printf("%s",reply);
send(sd,reply,sizeof(reply),0);
}
else{
bzero(reply,sizeof(reply));
sprintf(reply,"Group %05d created..!\n",new_group->group_id);
printf("%s",reply);
new_group->members[new_group->no_of_clients++] = sd;
send(sd,reply,sizeof(reply),0);
grps[group_count++] = new_group;
}
}

```

```

}
}
else if(strcmp(start_token,">LISTGROUPS") == 0){
bzero(buf,sizeof(buf));
strcat(buf,"Server:\n");
char grp_id[256];
int check = 0;
for(int j=0;j<10;j++){
if(grps[j] != NULL){
check = 1;
sprintf(grp_id,"%05d\n",grps[j]->group_id);
strcat(buf,grp_id);
}
}
if(check == 0){
send(sd,"No active groups..!\n",strlen("No active groups..!\n"),0);
printf("No active groups..!\n");
}
else{
send(sd,buf,sizeof(buf),0);
printf("%s",buf);
}
}
else if(strcmp(start_token,">ACTIVE") == 0){
start_token = strtok(NULL, " ");
bzero(buf,sizeof(buf));
if(start_token != NULL){
send(sd,"Invalid format..!\n",strlen("Invalid format..!\n"),0);
printf("Invalid format..!\n");
}
else{
strcat(buf,"Server:\n");
char client_id[256];
for(int j=0;j<total_clients;j++){
if(clients[j] == -1)
continue;
else if(clients[j] != sd){
sprintf(client_id, "%05d\n", clients[j]);
strcat(buf,client_id);
}
}
sprintf(client_id,"%05d * That is you\n",sd);
strcat(buf,client_id);
printf("Active clients:\n%s\n",buf);
send(sd,buf,sizeof(buf),0);
}
}
else if((strcmp(start_token,">JOINGROUP") == 0)){
char* token = strtok(buf, " ");
token = strtok(NULL, " ");
int grp_id=0,check=0;
char rply[256];
while(token != NULL){
if(check != 0){
check = -1;
send(sd,"Invalid format..!",strlen("Invalid format..!"),0);
printf("Invalid format..!\n");
break;
}
grp_id = atoi(token);
check++;
token = strtok(NULL, " ");
}
if(grp_id > group_count || grp_id == 0){
bzero(rply,sizeof(rply));
sprintf(rply,"Invalid group id..!\n");
printf("Invalid group id..!\n");
send(sd,rply,sizeof(rply),0);
}
else if(check == 1){
for(int j=0;j<10;j++){
if(grps[j] != NULL){
if(grp_id == grps[j]->group_id){
for(int k=0;k<grps[j]->no_of_clients;k++){
if(grps[j]->members[k] == sd){
bzero(rply,sizeof(rply));

```

```

sprintf(rply,"You are already in group %05d\n",grp_id);
printf("%05d already in group %05d\n",sd,grp_id);
send(sd,rply,sizeof(rply),0);
check = 0;
break;
}
}
}
}
}
if(check == 1){
int temp = 0;
for(int l=0;l<10;l++){
if(grp_reqs[l] != NULL && grp_reqs[l]->g_id == grp_id){
for(int k=0;k<grp_reqs[l]->no_of_clients;k++){
if(grp_reqs[l]->members[k] == sd){
bzero(rply,sizeof(rply));
if(grp_reqs[l]->processed[k] == -8888){
sprintf(rply,"already declined..!\n");
send(sd,rply,sizeof(rply),0);
}
else if(grp_reqs[l]->processed[k] == -8989){
sprintf(rply,"waitig for others to accept\n");
printf("waitig for others to accept\n");
send(sd,rply,sizeof(rply),0);
}
else{
grp_reqs[l]->no_of_rplys++;
grp_reqs[l]->processed[k] = -8989;
sprintf(rply,"You are request for being added to group %05d is processed; Waiting for others to join..!\n",grp_id);
printf("%05d %s",sd,rply);
send(sd,rply,sizeof(rply),0);
bzero(rply,sizeof(rply));
sprintf(rply,"%05d has accepted your request for joining the group..!\n",sd);
send(grp_reqs[l]->admin,rply,sizeof(rply),0);
}
check =0;
break;
}
}
if(grp_reqs[l]->no_of_rplys + grp_reqs[l]->dec_count == grp_reqs[l]->no_of_clients){
for(int k=0;k<10;k++){
if(grps[k] == NULL){
struct group *add = (struct group*)malloc(sizeof(struct group));
grps[k] = add;
grps[k]->admin = grp_reqs[l]->admin;
grps[k]->no_of_clients = grp_reqs[l]->no_of_clients;
grps[k]->group_id = grp_reqs[l]->g_id;
for(int m=0;m<grps[k]->no_of_clients;m++){
if(grp_reqs[l]->members[m] != -1){
grps[k]->members[m] = grp_reqs[l]->members[m];
bzero(rply,sizeof(rply));
sprintf(rply,"You were added to the group %05d\n",grps[k]->group_id);
printf("%05d %s",grps[k]->members[m],rply);
send(grps[k]->members[m],rply,sizeof(rply),0);
}
}
//admin of the group is created
bzero(rply,sizeof(rply));
sprintf(rply,"Group %05d is created..!\n",grp_reqs[l]->g_id);
printf("%s",rply);
grps[k]->members[grps[k]->no_of_clients++] = grp_reqs[l]->admin;
send(grp_reqs[l]->admin,rply,sizeof(rply),0);
break;
}
}
}
}
}
}
if(check == 1){
bzero(rply,sizeof(rply));
sprintf(rply,"Sorry! You don't have a request to join the group %05d\n",grp_id);
send(sd,rply,sizeof(rply),0);
}
}
}

```

```

}
else if(strcmp(start_token,">SENDGROUP") == 0){
char temp[1024];
strncpy(temp,buf,strlen(buf));
char* token = strtok(NULL, " ");
if(token!= NULL){
int grp_id = atoi(token);
int start_index = strlen(start_token) + strlen(token) + 2;
printf("grp%s\n",token);
char msg[1024];
bzero(msg,sizeof(msg));
strcpy(msg,buf+start_index);
bzero(buf,sizeof(buf));
sprintf(buf,"%05d:(group) ",grp_id);
strcat(buf,msg);
//strcat(buf,msg);
int check = 0;
char rply[1024];
for(int j=0;j<group_count;j++){
if(grps[j] != NULL && grp_id == grps[j]->group_id){
for(int k = 0;k<grps[j]->no_of_clients;k++){
if(sd == grps[j]->members[k]){
check = 1;
break;
}
}
}
if(check == 1){
for(int k = 0;k<grps[j]->no_of_clients;k++){
printf("Sending Message...!\n");
send(grps[j]->members[k],buf,sizeof(buf),0);
}
bzero(rply,sizeof(rply));
sprintf(rply,"Message delivered to all members in group %05d..!\n",grp_id);
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
else{
send(sd,"You are not in the group, you cannot send a message\n",strlen("You are not in the group\n"),0);
check = 1;
}
break;
}
}
if(check == 0){
bzero(rply,sizeof(rply));
//printf("no grp\n");
sprintf(rply,"No group is existed with id %s\n",token);
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
}
else{
printf("Invalide format..!\n");
send(sd,"Invalide format..!\n",strlen("Invalide format..!\n"),0);
}
}
else if(strcmp(start_token,">MAKEGROUPREQUEST") == 0){
char* token = strtok(buf, " ");
token = strtok(NULL, " ");
if(token == NULL){
printf("INVALID format ..!\n");
send(sd,"INVALID format ..!\n",strlen("INVALID formta ..!\n"),0);
}
else{
int client_id,check = 0;
char rply[1024];
struct create_grp *grp_req = (struct create_grp*)malloc(sizeof(struct create_grp));
grp_req->admin = sd;
grp_req->no_of_clients = 0;
int create_req = 0,count=0;
grp_req->g_id = group_count++;
while(token != NULL){
check = 0;
client_id = atoi(token);
for(int j=0;j<total_clients;j++){
if(clients[j] == client_id && client_id != sd){

```



```
bzero(rply,sizeof(rply));
sprintf(rply,"%05d is requesting you to be in the group %05d...\n",sd,grp_req->g_id);
send(clients[j],rply,sizeof(rply),0);
bzero(rply,sizeof(rply));
sprintf(rply,"add request send to %05d...\n",client_id);
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
grp_req->members[grp_req->no_of_clients++] = client_id;
check = 1;
create_req = 1;
break;
}
}
if(client_id == sd){
bzero(rply,sizeof(rply));
sprintf(rply,"Admin is automatically added to group");
send(sd,rply,sizeof(rply),0);
}
else if(check == 0){
bzero(rply,sizeof(rply));
sprintf(rply,"No client existed with id %s..!",token);
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
token = strtok(NULL, " ");
}
if(create_req == 0){
free(grp_req);
}
else if(grp_req->no_of_clients == 0){
bzero(rply,sizeof(rply));
sprintf(rply,"Please provide atleast one client without admin to create group..\n");
send(sd,rply,sizeof(rply),0);
}
else{
grp_req->no_of_rplys = 0;
grp_req->dec_count = 0;
grp_reqs[req] = grp_req;
req++;
}
}
}
else if(strcmp(start_token,>DECLINEGROUP") == 0){
char* token = strtok(buf, " ");
token = strtok(NULL, " ");
int grp_id=0,check=0;
char rply[256];
while(token != NULL){
if(check != 0){
check = -1;
send(sd,"Invalid format...\n",strlen("Invalid format...\n"),0);
printf("Invalid format...\n");
break;
}
grp_id = atoi(token);
check++;
token = strtok(NULL, " ");
}
if(grp_id > group_count || grp_id == 0){
bzero(rply,sizeof(rply));
sprintf(rply,"Invalid group id...\n");
printf("%s",rply);
send(sd,rply,sizeof(rply),0);
}
else if(check == 1){
for(int j=0;j<10;j++){
if(grps[j] != NULL){
if(grp_id == grps[j]->group_id){
for(int k=0;k<grps[j]->no_of_clients;k++){
if(grps[j]->members[k] == sd){
bzero(rply,sizeof(rply));
sprintf(rply,"You are already in group %05d\n",grp_id);
printf("%05d %s",sd,rply);
send(sd,rply,sizeof(rply),0);
check = 0;
break;
}
```

```

}
}
}
}
}
if(check == 1){
int temp = 0;
for(int l=0;l<10;l++){
if(grp_reqs[l] != NULL && grp_reqs[l]->g_id == grp_id){
for(int k=0;k<grp_reqs[l]->no_of_clients;k++){
if(grp_reqs[l]->members[k] == sd){
bzero(rply,sizeof(rply));
if(grp_reqs[l]->processed[k] == -8989){
sprintf(rply,"You already accepted to add group..!\n");
send(sd,rply,sizeof(rply),0);
}
else if(grp_reqs[l]->processed[k] == -8888){
sprintf(rply,"already declined..!\n");
send(sd,rply,sizeof(rply),0);
}
else{
grp_reqs[l]->processed[k] = -8888;
grp_reqs[l]->members[k] = -1;
grp_reqs[l]->dec_count++;
sprintf(rply,"You are request to reject being added to group %05d is processed..!\n",grp_id);
send(sd,rply,sizeof(rply),0);
bzero(rply,sizeof(rply));
sprintf(rply,"%05d has declined your request to join the group..!\n",sd);
printf("%s",rply);
send(grp_reqs[l]->admin,rply,sizeof(rply),0);
}
}
check =0;
break;
}
}
if(grp_reqs[l]->no_of_rplys + grp_reqs[l]->dec_count == grp_reqs[l]->no_of_clients && grp_reqs[l]->no_of_rplys != 0){
for(int k=0;k<10;k++){
if(grps[k] == NULL){
struct group *add = (struct group*)malloc(sizeof(struct group));
grps[k] = add;
grps[k]->admin = grp_reqs[l]->admin;
grps[k]->no_of_clients = grp_reqs[l]->no_of_clients;
grps[k]->group_id = grp_reqs[l]->g_id;
for(int m=0;m<grps[k]->no_of_clients;m++){
if(grp_reqs[l]->members[m] != -1){
grps[k]->members[m] = grp_reqs[l]->members[m];
bzero(rply,sizeof(rply));
sprintf(rply,"You were added to the group %05d\n",grps[k]->group_id);
printf("%05d %s",grps[k]->members[m],rply);
send(grps[k]->members[m],rply,sizeof(rply),0);
}
}
//admin is generated of the group
bzero(rply,sizeof(rply));
grps[k]->members[grps[k]->no_of_clients++] = grp_reqs[l]->admin;
sprintf(rply,"Group %05d is created..!\n",grp_reqs[l]->g_id);
printf("%s",rply);
send(grp_reqs[l]->admin,rply,sizeof(rply),0);
break;
}
}
}
else if(grp_reqs[l]->no_of_clients == grp_reqs[l]->dec_count){
bzero(rply,sizeof(rply));
sprintf(rply,"Group is not created Only with you..!\n");
send(grp_reqs[l]->admin,rply,sizeof(rply),0);
}
else{
}
}
}
}
if(check == 1){
bzero(rply,sizeof(rply));
sprintf(rply,"You don't have any request to decline the group %05d\n",grp_id);
send(sd,rply,sizeof(rply),0);
}

```

```

}
}
}
//help prints list of all the commands to run the program.
else if(strcmp(start_token,">HELP") == 0){
printf(">SEND <client-id> : [SENDS MESSAGE TO THE CLIENT ID SPECIFIED]\n");
printf(">SENDGROUP <group-id> : [SENDS MESSAGE TO THE CLIENT ID SPECIFIED]\n");
printf(">ACTIVE : [RETURNS A LIST OF CLIENTS THAT ARE ACTIVE]\n");
printf(">LISTGROUPS : [RETURNS A LIST OF GROUPS THAT ARE ACTIVE]\n");
printf(">MYGROUPS <client-id> : [SHOWS A LIST OF GROUPS THAT THE CLIENT IS A PART OF.]\n");
printf(">BROADCAST : [SENDS THE MESSAGE TO ALL ACTIVE CLIENTS.]\n");
printf(">MAKEGROUP <client-id><client-id>... : [MAKES A GROUP OF ALL THE CLIENT IDS THAT ARE ENTERED.]\n");
printf(">MAKEGROUPREQUEST <client-id><client-id>... : [MAKES A TEMP GROUP OF ALL THE CLIENT IDS THAT ARE ENTERED, SENDS THEM A JO
printf(">JOINGROUP <group-id> : [ACCEPTS THE REQUEST CREATED BY THE CLIENT TO JOIN A GROUP.]\n");
printf(">DECLINEGROUP <group-id> : [DECLINES THE REQUEST CREATED BY THE CLIENT TO JOIN A GROUP.]\n");
}
else{
//checking for non exit condition
printf("Invalid Request...\n");
send(sd,"Invalid Request...\n",strlen("Invalid Request...\n"),0);
//bzero(buf,sizeof(buf));
}
}
}
}
}
}
}

```

CLIENT CODE

```

/*****
Group [11]
Rutvi Bhatt [202112068]
Atharva Vaze [202112015]
Saksham Jain [202112029]
*****/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include <signal.h>
int main(int argc,char *argv[]){
int client_socket,port;
struct sockaddr_in serv_addr;
char buff[1024];
//checking required no of argument provided or not
if(argc < 2){
perror("ERROR: port no is required...\n");
exit(0);
}
//creating socket
client_socket = socket(AF_INET,SOCK_STREAM,0);
if(client_socket < 0){
perror("ERROR: While creating client socket...\n");
exit(0);
}
printf("Client socket created succussfully...\n");
bzero(&serv_addr,sizeof(serv_addr));
port = atoi(argv[1]);
//assigning PORT and IP
serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(port);
serv_addr.sin_addr.s_addr = INADDR_ANY;
//connecting to server
int con_val = connect(client_socket,(struct sockaddr*)&serv_addr,sizeof(serv_addr));
if(con_val != 0){
perror("ERROR: While conneting to socket...\n");
exit(0);
}
}

```

```

bzero(buff, sizeof(buff));
if(recv(client_socket, buff, 1024, 0) < 0){
    printf("Error in receiving data...\n");
    exit(0);
}
else{
    printf("%s", buff);
    if(strcmp(buff, "Connection Limit Exceeded !!\n") == 0){
        exit(0);
    }
}
//infinite loop untill client exit
int pid = fork();
if(pid == 0){
    while(1){
        bzero(buff, sizeof(buff));
        //reading input
        fgets(buff, 1024, stdin);
        send(client_socket, buff, strlen(buff), 0);
        //checking for exit condition
        if(strncmp(buff, "/quit", 5) == 0){
            //closing the socket discriptor
            close(client_socket);
            printf("Disconnected from server...\n");
            break;
        }
    }
}
else{
    while(1){
        char buf[1024];
        bzero(buf, sizeof(buf));
        if(recv(client_socket, buf, 1024, 0) < 0){
            printf("Error in receiving data...\n");
        }
        else{
            printf("%s\n", buf);
        }
    }
}
kill(pid, SIGKILL);
//closing socket discriptor
close(client_socket);
}

```

OUTPUT

```
Activities Terminal Thu 22:39 rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
rutvi@rutvi-VirtualBox:~$ cd Desktop
rutvi@rutvi-VirtualBox:~/Desktop$ cd CN
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ls
client.c server.c
rutvi@rutvi-VirtualBox:~/Desktop/CN$ gcc client.c -o client
rutvi@rutvi-VirtualBox:~/Desktop/CN$ gcc server.c -o server
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ./server 3009
Succussfully Server Socket is created..
Server Socket successfully binded...
Server listening..
Waiting for connection...
socket desrcipoter 4
Connection accepted from 127.0.0.1:52276
online = 1 total = 5
socket desrcipoter 5
Connection accepted from 127.0.0.1:52280
online = 2 total = 5
socket desrcipoter 6
Connection accepted from 127.0.0.1:52284
online = 3 total = 5
From client>MAKEGROUP 00006 00004
00004 added to group 00001
Group 00001 created..!
From client>HELLO GROUP 11
Invalid Request..!
From client>SENDGROUP 00001 HELLO
```

```
Activities Terminal Thu 22:40 rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
grp00001
Sending Message....!
Sending Message....!
Message delivered to all members in group 00001..!
From client>SENDGROUP 00001 GROUP_11
grp00001
Sending Message....!
Sending Message....!
Message delivered to all members in group 00001..!
From client>SENDGROUP 00001 RUTVI - 202112068
grp00001
Sending Message....!
Sending Message....!
Message delivered to all members in group 00001..!
From client>SENDGROUP 00001 ATHARVA - 202112015
grp00001
Sending Message....!
Sending Message....!
Message delivered to all members in group 00001..!
From client>SENDGROUP 00001 SAKSHAM - 202112029
grp00001
Sending Message....!
Sending Message....!
Message delivered to all members in group 00001..!
```

```
Activities Terminal Thu 22:41 rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
rutvi@rutvi-VirtualBox:~$ cd Desktop
rutvi@rutvi-VirtualBox:~/Desktop$ cd CN
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ls
client client.c server server.c
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ./client 3009
Client socket created succussfully...
Succussfully connected to server...
Welcome to our Chat Application...
Your user id is 00004
You were added to group with id 00001
00001:(group) HELLO
00001:(group) GROUP_11
00001:(group) RUTVI - 202112068
00001:(group) ATHARVA - 202112015
00001:(group) SAKSHAM - 202112029
```

```
Activities Terminal Thu 22:41 rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
rutvi@rutvi-VirtualBox:~$ cd Desktop
rutvi@rutvi-VirtualBox:~/Desktop$ cd CN
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ls
client client.c server server.c
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ./client 3009
Client socket created succussfully...
Succussfully connected to server...
Welcome to our Chat Application...
Your user id is 00005
```

```
Activities Terminal Thu 22:43
rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
rutvi@rutvi-VirtualBox:~/Desktop/CN$ ./client 3009
Client socket created succussfully...
Succussfully connected to server...
Welcome to our Chat Application...
Your user id is 00006
>MAKEGROUP 00006 00004
Group 00001 created..!

HELLO GROUP 11
Invalid Request..!
>SENDGROUP 00001 HELLO
00001:(group) HELLO

Message delivered to all members in group 00001..!

>SENDGROUP 00001 GROUP_11
00001:(group) GROUP_11

Message delivered to all members in group 00001..!

>SENDGROUP 00001 RUTVI - 202112068
00001:(group) RUTVI - 202112068

Message delivered to all members in group 00001..!

>SENDGROUP 00001 ATHARVA - 202112015
00001:(group) ATHARVA - 202112015

Message delivered to all members in group 00001..!
```

```
Activities Terminal Thu 22:44
rutvi@rutvi-VirtualBox: ~/Desktop/CN
File Edit View Search Terminal Help
HELLO GROUP 11
Invalid Request..!
>SENDGROUP 00001 HELLO
00001:(group) HELLO

Message delivered to all members in group 00001..!

>SENDGROUP 00001 GROUP_11
00001:(group) GROUP_11

Message delivered to all members in group 00001..!

>SENDGROUP 00001 RUTVI - 202112068
00001:(group) RUTVI - 202112068

Message delivered to all members in group 00001..!

>SENDGROUP 00001 ATHARVA - 202112015
00001:(group) ATHARVA - 202112015

Message delivered to all members in group 00001..!

>SENDGROUP 00001 SAKSHAM - 202112029
00001:(group) SAKSHAM - 202112029

Message delivered to all members in group 00001..!
```

