

EX 3: CHAT USING TCP || SRINITHYEE SK || 185001166

CLIENT:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<string.h>
int main(int argc, char **argv)
{
    int len;
    int sockfd, n = 0;
    struct sockaddr_in servaddr, cliaddr;
    char str[1000];
    char buff[1024];
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if(sockfd < 0)
        perror("Cannot create socket!\n");
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr(argv[1]);
    servaddr.sin_port = htons(7228);
    connect(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    while(1) {
        scanf("%s", buff);
        n = write(sockfd, buff, sizeof(buff));
    }
    close(sockfd);
    return 0;
}
```

SERVER:

```
#include<stdio.h>
#include<fcntl.h>
#include<arpa/inet.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
int main(int argc, char **argv)
{
    int len, clients[5], selsd;
    int sockfd, newfd, n = 0;
    struct sockaddr_in servaddr, cliaddr;
    fd_set readfds;
    char buff[1024];
    char str[1000];
    for(int i = 0; i < 5; i++) clients[i] = 0;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if(sockfd < 0)
        perror("Cannot create socket!\n");
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(7228);
    if(bind(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr)) < 0)
        perror("Bind error!\n");
    listen(sockfd, 5);
    len = sizeof(cliaddr);
    while(1) {
        FD_ZERO(&readfds);
        FD_SET(sockfd, &readfds);
        selsd = sockfd;
        for(int i = 0; i < 5; i++) {
            int sd = clients[i];
            if(sd > 0) FD_SET(sd, &readfds);
            if(sd > selsd) selsd = sd;
        }
        int activity = select(selsd + 1, &readfds, NULL, NULL, NULL);
```

EX 3: CHAT USING TCP || SRINITHYEE SK || 185001166

```
if(FD_ISSET(sockfd, &readfds)) {
    newfd = accept(sockfd, (struct sockaddr*)&cliaddr, &len);
    for(int i = 0; i < 5; i++) {
        if(clients[i] == 0) {
            clients[i] = newfd;
            break;
        }
    }
}
for(int i = 0; i < 5; i++) {
    if(FD_ISSET(clients[i], &readfds))
    {
        read(clients[i], buff, 1024);
        if(buff[0] == '*')
        {
            close(clients[i]);
            clients[i] = 0;
            printf("Client %d closed!\n", i);
        }
        else {
            printf("Client %d: %s \n", i, buff);
            strcpy(buff, "");
        }
    }
}
close(sockfd);
return 0;
}
```