

```

//SERVER

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h> // read,write,close

#include <netdb.h>
#include <netinet/in.h>

#include <sys/socket.h>
#include <sys/types.h>

#define MAX 80
#define PORT 8080
#define sa struct sockaddr

void chat_func(int clfd)
{
    char buff[MAX];
    int n;
    for (;;)
    {
        bzero(buff, MAX);
        read(clfd, buff, sizeof(buff));
        printf("Client texted : %s \n Your message : ", buff);
        n = 0;
        while ((buff[n++] = getchar()) != '\n')
            ;
        write(clfd, buff, sizeof(buff));
        if (strncmp("exit", buff, 4) == 0)
        {
            printf("Server Exiting...\n");
            break;
        }
    }
    return;
}

int main()
{
    printf("Starting Server!");
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in server_addr, client_addr;

```

```

int option_value_sockopt = 1;
if (sockfd == -1)
{
    perror("Socket creation failed");
    exit(EXIT_FAILURE);
}
else
    printf("Socket successfully created for Server!\n");
int status = setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR,
&option_value_sockopt, sizeof(option_value_sockopt));

if (status < 0)
{
    perror("Socket option settings failed");
    exit(EXIT_FAILURE);
}
else
    printf("Server socket options set!\n");

// setting server address struct stuff
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(PORT);
server_addr.sin_addr.s_addr = htonl(INADDR_ANY);

if (bind(sockfd, (sa *)&server_addr, sizeof(server_addr)) != 0)
{
    printf("Socket unable to bind on port: %d!", PORT);
    exit(0);
}
else
    printf("Socket successfully binded on %d\n", PORT);

// backlog of 5, ie 5 requests can be entertained.
if (listen(sockfd, 5) != 0)
{
    perror("Socket listen failed");
    exit(EXIT_FAILURE);
}
else
    printf("Server is listening on Port : %d\n", PORT);
int cli_len = sizeof(client_addr);
int clifd = accept(sockfd, (sa *)&client_addr, &cli_len);
if (clifd < 0)
{

```

```

        perror("Socket unable to accept connections!");
        exit(EXIT_FAILURE);
    }
    chat_func(clifd);
    close(sockfd);
    return 0;
}

```

// CLIENT

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>

#include <netdb.h>
#include <arpa/inet.h>

#include <sys/socket.h>
#include <sys/types.h>

#define MAX 80
#define PORT 8080
#define sa struct sockaddr

void chat_func(int sockfd)
{
    char buff[MAX];
    int n;
    for (;;)
    {
        bzero(buff, sizeof(buff));
        printf("Type your message : ");
        n = 0;
        while ((buff[n++] = getchar()) != '\n')
            ;
        write(sockfd, buff, sizeof(buff));
        if (strncmp("exit", buff, 4) == 0)
        {
            printf("Client Exit...\n");
            break;
        }
        bzero(buff, sizeof(buff));
        read(sockfd, buff, sizeof(buff));
    }
}

```

```

        printf("From Server : %s", buff);
    }
    return;
}

int main()
{
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in server_addr, client_addr;

    if (sockfd == -1)
    {
        perror("Socket creation failed");
        exit(EXIT_FAILURE);
    }
    else
        printf("Socket successfully created for Client!\n");
    bzero(&server_addr, sizeof(server_addr));

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);
    server_addr.sin_addr.s_addr = htonl(INADDR_LOOPBACK);

    if (connect(sockfd, (sa *)&server_addr, sizeof(server_addr)) != 0)
    {
        perror("Couldn't connect to server!");
        exit(EXIT_FAILURE);
    }
    else
        printf("Connected to server! Start Messaging!\n");

    chat_func(sockfd);
    close(sockfd);
    return 0;
}

```