

BUDGE BUDGE INSTITUTE OF TECHNOLOGY



Name: Sudip Basu

Subject: Computer Network

Department: Computer Science and Technology

Roll:16

Semester:4th

Socket Programming in C

TCP Server-Client implementation in C

TCP Server -

- 1.using create(), Create TCP socket.
- 2.using bind(), Bind the socket to server address.
- 3.using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection
- 4.using accept(), At this point, connection is established between client and server, and they are ready to transfer data.
- 5.Go back to Step 3.

TCP Client -

- 1.Create TCP socket.
- 2.connect newly created client socket to server.

Source Code Of TCP Server (C Programming)-

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr

// Function designed for chat between client and server.
void func(int sockfd)
{
    char buff[MAX];
    int n;
    // infinite loop for chat
    for (;;) {
        bzero(buff, MAX);

        // read the message from client and copy it in buffer
```

```

        read(sockfd, buff, sizeof(buff));
        // print buffer which contains the client contents
        printf("From client: %s\t To client : ", buff);
        bzero(buff, MAX);
        n = 0;
        // copy server message in the buffer
        while ((buff[n++] = getchar()) != '\n')
            ;

        // and send that buffer to client
        write(sockfd, buff, sizeof(buff));

        // if msg contains "Exit" then server exit and chat ended.
        if (strncmp("exit", buff, 4) == 0) {
            printf("Server Exit...\n");
            break;
        }
    }
}

```

// Driver function

```

int main()
{
    int sockfd, connfd, len;
    struct sockaddr_in servaddr, cli;

    // socket create and verification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));

    // assign IP, PORT
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);
}

```

```

// Binding newly created socket to given IP and verification
if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
    printf("socket bind failed...\n");
    exit(0);
}
else
    printf("Socket successfully binded..\n");

// Now server is ready to listen and verification
if ((listen(sockfd, 5)) != 0) {
    printf("Listen failed...\n");
    exit(0);
}
else
    printf("Server listening..\n");
len = sizeof(cli);

// Accept the data packet from client and verification
connfd = accept(sockfd, (SA*)&cli, &len);
if (connfd < 0) {
    printf("server accept failed...\n");
    exit(0);
}
else
    printf("server accept the client...\n");

// Function for chatting between client and server
func(connfd);

// After chatting close the socket
close(sockfd);
return 0;
}

```

Source Code Of TCP Client (C Programming)-

```

#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>

```

```

#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void func(int sockfd)
{
    char buff[MAX];
    int n;
    for (;;) {
        bzero(buff, sizeof(buff));
        printf("Enter the string : ");
        n = 0;
        while ((buff[n++] = getchar()) != '\n')
            ;
        write(sockfd, buff, sizeof(buff));
        bzero(buff, sizeof(buff));
        read(sockfd, buff, sizeof(buff));
        printf("From Server : %s", buff);
        if ((strcmp(buff, "exit", 4)) == 0) {
            printf("Client Exit...\n");
            break;
        }
    }
}

int main()
{
    int sockfd, connfd;
    struct sockaddr_in servaddr, cli;

    // socket create and varification
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        printf("socket creation failed...\n");
        exit(0);
    }
    else
        printf("Socket successfully created..\n");
    bzero(&servaddr, sizeof(servaddr));

    // assign IP, PORT

```

```

servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
servaddr.sin_port = htons(PORT);

// connect the client socket to server socket
if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
    printf("connection with the server failed...\n");
    exit(0);
}
else
    printf("connected to the server..\n");

// function for chat
func(sockfd);

// close the socket
close(sockfd);
}

```

Compilation -

Server side:

```
gcc -o tcp_server tcp_server.c
./tcp_server
```

Client side:

```
gcc -o tcp_client tcp_client.c
./tcp_client
```

Output (Communication Between Server Side and Client Side):-

ActivitiesTerminalJun 28 14:22

Terminal Window 1: sudip@sudip-Vostro-3578: ~/Desktop/Socket Programming

Terminal Window 2: sudip@sudip-Vostro-3578: ~/Desktop/Socket Programming

```
sudip@sudip-Vostro-3578:~/Desktop/Socket Programming$ ./tcp_server
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hi
    To client : hello
From client: What is your name?
    To client : Sudip
From client: In which college you are studying?
    To client : I am studying in BBIT College in CST Dept.
From client: oh,great!!!!
    To client : Ok
From client: So Bye
    To client : so,after diploma , which carrier you will be chossen?
From client: Btech
    To client : 
```

```
sudip@sudip-Vostro-3578:~/Desktop/Socket Programming$ ./tcp_client
Socket successfully created..
connected to the server..
Enter the string : hi
From Server : hello
Enter the string : What is your name?
From Server : Sudip
Enter the string : In which college you are studying?
From Server : I am studying in BBIT College in CST Dept.
Enter the string : oh,great!!!!
From Server : Ok
Enter the string : So Bye
From Server : so,after diploma , which carrier you will be chossen?
Enter the string : Btech

```