

# EX 6: HALF DUPLEX CHAT USING TCP/IP

**Name: Rahul Goel**

**Reg.no: RA1911030010094**

**GIVEN REQUIREMENTS:** There are two hosts, Client and Server. Both the Client and the Server exchange message i.e. they send messages or receive message from the other. There is only a single way communication between them.

**TECHNICAL OBJECTIVE:** To implement a half duplex application, where the Client establishes a connection with the Server. The Client can send and the server will receive messages at the same time.

**METHODOLOGY: Server:**

- Include the necessary header files.
- Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin\_family to AF\_INET, sin\_addr to INADDR\_ANY, sin\_port to dynamically assigned port number.
- Bind the local host address to socket using the bind function.
- Listen on the socket for connection request from the client.
- Accept connection request from the Client using accept function.
- Fork the process to receive message from the client and print it on the console.
- Read message from the console and send it to the client.

**Client:**

- Include the necessary header files.
- Create a socket using socket function with family AF\_INET, type as SOCK\_STREAM.
- Initialize server address to 0 using the bzero function.
- Assign the sin\_family to AF\_INET.
- Get the server IP address and the Port number from the console.
- Using gethostbyname function assign it to a hostent structure, and assign it to sin\_addr of the server address structure.

- Request a connection from the server using the connect function.
- Fork the process to receive message from the server and print it on the console.

➤ Read message from the console and send it to the server.

➤

## ➤ SERVER CODE:

```
#include<stdio.h>
#include<netdb.h>
int main(int argc, char *argv[])
{
    int n, sd, cd;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t servlen, clien;
    char buff[1000], buff1[1000];
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = inet_addr(argv[1]);
    servaddr.sin_port = htons(7510);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    cd = connect(sd, (struct sockaddr*)&servaddr, sizeof(servaddr)); while(1)
    {
        bzero(&buff, sizeof(buff));
        printf("%s\n", "Enter the input data:");
        fgets(buff, 1000, stdin);
        send(sd, buff, strlen(buff)+1, 0);
        printf("%s\n", "Data sent");
        n=1;
        while(n==1)
        {
            bzero(&buff1, sizeof(buff1));
            recv(sd, buff1, sizeof(buff1), 0);
            printf("Received from the server:%s\n", buff1);
            n=n+1;
        }
    }
    return 0;
}
```

bash - "ip-172-31-13-91" x Immediate x Bot

RA1911030010094:~/environment \$

The screenshot shows a web-based IDE interface for a C program. The editor has a dark theme and a sidebar on the left showing a file explorer. The main editor area contains a C program for a simple TCP server. The code includes standard headers, defines a buffer, and implements a main function that listens for connections and reads data from the client. The terminal at the bottom shows the command to run the program and its output, which includes a warning about an implicit declaration of the 'bzero' function and the message 'server is running.'.

```
#include <sys/types.h>
#include <stdio.h>
#include <netdb.h>
#include <csys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <netinet/in.h>
int main(int argc, char *argv[])
{
    int n, sd, ad;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t clien, servlen;
    char buff[1000], buff1[1000];
    bzero(servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(7514);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    bind(sd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    listen(sd, 5);
    printf("%s\n", "server is running.");
    ad = accept(sd, (struct sockaddr*)&cliaddr, &clien);
    while(1)
    {
        bzero(buff, sizeof(buff));
        recv(ad, buff, sizeof(buff), 0);
        printf("Receive from the client: %s\n", buff);
        n = 1;
        while(n == 1)
        {
            bzero(buff1, sizeof(buff1));
            printf("%s\n", "Enter the input data:");
        }
    }
}
```

Terminal output:

```
bash - "ip-172-31-13-91" x Immediate
RA1911030010094:~/environment $
```

The screenshot shows a web-based IDE interface for a C program, similar to the one above. The editor has a dark theme and a sidebar on the left showing a file explorer. The main editor area contains a C program for a simple TCP client. The code includes standard headers, defines a buffer, and implements a main function that connects to the server and sends data. The terminal at the bottom shows the command to run the program and its output, which includes a warning about an implicit declaration of the 'bzero' function and the message 'server is running.'.

```
#include <sys/types.h>
#include <stdio.h>
#include <netdb.h>
#include <csys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    int n, sd, ad;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t clien, servlen;
    char buff[1000], buff1[1000];
    bzero(servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(7514);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    bind(sd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    listen(sd, 5);
    printf("%s\n", "server is running.");
    ad = accept(sd, (struct sockaddr*)&cliaddr, &clien);
    while(1)
    {
        bzero(buff, sizeof(buff));
        recv(ad, buff, sizeof(buff), 0);
        printf("Receive from the client: %s\n", buff);
        n = 1;
        while(n == 1)
        {
            bzero(buff1, sizeof(buff1));
            printf("%s\n", "Enter the input data:");
        }
    }
}
```

Terminal output:

```
bash - "ip-172-31-12-196" x Immediate (JavaScript (bro x))
Cnsvinothkumar:~/environment/RA1911030010099 $ vi e6server1.c
Cnsvinothkumar:~/environment/RA1911030010099 $ gcc e6server1.c
e6server1.c: In function 'main':
e6server1.c:14:9: warning: implicit declaration of function 'bzero' [-Wimplicit-function-declaration]
    bzero(servaddr, sizeof(servaddr));
    ^~~~~
e6server1.c:14:9: warning: incompatible implicit declaration of built-in function 'bzero'
Cnsvinothkumar:~/environment/RA1911030010099 $ ./a.out
server is running.
Receive from the client: hi reshvanth

Enter the input data:
hello
Data sent
```