

18CSC302J (Computer Networks Lab)

Lab session 6 - Half Duplex Chat Using TCP/IPSERVER

Name :- Puneet Sharma

Reg. No. :- RA1911003010331

Class :-CSE F1

SERVER CODE:-

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "netdb.h"
#include "arpa/inet.h"
#define MAX 1000
#define BACKLOG 5 // how many pending connections queue will hold
int main()
{
    char serverMessage[MAX];
    char clientMessage[MAX];
    //create the server socket
    int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_port = htons(8087);
    serverAddress.sin_addr.s_addr = INADDR_ANY;

    //calling bind function to oir specified IP and port
    bind(socketDescriptor, (struct sockaddr*)&serverAddress,
    sizeof(serverAddress));

    listen(socketDescriptor, BACKLOG);
    //starting the accepting
    int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);

    while (1)
    {
```

```

printf("\nEnter message:");
scanf("%s", serverMessage);
send(clientSocketDescriptor, serverMessage, sizeof(serverMessage) , 0);
//recieve the data from the server
recv(clientSocketDescriptor, &clientMessage, sizeof(clientMessage), 0) ;
//recieved data from the server successfully then printing the data obtained
from the server
printf("\nClient: %s", clientMessage);

}

//close the socket
close(socketDescriptor);
return 0;}

```

```

half_duplex_server.c
1  #include "stdio.h"
2  #include "stdlib.h"
3  #include "string.h"
4  #include <sys/types.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <unistd.h>
8  #include "netdb.h"
9  #include "arpa/inet.h"
10 #define MAX 1000
11 #define BACKLOG 5 // how many pending connections queue will hold
12 int main()
13 {
14     char serverMessage[MAX];
15     char clientMessage[MAX];
16     //create the server socket
17     int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
18
19
20     struct sockaddr_in serverAddress;
21     serverAddress.sin_family = AF_INET;
22     serverAddress.sin_port = htons(8087);
23     serverAddress.sin_addr.s_addr = INADDR_ANY;
24
25     //calling bind function to oir specified IP and port
26     bind(socketDescriptor, (struct sockaddr*)&serverAddress, sizeof(serverAddress));
27
28     listen(socketDescriptor, BACKLOG);
29
30     //starting the accepting
31     int clientSocketDescriptor = accept(socketDescriptor, NULL, NULL);
32
33     while (1)
34     {
35         printf("\nEnter message:");

```

331/half_duplex_ser

Stop Run Config Com 331/half_duplex_server.c Runner: C CWD ENV

```

Client: 55
Enter message:22

Client: 55
Enter message:55

```

CLINT CODE:-

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "netdb.h"
#include "arpa/inet.h"
#define h_addr h_addr_list[0]

#define PORT 8087 // port number
#define MAX 1000 //maximum buffer size

//main function
int main(){
    char serverResponse[MAX];
    char clientResponse[MAX];

    //creating a socket
    int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);

    //placeholder for the hostname and my ip address
    char hostname[MAX], ipaddress[MAX];
    struct hostent *hostIP; //placeholder for the ip address
    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_port = htons(PORT);
    serverAddress.sin_addr.s_addr = INADDR_ANY;

    connect(socketDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));

    while (1)
    { //recieve the data from the server
        recv(socketDescriptor, serverResponse, sizeof(serverResponse), 0);
        //recieved data from the server successfully then printing the data obtained from the server
        printf("\nServer : %s", serverResponse);

        printf("\nEnter message:");
        scanf("%s", clientResponse);
        send(socketDescriptor, clientResponse, sizeof(clientResponse), 0);
    }

    //closing the socket
    close(socketDescriptor);
    return 0;
}
```

```
half_duplex_clint.c
1  #include "stdio.h"
2  #include "stdlib.h"
3  #include "string.h"
4  #include <sys/types.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <unistd.h>
8  #include "netdb.h"
9  #include "arpa/inet.h"
10 #define h_addr h_addr_list[0]
11
12 #define PORT 8087 // port number
13 #define MAX 1000 //maximum buffer size
14
15 //main function
16 int main(){
17     char serverResponse[MAX];
18     char clientResponse[MAX];
19
20     //creating a socket
21     int socketDescriptor = socket(AF_INET, SOCK_STREAM, 0);
22
23     //placeholder for the hostname and my ip address
24     char hostname[MAX], ipaddress[MAX];
25     struct hostent *hostIP; //placeholder for the ip address
26     struct sockaddr_in serverAddress;
27     serverAddress.sin_family = AF_INET;
28     serverAddress.sin_port = htons(PORT);
29     serverAddress.sin_addr.s_addr = INADDR_ANY;
30
31     connect(socketDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
32
33     while (1)
34     {
35         //receive the data from the server
36     }
37 }
```

48:2 C and C++ Spaces: 4

331/half_duplex_clin +

Stop Run Config Cor 331/half_duplex_clint.c Runner: C CWD EN

Server : 522
Enter message:55

Server : 55
Enter message: