Output

Server

```
Socket Creation Successful.
Socket Bind Successful.
Server Listening...
Server Accepted the Client.
Message from Client: hi
Message to Client: exit
Server Exiting...
```

Client

```
Socket Creation Successful
Connected to the Server.
Message to Server: hi
Message from Server: exit
Client Exiting...
```

O/p Verified

Lab: 5

20/2/2020

CLIENT - SERVER USING TCP

Aim

To implement client-server communication using TCP/IP.

Program

Server

```
#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
void func(int sockfd)
{
    char buff[MAX];
    int n;
    for(;;) {
        bzero(buff, MAX);
        read(sockfd, buff, sizeof(buff));
        printf("Message from Client:%s Message to Client:", buff);
        bzero(buff, MAX);
        n=0;
        while((buff[n++] = getchar()) != '\n');
        write(sockfd, buff, sizeof(buff));
        if(strcmp("exit", buff, 4)==0) {
            printf("Server Exiting...\n");
            break;
        }
        }
}
void main()
{
    int sockfd, connfd, len;
    struct sockaddr_in servaddr, cli;
    sockfd=socket(AF_INET, SOCK_STREAM, 0);
    if(sockfd==-1){
        printf("Socket Creation Failed!!! \n");
        exit(0);
    }
    else
        printf("Socket Creation Successful.\n");
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);
    if((bind(sockfd, (SA*)&servaddr, sizeof(servaddr)))!=0){
        printf("Socket Bind Failed!!!\n");
        exit(0);
    }
    else
        printf("Socket Bind Successful.\n");
    if((listen(sockfd,5))!=0) {
        printf("Listen Failed!!!\n");
        exit(0);
    }
    else
        printf("Server Listening...\n");
    len=strlen(cli);
    connfd=accept(sockfd, (SA*)&cli, &len);
    if(connfd<0){
        printf("Server Accept Failed!!! \n");
        exit(0);
    }
    else
        printf("Server Accepted the Client.\n");
    func(connfd);
```

```
        close (sockfd);
}
```

Client

```
#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("Message to Server:");
        n=0;
        while ((buff[n++] = getchar()) != '\n');
        write (sockfd, buff, sizeof (buff));
        bzero (buff, sizeof(buff));
        read (sockfd, buff, sizeof (buff));
        printf ("Message from Server: %s", buff);
        if ((strncmp (buff, "exit", 4)) == 0) {
            printf("Client Exiting..\n");
            break;
        }
    }
}

void main()
{
    int sockfd, connfd;
    struct sockaddr_in servaddr, cli;
    sockfd = socket (AF_INET, SOCK_STREAM, 0);
if(sockfd==-1) {
    printf("Socket Creation Failed!!!\n");
    exit(0);
}
else
    printf("Socket Creation Successful.\n");
bzero(&servaddr, sizeof(servaddr));

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

if(connect(sockfd, (SA*)&servaddr, sizeof(servaddr))!=0)
    printf("Connection with the Server Failed!!!\n");
    exit(0);
}
else
    printf("Connected to the Server.\n");
func(sockfd);
close(sockfd);
}
```

Result

Implemented client-server communication using TCP/IP

8
20/2/20