NAME- ASHUTOSH ARDU
REG NO.- 20BRS1262
DATE- 19-04-2021

CSE1004 LAB-5

ALGORITHM

- First establish a connection between a Client and a Server using socket programming.
- Then send a message from server to client and then receive that particular message at the client end.
- Then resend that previous message this time from client to server.
- Hence, echoing the message that was first sent from server to client back to the server from the client.

CODE

SERVER SIDE

```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<string.h>
// Server Side
int main()
{
    int s,b,sport,key;
    char mess[100];
    printf("Enter the message\n");
    scanf("%s",mess);
    struct sockaddr_in saddr,caddr;
    printf("Enter any Desired port number\n");
    scanf("%d",&sport);
    int clen=sizeof(caddr);
    s=socket(AF_INET,SOCK_STREAM,0);
    if(s<0)
    {
        printf("Error while socket creation\n");
    }
    else
        printf("Socket created successfully\n");
    saddr.sin_family=AF_INET;
    saddr.sin_addr.s_addr=htonl(INADDR_ANY);
    saddr.sin_port=htons(sport);
    b=bind(s,(struct sockaddr*)&saddr,sizeof(saddr));
    if(b==0)
        printf("Interface binded to the socket\n");
    else
        printf("Interface not binded to the socket\n");
    listen(s,5);
    key=accept(s,(struct sockaddr*)&caddr,&clen);
    if(key<0)
        printf("Error\n");
    else
        printf("Well Connected\n");
    send(key,mess,sizeof(mess),0);
    char buffer[100];
    recv(key,buffer,sizeof(buffer),0);
    printf("The echoed message is %s",buffer);

}
```

CLIENT SIDE

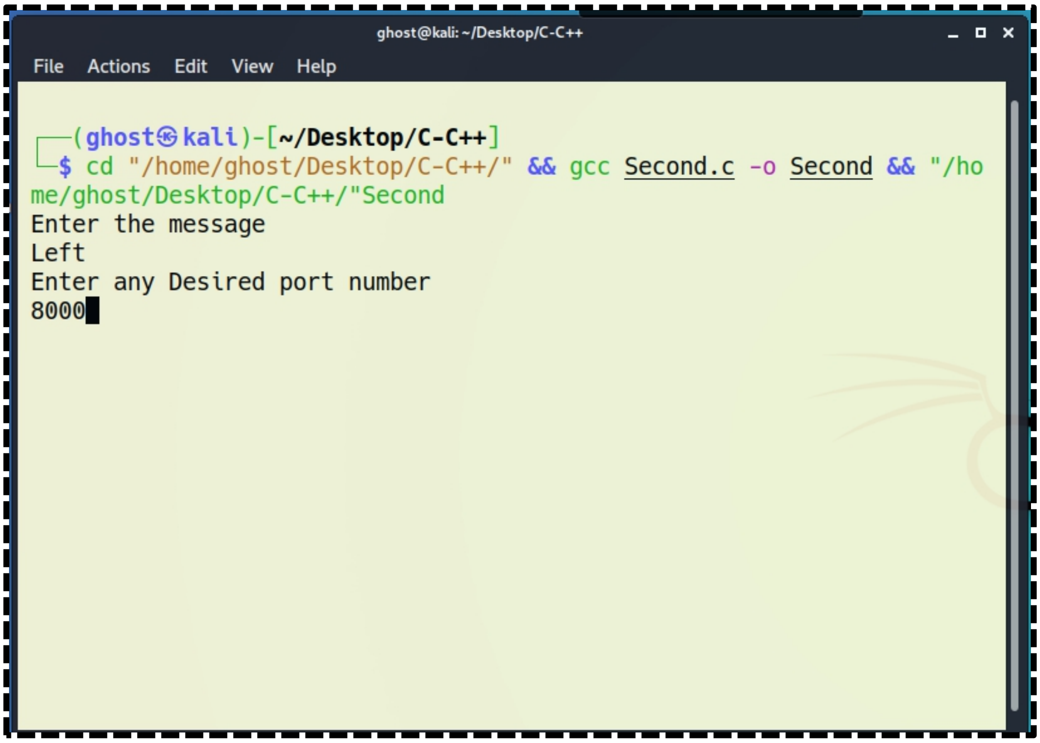
```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
// Client Side
int main()
{
    int soc,port;
    char mess[100];
    struct sockaddr_in server,client;
    printf("Enter the port no.\n");
    scanf("%d",&port);
    soc=socket(AF_INET,SOCK_STREAM,0);
    if(soc<0)
        printf("Socket not created\n");
    else
        printf("Socket Created\n");
    server.sin_family=AF_INET;
    server.sin_addr.s_addr=htonl(INADDR_ANY);
```

```
        server.sin_port=htons(port);


        if(connect(soc,(struct sockaddr*)&server,sizeof(server))<0)
            printf("Can't connect\n");
        else
            printf("Connected\n");
        recv(soc,mess,sizeof(mess),0);
        printf("the message is \n%s\n",mess);
        char buffer[100];
        strcpy(buffer,mess);
        send(soc,buffer,sizeof(buffer),0);
        return 0;
    }
```

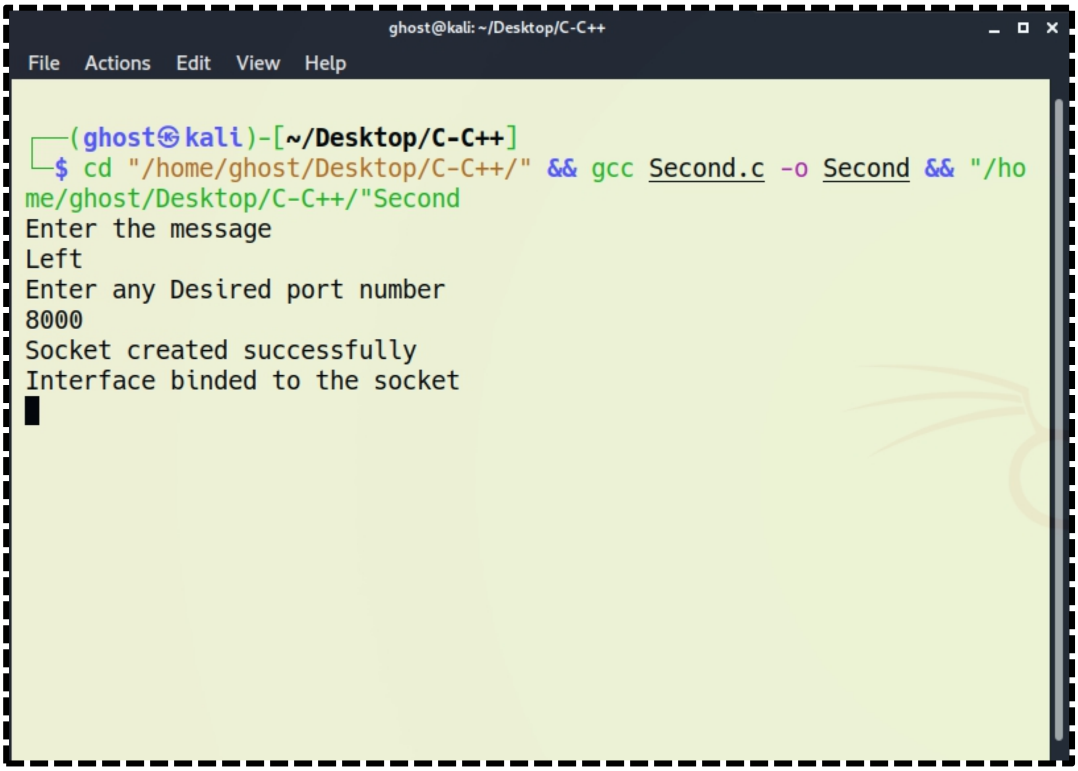
OUTPUT

ENTER THE MESSAGE AND SELECT A PORT NUMBER
IN WHICH THE MESSAGE WILL GET TRANSFERRED.



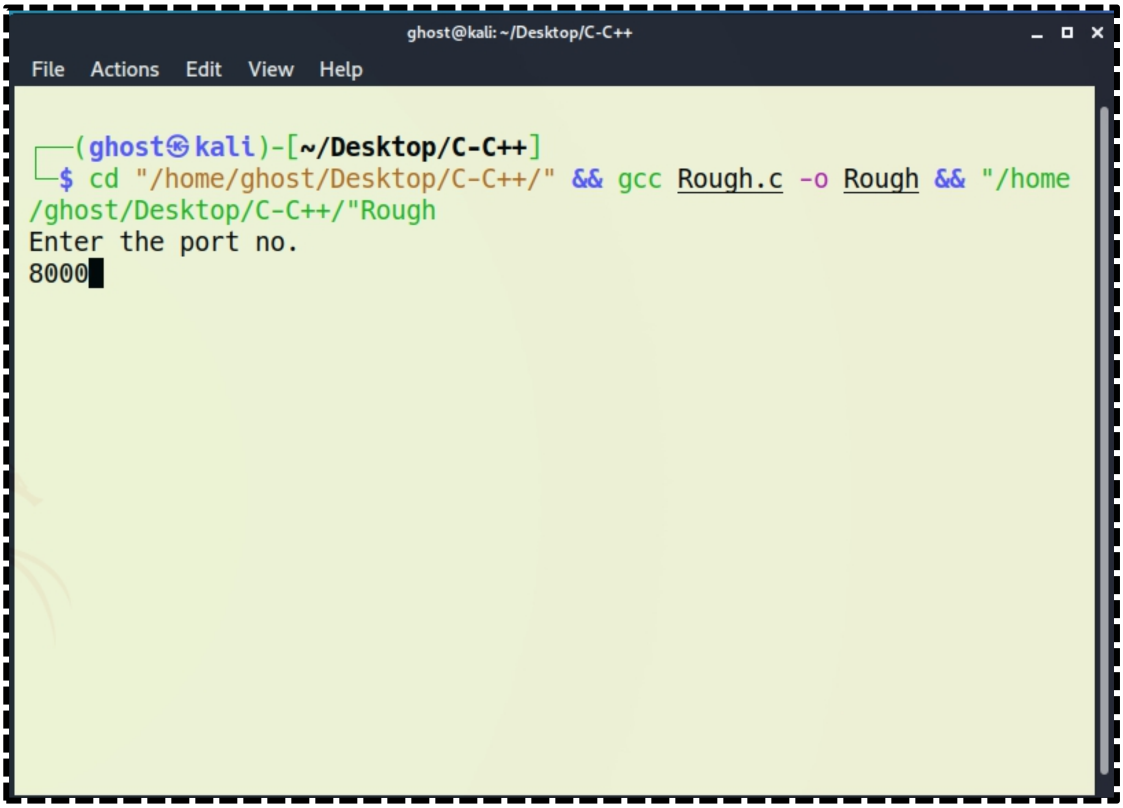
```
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/" && gcc Second.c -o Second && "/home/ghost/Desktop/C-C++/"Second
Enter the message
Left
Enter any Desired port number
8000
```

AFTER ENTERING THE PORT NUMBER, WE CREATE A SOCKET
AND BIND OUR SERVER TO IT.



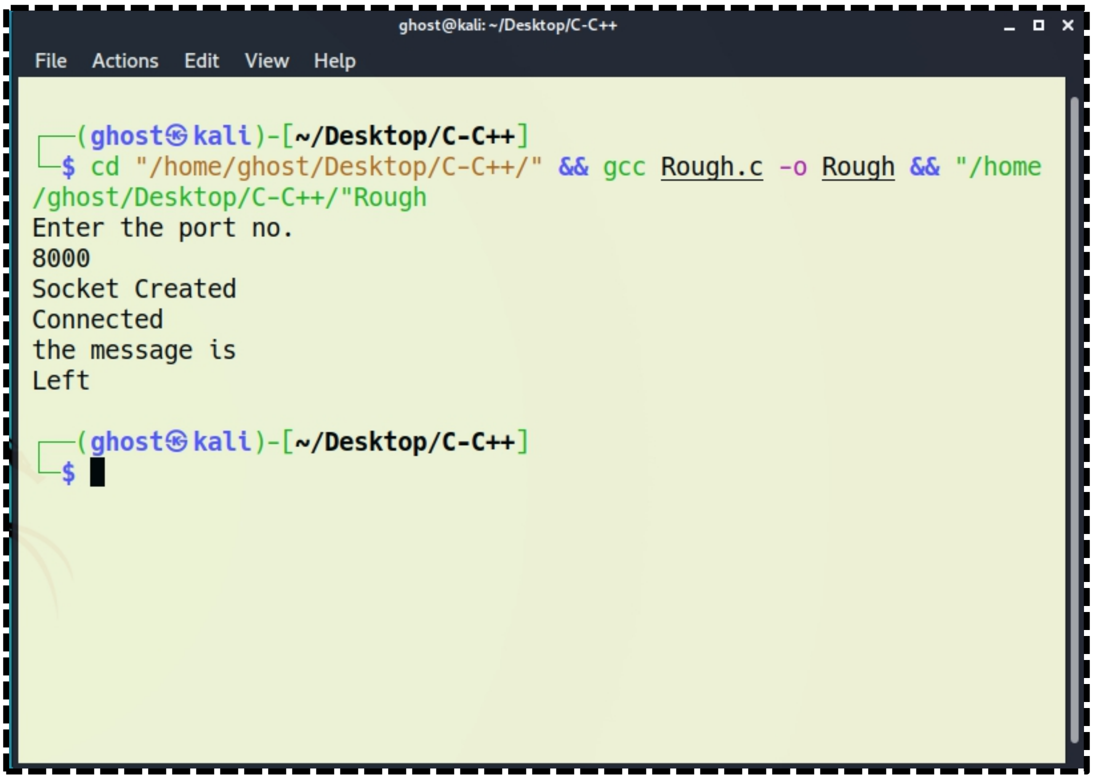
```
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/" && gcc Second.c -o Second && "/home/ghost/Desktop/C-C++/"Second
Enter the message
Left
Enter any Desired port number
8000
Socket created successfully
Interface binded to the socket
```

ON THE CLIENT SIDE WE CONNECT OUR CLIENT
TO THE PORT TO WHICH OUR SERVER IS BINDED TO.



```
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/" && gcc Rough.c -o Rough && "/home/ghost/Desktop/C-C++/"Rough
Enter the port no.
8000
```

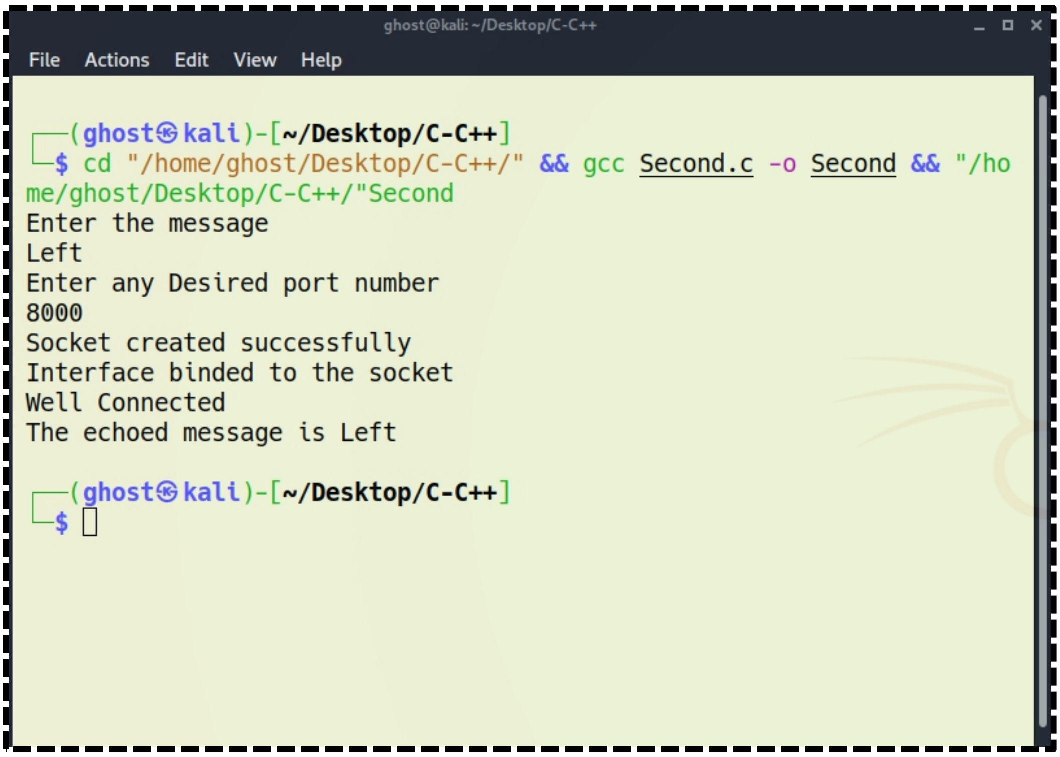
AFTER THE CLIENT GETS CONNECTED TO THE SOCKET
IT SUDDENLY RECEIVES THE MESSAGE SENT BY THE SERVER
AFTER THAT CLIENT RESEND THE MESSAGE BACK TO THE
SERVER.



```
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/" && gcc Rough.c -o Rough && "/home
/ghost/Desktop/C-C++/"Rough
Enter the port no.
8000
Socket Created
Connected
the message is
Left

┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ 
```

ON THE SERVER SIDE
IT RECEIVES THE ECHOED MESSAGE FROM THE CLIENT



```
┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ cd "/home/ghost/Desktop/C-C++/" && gcc Second.c -o Second && "/ho
me/ghost/Desktop/C-C++/"Second
Enter the message
Left
Enter any Desired port number
8000
Socket created successfully
Interface binded to the socket
Well Connected
The echoed message is Left

┌──(ghost㉿kali)-[~/Desktop/C-C++]
└─$ 
```