

## ***TCP Server-Client***

### ***Explanation:***

In the TCP Server Client program , the following are the main steps that take place in :

**In the TCP Server the following steps are performed:**

1. A TCP socket is created using create().
2. Then this socket is bind to the server address using bind(). Bind() operation is usually assigning a name to a socket
3. Then we use a listen() call , which put the socket in a passive mode. In this mode, server sockets wait for the client to approach the server to make a connection.
4. Then we use accept(), that accepts the connection , connection between the client and server is made and they are ready to transfer data
5. Then it goes back to the third step to see listen if any other connection call is made

**In the TCP Client the following steps are performed:**

1. Here also , a TCP socket is created using create()
2. Then this newly created client socket is connected to the server socket

### **TCP -Client Server Code with Documentation**

#### **server.c**

```
/*
```

```
    TCP-Server
```

```
*/
```

```
#include<sys/types.h>
```

```
#include<sys/socket.h>
```

```
#include<stdio.h>
```

```

#include<netinet/in.h>
#include <unistd.h>
#include<string.h>
#include <arpa/inet.h>

void main()
{
int sockfd,connfd,sin_size,l,n,len;
char operator;
int op1,op2,result;
//socket creation
if((sockfd=socket(AF_INET,SOCK_STREAM,0))>0)
printf("socket created sucessfully\n");
struct sockaddr_in servaddr;
struct sockaddr_in clientaddr;
//assigning IP,Port
servaddr.sin_family=AF_INET;//Server Domain
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");//Server IP Address
servaddr.sin_port=6006;//Port on which the data to be sent
//Binding newly created socket to the given IP
if((bind(sockfd, (struct sockaddr *)&servaddr,sizeof(servaddr)))==0)
printf("bind sucessful\n");
//listen for connections on a socket
if((listen(sockfd,5))==0)
printf("listen sucessful\n");

```

```

sin_size = sizeof(struct sockaddr_in);

//Server Accepts the data packet from the client
if((connfd=accept(sockfd,(struct sockaddr *)&clientaddr,&sin_size))>0);
printf("accept sucessful\n");

//Server recives the data from the Client
read(connfd, &operator,sizeof(operator));
read(connfd,&op1,sizeof(op1));
read(connfd,&op2,sizeof(op2));

//code for calculator functioning
switch(operator)
{
case '+':
    result=op1 + op2;
    printf("Result is: %d + %d = %d\n",op1, op2, result);
    break;
case '-':
    result=op1 - op2;
    printf("Result is: %d - %d = %d\n",op1, op2, result);
    break;
case '*':
    result=op1 * op2;
    printf("Result is: %d * %d = %d\n",op1, op2, result);
    break;
case '/':

```

```

    result=op1 / op2;
    printf("Result is: %d / %d = %d\n",op1, op2, result);
    break;
default:
    printf("ERROR: Unsupported Operation");
}
//Server is sending the result to the Client
write(connfd,&result,sizeof(result));
//Close Socket
close(sockfd);

}

```

//Code Ends Below is the Client Code

client.c

```

/*
    TCP-Client
*/
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<netinet/in.h>
#include <unistd.h>

```

```

#include<string.h>
#include<strings.h>
#include <arpa/inet.h>
void main()
{
    int sockfd,sin_size,con,n,len;
    char operator;
    int op1,op2,result;
    if((sockfd=socket(AF_INET,SOCK_STREAM,0))>0)//Here TCP Socket is
Created
    printf("socket created sucessfully\n");
    struct sockaddr_in servaddr;
    //assigning IP,Port
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
    servaddr.sin_port=6006;
    sin_size = sizeof(struct sockaddr_in);
    //Connecting client socket to server socket
    if((con=connect(sockfd,(struct sockaddr *) &servaddr, sin_size))==0); //initiate a
connection on a socket
    printf("connect sucessful\n");

    printf("Enter any one operation:\n + \n -\n / \n* \n");

    scanf("%c", &operator);
    printf("\n");

```

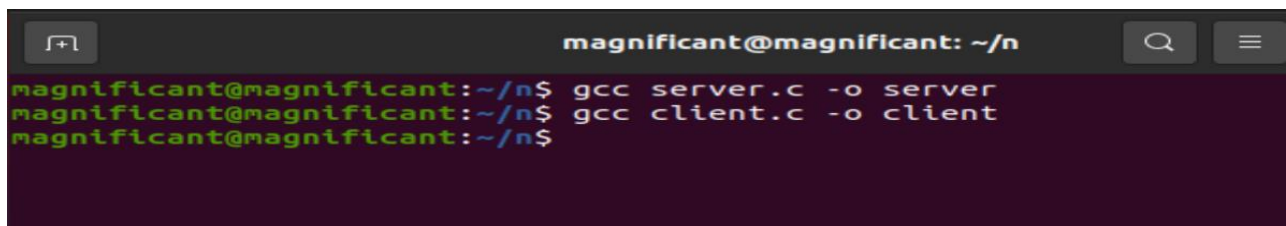
```

printf("Enter operands:\n");
scanf("%d %d",&op1, &op2);
//sending the input values to the server
write(sockfd,&operator,sizeof(operator));
write(sockfd,&op1,sizeof(op1));
write(sockfd,&op2,sizeof(op2));
//Receiving and displaying the message received from the Server
read(sockfd,&result,sizeof(result));
printf("Operation result from server=%d\n",result);
//Close the socket
close(sockfd);
}

```

## OUTPUT (Left Screen is Server Side, Right Screen is Client Side)

Firstly, we compile the c program using the following :

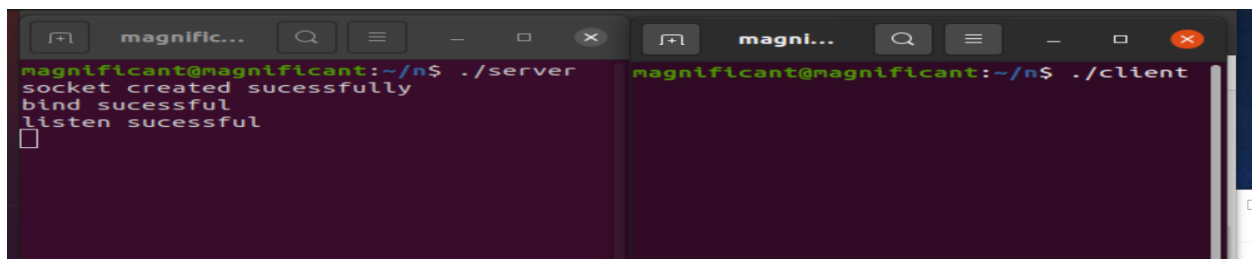


```

magnificent@magnificent: ~/n
magnificent@magnificent:~/n$ gcc server.c -o server
magnificent@magnificent:~/n$ gcc client.c -o client
magnificent@magnificent:~/n$

```

Then we run the server by using ./server command



```

magnificent@magnificent:~/n$ ./server
socket created successfully
bind successful
listen successful
□

magnificent@magnificent:~/n$ ./client

```

Then we run the client by using ./client command

```
magnificent@magnificent:~/n$ ./server
socket created successfully
bind successful
listen successful
accept successful
□

magnificent@magnificent:~/n$ ./client
socket created successfully
connect successful
Enter any one operation:
+
-
/
*
□
```

Here the server sends the output according to the user information entered :

```
magnificent@magnificent:~/n$ ./server
socket created successfully
bind successful
listen successful
accept successful
Result is: 4 + 5 = 9
magnificent@magnificent:~/n$ □

magnificent@magnificent:~/n$ ./client
socket created successfully
connect successful
Enter any one operation:
+
-
/
*
+
Enter operands:
4
5
Operation result from server=9
magnificent@magnificent:~/n$
```

```
magnificent@magnificent:~/n$ ./server
socket created successfully
bind successful
listen successful
accept successful
Result is: 4 + 5 = 9
magnificent@magnificent:~/n$ ./server
socket created successfully
bind successful
listen successful
accept successful
Result is: 10 - 6 = 4
magnificent@magnificent:~/n$ □

magnificent@magnificent:~/n$ ./client
Enter any one operation:
+
-
/
*
-
Enter operands:
10
6
Operation result from server=4
magnificent@magnificent:~/n$
```

```
magnificent@magnificent:~/n$ ./server
socket created sucessfully
bind sucessful
listen sucessful
accept sucessful
Result is: 4 * 3 = 12
magnificent@magnificent:~/n$
```

```
magnificent@magnificent:~/n$ ./client
socket created sucessfully
connect sucessful
Enter operation:
+
-
/
*
*
Enter operands]:
4
3
Operation result from server=12
magnificent@magnificent:~/n$
```

```
magnificent@magnificent:~/n$ ./server
socket created sucessfully
bind sucessful
listen sucessful
accept sucessful
Result is: 4 / 2 = 2
magnificent@magnificent:~/n$
```

```
magnificent@magnificent:~/n$ ./client
socket created sucessfully
connect sucessful
Enter operation:
+
-
/
*
/
Enter operands]:
4
2
Operation result from server=2
magnificent@magnificent:~/n$
```