**Example 1. Client Server - TCP**

**Server**

**Algorithm**

The activity involved in a tcp server are..........

1.Create a socket.

2.Bind it to the operating system.

3.Listen over it.

4.Accept connections.

5.Read/Write processes.

6.Close the socket.

**Program**

/\*Program to demonstrate the creation and usage of tcp socket.

\* This module acts as the tcp server which listens to a socket and accepts connection.

\* The server initially reads a message from the client and then writes a message to it.

\* This module should be compiled using the command. 'c++ tcpserver.cpp' and execute

'./a.out'.\*/

**//inculsion.**

#include<iostream>

#include<stdio.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<netdb.h>

#include<strings.h>

#include<unistd.h>

#include<stdlib.h>

int main()

{

int serversocket,clientsocket; **//Variables to store the socket id**

sockaddr\_in serveraddr,clientaddr; **//Variable to store the network host address**

socklen\_t len; **//Variable to store the address length**

//hostent\*server;

char message[50];

serversocket=socket(AF\_INET,SOCK\_STREAM,0); **//Creating Socket**

bzero((char\*)&serveraddr,sizeof(serveraddr)); **//Steps to include the host address**

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_port=htons(5030);

serveraddr.sin\_addr.s\_addr=INADDR\_ANY;

**//Binding the socket to the operating system**

bind(serversocket,(sockaddr\*)&serveraddr,sizeof(serveraddr));

bzero((char\*)&clientaddr,sizeof(clientaddr));

len=sizeof(clientaddr);

listen(serversocket,5); **//Listening over the socket**

printf("\n Waiting for client connectivity\n");

**//Accepting the connection**

clientsocket=accept(serversocket,(sockaddr\*)&clientaddr,&len);

printf("\n Client coonectivity recieved\n");

printf("\n Reading message from the client\n");

read(clientsocket,message,sizeof(message)); **//Reading activity**

printf("\n The client has send:\t%s\n",message);

printf("\n sending message to the client.\n");

**//writing activity**

write(clientsocket,"YOURMESSAGERECEIVED.",sizeof("YOURMESSAGERECEIVED."));

close(clientsocket);

close(serversocket);

}

**Client**

**Algorithm**

The different process involved in a TCP client process are.......

1.Create a socket

2.Connect newly created client socket to server 3.Read/Write processes.

4.Close the socket

**Program**

/\* Program to demonstrate the creation and use of tcp socket.

\* This module act as a tcp client which establishes a connection with a server socket

listening over a port.

\* This module initially establishes a connection,then sends a message to the client and finally

reads a message from it.

\* This module should be compiled into another folder say 'b' using the command.......'c++

tcpclient.cpp -o b' and execute using './b'

\* While executing the 'tcpserver' module should be executed first,then the 'tcpclient' module.

**//inculsion.**

#include<iostream>

#include<sys/types.h>

#include<stdio.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdlib.h>

#include<strings.h>

#include<unistd.h>

int main()

{

**//Variables to store the socket id**

int clientsocket;

**//Variable to store network host address**

sockaddr\_in serveraddr;

**//Variable to store address length**

socklen\_t len;

**//Variable to store the network byte order address**

hostent\*server;

char message[50];

**//Create a socket**

clientsocket=socket(AF\_INET,SOCK\_STREAM,0);

**//Steps involved in the server address creation**

bzero((char\*)&serveraddr,sizeof(serveraddr));

len=sizeof(serveraddr);

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_port=htons(5030);

server=gethostbyname("127.0.0.1");

bcopy((char\*)server->h\_addr,(char\*)&serveraddr.sin\_addr.s\_addr,sizeof(server->h\_addr));

printf("\n PRESS ENTER TO START THE CONNECTION PROCESS.\n");

fgets(message,2,stdin);

connect(clientsocket,(sockaddr\*)&serveraddr,sizeof(serveraddr));

printf("\n Sending message for server connection\n");

**//Sending the message**

write(clientsocket," HI I AM CLIENT",sizeof(" HI CLIENT"));

printf("\n Recieving message from server\n");

**//Receiving messages**

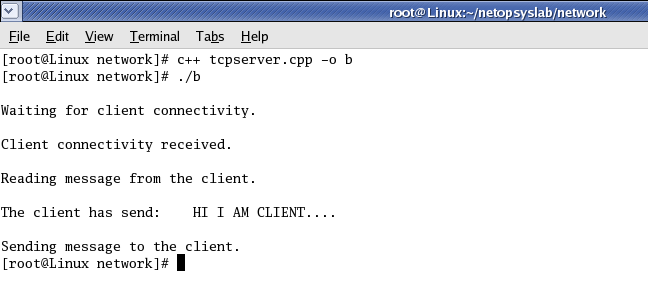
read(clientsocket,message,sizeof(message));

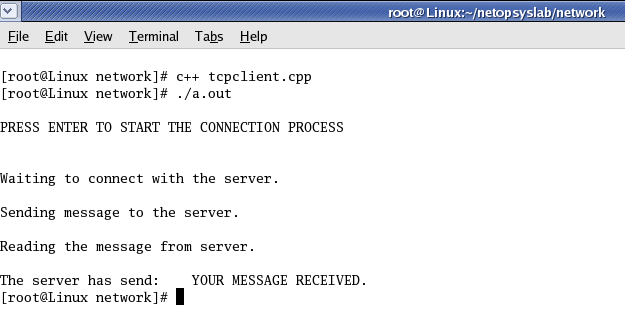
printf("\n Message recievd,\t%s",message);

close(clientsocket);

}

**Output**

****



**Example 2. CLIENT SERVER COMMUNICATION - TCP PROTOCOL**

**//Tcp server**

#include<stdio.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include<unistd.h>

int main()

{

int clntsocket,serversocket;

char msg[50],msg1[50];

struct sockaddr\_in serveraddr,clntaddr;

socklen\_t len;

bzero(msg,sizeof(msg));

serversocket=socket(AF\_INET,SOCK\_STREAM,0);

printf("\n SERVER PROCESS\n");

bzero((char \*)&serveraddr,sizeof(serveraddr));

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_port=htons(5015);

serveraddr.sin\_addr.s\_addr=INADDR\_ANY;

bind(serversocket,(struct sockaddr \*)&serveraddr,sizeof(serveraddr));

bzero((char\*)&clntaddr,sizeof(clntaddr));

len=sizeof(clntaddr);

printf("\n\*\*\*\*\*\n");

listen(serversocket,5);

printf("\nWaiting for client connectivity\n");

clntsocket=accept(serversocket,(struct sockaddr\*)&clntaddr,&len);

printf("\nClient connectivity is received\n");

printf("\nReading message from client\n");

while(1)

{

read(clntsocket,msg,sizeof(msg));

printf("\nClient message is \t%s",msg);

printf("\nsending acknowledgement to client\n");

write(clntsocket,"your msg is received",sizeof("your msg is received"));

}

close(clntsocket);

close(serversocket);

}

**// Tcp client**

#include<stdio.h>

#include<sys/types.h>

#include<netdb.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include<unistd.h>

int main()

{

int clntsocket;

struct sockaddr\_in serveraddr;

socklen\_t len;

struct hostent \*server;

char msg[50],msg1[50];

bzero(msg,sizeof(msg));

clntsocket=socket(AF\_INET,SOCK\_STREAM,0);

bzero((char \*)&serveraddr,sizeof(serveraddr));

len=sizeof(serveraddr);

serveraddr.sin\_family=AF\_INET;

serveraddr.sin\_port=htons(5015);

server=gethostbyname("127.0.0.1");

bcopy((char \*)server->h\_addr,(char \*)&serveraddr.sin\_addr.s\_addr,sizeof(server->h\_addr));

connect(clntsocket,(struct sockaddr \*)&serveraddr,sizeof(serveraddr));

printf("\nConnection process\n");

printf("\nsending message to server\n");

while(1)

{

write(clntsocket,"i am client",sizeof("i am client"));

printf("\nreceiving acknowledgement from server\n");

read(clntsocket,msg1,sizeof(msg1));

printf("\nacknowledgement is %s",msg1);

}

close(clntsocket);

return(0);

}

**CLIENT SERVER COMMUNICATION - UDP PROTOCOL**

**Udp server:**

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdio.h>

main()

{

struct sockaddr\_in server,client;

int s,n;

char a[10],b[50];

s=socket(AF\_INET,SOCK\_DGRAM,0);

server.sin\_port=1240;

server.sin\_family=AF\_INET;

server.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

bind(s,(struct sockaddr\*)&server,sizeof(server));

n=sizeof(client);

while(1)

{

printf("\nreceived from client");

recvfrom(s,a,sizeof(a),0,(struct sockaddr\*)&client,&n);

printf("%s",a);

printf("\n send to client");

scanf("%s",b);

if(strcmp(b,"close")==0)

break;

sendto(s,b,sizeof(b),0,(struct sockaddr\*)&client,n);

}

}

**Udp client:**

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdio.h>

main()

{

struct sockaddr\_in server;

int s,n;

char a[10],b[50];

s=socket(AF\_INET,SOCK\_DGRAM,0);

server.sin\_port=1240;

server.sin\_family=AF\_INET;

server.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

while(1)

{

printf("\nclient");

scanf("%s",b);

if(strcmp(b,"close")==0)

break;

sendto(s,b,sizeof(b),0,(struct sockaddr\*)&server,sizeof(server));

printf("\nclient2 ");

recvfrom(s,a,sizeof(a),0,NULL,NULL);

printf("%s",a);

}

}