**Socket Programming:**

1. **Message passing single client server :-**

**Server Code:**

 #include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<arpa/inet.h>

int main(){

int welcomeSocket, newSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

struct sockaddr\_storage serverStorage;

socklen\_t addr\_size;

welcomeSocket = socket(PF\_INET, SOCK\_STREAM, 0);

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(7891);

serverAddr.sin\_addr.s\_addr = inet\_addr("10.30.154.76");

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

bind(welcomeSocket, (struct sockaddr \*) &serverAddr, sizeof(serverAddr));

if(listen(welcomeSocket,5)==0)

printf("Listening\n");

else

 printf("Error\n");

 addr\_size = sizeof serverStorage;

newSocket = accept(welcomeSocket, (struct sockaddr \*) &serverStorage,

&addr\_size);

strcpy(buffer,"Hello World\n");

send(newSocket,buffer,13,0);

return 0;

}

**Client :-**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<arpa/inet.h>

int main(){

int welcomeSocket, newSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

struct sockaddr\_storage serverStorage;

socklen\_t addr\_size;

welcomeSocket = socket(PF\_INET, SOCK\_STREAM, 0);

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(7891);

serverAddr.sin\_addr.s\_addr = inet\_addr("10.30.154.76");

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

bind(welcomeSocket, (struct sockaddr \*) &serverAddr, sizeof(serverAddr));

if(listen(welcomeSocket,5)==0)

printf("Listening\n");

else

 printf("Error\n");

addr\_size = sizeof serverStorage;

newSocket = accept(welcomeSocket, (struct sockaddr \*) &serverStorage,

&addr\_size);

strcpy(buffer,"Hello World\n");

send(newSocket,buffer,13,0);

return 0;

}

1. **Multiple client message :-**

**Client :-**

 #include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include<arpa/inet.h>

#include <string.h>

int main(){

int clientSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

socklen\_t addr\_size;

clientSocket = socket(PF\_INET, SOCK\_STREAM, 0);

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(7891);

serverAddr.sin\_addr.s\_addr = inet\_addr("10.30.154.76");

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

addr\_size = sizeof serverAddr;

connect(clientSocket, (struct sockaddr \*) &serverAddr, addr\_size);

recv(clientSocket, buffer, 1024, 0);

printf("Data received: %s",buffer);

return 0;

}

**Server :-**

 #include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

 int welcomeSocket, newSocket;

 char buffer[1024];

 struct sockaddr\_in serverAddr;

 struct sockaddr\_storage serverStorage;

 socklen\_t addr\_size;

 serverAddr.sin\_family = AF\_INET;

  serverAddr.sin\_port = htons(7891);

 serverAddr.sin\_addr.s\_addr = inet\_addr("10.30.154.75");

 memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

 bind(welcomeSocket, (struct sockaddr \*) &serverAddr,

sizeof(serverAddr));

 if(listen(welcomeSocket,5)==0)

 printf("Listening\n");

 else

 printf("Error\n");

 addr\_size = sizeof serverStorage;

 newSocket = accept(welcomeSocket, (struct sockaddr \*) &serverStorage,

&addr\_size);

 strcpy(buffer,"Hello World\n");

 send(newSocket,buffer,13,0);

 return 0;

}

**TCP Chat Application**

**Sever.c**

 #include<sys/socket.h>

#include<sys/types.h>

#include<stdio.h>

#include<arpa/inet.h>

#include<netinet/in.h>

#include<string.h>

#include<unistd.h>

#define SER\_PORT 1200

int main()

{

int a,sersock,newsock,n;

char str[25],str2[25];

struct sockaddr\_in seraddr;

struct sockaddr\_in cliinfo;

socklen\_t csize=sizeof(cliinfo);

seraddr.sin\_family=AF\_INET;

seraddr.sin\_port=htons(SER\_PORT);

seraddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

if((sersock=socket(AF\_INET,SOCK\_STREAM,0))<0)

{

error("\n socket");

exit(0);

}

if(bind(sersock,(struct sockaddr \*)&seraddr,sizeof(seraddr))<0)

{

error("\nBIND");

exit(0);

}

if(listen(sersock,1)<0)

{

error("\n LISTEN");

}

if((newsock=accept(sersock,(struct sockaddr \*)&cliinfo,&csize))<0)

{

error("\n ACCEPT");

exit(0);

}s

else

printf("\n now connected to %s\n",inet\_ntoa(cliinfo.sin\_addr));

read(newsock,str,sizeof(str));

do

{

printf("\n client msg:%s",str);

printf("\n server msg:");

scanf("%s",str2);

write(newsock,str2,sizeof(str2));

listen(newsock,1);

read(newsock,str,sizeof(str));

n=strcmp(str,"BYE");

a=strcmp(str2,"BYE");

}

while(n!=0||a!=0);

close(newsock);

close(sersock);

return 0;

}

**Client.c**

 #include<stdio.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<netinet/in.h>

#include<unistd.h>

#define SER\_PORT 1200

int main(int count,char\*arg[])

{

int a,clisock;

char str[20],str2[20];

struct sockaddr\_in cliaddr;

cliaddr.sin\_port=htons(SER\_PORT);

cliaddr.sin\_family=AF\_INET;

cliaddr.sin\_addr.s\_addr=inet\_addr("10.30.154.76");

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

if(clisock<0)

{

perror("\n SOCKET");

exit(0);

}

if(connect(clisock,(struct sockaddr\*)&cliaddr,sizeof(cliaddr))<0)

{

perror("\n CONNECT");

exit(0);

}

printf("\nclient connected to %s",arg[1]);

printf("\nCLIENT");

scanf("%s",&str);

if(write(clisock,str,sizeof(str))<0)

{

printf("\n data could not be sent");

}

do

{

listen(clisock,1);

read(clisock,str2,sizeof(str2));

printf("\nserver msg:%s",str2);

printf("\nclient msg:");

scanf("%s",&str);

a=strcmp(str2,"BYE");

write(clisock,str2,sizeof(str2));

}

while(a!=0);

close(clisock);

return 0;

}

**UDP Chat Application**

**Server**

 #include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main(int argc, char \*\*argv){

if (argc != 2) {

 printf("Usage: %s <port>\n", argv[0]);

exit(0);

 }

char \*ip = "127.0.0.1";

int port = atoi(argv[1]);

int sockfd;

struct sockaddr\_in server\_addr, client\_addr;

char buffer[1024];

socklen\_t addr\_size;

int n;

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

if (sockfd < 0) {

 perror("[-]socket error");

 exit(1);

 }

memset(&server\_addr, '\0', sizeof(server\_addr));

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(port);

server\_addr.sin\_addr.s\_addr = inet\_addr(ip);

n = bind(sockfd, (struct sockaddr\*)&server\_addr, sizeof(server\_addr));

if (n < 0){

 perror("[-]bind error");

 exit(1);

 }

bzero(buffer, 1024);

addr\_size = sizeof(client\_addr);

recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr\*)&client\_addr,

&addr\_size);

printf("[+]Data recv: %s\n", buffer);

bzero(buffer, 1024);

strcpy(buffer, "Welcome to the UDP Server.");

sendto(sockfd, buffer, 1024, 0, (struct sockaddr\*)&client\_addr,

sizeof(client\_addr));

printf("[+]Data send: %s\n", buffer);

return 0;

}

**Client**

 #include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main(int argc, char \*\*argv){

if (argc != 2) {

 printf("Usage: %s <port>\n", argv[0]);

 exit(0);

 }

char \*ip = "127.0.0.1";

int port = atoi(argv[1]);

int sockfd;

struct sockaddr\_in addr;

char buffer[1024];

socklen\_t addr\_size;

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

memset(&addr, '\0', sizeof(addr));

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(port);

addr.sin\_addr.s\_addr = inet\_addr(ip);

bzero(buffer, 1024);

strcpy(buffer, "Hello World!");

sendto(sockfd, buffer, 1024, 0, (struct sockaddr\*)&addr,

sizeof(addr));

printf("[+]Data send: %s\n", buffer);

bzero(buffer, 1024);

addr\_size = sizeof(addr);

recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr\*)&addr,

&addr\_size);

printf("[+]Data recv: %s\n", buffer);

return 0;

}