**WAP for concurrent server and client communication.**

**serverCC.c**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<stdlib.h>

#include<arpa/inet.h>

#include <string.h>

void msg(int sock);

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

int socketfd, newsockfd, portno, clilen;

char buffer[256];

struct sockaddr\_in serv\_addr, cli\_addr;

int n, pid;

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

if (socketfd < 0) {

printf("ERROR opening socket");

return 0;

}

memset((char \*) &serv\_addr,'\0',sizeof(serv\_addr));

portno = 7000;

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

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

serv\_addr.sin\_port = htons(portno);

if (bind(socketfd, (struct sockaddr \*) &serv\_addr, sizeof(serv\_addr)) < 0) {

printf("ERROR on binding");

return 0;

}

listen(socketfd,5);

clilen = sizeof(cli\_addr);

while (1) {

newsockfd = accept(socketfd, (struct sockaddr \*) &cli\_addr, &clilen);

if (newsockfd < 0) {

printf("error on accept");

return 0;

}

pid = fork();

if (pid < 0) {

printf("error occured while forking");

return 0;

}

if (pid == 0) {

while(1){

close(socketfd);

msg(newsockfd);

}

close(newsockfd);

}

else {

close(newsockfd);

}

}

close(socketfd);

return 0;

}

void msg(int sock) {

int n;

char buffer[256],msg[1000];

bzero(buffer,256);

n = read(sock,buffer,255);

if (n < 0) {

printf("error reading from socket");

return 0;

}

printf("Here is the message: %s\n",buffer);

printf("server:");

scanf("%[^\n]s",msg);

//gets(msg);

n=send(sock,msg,150,0);

if (n < 0) {

printf("error writing to socket");

return ;

}

//close(sock);

}

**clientCC.c**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<stdlib.h>

#include<arpa/inet.h>

#include <string.h>

int main() {

int sockfd, portno, n;

struct sockaddr\_in serv\_addr;

// struct hostent \*server;

char buffer[256];

portno = 7000;

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

if (sockfd < 0) {

printf("error opening socket");

exit(1);

}

memset((char \*) &serv\_addr,'\0',sizeof(serv\_addr));

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

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

serv\_addr.sin\_port = htons(portno);

if (connect(sockfd, (struct sockaddr\*)&serv\_addr, sizeof(serv\_addr)) < 0) {

printf("error connecting");

return 0;

}

while(1)

{

printf("Please enter the message: ");

memset(buffer,' ',256);

fgets(buffer,255,stdin);

n = write(sockfd, buffer, strlen(buffer));

if (n < 0) {

printf("ERROR writing to socket");

return 0;

}

memset(buffer,' ',256);

n = recv(sockfd, buffer, 250,0);

if (n < 0) {

printf("ERROR reading from socket");

return 0;

}

printf("server:%s\n",buffer);

}

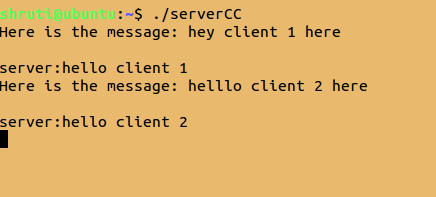
close(sockfd);

return 0;

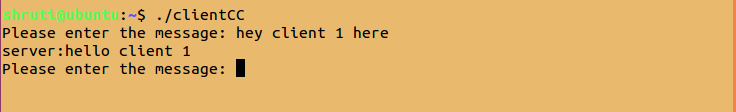
}

**OUTPUT:**

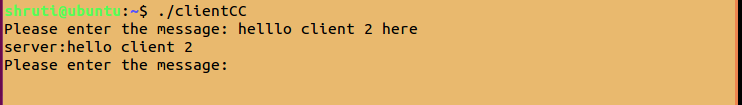
**Server:**



**Client 1:**



**Client 2:**



**WAP for communication between UDP client and server with the use of connect function.**

**client.c**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<stdlib.h>

#include<arpa/inet.h>

int main(){

int clientSocket;

char \*buffer,str[1024];

int sz=1024,n=0;

buffer=(char\*)malloc(sz\*sizeof(char));

struct sockaddr\_in serverAddr;

socklen\_t addr\_size;

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

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(7006);

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

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

addr\_size = sizeof serverAddr;

int c;

c=connect(clientSocket,(struct sockaddr \*)&serverAddr,sizeof(serverAddr));

if(c<0)

{

printf("connetion error!!");

return 0;

}

while(1)

{

printf("enter:");

getline(&buffer,&sz,stdin);

if(strcmp(buffer,"bye")==0)

{

n=write(clientSocket,buffer,1024);

if(n<0)

{

printf("error while sending\n");

return 0;

}

break;

}

n=write(clientSocket,buffer,1024);

if(n<0)

{

printf("error while sending\n");

return 0;

}

}

return 0;

}

**server.c**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

#include<stdlib.h>

#include<arpa/inet.h>

int main(){

int welcomeSocket, newSocket;

char buffer[1024];

char stringCpy[1024];

struct sockaddr\_in serverAddr,cliAddr;

int n=0;

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

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(7006);

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

bzero(serverAddr.sin\_zero, sizeof serverAddr.sin\_zero);

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

int sz=sizeof(cliAddr);

while(1){

printf("client:");

n=read(welcomeSocket,buffer,1024);

if(n<0)

{

printf("error occured while recieving\n");

return 0;

}

if(strcmp(buffer,"bye")==0)

{

printf("%s",buffer);

break;

}

printf("%s",buffer);

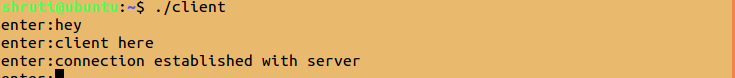
}

return 0;

}

**OUTPUT:**

**client:**



**server:**

