                          Socket program to established connection between client and server.

Server:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

main()

{

struct sockaddr\_in servaddr,cliaddr;

int p,b,nsd,sd,clilen;

printf("Enter the port number: ");

scanf("%d",&p);

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

if(nsd>=0)

printf("Socket value ok %d\n",p);

else

printf("Socket value not ok\n");

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(p);

b=bind(nsd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

if(b>=0)

printf("Connection is binded\n");

else

printf("Connection is not binded\n");

listen(nsd,5);

clilen=sizeof(cliaddr);

nsd=accept(nsd,(struct sockaddr\*)&cliaddr,&clilen);

if(nsd>=0)

printf("Connection is accepted\n");

else

printf("Connection is not accepted\n");

close(nsd);

}

CLIENT

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

main()

{

struct sockaddr\_in servaddr,cliaddr;

int c,pno,b,csd;

printf("Enter the port number: ");

scanf("%d",&pno);

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

if(csd>=0)

printf("Socket value is ok.\n");

else

printf("Socket value is not ok.\n");

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(pno);

c=connect(csd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

printf("%d",c);

if(c>=0)

printf("Connection Established.\n");

else

printf("Connection not Established.\n");

close(csd);

}

                                                                Socket Program to read current time from server and display it in client machine

Server:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<time.h>

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

{

    int listenfd = 0, connfd = 0;

    struct sockaddr\_in serv\_addr;

    char sendBuff[1025];

    time\_t ticks;

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

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

    memset(sendBuff, '0', sizeof(sendBuff));

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

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

    serv\_addr.sin\_port = htons(5000);

    bind(listenfd, (struct sockaddr\*)&serv\_addr, sizeof(serv\_addr));

    listen(listenfd, 10);

    while(1)

    {

        connfd = accept(listenfd, (struct sockaddr\*)NULL, NULL);

        ticks = time(NULL);

        snprintf(sendBuff, sizeof(sendBuff), "%.24s\r\n", ctime(&ticks));

        write(connfd, sendBuff, strlen(sendBuff));

        close(connfd);

        sleep(1);

     }

}

Client:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

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

{

    int sockfd = 0, n = 0;

    char recvBuff[1024];

    struct sockaddr\_in serv\_addr;

    if(argc != 2)

    {

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

        return 1;

    }

    memset(recvBuff, '0',sizeof(recvBuff));

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

    {

        printf("\n Error : Could not create socket \n");

        return 1;

    }

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

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

    serv\_addr.sin\_port = htons(5000);

    if(inet\_pton(AF\_INET, argv[1], &serv\_addr.sin\_addr)<=0)

    {

        printf("\n inet\_pton error occured\n");

        return 1;

    }

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

    {

       printf("\n Error : Connect Failed \n");

       return 1;

    }

    while ( (n = read(sockfd, recvBuff, sizeof(recvBuff)-1)) > 0)

    {

        recvBuff[n] = 0;

        if(fputs(recvBuff, stdout) == EOF)

        {

            printf("\n Error : Fputs error\n");

        }

    }

    if(n < 0)

    {

        printf("\n Read error \n");

    }

    return 0;

}

                                                                                                                                                                                CHATTING TCP

Server Side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include <time.h>

#include<string.h>

int main()

{

                struct sockaddr\_in servaddr,cliaddr;

                int port;

                printf("Enter port no: ");

                scanf("%d",&port);

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

                if (sd<0)

                printf("\nSocket not created.");

                else

                {

                                printf("\nSocket Created");

                                servaddr.sin\_family=AF\_INET;

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

                                servaddr.sin\_port=htons(port);

                                int b = bind(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

                                if (b<0)

                                printf("\nNot binded.");

                                else

                                {

                                                printf("\nBinded");

                                                int l = listen(sd,4);

                                                int clien = sizeof(cliaddr);

                                                                printf("\nListened");

                                                                int a= accept(sd,(struct sockaddr\*)&cliaddr,&clien);

                                                                if(a<0)

                                                                printf("\nConnection not established");

                                                                else

                                                                printf("\nConnection Established");

                                                                                char str[100];

                                                                                int ch=1,sen;

                                                                while(ch!=0)

                                                                {

                                                                                printf("\nPress 1 to send otherwise 0: ");

                                                                                scanf("%d",&sen);

                                                                                if(sen)

                                                                                {

                                                                                                printf("\nEnter the Message: ");

                                                                                                scanf("%s",str);

                                                                                                send(a,str,100,0);

                                                                                }

                                                                                recv(a,str,100,0);

                                                                                printf("\nRecieved Message: '%s'",str);

                                                                                printf("\nEnter 1 to continue or 0 to exit: ");

                                                                                scanf("%d",&ch);

                                                                }

                                }

                }

}

Client Side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<string.h>

main()

{

                struct sockaddr\_in servaddr,cliaddr;

                int port;

                printf("Enter port no: ");

                scanf("%d",&port);

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

                if (sd<0)

                printf("\nSocket not created.");

                else

                {

                                printf("\nSocket Created");

                                servaddr.sin\_family=AF\_INET;

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

                                servaddr.sin\_port=htons(port);

                                int c = connect(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

                                if (c<0)

                                printf("\nNot connected.");

                                else

                                {

                                                printf("\nConnected");

                                                char str[100];

                                                int ch=1,sen;

                                                while(ch!=0)

                                                                {

                                                                                printf("\nPress 1 to send otherwise 0: ");

                                                                                scanf("%d",&sen);

                                                                                if(sen)

                                                                                {

                                                                                                printf("\nEnter the Message: ");

                                                                                                scanf("%s",str);

                                                                                                send(sd,str,100,0);

                                                                                }

                                                                                recv(sd,str,100,0);

                                                                                printf("\nRecieved Message: '%s'",str);

                                                                                printf("\nEnter 1 to continue or 0 to exit: ");

                                                                                scanf("%d",&ch);

                                                                }

                                                close(sd);

                                }

                }

}

                                                                                                                                                                                ECHOOOOOOOO TCPP

Server:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<sys/socket.h>

void error(char \*msg)

{

perror(msg);

exit(1);

}

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

{

int sockfd, newsockfd, portno, clilen;

char buffer[256];

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

int n;

if (argc < 2)

{

fprintf(stderr,"ERROR, no port provided\n");

exit(1);

}

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

if (sockfd < 0)

{

error("ERROR opening socket");

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(argv[1]);

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

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

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

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

{

error("ERROR on binding");

}

listen(sockfd,5);

clilen = sizeof(cli\_addr);

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

if (newsockfd < 0)

{

error("ERROR on accept");

}

bzero(buffer,256);

n = read(newsockfd,buffer,255);

if (n < 0)

{

error("ERROR reading from socket");

}

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

n = write(newsockfd,"I got your message",18);

if (n < 0)

{

error("ERROR writing to socket");

}

return 0;

}

Client:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

void error(char \*msg)

{

perror(msg);

exit(0);

}

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

{

int sockfd, portno, n;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256];

if (argc < 3)

{

fprintf(stderr,"usage %s hostname port\n", argv[0]);

exit(0);

}

portno = atoi(argv[2]);

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

if (sockfd h\_addr,(char \*)&serv\_addr.sin\_addr.s\_addr,server->h\_length);

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

if (connect(sockfd,&serv\_addr,sizeof(serv\_addr)) < 0)

{

error("ERROR connecting");

}

printf("Please enter the message: ");

bzero(buffer,256);

fgets(buffer,255,stdin);

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

if (n < 0)

{

error("ERROR writing to socket");

}

bzero(buffer,256);

n = read(sockfd,buffer,255);

if (n < 0)

{

error("ERROR reading from socket");

}

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

return 0;

}

                                                                                                                                                                                ECHOOO UDDPPP

Server side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<string.h>

#include<arpa/inet.h>

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

{

int sockfd;

int n;

socklen\_t len;

char msg[1024];

struct sockaddr\_in servaddr,cliaddr;

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

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(5035);

printf("\nCircuit Binded");

bind(sockfd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

printf("\n Circuit is connected");

for(;;)

{

     printf("\n ");

     len=sizeof(cliaddr);

     n=recvfrom(sockfd,msg,1024,0,(struct sockaddr\*)&cliaddr,&len);

     printf("\n Client's Message : %s\n",msg);

     if(n<6)

     perror("send error");

     sendto(sockfd,msg,n,0,(struct sockaddr\*)&cliaddr,len);

}

return 0;

}

Client side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<string.h>

#include<arpa/inet.h>

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

{

int sockfd;

int n;

socklen\_t len;

char sendline[1024],recvline[1024];

struct sockaddr\_in servaddr;

strcpy(sendline,"");

printf("\n Enter the message : ");

scanf("%s",sendline);

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

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(5035);

connect(sockfd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

len=sizeof(servaddr);

sendto(sockfd,sendline,1024,0,(struct sockaddr\*)&servaddr,len);

n=recvfrom(sockfd,recvline,1024,0,NULL,NULL);

recvline[n]=0;

printf("\n Server's Echo : %s\n\n",recvline);

return 0;

}

                                                                                                                                                HOW MANY IP ADDR ARE CONECTED WHAT R THEY

Server side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

#include<arpa/inet.h>

main()

{

struct sockaddr\_in servaddr,cliaddr;

int p,b,nsd,sd,clilen;

char buffer[256];

printf("Enter the port number: ");

scanf("%d",&p);

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

if(nsd>=0)

printf("Socket value ok %d\n",p);

else

printf("Socket value not ok\n");

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(p);

b=bind(nsd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

if(b>=0)

printf("Connection is binded\n");

else

printf("Connection is not binded\n");

listen(nsd,5); // How many Clients I can Handle

clilen=sizeof(cliaddr);

nsd=accept(nsd,(struct sockaddr\*)&cliaddr,&clilen);

if(nsd>=0)

{

printf("Connection is accepted\n");

printf("%s\n",inet\_ntoa(( (struct sockaddr\_in \*)&servaddr.sin\_addr)->sin\_addr));

//printf("%s\n",inet\_ntoa(cliaddr.sin\_addr.s\_addr)); inet\_ntoa(( (struct sockaddr\_in \*)&servaddr.servaddr\_addr )->sin\_addr)

}

else

printf("Connection is not accepted\n");

close(nsd);

}

Client Side:

#include<sys/types.h>

#include<netinet/in.h>

#include<stdio.h>

main()

{

struct sockaddr\_in servaddr,cliaddr;

int c,pno,b,csd;  //socketfd = csd

printf("Enter the port number: ");

scanf("%d",&pno);

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

if(csd>=0)

printf("Socket value is ok.\n");

else

printf("Socket value is not ok.\n");

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(pno);

c=connect(csd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

printf("%d",c);

if(c>=0)

printf("Connection Established.\n");

else

printf("Connection not Established.\n");

close(csd);

}

                                                                                                                                                                CLIENT SIDE ENCRYPTIONNNNNNNNNNN

SERVER CODE:

#include<stdio.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<stdlib.h>

#include<string.h>

main()

{

                struct sockaddr\_in servaddr,cliaddr;

                int port;

                printf("enter the port no: ");

                scanf("%d",&port);

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

                if(sd<0)

                printf("\nsocket not created");

                else

                {

                                printf("\nsocket created");

                                servaddr.sin\_family=AF\_INET;

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

                                servaddr.sin\_port=htons(port);

                                int b=bind(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

                                if(b<0)

                                printf("\nnot binded");

                                else

                                {

                                                printf("\nBinded");

                                                int l=listen(sd,4);

                                                int clien;

                                                clien=sizeof(cliaddr);

                                                printf("\nlistened");

                                                int a=accept(sd,(struct sockaddr\*)&cliaddr,&clien);

                                                if(a<0)

                                                {

                                                                printf("not connected");

                                                }

                                                else

                                                {

                                                                printf("\nconnected");

                                                                int ch,sen;

                                                                char str[100];

                                                                recv(a,str,100,0);

                                                                printf("\nMessage Recieved: %s",str);

                                                                char str1[100];

                                                                int i;

                                                                for(i=0;i<strlen(str);i++)

                                                                                str1[i]=str[i]+2;

                                                                printf("\nEncrypted Message: %s",str1);

                                                                send(a,str1,100,0);

                                                }

                                }

                }

}

CLIENT CODE:

#include<stdio.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<string.h>

main()

{

                struct sockaddr\_in servaddr,cliaddr;

                int port;

                printf("enter the port no: ");

                scanf("%d",&port);

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

                if(sd<0)

                printf("socket not created\n");

                else

                {

                                printf("socket created\n");

                                servaddr.sin\_family=AF\_INET;

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

                                servaddr.sin\_port=htons(port);

                                int c=connect(sd,(struct sockaddr\*)&servaddr,sizeof(servaddr));

                                if(c<0)

                                printf("not connected");

                                else

                                {

                                                printf("\nConnected");

                                                char str[100],str1[100];

                                                printf("\nEnter the message: ");

                                                scanf("%s",str);

                                                send(sd,str,100,0);

                                                recv(sd,str,100,0);

                                                printf("\nSecret Code: %s",str);

                                                int i;

                                                for(i=0;i<strlen(str);i++)

                                                                                str1[i]=str[i]-2;

                                                printf("\nOrignal Message: %s",str1);

                                }

                                close(sd);

                }

}