**Client**

#include<stdio.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<unistd.h>

#include<stdlib.h>

int main()

{

int sd,len,cport;

struct sockaddr\_in ser;

char smsg[20],rmsg[20];

printf("\n\t Enter the Port : ");

scanf("%d",&cport);

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

{

printf("\n\t Error in socket");

return 0;

}

bzero(&ser,sizeof(ser));

ser.sin\_family=AF\_INET;

ser.sin\_port=htons(cport);

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

if(connect(sd,(struct sockaddr\*)&ser,sizeof(ser)) <0)

{

printf("\n\t Error in connect");

exit(-1);

}

do

{

printf("\n\t Enter the Client String :");

scanf("%s",smsg);

send(sd,smsg,20,0);

wait(20);

recv(sd,rmsg,20,0);

printf("\n\t Server Msg Received : %s ",rmsg);

}while((strcmp(rmsg,"exit"))!=0);

close(sd);

}

**Server**

#include<stdio.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<unistd.h>

#include<stdlib.h>

int main()

{

int sd,sd1,i,clilen,len,sport;

struct sockaddr\_in ser,cli;

char smsg[20],rmsg[20];

printf("\n\t Enter the Port : ");

scanf("%d",&sport);

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

{

printf("\n\t Error in socket");

return 0;

}

bzero(&ser,sizeof(ser));

ser.sin\_family=AF\_INET;

ser.sin\_port=htons(sport);

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

if(bind(sd,(struct sockaddr\*)&ser,sizeof(ser)) <0)

{

printf("\n\t Error in bind");

return 0;

}

if(listen(sd,5) <0)

{

printf("\n\t Error in Listen");

return 0;

}

do

{

printf("\n\t Enter the client no to be communicate : ");

scanf("%d",&i);

if(i==0)

exit(0);

printf("\n\t Client %d is connected",i);

clilen=sizeof(cli);

if((sd1=accept(sd,(struct sockaddr\*)&cli,&clilen)) <0)

{

printf("\n\t Error in Accept");

return 0;

}

printf("\n\t Accepted ");

do

{

recv(sd1,rmsg,20,0);

printf("\n\t Clinet Msg Received : %s ",rmsg);

printf("\n\t Enter Server String : ");

scanf("%s",smsg);

send(sd1,smsg,20,0);

wait(20);

} while((strcmp(smsg,"exit"))!=0);

} while(i!=0);

close(sd);

}