**Client**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<netdb.h>

#include<errno.h>

int main(){

int sock,byte\_rec,port;

char send\_data[1024],rec\_data[1024];

struct hostent \*host;

struct sockaddr\_in ser,cli;

host=gethostbyname("127.0.0.1");

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("Socket\n");

exit(1);

}

printf("socket created\n");

ser.sin\_family=AF\_INET;

printf("Enter the port number : ");

scanf("%d",&port);

ser.sin\_port=htons(port);

ser.sin\_addr= \*((struct in\_addr \*)host->h\_addr);

bzero(&(ser.sin\_zero),8);

if(connect(sock,(struct sockaddr\*)&ser,sizeof(struct sockaddr))==-1)

{

perror("connect\n");

exit(1);

}

while(1)

{

printf("\nsend:");

scanf("%s",send\_data);

send(sock,send\_data,strlen(send\_data),0);

byte\_rec=recv(sock,rec\_data,1024,0);

rec\_data[byte\_rec]='\0';

printf("Rec data : %s\n",rec\_data);

if(strcmp(rec\_data,"end")==0)

break;

}

close(sock);

return 0;

}

**Server**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<errno.h>

int main()

{

int sock,con,byte\_rec,flag=0,port;

char send\_data[1024],rec\_data[1024];

struct sockaddr\_in ser,cli;

int sin\_size;

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("Socket\n");

exit(1);

}

ser.sin\_family=AF\_INET;

printf("Enter the port number : ");

scanf("%d",&port);

ser.sin\_port=htons(port);

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

bzero(&(ser.sin\_zero),8);

if(bind(sock,(struct sockaddr\*)&ser,sizeof(struct sockaddr))==-1)

{

perror("can't bind\n");

exit(1);

}

if(listen(sock,3)==-1)

{

perror("listen\n");

exit(1);

}

printf("Waiting \n");

sin\_size=sizeof(struct sockaddr\_in);

con=accept(sock,(struct sockaddr\*)&cli,&sin\_size);

printf("Connected\n\n");

while(1)

{

byte\_rec=recv(con,rec\_data,1024,0);

rec\_data[byte\_rec]='\0';

printf("received data : %s\n\n",rec\_data);

send(con,rec\_data,strlen(rec\_data),0);

if(strcmp(rec\_data,"end")==0)

break;

}

close(con);

close(sock);

return 0;

}