client server communication using tcp

Cli.c

#include<stdio.h>

#include <unistd.h>

#include<string.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main()

{ int client;

struct sockaddr\_in serveraddr;

char buffer[1024];

char s[1024];

int port=1026;

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

serveraddr.sin\_family=AF\_INET;

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

serveraddr.sin\_port = htons( port );

if( connect(client,(struct sockaddr\*)&serveraddr,sizeof(serveraddr))>0)

printf("connected");

while(1)

{ printf("Enter the Data to send ");

scanf("%s",s);

send(client,s,strlen(s),0);

if (strcmp(s,"stop")==0)

break;

int n=read(client,buffer,1023);

buffer[n]='\0';

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

}

}

Ser.c

#include<stdio.h>

#include <unistd.h>

#include<string.h>

#include <netinet/in.h>

#include <arpa/inet.h>

//#include <sys/socket.h>

//#include<stdlib.h>

int main()

{

int server,client;

char buffer[1024];

char s[1024];

int PORT=1026;

struct sockaddr\_in servaddr,clientaddr;

int addrlen = sizeof(clientaddr);

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

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons( PORT );

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

listen(server,3);

client=accept(server,(struct sockaddr\*)&clientaddr,&addrlen);

printf("connected");

while(1)

{

int n= read(client,buffer,1023);

buffer[n]='\0';

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

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

break;

printf("Enter The Data to send");

scanf("%s",s);

send(client,s,strlen(s),0);

}

}

**Time server udp**

Client.c

#include<sys/socket.h>

#include<stdio.h>

#include<string.h>

#include <arpa/inet.h>

void main()

{

int servsock,len;

char buffer[1024];

struct sockaddr\_in clientaddr,servaddr;

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

clientaddr.sin\_family=AF\_INET;

clientaddr.sin\_port=htons(2034);

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

sendto(servsock,"Time",strlen("Time"),MSG\_CONFIRM,(const struct sockaddr

\*)&clientaddr,sizeof(clientaddr));

int i=recvfrom(servsock,buffer,1024,MSG\_WAITALL,(struct sockaddr\*)&clientaddr,&len);

buffer[len]='\0';

printf("received %s",buffer);

}

Ser.c

#include<sys/socket.h>

#include<stdio.h>

#include<string.h>

#include <arpa/inet.h>

#include<time.h>

void main()

{

int servsock,len;

char buffer[1024];

time\_t tim;

struct sockaddr\_in clientaddr,servaddr;

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

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(2034);

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

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

int i=recvfrom(servsock,(char \*)buffer,1024,MSG\_WAITALL,(struct

sockaddr\*)&clientaddr,&len);

buffer[len]='\0';

printf("received %s",buffer);

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

{

time(&tim);

sendto(servsock,ctime(&tim),strlen(ctime(&tim)),MSG\_CONFIRM,( struct sockaddr

\*)&clientaddr,len);

}

}

**selective recive**

**Recevierselective.c**

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include<stdlib.h>

#include <arpa/inet.h>

struct packet

{ int data;

int seq;

};

struct ackn

{

int seq;

int ack;

};

void main()

{ int receiver,sender,len,c,t,wind,j,l,k,r,ran,f,i,n;

int opt=1;

struct ackn a[20];

char buffer[1024];

struct packet p[25];

struct sockaddr\_in senderadd,recvadr;

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

recvadr.sin\_family=AF\_INET;

recvadr.sin\_port=htons(2069);

recvadr.sin\_addr.s\_addr=inet\_addr("192.168.1.19");

setsockopt(receiver, SOL\_SOCKET,

SO\_REUSEADDR | SO\_REUSEPORT, &opt,

sizeof(opt));

if (bind(receiver,(struct sockaddr\*)&recvadr,sizeof(recvadr))<0)

printf("Not Connected\n");

listen(receiver,3);

sender=accept(receiver,(struct sockaddr\*)&senderadd,&len);

i=read(sender,buffer,1024);

buffer[i]='\0';

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

send(sender,"hello",strlen("hello"),0);

j=1;

k=1;

c=0;

while(j<=4)

{ r=read(sender,(char \*)&p[j],sizeof(p[j]));

j++;

}

t=4;

j=1;

n=8;

while(c<8)

{

ran=rand()%3;

if(ran==0)

{ a[j].ack=-1;

a[j].seq=p[j].seq;

}

else{ a[j].ack=1;

a[j].seq=p[j].seq;

}

if(a[j].ack==1)

{ printf("Received Packet with data %d and seq

%d\n",p[j].data,p[j].seq);

send(sender,(char\*)&a[j],sizeof(a[j]),0);

c++;

j++;

}

else

{ send(sender,(char\*)&a[j],sizeof(a[j]),0);

printf("\n\n");

j++; n++;

}

if(t< n)

{ t++;read(sender,(char\*)&p[t],sizeof(p[t])); }

}

close(receiver);

}

Senderselective

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include <arpa/inet.h>

struct packet

{ int data;

int seq;

};

struct ackn

{

int seq;

int ack;

};

void main()

{

int sender,len,wind,j,l,k,c,i,r;

struct ackn a;

char buffer[1024];

struct packet p[25];

struct sockaddr\_in receiver;

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

receiver.sin\_family=AF\_INET;

receiver.sin\_port=htons(2069);

receiver.sin\_addr.s\_addr=inet\_addr("192.168.1.19");

connect(sender,(struct sockaddr\*)&receiver,sizeof(receiver));

send(sender,"hai",strlen("hai"),0);

i=read(sender,buffer,1024);

buffer[i]='\0';

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

j=1;

l=1;

k=1;

printf("Sending frame \nNo: of frames=8\nWindow size=4\n");

while(j<=8)

{ printf("Enter frame data");

scanf("%d",&p[j].data);

p[j].seq=j;

j++;

}

while(l<=4)

{ send(sender,(char\*)&p[l],sizeof(p[l]),0);

printf("sending packet with data %d and seq no %d\n",p[l].data,p[l].seq);

l++;

}

c=0;

n=8;

while(c<8)

{

r=read(sender,(char \*)&a,sizeof(a));

if (a.ack==1)

{ c=c+1;

printf("received ack for packet %d\n",a.seq);

if(l<=8)

{printf("sending packet with data %d and seq no

%d\n",p[l].data,p[l].seq);

send(sender,(char\*)&p[l],sizeof(p[l]),0);

l++;}

}

else if (a.ack==-1)

{ printf("\n\ntime expired for packet %d\n",a.seq);

printf("resending packet with data %d and seq no

%d\n",p [a.seq].data,a.seq);

send(sender,(char\*)&p[a.seq],sizeof(p[a.seq]),0);

}

}

close(sender);

}

**Stop and wait**

Sender.c

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include <arpa/inet.h>

struct packet

{ int data;

int seq;

};

void main()

{

int sender,len,i,r,flag;

int ack;

char buffer[1024];

int opt=1;

struct packet p;

struct sockaddr\_in receiver,sendadr;

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

receiver.sin\_family=AF\_INET;

receiver.sin\_port=htons(2065);

receiver.sin\_addr.s\_addr=inet\_addr("192.168.1.19");

if( connect(sender,(struct sockaddr\*)&receiver,sizeof(receiver))>0)

printf("connected");

int x= send(sender,"hai",strlen("hai"),0);

printf("sent hai\n");

i=read(sender,buffer,1023);

buffer[i]='\0';

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

i=1;

flag=0;

while(i<10)

{ if (flag==0)

{printf("Sender : Enter the data to sent:");

scanf("%d",&p.data);

p.seq=i;

}

send(sender,(char\*)&p,sizeof(p),0);

printf("send packet with data %d with seq no: %d\n",p.data,p.seq);

r=read(sender,(char\*)&ack,sizeof(ack));

if(ack==0 || ack==1)

{ sleep(3); printf("\n\ntime expired resending the packet\n");flag=1;}

else

{ printf("Received ack\n"); i++;flag=0;}

}

close(sender);

}

Receiver.c

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include <arpa/inet.h>

#include<stdlib.h>

struct packet

{ int data;

int seq;

};

void main()

{

int receiver,sende,len,i,r,x;

int ack;

char buffer[1024];

int opt=1;

struct packet p;

struct sockaddr\_in sender,receiv;

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

receiv.sin\_family=AF\_INET;

receiv.sin\_port=htons(2065);

receiv.sin\_addr.s\_addr=inet\_addr("192.168.1.19");

setsockopt(receiver, SOL\_SOCKET,SO\_REUSEADDR | SO\_REUSEPORT, &opt,sizeof(opt));

bind(receiver,(struct sockaddr\*)&receiv,sizeof(receiv));

listen(receiver,3);

sende=accept(receiver,(struct sockaddr\*)&sender,&len);

i= read(sende,buffer,1023);

buffer[i]='\0';

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

send(sende,"hai",strlen("hai"),0);

printf("sent hai\n");

i=1;

while(i<10)

{ r=read(sende,(char\*) &p,sizeof(p));

x=rand()%6;

send(sende,(char \*)&x,sizeof(x),0);

if(x==0)

printf("Received Packet data %d with seq %d sending ack\n\n\n packet duplicated: ",p.data,p.seq);

else if (x!=1)

{ i++; printf("Received Packet data %d with seq %d sending ack\n",p.data,p.seq);}

}

close(receiver);

}

**GO BACK N**

Sender.c

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include <arpa/inet.h>

struct packet

{ int data;

int seq;

};

void main()

{

int sender,len,i,r,flag;

int ack;

char buffer[1024];

int opt=1;

struct packet p;

struct sockaddr\_in receiver,sendadr;

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

receiver.sin\_family=AF\_INET;

receiver.sin\_port=htons(2065);

receiver.sin\_addr.s\_addr=inet\_addr("192.168.43.58");

if( connect(sender,(struct sockaddr\*)&receiver,sizeof(receiver))>0)

printf("connected");

int x= send(sender,"hai",strlen("hai"),0);

printf("sent hai\n");

i=read(sender,buffer,1023);

buffer[i]='\0';

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

i=1;

flag=0;

while(i<10)

{ if (flag==0)

{printf("Sender : Enter the data to sent:");

scanf("%d",&p.data);

p.seq=i;

}

send(sender,(char\*)&p,sizeof(p),0);

printf("send packet with data %d with seq no: %d\n",p.data,p.seq);

r=read(sender,(char\*)&ack,sizeof(ack));

if(ack==0 || ack==1)

{ sleep(3); printf("\n\ntime expired resending the packet\n");flag=1;}

else

{ printf("Received ack\n"); i++;flag=0;}

}

close(sender);

}

Receiver.c

#include<sys/socket.h>

#include<stdio.h>

#include<unistd.h>

#include<string.h>

#include <arpa/inet.h>

#include<stdlib.h>

struct packet

{ int data;

int seq;

};

void main()

{

int receiver,sende,len,i,r,x;

int ack;

char buffer[1024];

int opt=1;

struct packet p;

struct sockaddr\_in sender,receiv;

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

receiv.sin\_family=AF\_INET;

receiv.sin\_port=htons(2065);

receiv.sin\_addr.s\_addr=inet\_addr("192.168.43.58");

setsockopt(receiver, SOL\_SOCKET,SO\_REUSEADDR | SO\_REUSEPORT, &opt,

sizeof(opt));

bind(receiver,(struct sockaddr\*)&receiv,sizeof(receiv));

listen(receiver,3);

sende=accept(receiver,(struct sockaddr\*)&sender,&len);

i= read(sende,buffer,1023);

buffer[i]='\0';

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

send(sende,"hai",strlen("hai"),0);

printf("sent hai\n");

i=1;

while(i<10)

{ r=read(sende,(char\*) &p,sizeof(p));

x=rand()%6;

send(sende,(char \*)&x,sizeof(x),0);

if(x==0)

printf("Received Packet data %d with seq %d sending ack\n\n\n packet duplicated: ",p.data,p.seq);

else if (x!=1)

{ i++; printf("Received Packet data %d with seq %d sending ack\n",p.data,p.seq);}

}

close(receiver);

}

**FTP**

Server.c

#include<stdio.h>

#include <unistd.h>

#include<string.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include<stdlib.h>

int main()

{ char buffer[1024];

char s[1024];

int opt=1;

int PORT=1026;

int server,client;

FILE \*fp;

struct sockaddr\_in servaddr,clientaddr;

int addrlen = sizeof(clientaddr);

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

if (server<0)

{ printf("socket doest not created\n");

exit(0);

}

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons( PORT );

setsockopt(server, SOL\_SOCKET,

SO\_REUSEADDR | SO\_REUSEPORT, &opt,

sizeof(opt));

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

if (b<0)

{ printf("Error in binding\n");

exit(0);

}

listen(server,3);

printf("\n");

client=accept(server,(struct sockaddr\*)&clientaddr,&addrlen);

if (client<0)

{ exit(0);

}

printf("Connection accepted from %s\n",inet\_ntoa(clientaddr.sin\_addr));

int n= read(client,buffer,1024);

buffer[n]='\0';

printf("file Name received is %s\n",buffer);

fp=fopen(buffer,"r");

if (fp==NULL)

send(client,"File Not Present",strlen("File Not Present"),0);

else

{

while(fgets(buffer,1024,fp)!=NULL)

{ send(client,buffer,strlen(buffer),0);

}

printf("file data sent");

fclose(fp);

}

close( client);

close(server);

}

Client.c

#include<stdio.h>

#include <unistd.h>

#include<string.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int main()

{ int client;

struct sockaddr\_in serveraddr;

char buffer[1024];

char s[1024];

int port=1026;

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

serveraddr.sin\_family=AF\_INET;

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

serveraddr.sin\_port = htons( port );

if( connect(client,(struct sockaddr\*)&serveraddr,sizeof(serveraddr))>0)

printf("connected");

send(client,"udplient.c",strlen("udpclient.c"),0);

while(1)

{ int n=read(client,buffer,1024);

if(n==0)

break;

buffer[n]='\0';

printf("%s",buffer);

}

close(client);

}