**A chat between client and server-TCP**

**SERVER:**

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#define MAX 80

#define PORT 347634

#define SA struct sockaddr

void func(int sockfd)

{

char buff[MAX];

int n,a;

for(;;)

{

bzero(buff,sizeof(buff));

printf("Enter the string : ");

n=0;

while((buff[n++]=getchar())!='\n');

write(sockfd,buff,sizeof(buff));

bzero(buff,sizeof(buff));

read(sockfd,buff,sizeof(buff));

printf("From Server : %s",buff);

if((strncmp(buff,"exit",4))==0)

{

printf("Client Exit...\n");

break;

}

}

}

int main()

{

int sockfd,connfd;

struct sockaddr\_in servaddr,cli;

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

if(sockfd==-1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_addr.s\_addr=inet\_addr("127.0.0.2");

servaddr.sin\_port=htons(PORT);

if(connect(sockfd,(SA \*)&servaddr,sizeof(servaddr))!=0)

{

printf("connection with the server failed...\n");

exit(0);

}

else

printf("connected to the server..\n");

func(sockfd);

close(sockfd);

}

**OUTPUT:**

[211716205052@Putty ~]$ vi chatserver.c

[211716205052@Putty ~]$ gcc chatserver.c

[211716205052@Putty ~]$ ./a.out

Socket successfully created..

connected to the server..

Enter the string : hai client

From Server : hey server

Enter the string : fine.bye

From Server : bye server

Enter the string : exit

From Server : exit

Client Exit...

**CLIENT:**

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<stdlib.h>

#include<string.h>

#define MAX 80

#define PORT 347634

#define SA struct sockaddr

void func(int sockfd)

{

char buff[MAX];

int n;

for(;;)

{

bzero(buff,MAX);

read(sockfd,buff,sizeof(buff));

printf("From client: %s\t To client : ",buff);

bzero(buff,MAX);

n=0;

while((buff[n++]=getchar())!='\n');

write(sockfd,buff,sizeof(buff));

if(strncmp("exit",buff,4)==0)

{

printf("Server Exit...\n");

break;

}

}

}

int main()

{

int sockfd,connfd,len;

struct sockaddr\_in servaddr,cli;

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

if(sockfd==-1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(PORT);

if((bind(sockfd,(SA\*)&servaddr, sizeof(servaddr)))!=0)

{

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded..\n");

if((listen(sockfd,5))!=0)

{

printf("Listen failed...\n");

exit(0);

}

else

printf("Server listening..\n");

len=sizeof(cli);

connfd=accept(sockfd,(SA \*)&cli,&len);

if(connfd<0)

{

printf("server acccept failed...\n");

exit(0);

}

else

printf("server acccept the client...\n");

func(connfd);

close(sockfd);

}

**OUTPUT:**

[211716205052@Putty ~]$ vi chatclient.c

[211716205052@Putty ~]$ gcc chatclient.c

[211716205052@Putty ~]$ ./a.out

Socket successfully created..

Socket successfully binded..

Server listening..

server acccept the client...

From client: hai client

To client : hey server

From client: fine.bye

To client : bye server

From client: exit

To client : exit

Server Exit...

**A chat between client and server-UDP**

**SERVER:**

#include<stdio.h>

#include<netinet/in.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#define MAX 80

#define PORT 454545

#define SA struct sockaddr

void func(int sockfd)

{

char buff[MAX];

int n,clen;

struct sockaddr\_in cli;

clen=sizeof(cli);

for(;;)

{

bzero(buff,MAX);

recvfrom(sockfd,buff,sizeof(buff),0,(SA \*)&cli,&clen);

printf("From client--> %s To client-->",buff);

bzero(buff,MAX);

n=0;

while((buff[n++]=getchar())!='\n');

sendto(sockfd,buff,sizeof(buff),0,(SA \*)&cli,clen);

if(strncmp("exit",buff,4)==0)

{

printf("Server Exit...\n");

break;

}

}

}

int main()

{

int sockfd;

struct sockaddr\_in servaddr;

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

if(sockfd==-1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(PORT);

if((bind(sockfd,(SA \*)&servaddr,sizeof(servaddr)))!=0)

{

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded..\n");

func(sockfd);

close(sockfd);

}

**OUTPUT:**

[211716205052@Putty ~]$ vi chatserverudp.c

[211716205052@Putty ~]$ gcc chatserverudp.c

[211716205052@Putty ~]$ ./a.out

Socket successfully created..

Socket successfully binded..

From client-->hi

To client-->hi

From client-->hi

To client-->bye

From client-->bye

To client-->exit

Server Exit…

**CLIENT:**

#include<sys/socket.h>

#include<netdb.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

#define MAX 80

#define PORT 454545

#define SA struct sockaddr

int main()

{

char buff[MAX];

int sockfd,len,n;

struct sockaddr\_in servaddr;

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

if(sockfd==-1)

{

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr,sizeof(len));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(PORT);

len=sizeof(servaddr);

for(;;)

{

printf("\nEnter string : ");

n=0;

while((buff[n++]=getchar())!='\n');

sendto(sockfd,buff,sizeof(buff),0,(SA \*)&servaddr,len);

bzero(buff,sizeof(buff));

recvfrom(sockfd,buff,sizeof(buff),0,(SA \*)&servaddr,&len);

printf("From Server : %s\n",buff);

if(strncmp("exit",buff,4)==0)

{

printf("Client Exit...\n");

break;

}

}

close(sockfd);

}

**OUTPUT:**

[211716205052@Putty ~]$ vi chatserverudp.c

[211716205052@Putty ~]$ gcc chatserverudp.c

[211716205052@Putty ~]$ ./a.out

Socket successfully created..

Enter string : hi

hi

From server : hi

Enter string : bye

From server : bye

Enter string : From Server : exit

Client Exit…