**EX:1** **SOCKET PROGRSMMING**

**SERVER**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

int welcomeSocket, newSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

struct sockaddr\_storage serverStorage;

socklen\_t addr\_size;

/\*---- Create the socket. The three arguments are: ----\*/

/\* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in this case) \*/

welcomeSocket = socket(PF\_INET, SOCK\_STREAM, 0);

/\*---- Configure settings of the server address struct ----\*/

/\* Address family = Internet \*/

serverAddr.sin\_family = AF\_INET;

/\* Set port number, using htons function to use proper byte order \*/

serverAddr.sin\_port = htons(8060);

/\* Set IP address to localhost \*/

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

/\* Set all bits of the padding field to 0 \*/

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

/\*---- Bind the address struct to the socket ----\*/

bind(welcomeSocket, (struct sockaddr \*) &serverAddr, sizeof(serverAddr));

/\*---- Listen on the socket, with 5 max connection requests queued ----\*/

if(listen(welcomeSocket,5)==0)

printf("Listening\n");

else

printf("Error\n");

/\*---- Accept call creates a new socket for the incoming connection ----\*/

addr\_size = sizeof serverStorage;

newSocket = accept(welcomeSocket, (struct sockaddr \*) &serverStorage, &addr\_size);

strcpy(buffer,"Hello World\n");

send(newSocket,buffer,13,0);

return 0;

**OUTPUT:**

[211716205024@Putty ~]$ vi server.c

[211716205024@Putty ~]$ gcc server.c

[211716205024@Putty ~]$ ./a.out

Listening

**CLIENT:**

#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

int clientSocket;

char buffer[1024];

struct sockaddr\_in serverAddr;

socklen\_t addr\_size;

/\*---- Create the socket. The three arguments are: ----\*/

/\* 1) Internet domain 2) Stream socket 3) Default protocol (TCP in this case) \*/

clientSocket = socket(PF\_INET, SOCK\_STREAM, 0);

/\*---- Configure settings of the server address struct ----\*/

/\* Address family = Internet \*/

serverAddr.sin\_family = AF\_INET;

/\* Set port number, using htons function to use proper byte order \*/

serverAddr.sin\_port = htons(8060);

/\* Set IP address to localhost \*/

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

/\* Set all bits of the padding field to 0 \*/

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

/\*---- Connect the socket to the server using the address struct ----\*/

addr\_size = sizeof serverAddr;

connect(clientSocket, (struct sockaddr \*) &serverAddr, addr\_size);

/\*---- Read the message from the server into the buffer ----\*/

recv(clientSocket, buffer, 1024, 0);

/\*---- Print the received message ----\*/

printf("Data received: %s",buffer);

return 0;

}

**OUTPUT:**

[211716205024@Putty ~]$ vi client.c

[211716205024@Putty ~]$ vi client.c

[211716205024@Putty ~]$ gcc client.c

[211716205024@Putty ~]$ ./a.out

Data received: Hello World

**EX:2 DATE AND TIME**

**SERVER:**

#include"netinet/in.h"

#include"sys/socket.h"

#include"stdio.h"

#include"string.h"

#include"time.h"

main( )

{

struct sockaddr\_in sa;

struct sockaddr\_in cli;int sockfd,conntfd;int len,ch;char str[100];

time\_t tick;

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

if(sockfd<0)

{

printf("error in socket\n");

exit(0);

}

else printf("Socket opened");

bzero(&sa,sizeof(sa));

sa.sin\_port=htons(2108);

sa.sin\_addr.s\_addr=htonl(0);

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

{

printf("Error in binding\n");

}

else

printf("Binded Successfully");

listen(sockfd,50);

for(;;)

{

len=sizeof(ch);

conntfd=accept(sockfd,(struct sockaddr\*)&cli,&len);

printf("Accepted");

tick=time(NULL);

snprintf(str,sizeof(str),"%s",ctime(&tick));

printf("%s",str);write(conntfd,str,100);

}

}

**OUTPUT:**

[211716205024@Putty ~]$ vi serverdatetime.c

[211716205024@Putty ~]$ gcc serverdatetime.c

[211716205024@Putty ~]$ ./a.out

Socket openedBinded SuccessfullyAcceptedFri Jul 6 14:26:41 2018

**CLIENT:**

#include"netinet/in.h"

#include"sys/socket.h"

#include"stdio.h"

main()

{

struct sockaddr\_in sa,cli;

int n,sockfd;

int len;char buff[100];

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

if(sockfd<0){ printf("\nError in Socket");

exit(0);

}

else printf("\nSocket is Opened");

bzero(&sa,sizeof(sa));

sa.sin\_family=AF\_INET;

sa.sin\_port=htons(2108);

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

{

printf("\nError in connection failed");

exit(0);

}

else

printf("\nconnected successfully");

if(n=read(sockfd,buff,sizeof(buff))<0)

{

printf("\nError in Reading");

exit(0);

}

else

{printf("\nMessage Read %s",buff);

}}

**OUTPUT**:

[211716205024@Putty ~]$ vi clientdatetime.c

[211716205024@Putty ~]$ gcc clientdatetime.c

[211716205024@Putty ~]$ ./a.out

Socket is Opened

connected successfully

Message Read Fri Jul 6 14:26:41 2018

**EX:3a 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:**

[211716205024@Putty ~]$ vi chatserver.c

[211716205024@Putty ~]$ gcc chatserver.c

[211716205024@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:**

[211716205024@Putty ~]$ vi chatclient.c

[211716205024@Putty ~]$ gcc chatclient.c

[211716205024@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...

**EX:3b 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:**

[211716205024@Putty ~]$ vi chatserverudp.c

[211716205024@Putty ~]$ gcc chatserverudp.c

[211716205024@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:**

[211716205024@Putty ~]$ vi chatserverudp.c

[211716205024@Putty ~]$ gcc chatserverudp.c

[211716205024@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…