**CN LAB**

**ASSIGNMENT**

1906368

**1. Write the client side and server side program for connection oriented socket (TCP) using C where the Client sends two numbers to Server. Now Server calculates the sum of that two numbers and sends that sum result to the Client. Then Client displays that sum.**

**SERVER**

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd,confd;

int buffer[1024];

int hello=0;

struct sockaddr\_in servaddr, cliaddr;

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

printf("Socket creation failed\n");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

memset(&cliaddr, 0, sizeof(cliaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(1908);

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

{

printf("bind failed");

exit(EXIT\_FAILURE);

}

int len, n;

len = sizeof(cliaddr);

listen(sockfd,1);

if((confd=accept(sockfd,(struct sockaddr\*)&cliaddr,&len))<0)

{

printf("Acceptance Failed\n");

exit(0);

}

n = recv(confd, buffer,sizeof(buffer),0);

printf("Client:\n");

for(int i=0;i<2;i++)

{

printf("%d ",buffer[i]);

}

for(int i=0;i<2;i++)

{

hello+=buffer[i];

}

send(confd,&hello,sizeof(hello),0);

close(sockfd);

close(confd);

return 0;

}

**CLIENT**

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd;

int buffe;

int hell[1024];

struct sockaddr\_in servaddr;

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

printf("Socket Cretion Failed\n");

exit(EXIT\_FAILURE);

}

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(1908);

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

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

{

printf("Connect failed\n");

exit(0);

}

int n, len;

int n1=2;

printf("Enter the 2 number:\n");

for(int i=0;i<n1;i++)

{

scanf("%d",&hell[i]);

}

send(sockfd,hell,sizeof(hell),0);

n=recv(sockfd,&buffe,sizeof(buffe),0);

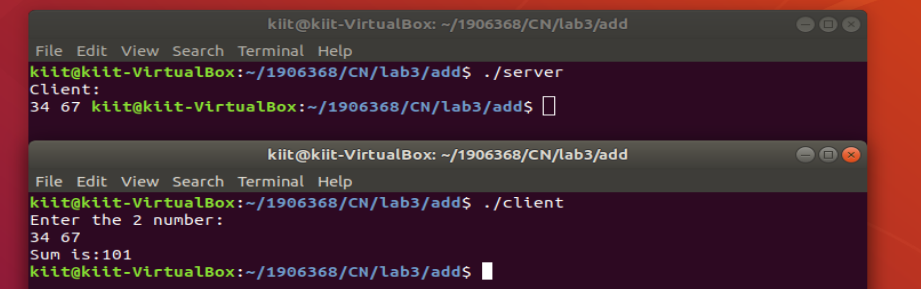
printf("Sum is:%d\n",buffe);

close(sockfd);

return 0;

}

**Output:**

****

**2. Write the client side and server side program for connection oriented socket (TCP) using C where the Client sends an Array with size 'n' to Server. Now Server sorts that array in ascending order and sends that sorted array to the Client. Now Client displays that sorted array.**

**SERVER**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd,confd;

int buffer[1024];

struct sockaddr\_in servaddr, cliaddr;

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

printf("Socket creation failed\n");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

memset(&cliaddr, 0, sizeof(cliaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(1908);

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

{

printf("bind failed");

exit(EXIT\_FAILURE);

}

int len, n;

len = sizeof(cliaddr);

listen(sockfd,1);

if((confd=accept(sockfd,(struct sockaddr\*)&cliaddr,&len))<0)

{

printf("Acceptance Failed\n");

exit(0);

}

n = recv(confd, buffer,sizeof(buffer),0);

printf("Client: ");

for(int i=0;i<5;i++)

{

printf("%d ",buffer[i]);

}

for(int i=0;i<5;i++)

{for(int j=i+1;j<5;j++)

if(buffer[i]>buffer[j])

{

int t=buffer[i];

buffer[i]=buffer[j];

buffer[j]=t;

}

}

send(confd,&buffer,sizeof(buffer),0);

close(sockfd);

close(confd);

return 0;

}

**CLIENT**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd;

int buffe[1024];

int hell[1024];

struct sockaddr\_in servaddr;

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

printf("Socket Cretion Failed\n");

exit(EXIT\_FAILURE);

}

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(1908);

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

if(connect(sockfd,(const struct sockaddr \*)&servaddr,sizeof(servaddr))<0){

printf("Connect failed\n");

exit(0);

}

int n, len;

int n1=5;

printf("Enter the array:-\n");

for(int i=0;i<n1;i++){

scanf("%d",&hell[i]);

}

send(sockfd,hell,sizeof(hell),0);

n=recv(sockfd,&buffe,sizeof(buffe),0);

printf("Sorted array:-\n");

for(int i=0;i<5;i++){

printf("%d ",buffe[i]);

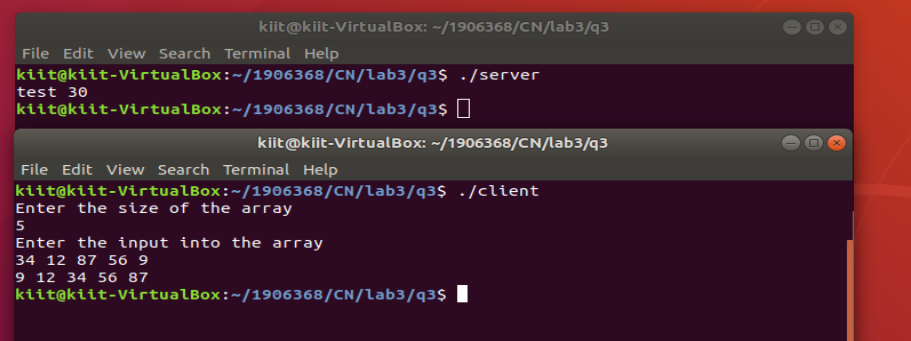
}

close(sockfd);

return 0;

}

**OUTPUT:**

****

1. **Write the client side and server side program for connection oriented socket (TCP) using C where both of them will exchange messages with each other. If any of them will receive the “exit” message from the other end then both of them will close the connection.**

**SERVER**

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd,confd;

char buffer[1024];

char hello[1024];

struct sockaddr\_in servaddr, cliaddr;

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

printf("Socket creation failed\n");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

memset(&cliaddr, 0, sizeof(cliaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(9991);

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

{

printf("bind failed");

exit(EXIT\_FAILURE);

}

int len, n;

len = sizeof(cliaddr);

listen(sockfd,1);

if((confd=accept(sockfd,(struct sockaddr\*)&cliaddr,&len))<0)

{

printf("Acceptance Failed\n");

exit(0);

}

while(1)

{ n = recv(confd,buffer,1024,0);

buffer[n]='\0';

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

{

printf("Ending chat\n");

break;

}

printf("Client : %s\n", buffer);

gets(hello);

send(confd,hello,1024,0);

}

close(sockfd);

close(confd);

return 0;

}

**CLIENT**

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#include<sys/socket.h>

#include<sys/types.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<netinet/in.h>

int main() {

int sockfd;

char hello[1024];

char buffer[1024];

printf("Start the chat:-\n");

struct sockaddr\_in servaddr;

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

printf("Socket Cretion Failed\n");

exit(EXIT\_FAILURE);

}

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(9991);

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

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

{

printf("Connect failed\n");

exit(0);

}

int n, len;

while(1)

{gets(hello);

send(sockfd,hello, 1024,0);

if(strcmp(hello,"exit")==0)

break;

n=recv(sockfd,buffer,1024,0);

buffer[n]='\0';

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

}

close(sockfd);

return 0;

}

**OUTPUT:**

****

**1906368**