**Exercise 2**

Design TCP iterative Client and Server application to reverse the given Input Sequence

Program:

The following program shows the contents of Filename : **TCP\_server.c**.

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <sys/wait.h>

#include <fcntl.h>

#include <unistd.h>

#define MYPORT 13600 /\*The port users will be connecting to\*/

void readstring(int,char \*);

int main(int C, char \*V[] )

{

int listensocket,connectionsocket,retbind;

struct sockaddr\_in

serveraddress,cliaddr;

socklen\_t len;

char buf[100],databuf[1024];

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

if (listensocket < 0 )

{

perror("socket" );

exit(1);

}

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

serveraddress.sin\_family = AF\_INET;

serveraddress.sin\_port = htons(MYPORT);/\*PORT NO\*/

serveraddress.sin\_addr.s\_addr =

htonl(INADDR\_ANY);/\*ADDRESS\*/

retbind=bind(listensocket,(struct sockaddr\*)&serveraddress,

sizeof(serveraddress));

/\*Check the return value of bind for error\*/

if(-1==retbind)

{

perror("BIND ERROR\n");

exit(1);

}

listen(listensocket,5);

/\*Beginning of the Main Server Processing Loop\*/

for (;;)

{

printf("Server:I am waiting-----Start of Main Loop\n");

len=sizeof(cliaddr);

connectionsocket=accept(listensocket,

(struct sockaddr\*)&cliaddr,&len);

if (connectionsocket < 0)

{

if (errno == EINTR)

printf("Interrupted system call ??");

continue;

}

printf("Connection from %s\n",

inet\_ntop(AF\_INET,&cliaddr.sin\_addr,buf,sizeof(buf)));

readstring(connectionsocket , databuf);

close(connectionsocket);

printf("Finished Serving One Client\n");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FUNCTION NAME:readstring

\* DESCRIPTION: Reads the string sent by the client over the

\* socket and stores it in the array fname .

\* NOTES : No Error Checking is done .

\* RETURNS :void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void readstring(

int connectionsocket, /\*Socket Descriptor\*/

char \*fname) /\*Array , to be populated by the string from client\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

{

int pointer=0,n;

int len=0,a,b;

char rev[50],temp[50],temp1[50];

int k,i;

while ((n=read(connectionsocket,(fname + pointer),1024))>0)

{

pointer=pointer+n;

}

fname[pointer]='\0';

printf("enter the string\n");

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

//strcpy(temp,fname);

k=strlen(fname);

// for(k=0;temp[k]!=0;k++);

// len=k;

a=0;

for(i=k-1;i>=0;i--)

temp[a++]=fname[i];

temp[a]='\0';

printf("\nrev is %s\n", temp);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**/\* client \*/**

The following program shows the contents of Filename : **TCP\_client.c**.

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <errno.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <fcntl.h>

#define MAXBUFFER 1024

void sendstring(int , char \*);

int main( int C, char \*V[] )

{

int sd,fd;

char c;

struct sockaddr\_in serveraddress;

char text[100];

int i=0;

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

if( sd < 0 ) {

perror( "socket" );

exit( 1 );

}

if (V[1] == NULL ) {

printf ("PL specfiy the server's IP Address \n");

exit(0);

}

if (V[2] == NULL ) {

printf ("PL specify the server's Port No \n");

exit(0);

}

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

serveraddress.sin\_family = AF\_INET;

serveraddress.sin\_port = htons(atoi(V[2]));//PORT NO

serveraddress.sin\_addr.s\_addr = inet\_addr(V[1]);//ADDRESS

if (connect(sd,(struct sockaddr\*)&serveraddress,

sizeof(serveraddress))<0)

{

printf("Cannot Connect to server");

exit(1);

}

printf("enter sentence to end enter #");

while(1)

{

c=getchar();

if(c=='#')

break;

text[i++]=c;

}

text[i]='\0';

sendstring(sd,text);

close(sd);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FUNCTION NAME:sendstring

\* DESCRIPTION: sends a string over the socket .

\* NOTES : No Error Checking is done .

\* RETURNS :void

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void sendstring(

int sd, /\*Socket Descriptor\*/

char \*fname) /\*Array Containing the string \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

{ int n , byteswritten=0 , written ;

char buffer[MAXBUFFER];

strcpy(buffer , fname);

n=strlen(buffer);

while (byteswritten<n)

{

written=write(sd , buffer+byteswritten,(n-byteswritten));

byteswritten+=written;

}

printf("String : %s sent to server \n",buffer);

}

**Execution & results:**

* 1. **1. Compiling and running server.**

[user@localhost week9]$ cc tcpserver.c

[user@localhost week9]$ mv a.out tcpserver

[user@localhost week9]$ ./tcpserver

Server:I am waiting-----Start of Main Loop

Connection from 127.0.0.1

enter the string

Server :Received Network Programming

Rev string is gnimmargorP krowteN

Finished Serving One Client

Server:I am waiting-----Start of Main Loop

* 1. **2. Compiling and running client.**

[user@localhost week9]$ cc tcpclient.c

[user@localhost week9]$ mv a.out tcpclient

[user@localhost week9]$./tcpclient 127.0.0.1 13153

enter sentence to end enter #Network Programming#

String : Network Programming sent to server