NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. Vth Sem

EXAM SET 4

DATE: 7th December, 2021

Subject Code: CS-311

Subject Name: Computer Network Laboratory

Submitted By:

Name	: Subhojit Ghimire
Sch. Id.	: 1912160
Branch	: CSE – B

1. **Write a program for connection-oriented service in which the server reverses the string sent by the client and sends it back.**

➔ **AIM:** TO IMPLEMENT CLIENT-SERVER PROTOCOL THAT REVERSES THE STRING SENT BY THE CLIENT USING CONNECTION-ORIENTED SERVICE (I.E., TCP) IN CPP.

CODE:

// SERVER SIDE IMPLEMENTATION CODE

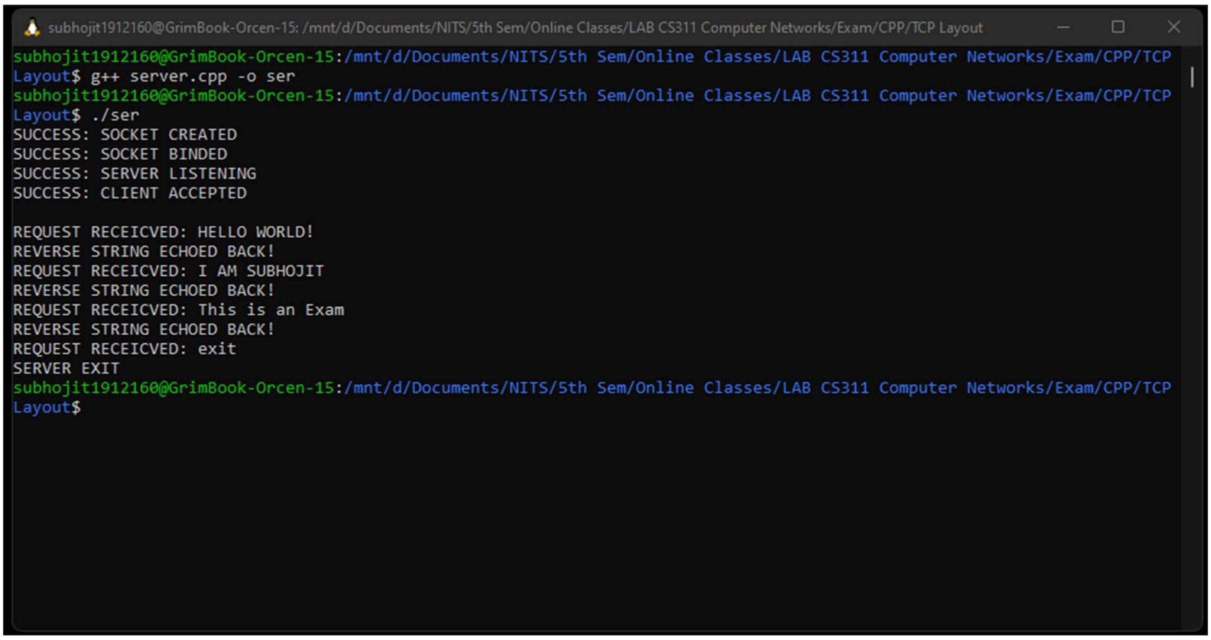
```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>

#define MAX 1024
#define PORT 8080

using namespace std;

int main () {
        int sockfd = socket (AF_INET, SOCK_STREAM, 0);
        struct sockaddr_in servaddr;
        if (sockfd != -1)
                cout <<"SUCCESS: SOCKET CREATED\n";
        else {
                perror ("ERROR: SOCKET CREATION");
                exit (EXIT_FAILURE);
        }
        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, (struct sockaddr*)&servaddr, sizeof (servaddr))))
                cout <<"SUCCESS: SOCKET BINDED\n";
        else {
                perror ("ERROR: SOCKET BINDING");
                exit (EXIT_FAILURE);
        }

        struct sockaddr_in cli;
        unsigned int len = sizeof (cli);
        if (!(listen (sockfd, 5)))
                cout <<"SUCCESS: SERVER LISTENING\n";
        else {
                perror ("ERROR: SERVER LISTEN");
                exit (EXIT_FAILURE);
        }

        int connfd = accept (sockfd, (struct sockaddr*)&cli, &len);
        if (connfd >= 0)
                cout <<"SUCCESS: CLIENT ACCEPTED\n\n";
        else {
                perror ("ERROR: CLIENT ACCEPTION");
                exit (EXIT_FAILURE);
        }

        int ii, jj;
        char buffer[MAX], buff[MAX];
        while (1) {
                bzero (buffer, MAX);
                bzero (buff, MAX);
                read (connfd, buffer, sizeof (buffer));
                strcpy (buff, buffer);
                buff [strlen (buff) - 1] = '\0';
                cout << "REQUEST RECEICVED: " << buff <<endl;
                if (!strncmp (buff, "exit", 4))
                        break;
                bzero (buffer, MAX);
                for (ii=0, jj=strlen(buff)-1; jj>=0; --jj, ++ii)
                        buffer [ii] = buff [jj];
                cout << "REVERSE STRING ECHOED BACK!" <<endl;
                buffer [ii] = '\0';
                write (connfd, buffer, sizeof (buffer));
        }
        cout <<"SERVER EXIT\n";
        close (sockfd);
        return 0;
}
```

```
// CLIENT SIDE IMPLEMENTATION CODE

#include <iostream>
#include <cstdlib>
#include <cstring>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <unistd.h>

#define MAX 1024
#define PORT 8080
```

```
using namespace std;

int main () {
        int sockfd = socket (AF_INET, SOCK_STREAM, 0);
        struct sockaddr_in servaddr;
        if (sockfd != -1)
                cout <<"SUCCESS: SOCKET CREATED\n";
        else {
                perror ("ERROR: SOCKET CREATION");
                exit (EXIT_FAILURE);
        }
        bzero (&servaddr, sizeof (servaddr));

        servaddr.sin_family = AF_INET;
        servaddr.sin_addr.s_addr = htonl (INADDR_ANY);
        servaddr.sin_port = htons (PORT);

        if (!(connect (sockfd, (struct sockaddr*)&servaddr, sizeof (servaddr))))
                cout <<"SUCCESS: CONNECTED TO SERVER\n\n";
        else {
                perror ("ERROR: SERVER CONNECTION");
                exit (EXIT_FAILURE);
        }

        cout<<"ENTER STRING TO REVERSE:";
        char buffer [MAX], buff [MAX];
        while (1) {
                bzero (buff, sizeof (buff));
                bzero (buffer, sizeof (buffer));
                cout <<"\n> ";
                fgets (buff, MAX, stdin);
                write (sockfd, buff, sizeof (buff));
                if (!strncmp (buff, "exit", 4))
                        break;
                read (sockfd, buffer, sizeof (buffer));
                cout <<"REVERSE IS: " << buffer <<endl;
        }
        cout <<"CLIENT EXIT\n";
        close (sockfd);
        return 0;
}
```

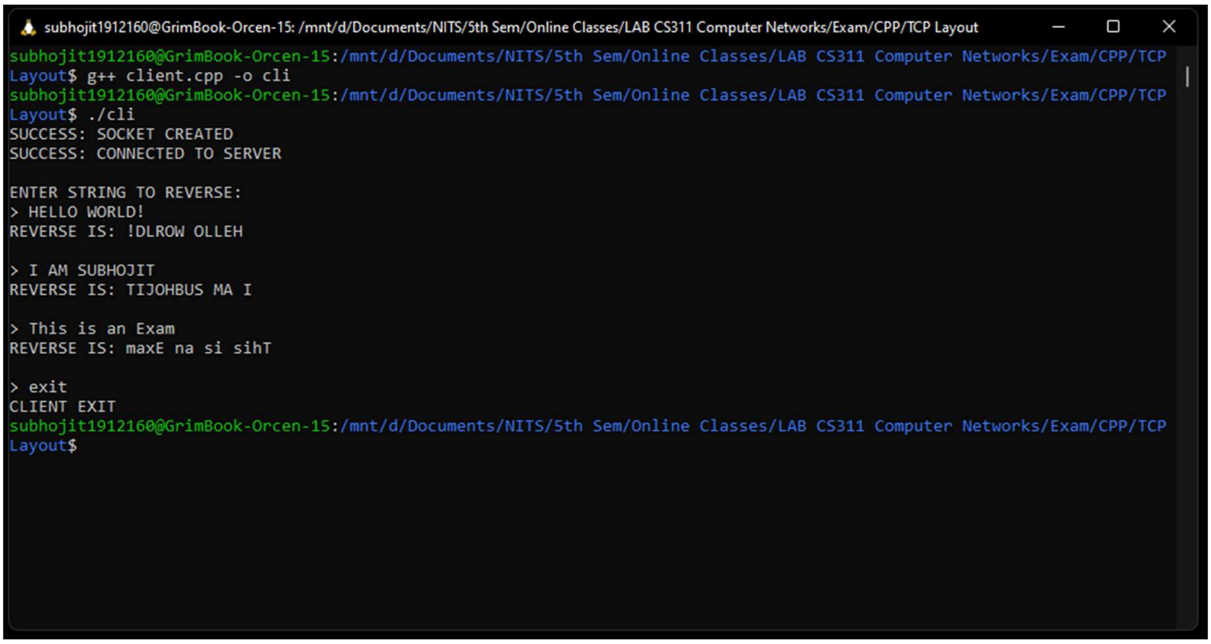
OUTPUT:

// SERVER


subhojit1912160@GrimBook-Orcen-15: /mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP Layout

```
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$ g++ server.cpp -o ser
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$ ./ser
SUCCESS: SOCKET CREATED
SUCCESS: SOCKET BINDED
SUCCESS: SERVER LISTENING
SUCCESS: CLIENT ACCEPTED

REQUEST RECEICVED: HELLO WORLD!
REVERSE STRING ECHOED BACK!
REQUEST RECEICVED: I AM SUBHOJIT
REVERSE STRING ECHOED BACK!
REQUEST RECEICVED: This is an Exam
REVERSE STRING ECHOED BACK!
REQUEST RECEICVED: exit
SERVER EXIT
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$
```

// CLIENT


subhojit1912160@GrimBook-Orcen-15: /mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP Layout

```
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$ g++ client.cpp -o cli
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$ ./cli
SUCCESS: SOCKET CREATED
SUCCESS: CONNECTED TO SERVER

ENTER STRING TO REVERSE:
> HELLO WORLD!
REVERSE IS: !DLROW OLLEH

> I AM SUBHOJIT
REVERSE IS: TIJOHBUS MA I

> This is an Exam
REVERSE IS: maxE na si sihT

> exit
CLIENT EXIT
subhojit1912160@GrimBook-Orcen-15:/mnt/d/Documents/NITS/5th Sem/Online Classes/LAB CS311 Computer Networks/Exam/CPP/TCP
Layout$
```

Output Explanation:

The client and server programs are compiled and executed. We will send the string from the client side. The server receives the message request and reverses the string and sends it back to the client. The client then accepts the echoed back message from server and displays it on the client side terminal. In the above output examples, I firstly sent the message “Hello World!” to the server. The server reversed the string to “!dlrow olleh” and sent it back to client to display on the client terminal. Similarly, the program works until the client sends the message “exit” which terminates both the server and the client.