THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Name: Manasi Bhagwat
Roll No: 50

Date: 05/04/2021

Lab Assignment No: 8

Aim: Socket Programming with C/Java – TCP Client, TCP Server.

Lab Outcome Attained: To implement client-server socket programs.

Theory:

Socket Programming

Sockets provide the communication mechanism between two computers using TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

When the connection is made, the server creates a socket object on its end of the communication. The client and the server can now communicate by writing to and reading from the socket. The java.net.Socket class represents a socket, and the java.net.ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

The following steps occur when establishing a TCP connection between two computers using sockets −

- The server instantiates a ServerSocket object, denoting which port number communication is to occur on.
- The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.
- After the server is waiting, a client instantiates a Socket object, specifying the server name and the port number to connect to.

- The constructor of the Socket class attempts to connect the client to the specified server and the port number. If communication is established, the client now has a Socket object capable of communicating with the server.
- On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

After the connections are established, communication can occur using I/O streams. Each socket has both an OutputStream and an InputStream. The client's OutputStream is connected to the server's InputStream, and the client's InputStream is connected to the server's OutputStream.

TCP is a two-way communication protocol, hence data can be sent across both streams at the same time.

Program :

TCP-SERVER –

Some packets are imported.

The java.io package is imported for providing a set of input and output streams used to read and write data to files or other input and output sources.

```
import java.io.*;
```

The java.net package is imported for network related programming.

```
import java.net.*;

class tcpserver1

{

public static void main(String argv[]) throws Exception

{

String clientSentence;

String str;

int count;
```

A server runs on a specific computer on the network and has a socket that is bound to a specific port number.

Here , we use the same computer as the client and the server is started on port number 6789 .

```
ServerSocket welcomeSocket = new ServerSocket(6789);

while(true)

{

count=0;
```

The server waits for the client to make a connection request. When the server code encounters the accept method, it blocks until a client makes a connection request to it.

```
Socket connectionSocket = welcomeSocket.accept();
```

This command creates an input stream attached to socket.

```
BufferedReader inFromClient = new

BufferedReader(new

inputStreamReader(connectionSocket.getInputStream()));
```

This command creates an output stream attached to socket.

```
DataOutputStream outToClient =

new DataOutputStream(connectionSocket.getOutputStream());
```

This command reads in line from socket.

```
clientSentence = inFromClient.readLine();

System.out.println("Received: " + clientSentence);
```

We count the number of vowels in the given string.

```
str = clientSentence.toLowerCase();

for(int i = 0; i < str.length(); i++)

{   if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' ||
str.charAt(i) == 'o' || str.charAt(i) == 'u')

            {count++;  }  }

String s=String.valueOf(count) + '\n';
```

This command writes out the final output to socket.

```
outToClient.writeBytes(s); }}}
```

TCP- CLIENT-

The java.io package is imported for providing a set of input and output streams used to read and write data to files or other input and output sources.

```
import java.io.*;
```

The java.net package is imported for network related programming.

```
import java.net.*;

class tcpclient1

{

public static void main(String arg[]) throws Exception

{

String sentence;

String noofvowels;
```

This command creates an input stream.

```
BufferedReader inFromUser =

new BufferedReader( new InputStreamReader(System.in));
```

This command creates a client Socket on the same computer and port number 6789.

```
Socket clientSocket = new Socket("localhost", 6789);
```

This command creates an output stream attached to socket.

```
DataOutputStream outToServer =

new DataOutputStream(clientSocket.getOutputStream());
```

This command creates an input stream attached to socket.

```
BufferedReader inFromServer = new

BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

sentence = inFromUser.readLine();
```

This command sends the input string to the server.

```
outToServer.writeBytes(sentence + '\n');
```

This command is used to receive the output line from server.

```
noofvowels = inFromServer.readLine();
```

The output is printed onto the screen.

```
System.out.println("NO OF VOWELS: " + noofvowels);
```

This command is used to close the connection with the server.

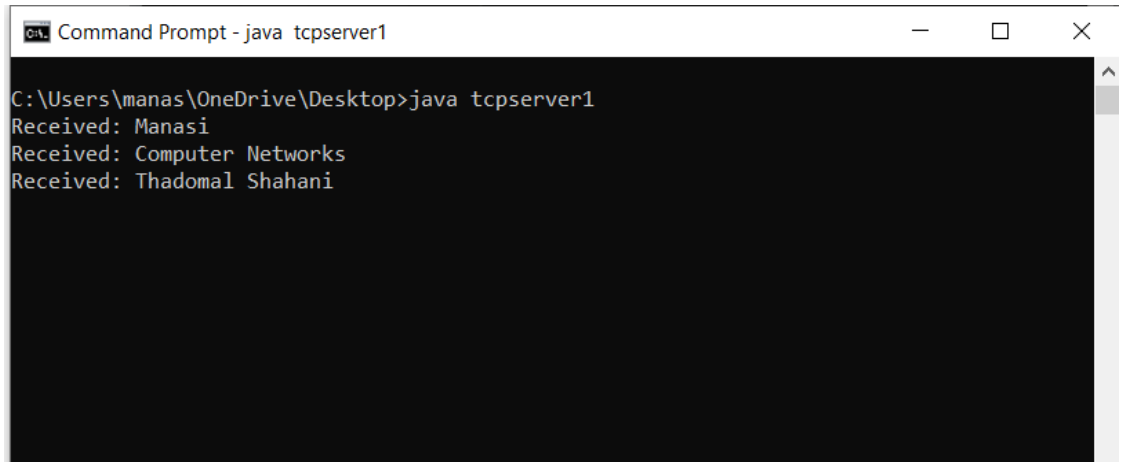
```
clientSocket.close();

}}
```

OUTPUT –

Client -

Server –



Conclusion

We have successfully implemented Socket Programming with Java – TCP Client, TCP Server.