COMPUTER NETWORKS LAB ASSESSMENT- 1

Suryakumar P 21MIS1146

Assessment -1

(a) Basic Network commands.

(b) Simple message transfer using TCP Socket Programming.

(c) Write a TCP socket program, the sum of random numbers generated by the server display in the client.

(d) Write a program to implement a chat server and client in Java using TCP sockets.

(e) Using TCP Sockets write a program to display the current date and time.

A. Basic Network Commands

Command	Description
Ping	**The ping command is a networking utility used to test the reachability of a host on an Internet Protocol (IP) network. It can be used to determine whether a specific IP address is accessible and also measures the round-trip time for messages sent from the local host to a remote host.**
NetStat	**NetStat is a networking utility that can be used to display all active network connections and their status. It can be used to identify which applications are using which ports and can be helpful in troubleshooting networking issues.**
NbtStat	**NbtStat is a networking utility in Windows that helps users troubleshoot NetBIOS over TCP/IP problems. It can be used to display a variety of information about the current state of the NetBIOS over TCP/IP protocol on both local and remote computers.**
Ip Config	**IP Config is a command-line tool that is used to display the current IP address configuration of a Windows machine. This includes the IP address, subnet mask, and default gateway.**
Hostname	**If you know which switch to use with the command, NbtStat can provide you with the hostname that has been assigned to a Windows device, as discussed earlier in this piece.**

ARP	**The Address Resolution Protocol, or ARP, is a networking utility used for mapping network addresses to physical addresses.**
Nslookup	**Nslookup is a command-line networking tool used for querying Domain Name System (DNS) to obtain domain name or IP address mapping, or other DNS records.**
Tracert	**The tracert command is a Command Prompt command that displays the network packets being sent and received, as well as the number of hops required for them to reach their destination.**
GetMac	**The MAC address is a unique identifier for every network-capable device on the internet. The number is assigned during the manufacturing process and is stored in the device's hardware.**
Route	**The Route networking command is one of the most essential networking commands for Windows administrators.**
NetDiag	**NetDiag is a networking diagnostic tool that is included with the Windows operating system.**

B. Simple message transfer using TCP Socket Programming.

Problem Definition:

To create a simple message transfer using TCP Socket Programming.

Method:

Socket Programming using JAVA.

1. Create a Socket Server.
2. Create a client program.
3. Connect them using IP Address and a Port.
4. Parse the message from client to the Server and print it.

Code:

```
//MyServer.java
import java.io.*;
import java.net.*;
public class MyServer {
public static void main(String[] args){
try{
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes connection
DataInputStream dis=new DataInputStream(s.getInputStream());
String  str=(String)dis.readUTF();
System.out.println("message= "+str);
ss.close();
}catch(Exception e){System.out.println(e);}
}
}
```

```
//MyClient.java
import java.io.*;
import java.net.*;
public class MyClient {
public static void main(String[] args) {
try{
Socket s=new Socket("localhost",6666);
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
dout.writeUTF("Hello Server, This is Suryakumar P 21MIS1146");
dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}
```

Output:


PROBLEMS
OUTPUT
TERMINAL
JUPYTER
DEBUG CONSOLE
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client> javac MyServer.java
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client> java MyServer
message= Hello Server, This is Suryakumar P 21MIS1146
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client>
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client> javac MyClient.java
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client> java MyClient
PS D:\VIT\Fall Inter Semester 23-24\SWE2002 - Computer Networks\Lab\Lab - 1 Server-Client>

C. Write a TCP socket program, the sum of random numbers generated by the server display in the client.

Problem Definition:

To write a TCP socket program, the sum of random numbers generated by the server display in the client.

Method:

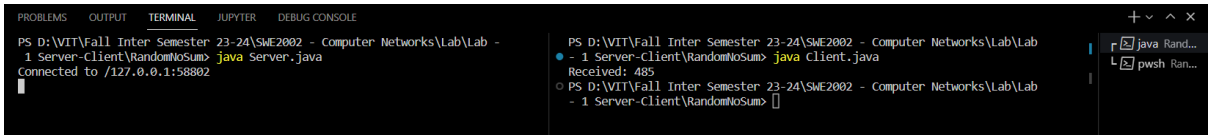
1. Creating sockets using Socket Programming using JAVA.
2. Using for loop generate random numbers multiple time.
3. Use a variable and add the generated numbers using the loop.
4. Send the final sum to the client via server.
5. Print the total sum in the client program.

Code:

```
//Server.java
import java.io.*;
import java.net.*;
import java.util.Random;

public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(12345);
        while (true) {
            Socket clientSocket = serverSocket.accept();
            System.out.println("Connected to " +
clientSocket.getRemoteSocketAddress());
            Random rand = new Random();
            int total = 0;
            for (int i = 0; i < 10; i++) {
                total += rand.nextInt(100) + 1;
            }
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
            out.println(total);
            clientSocket.close();
        }
    }
}
```

```java
//Client.java
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        Socket clientSocket = new Socket("localhost", 12345);
        BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        String data = in.readLine();
        System.out.println("Received: " + data);
        clientSocket.close();
    }
}
```

Output:



D. Write a program to implement a chat server and client in Java using TCP sockets.

Problem Definition:

To write a program to implement a chat server and client in Java using TCP sockets.

Method:

By using Socket Programming in JAVA

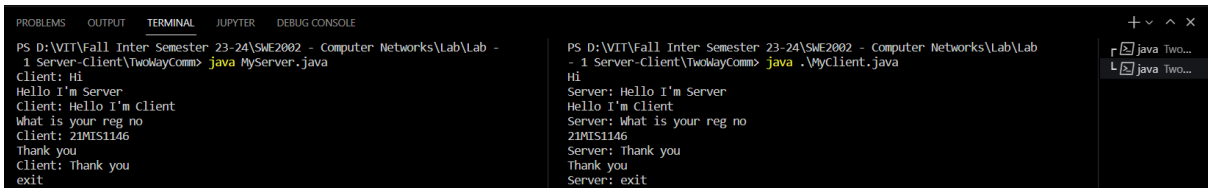
1. Create a Socket Server that iterates by sending messages again and again to the client.
2. Similarly in Client side iterate messages to the server.
3. The iteration stops when the entered message is “exit”.
4. And the two-way communication chat server is successfully implemented.

Code:

```
//MyServer.java
import java.net.*;
import java.io.*;
class MyServer{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333);
Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("exit")){
str=din.readUTF();
System.out.println("Client: "+str);
str2=br.readLine();
dout.writeUTF(str2);
dout.flush();
}
din.close();
s.close();
ss.close();
}}
```

```java
//MyClient.java
import java.net.*;
import java.io.*;
class MyClient{
public static void main(String args[])throws Exception{
Socket s=new Socket("localhost",3333);
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("exit")){
str=br.readLine();
dout.writeUTF(str);
dout.flush();
str2=din.readUTF();
System.out.println("Server: "+str2);
}

dout.close();
s.close();
}}
```

Output:



E. Using TCP Sockets write a program to display the current date and time.

Problem Definition:

To write a program to display the current date and time using TCP Sockets.

Method:

1. Create Server-Client using Java Socket Programming.
2. Import the *java.util.Date* package.
3. Now send the current date and time using the package from the Server to the Client and print it.
4. Date and time are displayed successfully using TCP Sockets.

Code:

```
//DateServer.java
import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

public class DateServer {
    public static void main(String[] args) throws IOException {
        try (ServerSocket listener = new ServerSocket(59090)) {
            while (true) {
                try (Socket socket = listener.accept()) {
                    PrintWriter out = new
PrintWriter(socket.getOutputStream(), true);
                    out.println(new Date().toString());
                }
            }
        }
    }
}
```

```java
//DateClient.java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;

public class DateClient {
    public static void main(String[] args) throws IOException {
        try (Socket socket = new Socket("localhost", 59090)) {
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            String response = in.readLine();
            System.out.println("Current date and time: " + response);
        }
    }
}
```

Output: