

**NAME :** Bhavik Ransubhe

**CLASS :** TE (B) COMP

**ROLL NO :** 39055

## **PROBLEM STATEMENT :**

Write a program using TCP socket for wired network for following (Use C/C++)

- a. Say Hello to Each other
- b. File transfer
- c. Calculator (Arithmetic)
- d. Calculator (Trigonometry)

Demonstrate the packets captured traces using Wireshark Packet Analyzer

Tool for peer to

peer mode.

---

### **1) Say Hello to Each other:-**

#### **CODE:**

##### **CLIENT SIDE:-**

```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);

        Socket s = null;           //client socket
        DataInputStream in = null;  //data input from socket
        DataOutputStream out = null; //data output for socket

        try {

            Socket socket = new Socket("localhost", 1401);
            in = new DataInputStream(socket.getInputStream());
            out = new DataOutputStream(socket.getOutputStream());

            System.out.println(in.readUTF());
            out.writeUTF("\n Hello from client");
            out.flush();

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
```

```

        if (in != null) in.close();
        if (out != null) out.close();
    }
}
}

```

## SERVER SIDE:-

```

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {

        Scanner sc=new Scanner(System.in);
        System.out.print("Server created");

        Socket s = null;           //client socket
        ServerSocket ss = null;     //server socket object
        DataInputStream in = null;  //data input from socket
        DataOutputStream out= null; //data output for socket

        try {

            ss=new ServerSocket(1401); //create serversocket with port number 8008
            s=ss.accept();
            in=new DataInputStream(s.getInputStream());
            out=new DataOutputStream(s.getOutputStream());

            out.writeUTF("Hi from server\n"); //send hi message to client
            out.flush();                      //flush all data to stream
            System.out.println(in.readUTF()); //read hi from client

        } catch (IOException e) {
            System.out.println(e);
        } finally {
            //close all allocated resource
            if(s!=null) s.close();
            if(ss!=null) ss.close();
            if(in!=null) in.close();
            if(out!=null) out.close();
        }
    }
}

```

## OUTPUT:

### CLIENT SIDE:-

Hi from server

### SERVER SIDE:-

Server created

Hello from client

---

## 2) FILE Transfer :-

### CODE:

#### CLIENT SIDE:-

```
import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.InetAddress;
import java.net.Socket;
public class Main {
    public static void main(String[] args) throws Exception{
        //Initialize socket
        Socket socket = new Socket(InetAddress.getByName("localhost"), 1401);
        byte[] contents = new byte[10000];
        //Initialize the FileOutputStream to the output file's full path.
        FileOutputStream fos = new FileOutputStream("d:\\file2.txt");
        BufferedOutputStream bos = new BufferedOutputStream(fos);
        InputStream is = socket.getInputStream();
        //No of bytes read in one read() call
        int bytesRead = 0;
        while((bytesRead=is.read(contents))!=-1)
            bos.write(contents, 0, bytesRead);
        bos.flush();
        socket.close();
        System.out.println("File saved successfully!");
    }
}
```

#### SERVER SIDE:-

```
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.OutputStream;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
public class Main
{
    public static void main(String[] args) throws Exception
    {
        //Initialize Sockets
        ServerSocket ssock = new ServerSocket(1401);
        Socket socket = ssock.accept();
        //The InetAddress specification
```

```

InetAddress IA = InetAddress.getByName("localhost");

//Specify the file
File file = new File("d:\\file1.txt");
FileInputStream fis = new FileInputStream(file);
BufferedInputStream bis = new BufferedInputStream(fis);
//Get socket's output stream
OutputStream os = socket.getOutputStream();
//Read File Contents into contents array
byte[] contents;
long fileLength = file.length();
long current = 0;
long start = System.nanoTime();
while(current!=fileLength){
    int size = 10000;
    if(fileLength - current >= size)
        current += size;
    else{
        size = (int)(fileLength - current);
        current = fileLength;
    }
    contents = new byte[size];
    bis.read(contents, 0, size);
    os.write(contents);
    System.out.print("Sending file ... "+(current*100)/fileLength+"% complete! \n");
}
os.flush();
//File transfer done. Close the socket connection!
socket.close();
sock.close();
System.out.println("File sent succesfully!");
}
}

```

## OUTPUT:

### CLIENT SIDE:-

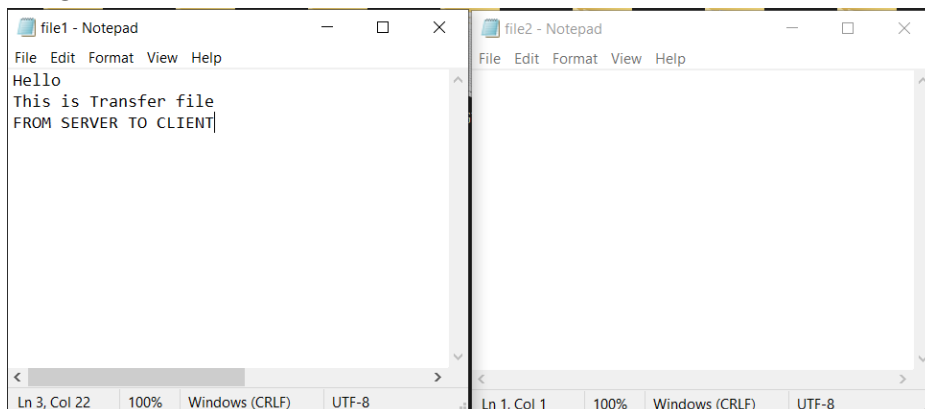
File saved successfully!

### SERVER SIDE:-

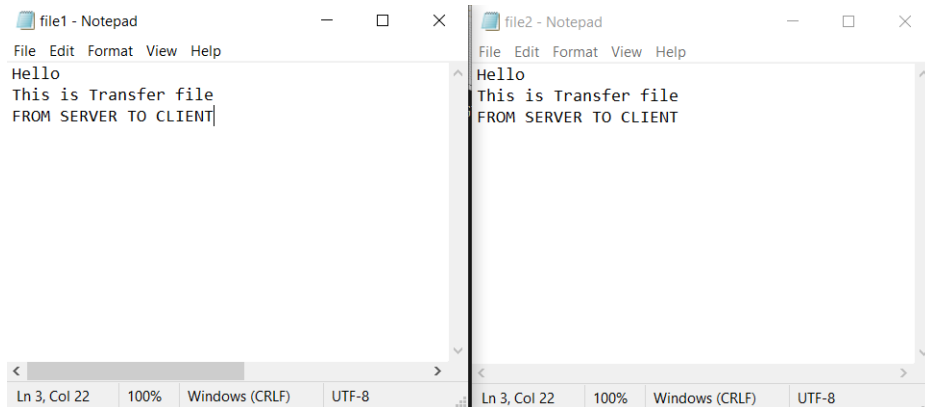
Sending file ... 100% complete!

File sent succesfully!

### BEFORE:



AFTER:



### 3) CALCULATOR (Arithmetic):-

CODE:

CLIENT SIDE:-

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) throws IOException
    {
        InetAddress ip = InetAddress.getLocalHost();
        int port = 4444;
        Scanner sc = new Scanner(System.in);

        // Step 1: Open the socket connection.
        Socket s = new Socket(ip, port);

        // Step 2: Communication-get the input and output stream
        DataInputStream dis = new DataInputStream(s.getInputStream());
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        while (true)
        {
            // Enter the equation in the form-
            // "operand1 operation operand2"
            System.out.print("Enter the equation in the form: ");
            System.out.println("operand operator operand");

            String inp = sc.nextLine();

            if (inp.equals("bye"))
                break;

            // send the equation to server
```

```

        dos.writeUTF(inp);

        // wait till request is processed and sent back to client
        String ans = dis.readUTF();
        System.out.println("Answer = " + ans);
    }
}
}

```

## SERVER SIDE:-

```

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.StringTokenizer;

public class Main
{
    public static void main(String args[]) throws IOException
    {

        // Step 1: Establish the socket connection.
        ServerSocket ss = new ServerSocket(4444);
        Socket s = ss.accept();

        // Step 2: Processing the request.
        DataInputStream dis = new DataInputStream(s.getInputStream());
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        while (true)
        {
            // wait for input
            String input = dis.readUTF();

            if(input.equals("bye"))
                break;

            System.out.println("Equation received: " + input);
            int result;

            // Use StringTokenizer to break the equation into operand and
            // operation
            StringTokenizer st = new StringTokenizer(input);

            int oprnd1 = Integer.parseInt(st.nextToken());
            String operation = st.nextToken();
            int oprnd2 = Integer.parseInt(st.nextToken());

            // perform the required operation.
            if (operation.equals("+"))
            {
                result = oprnd1 + oprnd2;
            }

            else if (operation.equals("-"))

```

```

    {
        result = oprnd1 - oprnd2;
    }
    else if (operation.equals("*"))
    {
        result = oprnd1 * oprnd2;
    }
    else
    {
        result = oprnd1 / oprnd2;
    }
    System.out.println("Sending the result...");

    // send the result back to the client.
    dos.writeUTF(Integer.toString(result));
}
}

```

## OUTPUT:

### CLIENT SIDE:-

Enter the equation in the form: 'operand operator operand'  
 50 - 27  
 Answer = 23  
 Enter the equation in the form: 'operand operator operand'  
 9 \* 3  
 Answer = 27  
 Enter the equation in the form: 'operand operator operand'  
 10 + 5  
 Answer = 15  
 Enter the equation in the form: 'operand operator operand'  
 10 / 3  
 Answer = 3  
 Enter the equation in the form: 'operand operator operand'  
 bye

Process finished with exit code 0

### SERVER SIDE:-

Equation received: 50 - 27  
 Sending the result...  
 Equation received: 9 \* 3  
 Sending the result...  
 Equation received: 10 + 5  
 Sending the result...  
 Equation received: 10 / 3  
 Sending the result...

Process finished with exit code 0

---

#### 4) CALCULATOR (Trigonometry):-

##### CODE:

##### CLIENT SIDE:-

```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);

        Socket s = null;           //client socket
        DataInputStream in = null;  //data input from socket
        DataOutputStream out = null; //data output for socket

        try {

            Socket socket = new Socket("localhost", 8008);
            in = new DataInputStream(socket.getInputStream());
            out = new DataOutputStream(socket.getOutputStream());

            while (true) {

                System.out.print("\nChoose Trigonometric operation :\n 1.sin\n 2.cos\n 3.tan\n 4.cot" +
                    "\n 5.sec\n 6.cosec\n 7.exit\n --->>>");
                int choice = sc.nextInt();

                if (choice < 7) {

                    System.out.print("\nEnter angle Degree:");
                    Double value = sc.nextDouble();
                    out.writeInt(choice);
                    out.writeDouble((Double) (value * 3.14 / 180)); //convert degree to radian
                    System.out.println("\nANS : " + in.readDouble()); //print ans from server

                } else {
                    out.writeInt('0'); //for end connection send y to server
                    sc.close();       //close all allocated resources
                    in.close();
                    out.close();
                    System.exit(0); //exit program
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {

            if (s != null) s.close();
            if (in != null) in.close();
            if (out != null) out.close();
        }
    }
}
```



## SERVER SIDE:-

```
import java.io.*;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {

        Scanner sc=new Scanner(System.in);
        System.out.print("Server created");

        Socket s = null;           //client socket
        ServerSocket ss = null;     //server socket object
        DataInputStream in = null;  //data input from socket
        DataOutputStream out= null; //data output for socket

        try {

            ss=new ServerSocket(8008); //create serversocket with port number 8008
            s=ss.accept();
            in=new DataInputStream(s.getInputStream());
            out=new DataOutputStream(s.getOutputStream());

            int choice=in.readInt();    //read operation choice from client
            while(choice!='0')
            {
                out.writeDouble(Calculation(choice,in.readDouble()));
                choice=in.readInt();
            }

        } catch (IOException e) {
            System.out.println(e);
        } finally {
            //close all allocated resource
            if(s!=null) s.close();
            if(ss!=null) ss.close();
            if(in!=null) in.close();
            if(out!=null) out.close();
        }

    }

    //trigonometric calculation
    static Double Calculation(int choice,Double value)
    {

        switch (choice)
        {
            case 1:
                System.out.print("\n Answer of sin value sent to client :");
```

```

        return Math.sin(value);
    case 2:
        System.out.print("\n Answer of cos value sent to client :");
        return Math.cos(value);

    case 3:
        System.out.print("\n Answer of tan value sent to client :");
        return Math.tan(value);
    case 4:
        System.out.print("\n Answer of cot value sent to client :");
        return 1/Math.tan(value);
    case 5:
        System.out.print("\n Answer of sec value sent to client :");
        return 1/Math.cos(value);

    case 6:
        System.out.print("\n Answer of cosec value sent to client :");
        return 1/Math.sin(value);
    }
    return -1.0;
}
}

```

## OUTPUT:

### CLIENT SIDE:-

Choose Trigonometric operation :

```

1.sin
2.cos
3.tan
4.cot
5.sec
6.cosec
7.exit
--->>>1

```

Enter angle Degree:60

ANS :0.8657598394923444

Choose Trigonometric operation :

```

1.sin
2.cos
3.tan
4.cot
5.sec
6.cosec
7.exit
--->>>3

```

Enter angle Degree:45

ANS :0.9992039901050427

Choose Trigonometric operation :

- 1.sin
  - 2.cos
  - 3.tan
  - 4.cot
  - 5.sec
  - 6.cosec
  - 7.exit
- >>>7

Process finished with exit code 0

### **SERVER SIDE:-**

Server created

Answer of sin value sent to client :

Answer of tan value sent to client :

Process finished with exit code 0

```
*****
*****
```