Lab Sheet 05

Distributed and Cloud Computing [ICT3253]

Submission By : W.M.M.M. Bandara (TG/2017/231)

Submitted To : Ms. Piyumi Wijerathne

Submission Date : 10.04.2021

Bachelor of Information and Communication Technology
Department of ICT
Faculty of Technology
University of Ruhuna

Question 01

01).

Java Socket programming is used for communication between the applications running on different JRE. Java Socket programming can be connection-oriented or connection-less. Socket and Server Socket classes are used for connection-oriented socket programming and Datagram Socket and Datagram Packet classes are used for connection-less socket programming. In socket programming there has two ways, there are client and server programming.

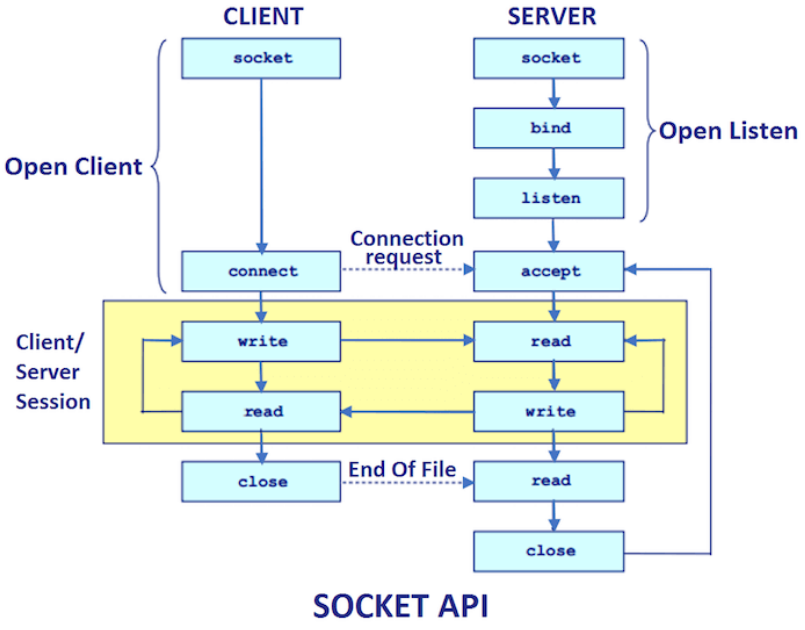
02).

- Stream sockets
- Datagram sockets
- Raw sockets
- Packet Sequenced sockets
- Hex sockets
- Socket Bit

03)

Server side

1. Create server socket on a particular port.
2. Listen to any attempt of connection to that port.
3. If a connection attempt succeeds, get the host of stream objects from the *Socket* and communicate with the client.
4. The communication, however, is established according to the agreed protocol.
5. Close the connection.



Client side

1. Connect to remote host.
2. Accept remote connections on the bounded port.
3. Bind to a port.
4. Listen to incoming data.
5. Send and receive data.
6. Close a connection

Question 02

Part 01

```java
package myclientservedemo;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.net.Socket;

/**
 *
 * @author ICT_Level3
 */
public class Client {
    public static void main(String[] args) {

        try {
            Socket s= new Socket("localhost",9000);
            DataOutputStream dataOutputStream = new DataOutputStream(s.getOutputStream());
            DataInputStream din = new DataInputStream(s.getInputStream());

            System.out.println("Client is Runing");
            dataOutputStream.writeUTF("Hello.....");
            String v = din.readUTF();
            System.out.println(v);
            dataOutputStream.flush();
            din.close();
            dataOutputStream.close();


            s.close();
        } catch (Exception e) {
            System.out.println("Eroor"+e.getMessage());
        }
    }
}
```

```java
package myclientservedemo;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author ICT_Level3
 */
public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket ss =   new ServerSocket(9000);
            Socket s = ss.accept();
            System.out.println("Server is Running.........");
            DataInputStream dataInputStream = new DataInputStream(s.getInputStream());
            DataOutputStream dataOutputStream = new DataOutputStream(s.getOutputStream());
            String value = dataInputStream.readUTF();
            dataOutputStream.writeUTF("byeeee");
            dataOutputStream.flush();
            dataOutputStream.close();
            System.out.println(value);
        } catch (IOException ex) {
            System.out.println("Error :"+ex.getMessage());
        }
    }
}
```

Part 02

```java
package myclientservedemo;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.net.Socket;

/**
 *
 * @author ICT_Level3
 */
public class SecondClient {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("localhost", 2000);
            File file = new File("C:\\Users\\ICT_Level3.LAB172.001\\Documents\\Original.txt");
            String value = "hiii server";
            byte []b = new byte[30];
            b= value.getBytes();
            FileOutputStream fout = new FileOutputStream(file);
            fout.write(b);
            fout.flush();
            fout.close();
            System.out.println("Transfer Complete");
        } catch (Exception e) {
        }
    }
}
```

```java
package myclientservedemo;

import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.net.ServerSocket;
import java.net.Socket;

/**
 *
 * @author ICT_Level3
 */
public class SecondServer {
    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(2000);
            Socket s = ss.accept();

            DataInputStream dis = new DataInputStream(s.getInputStream());
            byte []bv = new byte[30];
            System.out.println("File Transferred");

            FileInputStream fin = new FileInputStream("C:\\Users\\ICT_Level3.LAB172.001\\Documents\\Original.txt");
            FileOutputStream fos = new FileOutputStream("C:\\Users\\ICT_Level3.LAB172.001\\Documents\\Copy.txt");
            fin.read(bv);
            fos.write(bv);
            System.out.println("Successfully Cpoied");
            String data = new String(bv);
            System.out.println("file data : "+data);

        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```