

## **EX.NO 3(A) Applications using TCP sockets like Echo client and Echo server**

### **Aim**

To write a java program for applications using TCP sockets like Echo client and Echo server

### **Algorithm**

- 1.Start the program.
- 2.Get the frame size from the user
- 3.To create the framebased on the user request.
- 4.To send frames to server from the client side.
- 5.If your frames reach the server it will send ACK signal to client otherwise it will send NACK signal to client.
- 6.Stop the program

### **Program:**

#### **EchoServer.Java:**

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

public class EchoServer
{
    public EchoServer(int portnum)
    {
        try
        {
            server = new ServerSocket(portnum);
        }
        catch (Exception err)
        {
            System.out.println(err);
        }
    }

    public void serve()
    {
        try
```

```

        {
            while (true)
            {
                Socket client = server.accept();
                BufferedReader r = new BufferedReader(new
InputStreamReader(client.getInputStream()));
                PrintWriter w = new
PrintWriter(client.getOutputStream(),
true);
                w.println("Welcome to the Java EchoServer. Type 'bye'
to
close.");
                String line;
                do
                {
                    line = r.readLine();
                    if ( line != null )
                        w.println("Got: "+ line);
                    System.out.println (line);
                }
                while ( !line.trim().equals("bye") );
                client.close();
            }
        }
        catch (Exception err)
        {
            System.err.println(err);
        }
    }
    public static void main(String[] args)
    {

```

```

        EchoServer s = new EchoServer(9999);
        s.serve();
    }
    private ServerSocket server;
}

```

### **EchoClient.java :**

```

import java.io.*;
import java.net.*;
public class EchoClient
{
    public static void main(String[] args)
    {
        try
        {
            Socket s = new Socket("127.0.0.1", 9999);
            BufferedReader r = new BufferedReader(new
                InputStreamReader(s.getInputStream()));
            PrintWriter w = new PrintWriter(s.getOutputStream(), true);
            BufferedReader con = new BufferedReader(new
                InputStreamReader(System.in));

            String line;
            do
            {
                line = r.readLine();
                if ( line != null )
                    System.out.println(line);
                line = con.readLine();
                w.println(line);
            }
            while ( !line.trim().equals("bye") );
        }
    }
}

```

```

    }
    catch (Exception err)
    {
        System.err.println(err);
    }
}
}

```

### **Output:**

The screenshot shows two side-by-side Windows command prompt windows. The left window is titled 'C:\Windows\system32\cmd.exe - java EchoServer' and the right window is titled 'C:\Windows\system32\cmd.exe'. Both windows show the user navigating to the 'nwlab' directory and running 'javac EchoClient.java' and 'javac EchoServer.java'. The left window then runs 'java EchoServer', which outputs a welcome message and listens for input. The right window runs 'java EchoClient', which sends input to the server and receives the echoed output.

```

C:\Windows\system32\cmd.exe - java EchoServer
Microsoft Windows [Version 10.0.17763.615]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Karthik>
C:\Users\Karthik>cd nwlab
E:\nwlab>javac EchoClient.java
E:\nwlab>javac EchoServer.java
E:\nwlab>java EchoServer
Welcome
hello
Got: hello
welcome
Got: welcome
this is test for echo
Got: this is test for echo
tested
Got: tested
bye
E:\nwlab>

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.615]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Karthik>
C:\Users\Karthik>cd nwlab
E:\nwlab>java EchoClient
Welcome to the Java EchoServer. Type 'bye' to close.
hello
Got: hello
welcome
Got: welcome
this is test for echo
Got: this is test for echo
tested
Got: tested
bye
E:\nwlab>

```

### **Viva questions:**

1. Define server and what are the types of server?
2. What are the three types of socket function?
3. What are concurrent servers?
4. Define Iterative server
5. Compare Iterative server and concurrent server
6. Explain socket address structure
7. List some character stream support classes
8. What do you mean by socket programming?

### **Result:**

Thus the java program to concurrently communicate using TCP Sockets was executed successfully