|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SCSA2501** | **COMPUTER NETWORKS LAB** | **L** | **T** | **P** | **Credits** | **Total Marks** |
| **0** | **0** | **4** | **2** | **100** |

**SUGGESTED LIST OF EXPERIMENTS**

1.        To find IP address of a machine.

2.        Time and Date server.

3.        Echo UDP server.

4.        TCP sockets.

a.        Echo client & server

b.        Chat

c.        File transfer

5.        Daemon program.

6.        Implementing stop and wait protocol & sliding window protocol.

7.        Code simulating ARP/RARP protocols.

8.        Code simulating PING and TRACEROUTE commands.

9.        RPC (Remote Procedure Call).

10.      Study of Network Simulator (NS) and simulation of Congestion Control algorithms using NS.

11.      Case study

a.        Flooding

b.        Link state routing

c.        Distance vector routing

**Ex.No:1**

1. A simple program to print the IP address of the system

Algorithm

# Import all the required packages net, and io packages

1. Create a class ipclient

# Initialize InetAddress class and create the object ia.

1. Get the Ip address of the system using getLocalHost() method

# Print the IP address.

**PROGRAM**

import java.net.\*; import java.io.\*; class ipclient

{

public static void main(String args[])

{

try

{

InetAddress ia=InetAddress.getLocalHost(); System.out.println("The client system address is: "+ia);

}

catch(IOException e)

{

System.out.println("The exception is: "+e);

}

}

**Ex.No 1A To Print Client Address at the Server Side**

Aim : Design a socket program to print the client address at the server side **Algorithm**

Server:

1. Import all the required io and net packages
2. Create a class Sip
3. Initialize the classes ServerSocket and Socket and their respective objects.
4. Initialize the class DataInputStream for reading the data from socket through getInputStream() method.
5. Establish the connection with client system using the client IP and port number 8020
6. Read the IP address from the socket
7. Print the IP address at the server Client:
8. Import all the required io and net packages
9. Create a class Cip
10. Initialize the classes Socket and their respective objects.
11. Initialize the class PrintStream Class for writing data into the socket through getOutputStream() method.
12. Get the IP address of the system using InetAddress class with getLocalHost() method.
13. Establish the connection with server system using the IP address and port number 8020
14. Write the IP address into the socket
15. If any exception arise, print the error.

Program:

**Server**

import java.io.\*; import java.net.\*; class Sip

{

public static void main(String args[])

{

ServerSocket ss; Socket s; DataInputStream dis; String ip;

try

{

ss=new ServerSocket(8020); while(true)

{

s=ss.accept();

dis=new DataInputStream(s.getInputStream()); ip=dis.readLine();

System.out.println("Ip address of the client system is"+ip);

}

}

catch(IOException e)

{

System.out.println("The exception is: "+e);

}

}

}

**Client**

Import java.io.\*; import java.net.\*; class Cip

{

public static void main(String args[])

{

Socket soc;

PrintStream s;

try

{

InetAddress ia=InetAddress.getLocalHost(); soc=new Socket(ia,8020);

ps=new PrintStream(soc.getOutputStream()); ps.println(ia);

}

catch(IOException e)

{

System.out.println("The exception is: "+e);

}

}

}

**Ex.No:2**

# **2. CREATION OF DATE AND TIME SERVER.**

**SERVER PROGRAM**

import java.io.\*; import java.net.\*; import java.util.\*; class dateserver

{

public static void main(String args[])

{

ServerSocket ss; Socket s; PrintStream ps; DataInputStream dis; String inet;

try

{

ss=new ServerSocket(8020); while(true)

{

s=ss.accept();

ps=new PrintStream(s.getOutputStream()); Date d=new Date();

ps.println(d); ps.close();

}

}

catch(IOException e)

{

System.out.println("The exception is: "+e);

} } }

# CLIENT PROGRAM

import java.io.\*; import java.net.\*; class dateclient

{

public static void main(String args[])

{

Socket soc;

DataInputStream dis;

String sdate;

PrintStream ps;

try

{

InetAddress ia=InetAddress.getLocalHost(); soc=new Socket(ia,8020);

dis=new DataInputStream(soc.getInputStream()); sdate=dis.readLine();

System.out.println("The data in the server is: "+sdate);

}

catch(IOException e)

{

System.out.println("The exception is: "+e);

}

}

}

**Ex.No:3 Echo UDP server.**

**Echo UDP server**

**/\*UDP SERVER\*/**

import java.net.\*;

import java.io.\*;

public class udpserver

{

public static int client=789;

public static int server=790;

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

{

String s;

InetAddress id=InetAddress.getLocalHost();

BufferedReader dis=new BufferedReader(new InputStreamReader(System.in));

DatagramSocket ds=new DatagramSocket(server);

byte b[]=new byte[1024];

System.out.println("Server Side.... Sending....");

System.out.println("\n"+id);

while(true)

{

s=dis.readLine();

if(s.equals("end"))

{

b=s.getBytes();

DatagramPacket dp=new DatagramPacket(b,s.length(),id,client);

ds.send(dp);

break;

}

else

{

b=s.getBytes();

DatagramPacket dp=new DatagramPacket(b,s.length(),id,client);

ds.send(dp);

}

}

}

}

**/\*UDPCLIENT\*/**

import java.net.\*;

import java.io.\*;

public class udpclient

{

public static int client=789;

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

{

DatagramSocket ds=new DatagramSocket(client);

byte b[]=new byte[1024];

System.out.println("client....receiving....");

while(true)

{

DatagramPacket dp=new DatagramPacket(b,b.length);

ds.receive(dp);

String s=new String(dp.getData(),0,dp.getLength());

if(s.equals("end")) break;

else System.out.println(s);

}

} }

**Ex.No:4 a) TCP SOCKET-ECHO Client & 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

**Server**

import java.io.\*;

import java.net.\*;

class echos {

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

{

String echoin;

ServerSocket svrsoc;

Socket soc;

BufferedReader br;

try

{

svrsoc = new ServerSocket(2000);

soc = svrsoc.accept();

br = new BufferedReader (new InputStreamReader(soc.getInputStream()));

PrintStream ps = new PrintStream(soc.getOutputStream());

System.out.println("Connected for echo:");

while((echoin=br.readLine())!=null)

{

if(echoin.equals("end"))

{

System.out.println("Client disconnected");

br.close();

break;

}

else

ps.println(echoin);

}

}

catch(UnknownHostException e)

{

System.out.println(e.toString());

}

catch(IOException ioe)

{

System.out.println(ioe);

}

}

}

Client:

import java.io.\*;

import java.net.\*;

class echoc {

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

{

String sockin;

try

{

Socket csock = new Socket(InetAddress.getLocalHost(),2000);

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

BufferedReader br\_sock = new BufferedReader(new InputStreamReader(csock.getInputStream()));

PrintStream ps = new PrintStream(csock.getOutputStream());

System.out.println("Start echoing... type 'end' to terminate");

while((sockin=br.readLine())!=null)

{

ps.println(sockin);

if(sockin.equals("end"))

break;

else

System.out.println("echoed from server:"+br\_sock.readLine());

}

}

catch(UnknownHostException e)

{

System.out.println(e.toString());

}

catch(IOException ioe)

{

System.out.println(ioe);

}

}

}

**Ex.No:4 b) TCP SOCKET-Simple Chat Program**

**Client program**

//cchatserver.java

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class cchatserver extends Thread

{

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

{

ServerSocket ssocket=new ServerSocket(4000);

Socket csocket=ssocket.accept();

BufferedReader br=new BufferedReader(new InputStreamReader(csocket.getInputStream()));

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

PrintWriter out=new PrintWriter(csocket.getOutputStream(),true);

String s,t;

try

{

while(true)

{

System.out.println("server");

s=in.readLine();

out.println("server:"+s);

System.out.println(br.readLine());

}

}

catch(IOException e)

{

System.out.println("client has closed");

}

}

}

\\cchatclient.java

import java.net.\*;

import java.io.\*;

public class cchatclient

{

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

{

Socket s=new Socket(InetAddress.getLocalHost(),4000);

BufferedReader br=new BufferedReader(new InputStreamReader(s.getInputStream()));

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

PrintWriter out=new PrintWriter(s.getOutputStream(),true);

String input,t;

while(true)

{

System.out.println("client");

out.println("client:"+in.readLine());

System.out.println(br.readLine());

}

}

}

**Ex.No:4 c) TCP SOCKET-File Transfer Protocol**

**ftpclient.java**

import java.io.\*;

import java.net.\*;

public class ftpclient

{

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

{

Socket s= new Socket(InetAddress.getLocalHost(),4000);

BufferedReader in=new BufferedReader(new InputStreamReader(s.getInputStream()));

String tmp;

try

{

while((tmp=in.readLine())!=null)

System.out.println(tmp);

}

catch(Exception e)

{}

}

}

[**\\ftpserver.java**](file:///\\ftpserver.java)

import java.io.\*;

import java.net.\*;

public class ftpserver extends Thread

{

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

{

ServerSocket ss=new ServerSocket(4000);

Socket s=ss.accept();

String f1,data="",tmp="";

try

{

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

System.out.println("Enter the file name to open");

f1=in.readLine();

PrintWriter out=new PrintWriter(s.getOutputStream(),true);

File f=new File(f1);

if(f.isFile()&&f.canRead())

{

BufferedReader br=new BufferedReader(new FileReader(f1));

while((tmp=br.readLine())!=null)

data=data+tmp+"\n";

}

else

data="error in input file";

out.println(data);

}

catch(Exception e)

{}

}

}

**Ex.No:5 Daemon program**

public class DaemonThread extends Thread

{

public DaemonThread(String name){ super(name);

}

public void run()

{

// Checking whether the thread is Daemon or not

if(Thread.currentThread().isDaemon())

{

}

else

{

}

}

Public static void main(String[] args)

{

DaemonThread t1 = new DaemonThread("t1"); DaemonThread t2 = new DaemonThread("t2"); DaemonThread t3 = new DaemonThread("t3");

// Setting user thread t1 to Daemon t1.setDaemon(true);

// starting first 2 threads t1.start();

t2.start();

// Setting user thread t3 to Daemon t3.setDaemon(true);

t3.start();

}

}

Output:

t1 is Daemon thread t3 is Daemon thread t2 is User thread

**Ex.No:6 Code simulating PING and TRACEROUTE commands.**

**Trace Route Program**

import java.io.BufferedReader; import java.io.InputStreamReader;

public class traceroutecmd

{

public static void runSystemCommand(String command)

{

try

{

Process p = Runtime.getRuntime().exec(command); BufferedReader inputStream = new BufferedReader(

new InputStreamReader(p.getInputStream()));

String s = "";

while ((s = inputStream.readLine()) != null) System.out.println(s);

}

catch (Exception e)

{

}

}

public static void main(String[] args)

{

// String ip = "[www.google.co.in](http://www.google.co.in/)";

// String ip = "127.0.0.1";

String ip = "[www.drranurekha.com](http://www.drranurekha.com/)"; runSystemCommand("tracert " + ip);

}

}

Ping

Program:

import java.io.\*; public class ping1

{

public static void runSystemCommand(String Command)

{

try{

Process p=Runtime.getRuntime().exec(Command);

BufferedReader InputStream=new BufferedReader(new InputStreamReader(p.getInputStream())); String s="vvv";

while((s=InputStream.readLine())!=null)

{

System.out.println(s);

}

}

catch(Exception e)

{

e.printStackTrace();

}

}

public static void main(String[]args)

{

String Ip="localhost";

runSystemCommand("ping " +Ip); java.util.Date date=new java.util.Date(); System.out.println(date);

}

}

Ex.No:7 Program to implement HTTP protocol

**Aim:** Program to implement HTTP protocol and to print URl for the Client.

# Algorithm:

STEP 1: Create the URL with Http URL Connections

STEP 2: Define the Http Protocol for Client Connections.

STEP3: Get the Http Connection. STEP4:Print the URL for the Client.

# Program:

import java.io.\*; import java.net.\*; public class myhttp

{

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

{

URL url=new URL("<http://www.sathyabama.ac.in/>"); URLConnection conn=url.openConnection(); conn.connect();

InputStreamReader content= new InputStreamReader(conn.getInputStream()); FileWriter f=new FileWriter ("abc.html");

for(int i=0;i!=-1;i= content.read())

{

f.write((char) i);

}

}

}