EXP:7 STUDY OF SOCKET PROGRAMMING AND CLIENT SERVER MODEL USING TCP AND UDP

Objective:

To implement socket programming date and time display from client to server using TCP and UDP Sockets.

Learning Outcomes:

After the completion of this experiment, student will be able to

 Write, execute and debug programs which use TCP Socket API.

 understand the use of client/server architecture in application development

 Develop a client-server application by using TCP and UDP

 Understand how the connection is established using a socket between a client and a server and also understands time and date retrieval from the server.

Problem Statement:

 A server program to establish the socket connection with the client.

 A client program which on establishing a connection retrieves the time and date of the system and displays it.

Concept:

Socket Programming

 Sockets provide the communication mechanism between two computers. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.

 When the connection is made, the server creates a socket object on its end of the communication. The client and server can now communicate by writing to and reading from the socket.

Client–server model

 The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

 Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system.

 A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function.

 Clients therefore initiate communication sessions with servers which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

System and Software tools required:

Package Required : Java Compiler

Operating System : UBUNTU

Minimum Requirement : Pentium III or Pentium IV with 2GB RAM 40 GB hard disk

Algorithm: Server:

Step1: Create a server socket and bind it to port.

Step 2: Listen for new connection and when a connection arrives, accept it.

Step 3: Send server's date and time to the client.

Step 4: Read client's IP address sent by the client.

Step 5: Display the client details.

Step 6: Repeat steps 2-5 until the server is terminated.

Step 7: Close the server socket.

Client

Step1: Create a client socket and connect it to the server's port number.

Step2: Retrieve its own IP address using built-in function.

Step3: Send its address to the server.

Step4: Display the date & time sent by the server.

Step5: Close the client socket

Execution of program:

Compiling the program: javac file name.java

Executing the program : java file name

Sample coding: TCP Program: dateserver.java

//import java packages

import java.net.*; import java.io.*; importjava.util.*;

/*… Register service on port 8020…*/

ss=new ServerSocket(8020);

/*… Wait and accept a connection…*/

s=ss.accept();

/*… Get a communication stream associated with the socket…*/

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

/* …To get system time…*/

Date d=new Date();

ps.println(d);

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

inet=dis.readLine();System.out.println("THE CLIENT SYSTEM ADDRESS IS :"+inet);

/* …This method is used to request for closing or terminating an object…*/

ps.close();}}

dateclient.java

/* …Socket class is having a constructor through this Client program can request to server to get connection…*/

Socket soc;

DataInputStream dis;

String sdate;

PrintStreamps;

/*…getLocalHost() method: Returns the name of the local computer…*/

InetAddressia=InetAddress.getLocalHost();

/*… Open your connection to a server, at port 8020…*/

soc=new Socket(ia,8020);

/*… Get an input file handle from the socket and read the input…*/

/*…getInputStream()-This method take the permission to write the data from client program to server program and server program to client program…*/

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

sdate=dis.readLine();

System.out.println("THE date in the server is:"+sdate);

/* …getOutputStream()-This method is used to take the permission to read data from client system by the server or from the server system by the client…*/

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

ps.println(ia);}