Experiment No: 5

// Aim : Implement token ring based mutual exclusion algorithm.

//Step 1 – Write and Compile Server program

import java.io.\*; import

java.net.\*;

public class MutualServer implements Runnable

{

Socket socket=null; static

ServerSocket ss;

MutualServer(Socket newSocket)

{

this.socket=newSocket;

}

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

{

ss=new ServerSocket(7000);

System.out.println("Server Started");

while(true)

{

Socket s = ss.accept();

MutualServer es = new MutualServer(s); Thread

t = new Thread(es);

t.start();

}

}

public void run()

{

try

{BufferedReader

in

=

new

BufferedReader(new

InputStreamReader(socket.getInputStream()));

while(true)

{

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

}

}

catch(Exception e){ }

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Step 2 – Write and Compile First client program ClientOne.java.

import java.io.\*;

import java.net.\*; public

class ClientOne

{

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

{

Socket s=new Socket("localhost",7000);

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

ServerSocket ss = new ServerSocket(7001);

Socket s1 = ss.accept();

BufferedReader in1 = new BufferedReader(new

InputStreamReader(s1.getInputStream()));

PrintStream out1 = new PrintStream(s1.getOutputStream()); BufferedReader br = new

BufferedReader(new InputStreamReader(System.in));

String str="Token";

while(true)

{

if(str.equalsIgnoreCase("Token"))

{

System.out.println("Do you want to send some data");

System.out.println("Enter Yes or No"); str=br.readLine();

if(str.equalsIgnoreCase("Yes"))

{System.out.println("Enter the data");

str=br.readLine();

out.println(str);

}

out1.println("Token");

}

System.out.println("Waiting for Token");

str=in1.readLine();

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*88

//Step 3 – Write and Compile Second client program

import java.io.\*; import

java.net.\*; public class

ClientTwo

{

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

{

Socket s=new Socket("localhost",7000);

PrintStream out = new PrintStream(s.getOutputStream()); Socket

s2=new Socket("localhost",7001); BufferedReader in2 = new

BufferedReader(new InputStreamReader(s2.getInputStream()));

PrintStream out2 = new PrintStream(s2.getOutputStream()); BufferedReader br = new

BufferedReader(new InputStreamReader(System.in));

String str;

while(true)

{

System.out.println("Waiting for Token");

str=in2.readLine();

if(str.equalsIgnoreCase("Token"))

{

System.out.println("Do you want to send some data");

System.out.println("Enter Yes or No"); str=br.readLine();

if(str.equalsIgnoreCase("Yes")){

System.out.println("Enter the data"); str=br.readLine();

out.println(str);

}

out2.println("Token");

}

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Step 4 – Run Server Program and keep it running till we connect the clients.

Step 5 – Open new Command prompt and Run ClientOne Program on it and keep it running till ClientTwo starts.

Step 6 – Open one more Command prompt to Run ClientTwo Program. The output allows both the

clients to use token and share their messages with each other using Token Ring. To send the message,

the client has to accept the token by typing type Yes followed by the message alternately and has to

type No to release the token.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

c:\javac MutualServer.java

c:\javac ClientOne.java

c:\javac ClientTwo.java

Step 4 – Run Server Program and keep it running till we connect the clients.

c:\java MutualServer

Step 5 – Open new Command prompt and Run ClientOne Program on it and keep it running till ClientTwo starts.

c:\java ClientOne

Step 6 – Open one more Command prompt to Run ClientTwo Program. The output allows both the

clients to use token and share their messages with each other using Token Ring. To send the message,

c:\java ClientTwo

Output:





