**PRACTICAL NO.5**

**Aim:** Implementation of mutual exclusion using Token Ring Technique.

**TokenRing.java**

import java.net.\*;

import java.io.\*;

public class TokenRing {

public static DatagramSocket ds;

public static DatagramPacket dp;

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

try {

ds=new DatagramSocket(1000); }

catch(Exception e) {e.printStackTrace();}

while(true) {

byte buff[]=new byte[1024];

ds.receive(dp=new DatagramPacket(buff, buff.length));

String str=new String(dp.getData(),0,dp.getLength()); System.out.println("Message from "+str); }}}

**TokenRingClient1.java**

import java.net.\*;

import java.io.\*;

public class TokenRingClient1 {

public static DatagramSocket ds;

public static DatagramPacket dp;

public static BufferedReader br;

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

boolean hasToken;

try {

ds=new DatagramSocket(100); }

catch(Exceptione) {e.printStackTrace();}

hasToken=true;

while(true) {

if(hasToken==true) {

System.out.println("Do you want to enter data? (yes/no): ");

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

String ans=br.readLine();

if(ans.equalsIgnoreCase("yes")) {

System.out.println("Ready to send.");

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

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

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

String str="Client-1===> "+br.readLine();

byte buff[]=new byte[1024];

buff=str.getBytes();

ds.send(new DatagramPacket (buff, buff.length,InetAddress.getLocalHost(), 1000));

System.out.println("Now sending..."); }

else if(ans.equalsIgnoreCase("no")) {

System.out.println("I am busy.");

//Sending message to client 2

String msg="Token";

byte buff1[]=new byte[1024];

buff1=msg.getBytes();

ds.send(new DatagramPacket(buff1, buff1.length, InetAddress.getLocalHost(),200));

hasToken=false;

//receiving message from Client 2

byte buff2[]=new byte[1024];

ds.receive(dp=new DatagramPacket(buff2, buff2.length));

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

System.out.println("The data is: "+clientmsg);

if(clientmsg.equals("Token"))

hasToken=true;

System.out.println("I am leaving busy state.");

} }

else {

System.out.println("Enter in receive mode.");

byte buff[]=new byte[1024];

ds.receive(dp=new DatagramPacket(buff, buff.length));

String clientmsg1=new String(dp.getData(),0,dp.getLength());

System.out.println("The data is: "+clientmsg1);

if(clientmsg1.equals("Token")) {

hasToken=true; } } } } }

**TokenRingClient2.java**

import java.net.\*;

import java.io.\*;

public class TokenRingClient2 {

public static DatagramSocket ds;

public static DatagramPacket dp;

public static BufferedReader br;

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

boolean hasToken;

try {

ds=new DatagramSocket(200); }

catch(Exception e) {e.printStackTrace();}

hasToken=false;

while(true) {

if(hasToken==true) {

System.out.println("Do you want to enter data? (yes/no): ");

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

String ans=br.readLine();

if(ans.equalsIgnoreCase("yes")) {

System.out.println("Ready to send.");

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

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

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

String str="Client-2===> "+br.readLine();

byte buff1[]=new byte[1024];

buff1=str.getBytes();

ds.send(new DatagramPacket(buff1, buff1.length, InetAddress.getLocalHost(),1000));

System.out.println("Data sent."); }

else {

//Sending message to client 1

String clientmsg="Token";

byte buff2[]=new byte[1024];

buff2=clientmsg.getBytes();

ds.send(new DatagramPacket(buff2, buff2.length, InetAddress.getLocalHost(),100));

hasToken=false; } }

else {

try {

byte buff[]=new byte[1024];

System.out.println("Enter in receive mode.");

ds.receive(dp=new DatagramPacket(buff, buff.length));

String clientmsg1=new String(dp.getData(),0,dp.getLength());

System.out.println("The data is: "+clientmsg1);

if(clientmsg1.equals("Token"))

hasToken=true; }

catch(Exception e){e.printStackTrace();}}}}}