**Mutual Exclusion Using Token Ring Algorithm**

**SOURCE CODE:**

**TokenServer.java**

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

public class TokenServer  
{  
    public static void main(String agrs[])throws Exception  
        {  
             
            while(true)  
            {  
            Server sr=new Server();  
            sr.recPort(8000);  
            sr.recData();  
            }  
        }  
}

class Server  
{  
     
    boolean hasToken=false;  
    boolean sendData=false;  
    int recport;  
     
    void recPort(int recport)  
    {  
        this.recport=recport;  
    }

    void recData()throws Exception  
    {  
        byte buff[]=new byte[256];  
        DatagramSocket ds;  
        DatagramPacket dp;  
        String str;  
         
        ds=new DatagramSocket(recport);  
        dp=new DatagramPacket(buff,buff.length);  
        ds.receive(dp);  
        ds.close();  
         
        str=new String(dp.getData(),0,dp.getLength());  
        System.out.println(“The message is “+str);  
    }  
}

**TokenClient1.java**

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

public class TokenClient1  
{  
    public static void main(String arg[]) throws Exception  
        {  
            InetAddress lclhost;  
            BufferedReader br;  
            String str=””;  
            TokenClient12 tkcl,tkser;  
            boolean hasToken;  
            boolean setSendData;  
             
            while(true)  
            {  
            lclhost=InetAddress.getLocalHost();  
            tkcl = new TokenClient12(lclhost);  
            tkser = new TokenClient12(lclhost);         
            //tkcl.setSendPort(9001);  
            tkcl.setSendPort(9004);  
            tkcl.setRecPort(8002);  
            lclhost=InetAddress.getLocalHost();  
            tkser.setSendPort(9000);

            if(tkcl.hasToken == true)  
            {  
                 
System.out.println(“Do you want to enter the Data –> YES/NO”);  
                br=new BufferedReader(new InputStreamReader(System.in));  
                str=br.readLine();  
                if(str.equalsIgnoreCase(“yes”))  
                {    
                    System.out.println(“ready to send”);  
                    tkser.setSendData = true;  
                    tkser.sendData();  
                    tkser.setSendData = false;  
                }  
                else if(str.equalsIgnoreCase(“no”))  
                {  
                    System.out.println(“i m in else”);  
                    //tkcl.hasToken=false;  
                    tkcl.sendData();  
                    tkcl.recData();  
                System.out.println(“i m leaving else”);  
                }  
            }  
            else  
            {  
            System.out.println(“ENTERING RECEIVING MODE…”);  
                tkcl.recData();  
            }  
    }  
}

}

class TokenClient12  
{  
    InetAddress lclhost;  
    int sendport,recport;  
    boolean hasToken = true;  
    boolean setSendData = false;  
    TokenClient12 tkcl,tkser;  
    TokenClient12(InetAddress lclhost)  
    {  
         
        this.lclhost = lclhost;  
    }  
     
    void setSendPort(int sendport)  
    {  
        this.sendport = sendport;  
    }

    void setRecPort(int recport)     
    {  
        this.recport = recport;  
    }

void sendData() throws Exception  
        {  
        BufferedReader br;  
        String str=”Token”;  
        DatagramSocket ds;  
        DatagramPacket dp;  
         
     
        if(setSendData == true)  
        {  
            System.out.println(“sending “);  
            System.out.println(“Enter the Data”);  
            br=new BufferedReader(new InputStreamReader(System.in));  
            str = “ClientOne…..” + br.readLine();  
            System.out.println(“now sending”);  
                     
        }  
            ds = new DatagramSocket(sendport);  
            dp = new DatagramPacket(str.getBytes(),str.length(),lclhost,sendport-1000);  
            ds.send(dp);  
            ds.close();  
            setSendData = false;  
            hasToken = false;  
    }  
     
    void recData()throws Exception  
    {  
        String msgstr;  
        byte buffer[] = new byte[256];  
        DatagramSocket ds;  
        DatagramPacket dp;

        ds = new DatagramSocket(recport);  
        dp = new DatagramPacket(buffer,buffer.length);  
        ds.receive(dp);  
        ds.close();  
        msgstr = new String(dp.getData(),0,dp.getLength());  
        System.out.println(“The data is “+msgstr);  
         
        if(msgstr.equals(“Token”))  
            {  
                hasToken = true;  
            }  
    }  
     
}  
       

**TokenClient2.java**

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

public class TokenClient2  
{  
    static boolean setSendData ;  
    static boolean hasToken ;  
    public static void main(String arg[]) throws Exception  
        {  
            InetAddress lclhost;  
            BufferedReader br;  
            String str1;  
            TokenClient21 tkcl;  
            TokenClient21 ser;  
            while(true)  
            {  
            lclhost=InetAddress.getLocalHost();  
            tkcl = new TokenClient21(lclhost);  
            tkcl.setRecPort(8004);         
            tkcl.setSendPort(9002);  
            lclhost=InetAddress.getLocalHost();  
            ser = new TokenClient21(lclhost);  
            ser.setSendPort(9000);  
            System.out.println(“entering if”);     
            if(hasToken == true)  
            {  
                 
System.out.println(“Do you want to enter the Data –> YES/NO”);  
                br=new BufferedReader(new InputStreamReader(System.in));  
                str1=br.readLine();  
                if(str1.equalsIgnoreCase(“yes”))  
                {     
                    System.out.println(“ignorecase”);  
                    ser.setSendData = true;  
                    ser.sendData();  
                    }  
                else if(str1.equalsIgnoreCase(“no”))  
                {  
                    tkcl.sendData();  
                    hasToken=false;  
                }  
            }  
            else  
            {  
            System.out.println(“entering recieving mode”);     
                tkcl.recData();  
                hasToken=true;  
            }  
        }  
    }  
}  
class TokenClient21  
{  
    InetAddress lclhost;  
    int sendport,recport;  
    boolean setSendData = false;  
    boolean hasToken = false;  
    TokenClient21 tkcl;  
    TokenClient21 ser;

    TokenClient21(InetAddress lclhost)  
    {  
         
        this.lclhost = lclhost;  
    }  
     
    void setSendPort(int sendport)  
    {  
        this.sendport = sendport;  
    }

    void setRecPort(int recport)     
    {  
        this.recport = recport;  
    }

    void sendData() throws Exception  
    {  
        System.out.println(“case”);  
        BufferedReader br;  
        String str=”Token”;  
        DatagramSocket ds;  
        DatagramPacket dp;  
         
        if(setSendData == true)  
        {  
            System.out.println(“Enter the Data”);  
            br=new BufferedReader(new InputStreamReader(System.in));  
            str = “ClientTwo…..” + br.readLine();  
        }  
            ds = new DatagramSocket(sendport);  
            dp = new DatagramPacket(str.getBytes(),str.length(),lclhost,sendport-1000);  
            ds.send(dp);  
            ds.close();  
            System.out.println(“Data Sent”);  
            setSendData = false;  
            hasToken = false;  
         
    }  
   

void recData()throws Exception  
    {  
        String msgstr;  
        byte buffer[] = new byte[256];  
        DatagramSocket ds;  
        DatagramPacket dp;  
ds = new DatagramSocket(recport);  
        //ds = new DatagramSocket(4000);  
        dp = new DatagramPacket(buffer,buffer.length);  
        ds.receive(dp);  
        ds.close();  
msgstr = new String(dp.getData(),0,dp.getLength());  
        System.out.println(“The data is “+msgstr);  
        if(msgstr.equals(“Token”))  
            {  
                hasToken = true;  
            }  
    }  
     
}

**OUTPUT :**

**TokerServer.java**

>java TokenServer  
The message is ClientOne…..hello  
The message is ClientTwo…..hii

**TokelClient1.java**

>java TokenC  
Client1  
Do you want to enter the Data –> YES/NO  
yes  
ready to send  
sending  
Enter the Data  
hello  
now sending  
Do you want to enter the Data –> YES/NO  
no  
i m in else

**TokenClient2.java**

>java TokenClient2  
entering if

entering recieving mode  
The data is Token  
entering if  
Do you want to enter the Data –> YES/NO  
yes  
ignorecase  
case  
Enter the Data  
hii  
Data Sent  
entering if  
Do you want to enter the Data –> YES/NO