**Experiment No: 07**

Name: Suraj P. Patil Roll: 4120

Class: B.Tech URN: 20131086

Div: B Date: 07-10-2023

**Title:** Implementation of Mutual Exclusion (Tocken Ring) Algorithm.

**Source Code:**

import java.io.\*;

import javax.swing.\*;

import java.net.\*;

import java.awt.\*;

import java.awt.event.\*;

public class TockenRingnew extends JFrame implements Runnable,ActionListener

{

Label l1,l[];

TextArea sl;

Button b1,b[];

TextField tf1,tf[];

JFrame f[],sf;

int nop=0;

Container c;

int Tocken[];

static int j=0;

int CRUsing=0;

Thread t[],at;

int Que[];

TockenRingnew()throws Exception

{

super("Distributed System");

setSize(200,200);

setVisible(true);

c=getContentPane();

c.setLayout(new FlowLayout());

l1=new Label("Enter No. Of Processes In your Distributed Systems");

tf1=new TextField(10);

b1=new Button("Create");

c.add(l1);c.add(tf1);c.add(b1);

validate();

b1.addActionListener(this);

}

public void run()

{

try{

while(true)

{

if(CRUsing!=1)

{

Tocken[j]=0;

j++;

System.out.println("Tocken given to Process no."+j);

sl.setText(sl.getText()+"\nTocken given to Process no."+j);

if (j==nop)

{

j=0;

}

Tocken[j]=1;

if (Que[j]==1)

{

System.out.println("You Can Enter in CR");

CRUsing=1;

l[j].setText("Click \"Task Completed\" While Releasing CR");

b[j].setLabel("Task Completed");

b[j].setEnabled(true);

sl.setText(sl.getText()+"\nIn que of process "+j+" Req is present , it is entering in CR");

at.suspend();

}

Thread.sleep(6000);

System.out.println(j);

}

}}catch(Exception dd){System.out.println(dd);}

}

public void actionPerformed(ActionEvent ae)

{

try{

String cmd=ae.getActionCommand();

if(cmd.equals("CRITICAL REGION"))

{

System.out.println("CRITICAL REGION");

for (int jj=0;jj<nop ;jj++ )

{

if (b[jj].equals(ae.getSource()))

{

if (Tocken[jj]==1)

{

sl.setText(sl.getText()+"\nProcess No."+jj+"Want to Enter in CR , It can Enter as Tocken is Localy available");

System.out.println("You Can Enter in CR Tocken localy available");

b[jj].setLabel("Task Completed");

l[jj].setText("Click \"Task Completed\" While Releasing CR");

CRUsing=1;

at.suspend();

}

else

{

Que[jj]=1;

b[jj].setLabel("WAIT");

b[jj].setEnabled(false);

sl.setText(sl.getText()+"\nProcess No."+jj+"Want to Enter in CR "+",It is Waiting For Entering in CR");

}

}

}

}

else if (cmd.equals("Task Completed"))

{

at.resume();

for (int jj=0;jj<nop ;jj++ )

{

if (b[jj].equals(ae.getSource()))

{

b[jj].setLabel("CRITICAL REGION");

l[jj].setText("If You Want to Enter in CR PRESS \"CRITICAL REGION\" Button");

Que[jj]=0;

CRUsing=0;

}

}

}

else if(cmd.equals("Create"))

{

sf=new JFrame("Status Frame");

sl=new TextArea(20,60);

sl.setText("In Logical Ring "+nop+" Number of processes are included");

sf.setSize(400,200);

sf.setVisible(true);

sf.getContentPane().setLayout(new FlowLayout());

sf.getContentPane().add(sl);

sl.setEditable(false);

setVisible(false);

nop=Integer.parseInt(tf1.getText());

f=new JFrame[nop];

Tocken=new int[nop];

t=new Thread[nop];

l=new Label[nop];

tf=new TextField[nop];

b=new Button[nop];

Que=new int[nop];

for (int i=0;i<nop ;i++ )

{

Tocken[0]=0;

}

Tocken[0]=1;

at=new Thread(this);

at.start();

for (int i=0;i<nop ;i++ )

{

f[i]=new JFrame("Process "+i);

f[i].setSize(300,100);

f[i].setVisible(true);

f[i].getContentPane().setLayout(new FlowLayout());

tf[i]=new TextField(10);

l[i]=new Label("If You Want to Enter in CR PRESS \"CRITICAL REGION\" Button");

b[i]=new Button("CRITICAL REGION");

f[i].getContentPane().add(l[i]);

f[i].getContentPane().add(b[i]);

b[i].addActionListener(this);

f[i].validate();

}//for

}//if

}catch(Exception e){System.out.println(e);}

}

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

{

TockenRingnew tr=new TockenRingnew();

}

}

Output :

