|  |  |
| --- | --- |
| Name: Sayyed Sohail Rashid | Course Name: DC-LAB |
| Class: BE-CO | Batch: 01 |
| Roll no: 18CO48 | Experiment No: 06 |

Aim : To Implement the Lamport Clock Synchronization Algorithm.

Code:

Lamport.java

import java.io.\*;

import java.util.\*;

import java.util.Scanner;

import javax.swing.\*;

import java.awt.\*;

import java.awt.geom.\*;

public class lamport{

int e[][]=new int[10][10];

int en[][]=new int[10][10];

int ev[]=new int[10];

int i,p,j,k;

HashMap<Integer,Integer> hm=new HashMap<Integer,Integer>();

int xpoints[] =new int[5];

int ypoints[] =new int[5];

class draw extends JFrame{

private final int ARR\_SIZE = 4;

void drawArrow(Graphics g1, int x1, int y1, int x2, int y2) {

Graphics2D g = (Graphics2D) g1.create();

double dx = x2 - x1, dy = y2 - y1;

double angle = Math.atan2(dy, dx);

int len = (int) Math.sqrt(dx\*dx + dy\*dy);

AffineTransform at = AffineTransform.getTranslateInstance(x1, y1);

at.concatenate(AffineTransform.getRotateInstance(angle));

drawArrow(g, 30, 300, 300, 190);

}

}

}

for(i=1;i<=p;i++)

{

for(j=2;j<=ev[i];j++)

{

k=i\*10+j;

if(hm.get(k)==0)

{

en[i][j]=en[i][j-1]+1;

}

}

for(i=1;i<=p;i++)

{

for(j=1;j<=ev[i];j++)

{

System.out.println(en[i][j]);

}

}

JFrame jf=new draw();

jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jf.setSize(500,500);

jf.setVisible(true);

}

public static void main(String[] args){

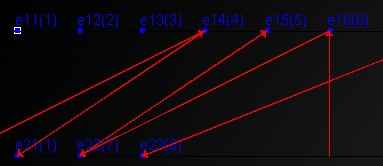
lamport lam=new lamport();

lam.calc();

}

}

Output:



Conclusion:

The two processes from which one has 6 events and next has 3 events is synchronized using the lamport clock synchronization algorithm.