

Programación Avanzada

Juan Manuel Fernández Peña Curso 2013

Simulación de un tinaco

El ejemplo que se incluye aquí representa la simulación de una casa donde se recibe agua a ciertas horas, almacenándola en un tinaco. Al mismo tiempo, en diferentes horarios, se utiliza agua a través de llaves. La simulación se asegura de que no haya conflictos en la región crítica, que es el tinaco. El mecanismo seleccionado para lograr evitar conflictos es el cerrojo. Inicialmente se muestran el cerrojo y el tinaco, que serán compartidos; luego los dos tipos de hilos y finalmente la clase Hogar que controla todo y tiene el main.

```
public class Cerrojo {
    private boolean protegido = false;

    public synchronized void abrir() throws InterruptedException {
        while (protegido) wait();
        protegido = true;
    }

    public synchronized void cierra(){
        protegido = false;
        notify();
    }
}
```

```
public class Tinaco {
    private int contenido;
    private int capacidad;

    public Tinaco(int cont, int cap){
        contenido = cont;
        capacidad = cap;
    }

    public int agrega(int cant){
        if (capacidad - contenido > cant)
            contenido += cant;
        else contenido = capacidad;
        return contenido;
    }

    public int saca(int cant){
        int resp =0;
        if (contenido > cant){
            resp = cant;
            contenido -= cant;
        }
        else {
            resp = contenido;
            contenido = 0;
        }
        return resp;
    }

    public int getContenido(){
        return contenido;
    }
}
```

```

import java.util.Calendar;
import java.util.Random;

public class Llegada extends Thread {
    private int lps;
    private int sueño1, sueño2;
    private Cerrojo chapa;
    private Tinaco tin;
    private boolean sigue = true;
    Random alea = new Random();

    public Llegada(int ll, int seg1, int seg2, Cerrojo c, Tinaco t){
        lps = ll;
        sueño1 = seg1 * 1000;
        sueño2 = seg2;
        chapa = c;
        tin = t;
    }

    public void run(){
        int lim = alea.nextInt(20)+5;
        int sue = sueño1 * (alea.nextInt(10));
        while (sigue){
            for (int ix=0;ix<lim;ix++){
                surte();
                try{
                    sleep(sueño2);
                }catch (InterruptedException ie){
                    //pendiente
                }
            }
            try{
                sleep(sue);
            }catch (InterruptedException ie){
                //pendiente
            }
        }
    }

    private void surte(){
        try{
            chapa.abrir();
            tin.agrega(lps);
            System.out.println(">>
"+Calendar.getInstance().getTimeInMillis()%100000+" >> Nivel "+tin.getContenido());
            chapa.cierra();
        }catch (InterruptedException ie){
            //pendiente
        }
    }

    public void detener(){
        sigue = false;
    }
}

```

```

import java.util.Calendar;
import java.util.Random;

public class SaleAgua extends Thread {

```

```

private int lps;
private int sueño1, sueño2;
private Cerrojo chapa;
private Tinaco tin;
private boolean sigue = true;
private static int número = 0;
private int minúmero;
private Random alea = new Random();

public SaleAgua(int ll, int seg1, int seg2, Cerrojo c, Tinaco t){
    lps = ll;
    sueño1 = seg1;
    sueño2 = seg2;
    chapa = c;
    tin = t;
    minúmero = número;
    número++;
}

public void run(){
    int lim = alea.nextInt(4)+2;
    int sue = sueño2 * (alea.nextInt(3));
    while (sigue){
        for (int ix=0;ix<lim;ix++){
            extrae();
            try{
                sleep(sueño1);
            }catch (InterruptedException ie){
                //pendiente
            }
        }
        try{
            sleep(sue);
        }catch (InterruptedException ie){
            //pendiente
        }
    }
}

private void extrae(){
    try{
        chapa.abrir();
        tin.saca(lps);
        System.out.println("<<
"+Calendar.getInstance().getTimeInMillis()%100000+" << "+minúmero+" Nivel
"+tin.getContenido());
        chapa.cierra();
    }catch (InterruptedException ie){
        //pendiente
    }
}

public void detener(){
    sigue = false;
}
}

```

```

public class Hogar {
    private Tinaco eltinaco;
    private Cerrojo cerr;
    private Llegada lle;
    private SaleAgua sal, sa2;
}

```

```
public Hogar(){
    eltinaco = new Tinaco(500,1000);
    cerr = new Cerrojo();
    sa2 = new SaleAgua(2,25, 6000, cerr, eltinaco);
    sa1 = new SaleAgua(1,50, 5000, cerr, eltinaco);
    lle = new Llegada(5,10,200, cerr, eltinaco);
}

public void actúa(){
    lle.start();
    sa1.start();
    sa2.start();
    try{
        Thread.sleep(100000);
    }catch (InterruptedException ie){
        //pendiente
    }
    lle.detener();
    sa1.detener();
    sa2.detener();
}

public static void main(String[] args) {
    Hogar h=new Hogar();
    h.actúa();
}

}
```