

PCTR Practica 9

Ignacio Rodríguez Pérez

Ejercicio 1: Monitor impresoras

1 Código

```
public class monitorImpresion
{
    private static int nImpresoras = 3;
    private int impresorasDisponibles;
    private boolean[] impresoras = new boolean[nImpresoras];
    private int[] procesos;

    public monitorImpresion(int n_procesos)
    {
        impresorasDisponibles = nImpresoras;
        procesos = new int[n_procesos];
        for (int i = 0; i < impresoras.length; i++)
            impresoras[i] = true;
    }

    public synchronized void pedir_impresora(int hilo)
    {
        while (impresorasDisponibles == 0)
        {
            try
            {
                System.out.println("Todas las impresoras ocupadas"
                    + ". Proceso " + hilo + " en espera");
                wait();
            } catch (InterruptedException e) { }
            int n = 0;
            while (!impresoras[n]) n++;
            --impresorasDisponibles;
            impresoras[n] = false;
            procesos[hilo] = n;
            System.out.println("El proceso " + hilo + " ha cogido a"
                + " la impresora " + procesos[hilo]);
        }
    }

    public synchronized void dejar_impresora(int hilo)
    {
        int impresora = procesos[hilo];
        impresoras[impresora] = true;
        ++impresorasDisponibles;
        procesos[hilo] = -1;
        System.out.println("El proceso " + hilo
            + " suelta la impresora " + impresora);
        notifyAll();
    }
}
```

2 Explicación

En este ejemplo, iniciamos con tres impresoras disponibles.

Teniendo en el método `run` un bucle infinito, con dos llamadas (una que pide una impresora, y otra que deja disponible la misma; pasándole a ambas el entero que representa el id del hilo), tenemos que primera llamada que se haga, será la petición de una impresora (dormimos un poco el hilo para que no se de el caso de que suelte una impresora que no haya dejado a coger).

En la llamada que solicita una impresora, hemos de comprobar que hay impresoras disponibles. Como puede darse el caso de que no haya en algún momento impresoras libres, mandamos a la cola del `wait-set` dicho hilo, y como se puede dar el caso de que, una vez se despierten a todos (con `notifyAll`), hemos de colocarle una condición de guarda para que, una vez se despierte, vuelva a comprobar que tiene impresoras disponibles.

Una vez tenemos el caso de que haya alguna impresora disponible, buscamos cual es la primera (con una simple búsqueda secuencial), de modo que la variable `n` es el índice en el vector de la primera impresora disponible, luego pasa a estar ocupada, lo cual se refleja con `impresora[n] = false`. Disminuimos en uno la cantidad de impresoras disponibles, y guardamos en otro vector que proceso ha guardado qué impresora, para luego mostrarlo por pantalla.

Por último, tenemos el caso de que el hilo acabe su ejecución y desee liberar la impresora. En este caso, como sabemos el hilo que libera dicha impresora (se le pasa al método), guardamos la impresora que será liberada, el vector en dicha posición lo actualizamos a verdadero (o sea, que estaremos indicando que esa impresora en concreto está disponible), incrementamos en uno el valor de las impresoras disponibles y mostramos por pantalla la impresora liberada por el hilo en ejecución.