

## Documentación del problema de los fumadores y el problema de los lectores-escriutores mediante el uso de monitores tipo SU

## Problema de los fumadores

El uso de monitores tipo SU simplifica mucho la resolución de este problema, frente a su contraparte con monitores.

Para resolver este problema creamos una clase derivada de la clase “HoareMonitor” llamada “Estanco”.

Esta clase tiene como componentes privadas un entero, que hace referencia al número de ingrediente que se encuentra en el mostrador en un momento determinado (por defecto se inicializa a 4) y 2 colas de condición: una para el estanquero y un array de CondVar con tantos elementos como fumadores haya (3 en este caso).

Tenemos como métodos:

- obtenerIngrediente(int n), que pone a cada fumador a esperar su respectivo ingrediente (el ingrediente depende del número del fumador, que es el parámetro n que recibe el método)
- ponerIngrediente(int ingrediente), que pone el ingrediente en el mostrador, asegurándose primero de que está vacío. El ingrediente lo recibe el método como parámetro, ya ha sido generado con anterioridad.
- EsperarRecogida(), que hace que el estanquero no comience a producir un nuevo ingrediente hasta que el anterior haya sido recogido.

El resultado obtenido es exactamente el mismo que el que se obtiene con semáforos, solo que la implementación del código con monitores se simplifica muchísimo en comparación a los semáforos.

## Problema de los lectores-escritores

Este problema consta de una serie de lectores, que escriben datos en una memoria compartida (un array) y unos lectores que leen los datos de la misma.

Tenemos las siguientes condiciones:

- No puede haber varios escritores escribiendo a la vez.
- No se puede escribir mientras se leen datos.
- Los lectores deben ejecutarse de forma concurrente.

Para la resolución del ejercicio creamos una clase llamada “Lec\_Esc” derivada de “HoareMonitor”.

Esta clase tiene las siguientes componentes privadas:

- Un booleano para saber si hay algún escritor escribiendo
- Un entero n\_lec que indica el número de lectores leyendo en un momento de tiempo determinado
- Un entero dato, que se usa de contador para los lectores
- Un entero valor, que se usa de contador para los escritores
- Un entero n\_datos que se usa para saber el número de datos que aún no han sido leídos
- Una cola de condición para los lectores
- Una cola de condición para los escritores

Y los siguientes métodos:

- ini\_lec(), que se asegura de que haya datos que leer y no haya escritores escribiendo, también avisa a los demás lectores si hay datos suficientes.
- Leer(), que lee el valor correspondiente.
- fin\_lec(), que si no quedan más lectores leyendo, avisa a un escritor.
- ini\_esc(), Si hay lectores leyendo u otro escritor escribiendo, espera.
- escribir(), escribe el valor correspondiente.
- fin\_esc(), si hay lectores esperando, avisa a un lector, y si no avisa a un escritor.

Como vemos, este problema relativamente complejo se puede resolver de una forma no muy complicada mediante el uso de monitores SU.