

Múltiplas Variáveis de Condição por Monitor

Grupo de Sistemas Distribuídos
Universidade do Minho

Objectivos

Uso das facilidades de biblioteca em Java relativas a variáveis de condição, para resolver problemas com monitores para os quais seja útil ter mais do que uma variável de condição.

Mecanismos

Variáveis de condição associadas a `ReentrantLock`:

- interface `Lock`: método `newCondition()`;
- interface `Condition`: métodos `await()`, `signal()`, `signalAll()`;

Exercícios propostos

1 Implemente uma classe `Warehouse` para permitir a gestão de um armazem acedido concorrentemente. Deverão ser disponibilizados os métodos:

```
void supply(String item, int quantity) throws InterruptedException { ... }  
void consume(Set<String> items) throws InterruptedException { ... }
```

A operação `supply` abastece o armazem com uma dada quantidade de um item; a operação `consume(Set<String> items)` obtém do armazem um conjunto de itens, bloqueando enquanto tal não for possível. Escreva duas versões da abstracção:

1. A versão “egoísta”, centrada nos clientes, em que cada cliente tenta apropriar-se dos itens o mais cedo possível.
2. A versão “cooperativa”, que optimiza o uso do armazém como um todo, não reservando itens para clientes que não possam ser satisfeitos no momento (porque faltam alguns).

Pode partir da versão sequencial fornecida. Esta está incompleta pois não funciona caso falte algum produto requisitado.

Exercícios Adicionais

2 A segunda versão do armazém poderá levar a situações de *starvation*, em que um dado cliente continua à espera indefinidamente, quando os itens desejados, e que foram entretanto chegando, vão sendo atribuídos a outros clientes. Modifique o armazém para, mantendo-se cooperativo, evitar este problema.