Monitores nativos em Java

Grupo de Sistemas Distribuídos
Universidade do Minho

Objectivos

Uso de monitores nativos de Java em problemas de ordem de execução.

Mecanismos

Keyword `synchronized`. Métodos `wait`, `notify` e `notifyAll` de `Object`.

Exercícios propostos

1 Escreva uma abstracção para permitir que N threads se sincronizem:

```
class Barrier {
    Barrier (int N) { ... }
    void await() throws InterruptedException { ... }
}
```

A operação `await` deverá bloquear até que as N threads o tenham invocado; nesse momento o método deverá retornar em cada thread. Escreva duas versões desta abstracção:

1. Suponha que cada thread apenas vai invocar `await` uma vez sobre o objecto.
2. Permita que a operação possa ser usada várias vezes por cada thread (barreira reutilizável), de modo a suportar a sincronização no fim de cada uma de várias fases de computação.

2 Generalize a abstracção de barreira para uma abstracção de `Agreement`:

```
class Agreement {
    Agreement (int N) { ... }
    int propose(int choice) throws InterruptedException { ... }
}
```

Esta deve permitir que as N threads se sincronizem para chegar a acordo num valor. Cada thread propõe um valor, ficando o método `propose` bloqueado até todas a N o terem feito; nesse momento o método deverá retornar em cada thread o máximo dos valores propostos. Tal como na barreira reutilizável, deverá ser possível uma sucessão de acordos.