

Sistemas Distribuídos

2ª Avaliação

28 de Janeiro de 2011

Duração: 1h30m

I

1 Considere o conceito de monitor e a sua implementação numa linguagem genérica. Discuta as vantagens e desvantagens da sua utilização relativamente a outras primitivas de controlo de concorrência que estudou.

[illegible]

2 Considere uma aplicação distribuída que requer o acesso mutuamente exclusivo por cada um dos participantes a um determinado recurso único. Explique sucintamente como implementaria o controlo de concorrência da aplicação com recurso às primitivas de comunicação que estudou de forma a não ocorrerem esperas activas.

[illegible]

II

Considere que dispõe da classe `Calculo` apresentada abaixo. Implemente um servidor TCP/IP que permita a invocação de métodos desta classe. O servidor deverá ser capaz de tratar um máximo de M clientes simultaneamente. Em cada conexão, um cliente poderá especificar a lista de métodos (e respectivos parâmetros) que pretende ver invocados no servidor. Por sua vez, o servidor devolverá ao cliente o valor médio dos resultados retornados pelos métodos indicados na conexão. No servidor os métodos poderão ser invocados concorrentemente, mas em cada momento não podem decorrer mais do que X execuções de cada método (globalmente). Procure maximizar a concorrência potencial da sua solução recorrendo ao mecanismos de *threads* e sincronização estudados nas aulas.

```

class Calculo {
    public static float m_1(float a, float b) // operação demorada
    // ...
    public static float m_N(float a, float b) // operação demorada

    public static float invocar(int n, float a, float b) // invocar metodo #n
}

```

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

Algumas primitivas relevantes

- Object o;
synchronized(o) { ... };
o.wait();
o.notify();
o.notifyAll();

- `Lock l = new ReentrantLock();`
`Condition c = l.newCondition();`
`l.lock();`
`l.unlock();`
`c.await();`
`c.signal();`
`c.signalAll();`