LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Programação Concorrente

Teste de Avaliação[1]

14 de Junho de 2016 Duração: 2h00m

I

1 Explique o conceito de *deadlock* e descreva um cenário onde este fenómeno poderá acontecer.

2 Diga o que entende por *spurious wakeup*, e explique as suas consequências na metodologia de programar com monitores. Dê um exemplo de um problema que seria resolvido trivialmente caso o fenómeno não existisse.

3 É comum ouvir "como esta thread apenas vai ler as variáveis, não necessito de usar um lock". Comente esta afirmação.

II

Pretende-se que escreva em Java, fazendo uso de primitivas baseadas em monitores, uma classe para monitorização de eventos num ambiente *multithreaded* em que os métodos a disponibilizar são a sinalização e a espera por eventos.

```
void sinaliza(int tipo);
void espera(int tipo1, int n1, int tipo2, int n2);
```

O método `espera` deverá bloquear até serem `sinaliza`dos, depois do momento em que é invocado: `n1` vezes um evento `tipo1`, e `n2` vezes um evento `tipo2`. Considere que o tipo de um evento é um inteiro entre 1 e E, especificado no construtor.

III

Apresente o código Erlang de um processo servidor relativamente à mesma situação descrita no grupo II. Suponha que os clientes são processos Erlang, que comunicam pelo mecanismo nativo de mensagens, e implemente também as funções de interface apropriadas para serem usadas por estes.

---

[1] Cotação — 5+8+7