LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Programação Concorrente

Exame de Recurso[1]

8 de junho de 2022 Duração: 2h00m

I

1 Explique o conceito de *starvation* e descreva um cenário que seja propício à ocorrência deste fenómeno.

2 É comum encontrar generalizações de semáforos com operações como `acquire(n)` e `release(n)`. Explique a semântica destas operaçoes e compare-as com as operações clássicas.

3 Explique porque motivos o `wait()` numa variável de condição de um monitor em linguagens actuais deve ser efectuado dentro de um ciclo que (re)testa um predicado.

II

Considere um sistema de controlo de acesso a dois recursos. Apenas um recurso pode estar a ser acedido em cada momento e no máximo podem estar T threads a aceder a um recurso. Pretende-se que escreva em Java, fazendo uso de primitivas baseadas em monitores, uma classe que implemente a interface:

```
interface Controller {
    int request_resource(int i);
    void release_resource(int i);
}
```

O `request_resource` tem como parâmetro o recurso pretendido (identificado por um número que pode ser 0 ou 1), e deverá bloquear até o recurso poder ser acedido; `release_resource` é invocado quando uma thread completou o uso do recurso correspondente. Tente evitar *starvation*.

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 — 6+8+6