

Programação Concorrente

*Exame de Recurso*¹

15 de Junho de 2021

Duração: 2h00m

I

- 1 Explique o conceito de *starvation* e descreva um cenário que seja propício à ocorrência deste fenómeno.
- 2 Se estivermos a usar monitores, podemos ser tentados a resolver qualquer problema de sincronização com apenas uma variável de condição. Explique que problemas tem tal abordagem. Dê um exemplo.
- 3 Explique para que servem variáveis *volatile* em Java. Dê um exemplo de uma situação que pode ser resolvida por este conceito e de outra situação para a qual este conceito não ajuda.

II

Suponha que pretende implementar um serviço de execução de tarefas, para limitar a concorrência na execução destas. Supondo que existem as prioridades 1, 2, e 3, o serviço deverá: permitir sempre iniciar a execução de tarefas de prioridade 3; iniciar tarefas de prioridade 2 quando estão menos de N tarefas concorrentes em execução; apenas permitir iniciar tarefas de prioridade 1 quando estão menos de N tarefas em execução e todas dessa prioridade. Deverá ser disponibilizada a interface:

```
interface Escalonador {  
    void executa(Runnable tarefa, int prioridade);  
}
```

em que `executa` deve invocar o método `run` do objecto `tarefa` passado como argumento, respeitando as restrições de concorrência acima enunciadas. Codifique em Java uma classe que implemente esta interface.

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.

¹Cotação — 6+8+6