LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Programação Concorrente

Época Especial[1]

20 de julho de 2022 Duração: 2h00m

I

1 Suponha que tem várias threads, cada uma a efectuar operações que envolvam vários recursos partilhados. Explique que problemas podem surgir e descreva uma técnica de controlo de concorrência que os resolva.

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 Explique porque em Erlang, para esperar por uma mensagem de resposta a um pedido, não deve ser utilizado `receive X -> use(X) end`, e diga como deve ser feito.

II

Pretende-se que escreva em Java, fazendo uso de primitivas baseadas em monitores, código que permita jogadores, cada um representado por uma thread, participarem num jogo. Cada partida envolve os jogadores que manifestaram a intenção de *participar*, no intervalo de tempo de um minuto contado desde que o primeiro jogador manifestou o interesse (findo esse tempo, o próximo jogador começará a contagem de tempo para a partida seguinte). Podem estar várias partidas a decorrer em simultâneo. Implemente as interfaces:

```
interface Jogo {
    Partida participa();
}

interface Partida {
    boolean aposta(int n, int media);
}
```

A operação `participa()` deverá bloquear até poder começar uma partida, devolvendo o objecto que a representa. Este objecto é usado pelos jogadores dessa partida, que consiste em cada jogador escolher um número n entre 1 e 10, e tentarem adivinhar qual é a média dos números escolhidos por todos. Para tal, cada jogador faz uma única invocação da operação `aposta(n, media)`, que deverá devolver `true` se o jogador acertou na média dos números escolhidos por todos, ou `false` caso contrário.

III

Apresente o código Erlang para implementar o mesmo sistema descrito no grupo II, através de um ou mais processos. Supondo que os jogadores são representados por processos Erlang que comunicam pelo mecanismo nativo de mensagens, implemente também as funções de interface apropriadas para serem usadas por estes para interagirem com o(s) processos relevante(s).

[1] Cotação — 6+8+6