

## Exercício - Capítulo 7 – Sincronismo

1) Defina o que é uma aplicação concorrente e dê um exemplo de sua utilização.

R: É uma aplicação estruturada de maneira que partes diferentes do código do programa possam executar concorrentemente. Este tipo de aplicação tem como base a execução cooperativa de múltiplos processos ou threads, que trabalham em uma mesma tarefa na busca de um resultado comum.

2) Considere uma aplicação que utilize uma matriz na memória principal para a comunicação entre vários processos concorrentes. Que tipo de problema pode ocorrer quando dois ou mais processos acessam uma mesma posição da matriz?

R: Caso não haja uma gerência no uso concorrente dos recursos compartilhados, inconsistências nos dados podem ocorrer.

3) O que é exclusão mútua e como é implementada?

R: É impedir que dois ou mais processos acessem um mesmo recurso simultaneamente. Para isso, enquanto um processo estiver acessando determinado recurso, todos os demais processos que queiram acessá-lo deverão esperar pelo término da utilização do recurso

4) Como seria possível resolver os problemas decorrentes do compartilhamento da matriz, apresentado anteriormente, utilizando o conceito de exclusão mútua?

R: Garantindo na aplicação que somente um único processo pode estar acessando a matriz por vez.

5) O que é starvation e como podemos solucionar esse problema?

R: Starvation é a situação onde um processo nunca consegue executar sua região crítica e, conseqüentemente, acessar o recurso compartilhado. A solução para o problema depende de estabelecimentos de mecanismos de acesso pelo sistema operacional que garantam o acesso ao recurso por todos os processos que solicitarem uso.

6) Qual o problema com a solução que desabilita as interrupções para implementar a exclusão mútua?

R: Essa solução apesar de simples, apresenta algumas limitações. Primeiramente, a multiprogramação pode ficar seriamente comprometida, já que a concorrência entre processos tem como base o uso de interrupções. Um caso mais grave poderia ocorrer caso um processo desabilitasse as interrupções e não tornasse a habilitá-las. Nesse caso, o sistema, provavelmente, teria seu funcionamento seriamente comprometido.

Em sistemas com múltiplos processadores, esta solução torna-se ineficiente devido ao tempo de propagação quando um processador sinaliza aos demais que as interrupções devem ser habilitadas ou desabilitadas. Outra consideração é que o mecanismo de clock do sistema é implementado através de interrupções, devendo esta solução ser utilizada com bastante critério

7) O que é espera ocupada e qual o seu problema?

R: Na espera ocupada, toda vez que um processo não consegue entrar em sua região crítica, por já existir outro processo acessando o recurso, o processo permanece em looping, testando uma condição,

até que lhe seja permitido o acesso. Dessa forma, o processo em looping consome tempo do processador desnecessariamente, podendo ocasionar problemas ao desempenho do sistema.

8) Explique o que é sincronização condicional e dê um exemplo de sua utilização.

R: Sincronização condicional é uma situação onde o acesso ao recurso compartilhado exige a sincronização de processos vinculada a uma condição de acesso. Um recurso pode não se encontrar pronto para uso devido a uma condição específica. Nesse caso, o processo que deseja acessá-lo deverá permanecer bloqueado até que o recurso fique disponível. Um exemplo clássico desse tipo de sincronização é a comunicação entre dois processos através de operações de gravação e leitura em um buffer.

9) Explique o que são semáforos e dê dois exemplos de sua utilização: um para a solução da exclusão mútua e outro para a sincronização condicional.

R: Um semáforo é uma variável inteira, não negativa, que só pode ser manipulada por duas instruções: DOWN e UP. Ver slide 20,21 e 22.

10) Explique o que são monitores e dê dois exemplos de sua utilização: um para a solução da exclusão mútua e outro para a sincronização condicional.

R: Monitores são mecanismos de sincronização de alto nível que torna mais simples o desenvolvimento de aplicações concorrentes. Ver slides 23,24, e 25

11) Qual a vantagem da forma assíncrona de comunicação entre processos e como esta pode ser implementada?

R: A vantagem deste mecanismo é aumentar a eficiência de aplicações concorrentes. Para implementar essa solução, além da necessidade de buffers para armazenar as mensagens, devem haver outros mecanismos de sincronização que permitam ao processo identificar se uma mensagem já foi enviada ou recebida.

12) O que é deadlock, quais as condições para obtê-lo e quais as soluções possíveis?

R: Deadlock é a situação em que um processo aguarda por um recurso que nunca estará disponível ou um evento que não ocorrerá. Para que ocorra a situação de deadlock, quatro condições são necessárias simultaneamente:

- ⌚ exclusão mútua: cada recurso só pode estar alocado a um único processo em um determinado instante;
- ⌚ espera por recurso: um processo, além dos recursos já alocados, pode estar esperando por outros recursos;
- ⌚ não-preempção: um recurso não pode ser liberado de um processo só porque outros processos desejam o mesmo recurso;
- ⌚ espera circular: um processo pode ter de esperar por um recurso alocado a outro processo e vice-versa.

Para prevenir a ocorrência de deadlocks, é preciso garantir que uma das quatro condições apresentadas, necessárias para sua existência, nunca se satisfaça. A prevenção de deadlocks evitando-se a ocorrência de qualquer uma das quatro condições é bastante limitada e, por isso, na prática não é utilizada