Lista de Exercícios 7 – Sincronização de processos Parte 1 – Exclusão mútua e Semáforos.

Lista Teórica

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

   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?**

   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?**

   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?**

   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?**

   Starvation é a situação onde um processo nunca consegue executar sua região crítica e, consequentemente, 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?**

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?

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.

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 um exemplo de sua utilização para a solução da exclusão mútua.

Semáforos são mecanismos que resolvem o problema de exclusão mútua. Um semáforo pode ser visto como um objeto que pode sofrer dois tipos de operação sobre ele: trancando e destrancando a execução de instruções (p. ex., operações UP e DOWN, P e V). As operações sobre um semáforo são atômicas.

A exclusão mútua pode ser implementada através de um semáforo binário associado ao recurso compartilhado. A principal vantagem desta solução em relação aos algoritmos anteriormente apresentados é a não ocorrência da espera ocupada. Assim, as intruções DOWN e UP funcionam como protocolos de entrada e saída.