Questionário - 10

Arthur C. M. Barcella e Matheus P. Salazar

Explique o que são condições de disputa, mostrando um exemplo real.

Condições de disputa ocorrem quando duas ou mais tarefas alteram uma região de memória compartilhada ao mesmo tempo, podendo corromper a informação que ali está contida.

Pegamos o exemplo do depósito por dois processos, ao tentar incrementar o valor contido em uma memória compartilhada, caso não haja coordenação, o processo um irá ler o valor contido na posição de memória e incrementar seu valor, entretanto, o processo dois, irá ler o mesmo valor visto pelo processo um e incrementar, corrompendo o valor que deveria conter também a soma do processo um.

Sobre as afirmações a seguir, relativas aos mecanismos de coordenação, indique quais são incorretas, justificando sua resposta:

a. A estratégia de inibir interrupções para evitar condições de disputa funciona em sistemas multi-processados.

b. Os mecanismos de controle de entrada nas regiões críticas provêem exclusão mútua no acesso às mesmas.

c. Os algoritmos de busy-wait se baseiam no teste contínuo de uma condição.

d. Condições de disputa ocorrem devido às diferenças de velocidade na execução dos processos. FALSO (As condições ocorrem devido ao acesso às informações sem coordenação por parte das tarefas)

e. Condições de disputa ocorrem quando dois processos tentam executar o mesmo código ao mesmo tempo. (FALSO: Essa condição ocorre quando duas tarefas tentam modificar a mesma área de memória ao mesmo tempo)

f. Instruções do tipo Test&Set Lock devem ser implementadas pelo núcleo do SO. (FALSO: Devem ser implementadas pelo desenvolvedor da aplicação)

g. O algoritmo de Peterson garante justiça no acesso à região crítica. (FALSO: O algoritmo balanceia de melhor maneira o acesso a região, mas não garante justiça no acesso).

h. Os algoritmos com estratégia busy-wait otimizam o uso da CPU do sistema. (FALSO: Estes algoritmos fazem testes contínuos desnecessários reduzindo a eficiência da CPU).

i. Uma forma eficiente de resolver os problemas de condição de disputa é introduzir pequenos atrasos nos processos envolvidos. (FALSO: Essa forma é ineficiente pois o processador precisa aguardar além de aumentar o tempo de execução do processo).

Explique o que é espera ocupada e por que os mecanismos que empregam essa técnica são considerados ineficientes.

A espera ocupada se baseia no teste repetitivo de uma condição por um processo para que um conjunto de processos acessem uma mesma área de memória, mas apenas um de cada vez.

O teste se baseia em verificar o estado da variável de acesso a região de memória, quando um processo acessa a região de memória ele altera a variável de tal maneira que os demais processos não conseguem mais acessar a área de memória.

Quando o processo atual termina de usar a região de memória ele altera novamente a variável de acesso permitindo outros processos.

Em que circunstâncias o uso de espera ocupada é inevitável?

Nos casos onde o escalonador não permite criar uma fila de processos que desejam acessar uma mesma região de memória, o uso da espera ocupada é necessário para garantir que não ocorrerá uma condição de disputa. Entretanto, ao utilizar essa função, a eficiência da CPU irá ser alterada, devido aos testes contínuos realizados pela CPU.

Considere `ocupado` uma variável inteira compartilhada entre dois processos A e B (inicialmente, `ocupado = 0`). Sendo que ambos os processos executam o trecho de programa abaixo, explique em que situação A e B poderiam entrar simultaneamente nas suas respectivas regiões críticas.

```
1   while (true) {
2           regiao_nao_critica();
3           while (ocupado) {};
4           ocupado = 1;
5           regiao_critica();
6           ocupado = 0;
7   }
```

Considere `ocupado` uma variável inteira compartilhada entre dois processos A e B (inicialmente, `ocupado = 0`). Sendo que ambos os processos executam o trecho de programa abaixo, explique em que situação A e B poderiam entrar simultaneamente nas suas respectivas regiões críticas.

Os processos poderiam acessar a região crítica no momento do teste de verificação da variável ocupado, como é realizado primeiro um teste para depois setar a variável, enquanto um processo sai do teste e se encaminha para alterar a variável ocupado, outro processo poderia realizar o teste de while com a variável ainda em 0.

Ao alterar a variável para um afirmando que a região já está ocupada, o outro processo já teria passado pelo teste, permitindo que dois processos acessassem a área crítica ao mesmo tempo.

Questionário - 10

Arthur C. M. Barcella e Matheus P. Salazar