

Atividade 5 – Sistemas Operacionais (SO)

Júlia Simone Araújo

1.

Condições de disputa podem acontecer quando os processos concorrentes de um sistema têm acesso a recursos (memória, arquivos, conexão de rede etc.) compartilhados de forma não ordenada. Isso pode causar problemas de consistência no valor dos dados ou estado dos recursos acessados.

Um exemplo comum do mundo real é o de transação bancária. Ao querer depositar um valor em uma conta, a função (seção crítica) que faz parte de um sistema de controle de contas, pode ser chamada simultaneamente por inúmeros usuários. Ao levarmos em consideração o formato de chamada e execução de funções por uma máquina X86 e um ambiente concorrente, multitarefas e preemptivo, se dois usuários (t_1 e t_2) quiserem depositar valores em uma mesma conta ao mesmo tempo, pode ocorrer troca de contexto antes de cada operação ser finalizada, entrelaçando t_1 e t_2 . Isso pode gerar alteração no resultado esperado, gerando um resultado inadequado.

Esse tipo de problema pode inclusive permanecer por anos, por não ser identificado no código fonte, apenas na execução.

2.

a. Incorreta, pois só funciona em sistemas monoprocessados.

d. Incorreta, se baseiam na verdade pela ausência de ordenação entre as operações.

f. Incorreta, as instruções do tipo Test&Set são implementados pelo processador.

g. Incorreta, não há garantia tendo em vista que não há ordem no acesso a seção crítica.

h. Incorreta, na realidade esse tipo de algoritmo consome muito da CPU pelo teste contínuo.

i. Incorreta, atrasar os processos não garante a solução (exclusão mútua).

3.

Espera ocupada é um mecanismo que propõe uma tentativa de solução a condições de disputa em um sistema. Consiste em testar continuamente uma condição que indica se a seção crítica a ser acessada está livre ou ocupada. Caso esteja ocupada, a tarefa fica em loop até que a seção seja liberada. Acontece que pela atribuição do estado de ocupado e o teste da condição serem feitos em momentos distintos, isso pode levar a uma troca de contexto e possível entrelaçamento. O que torna o mecanismo ineficiente para o problema.

4.

Ao controlar o acesso a seções críticas, como: conexões de rede, buffers e descritores de tarefas.

5.

Como o teste de condição (linha 3) e a atribuição (linha 4) são feitos em momentos diferentes, pode ocorrer uma troca de contexto entre os momentos levando a condição de disputa e possível entrelaçamento. Ou seja, essas linhas do código são consideradas regiões críticas.