

1 - É uma situação crítica que pode ocorrer quando em um sistema você tem acesso a recursos compartilhados em outras palavras, uma condição de disputa ocorre quando dois processos acessam "simultaneamente" os dados compartilhados entre eles. Um exemplo seria um software que permite vários usuários utilizarem a mesma impressora, por ordem de solicitação e, em algum momento, uma das solicitações de impressão sobrepõe outra.

2 -

a - Falso, pois em máquinas multiprocessadas duas tarefas concorrentes podem ser executadas simultaneamente em processadores separados.

d - Falso, o que causa as condições de disputa é o acesso simultâneo ao mesmo dado, independentemente da velocidade dos processos.

f - Falso, estas instruções podem ser vistas em núcleos de SO, não são necessariamente soluções obrigatórias de um núcleo.

g - Falso, soluções despreocupadas não possuem garantia da ordem na acesso à seção crítica.

h - Falso, este tipo de algoritmo é de teste contínuo, ou seja, consomem tempo do processador sem necessidade.

3 - Consiste em testar continuamente uma condição que indica se a seção desejada está livre ou ocupada. Este tipo de solução é ineficiente para aplicações de nível de usuário, apesar de funcionarem, pois estas tarefas ficam testando continuamente, ou seja, consomem tempo do processador sem necessidade. A melhor forma de implementá-las seria suspendendo estas tarefas, assim que seu quantum de tempo acabar, até que a seção crítica solicitada seja liberada.

4 - Seu uso é inevitável durante a construção de estruturas de controle de concorrência, dentro do núcleo do sistema operacional.

5 - Haverá um acesso à região crítica por ambas as partes durante a troca de contexto, do valor da variável ocupado = 0 para ocupado = 1 (sistemas mono e multiprocessadores) ou, quando duas threads alterarem simultaneamente o valor 0 para 1 (Sistemas multiprocessadores)