

1) Explique o que são condições de disputa, mostrando um exemplo real. R= Condições de disputa são em alguns casos, o processo que compartilha a mesma área ou região, de processamento na memória principal. Ex.: Spool de impressão. Existem dois processos. Um gera um nome de arquivo para ser impresso em um Diretório de Spool, um outro processo, Daemons de impressão, verifica constantemente este diretório para checar arquivos a serem impressos. Encontrando arquivos ele os remove e imprime.

2) Letra C = II e III 1 - F 2 - V 3 - V 4 - F 5 - F (I) Em caso de sistemas com mais de um núcleo, torna-se necessário usar outros mecanismos de controle de concorrência, como operações TSL, para proteger a integridade interna do semáforo. (IV) Condições de disputa decorrem do acesso entrelaçado a um recurso Compartilhado (V) ?

3) Explique o que é espera ocupada e por que os mecanismos que empregam essa técnica são considerados ineficientes. R= Espera ocupada é um modelo de programação paralela que é caracterizado por testar repetidamente uma condição que impede o progresso de um processo a qual só pode ser alterada por outro processo. Esse modelo de programação tem a grande desvantagem em relação a perda de tempo no caso de monoprocessador, já que ele passa muito tempo testando condições, deixando assim, de estar realizando algum outro processamento.

4) Em que circunstâncias o uso de espera ocupada é inevitável? R: Por não haver garantia de ordem no acesso à seção crítica; dependendo da duração de quantum e da política de escalonamento, uma tarefa poderá entrar e sair da seção crítica várias vezes, antes que outras tarefas consigam acessá-la.

5) Letra C = II e V 1 - F 2 - V 3 - F 4 - F 5 - V (I) São na verdade implementadas pelo hardware (III)? (IV)?

6) -

7) Inicializado em 0: ? Inicializado em 1: proteger o acesso à área compartilhada Inicializado em $N > 1$: ?

8) Porque cabe ao programador garantir que a tarefa corrente vai liberar o mutex e não vai alterar o estado associado à variável de condição.

9) Mostre como pode ocorrer violação da condição de exclusão mútua se as operações down(s) e up(s) sobre semáforos não forem implementadas de forma atômica. R) As operações de acesso aos semáforos são geralmente implementadas pelo núcleo do sistema operacional, na forma de chamadas de sistema. É importante observar que a execução das operações Down(s) e Up(s) deve ser atômica, ou seja, não devem ocorrer acessos concorrentes às variáveis internas do semáforo, para evitar condições de disputa sobre as mesmas.