



Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
Professores: Valmir C. Barbosa e Felipe M. G. França
Assistente: Alexandre H. L. Porto

Quarto Período
AP1 - Segundo Semestre de 2011

Nome -

Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (1,5) Qual foi o objetivo principal da definição do conceito de multiprogramação, o qual passou a ser usado nas duas últimas gerações de computadores?

Resp.: Com o passar das gerações, a velocidade de processamento dos computadores se tornou cada vez maior. A velocidade dos dispositivos físicos também aumentou, mas muito mais lentamente do que a velocidade de processamento. Com isso, o tempo de ociosidade do processador quando o programa em execução fazia operações de E/S ficou cada vez maior, pois cada programa era executado até terminar, sem interrupções. Além disso, existia uma grande demora para se obter os resultados dos programas. Para evitar esses problemas, o conceito de multiprogramação, que permite que mais de um programa esteja em execução no sistema através da divisão do tempo de processamento entre os programas, foi definido a partir da terceira geração.

2. (2,5) Diga se as seguintes afirmativas são falsas ou verdadeiras. Para responder, escreva apenas F ou V para cada item em seu caderno de respostas.
 - (a) (0,5) Em uma árvore de processos, uma aresta ligando dois processos em níveis consecutivos da árvore significa que o processo do nível superior criou o processo do nível inferior.

Resp.: V (Verdadeira).

- (b) (0,5) Um *pipe* é usado para facilitar o gerenciamento de um dispositivo físico, pois podemos enviar dados para um dispositivo ao escrever nele e receber dados do dispositivo ao ler dele.

Resp.: F (Falsa, porque um *pipe* é usado para facilitar a troca de dados entre dois processos, ao conectar a saída de um deles com a entrada do outro. Isso permite que um processo envie dados para um outro processo com uma simples operação de escrita no *pipe* pelo processo que está enviando e uma operação de leitura do *pipe* pelo processo que está recebendo).

- (c) (0,5) Normalmente, para podermos acessar um sistema de arquivos diferente daquele com o sistema operacional, precisamos primeiramente montá-lo em algum diretório do sistema de arquivos com o sistema operacional.

Resp.: V (Verdadeira).

- (d) (0,5) Ao executar uma chamada ao sistema operacional, sempre precisamos mudar o processador do modo usuário para o modo supervisor, através da execução da instrução TRAP, porque o código que trata essa chamada executa no modo supervisor.

Resp.: V (Verdadeira).

- (e) (0,5) As bibliotecas devem sempre ser usadas ao fazermos as chamadas ao sistema operacional, porque somente nelas podemos executar a instrução TRAP necessária à execução de uma chamada.

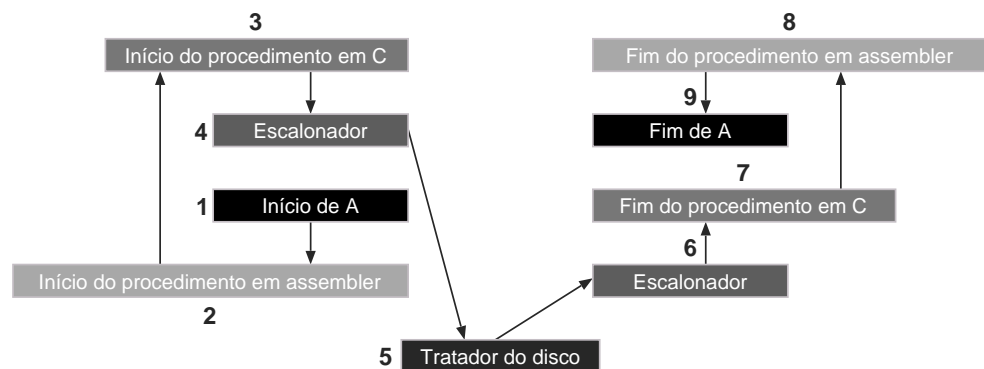
Resp.: F (Falsa, porque a instrução TRAP pode ser executada dentro do código do processo. A biblioteca é usada porque fornece uma interface bem melhor, muitas vezes até estendida, para uma chamada ao sistema operacional).

3. (1,5) Como o monitor de máquina virtual permite que diversos sistemas operacionais executem no mesmo computador sem interferirem uns com os outros?

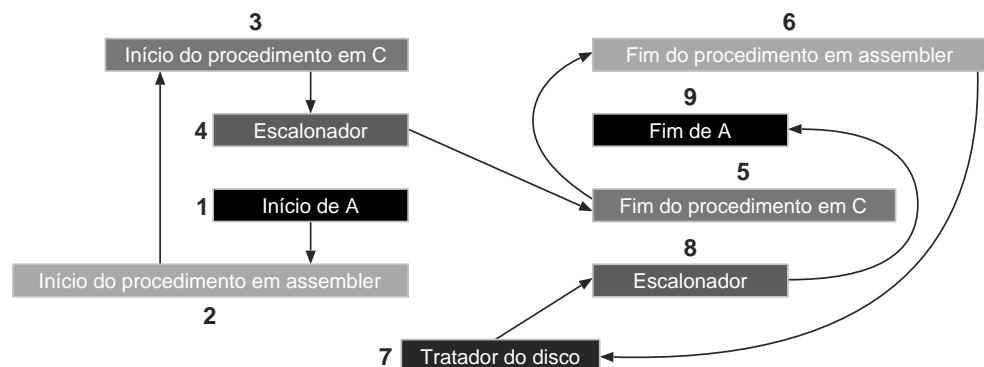
Resp.: Como vimos na aula 3, o monitor de máquina virtual cria máquinas virtuais, que são cópias exatas da máquina real, ou seja, não são máquinas estendidas, sendo portanto tão difíceis de serem usadas quanto a máquina real. Para que os sistemas operacionais não interfiram uns com os outros, basta executar cada sistema em uma máquina virtual diferente. Agora, como cada sistema operacional acha que está executando na máquina real, basta então o monitor garantir que as máquinas virtuais em execução compartilhem os recursos do hardware sem que ocorra algum tipo de condição de corrida que faça um dos

sistemas interferir no funcionamento de um outro sistema.

4. (1,5) Um aluno de sistemas operacionais disse que o diagrama a seguir representa, durante o tratamento de uma interrupção ocorrida durante a execução de um processo A, a correta ordem das ações executadas. O diagrama do aluno está correto? Se você achar que sim, basta dizer isso mas, se você achar que não, desenhe o diagrama que corretamente representa a ordem de execução das ações.



Resp.: O diagrama do aluno está errado. Na figura a seguir damos o diagrama correto. A seguir também damos uma breve descrição do novo diagrama, mas não é preciso tê-la dado na resposta. Note que o processo tratador do disco, executado devido à primeira chamada ao escalonador, somente é executado após o tratamento da interrupção ser finalizado. Já a segunda chamada ao escalonador faz com que A imediatamente inicie a sua execução no ponto em que parou anteriormente (note que, nesse caso, A precisa ser o processo mais prioritário).



5. (1,5) Considere um semáforo binário usado para garantir a exclusão mútua ao acessar um recurso compartilhado. O que ocorrerá se um processo executar somente a operação **P** imediatamente antes de entrar na sua seção crítica e não a operação **V** ao término? E se o processo executar somente a operação **V** imediatamente depois de sair da sua seção crítica e não a operação **P** no início?

Resp.: -Se a operação **V** não for executada depois da operação **P**, então mais nenhum processo poderá executar a sua seção crítica, já que o acesso ao recurso em questão não foi liberado. Assim, qualquer processo que executar, no futuro, a operação **P** para tentar acessar o recurso compartilhado, ficará bloqueado para sempre, a não ser que um outro processo cometa o erro descrito na resposta dada a seguir para a segunda pergunta da questão.

-Se a operação **P** não for executada antes da operação **V**, então o processo acessará o recurso compartilhado sem garantir o acesso exclusivo a ele. Isso poderá gerar resultados incorretos na computação devido a possíveis condições de corrida, já que esse acesso poderá ocorrer concorrentemente com o de outro processo.

6. (1,5) Suponha que o sistema operacional use o algoritmo *round robin*, com um quantum de 1 unidade de tempo, ao escalonar os processos, e que a ordem de execução dos três únicos processos, A, B e C, até terminarem tenha sido ABCABCACACAA. Suponha agora que o sistema operacional tenha passado a usar o algoritmo de prioridades, sendo cada prioridade reduzida em 1 unidade a cada 2 unidades de tempo,

e sendo que cada processo executa até a sua prioridade deixar de ser a maior. Quais serão os novos tempos de término dos procesos se a prioridade inicial de cada processo for igual ao número de unidades de tempo que ele precisa executar até terminar? Justifique a sua resposta.

Resp.: Pela ordem de execução, vemos que o processo A executou por 6 unidades de tempo, o processo B por 2 unidades de tempo, e o processo C por 4 unidades de tempo. Agora, como a prioridade inicial de cada processo é igual ao seu tempo execução, então as prioridades dos processos A, B e C são de, respectivamente, 6, 2 e 4. A tabela dada a seguir mostra a ordem de execução se agora usarmos o algoritmo por prioridades, sendo que cada processo executa até terminar ou até algum outro processo possuir prioridade maior, e supondo que a prioridade é reduzida de 1 unidade a cada 2 unidades de tempo. Nessa tabela, a primeira linha mostra unidades de tempo. A segunda linha mostra os processos executados em cada unidade de tempo. Finalmente, a terceira linha mostra, para o processo de cada coluna, a prioridade do processo antes de ele executar na unidade de tempo correspondente. Como podemos ver pela tabela, os novos tempos de término dos processos A, B e C serão de, respectivamente, 6, 10 e 12 unidades de tempo.

0	2	4	6	8	10
A	A	A	C	C	B
6	5	4	4	3	2