



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
AP3 - Primeiro Semestre de 2008

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. (2.0) Quando um processo executa uma chamada ao sistema operacional, o processador passa do modo usuário para o modo supervisor antes de a chamada ser tratada. Por que isso é feito? Justifique a sua resposta.

Resp.: Como vimos nas aulas introdutórias da disciplina, uma das principais funções do sistema operacional é a de gerenciar os componentes do hardware do computador. Para poder executar esta função, o núcleo do sistema operacional precisa executar no modo supervisor, pois somente neste modo pode-se acessar diretamente o hardware. Também vimos que as chamadas ao sistema operacional, que são executadas pelo núcleo do sistema, existem para que os processos em execução possam requisitar serviços ao sistema. Logo, o processador deverá mudar do modo usuário para o modo supervisor, usando a instrução TRAP, ao executarmos uma chamada ao sistema. Note que isso ocorrerá não somente porque o núcleo executa no modo supervisor, como também porque alguns dos serviços executados pelas chamadas podem precisar do acesso direto a algum dos componentes do hardware.

2. (2.0) Suponha que dois processos, A e B, compartilham uma região de memória R que funciona como um contador. O processo B incrementa o valor deste contador em uma unidade. O processo A decrementa o contador em uma unidade (se o contador tem valor 0, o processo espera que esse valor aumente para efetuar o decremento). Como deveremos usar os semáforos para garantir a exclusão mútua ao acessarmos a região de memória R , e para garantirmos o correto funcionamento dos processos A e B?

Resp.: Deveremos usar dois semáforos binários: o semáforo *bloqueio*, usado para bloquear o processo A caso o valor do contador dado em R seja igual a 0, e o semáforo *acesso*, usado para garantir o acesso exclusivo à região R . Como o processo A somente executará a operação **P** sobre *bloqueio* quando o valor do contador for 0, então o valor inicial deste semáforo deverá ser 0. Já o valor inicial do semáforo *acesso* será 1, pois inicialmente A e B não estarão acessando R . O procedimento *IncrementaContador* dado a seguir deverá ser usado pelo processo B

para incrementar o valor do contador de R em 1 unidade. Note que este procedimento executa a operação **V** sobre *bloqueio* somente se o valor do contador era 0 antes do incremento, pois A somente bloquearia se o valor do contador fosse 0. Já o procedimento *DecrementaContador*, também dado a seguir, deverá ser usado pelo processo A para decrementar o valor do contador de R em 1 unidade. Note que o valor do contador será maior do 0 depois de executarmos a operação **P** sobre *bloqueio*. Isso ocorrerá porque se A não ficar bloqueado em *bloqueio*, então B executou o procedimento *IncrementaContador*, que executou a operação **V** sobre *bloqueio* e incrementou o contador. Note também que A deverá executar a operação **V** sobre *acesso* antes de bloquear, e novamente executar a operação **P** sobre *acesso* após desbloquear, para garantir que B terá uma chance de incrementar o contador, e que A terá acesso exclusivo ao contador após desbloquear.

<pre> void IncrementaContador(void) { P(acesso); if (R == 0) V(bloqueio); R = R + 1; V(acesso); } </pre>	<pre> void DecrementaContador(void) { P(acesso); if (R == 0) { V(acesso); P(bloqueio); P(acesso); } R = R - 1; V(acesso); } </pre>
--	--

3. (2.0) Suponha que dois processos, A e B, compartilham três recursos: uma impressora, um disco e uma unidade de fita magnética. Suponha ainda que o processo A obteve a impressora e o processo B obteve a unidade de fita magnética. Se os processos A e B desejarem obter um dos recursos que não possuem, sem liberar os recursos alocados a eles, que escolhas de A e B gerariam um impasse?

Resp.: Como o disco é um recurso preemptivo e a impressora e a unidade de fita são recursos não-preemptivos, então não ocorrerá um

impasse se A ou B desejarem o disco. Vamos então supor que A deseja a unidade de fita e que B deseja a impressora. Como os processos A e B serão bloqueados esperando por, respectivamente, a unidade de fita e a impressora, sem liberarem os recursos que possuem, então teremos um impasse. Ele ocorrerá porque A e B estarão ambos bloqueados, e porque A somente será desbloqueado quando B liberar a unidade de fita e B somente será desbloqueado quando A liberar a impressora.

4. (2.0) Suponha que o computador tem uma memória física de 512MB, dividida em molduras de página com 16KB de tamanho. Se pudermos armazenar 1/8 das páginas virtuais nas molduras de página, como será dividido o endereço virtual usando uma tabela de páginas de dois níveis com mesmo tamanho em cada nível?

Resp:. Como a memória tem 512MB, isto é, 2^{29} bytes, e como o tamanho de cada moldura de página é de 16KB, ou seja, 2^{14} bytes, então teremos $2^{29}/2^{14} = 2^{15}$ molduras de página (isto é, 32768 molduras). Agora, como poderemos armazenar até 1/8 das páginas virtuais nas molduras de página, o número de páginas virtuais será de $8 \times 2^{15} = 2^3 \times 2^{15} = 2^{18}$ (ou seja, 262144 páginas virtuais). Logo, usaremos os 18 bits superiores do endereço virtual para representar a página virtual. Agora, como o sistema operacional usa uma tabela de páginas com dois níveis cujas tabelas possuem o mesmo tamanho, então este campo de 18 bits será dividido em dois campos de 9 bits. Os 9 bits superiores deste campo definirão a entrada da tabela de páginas do primeiro nível, e os 9 bits inferiores definirão a entrada da tabela de páginas do segundo nível (que foi definida pela entrada escolhida da tabela do primeiro nível). Finalmente, como o tamanho da página virtual é o mesmo tamanho da moldura de página (que é igual a 2^{14} bytes), então usaremos os 14 bits inferiores do endereço virtual para representar o deslocamento dentro da página.

5. (2.0) Suponha que o computador tem um disco de 2GB com blocos com tamanho de 32KB. Qual será a memória necessária para gerenciar os blocos do disco se o sistema operacional usar uma lista encadeada baseada em um índice ao gerenciar os blocos usados, e um mapa de

bits ao gerenciar os blocos livres?

Resp.: Como o disco tem 2GB de tamanho, isto é, 2^{31} bytes, e como cada bloco tem 32KB de tamanho, ou seja, 2^{15} bytes, então teremos $2^{31}/2^{15} = 2^{16} = 65536$ blocos no disco. O espaço gasto pelo mapa de bits, usado para gerenciar os blocos livres, será de 65536 bits, ou seja, 8192 bytes, pois cada bloco será representado no mapa por 1 bit. Já a tabela da alocação baseada em um índice, usada para gerenciar os blocos usados, possuirá uma entrada para cada bloco, isto é, 65536 entradas. Agora, como cada entrada deverá ter 2 bytes de tamanho (pois ela deverá armazenar o número de um dos 65536 blocos do disco), então gastaremos $65536 \times 2 = 131072$ bytes para armazenar esta tabela. Logo, o número total de bytes usados para armazenar o mapa de bits e a tabela será de $8192 + 131072$, ou seja, 139264 bytes.