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
Gabarito da AD2 - Primeiro Semestre de 2018

Atenção: Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um, 1/3 dos pontos daquela questão.

Nome -
Assinatura -

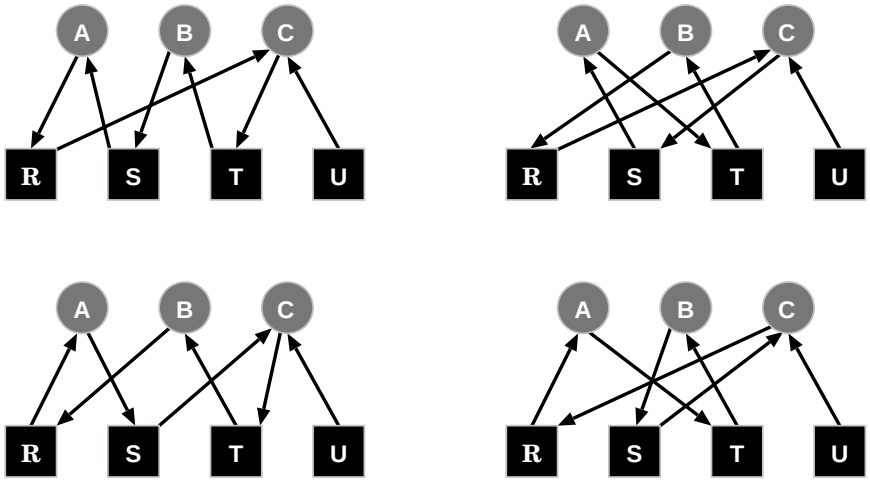
---

1. (1,5) Suponha que um processo A possua dois recursos não-preemptivos, R e S, que um processo B possua um recurso não-preemptivo, T, e que o processo B, junto com um outro processo C, possuam um recurso preemptivo e compartilhável, U. Responda, justificando a sua resposta:

   (a) (1,0) Como podemos gerar um impasse envolvendo todos os processos?

   Resp.: Como C possui somente U, que não pode pertencer a um impasse porque é preemptivo, é necessário primeiramente esperar

A, que possui dois recursos não-preemptivos R e S, liberar um deles. Se o recurso liberado e depois alocado a C for R, então teremos as possibilidades dadas nos grafos na parte superior da figura a seguir; se for S, teremos as possibilidades dadas nos grafos na parte inferior da figura. Para a sua resposta ser considerada correta, basta fornecer um dos grafos.



(b) (0,5) Quantos dos recursos não-preemptivos precisariam ser preemptivos para que os impasses nunca ocorressem?

Resp.: Para evitar completamente os impasses, somente um recurso pode ser não-preemptivo, porque com dois (por exemplo, se A tem R, B tem T, A solicita T e B solicita R) ou três (como vimos na resposta do item anterior) podemos ter um impasse. Como três recursos são não-preemptivos, então dois deles devem passar a ser preemptivos.

2. (1,5) Considere o seguinte mapeamento entre endereços virtuais e endereços físicos em um sistema onde o tamanho da página é 16KB. Considere, ainda, que o endereço virtual tem 18 bits e o físico tem 17 bits. Responda, justificando a sua resposta:

Endereço virtual	Endereço físico
250052	4292
77881	127033
1234	99538
234567	87111

(a) (0,5) Como são divididos o endereço virtual e o endereço físico?

Resp.: Como cada página virtual e cada moldura de página tem o mesmo tamanho de 16KB, então 14 bits são necessários para representar o deslocamento nos endereços virtuais e físicos, pois $2^{14} = 16384 = 16K$. Logo, o endereço virtual de 18 bits é dividido no campo número da página virtual com os 4 bits superiores do endereço e o campo deslocamento com os 14 bits inferiores do endereço. Similarmente, o endereço físico de 17 bits é dividido no campo número da moldura de página com os 3 bits superiores do endereço e o campo deslocamento com os 14 bits inferiores do endereço.

(b) (1,0) Dados os endereços virtuais 45311, 8999 e 186432, quais deles estão mapeados em molduras já ocupadas? Para estes, quais são os endereços físicos?

Resp.: A seguir damos uma versão estendida da tabela do enunciado em que mostramos também, usando a divisão dos endereços virtuais e físicos em binário, o número da página virtual, o deslocamento (mostramos o deslocamento somente uma vez porque, como sabemos, ele dentro da página virtual é igual a dentro da moldura de página), e o número da moldura de página. Além disso mostramos, nas últimas três linhas, separadas, o número da página virtual e o deslocamento para cada endereço dado no enunciado e, caso essa página esteja mapeada, ou seja, apareça em uma das linhas originais da tabela, o número da moldura em que ela está mapeada e o endereço físico no qual o endereço virtual está mapeado. Pela tabela, vemos que somente o endereço virtual 8999 está mapeado no endereço físico 107303.

Endereço virtual	Página virtual	Deslocamento	Endereço físico	Moldura de página
250052	1111	01000011000100	4292	000
77881	0100	11000000111001	127033	111
1234	0000	00010011010010	99538	110
234567	1110	01010001000111	87111	101
45311	0010	11000011111111	—	—
8999	0000	10001100100111	107303	110
186432	1011	01100001000000	—	—

3. (2,0) Suponha que um processo acesse, em ordem, as páginas virtuais 1, 2, 3, 0, 0, 2, 2, 1, 3, 1, 0 e 2. Se o algoritmo FIFO for usado, quantas falhas de página ocorrerão e que página será substituída quando cada falha ocorrer? E se o algoritmo LRU for usado? Justifique a sua resposta.

   Resp.: Como foi informado em uma mensagem do fórum na plataforma, foram alocadas três molduras, inicialmente vazias, ao processo. Primeiramente então vamos mostrar, na tabela a seguir, a sequência de acessos às páginas virtuais dadas na questão para o algoritmo FIFO. Como vimos na aula 9, no algoritmo FIFO, as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo da sua cópia para a memória. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página copiada há mais tempo para a memória. Na tabela dada a seguir mostramos, em cada linha, o que ocorre ao acessarmos as páginas na ordem dada no enunciado. Para cada uma dessas linhas mostramos na primeira coluna a página que é acessada, na segunda coluna a ordem em que as páginas devem ser escolhidas e, na última coluna, se o acesso à página gerou um não uma falha de página. Como podemos ver pela tabela, ocorrem 6 falhas de página quando o algoritmo FIFO é usado.

Páginas	Ordenação			Ocorreu uma falha?
1	1			Sim
2	1	2		Sim
3	1	2	3	Sim
0	2	3	0	Sim
0	2	3	0	Não
2	2	3	0	Não
2	2	3	0	Não
1	3	0	1	Sim
3	3	0	1	Não
1	3	0	1	Não
0	3	0	1	Não
2	0	1	2	Sim

- Agora vamos mostrar, na tabela a seguir, a sequência de acessos às páginas virtuais dadas na questão para o algoritmo LRU. Como vimos na aula 9, no algoritmo LRU, as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo do seu último acesso. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página não acessada há mais tempo. A tabela a seguir tem a mesma estrutura da anterior, mas agora mostra, na segunda coluna, a ordem de escolha segundo o algoritmo LRU. Como podemos ver pela tabela, ocorrem 8 falhas quando o algoritmo LRU é usado, implicando portanto que o número da falhas aumenta em 2.

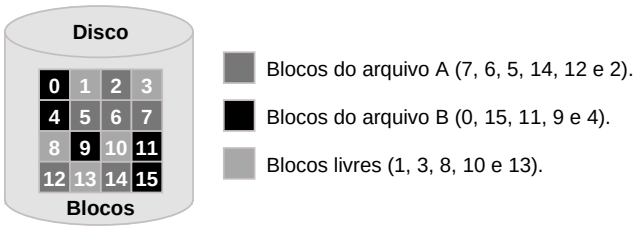
Páginas	Ordenação			Ocorreu uma falha?
1	1			Sim
2	1	2		Sim
3	1	2	3	Sim
0	2	3	0	Sim
0	2	3	0	Não
2	3	0	2	Não
2	3	0	2	Não
1	0	2	1	Sim
3	2	1	3	Sim
1	2	3	1	Não
0	3	1	0	Sim
2	1	0	2	Sim

4. (2,0) Suponha que um computador use a segmentação com paginação ao gerenciar a memória física do computador, de 4GB, e que os processos A, B, C, D e E, com tamanhos de, respectivamente, 256MB, 64KB, 2GB, 1024KB e 512MB, sejam os únicos em execução. Se o tamanho da página virtual for de 2^aKB, qual deverá ser o valor máximo de a que permitirá armazenar integralmente todos os cinco processos na memória do computador? Justifique a sua resposta.

   Resp.: Primeiramente, vamos converter os tamanhos dos processos para os equivalentes em KB, usando potências de 2. Os tamanhos convertidos dos processos A, B, C, D e E são, respectivamente, 2^{18}KB, 2^6KB, 2^{21}KB, 2^{10}KB e 2^{19}KB. Como o tamanho da página pode ser no mínimo 1 byte e no máximo igual ao tamanho da memória de 4GB = 2^{22}KB, então o valor de a pode variar de -10 (valores negativos permitem páginas com tamanhos menores do que 1KB) até 22. Note que se conseguirmos armazenar todos os processos para um valor a, então poderemos armazenar todos os processos para todos os possíveis valores menores do que a, porque quando a é reduzido em 1 unidade, o número total de páginas virtuais dobra e, no pior caso, o número de páginas usadas por cada processo também dobra (o número pode ser igual ao dobro menos 1, caso o processo use menos da metade da última página alocada a ele). Note também que precisaremos de pelo

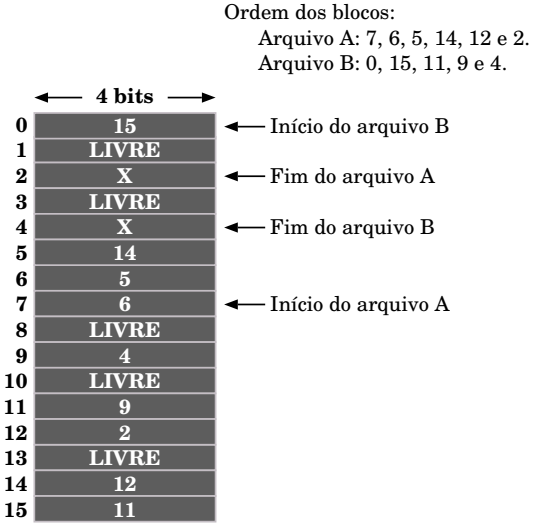
menos 5 páginas para armazenar os processos, porque cada processo ocupa no mínimo 1 página. Logo, a não poderá ser igual a 22, 21 ou 20, porque o número total de páginas para esses valores é de, respectivamente, 1, 2 e 4. Agora, para $a = 19$, ou seja para uma página com 2^{19}KB ou 512MB, teremos 8 páginas virtuais, e os processos A, B, C, D e E usarão, respectivamente, 1, 1, 4, 1 e 1 páginas, ou seja, também 8 páginas no total. Portanto, podemos concluir que o valor máximo de a é 19.

5. (1,5) Considere a alocação do disco dada na figura a seguir. Responda, justificando a sua resposta:



   (a) (1,0) Como será a tabela da alocação por lista encadeada utilizando um índice para a alocação dada na figura?

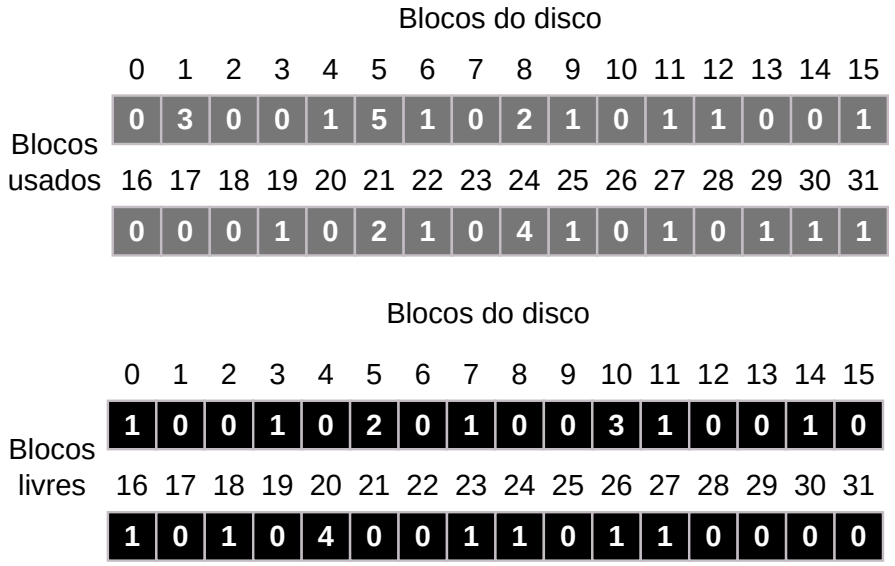
   Resp.: Se a técnica de alocação por lista encadeada utilizando um índice for usada, vamos obter a tabela dada na figura a seguir. Note que ela tem 16 entradas, referenciadas pelos endereços dos blocos, pois temos 16 blocos no disco, numerados de 0 até 15. Nessa tabela, um "X" na entrada indica que o bloco associado a ela é o último bloco do arquivo.



(b) (0,5) Se a alocação contígua passar a ser usada, quantos blocos um novo arquivo C poderá ter se o arquivo A for armazenado no ínicio do disco e o arquivo B no final do disco?

Resp.: Como vimos na aula 11, quando a alocação contígua é usada, um arquivo é armazenado em blocos consecutivos do disco. Pelo enunciado, vemos que o número de blocos usados pelos arquivos A e B é, respectivamente, 6 e 5. Como o arquivo A será agora armazenado nos blocos de 0 até 5 e o arquivo B nos blocos de 11 até 15, e como isso implicará em que os blocos de 6 até 10 estarão livres, então C poderá ter até 5 blocos.

6. (1,5) Suponha que, depois de uma verificação da consistência do sistema de arquivos, o sistema operacional tenha criado as tabelas dadas a seguir, as quais, para cada bloco, indicam quantas vezes ele foi encontrado em um arquivo (primeira tabela) e quantas vezes ele foi contado como sendo livre (segunda tabela). Responda:



(a) (1,0) Caso existam problemas de consistência no sistema de arquivos, indique quais são eles.

Resp.: Como, em um sistema de arquivos consistente, cada bloco do disco ou está livre ou está alocado a um único arquivo, então não poderiam existir entradas nas tabelas com valores maiores do que 1 e, além disso, para cada bloco, se a entrada em uma tabela fosse igual a 1, a entrada na outra tabela deveria ser igual a 0. Logo, o sistema de arquivos está inconsistente, porque as tabelas da figura do enunciado não satisfazem as condições descritas. O primeiro erro ocorre com os blocos 5, 11, 24 e 27 que, além de estarem alocados a pelo menos um arquivo, também estão marcados como livres pelo menos uma vez, o que pode, em consequência, fazer com que sejam alocados a mais arquivos. O segundo erro ocorre com os blocos 1, 8 e 21, pois eles estão alocados a mais de um arquivo. O terceiro erro, que ocorre com os blocos 10 e 20, está no fato de eles estarem marcados mais de uma vez como livres, o que permite que possam ser alocados a mais de um arquivo. Finalmente, os blocos 2, 13, 17 e 28 são blocos ausentes porque, segundo as tabelas, nem estão marcados como livres nem associados a um arquivo.

(b) (0,5) Quais desses problemas poderiam ter sido evitados se uma lista encadeada, similar à descrita na aula 12, tivesse sido usada para gerenciar os blocos livres do disco?

Resp.: Somente podemos evitar que cada bloco seja marcado mais de uma vez como livre quando garantimos que a lista está corretamente implementada, ou seja, quando cada bloco livre do disco está armazenado em somente uma entrada da lista e além disso não existe nenhum bloco usado armazenado na lista. Logo, além de os blocos 2, 13, 17 e 28 deixarem de ser ausentes, os blocos 10 e 20 poderiam ser alocados a somente um arquivo, e os blocos 5, 11, 24 e 27 não poderiam ser alocados a mais arquivos.