



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

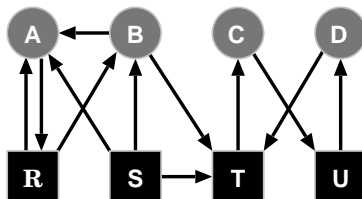
Quarto Período
Gabarito da AD2 - Segundo Semestre de 2014

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.

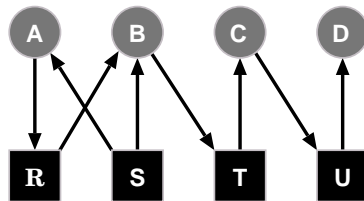
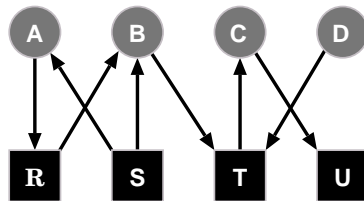
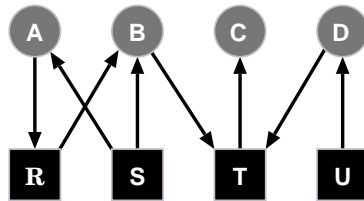
Nome -

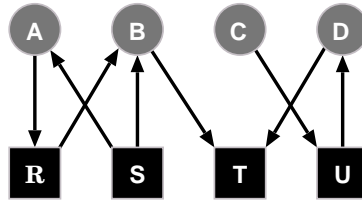
Assinatura -

-
1. (1,0) Removendo somente quatro arestas, modifique o grafo dado a seguir de modo a que ele seja um grafo de recursos correto e sem impasses, supondo que os recursos R, T e U são não-preemptivos e que o recurso S é preemptivo.



Resp.: Note que precisamos remover obrigatoriamente as arestas orientadas de B para A e de S para T, pois no grafo de recursos não existem arestas entre dois vértices que representem processos ou entre dois vértices que representem recursos. A terceira aresta que precisa ser removida é uma das arestas do ciclo C-U-D-T-C, pois somente assim o impasse envolvendo os recursos T e U e os processos C e D não ocorrerá. Finalmente, a última aresta a ser removida é a que indica que o recurso R foi alocado ao processo A, porque um recurso não pode simultaneamente pertencer a um processo e estar sendo requisitado por ele. Note que a remoção da aresta orientada de A para R implicaria na remoção também da aresta orientada de R para B, porque um recurso não-preemptivo não pode estar alocado a mais de um processo. Com isso, 5 arestas seriam removidas. Logo, o grafo correto é qualquer um dos seguintes:





2. (2,0) Considere um computador com uma memória virtual de 256KB, com páginas virtuais de 16KB de tamanho. Considere ainda que a memória física tenha 50% do tamanho da memória virtual. Como será preenchida a tabela de páginas se os endereços virtuais 123456, 262143, 8192, 173716 e 127033 forem mapeados, respectivamente, nos endereços físicos 74304, 16383, 106496, 59028 e 77881? Na sua resposta, use o mesmo estilo usado no *slide* 20 da aula 8. Suponha que, inicialmente, todas as molduras estão vazias.

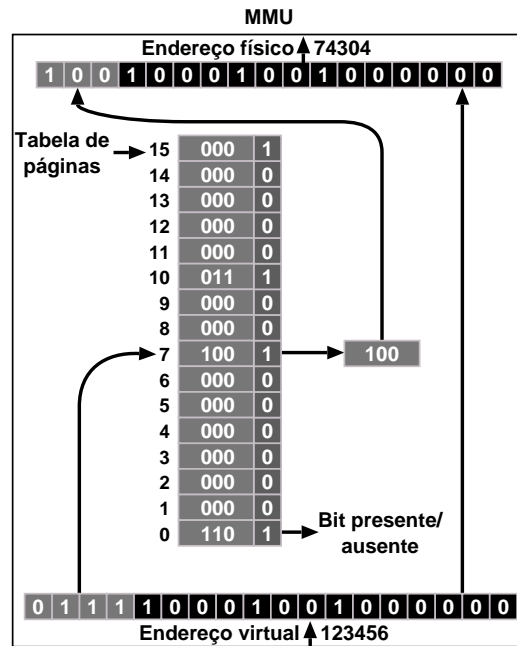
Resp.: Pela questão, vemos que a memória virtual, com 256KB, requer endereços de 18 bits (pois $2^{18} = 256K$), e que ela é dividida em 16 páginas virtuais de 16KB. Assim, os 4 bits superiores (pois $2^4 = 16$) do endereço virtual definem o número da página e os 14 bits inferiores (pois $2^{14} = 16K$) o deslocamento dentro dessa página. Também pela questão, vemos que a memória física tem 128KB (50% ou 1/2 do tamanho da memória virtual), requerendo endereços de 17 bits (pois $2^{17} = 128K$). Como o tamanho da moldura de página é igual ao da página virtual, então a memória física é dividida em 8 molduras de página. Assim, os 3 bits superiores do endereço físico indicam o número da moldura (pois $2^3 = 8$) e os 14 bits inferiores o deslocamento dentro dessa moldura. Logo, pelo mapeamento dado na questão, observamos o seguinte:

- O endereço virtual 123456, que em binário é 011110001001000000, estar mapeado no endereço físico 74304, que em binário é 10010001-001000000, significa que a página virtual 0111 em binário ou 7 em decimal está mapeada na moldura de página 100 em binário ou 4 em decimal.
- O endereço virtual 262143, que em binário é 11111111111111111, estar mapeado no endereço físico 16383, que em binário é 00011111-111111111, significa que a página virtual 1111 em binário ou 15

em decimal está mapeada na moldura de página 000 em binário ou 0 em decimal.

- O endereço virtual 8192, que em binário é 0000100000000000000, estar mapeado no endereço físico 106496, que em binário é 1101000-0000000000, significa que a página virtual 0000 em binário ou 0 em decimal está mapeada na moldura de página 110 em binário ou 6 em decimal.
- O endereço virtual 173716, que em binário é 101010011010010100, estar mapeado no endereço físico 59028, que em binário é 01110011-010010100, significa que a página virtual 1010 em binário ou 10 em decimal está mapeada na moldura de página 011 em binário ou 3 em decimal.
- O endereço virtual 127033, que em binário é 011111000000111001, estar mapeado no endereço físico 14262, que em binário é 10011000-000111001, significa que a página virtual 0111 em binário ou 7 em decimal está mapeada na moldura de página 100 em binário ou 4 em decimal. Note que o mapeamento deste endereço representa o mesmo mapeamento do endereço 123456.

Logo, a tabela de páginas é a dada na figura a seguir, lembrando que o bit presente/ausente é igual a 1 para as páginas virtuais mapeadas em alguma moldura de página e 0 para as páginas não mapeadas. Nessa figura mostramos, como ilustração, a conversão do endereço virtual 123456 para o endereço físico 74304.

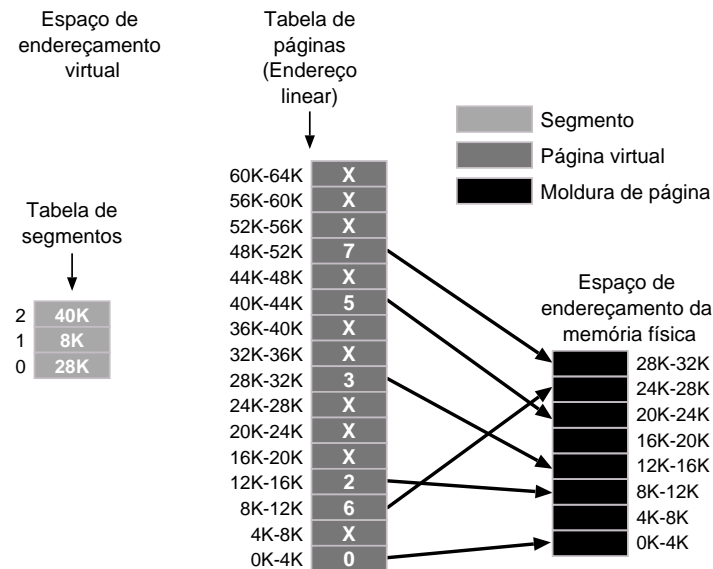


3. (2,0) Suponha que um processo tenha acessado, em ordem, as páginas virtuais 1, 2, 3, 2, 2, 1, 3, 3, 2 e 1, e que tenham sido alocadas 4 molduras, inicialmente vazias, para o processo. Suponha ainda que, em todos os acessos, a página virtual 1 tenha sido acessada somente para escrita e as outras somente para leitura. Se o bit **refenciada** de uma página for ligado quando ela for copiada para a memória e permanecer ligado até outra página ser acessada, quantas falhas de página serão geradas pelos acessos à página 1 se o algoritmo NRU for usado ao substituir as páginas? E quantas falhas de página serão geradas pela página 2 com o algoritmo LRU? Justifique a sua resposta.

Resp.: Como o número de molduras é superior ao número de páginas acessadas, o número de falhas geradas por acessos a qualquer das três páginas acessadas será sempre 1, independentemente do algoritmo utilizado.

4. (1,5) Suponha que um computador utilize uma técnica de segmentação com paginação similar à do processador Pentium da Intel. Na figura a seguir mostramos, na tabela de segmentos, o endereço linear inicial no

qual cada segmento usado por um processo é mapeado. Nesta figura também mostramos, na tabela de páginas do espaço de endereçamento linear, como as páginas virtuais são mapeadas nas molduras de página da memória física. Usando essas tabelas, forneça o endereço linear e o endereço físico de cada um dos seguintes endereços virtuais bidimensionais (segmento, endereço dentro do segmento), usando todos as possíveis molduras livres caso uma falha de páginas ocorra:



- (a) (0,5) (0,1234).
 (b) (0,5) (1,12021).
 (c) (0,5) (2,0).

Resp.: No Pentium, cada segmento é mapeado a partir do endereço inicial dado na tabela de segmentos. Todos os endereços iniciais dos segmentos pertencem a um espaço de endereçamento comum, chamado de **espaço de endereçamento linear**, que é paginado do mesmo modo que o espaço de endereçamento virtual visto na aula 9. Logo, o endereço linear correspondente a um endereço de um segmento é igual à soma deste endereço com o endereço linear a partir do qual o segmento é mapeado. Pela figura vemos que o espaço de endereçamento linear tem 64K, com endereços de 16 bits variando de 0 até 65535. Já

o espaço de endereçamento físico tem 32K, com endereços de 15 bits variando de 0 até 32767. Como as páginas possuem 4K de tamanho, o espaço de endereçamento linear é dividido em 16 páginas virtuais, e o espaço de endereçamento físico é dividido em 8 molduras de página. Assim, cada endereço linear é dividido do seguinte modo: 4 bits para o número da página e 12 bits para o deslocamento. No que segue, cada endereço linear também é dado em binário, separando o número da página e o deslocamento por “|”.

- (a) Endereço virtual (0, 1234): como o segmento 0 é, segundo a tabela, mapeado para o endereço linear 28K ou 28672, então o endereço linear correspondente ao endereço virtual é 28672 mais 1234, ou seja, é igual a 29906. Agora, para o endereço linear 29906 (0111 | 010011010010), vemos que estamos acessando a palavra 1234 da página virtual 7 (com endereços de 28672 até 32767). Pela figura, vemos que a página virtual 7 está mapeada na moldura de página 3 (com endereços de 12288 até 16383). Logo, o endereço físico é 12288 (o primeiro endereço da moldura) mais 1234 (o deslocamento), isto é, é 13522.
- (b) Endereço virtual (1, 12021): já para o segmento 1, que segundo a tabela está mapeado no endereço linear 8K ou 8192, o endereço linear correspondente ao endereço virtual é 8192 mais 12021, ou seja, é igual a 20213. No caso do endereço linear 20213 (0100 | 111011110101), estamos acessando a palavra 3829 da página virtual 4 (com endereços de 16384 até 20479). Como a página virtual 4 não está mapeada na memória, e como as molduras de página 1 e 4 estão livres, então temos dois possíveis mapeamentos, de acordo com a moldura escolhida: se a moldura 1 é a escolhida (com endereços de 4096 até 8191), então o endereço físico é 4096 (o primeiro endereço da moldura) mais 3829 (o deslocamento), isto é, 7925; já se a moldura 4 é a escolhida, então o endereço físico é também 20213, pois o número da moldura é igual ao da página virtual.
- (c) Endereço virtual (2, 0): no caso do segmento 2, que segundo a tabela de segmentos é mapeado a partir do endereço linear 40K ou 40960, o endereço linear correspondente ao endereço virtual é também 40960 (pois a segunda componente do endereço virtual é

0). Esse endereço linear 40960 (1010 | 000000000000) está na palavra 0 da página virtual 10 (com endereços de 40960 até 45055), que está mapeada na moldura de página 5 (com endereços de 20480 até 24575). Logo, o endereço físico é 20480, igual ao primeiro endereço da moldura, pois o descolamento é 0.

5. (1,5) Imagine que você deseje acessar as posições do arquivo prova.pdf dado na figura a seguir. Quantas vezes cada um dos blocos será acessado ao usarmos o acesso sequencial, se desejarmos:

Arquivo prova.pdf (12 posições)

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

- (a) (0,5) acessar os 7 primeiros blocos em ordem decrescente e os 5 últimos blocos em ordem crescente?
- (b) (0,5) acessar todos os blocos ímpares e depois todos os blocos pares?
- (c) (0,5) acessar os blocos na seguinte ordem: 1, 10, 3, 4, 12, 9, 7, 8, 5, 11, 6 e 2?

Resp.: Quando usamos o acesso aleatório, as posições do arquivo podem ser acessadas independentemente umas das outras. Isso não ocorre no acesso sequencial, em que o acesso a uma posição x do arquivo pode exigir que acessemos outras posições também. Se a posição x foi a primeira a ser acessada ou se a posição anteriormente acessada se localiza no arquivo após x , precisamos acessar todas as posições até x , inclusive. Se a posição anteriormente acessada está antes de x , precisamos acessar, além de x , todas as posições entre a posição anterior e x . As respostas às questões são dadas em cada uma das tabelas a seguir. Cada tabela mostra as posições que precisam ser acessadas (indicadas por um “X”) para cada x . A última linha de cada tabela mostra, para cada posição, o número total de acessos.

(a) 7 primeiras em ordem decrescente e depois
e as 5 últimas em ordem crescente

x	Precisam ser acessadas											
	1	2	3	4	5	6	7	8	9	10	11	12
7	X	X	X	X	X	X	X					
6	X	X	X	X	X	X						
5	X	X	X	X	X							
4	X	X	X	X								
3	X	X	X									
2	X	X										
1	X											
8		X	X	X	X	X	X	X				
9									X			
10										X		
11											X	
12												X
Total	7	7	6	5	4	3	2	1	1	1	1	1

(b) todos os blocos ímpares e depois
todos os blocos pares

x	Precisam ser acessadas											
	1	2	3	4	5	6	7	8	9	10	11	12
1	X											
3		X	X									
5				X	X							
7						X	X					
9								X	X			
11										X	X	
2	X	X										
4			X	X								
6					X	X						
8							X	X				
10									X	X		
12											X	X
Total	2	2	2	2	2	2	2	2	2	2	2	1

(c) as posições 1, 10, 3, 4, 12, 9,
7, 8, 5, 11, 6 e 2

x	Precisam ser acessadas											
	1	2	3	4	5	6	7	8	9	10	11	12
1	X											
10		X	X	X	X	X	X	X	X	X		
3	X	X	X									
4				X								
12					X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X			
7	X	X	X	X	X	X	X					
8								X				
5	X	X	X	X	X							
11						X	X	X	X	X	X	
6	X	X	X	X	X	X						
2	X	X										
Total	7	7	6	6	6	6	5	5	4	3	2	1

6. (2,0) Suponha que o sistema operacional use um mapa de bits para gerenciar os blocos livres de um disco com n blocos e que a alocação contígua seja usada ao gerenciar os blocos usados. Suponha ainda que dois arquivos, A e B, estejam armazenados no disco e que tenham, respectivamente, a e b blocos. Responda, justificando a sua resposta:

- (a) (1,0) Como será o mapa de bits do disco se A for armazenado a partir da posição i do disco e B a partir da posição j do disco, supondo que não existam inconsistências no sistema de arquivos?

Resp.: Como não existem inconsistências no sistema de arquivos, então podemos concluir que os arquivos não têm blocos em comum e que ambos os arquivos estão totalmente armazenados no disco. Além disso, como vimos na aula 12, o mapa de bits do disco tem n bits, sendo que o bit i , $0 \leq i < n$ (os blocos são numerados de 0 até $n - 1$), indica se o bloco i do disco está sendo usado (se for 0) ou livre (se for 1). Agora, como na alocação contígua os blocos de um arquivo são consecutivos do disco, A usa os blocos de i até $i + a - 1$ e B os blocos de j até $j + b - 1$. Com isso, os bits de i até $i + a - 1$ e de j até $j + b - 1$ do mapa de bits são todos iguais

a 0 e os outros bits restantes do mapa são todos iguais a 1.

- (b) (1,0) Quais serão os possíveis blocos iniciais de A no disco se o sistema de arquivos estiver inconsistente devido a A e B estarem compartilhando x blocos?

Resp.: Primeiramente notemos que $a \geq x$ e $b \geq x$, pois A e B compartilham x blocos do disco. Como a alocação contígua é usada, então temos as seguintes possibilidades, dependendo de qual arquivo é o primeiro armazenado no disco. Se A for armazenado antes de B, então os x blocos finais de A são os mesmos x blocos iniciais de B, o que implica que devem existir exatamente $a - x$ blocos entre os blocos i e $j - 1$, ou seja, $j = i + a - x$. Agora, como somente existe a inconsistência descrita no enunciado, então $j + b \leq n$, ou seja $i + a - x + b \leq n$ ou $i \leq n - a - b + x$. Como A é o primeiro arquivo, ele pode começar a partir do bloco inicial e, com isso, neste primeiro caso teremos que $0 \leq i \leq n - a - b + x$. No caso de B ser armazenado antes de A, os x blocos finais de B são os mesmos x blocos iniciais de A, o que implica que devem existir exatamente $b - x$ blocos entre os blocos j e $i - 1$, ou seja, pelo menos $b - x$ blocos antes da posição i , isto é, $i \geq b - x$. Agora, como novamente somente temos uma inconsistência no disco, então $i + a \leq n$ ou $i \leq n - a$. Com isso, no segundo caso, temos que $b - x \leq i \leq n - a$.