



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 2017

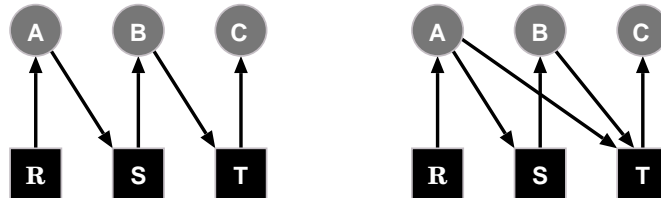
**Atenção:** Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um,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

Nome -  
Assinatura -

- 
1. (1,5) Suponha que três recursos não-preemptivos, R, S e T, estejam alocados, respectivamente, aos processos A, B e C. Responda, justificando a sua resposta:
    - (a) (1,0) Quantas solicitações de recursos podem no máximo ser feitas sem que um impasse ocorra?

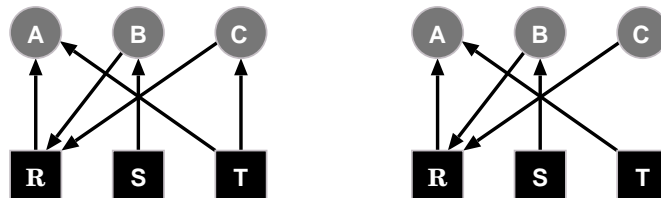
**Resp.:** Se uma solicitação for composta por somente um recurso, então somente duas solicitações adicionais poderão ser feitas sem que ocorra um impasse. Um exemplo é mostrado na figura a seguir (lado esquerdo). Caso uma solicitação possa ser composta

por mais de um recurso, esse número passa de 2 para 3, conforme ilustrado no lado direito da figura.



- (b) (0,5) O que ocorrerá se T passar a ser preemptivo e, depois disso, A solicitar T e B e C solicitarem R?

**Resp.:** Neste caso não teremos um impasse pois, independentemente de T ser ou não compartilhável, ou seja, poder ou não ser alocado a mais de um processo ao mesmo tempo, não teremos um ciclo orientado no grafo de recursos. A parte esquerda da figura a seguir supõe que T é compartilhável, e a parte direita supõe que T não é compartilhável e que T foi removido de C e depois alocado a A.

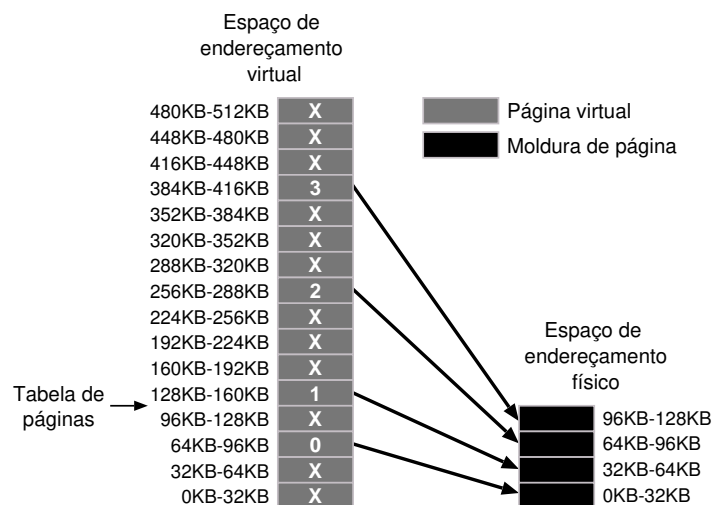


2. (1,5) Suponha que um computador tenha uma memória virtual de 512KB e uma memória física de 128KB. Suponha ainda que as páginas virtuais tenham 32KB e que, inicialmente, nenhuma página virtual esteja mapeada em uma moldura de página. Cada página virtual  $p$  sempre é mapeada na moldura de página  $\lfloor p/t \rfloor$ , onde  $t$  é a razão entre o número de páginas virtuais e o número de molduras de página, substituindo a página que esteja na moldura, se necessário. Como será o mapeamento das páginas nas molduras após um processo acessar, em ordem, os endereços virtuais 524287, 0, 65536, 263388, 157283 e 404517? Para responder, use uma figura similar à dada no slide 19 da

aula 8. Justifique a sua resposta.

**Resp.:** Pelo enunciado da questão, vemos que o espaço de endereçamento virtual possui 512KB, com endereços de 19 bits variando de 0 até 524287. Já o espaço de endereçamento físico possui 128KB, com endereços de 17 bits variando de 0 até 131071. As páginas virtuais possuem 32KB de tamanho e, com isso, temos 16 páginas virtuais e 4 molduras de página, pois o tamanho de cada moldura é igual ao da página. Logo, um endereço virtual é dividido em 4 bits para o número da página virtual e 15 bits para o deslocamento, um endereço físico é dividido em 2 bits para o número da moldura de página e 15 bits para o deslocamento, e  $t = 16/4 = 4$ . A tabela a seguir mostra, para cada endereço virtual, supondo a ordem de acesso dada no enunciado, como ele é na base binária e como ele é dividido, a página virtual associada a ele e, finalmente, caso o acesso gere uma falha de página, a moldura escolhida segundo o critério dado no enunciado. Com base nessa tabela, obtemos a figura dada logo a seguir, a qual mostra todos os mapeamentos das páginas virtuais nas molduras de página após todos os acessos.

Endereço virtual	Endereço em binário (página   deslocamento)	Página virtual	Moldura escolhida
524287	1111   111111111111111	15	$\lfloor 15/4 \rfloor = 3$
0	0000   000000000000000	0	$\lfloor 0/4 \rfloor = 0$
65536	0010   000000000000000	2	$\lfloor 2/4 \rfloor = 0$
263388	1000   000010011011100	8	$\lfloor 8/4 \rfloor = 2$
157283	0100   110011001100011	4	$\lfloor 4/4 \rfloor = 1$
404517	1100   010110000100101	12	$\lfloor 12/4 \rfloor = 3$



3. (2,5) Considere a execução do algoritmo FIFO dada a seguir, que mostra como as páginas virtuais são substituídas conforme um processo acessa, de cima para baixo, as páginas dadas na tabela. Responda, justificando a sua resposta, supondo que todas as molduras estejam inicialmente vazias:

Páginas	Ordenação	Ocorreu uma falha?
1	1	Sim
3	1 3	Sim
2	1 3 2	Sim
4	3 2 4	Sim
0	2 4 0	Sim
1	4 0 1	Sim
4	4 0 1	Não
2	0 1 2	Sim
3	1 2 3	Sim
1	1 2 3	Não

- (a) (1,5) Se o sistema operacional tivesse usado o algoritmo NRU, teríamos tido o mesmo número de falhas de página? Suponha que o bit **referenciada** seja ligado quando a página é acessada e desligado quando duas páginas diferentes dela são acessadas, e

que somente as páginas 1 e 3 sejam alteradas quando acessadas, respectivamente, pela segunda e primeira vez.

**Resp.:** Como vimos na aula 9, no algoritmo NRU as páginas são primeiramente divididas em 4 classes diferentes, de acordo com o valor dos bits **modificada** ( $M$ ) e **referenciada** ( $R$ ): Classe 0: ambos os bits  $R$  e  $M$  são 0; Classe 1: o bit  $R$  é 0 e o bit  $M$  é 1; Classe 2: o bit  $R$  é 1 e o bit  $M$  é 0; e Classe 3: ambos os bits  $R$  e  $M$  são 1. Depois de dividir as páginas em classes, a página a ser substituída será uma das páginas da classe não vazia com o menor número. No caso desta questão, como há três molduras de página e como são necessários dois acessos a páginas diferentes para que uma dada página tenha seu bit  $R$  mudado de 1 para 0, para cada página na lista de acesso dada é necessário informar dois estados das quatro classes: o primeiro é o estado após o acesso mas antes de a página ser copiada para a memória, se necessário; já o segundo é o estado que já leva em conta essa cópia. Como existiam duas possíveis páginas, a 1 e a 3, para serem escolhidas quando a página 2 foi acessada pela segunda vez, mostramos a seguir duas tabelas, uma para cada escolha. Na primeira tabela, existem duas possíveis escolhas de página a ser substituída, a 2 ou a 4, quando a página 1 é acessada pela terceira vez, mas a única diferença é que, caso a página 4 fosse a escolhida, a última linha da tabela teria a página 2 na classe 0. Além disso supomos, por coerência com o enunciado da questão, que a página será salva no disco caso esteja na memória e seja acessada para leitura depois de ter sido acessada para escrita. Em cada tabela dada a seguir, mostramos esses estados para cada um dos acessos. Também informarmos se cada acesso gerou ou não uma falha de página. As classes são mostradas nas colunas de 2 até 5, e a coluna 6 indica se ocorreu ou não uma falha. Como podemos ver, independentemente da escolha, ocorreu uma falha a mais quando o sistema operacional usou o algoritmo NRU.

Página virtual	Classes				Falha de página?
	0	1	2	3	
1	-	-	-	-	Sim
$R = 1, M = 0$	-	-	1	-	
3	-	-	1	-	Sim
$R = 1, M = 1$	-	-	1	3	
2	1	-	-	3	Sim
$R = 1, M = 0$	1	-	2	3	
4	1	3	2	-	Sim
$R = 1, M = 0$	-	3	2,4	-	
0	2	3	4	-	Sim
$R = 1, M = 0$	-	3	0,4	-	
1	4	3	0	-	Sim
$R = 1, M = 1$	-	3	0	1	
4	0	3	-	1	Sim
$R = 1, M = 0$	-	3	4	1	
2	-	1,3	4	-	Sim
$R = 1, M = 0$	-	3	2,4	-	
3	4	-	2,3	-	Não
$R = 1, M = 0$	4	-	2,3	-	
1	2,4	-	3	-	Sim
$R = 1, M = 0$	4	-	1,3	-	

Página virtual	Classes				Falha de página?
	0	1	2	3	
1	-	-	-	-	Sim
$R = 1, M = 0$	-	-	1	-	
3	-	-	1	-	Sim
$R = 1, M = 1$	-	-	1	3	
2	1	-	-	3	Sim
$R = 1, M = 0$	1	-	2	3	
4	1	3	2	-	Sim
$R = 1, M = 0$	-	3	2,4	-	
0	2	3	4	-	Sim
$R = 1, M = 0$	-	3	0,4	-	
1	4	3	0	-	Sim
$R = 1, M = 1$	-	3	0	1	
4	0	3	-	1	Sim
$R = 1, M = 0$	-	3	4	1	
2	-	1,3	4	-	Sim
$R = 1, M = 0$	-	1	2,4	-	
3	4	1	2	-	Sim
$R = 1, M = 0$	-	1	2,3	-	
1	2	-	1,3	-	Não
$R = 1, M = 0$	2	-	1,3	-	

- (b) (1,0) Se o sistema operacional tivesse usado o algoritmo LRU, teríamos tido a mesma sequência de substituições?

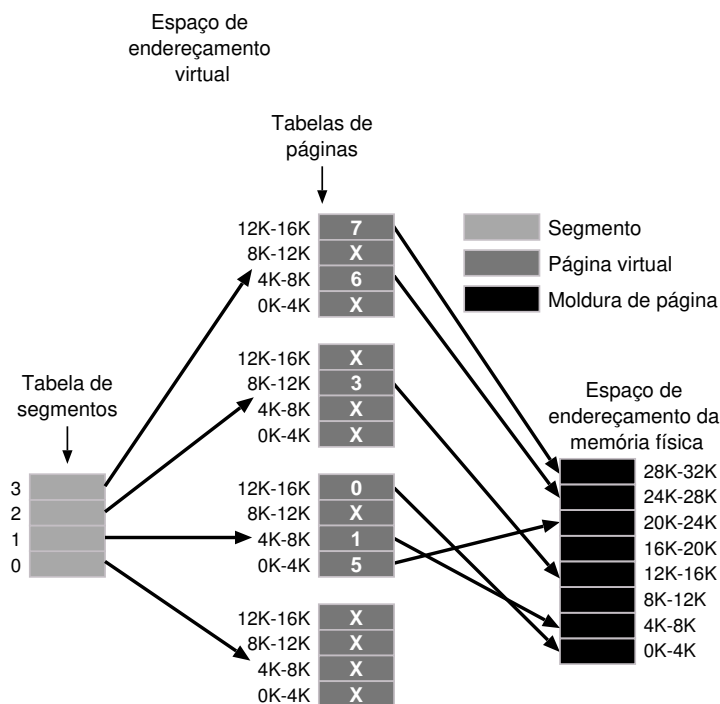
**Resp.:** Agora vamos mostrar, na tabela a seguir, a sequência de acessos às páginas virtuais dadas na questão para o algoritmo LRU. Como vimos na aula 9, no algoritmo LRU as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo do seu último acesso. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página não acessada há mais tempo. Na tabela dada a seguir mostramos, em cada linha, o que ocorre ao acessarmos as páginas na ordem dada na tabela do enunciado, usando o mesmo número de molduras. 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, a sequência de páginas substituídas será diferente, pois será 1, 3, 2, 0, 1 e 4 ao invés de 1, 3, 2, 4 e 0.

Páginas	Ordenação			Ocorreu uma falha?
1	1			Sim
3	1	3		Sim
2	1	3	2	Sim
4	3	2	4	Sim
0	2	4	0	Sim
1	4	0	1	Sim
4	0	1	4	Não
2	1	4	2	Sim
3	4	2	3	Sim
1	2	3	1	Sim

4. (1,5) Suponha que o sistema operacional use a técnica de segmentação com paginação, e que o computador tenha um espaço de endereçamento virtual dividido como na figura dada a seguir. Responda, justificando a sua resposta:





- (a) (0,5) Para cada segmento, que faixas de endereços geram falhas de página ao serem acessadas, sendo que “X” em uma entrada de uma tabela de páginas indica que a página não está na memória?

**Resp.:** Para o segmento 0, vemos que uma falha de página ocorrerá ao acessarmos qualquer endereço do segmento, que varia de 0 até 16383 (faixas 0K–4K, 4K–8K, 8K–12K e 12K–16K). Para o segmento 1, uma falha de página será gerada se acessarmos os endereços de 8192 até 12287 (faixa 8K–12K). Para o segmento 2, uma falha de página será gerada ao acessarmos os endereços de 0 até 8191 (faixas 0K–4K e 4K–8K) ou os endereços de 12288 até 16383 (faixa 12K–16K). Finalmente, para o segmento 3, os acessos aos endereços de 0 até 4095 (faixa 0K–4K) ou aos endereços de 8192 até 12287 (faixa 8K–12K) gerarão falhas de página.

- (b) (0,5) Que endereço virtual está mapeado no endereço físico 14350?

**Resp.:** No caso do endereço físico 14350 (em binário, 011 | 100000-

001110, sendo que separamos a moldura de página do deslocamento por um “|”), vemos que estamos acessando a palavra 2062 da moldura de página 3 (com endereços de 12288 até 16383). Agora, como a página virtual 2 do segmento 2 está mapeada nesta moldura (com endereços de 8192 até 12287), então o endereço dentro do segmento 2 é 8192 (o primeiro endereço da página) mais 2062 (o deslocamento), isto é, 10254. Logo, o endereço virtual (2, 10254) está mapeado no endereço físico 14350.

- (c) (0,5) Que endereço físico está mapeando o endereço virtual (3, 1234)? Se o endereço virtual não estiver mapeado, dê o endereço físico que resultaria do uso de cada moldura ainda não preenchida.

**Resp.:** Para o endereço virtual (3, 1234), que está no segmento 3 e no endereço 1234 (em binário, 00 | 010011010010, sendo que separamos a página virtual do deslocamento por um “|”) dentro desse segmento, vemos que estamos acessando a palavra 1234 da página virtual 0 (com endereços de 0 até 4095). Como a página virtual 0 do segmento 3 não está mapeada na memória, e como as molduras de página 2 e 4 estão livres, então temos dois possíveis mapeamentos, de acordo com a moldura escolhida: se a moldura 2 é a escolhida (com endereços de 8192 até 12287), então o endereço físico é 8192 (o primeiro endereço da moldura) mais 1234 (o deslocamento), isto é, 9426; já se a moldura 4 é a escolhida (com endereços de 16384 até 20479), então o endereço físico é agora 16384 (o primeiro endereço da moldura) mais 1234 (o deslocamento), isto é 17618.

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, usando o acesso sequencial, a mais do que usando o acesso aleatório, se desejarmos:

**Arquivo prova.pdf (14 posições)**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	----	----	----	----	----

- (a) (0,8) acessar os blocos pares em ordem decrescente e depois os blocos ímpares em ordem crescente?
- (b) (0,7) acessar os blocos na seguinte ordem: 1, 10, 14, 7, 8, 2, 3, 9, 11, 12, 5, 4, 6 e 13?

**Resp.:** Quando usamos o acesso aleatório, as posições do arquivo podem ser acessadas independentemente umas das outras. Logo, cada posição é acessada somente uma vez com esse acesso. 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 feitos a mais pelo acesso sequencial em relação ao aleatório, que é igual ao número total de acessos usando o acesso sequencial menos um para todas as posições.

(a) os blocos pares em ordem decrescente,  
depois os blocos ímpares em ordem crescente.

$x$	Precisam ser acessadas													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
14	X	X	X	X	X	X	X	X	X	X	X	X	X	X
12	X	X	X	X	X	X	X	X	X	X	X	X		
10	X	X	X	X	X	X	X	X	X	X				
8	X	X	X	X	X	X	X	X						
6	X	X	X	X	X	X								
4	X	X	X	X										
2	X	X												
1	X													
3		X	X											
5				X	X									
7						X	X							
9								X	X					
11										X	X			
13												X	X	
Total	7	7	6	6	5	5	4	4	3	3	2	2	1	0

(b) os blocos na seguinte ordem: 1, 10, 14, 7, 8, 2, 3,  
9, 11, 12, 5, 4, 6 e 13.

$x$	Precisam ser acessadas													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	X													
10		X	X	X	X	X	X	X	X	X				
14											X	X	X	X
7	X	X	X	X	X	X	X							
8								X						
2	X	X												
3			X											
9				X	X	X	X	X	X					
11										X	X			
12												X		
5	X	X	X	X	X									
4	X	X	X	X										
6					X	X								
13							X	X	X	X	X	X	X	
Total	4	4	4	4	4	3	3	3	2	2	2	2	1	0

6. (1,5) Suponha que um disco tenha  $n$  blocos de  $k$  bytes, e que o mapa de bits do disco seja armazenado no início dele. Como é o mapa de bits desse disco após um arquivo A, com  $xk$  bytes, ser armazenado a partir do bloco  $i$  do disco usando a alocação contígua.

**Resp.:** Como o disco tem  $n$  blocos, que supomos serem numerados de 0 até  $n - 1$ , e como cada bloco do disco está associado a um bit diferente do mapa de bits, então o mapa tem  $n$  bits, ocupando portanto  $\lceil n/8k \rceil$  blocos do disco. Agora, como o mapa é armazenado a partir do bloco 0 do disco, ele usará os blocos de 0 até  $\lceil n/8k \rceil - 1$  do disco. Além disso, como o arquivo está armazenado a partir do bloco  $i$ , tem  $xk$  bytes, e como a alocação contígua é usada, então o arquivo é armazenado nos blocos de  $i$  até  $i + x - 1$ . Como os outros blocos não especificados anteriormente estão livres então, no mapa de bits, os bits de 0 até  $\lceil n/8k \rceil - 1$  e de  $i$  até  $i + x - 1$  são todos iguais a 0 e os bits restantes são todos iguais a 1.