



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 2014

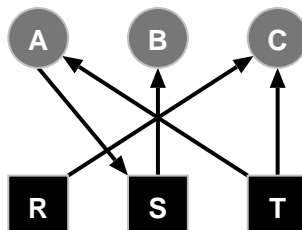
Atenção: Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um, 1/3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

Nome -
Assinatura -

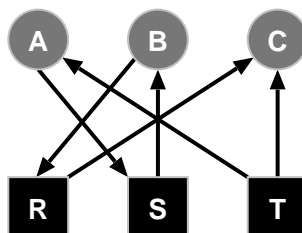
1. (1,0) Suponha que três processos, A, B e C, compartilhem dois recursos não-preemptivos, sendo estes uma impressora e um gravador de DVD, e um recurso preemptivo, o disco. Suponha ainda que A tenha obtido o disco, B o gravador de DVD e C a impressora, que A tenha sido bloqueado ao tentar obter o gravador de DVD e que C tenha obtido o disco. Existirá um impasse se B tentar obter o disco? E se B tentar obter a impressora? Justifique a sua resposta.

Resp.: -Na figura seguir mostramos o grafo de recursos antes de B tentar obter o segundo recurso. Na figura, R representa a impressora,

S o gravador de DVD e T o disco. Como não existem impasses no grafo, então, se B tentar obter o disco, ainda não teremos um impasse porque o disco é um recurso preemptivo.



-Agora, se B tentar obter a impressora, teremos o grafo de recursos dado a seguir. Como podemos ver pelo grafo, apesar de a impressora ser um recurso não-preemptivo, continuaremos não tendo um impasse porque não existe um ciclo orientado (o qual seria necessário, já que os recursos são todos não-preemptivos).

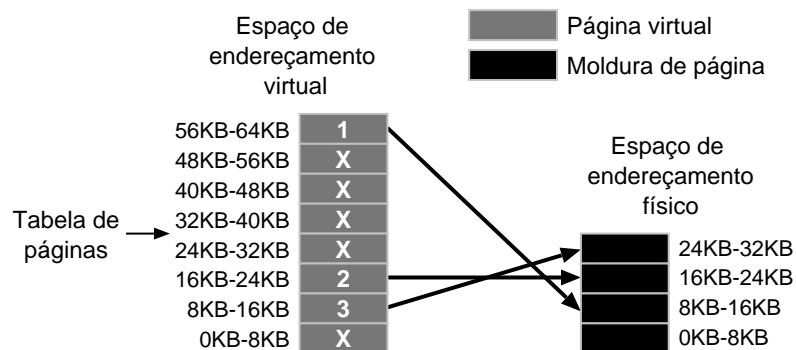


2. (3,0) Suponha que um computador tenha uma memória virtual de 64KB e uma memória física de 32KB. Suponha ainda que as páginas virtuais tenham 8KB de tamanho e que, inicialmente, nenhuma página virtual esteja mapeada em uma moldura de página. Responda:
 - (a) (1,0) Se o processo acessar os endereços virtuais dados na parte esquerda da tabela a seguir, e se cada endereço virtual dado em uma linha da tabela for mapeado para o endereço físico dado na coluna ao lado, como será o mapeamento das páginas virtuais nas molduras de página, usando uma figura similar à dada na transparência 19 da aula 8?

Endereço virtual	Endereço físico
12345	28729
60000	10848
17331	17331
58769	9617

Resp.: Pelo enunciado da questão, vemos que o espaço de endereçamento virtual possui 64KB, com endereços de 16 bits variando de 0 até 65535. Já o espaço de endereçamento físico possui 32KB, com endereços de 15 bits variando de 0 até 32767. As páginas possuem 8KB de tamanho e, com isso, temos 8 páginas virtuais e 4 molduras de página. Logo, um endereço virtual é dividido em 3 bits para o número da página virtual e 13 bits para o deslocamento, e um endereço físico é dividido em 2 bits para o número da moldura de página e 13 bits para o deslocamento. A seguir damos uma nova versão da tabela mostrando, separadas por uma barra vertical, a divisão de cada endereço virtual nos campos número da página virtual e deslocamento, e a divisão de cada endereço físico nos campos número da moldura de página e deslocamento (note que todos os deslocamentos são coerentes, o que indica que o mapeamento da tabela está correto). Usando os campos com os números das páginas virtuais e os campos com os números das molduras de página obtemos a figura dada a seguir, abaixo da tabela e similar à dada na aula, com os mapeamentos.

Endereço virtual	Endereço físico
12345 (001 1000000111001)	28729 (11 1000000111001)
60000 (111 0101001100000)	10848 (01 0101001100000)
17331 (010 0001110110011)	17331 (10 0001110110011)
58769 (111 0010110010001)	9617 (01 0010110010001)



- (b) (1,0) Considerando o mapeamento obtido no item (a), o endereço físico 0 está mapeando algum endereço virtual?

Resp.: Não, pois o endereço físico 0 está na moldura de página 0 e, pela figura, vemos que nenhuma página virtual está mapeada nessa moldura.

- (c) (1,0) Considerando novamente o mapeamento obtido no item (a), o que ocorreria se o endereço virtual 32767 fosse agora acessado?

Resp.: Usando a mesma divisão dada na tabela da resposta do item (a), o endereço virtual 32767 em binário é 011|111111111111. Logo, esse endereço está na página virtual 3, e o deslocamento dentro da página virtual é 111111111111 em binário ou 8191 em decimal. Como a página virtual 3 não está mapeada em nenhuma moldura de página, o acesso geraria uma falha de página e, depois disso, a moldura escolhida dependeria do algoritmo de substituição de páginas. Na tabela a seguir mostramos o endereço físico obtido para cada uma das possíveis molduras de página, novamente usando a divisão dada na tabela do item (a):

Número da moldura de página	Endereço físico
0	8191 (00 111111111111)
1	16383 (01 111111111111)
2	24575 (10 111111111111)
3	32767 (11 111111111111)

3. (1,0) Suponha que quatro molduras de página, inicialmente vazias, tenham sido alocadas a um processo. Se o processo acessar, em ordem, as páginas 1, 2, 3, 0, 3, 4, 1 e 2, ocorrerá uma falha de página se a página 4 for acessada em seguida e o algoritmo FIFO for usado pelo sistema operacional? E se a página 0 for acessada e o algoritmo LRU for usado pelo sistema? Justifique a sua resposta.

Resp.: -Primeiramente vamos mostrar como as páginas são substituídas de acordo com o algoritmo FIFO. Como vimos na aula 9, no algoritmo FIFO as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo em que elas foram copiadas para a memória. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página copiada há mais tempo. 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 e na segunda coluna a ordem em que as páginas devem ser escolhidas. Como podemos ver pela última linha da tabela, não ocorrerá uma falha de página porque a página 4 já está na memória.

Página	Ordenação
1	1
2	1 2
3	1 2 3
0	1 2 3 0
3	1 2 3 0
4	2 3 0 4
1	3 0 4 1
2	0 4 1 2

-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, na primeira coluna, damos a página acessada, sendo que as páginas são mostradas na mesma ordem dada no enunciado. Finalmente, na segunda coluna, mostramos a ordem em que as páginas são escolhidas pelo algoritmo. Como podemos ver pela tabela, ocorrerá uma falha de página porque a página 0 não está na memória.

Página	Ordenação
1	1
2	1 2
3	1 2 3
0	1 2 3 0
3	1 2 0 3
4	2 0 3 4
1	0 3 4 1
2	3 4 1 2

4. (1,0) Suponha que um computador tenha uma memória física dividida em x molduras de página com tamanhos de k bytes. Suponha ainda que o sistema operacional use a segmentação com paginação, e que um dado processo use n segmentos. Se o tamanho do segmento i , $1 \leq i \leq n$, for $y_i k$, que restrição sobre os tamanhos dos segmentos do processo permitiria que todos eles fossem armazenados na memória? Justifique a sua resposta.

Resp.: Pelo enunciado, notamos que o segmento i , $1 \leq i \leq n$, é dividido em exatamente y_i páginas virtuais, porque o segmento tem tamanho de $y_i k$ bytes e cada página virtual, cujo tamanho é igual ao da moldura de página, tem k bytes. Como desejamos armazenar integralmente todos os n segmentos do processo na memória, então cada segmento i , $1 \leq i \leq n$, precisará de exatamente y_i molduras de página. Logo, para podermos armazenar todas os segmentos na memória, o número total de molduras de página, x , não poderá ser menor do que o número de molduras necessárias por todos os n segmentos do processo, ou seja, $\sum_{i=1}^n y_i \leq x$, o que significa que $\sum_{i=1}^n y_i k \leq xk$.

5. (2,0) Considere um computador com um disco de 32 blocos de 4KB. Suponha que o disco esteja inicialmente vazio. Responda:
- (a) (1,0) Se o sistema operacional usar a alocação contígua, e se um arquivo A com 64KB for armazenado a partir do bloco i , qual será o tamanho máximo que um arquivo B poderá ter?

Resp.: Pela aula 11, o disco terá 32 blocos, numerados de 0 até 31. Como o arquivo A tem 64KB e como cada bloco tem 4KB, então o arquivo A ocupará 16 blocos do disco. Além disso, como a alocação contígua obriga que todos os 16 blocos de A sejam blocos consecutivos do disco, então o arquivo A usará os blocos de i até $i + 15$ e, com isso, $i \geq 0$ e $i + 15 \leq 31$, ou seja, $0 \leq i \leq 16$. Logo, o arquivo B poderá ser armazenado, caso $i > 0$, nos $i - 1$ bloco(s) inicial(is) do disco ou, caso $i < 16$, nos $31 - (15 + i) = 16 - i$ bloco(s) final(is) do disco. Agora, como cada bloco tem 4KB, então podemos dizer que o tamanho máximo de B é igual a $4 \max\{i - 1, 16 - i\}$ KB.

- (b) (1,0) Qual será a tabela obtida se o sistema operacional usar a alocação por lista encadeada utilizando um índice, se um arquivo A usar, em ordem, os blocos 1, 3 e 0, um arquivo B usar, em ordem, os blocos 4, 31, 22, 15 e 30, e um arquivo C usar, em ordem, os blocos 25, 24, 23, 26, 27 e 11?

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 32 entradas, referenciadas pelos endereços dos blocos, pois temos 32 blocos no disco, numerados de 0 até 31. Nessa tabela, um “X” na entrada indica que o bloco associado a ela é o último bloco do arquivo.

0	X	← Fim do arquivo A
1	3	← Início do arquivo A
2	LIVRE	
3	0	
4	31	← Início do arquivo B
5	LIVRE	
6	LIVRE	
7	LIVRE	
8	LIVRE	
9	LIVRE	
10	LIVRE	
11	X	← Fim do arquivo C
12	LIVRE	
13	LIVRE	
14	LIVRE	
15	30	
16	LIVRE	
17	LIVRE	
18	LIVRE	
19	LIVRE	
20	LIVRE	
21	LIVRE	
22	15	
23	26	
24	23	
25	24	← Início do arquivo C
26	27	
27	11	
28	LIVRE	
29	LIVRE	
30	X	← Fim do arquivo B
31	22	

Ordem dos blocos:

arquivo A: 1,3 e 0.

arquivo B: 4, 31, 22, 15 e 30.

arquivo C: 25, 24, 23, 26, 27 e 11.

6. (2,0) Suponha que o sistema operacional use um domínio de proteção com os objetos e os domínios definidos pela matriz mostrada a seguir, similar à matriz que foi vista na aula 12. Responda:

Objetos

A: Arquivo1 **B:** Arquivo2
C: Arquivo3 **D:** Arquivo4
E: Arquivo5 **F:** Arquivo6
G: Impressora1 **H:** Plotter2
I: Domínio1 **J:** Domínio2
K: Domínio3

		Objeto										
		A	B	C	D	E	F	G	H	I	J	K
Domínio	1	R	RW		W		WX			Enter		Enter
	2	RWX		R		RW	X		R		Enter	
	3	X			RX		RWX		WX	Enter		

- (a) (1,0) Que objetos o domínio 3 pode acessar? Para cada um destes objetos, que operações podem ser executadas?

Resp.: -O domínio 3 pode acessar cada objeto que não seja um domínio (pois a operação **Enter** é usada para a troca de domínios) e cuja entrada na linha do domínio 3 possua alguma das operações **r**, **w** ou **x**. Logo, podem ser acessados os seguintes objetos: A (Arquivo1), D (Arquivo4), F (Arquivo6) e H (Plotter2).

-Na aula 12 vimos que **r** significa que podemos ler dados do objeto, que **w** significa que podemos salvar dados no objeto, e que **x** significa que podemos executar o objeto. Com isso:

- Podemos somente executar o objeto A, ou seja, podemos somente executar o Arquivo1;
- Podemos ler dados do objeto D ou executá-lo, ou seja, podemos ler ou executar o Arquivo4;
- Podemos ler, salvar, ou executar o objeto F, ou seja, podemos ler, alterar ou executar o Arquivo6;
- Podemos escrever dados no objeto H ou executá-lo. Note que como a operação executar é inadequada para um plotter como o Plotter2, então possivelmente existe uma inconsistência no

sistema. Porém a operação salvar está correta, pois significa, neste caso, imprimir no Plotter2.

- (b) (1,0) Estando no domínio 1, um processo pode passar, usando a operação **Enter**, para quais domínios?

Resp.: Para descobrirmos para quais domínios o processo pode passar, devemos olhar as colunas da linha do domínio 1 associadas a objetos que sejam domínios e que possuam a operação **Enter**. Pela matriz, vemos que podemos passar do domínio 1 para os domínios 1 (uma operação não lógica mas válida) e o 3.