



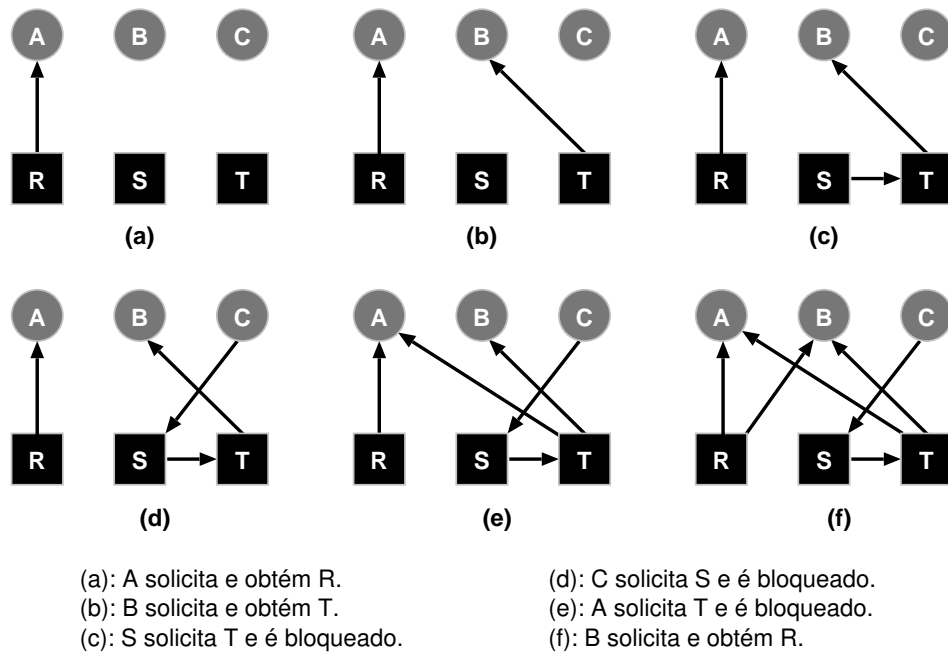
Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
Professores: Valmir C. Barbosa e Felipe M. G. França
Assistente: Alexandre H. L. Porto

Quarto Período
Gabarito da AD2 - Primeiro Semestre de 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) Um aluno de sistemas operacionais disse que a figura a seguir é um possível exemplo de uma sequência de alocações de recursos, sendo que R é um recurso preemptivo, S e T são recursos não-preemptivos e A , B e C são processos. A sequência do aluno está correta? Se você achar que sim, basta dizer isso mas, se você achar que não, diga quais foram os passos errados do aluno na sequência. Justifique a sua resposta.



Resp.: A sequência do aluno não está correta, pois as alocações dos passos de (c) até (e) estão erradas. No caso do passo (c), como somente processos podem solicitar recursos, então S não pode solicitar T. No caso do passo (d), o processo C pode obter o recurso S diretamente, porque ele não está alocado a nenhum outro processo. Já no passo (e), o erro está na orientação da aresta, pois quando o processo A tentasse obter o recurso T e fosse bloqueado porque T já foi alocado a B, a aresta deveria ser orientada de A para T.

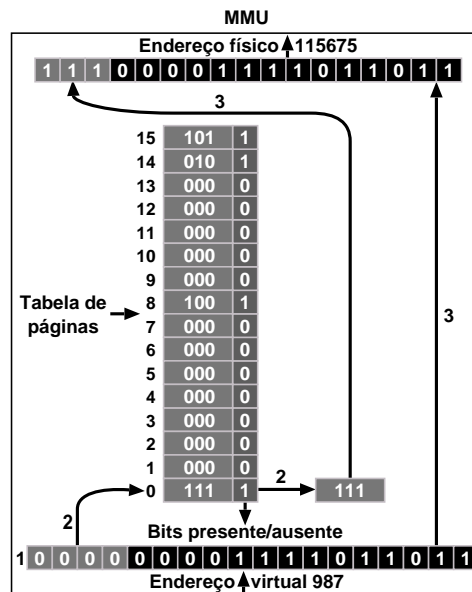
2. (1,5) Suponha que um computador tenha uma memória virtual de 256KB e uma memória física de 128KB. Suponha ainda que as páginas virtuais tenham 16KB de tamanho e que, inicialmente, nenhuma página virtual esteja mapeada em uma moldura de página. Responda:

- (a) (1,0) Se o processo acessar os endereços virtuais dados na parte esquerda da tabela a seguir, e se cada endereço virtual 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 20 da aula 8, na qual o endereço da primeira linha da tabela é dado como exemplo?

Endereço virtual	Endereço físico
987	115675
131072	65536
256000	92160
16383	131071
234567	37959

Resp.: Pelo enunciado da questão, vemos que o espaço de endereçamento virtual tem 256KB, com endereços de 18 bits variando de 0 até 262143. Já o espaço de endereçamento físico tem 128KB, com endereços de 17 bits variando de 0 até 131071. As páginas possuem 16KB de tamanho e, com isso, temos 16 páginas virtuais e 8 molduras de página. Logo, um endereço virtual é dividido em 4 bits para o número da página virtual e 14 bits para o deslocamento, e um endereço físico é dividido em 3 bits para o número da moldura de página e 14 bits para o deslocamento. A seguir apresentamos 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, os campos com os números das molduras de página, e o mapeamento dado na primeira linha da tabela como exemplo, obtemos a figura dada após a tabela.

Endereço virtual	Endereço físico
987 (0000 00001111011011)	115675 (111 00001111011011)
131072 (1000 00000000000000)	65536 (100 00000000000000)
256000 (1111 10100000000000)	92160 (101 10100000000000)
16383 (0000 11111111111111)	131071 (111 11111111111111)
234567 (1110 01010001000111)	37959 (010 01010001000111)



- (b) (0,5) Considerando o mapeamento obtido no item (a), o endereço físico 123456 está mapeando algum endereço virtual? E o endereço virtual 262143 está mapeado em algum endereço físico?

Resp.: -Quando convertemos o endereço físico 123456 para binário, usando a mesma divisão da tabela do item (a), obtemos 111|100010-01000000. Baseado no modo como a divisão foi feita, o endereço pertence à moldura de página 7 que, como podemos ver pelo mapeamento obtido no item (a), está mapeando a página virtual 0, ou 0000 em binário. Logo, o endereço está mapeando um endereço virtual que, em binário, novamente usando a divisão da tabela do item (a), é 0000|10001001000000, ou seja, o endereço virtual 8768. -Quando convertemos o endereço virtual 262143 para binário, usando a mesma divisão da tabela do item (a), obtemos 1111|111111-11111111, indicando que o endereço virtual está associado à página virtual 15. Agora, pelo mapeamento obtido no item (a), a página 15 está mapeada na moldura de página 5, ou 101 em binário. Logo, o endereço está mapeado em um endereço físico que, em binário, novamente usando a divisão da tabela do item (a), é 101|11111111111111, ou seja, o endereço físico 98303.

3. (1,5) Suponha que um computador possua um espaço de endereçamento virtual de 64 bits e um espaço de endereçamento físico de 32 bits. Suponha ainda que o tamanho de uma página virtual seja de 64KB. Responda:

- (a) (0,5) Quantas molduras de página existem no espaço de endereçamento físico? E quantas páginas virtuais existem no espaço de endereçamento virtual?

Resp.: -Pelo enunciado, vemos que o tamanho da moldura de página é de 64KB, ou 2^{16} bytes, porque é igual ao tamanho da página virtual, significando que o campo deslocamento do endereço físico tem 16 bits. Com isso, o endereço físico é dividido em um campo número da moldura de página com 16 bits e um campo deslocamento também com 16 bits. Ou seja, existem 2^{16} molduras de página.

-Agora, como vimos que o tamanho da página virtual é de 2^{16} bytes, então o campo deslocamento do endereço virtual também tem 16 bits. Logo, o endereço virtual é dividido em um campo número da página virtual com 48 bits e um campo deslocamento com 16 bits. Assim, existem 2^{48} páginas virtuais.

- (b) (0,5) Suponha que desejemos dividir o campo **número da página virtual** do endereço virtual em partes iguais, para podermos usar uma tabela de páginas multinível. De quantos modos poderemos dividir esse campo, supondo que cada parte não poderá ter menos do que 16 bits?

Resp.: Como o campo número da página virtual, pelo item (a), tem 48 bits, temos dois modos de dividi-lo, satisfazendo a restrição de cada parte ter o mesmo tamanho e pelo menos 16 bits: dividir em dois novos campos de 24 bits, criando uma tabela de páginas com dois níveis; ou dividir em três novos campos de 16 bits, criando uma tabela de páginas com três níveis.

- (c) (0,5) Se a MMU possuir uma TLB com 2^{16} entradas e se todas as páginas virtuais possuírem igual probabilidade de serem aces-

sadas, qual será a probabilidade de o mapeamento de uma página poder ser feito usando as informações armazenadas em uma das entradas da TLB?

Resp.: Como todas as páginas virtuais têm igual probabilidade de serem acessadas, a probabilidade de uma página estar na TLB é igual ao número de entradas na TLB, ou seja 2^{16} , dividido pelo número total de páginas virtuais que, pelo item (a), é igual a 2^{48} . Logo, a probabilidade é $\frac{2^{16}}{2^{48}} = 2^{-32}$.

4. (2,5) Suponha que o sistema operacional tenha disponibilizado 4 molduras de página para serem alocadas a dois processos A e B. Suponha ainda que os processos tenham acessado as páginas virtuais segundo a ordem 1, 2, 3, 4, 2, 0, 3, 0, 1, 2, 0 e 4. Se o algoritmo LRU for usado junto com uma política de alocação global, o que ocorrerá se a página 1 for novamente acessada? E o que ocorrerá se o algoritmo FIFO for usado junto com uma política de alocação local usando a alocação igualitária, supondo agora que a página 2 tenha sido novamente acessada? Para este último caso, suponha que A acesse somente as páginas 0 e 2 e B acesse somente as páginas 1, 3 e 4. Justifique a sua resposta.

Resp.: -Primeiramente vamos mostrar, na tabela a seguir, a sequência de acessos às páginas virtuais dada na questão para o algoritmo LRU usando a política de alocação global. 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. Além disso, na política de alocação global, as páginas acessadas por todos os processos devem ser consideradas pelo algoritmo LRU. 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, como a página 1 está na memória, não ocorrerá uma falha de página, mas a ordem de escolha das páginas passará a ser 2, 0, 4, 1.

Páginas	Ordenação
1	1
2	1 2
3	1 2 3
4	1 2 3 4
2	1 3 4 2
0	3 4 2 0
3	4 2 0 3
0	4 2 3 0
1	2 3 0 1
2	3 0 1 2
0	3 1 2 0
4	1 2 0 4

-Agora, se usarmos a política de alocação local igualitária, 2 molduras serão alocadas a cada processo. Além disso, quando uma falha de página ocorrer, somente as páginas nas molduras alocadas ao processo que gerou a falha serão consideradas. Como a página 2 somente é acessada pelo processo A, e como o número de molduras alocadas a A é igual ao número de páginas diferentes acessadas por ele, somente os primeiros acessos às páginas 0 e 2 gerarão falhas de página. Logo, o novo acesso à página 2 não gerará uma falha de página e, como a ordem de escolha do FIFO é sempre a ordem de cópia das páginas para a memória, a ordem de escolha das páginas será 2, 0.

5. (2,0) Suponha que um computador tenha um disco com 32 blocos numerados de 0 até 31, e que o sistema operacional use a técnica de alocação por lista encadeada baseada em um índice ao gerenciar os blocos do disco. Responda às seguintes perguntas justificando a sua resposta, supondo que a alocação dos blocos do disco seja dada pela tabela a seguir:

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

Ordem dos blocos:
moon.jpg: 3,1,9,8,13,31,5,0.
relat.pdf: 2,17,16,15,4,27,19,12,14,24,25,26.

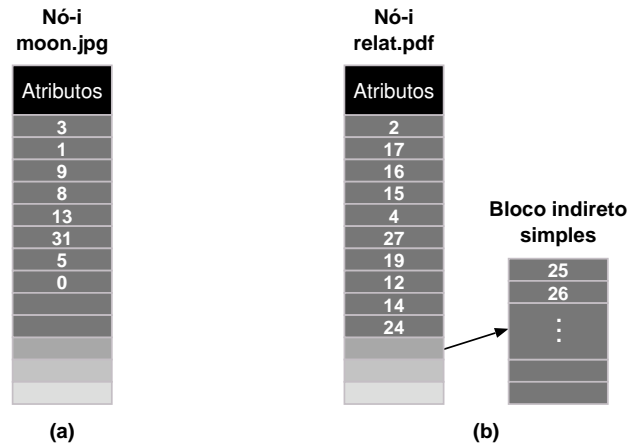
- (a) (1,0) Se a alocação contígua passar a ser usada, ainda vai ser possível armazenar os arquivos se supusermos que o último bloco lógico de cada arquivo continue sendo o mesmo?

Resp.: Não, porque, apesar de podermos armazenar todo o arquivo relat.pdf nos blocos de 15 até 26, não podemos armazenar todo o arquivo moon.jpg terminando no bloco 0 porque nenhum bloco o precede.

- (b) (1,0) Como os arquivos serão agora alocados se o sistema operacional passar a usar a técnica de alocação baseada em nós-i?

Resp.: Na figura a seguir mostramos como cada arquivo é alocado usando a técnica de alocação baseada em nós-i que vimos na aula 11. Nessa figura, mostramos o nó-i usado por moon.jpg na parte (a) e o nó-i usado por relat.pdf na parte (b). Na figura, somente preenchemos os campos dos nós-i associados aos blocos que foram usados (os campos com os atributos não precisavam ser

preenchidos e não serão considerados na correção da questão).



6. (1,0) Suponha que o sistema operacional use um domínio de proteção com os objetos e domínios definidos pela matriz mostrada a seguir, similar à matriz que foi vista na aula 12. Responda:

Objetos

A: AD1 B: AD2
C: GabaritoAD1 D: GabaritoAP1
E: Aula001 F: Aula012
G: ImpressoraHP H: PlotterHP
I: Domínio1 J: Domínio2
K: Domínio3

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

- (a) (0,5) Que objetos o domínio 2 pode acessar? Para cada um desses objetos, que operações podem ser executadas?

Resp.: -O domínio 2 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 2 possua alguma das operações **r**, **w** ou **x**. Logo, podem ser acessados os seguintes objetos: B (AD2), C (GabaritoAD1), E (Aula001), F (Aula012) e H (PlotterHP).

-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 B, ou seja, podemos somente executar o arquivo AD2;
 - Podemos ler ou escrever dados no objeto C, ou seja, podemos ler ou escrever no arquivo GabaritoAD1;
 - Podemos ler dados do objeto E ou executá-lo, ou seja, podemos ler ou executar o arquivo Aula001;
 - Podemos ler, salvar, ou executar o objeto F, ou seja, podemos ler, escrever ou executar o arquivo Aula012;
 - Podemos somente escrever dados no objeto H. Note que a operação salvar está correta porque para o objeto H isso significa que desejamos imprimir no PlotterHP.
- (b) (0,5) Estando no domínio 3, 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 3 associadas a objetos que sejam domínios e que possuam a operação **Enter**. Pela matriz, vemos que podemos passar do domínio 3 para os domínios 1 (objeto I) e 2 (objeto J).