



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 2019

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.

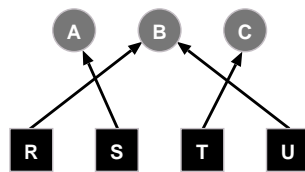
Nome -
Assinatura -

-
1. (1,5) Suponha que três processos, A, B e C, estejam em execução no sistema operacional, e que quatro recursos não-preemptivos, R, S, T e U, estejam disponíveis para serem usados pelos processos, sendo que em cada solicitação um processo somente pode tentar obter um recurso. Suponha ainda que A tenha obtido com sucesso S, B tenha obtido com sucesso R e U (em duas solicitações diferentes) e C tenha obtido com sucesso T. Se agora A tentar obter T, C tentar obter R e B tentar obter

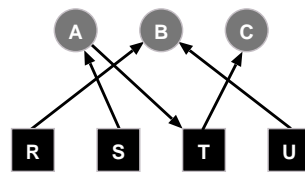
S responda, justificando a sua resposta:

- (a) (0,5) Como o grafo de recursos irá mudar conforme as solicitações adicionais de recursos forem feitas de acordo com a ordem dada no enunciado da questão?

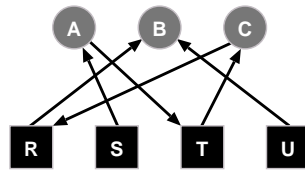
Resp.: O grafo de recursos mudará como mostrado nos itens da figura a seguir.



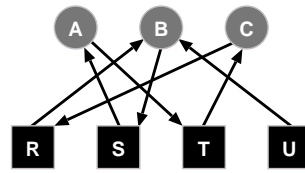
(a)



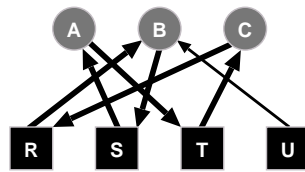
(b)



(c)



(d)



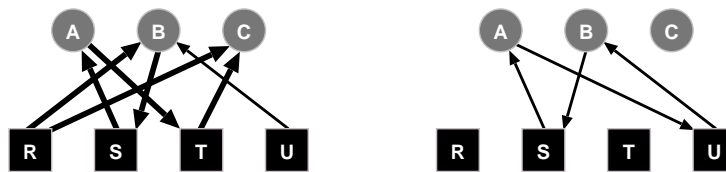
(e)

(a): Alocação inicial.
(b): A solicita T e é bloqueado.
(c): C solicita R e é bloqueado.

(d): B solicita S e é bloqueado.
(e): Impasse existente no grafo.

- (b) (1,0) Existe um impasse no grafo de recursos obtido após todas as solicitações terem sido feitas pelos processos? Caso exista um impasse, poderíamos ter evitado a ocorrência de qualquer impasse se somente um dos recursos fosse preemptivo e compartilhável?

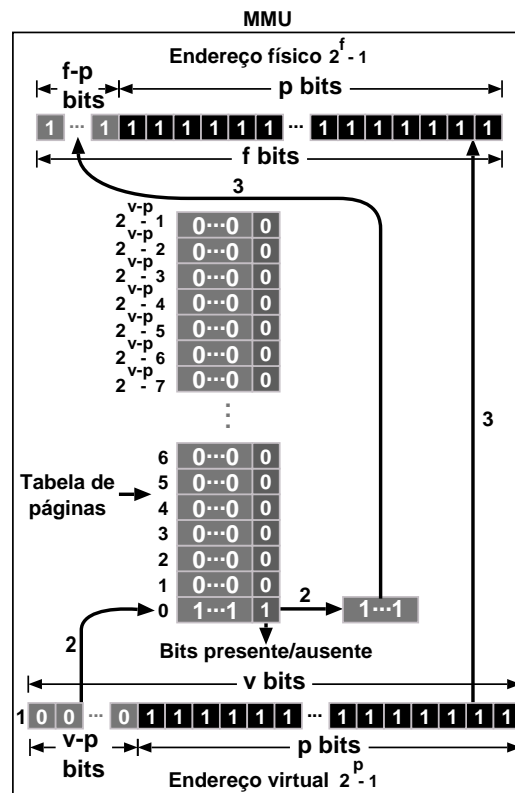
Resp.: Como podemos ver pelo grafo de recursos (e) dado na resposta do item anterior, existe um impasse devido ao ciclo orientado $A \rightarrow T \rightarrow C \rightarrow R \rightarrow B \rightarrow S \rightarrow A$. Para evitar especificamente este impasse, teria bastado que um dos recursos do ciclo, o R, o S ou o T, passasse a ser preemptivo e compartilhável, como exemplificado no grafo de recursos dado na parte esquerda da figura a seguir, no qual R é preemptivo e compartilhável. Isto permitiria que R fosse alocado simultaneamente a B e C. Note, pelas arestas destacadas, que o ciclo descrito anteriormente deixaria de existir no grafo. Porém, tornar somente um recurso preemptivo e compartilhável não é suficiente para evitar qualquer impasse, pois é possível existir um impasse envolvendo somente dois ou três recursos. Por exemplo, se partindo da mesma alocação inicial A solicitasse U e B solicitasse S, teríamos um impasse envolvendo A, B, S e U, devido ao ciclo orientado $A \rightarrow U \rightarrow B \rightarrow S \rightarrow A$. Este exemplo pode ser visto na parte direita da figura a seguir. Logo, para evitar qualquer impasse, três recursos devem se tornar preemptivos e compartilháveis.



2. (2,0) Suponha que o tamanho da memória virtual seja de 2^v bytes, que o tamanho da memória física seja de 2^f bytes, e que o tamanho de cada página virtual seja de 2^p bytes, sendo que $p < f < v$. Mostre uma figura, similar à dada no slide 20 da aula 8, com a conversão do endereço virtual $2^p - 1$ para o endereço físico correspondente, supondo que a página virtual com esse endereço seja mapeada na moldura de página com o maior número possível, e que inicialmente nenhuma página esteja mapeada na memória. Justifique a sua resposta.

Resp.: Como o tamanho da memória virtual é de 2^v bytes, ou seja, cada endereço virtual tem v bits, e como o tamanho da página virtual é de 2^p bytes, ou seja, são necessários p bits para representar todos os possíveis deslocamentos dentro de uma página, então cada endereço

virtual é dividido do seguinte modo: campo número da página virtual com os $v - p$ bits mais significativos; e campo deslocamento com os p bits menos significativos. Como o tamanho da moldura de página também é de 2^p bytes, e como o tamanho da memória física de 2^f bytes implica que cada endereço físico tem f bits, então, de modo similar ao endereço virtual, o endereço físico é dividido no campo número da moldura de página com os $f - p$ bits mais significativos e no campo deslocamento com os p bits menos significativos. Agora, como o endereço virtual a ser convertido é $2^p - 1$, o maior endereço da página virtual 0, então todos os $v - p$ bits do campo número da página virtual desse endereço são iguais a 0, e todos os bits do campo deslocamento são iguais a 1. Como pelo enunciado a página 0 é mapeada na moldura de página com o maior número possível, ou seja, $2^{f-p} - 1$, pois os números das molduras variam de 0 até $2^{f-p} - 1$, então todos os bits do campo número da moldura de página do endereço físico são iguais a 1 o que, por sua vez, implica que o endereço físico é $2^f - 1$, já que todos os bits do campo deslocamento também são iguais a 1. A figura resultante é mostrada a seguir.



3. (2,0) Suponha que um processo tenha acessado as páginas virtuais 1, 3, 2, 1, 4, 3, 2, 1, 4 e 2. Se três molduras de página, inicialmente vazias, estiverem disponíveis para serem usadas pelo processo, quantas falhas de página serão geradas se o algoritmo LRU for usado pelo sistema operacional? Suponha agora que o algoritmo de segunda chance passe a ser usado, sendo que o bit **referenciada** de cada página é ligado sempre que ela é acessada, e desligado ou quando duas páginas diferentes dela são acessadas em sequência ou quando o próprio algoritmo desliga o bit. Se agora quatro molduras de página, inicialmente vazias, estiverem disponíveis para o processo, e além disso a sequência de acesso for 1, 3, 5, 2, 1, 4, 3, 5, 2, 1, 4 e 2, o número de falhas de página será o mesmo? Justifique a sua resposta.

Resp.: -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 ocorrerá ao acessarmos as páginas na ordem dada no enunciado. Para cada uma dessas linhas mostramos, na primeira coluna, a página que será acessada. Já na segunda coluna mostramos a ordem em que as páginas deverão ser escolhidas para substituição. Finalmente, na última coluna, mostramos se o acesso à página gerará uma falha de página. Como podemos ver pela tabela, serão geradas 8 falhas de página.

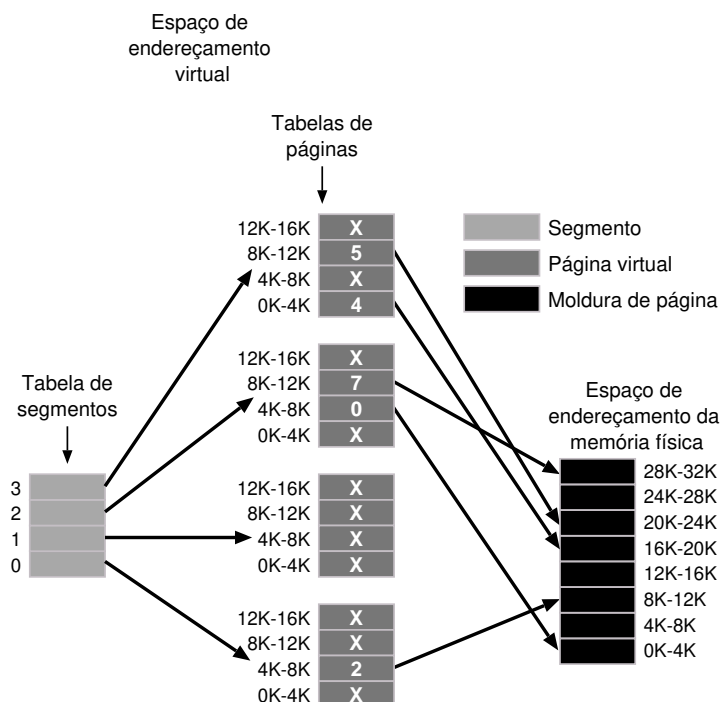
Páginas	Ordenação			Ocorreu uma falha?
1	1			Sim
3	1	3		Sim
2	1	3	2	Sim
1	3	2	1	Não
4	2	1	4	Sim
3	1	4	3	Sim
2	4	3	2	Sim
1	3	2	1	Sim
4	2	1	4	Sim
2	1	4	2	Não

-Agora vamos mostrar como as páginas serão substituídas de acordo com o algoritmo de segunda chance, usando o critério de atualização do bit **referenciada** dado no enunciado da questão. Como vimos na aula 9, nesse algoritmo, 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, se o seu bit **referenciada** for 0. Se o bit **referenciada** da página copiada há mais tempo na memória for 1, esse bit se tornará 0, a página deixará de ser a escolhida e será colocada no final da ordenação, e a próxima página da ordenação será considerada. A tabela a seguir é similar à anterior mas informamos, adicionalmente, para cada página dada na ordenação da segunda coluna, em parênteses, o valor atual do bit **referenciada**. Além disso, quando o bit **referenciada** da página copiada

há mais tempo na memória for 1, mostramos os passos feitos pelo algoritmo até ele escolher uma página com o bit **referenciada** igual a 0. Como podemos ver pela tabela, agora serão geradas 10 falhas de página, implicando portanto que o número de falhas gerado terá sido diferente do anterior.

Página	Ordenação	Falha?
1	1(1)	Sim
3	1(1) 3 (1)	Sim
5	1(0) 3 (1) 5 (1)	Sim
2	1(0) 3 (0) 5 (1) 2 (1)	Sim
1	1(1) 3 (0) 5 (0) 2 (1)	Não
4	3(0) 5 (0) 2 (0) 1 (0)	Sim
	5(0) 2 (0) 1 (0) 4 (1)	
3	2(0) 1 (0) 4 (1) 3 (1)	Sim
5	1(0) 4 (0) 3 (1) 5 (1)	Sim
2	4(0) 3 (0) 5 (1) 2 (1)	Sim
1	3(0) 5 (0) 2 (1) 1 (1)	Sim
4	5(0) 2 (0) 1 (1) 4 (1)	Sim
2	5(0) 2 (1) 1 (0) 4 (1)	Não

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:



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

Resp.: Primeiramente vamos notar que cada página virtual dada na figura do enunciado tem 4K ou 4096 endereços virtuais. Logo, o número total de endereços que geram falhas, para cada segmento, é o número de páginas não mapeadas deste segmento multiplicado por 4096. Como os números de páginas que geram falhas para os segmentos 0, 1, 2 e 3 são, respectivamente, 3, 4, 2 e 2, então o número total de endereços que geram falhas são, respectivamente, 12288, 16384, 8192 e 8192.

- (b) (0,5) Quais faixas de endereços físicos não estão mapeando faixas de endereços virtuais?

Resp.: Pela figura dada no enunciado, vemos que as molduras de página 1, 3 e 6 não estão mapeando páginas virtuais. Logo, as

faixas 4K–8K (endereços de 4096 até 8191), 12K–16K (endereços de 12288 até 16383) e 24K–28K (endereços de 24576 até 28671) não estão mapeando faixas de endereços virtuais.

- (c) (0,5) Para o endereço virtual (1,10537), mostre a divisão, em binário, nos campos número da página virtual e deslocamento dentro desta página. Caso a página virtual encontrada não esteja mapeada em nenhuma moldura, considere todos os mapeamentos possíveis usando molduras livres e, para cada um, mostre o endereço físico segundo este mapeamento e também a divisão desse endereço, em binário, nos campos número da moldura de página e deslocamento dentro desta moldura.

Resp.: Pela figura, vemos que o espaço de endereçamento de cada segmento tem 16K, e que cada endereço tem 14 bits. Este espaço de endereçamento, por sua vez, é dividido em 4 páginas virtuais com 4K de tamanho, numeradas de 0 até 3. Logo, para um dado endereço de um segmento, os 2 bits mais significativos definem o número da página virtual, e os 12 bits menos significativos definem o deslocamento dentro desta página. Já em relação ao espaço de endereçamento físico, vemos, pela figura, que este espaço de endereçamento tem 32K, e que cada endereço físico tem 15 bits. Como o tamanho de cada moldura de página é também de 4K, então os 3 bits mais significativos do endereço físico definem o número da moldura de página, e os 12 bits menos significativos definem o deslocamento dentro desta moldura. Agora, convertendo o endereço 10537 do segmento 1 para binário, obtemos $10 \mid 100100101001$, onde “|” é usado para separar os campos número da página virtual e deslocamento. Pelos campos, vemos que o endereço 10537 está na página virtual 2 (10 em binário), e que o deslocamento desse endereço dentro da página 1 é 2345 (100100101001 em binário). Pela figura, vemos que a página 2 do segmento 1 não está mapeada em uma moldura. Logo, precisaremos considerar todas as molduras livres, que pela figura são a 1, a 3, e a 6. Na tabela a seguir mostramos, na primeira coluna, cada uma das possíveis molduras em decimal e, entre parênteses, em binário. Já na segunda coluna mostramos o endereço físico

correspondente em binário, usando “|” para separar os campos número da moldura de página e deslocamento. Finalmente, na terceira coluna, mostramos o endereço dado na segunda coluna em decimal.

Moldura de página	Endereço físico em binário	Endereço físico em decimal
1 (001)	001 100100101001	6441
3 (011)	011 100100101001	14633
6 (110)	110 100100101001	26921

5. (1,5) Considere um disco com 16 blocos de b KB, numerados de 0 até 15. Suponha que um arquivo A, com 8 blocos, seja o único armazenado no disco. Considere também que os blocos lógicos pares do arquivo estejam associados, em ordem crescente, aos blocos iniciais do disco, e que os blocos lógicos ímpares estejam associados, em ordem decrescente, aos blocos finais do disco. Responda, justificando a sua resposta:

- (a) (1,0) Se a alocação por lista encadeada utilizando um índice for usada para gerenciar os blocos usados do disco, como será preenchida a tabela?

Resp.: Pelo enunciado, os blocos lógicos são associados aos blocos físicos como na tabela a seguir:

	blocos							
lógicos	0	1	2	3	4	5	6	7
físicos	0	15	1	14	2	13	3	12

Assim, a tabela relativa ao método de alocação será a seguinte:

Ordem dos blocos:

Arquivo A: 0, 15, 1, 14, 2, 13, 3 e 12.

← 4 bits →	
0	15
1	14
2	13
3	12
4	L
5	L
6	L
7	L
8	L
9	L
10	L
11	L
12	X
13	3
14	2
15	1

← Início do arquivo A

← Fim do arquivo A

Nessa tabela, “L” representa um bloco livre e um “X” o final do arquivo.

- (b) (0,5) Suponha agora que um arquivo B seja armazenado em blocos consecutivos do disco. Qual será o menor número para o primeiro bloco de B, e qual será o maior tamanho possível para B em KB?

Resp.: Como o arquivo B usará blocos consecutivos do disco, então B deverá começar no bloco 4 ou depois e terminar no bloco 11 ou antes. Logo, o menor número para o primeiro bloco de B será 4. Assim, o maior tamanho para B será $(11 - 4 + 1)b$ KB, ou seja, $8b$ KB.

6. (1,5) 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 estudada na aula 12. Responda:

Objetos

A: Aula012 **B:** Impressora
C: GabaritoAD2 **D:** Aula010
E: ListaAula12 **F:** Plotter
G: GravadorDVD **H:** Médias
I: Domínio 1 **J:** Domínio 2
K: Domínio 3

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

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

Resp.: -O domínio 1 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 1 possua alguma das operações **R**, **W** ou **X**. Logo, podem ser acessados os seguintes objetos: A (Aula012), C (GabaritoAD2), D (Aula010), F (Plotter) e H (Médias).

-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 ler o objeto A, ou seja, podemos somente ler o arquivo Aula012;
- Podemos somente executar o objeto C, ou seja, podemos somente executar o arquivo GabaritoAD2;
- Podemos salvar dados no objeto D ou executá-lo, ou seja, podemos escrever ou executar o arquivo Aula010;
- Podemos ler ou salvar dados no objeto F, ou seja, podemos ler ou escrever no Plotter. A leitura de dados do Plotter não está incorreta, pois podemos estar tentando obter, por

exemplo, informações sobre o estado do Plotter. Já a operação salvar está correta porque para o objeto F isso significa que desejamos imprimir no Plotter;

- Podemos ler, salvar, ou executar o objeto H, ou seja, podemos ler, escrever ou executar o arquivo Médias.

(b) (0,5) Estando no domínio 2, 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 2 associadas a objetos que sejam domínios e que possuam a operação **Enter**. Pela matriz, vemos que podemos passar do domínio 2 para os domínios 1 (objeto I) e 3 (objeto K).