



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 2016

**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 -

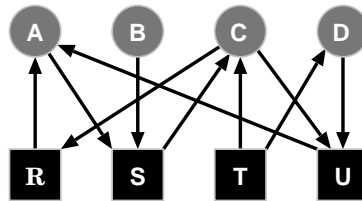
---

1. (1,5) Suponha que quatro processos, A, B, C e D, estejam em execução no sistema operacional, e que existam três recursos não-preemptivos, R, S e U, e um recurso T preemptivo e compartilhável, inicialmente não alocados a nenhum dos processos. Suponha ainda que um processo possa requisitar mais de um recurso ao mesmo tempo, ficando bloqueado até que todos os recursos não-preemptivos tenham sido obtidos. Alguma das duas sequências de alocação dadas a seguir irá gerar um impasse? Justifique a sua resposta.

(a) (0,8) Sequência 1:

- A requisita R e U;
- C requisita S e T;
- D requisita T e U;
- A requisita S;
- C requisita R e U;
- B requisita S.

**Resp.:** Na figura a seguir mostramos o grafo de recursos obtido após a sequência de alocações dada no enunciado do item (a) ter ocorrido. Como o recurso T é preemptivo e compartilhável, C e D podem usá-lo ao mesmo tempo. Pela figura, teremos impasses devidos a dois ciclos orientados: o ciclo orientado R-A-S-C-R, porque A deseja o recurso não-preemptivo S que C possui e C deseja o recurso não-preemptivo R que A possui; e o ciclo orientado U-A-S-C-U, porque A deseja o recurso não-preemptivo S que C possui e C deseja o recurso não-preemptivo U que A possui.

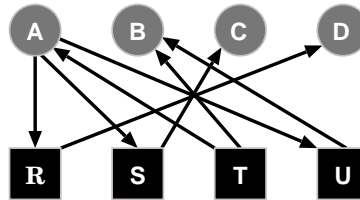


(b) (0,7) Sequência 2:

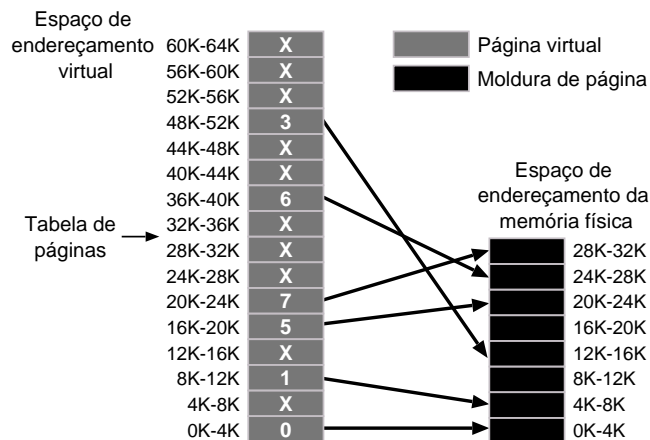
- B requisita U;
- A requisita T;
- D requisita R;
- C requisita S;
- B requisita T;
- A requisita R, S e U.

**Resp.:** A figura a seguir mostra o grafo de recursos obtido após a sequência de alocações dada no enunciado do item (b) ter ocorrido, sendo que mais uma vez o recurso T pode ser alocado a mais de um processo devido a ser preemptivo e compartilhável. Pela figura, vemos que agora não teremos impasses porque não existem

ciclos orientados no grafo.



2. (1,5) Utilizando a tabela de páginas dada a seguir, vista no final da aula 8, forneça o endereço físico correspondente a cada um dos endereços virtuais abaixo, supondo que todas as molduras estejam alocadas ao processo que acessou as páginas. Se o endereço não estiver na memória, então escolha uma das molduras livres para mapeá-lo:



- (a) (0,5) 12321.  
 (b) (0,5) 53158.  
 (c) (0,5) 12.

**Resp.:** Pela figura vemos que o espaço de endereçamento virtual 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 virtual é dividido em 16 páginas virtuais, e o espaço de endereçamento físico é dividido em 8 molduras de página. Assim, cada

endereço virtual é dividido em 4 bits para o número da página e 12 bits para o deslocamento, e cada endereço físico em 3 bits para a moldura e também 12 bits para o deslocamento. Na tabela a seguir mostramos cada endereço virtual em decimal e binário (com o número da página e o deslocamento separados por “|”), a página virtual que contém esse endereço (com sua faixa de endereços virtuais entre parênteses), e o deslocamento dentro dela.

Endereço virtual	Página virtual	Deslocamento
12321 (0011   000000100001)	3 (12288–16383)	33
53158 (1100   111110100110)	12 (49152–53247)	4006
12 (0000   000000001100)	0 (0–4095)	12

Pela figura vemos que as páginas virtuais 0 e 12 estão mapeadas, respectivamente, nas molduras de página 0 e 3, enquanto que a página virtual 3 não está mapeada em uma moldura de página, implicando que o endereço virtual 12321 não está mapeado em um endereço físico. Como duas molduras, a 2 e a 4, estão livres, então uma delas pode ser a escolhida na resposta para a questão. Na tabela a seguir mostramos para cada endereço virtual, a moldura (com o endereço físico inicial dela dado entre parênteses), o deslocamento dentro da moldura, igual ao deslocamento dentro da página virtual, e o endereço físico obtido pela soma do endereço físico inicial da moldura com o deslocamento. Para o endereço virtual 12321, mostramos duas linhas na tabela, para cada uma das possíveis molduras que poderão ser escolhidas quando a falha de página ocorrer.

Endereço virtual	Moldura de página	Deslocamento	Endereço físico
12321	2 (8192)	33	8225
	4 (16384)	33	16417
53158	3 (12288)	4006	16294
12	0 (0)	12	12

- (2,0) Suponha que um processo tenha acessado, em ordem, as páginas virtuais 1, 2, 0, 1, 0, 2, 1, 1, 0 e 2, e que tenham sido alocadas 2 molduras, inicialmente vazias, para o processo. Suponha ainda que o bit

**referenciada** de uma página seja ligado quando ela é acessada e permaneça ligado até duas páginas diferentes dela terem sido acessadas. Se o algoritmo NRU for usado, quais serão as páginas na memória, após os acessos dados no enunciado, supondo que todas as páginas sejam alteradas somente quando acessadas pela segunda vez? E quantas falhas cada uma das páginas acessadas irá gerar? Justifique a sua resposta.

**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 duas 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. Na 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 por esta tabela, as páginas na memória após os acessos serão a 0 e a 2, e o número total de falhas para as páginas 0, 1 e 2 serão de, respectivamente, 2, 3 e 3.

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

4. (1,5) Suponha que a memória de um computador tenha tamanho de 4GB. Suponha ainda que existam quatro processos, A, B, C e D, com códigos de, respectivamente, 77MB, 768MB, 500MB e 2GB. Se o hardware permitir páginas com tamanhos de 8MB, 32MB, 256MB e 1GB, então para quais desses tamanhos poderíamos fazer todos os códigos dos quatro processos caberem simultaneamente na memória? Justifique a sua resposta.

**Resp.:** Na tabela a seguir mostramos o número de páginas necessárias para cada um dos tamanhos de página dados no enunciado. Na primeira coluna mostramos o tamanho da página. Na segunda coluna mostramos o número total de páginas para o tamanho dado na primeira coluna. Nas colunas de 3 até 6 mostramos a quantidade de páginas usadas pelos processos. Finalmente, na última coluna, mostramos o número total de páginas necessárias para armazenar simultaneamente os processos

A, B, C e D na memória (igual à soma das quantidades de páginas nas colunas de 3 até 6). Como podemos ver pela tabela, não poderemos armazenar todos os códigos dos processos somente para o tamanho de página de 1GB.

Tamanho da página	Número de páginas	Processos				Total de páginas
		A	B	C	D	
8MB	512	10	96	63	256	425
32MB	128	3	24	16	64	107
256MB	16	1	3	2	8	14
1GB	4	1	1	1	2	5

5. (2,0) Considere um computador com um disco de 32 blocos de 32KB, numerados de 0 até 31. Suponha que o disco esteja inicialmente vazio. Responda, justificando a sua resposta:

- (a) (1,0) Se o sistema operacional usar a alocação contígua, e se um arquivo A com 520KB for armazenado a partir do bloco  $i$ , qual será o tamanho máximo que um arquivo B poderá ter em função de  $i$ ?

**Resp.:** Como o tamanho de cada bloco é de 32KB e o tamanho do arquivo A é de 520KB, então A usará 17 blocos consecutivos do disco (sendo o último bloco não usado integralmente), significando que A usará os blocos de  $i$  até  $i + 16$ ,  $0 < i < 16$ . Como o arquivo B poderá ser armazenado consecutivamente nos  $i - 1$  blocos iniciais do disco, ou nos blocos de  $i + 17$  até 31, então o tamanho máximo de B é igual a  $(32 \times \max\{i, 15 - i\})$ KB.

- (b) (1,0) Qual será a tabela obtida se o sistema operacional usar a alocação por lista encadeada utilizando um índice para armazenar os seguintes três arquivos simultaneamente? Arquivo A, usando em ordem crescente todos os blocos com número par a partir do bloco 15 do disco; arquivo B, usando os 8 primeiros blocos do disco na ordem dada pela alocação contígua; e arquivo C, usando em

ordem decrescente os quatro últimos blocos ímpares do disco.

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

Ordem dos blocos:

arquivo A: 16, 18, 20, 22, 24, 26, 28 e 30.

arquivo B: 0, 1, 2, 3, 4, 5, 6 e 7.

arquivo C: 31, 29, 27 e 25.

6. (1,5) Suponha que o sistema operacional use um mapa de bits para gerenciar os blocos livres de um disco com  $n$  blocos, numerados de 0 até  $n - 1$ , e que a alocação contígua seja usada ao gerenciar os blocos



usados. Suponha ainda que três arquivos, A, B e C, estejam armazenados no disco e que tenham, respectivamente,  $a$ ,  $b$  e  $c$  blocos, sendo  $a + b + c \leq n$ . Como será o mapa de bits do disco se A for armazenado no início do disco, B no final do disco, e C a partir do bloco  $i$  do disco, sendo  $a \leq i \leq n - b - c$ ? Justifique a sua resposta.

**Resp.:** Como vimos na aula 11, quando a alocação contígua é usada, cada arquivo é armazenado em um conjunto de blocos consecutivos do disco. Com isso, o arquivo A, com tamanho de  $a$  blocos e armazenado no início do disco, usa os blocos de 0 até  $a - 1$ . Já o arquivo B, com  $b$  blocos e armazenado no final do disco, usa os blocos de  $n - b$  até  $n - 1$ . Finalmente o arquivo C, com  $c$  blocos e armazenado a partir da posição  $i$  do disco, usa os blocos de  $i$  até  $i + c - 1$ . Os blocos restantes, de  $a$  até  $i - 1$ , se  $i > a$ , e de  $i + c$  até  $n - b - 1$ , se  $i < n - b - c$ , estão livres. Agora, como no mapa de bits existe um bit para cada bloco do disco, e como o valor desse bit é 0 se o bloco está usado e 1 se está livre, os bits de 0 até  $a - 1$ , de  $i$  até  $i + c$  e de  $n - b$  até  $n - 1$  são todos iguais a 0, enquanto que os bits de  $a$  até  $i - 1$ , se  $i > a$ , e de  $i + c$  até  $n - b - 1$ , se  $i < n - b - c$ , são todos iguais a 1. Veja a figura a seguir.

