



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 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.

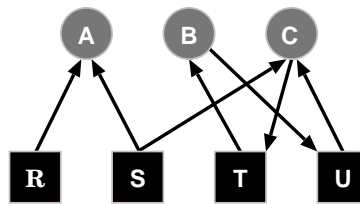
Nome -
Assinatura -

1. (1,0) Suponha que um processo A tenha obtido os recursos R e S, que um processo B tenha obtido o recurso T, e que um processo C tenha obtido os recursos S e U. Diga como um impasse poderá ocorrer nos seguintes cenários, usando o grafo de recursos ao justificar a sua resposta.

(a) (0,5) B tenta obter U e C tenta obter T.

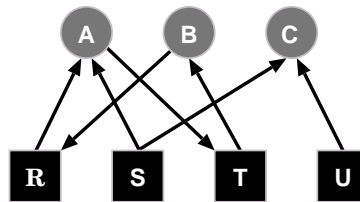
Resp.: Como B obteve T com sucesso e C obteve U com sucesso, então teremos um impasse se T e U forem não-preemptivos, pois

o grafo de recursos, como podemos ver na figura a seguir, terá o ciclo B-U-C-T-B. Note que se pelo menos um dos dois recursos for preemptivo, não teremos um impasse porque o recurso preemptivo, se for compartilhável, poderá ser alocado simultaneamente a B e C e, em caso contrário, poderá ser removido do processo que o possui e alocado ao processo que o deseja, evitando assim o impasse.



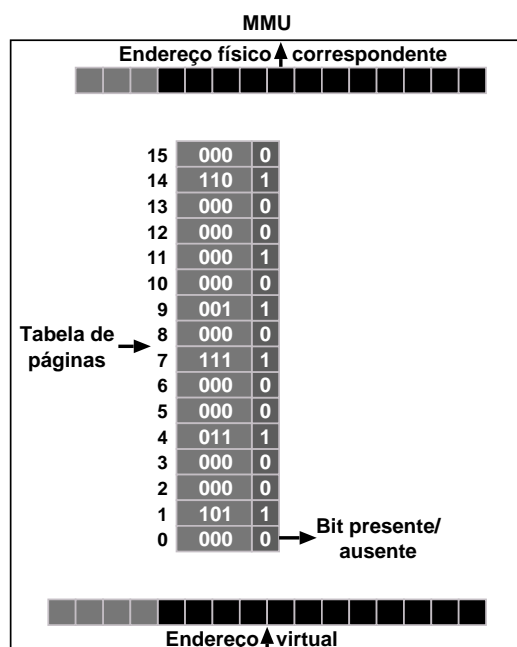
(b) (0,5) A tenta obter T, B tenta obter R e C tenta obter S.

Resp.: Assim como no item anterior, teremos um ciclo, A-T-B-R-A, como podemos ver na figura a seguir. E, assim como antes, somente teremos um impasse se R e T forem ambos não-preemptivos, sendo que se pelo menos um for preemptivo, poderemos usar o que foi descrito no item (a) para evitar o impasse. No caso de C tentar obter S, não teremos nenhum problema porque, como podemos ver pelo enunciado, C já possui S (que precisa ser preemptivo e compartilhável, devido a ter sido alocado também a A), e se C tentar obter S novamente, o sistema operacional poderá, caso S não seja dividido entre os processos e uma nova parte de S não possa ser obtida por C, gerar um erro.



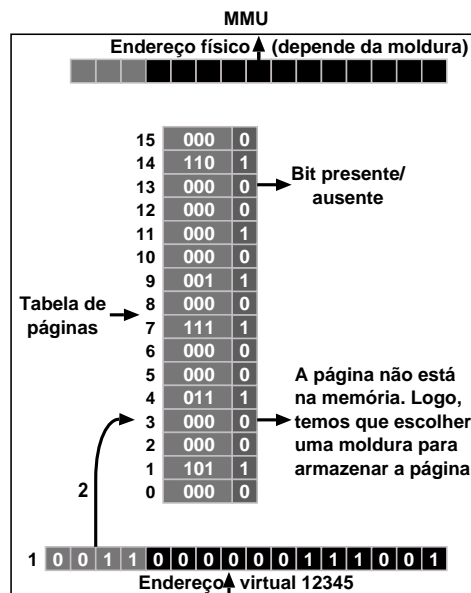
2. (2,0) A figura dada a seguir, vista na aula 8, possui algumas das informações necessárias à conversão dos endereços virtuais nos endereços

físicos correspondentes da memória. Gere duas versões da figura, para os endereços virtuais 12345 e 40131, descrevendo os passos executados pela MMU ao converter cada endereço virtual no endereço físico correspondente. Se você achar que o acesso a um dos endereços gera uma falha de página então, antes de mapear esse endereço, escolha uma das molduras disponíveis e depois mapeie a página virtual desse endereço na moldura escolhida.

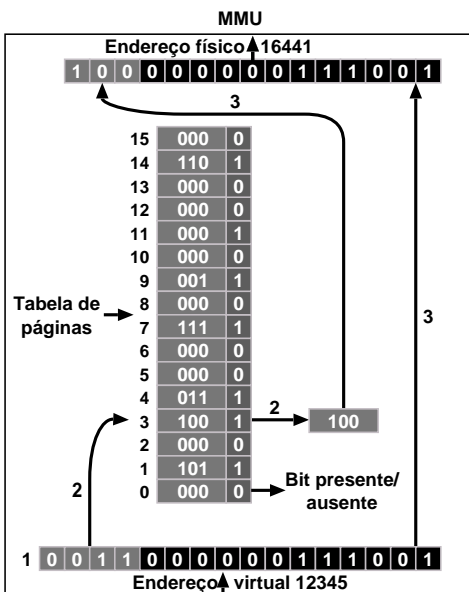
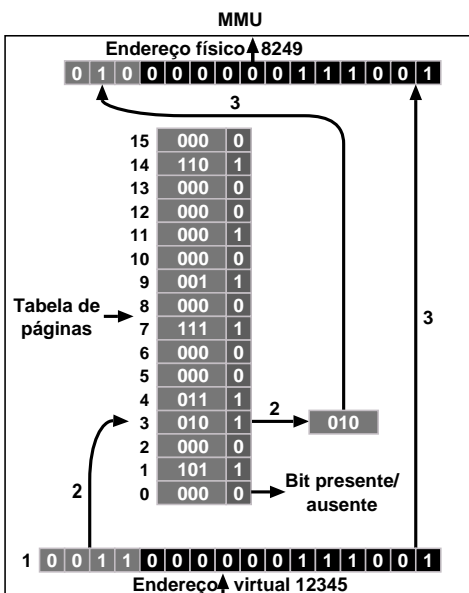


Resp.: Vamos primeiramente converter o endereço virtual 12345. Para determinar o endereço físico, primeiramente devemos converter o endereço virtual 12345 para binário, obtendo o valor 0011000000111001. Isso significa que o endereço está na página de número decimal 3 e que o deslocamento dentro dessa página é 57. Agora, como podemos ver pela figura dada no enunciado, a página virtual 3 não está armazenada na memória, porque o seu bit presente/ausente é 0. Nesse caso, antes de o processador finalizar a conversão do endereço virtual para o endereço físico (e, com isso, executar os passos 3 e 4 vistos na aula 8), o sistema operacional precisará escolher a moldura de página que deverá armazenar a página virtual. Como existem duas molduras de página não usadas (a 2 e a 4), então teremos um dos possíveis casos

dados na tabela a seguir. Na tabela, cada endereço está associado a uma das molduras de página. Em cada coluna da tabela damos, entre parênteses, o valor binário correspondente ao valor decimal. As figuras representando cada um dos casos são dadas após a tabela, lembrando que para a resposta ser considerada correta basta somente dar um dos casos.

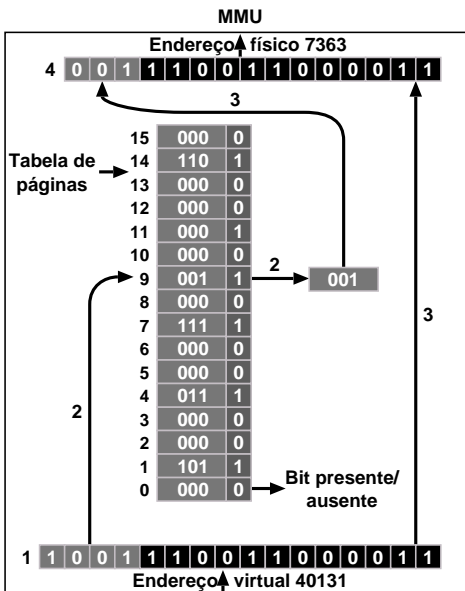


Página virtual	Endereço físico
2 (010)	8249 (010000000111001)
4 (100)	16441 (100000000111001)



Agora, ao converter o endereço virtual 40131 para binário, obtemos o valor 1001110011000011. Isso significa que o endereço está na página de número decimal 9, que está na memória e mapeada na moldura de página 1, pois o seu bit presente/ausente é 1, e que o deslocamento den-

tro dessa página é 3267. A seguir damos a figura para esse endereço virtual, onde vemos que o endereço físico é, em binário, 001110011000011, ou seja, o endereço é o 7363 da moldura de página 1.



3. (3,0) 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 o processo acesse as páginas da tabela na mesma ordem, e que o mesmo número de molduras, inicialmente vazias, esteja disponível para o processo:

Páginas	Ordenação	Ocorreu uma falha?
2	2	Sim
3	2 3	Sim
4	2 3 4	Sim
5	2 3 4 5	Sim
3	2 3 4 5	Não
0	3 4 5 0	Sim
2	4 5 0 2	Sim
1	5 0 2 1	Sim
4	0 2 1 4	Sim
6	2 1 4 6	Sim

- (a) (1,5) Se o sistema operacional passar a usar o algoritmo LRU, teremos o mesmo número de falhas de página?

Resp.: -Primeiramente 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 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, teremos 9 falhas de página, portanto o mesmo número.

Páginas	Ordenação	Ocorreu uma falha?
2	2	Sim
3	2 3	Sim
4	2 3 4	Sim
5	2 3 4 5	Sim
3	2 4 5 3	Não
0	4 5 3 0	Sim
2	5 3 0 2	Sim
1	3 0 2 1	Sim
4	0 2 1 4	Sim
6	2 1 4 6	Sim

- (b) (1,5) Se o sistema operacional passar a usar o algoritmo de idade, sendo que os contadores têm 4 bits e que uma interrupção é gerada a cada dois acessos às páginas, teremos a mesma sequência de substituições?

Resp.: -Agora, como vimos na aula 9, o algoritmo de idade define, para cada página que está na memória, um contador, no nosso caso com 4 bits, que é inicializado em 0 quando a página é copiada para a memória. A cada interrupção do temporizador, ou seja, a cada dois acessos às páginas, que estão na memória, o seu contador é deslocado um bit para a direita e o seu bit **referenciada** é copiado para o bit mais significativo desse contador. Quando precisarmos substituir uma página, a página com o menor contador será a escolhida para ser substituída. Como o enunciado não diz que o sistema operacional reseta o bit referenciada das páginas, e como ele sempre é ligado quando a página é acessada, então o bit nunca é desligado. Nas figuras dadas a seguir mostramos, na parte (a), o estado inicial antes de acessarmos qualquer página e, nas partes de (b) até (k), os valores dos contadores após acessarmos as páginas na ordem dada no enunciado (a linha da página acessada é destacada em negrito). Também damos, entre parênteses, após o número de cada página, o valor atual do seu bit **referenciada**. Também, nas partes de (e) até (k) da figura, as páginas escolhidas para serem substituídas são indicadas por uma seta. Note que nos três primeiros acessos não indicamos esta página, porque pela ta-

bela do enunciado notamos que quatro molduras foram alocadas ao processo. Também dizemos nas figuras, em cada parte com exceção da (a), se o acesso à página gerou ou não uma falha. Temos duas possibilidades devido ao empate ocorrido na parte (f) para os contadores das páginas 4 e 5. Pela figuras, vemos que as possíveis seqüências de substituições são, respectivamente, 4, 0, 1, 4 e 5, 0, 1 e, pelo enunciado, vemos que a seqüência original é 2, 4, 5, 3, 0. Logo, a seqüência de substituições não é igual em nenhum dos casos.

Páginas

- -
- -
- -
- -

Falha: -
(a)

Páginas

2 (1) 0000
- -
- -
- -

Falha: Sim
(b)

Páginas

2 (1) 1000
3 (1) 1000
- -
- -

Falha: Sim
(c)

Páginas

2 (1) 1000
3 (1) 1000
4 (1) 0000
- -

Falha: Sim
(d)

Páginas

2 (1) 1100
3 (1) 1100
4 (1) 1000 \Leftarrow
5 (1) 1000 \Leftarrow

Falha: Sim
(e)

Páginas

2 (1) 1100
3 (1) 1100
4 (1) 1000 \Leftarrow
5 (1) 1000 \Leftarrow

Falha: Não
(f)

Páginas

2 (1) 1110
3 (1) 1110
0 (1) 1000 \Leftarrow
5 (1) 1100

Falha: Sim
(g)

Páginas

2 (1) 1110
3 (1) 1110
0 (1) 1000 \Leftarrow
5 (1) 1100

Falha: Não
(h)

Páginas

2 (1) 1111
3 (1) 1111
1 (1) 1000 \Leftarrow
5 (1) 1110

Falha: Sim
(i)

Páginas

2 (1) 1111
3 (1) 1111
4 (1) 0000 \Leftarrow
5 (1) 1110

Falha: Sim
(j)

Páginas

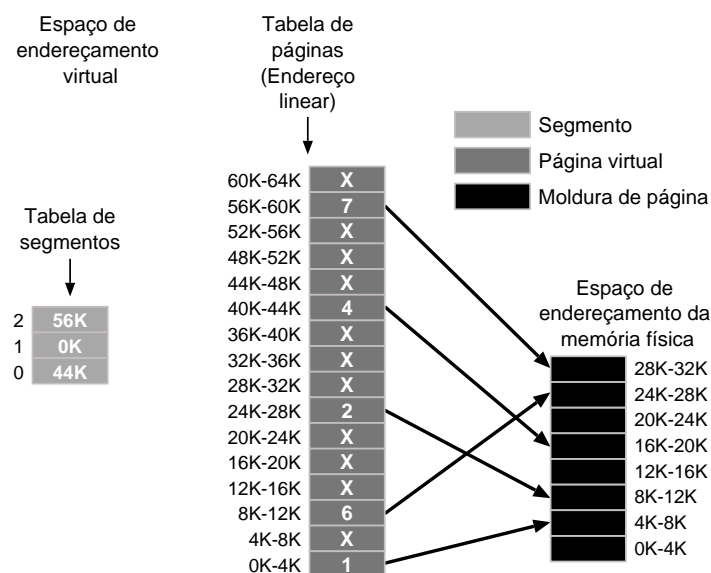
2 (1) 1111
3 (1) 1111
6 (1) 1000 \Leftarrow
5 (1) 1111

Falha: Sim
(k)

Páginas - - - - - - - -	Páginas 2 (1) 0000 - - - - - -	Páginas 2 (1) 1000 3 (1) 1000 - - - -
Falha: - (a)	Falha: Sim (b)	Falha: Sim (c)
Páginas 2 (1) 1000 3 (1) 1000 4 (1) 0000 - -	Páginas 2 (1) 1100 3 (1) 1100 4 (1) 1000 \Leftarrow 5 (1) 1000 \Leftarrow	Páginas 2 (1) 1100 3 (1) 1100 4 (1) 1000 \Leftarrow 5 (1) 1000 \Leftarrow
Falha: Sim (d)	Falha: Sim (e)	Falha: Não (f)
Páginas 2 (1) 1110 3 (1) 1110 4 (1) 1100 0 (1) 1000 \Leftarrow	Páginas 2 (1) 1110 3 (1) 1110 4 (1) 1100 0 (1) 1000 \Leftarrow	Páginas 2 (1) 1111 3 (1) 1111 4 (1) 1110 1 (1) 1000 \Leftarrow
Falha: Sim (g)	Falha: Não (h)	Falha: Sim (i)
Páginas 2 (1) 1111 3 (1) 1111 4 (1) 1110 1 (1) 1000 \Leftarrow	Páginas 2 (1) 1111 3 (1) 1111 4 (1) 1111 6 (1) 1000 \Leftarrow	
Falha: Não (j)	Falha: Sim (k)	

4. (1,5) Suponha que um computador utilize uma técnica de segmentação com paginação similar à do processador Pentium da Intel. Na figura a seguir mostramos, na tabela de segmentos, o endereço linear inicial no qual cada segmento usado por um processo é mapeado. Nesta figura

também mostramos, na tabela de páginas do espaço de endereçamento linear, como as páginas virtuais são mapeadas nas molduras de página da memória física. Usando essas tabelas, forneça o endereço linear e o endereço físico de cada um dos seguintes endereços virtuais bidimensionais (segmento, endereço dentro do segmento), usando todos as possíveis molduras livres caso uma falha de páginas ocorra:



- (a) (0,5) (0,9786).
- (b) (0,5) (1,20358).
- (c) (0,5) (2,7427).

Resp.: No Pentium, cada segmento é mapeado a partir do endereço inicial dado na tabela de segmentos. Todos os endereços iniciais dos segmentos pertencem a um espaço de endereçamento comum, chamado de **espaço de endereçamento linear**, que é paginado do mesmo modo que o espaço de endereçamento virtual visto na aula 9. Logo, o endereço linear correspondente a um endereço de um segmento é igual à soma deste endereço com o endereço linear a partir do qual o segmento é mapeado. Pela figura vemos que o espaço de endereçamento linear 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 linear é dividido em 16 páginas virtuais, e o espaço de endereçamento físico é dividido em 8 molduras de página. Assim, cada endereço linear é dividido do seguinte modo: 4 bits para o número da página e 12 bits para o deslocamento. Na tabela a seguir mostramos, para cada item (cada endereço virtual bidimensional), o endereço base do segmento, o endereço linear correspondente (igual a soma do endereço base mais a segunda componente do endereço virtual) em decimal e binário (com o número da página e o deslocamento separados por “|”) entre parênteses, a página virtual (com a faixa de endereços virtuais contidos na página entre parênteses) que contém este endereço, e o deslocamento dentro desta página virtual.

Endereço virtual	Endereço base	Endereço Linear	Página virtual	Deslocamento
(a) (0, 9786)	44K ou 45056	54842 (1101 011000111010)	13 (53248– 57343)	1594
(b) (1, 20358)	0K ou 0	20358 (0100 111110000110)	4 (16384– 20479)	3974
(c) (2, 7427)	56K ou 57344	64771 (1111 110100000011)	15 (61440– 65535)	3331

Como pela figura vemos que todas as páginas virtuais de todos os endereços lineares acessados não estão na memória, e como as molduras de página 0, 3 e 5 estão livres então, para cada endereço linear, teremos três possíveis endereços físicos, dados na tabela a seguir, na qual mostramos, para cada endereço linear, o deslocamento dentro da moldura, igual ao deslocamento dentro da página virtual, e o endereço físico para cada possível moldura, obtido pela soma do endereço inicial da moldura (dado entre parênteses para cada moldura) com o deslocamento.

Endereço linear	Deslocamento	Endereço físico		
		Molduras		
		0 (0)	3 (12288)	5 (20480)
(a) 54842	1594	1594	13882	22074
(b) 20358	3974	3974	16262	24454
(c) 64771	3331	3331	15619	23811

5. (1,5) Imagine que você deseje acessar as posições do arquivo ad2.pdf dado na figura a seguir. Quantas vezes cada um dos blocos será acessado usando o acesso sequencial, se desejarmos:

Arquivo ad2.pdf (11 posições)

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

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

Resp.: Quando usamos o acesso aleatório, as posições do arquivo podem ser acessadas independentemente umas das outras. 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.

- (a) 5 primeiros blocos em ordem crescente e depois
e os 6 últimos blocos em ordem decrescente

x	Precisam ser acessadas										
	1	2	3	4	5	6	7	8	9	10	11
1	X										
2		X									
3			X								
4				X							
5					X						
11						X	X	X	X	X	X
10	X	X	X	X	X	X	X	X	X	X	
9	X	X	X	X	X	X	X	X	X		
8	X	X	X	X	X	X	X	X			
7	X	X	X	X	X	X	X				
6	X	X	X	X	X	X					
Total	6	6	6	6	6	6	5	4	3	2	1

- (b) todos os blocos pares e depois
todos os blocos ímpares

x	Precisam ser acessadas										
	1	2	3	4	5	6	7	8	9	10	11
2	X	X									
4			X	X							
6					X	X					
8							X	X			
10									X	X	
1	X										
3		X	X								
5				X	X						
7						X	X				
9								X	X		
11										X	X
Total	2	2	2	2	2	2	2	2	2	2	1

(c) os blocos na ordem 1, 11, 3, 4, 8,
9, 10, 2, 5, 7 e 6

x	Precisam ser acessadas										
	1	2	3	4	5	6	7	8	9	10	11
1	X										
11		X	X	X	X	X	X	X	X	X	X
3	X	X	X								
4				X							
8					X	X	X	X			
9									X		
10										X	
2	X	X									
5			X	X	X						
7						X	X				
6	X	X	X	X	X	X					
Total	4	4	4	4	4	4	3	2	2	2	1

6. (1,0) Suponha que depois de uma verificação de consistência do sistema de arquivos, tenham obtido as seguintes tabelas que, como vimos na aula 12, contabilizam os blocos em utilização e os blocos livres, para um disco com 16 blocos. Responda:

Blocos do disco																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocos usados	0	1	0	3	1	0	4	0	2	1	0	1	1	0	1	1
Blocos livres	1	0	0	1	0	1	0	1	2	0	2	1	0	0	0	0

- (a) (0,5) Se existirem problemas de consistência neste sistema de arquivos, indique quais são eles.

Resp.: Existem sete problemas de consistência. Os blocos 2 e 13 do disco são blocos ausentes (blocos do disco que não podem ser usados) pois, em ambas as tabelas, os contadores deles são iguais a zero. O bloco 6 está associado a quatro arquivos diferentes pois, como podemos ver pela tabela Blocos usados, o seu

contador é igual a 4. Note que este problema é sério, pois certamente todos os arquivos estão corrompidos. O problema é pior ainda para os blocos 3 e 8 pois, além de eles estarem associados a, respectivamente, três e dois arquivos diferentes, também estão incorretamente marcados como livres na tabela Blocos livres, sendo que o bloco 8 ainda foi marcado duas vezes. Já o bloco 11 está associado a um arquivo e ao mesmo tempo está marcado como livre, o que pode fazer com que, no futuro, este bloco seja associado a um outro arquivo. Finalmente, o bloco 10 está marcado duas vezes como livre na tabela de Blocos livres, o que pode fazer, no futuro, ele ser associado a dois arquivos diferentes.

- (b) (0,5) Se existirem problemas de consistência, um deles deixaria de ocorrer se usássemos um mapa de bits para sabermos quais blocos estão livres no disco?

Resp.: Os únicos problemas que não poderiam ocorrer se usássemos um mapa de bits para gerenciar os blocos livres do disco seriam os blocos 8 e 10 estarem duplicados, com o contador igual a 2, na tabela Blocos livres. Isso é devido a nunca podermos ter os contadores da tabela Blocos livres com valores maiores do que 1 ao usarmos um mapa de bits, porque temos somente um bit associado a cada bloco do disco, que pode ser igual a 0 ou a 1.