



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  
AD2 - Segundo Semestre de 2008

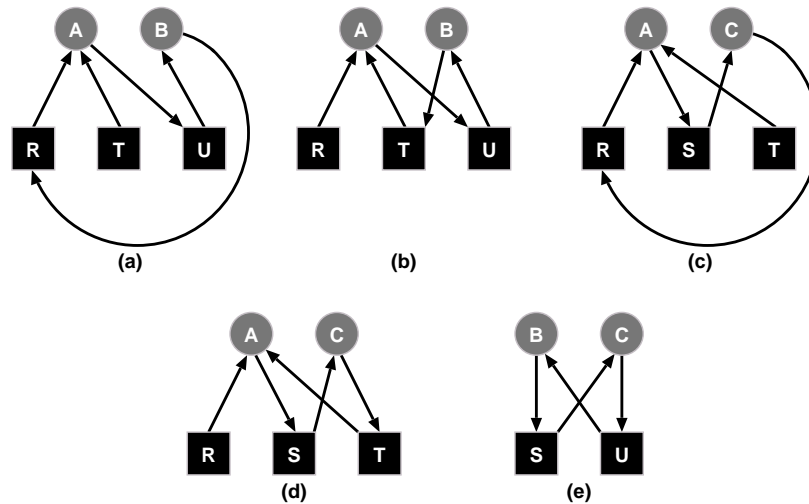
**Atenção:** Tem havido muita discussão sobre a importância de que cada aluno redija suas próprias respostas às questões da AD2. Os professores da disciplina, após refletirem sobre o assunto, decidiram o seguinte: Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras 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 estão compartilhando quatro recursos R, S, T e U. Suponha ainda que A possui R e T, B possui U e C possui S. O que ocorrerá se:
    - (a) (0.75) Todos os recursos forem não preemptivos e, para quaisquer dois processos, cada um solicitar todos os recursos do outro antes

de liberar os recursos que possui?

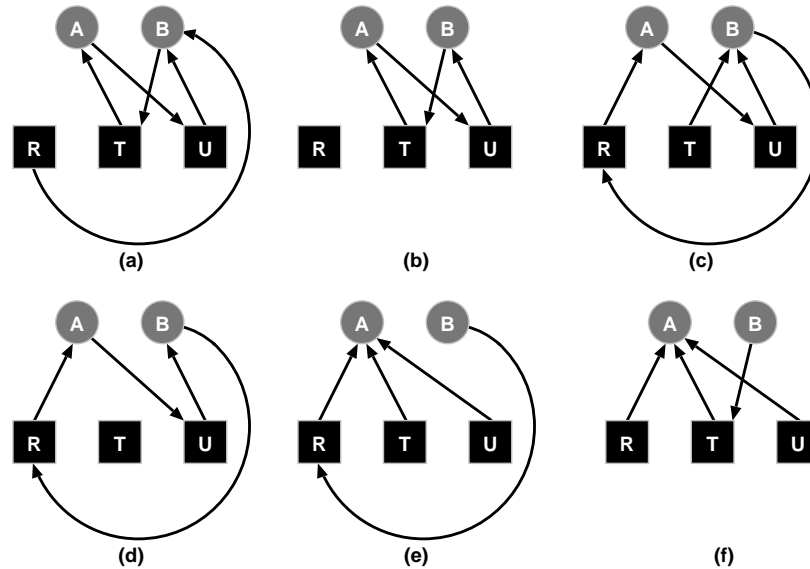
**Resp.:** Se todos os recursos forem não preemptivos, então sempre teremos um impasse, independentemente dos dois processos, pois nenhum deles liberará os recursos que possui, e cada um deles ficará bloqueado esperando pelo(s) recurso(s) do outro. Na figura a seguir mostramos os grafos de recursos com cada um destes impasses. Nas partes (a) e (b), envolvendo os processos A e B, mostramos o que ocorrerá se A solicitar U e B solicitar primeiramente R (parte (a)) ou T (parte (b)). Já nas partes (c) e (d), envolvendo os processos A e C, mostramos o que ocorrerá quando A solicitar S e C solicitar primeiramente R (parte (c)) ou T (parte (d)). Finalmente, na parte (e), envolvendo os processos B e C, mostramos o que ocorrerá se B solicitar S e C solicitar U.



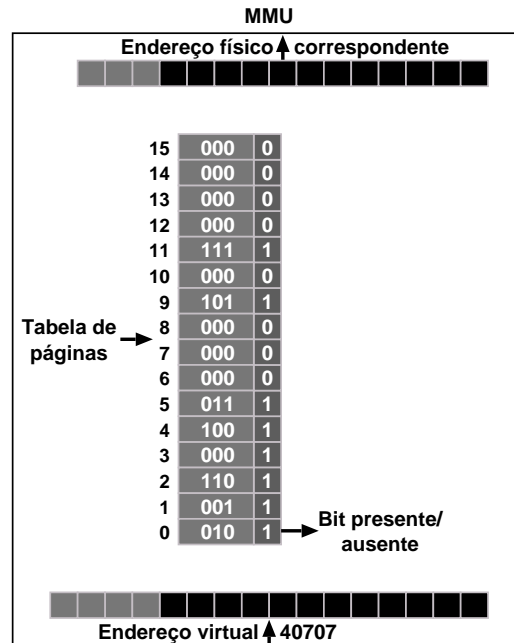
- (b) (0.75) Dois recursos forem não preemptivos, dois forem preemptivos e, para A e B, cada um solicitar todos os recursos do outro antes de liberar os recursos que possui?

**Resp.:** Como existem três recursos envolvidos (R, T e U), então pelo menos um deles deve ser preemptivo. Se dois recursos forem preemptivos, então não teremos um impasse, porque somente um dos processos precisa se bloquear até que o outro processo libere

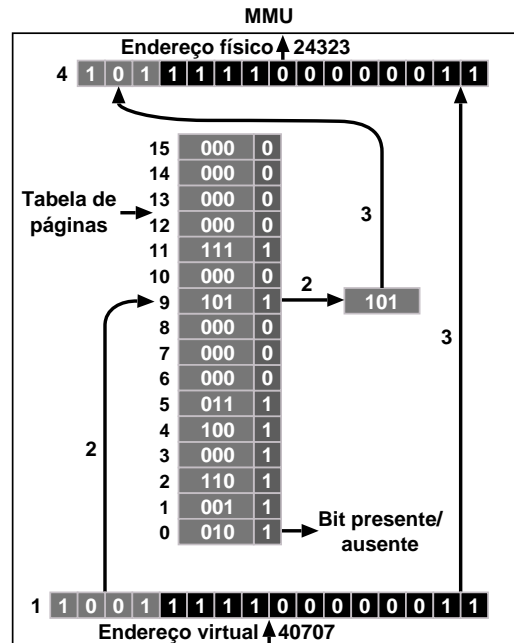
o único recurso não preemptivo. Agora, se o único recurso preemptivo for um dos recursos de A (R ou T), então teremos um impasse, pois este recurso poderia ser retirado de A e alocado a B, mas isso não poderia ser feito com o outro recurso de A. Agora, se U for o recurso preemptivo, então não teremos um impasse, pois o recurso poderia ser retirado de B e alocado a A e, após A liberar todos os recursos que possui, U poderia ser novamente alocado a B junto com R e T. Nas partes (a) e (b) da figura a seguir mostramos o que ocorre se R for o recurso preemptivo e B conseguir obtê-lo (parte (a)) ou não (parte (b)) antes de bloquear ao requisitar T. Nas partes (c) e (d) mostramos o que ocorre se T for o recurso preemptivo e B conseguir obtê-lo (parte (c)) ou não (parte (d)) antes de bloquear ao requisitar R. Finalmente, nas partes (e) e (f), mostramos o que ocorre se U for o recurso preemptivo e B bloquear ao requisitar R (parte (e)) ou T (parte (f)).



2. (1.0) A figura dada a seguir possui algumas das informações necessárias à conversão do endereço virtual 40707 em um endereço físico da memória. Complete esta figura, descrevendo os passos executados pela MMU ao converter este endereço virtual para o físico.



**Resp.:** A seguir mostramos a figura completa com o processamento executado pela MMU ao converter o endereço. No Passo 1, a MMU divide o endereço virtual 40707, cujo valor na base binária é 1001111100000011, em dois campos: os 4 bits superiores 1001 que representam a página virtual 9 e os 12 bits inferiores 111100000011 que representam o deslocamento 3843 dentro desta página. No Passo 2, o número da página virtual 9 é usado para acessar a entrada 9 desta página na tabela de páginas dada na figura. A partir desta entrada descobrimos que a página está na memória, pois o bit presente/ausente é 1 (isto é, não será gerada uma falha de página), e que ela está armazenada na moldura de página 5 (101 em binário). No Passo 3, o endereço físico correspondente ao endereço virtual é formado. Os seus 15 bits serão compostos pelos 3 bits superiores 101 (o valor em binário da moldura 5) e pelos 12 bits inferiores 111100000011 (o valor em binário do deslocamento 3843). Finalmente, no Passo 4, o endereço físico em binário 101111100000011, cujo valor na base decimal é 24323, é colocado no barramento, e o acesso a este endereço físico ocorre, garantindo a correta execução da instrução que acessou o endereço virtual 40707.



3. (1.5) Suponha que o sistema operacional alocou três molduras de página para um processo, inicialmente vazias. Suponha ainda que o processo acessou, nesta ordem, as páginas 1, 13, 5, 0, 7, 7, 5, 9, 0 e 1. Para cada um dos algoritmos dados a seguir, diga quantas falhas de página ocorreram, e qual página será substituída se o processo posteriormente acessar uma página diferente das que acessou:

- (a) (0.75) O algoritmo NRU, supondo que o processo altera o conteúdo das páginas 5 e 9, e que o bit referenciada de uma página é ligado quando ela é carregada na memória e desligado sempre que o processo acessar, em seqüência, duas páginas diferentes dela.

**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. Como vimos na Aula 9, o bit  $R$  sempre é ligado quando

a página é acessada. Na figura dada a seguir mostramos, em cada linha, como as páginas na memória são divididas em cada uma das 4 classes, após acessarmos a página dada na primeira coluna, de acordo com a ordem dada no enunciado. As classes são mostradas nas colunas de 2 até 5 e, na última coluna, é dito se o acesso à página causou uma falha de página. Também destacamos, em negrito, a(s) página(s) a ser(em) substituída(s). Como podemos ver por esta figura, ocorreram 8 falhas de página, e a próxima página a ser substituída seria a 9. Note que somente as páginas 5 e 9 podem estar nas classes 1 ou 3, pois somente para elas, que são alteradas, o bit  $M$  é 1.

Página	Classes				Ocorreu uma falha?
	0	1	2	3	
1	-	-	<b>1</b>	-	Sim
13	-	-	<b>1,13</b>	-	Sim
5	<b>1</b>	-	13	5	Sim
0	<b>13</b>	-	0	5	Sim
7	-	<b>5</b>	0,7	-	Sim
7	<b>0</b>	5	7	-	Não
5	<b>0</b>	-	7	5	Não
9	<b>7</b>	-	-	5,9	Sim
0	-	<b>5</b>	0	9	Sim
1	-	<b>9</b>	0,1	-	Sim

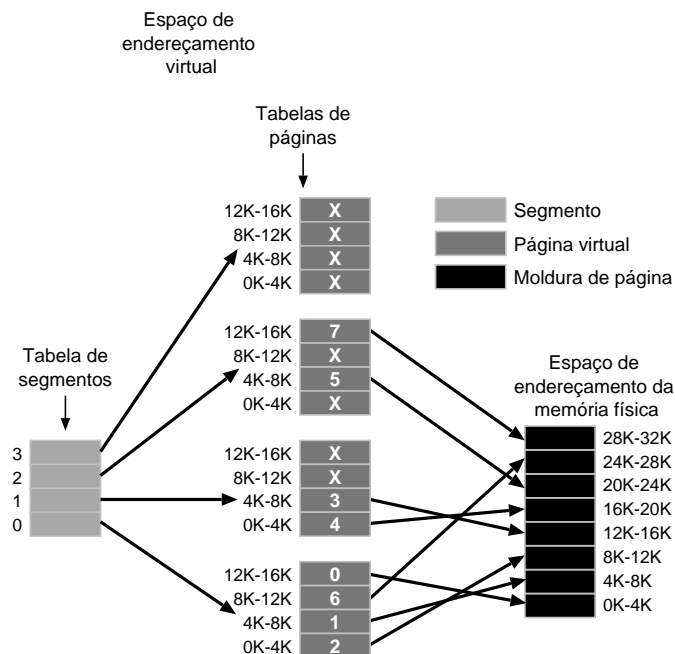
(b) (0.75) O algoritmo LRU.

**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 será a primeira página segundo esta ordenação, isto é, a página não acessada há mais tempo. Na figura dada a seguir mostramos, em cada linha, o que ocorre ao acessarmos as páginas na ordem dada no enunciado. Para cada uma destas linhas mostramos, na primeira coluna, a página que é acessada, na segunda coluna, a ordem em que as páginas devem ser escolhidas e finalmente, na terceira coluna, se o acesso gera uma falha de página. A primeira

página segundo a ordenação, que será a substituída, é mostrada em negrito. Como podemos ver por esta figura, teremos 8 falhas de página, e a próxima página a ser substituída será a 9.

Páginas	Ordenação	Ocorreu uma falha?
1	<b>1</b>	Sim
13	<b>1</b> 13	Sim
5	<b>1</b> 13 5	Sim
0	<b>13</b> 5 0	Sim
7	<b>5</b> 0 7	Sim
7	<b>5</b> 0 7	Não
5	<b>0</b> 7 5	Não
9	<b>7</b> 5 9	Sim
0	<b>5</b> 9 0	Sim
1	<b>9</b> 0 1	Sim

4. (2.25) Suponha que o sistema operacional usa a técnica de segmentação com paginação, e que o computador tem um espaço de endereçamento virtual dividido como na figura dada a seguir. Responda:



- (a) (0.75) Para cada segmento, quais faixas de endereços gerariam falhas de página ao acessá-las, se o “X” em uma entrada de uma tabela de páginas indicasse que a página não está na memória?

**Resp.:** Para o segmento 0, vemos que não ocorrerão falhas de páginas ao acessarmos um endereço deste segmento, pois todas as suas páginas estão mapeadas na memória. Para o segmento 1, vemos que uma falha de página será gerada se acessarmos os endereços de 8192 até 16383 (faixas 8K-12K e 12K-16K). Para o segmento 2, vemos que uma falha de página será gerada ao acessarmos os endereços de 0 até 4095 (faixa 0K-4K) ou os endereços de 8192 até 12287 (faixa 8K-12K). Finalmente, o acesso a qualquer endereço do segmento 3 irá gerar uma falha de página, pois nenhuma das páginas deste segmento está mapeada na memória.

- (b) (0.75) Suponha que o processo A está usando os segmentos 1 e 3, e que o processo B está usando os segmentos 0 e 2. Suponha ainda que a ordem de carga das páginas nas molduras foi a seguinte: 1, 7, 5, 2, 0, 6, 3 e 4. Qual será a página substituída quando o processo A acessar algum endereço do segmento 3, se o sistema operacional usar o algoritmo FIFO para substituir as páginas com uma política de alocação global? E se a política fosse local ao invés de global?

**Resp.:** Como vimos na Aula 9, no algoritmo FIFO, a página carregada há mais tempo na memória será a escolhida para ser substituída. E como podemos ver pela figura, as molduras 3 e 4 estão associadas ao processo A, e as outras molduras estão associadas ao processo B. Então:

- Se a política de alocação global for usada, todas as molduras deverão ser consideradas pelo algoritmo FIFO. Agora, como a ordem 1, 7, 5, 2, 0, 6, 3 e 4 está de acordo com a ordem crescente do tempo de carga das páginas nas molduras, então a página 1 do segmento 0 (associado ao processo B), que está na moldura 1, será a escolhida para ser substituída.



- Se a política de alocação for local, então somente as molduras associadas ao processo A, ou seja, as molduras 3 e 4, deverão ser consideradas. Como neste caso a ordenação, de acordo com o tempo de carga, será 3 e 4, então a página 1 do segmento 1, que está na moldura 3, será a escolhida para ser substituída.
- (c) (0.75) Suponha que uma TLB foi adicionada à MMU. Qual deveria ser o número de entradas desta TLB para 5% dos acessos a endereços de um segmento serem satisfeitos por ela?

**Resp.:** O enunciado desta questão ficou ambíguo, pois não foi dito se o endereço é o endereço virtual bidimensional, composto pelo par segmento, endereço linear dentro deste segmento, ou o endereço linear de um segmento. Logo, vamos considerar como corretas uma das duas respostas: a que considerar o endereço como o bidimensional ou a que considerar o endereço como o linear. A seguir damos as respostas para cada uma destas possibilidades:

- Para garantir que 5% dos acessos aos endereços lineares de cada um dos segmentos sejam satisfeitos pela TLB, então deveremos dividir o espaço da TLB em 4 partes iguais, e associar cada parte a um dos segmentos. Como cada segmento tem 16384 (16K) endereços lineares, então bastará que a TLB possua 820 entradas para cada segmento. Isso ocorrerá porque uma entrada da TLB dá o mapeamento de um dos endereços lineares do segmento, e 5% de 16384 é aproximadamente igual a 820 (pois  $16384 \times 0,05 = 819,2 \sim 820$ ). Logo, a TLB deverá ter 3280 entradas no total.
- Para garantir que 5% dos acessos aos endereços virtuais bidimensionais sejam satisfeitos pela TLB, então deveremos ter entradas suficientes para armazenar 5% destes endereços. Existem 65536 endereços virtuais, pois temos 4 segmentos com 16384 (16K) endereços lineares. Logo, a TLB deverá possuir 3277 entradas, pois uma entrada da TLB dá o mapeamento de um dos endereços virtuais, e 5% de 65536 é aproximadamente igual a 3277 (pois  $65536 \times 0,05 = 3276,8 \sim 3277$ ).

5. (1.5) Considere um computador com um disco de 16 blocos de 64KB, numerados de 0 até 15. Suponha que o arquivo A está armazenado, em ordem, nos blocos 3, 4, 7, 15 e 11 do disco, e o arquivo B está armazenado, em ordem, nos blocos 8, 9, 10, 13 e 14 do disco. Responda:
- (a) (0.75) Se desejarmos alocar um arquivo C no disco, qual tamanho ele poderá ter no máximo? Isso dependerá da técnica de alocação de blocos utilizada?

**Resp.:** O tamanho do arquivo dependerá da técnica de alocação usada, pois a técnica de alocação contígua, ao contrário das outras técnicas, exige que os blocos sejam consecutivos no disco. Como podemos ver pelo enunciado, temos 6 blocos livres de 64KB no disco: os blocos de 0 até 2 e os blocos 5, 6 e 12. Se a técnica de alocação contígua for usada, então o tamanho máximo do arquivo C será de 3 blocos, isto é,  $3 \times 64KB = 192KB$ , se ele for armazenado nos blocos de 0 até 2. Agora, se uma das outras técnicas de alocação for usada, então o arquivo C poderá usar todos os 6 blocos livres do disco e, com isso, o seu tamanho será de até  $6 \times 64KB = 384KB$ .

- (b) (0.75) Considere as técnicas de alocação de blocos contígua, por lista encadeada utilizando um índice e por nós-i. Para cada técnica que puder gerar a alocação de blocos dada no enunciado, descreva como esta alocação seria implementada de acordo com o funcionamento da técnica.

**Resp.:** -A técnica de alocação contígua não pode gerar a alocação dada no enunciado, pois os blocos, de ambos os arquivos A e B, não estão todos consecutivos no disco.

-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 16 entradas, referenciadas pelos endereços dos blocos, pois temos 16 blocos no disco. Nesta tabela, um “X” na entrada indica que o bloco associado a ela é o último bloco do arquivo.



vista na Aula 12 gastaria menos espaço no disco para armazenar a lista de blocos livres do disco se  $x$  blocos estiverem livres?

**Resp.:** Se usarmos a lista encadeada para armazenar os  $x$  blocos livres, precisaremos usar  $2x$  bytes do disco, pois existem 65536 blocos no disco, o que significa que cada endereço de um bloco precisará de 2 bytes para ser representado. Agora, se usarmos um mapa de bits, sempre precisaremos de 65536 bits, que ocuparão 8192 bytes do disco, pois cada bloco do disco deve sempre ser representado por 1 bit no mapa de bits. Logo, se  $2x < 8192$ , ou seja, se  $x < 4096$ , então a lista encadeada gastará menos espaço do disco. Agora, se  $x > 4096$ , o mapa de bits é que gastará menos espaço do disco. Finalmente, se  $x = 4096$ , tanto a lista encadeada como o mapa de bits gastarão 8192 bytes do disco.

- (b) (0.75) Quais seriam os significados das permissões de cada um dos objetos dados na figura a seguir?

Objeto	Lista com os domínios e as operações
Arquivo0	(hal, *, R-X)
Arquivo1	(vod, land, RWX)
Arquivo2	(hal, *, RW-), (lam, staff, RW-), (felipe, *, R-X)
Arquivo3	(*, cederj, --X)
Arquivo4	(rnp, *, ---), (*, cederj, R--)

**Resp.:** A seguir descrevemos o significado das permissões para cada um dos objetos dados na figura, lembrando que um “R” significa que o objeto pode ser lido, que um “W” significa que o objeto pode ser alterado, e que um “X” significa que o objeto pode ser executado:

- Objeto Arquivo0: A única entrada da lista significa que somente o usuário “hal”, pertencente a qualquer grupo (indicado por um “\*”) pode ler ou executar o objeto.
- Objeto Arquivo1: A única entrada da lista significa que somente o usuário “vod” do grupo “land” pode ler, alterar ou executar o objeto.
- Objeto Arquivo2: A primeira entrada da lista significa que o usuário “hal” de qualquer grupo (indicado por um “\*”) pode

ler ou alterar o objeto. A segunda entrada significa que o usuário “lam” do grupo “staff” pode também ler ou alterar o objeto. Finalmente, a terceira entrada significa que o usuário “felipe” de qualquer grupo (indicado por um “\*”), pode ler ou executar o objeto.

- Objeto Arquivo3: A única entrada significa que somente qualquer usuário (indicado por um “\*”) do grupo “cederj” pode executar o objeto.
- Objeto Arquivo4: Estas duas permissões são interessantes: a primeira entrada significa que o acesso ao objeto é negado (indicado por um “---”) ao usuário “rnp” de qualquer grupo (indicado por um “\*”). Logo, a segunda entrada significa que somente qualquer usuário (indicado por um “\*”) do grupo “cederj”, com exceção do usuário “rnp” (devido à primeira entrada), pode ler o objeto.

- (c) (0.75) Suponha que um bloco livre seja incorretamente alocado aos objetos Arquivo0 e Arquivo3 da figura do item anterior, após ambos aumentarem o seu tamanho em um bloco. Isso geraria uma inconsistência no sistema de arquivos? Como ela afetaria os dados destes arquivos?

**Resp.:** -Sim, isso geraria uma inconsistência, pois um mesmo bloco estaria associado a dois objetos (Arquivo0 e Arquivo1) e, quando o sistema de arquivos está consistente, cada um dos blocos do disco está livre ou associado a somente um dos objetos armazenados no sistema de arquivos.

-Os dados destes arquivos (note que os objetos são arquivos) seriam afetados como descrito a seguir. Se alterássemos os dados armazenados no bloco compartilhado em um dos arquivos, parte dos dados do outro arquivo, também armazenados neste bloco, seriam indiretamente alterados. Logo, a alteração dos dados de um dos arquivos pode alterar parte dos dados do outro arquivo, comprometendo o conteúdo deste arquivo.