



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 2013

**Atenção:** ADs enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.

**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.

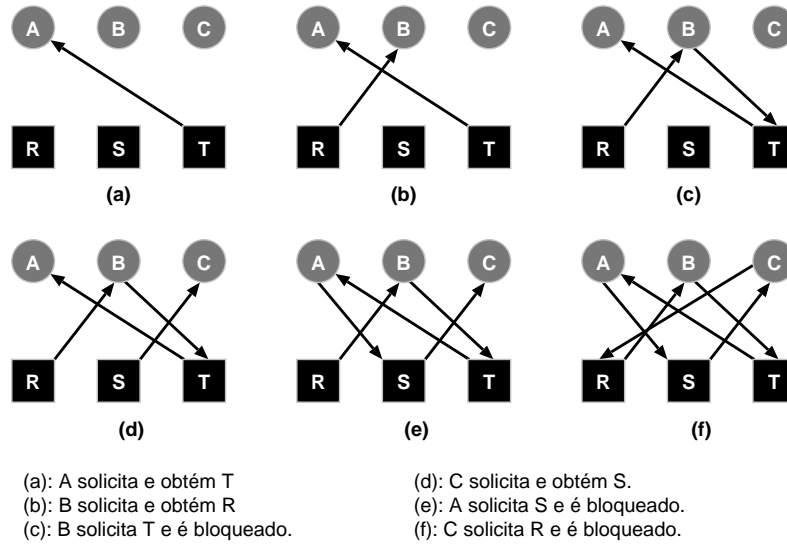
**Observação:** A questão 6 é opcional. Se você tentar resolvê-la, você poderá ganhar até 2,0 pontos adicionais na nota da prova, mas a sua nota ainda será limitada a 10,0 pontos.

Nome -

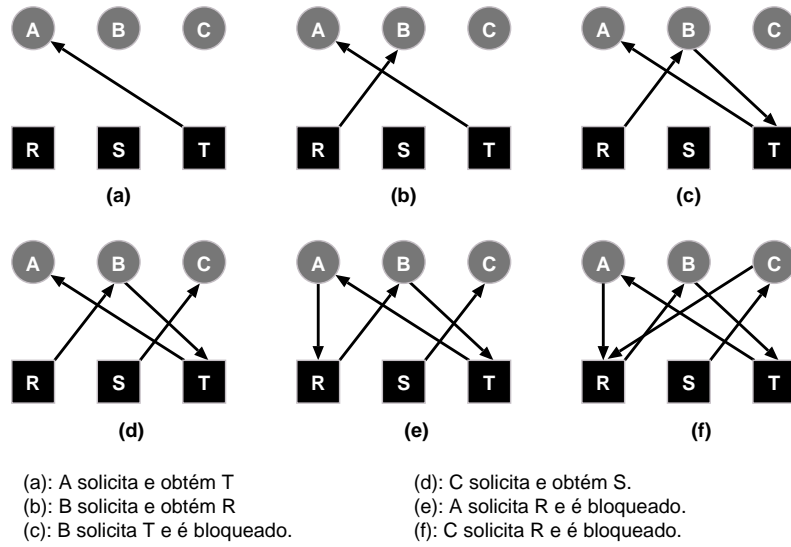
Assinatura -

- 
1. (2,0) Na figura dada a seguir mostramos um exemplo de um grafo de recursos com um impasse. Suponha que no passo (e) A tivesse solicitado

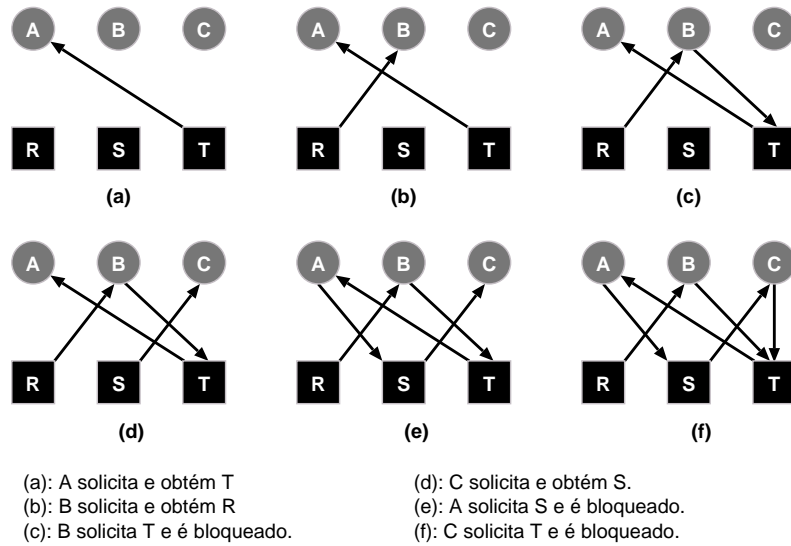
$R$  em vez de  $S$ . O impasse continuaria a existir? E se no passo (f)  $C$  tivesse solicitado  $T$  ao invés de  $R$ ? Justifique a sua resposta.



**Resp.:** Na figura dada no enunciado existe um impasse envolvendo todos os processos e todos os recursos, devido ao ciclo orientado  $A \rightarrow S \rightarrow C \rightarrow R \rightarrow B \rightarrow T \rightarrow A$  no grafo da parte (f) da figura. Como esse ciclo tem todos os recursos então todos eles são não-preemptivos. Agora, se o processo  $A$ , no passo (e), solicitar  $R$  em vez de  $S$ , obteremos a sequência de passos dadas na figura a seguir. Como podemos ver pela parte (f) da figura, novamente teremos um impasse, envolvendo os processos  $A$  e  $B$  e os recursos  $R$  e  $T$ , devido ao ciclo orientado  $A \rightarrow R \rightarrow B \rightarrow T \rightarrow A$ .

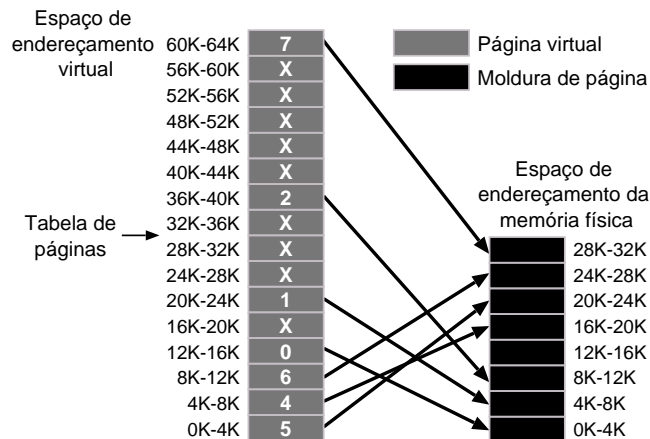


Finalmente, se o processo  $C$  solicitar  $T$  ao invés de  $R$ , também teremos um impasse como podemos ver pela parte (f) da figura a seguir, envolvendo os processos  $A$  e  $C$  e os recursos  $S$  e  $T$ , devido ao ciclo orientado  $A \rightarrow S \rightarrow C \rightarrow T \rightarrow A$ .



2. (2,0) Utilizando a tabela de páginas dada a seguir, vista no final da aula 8, forneça o endereço físico correspondente a cada um dos en-

dereços virtuais abaixo, supondo que todas as molduras são alocadas ao processo que acessou as páginas:



- (a) (0,6) 0.
- (b) (0,7) 65432.
- (c) (0,7) 18081.

**Resp.:** Pela figura vemos que o espaço de endereçamento virtual possui 64K, com endereços de 16 bits variando de 0 até 65535. Já o espaço de endereçamento físico possui 32K, com endereços de 15 bits variando de 0 até 32767. As páginas possuem 4K de tamanho e, com isso, temos 16 páginas virtuais e 8 molduras de página. Logo, um endereço virtual é dividido do seguinte modo: 4 bits para o número da página virtual e 12 bits para o deslocamento. A seguir mostramos como converter cada um dos endereços virtuais para os endereços físicos correspondentes. Mostramos, entre parênteses, o endereço na base binária, separando os seus dois campos pelo caractere “|”.

- (a) Endereço virtual 0 (0000 | 000000000000): neste caso, vemos que estamos acessando a palavra 0 da página virtual 0 (com endereços de 0 até 4095). Pela figura, vemos que a página virtual 0 está mapeada na moldura de página 5 (com endereços de 20480 até 24575). Logo, o endereço físico é 20480 (a primeira palavra da moldura) mais 0 (o deslocamento), ou seja, 20480.

- (b) Endereço virtual 65432 (1111 | 111110011000): para este endereço, vemos que estamos acessando a palavra 3992 da página virtual 15 (com endereços de 61440 até 65535). Agora, como a página 15 está mapeada na moldura de página 7 (com endereços de 28672 até 32767), o endereço físico é 28672 (a primeira palavra da moldura) mais 3992 (o deslocamento), isto é, 32664.
- (c) Endereço virtual 18081 (0100 | 011010100001): este endereço aponta para a palavra 1697 da página virtual 4 (com endereços de 16384 até 20479). A página virtual 4 não está mapeada na memória mas, depois de a falha de página ocorrer, ela poderá ser mapeada na moldura de página 3, livre, sem que seja necessária a execução do algoritmo de substituição de página. Depois disso, a página estará mapeada na moldura de página 3 (com endereços de 12288 até 16383), significando que o endereço físico é 12288 (a primeira palavra da moldura) mais 1697 (o deslocamento), isto é, 13985.
3. (2,0) Suponha que um computador possui um espaço de endereçamento virtual de 32 bits e um espaço de endereçamento físico de 24 bits. Suponha ainda que o tamanho de uma página virtual é de 32KB. Responda:
- (a) (0,7) Quantas molduras de página existem no espaço de endereçamento físico? Quantas páginas virtuais existem no espaço de endereçamento virtual?

**Resp.:** -O tamanho das molduras de página é o mesmo tamanho das páginas virtuais, isto é, 32KB. Logo, precisaremos de 15 bits no endereço físico para representar o deslocamento dentro da moldura, pois 32KB é igual a  $32768 = 2^{15}$  bytes. Agora, como um endereço físico tem 24 bits, os 9 bits restantes serão usados para representar o número da moldura de página e, com isso, teremos  $2^9 = 512$  molduras de página.

-O deslocamento dentro da página virtual também precisará de 15 bits para ser representado, pois o tamanho das páginas virtuais é 32KB. Com isso, o número da página virtual usará os 17 bits restantes do endereço virtual, o que fará com que o espaço de endereçamento virtual seja dividido em  $2^{17} = 131072$  páginas

virtuais.

- (b) (0,7) Suponha que desejamos dividir o campo número da página virtual do endereço virtual em partes iguais, para obtermos uma tabela de páginas multinível cujas tabelas possuem o mesmo número de entradas. Quais seriam os possíveis modos de dividirmos este campo, se não quisermos ter mais do que 4 níveis na tabela de páginas?

**Resp.:** Como o campo número de página virtual tem 17 bits, e 17 é um número primo, então não podemos dividir o campo em quatro ou menos partes iguais, pois uma das partes sempre seria maior do que as outras (por exemplo, se tentarmos dividir o campo em 4 partes, uma parte teria 5 bits e as três outras partes teriam 4 bits).

- (c) (0,6) Suponha que o hardware do computador foi aperfeiçoado, e que uma TLB com 32 entradas foi adicionada à MMU. Se as páginas virtuais possuírem igual probabilidade de serem acessadas, qual será a probabilidade de as informações de uma página acessada estarem armazenadas em uma das entradas da TLB?

**Resp.:** Como vimos na resposta do item (a), temos 131072 páginas virtuais. Agora, como todas essas páginas possuem a mesma probabilidade de serem acessadas, então a probabilidade de as informações de uma página estarem na TLB será de  $\frac{32}{131072} \approx 0,00024$ , isto é, será de aproximadamente 0,024%. Estamos supondo aqui que, ao ser necessária uma substituição na TLB, a página a ser retirada é escolhida aleatoriamente.

4. (2,0) Suponha que o sistema operacional possui 6 molduras de página disponíveis para serem alocadas a dois processos A e B. Suponha ainda que o sistema use o algoritmo LRU para alocar as molduras aos processos quando uma falha de páginas ocorrer. Se os processos acessaram, na ordem, as páginas virtuais 0, 1, 2, 3, 2, 2, 1, 3, 4, 5, 6 e 0, qual será a página a ser substituída no próximo acesso se uma falha de página ocorrer, supondo que a política de alocação global é usada? E se for

usada uma política de alocação local igualitária, supondo que A acessou as páginas 0, 2 e 4 e que B acessou a página que gerou a falha? Justifique a sua resposta.

**Resp.:** -Primeiramente vamos mostrar, na tabela a seguir, a sequência de acessos às páginas virtuais dadas 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. 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, a página 1 será substituída caso uma nova falha de página ocorra.

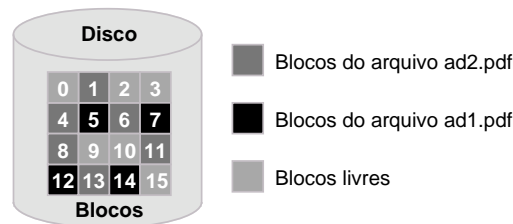
Páginas	Ordenação
0	0
1	0 1
2	0 1 2
3	0 1 2 3
2	0 1 3 2
2	0 1 3 2
1	0 3 2 1
3	0 2 1 3
4	0 2 1 3 4
5	0 2 1 3 4 5
6	2 1 3 4 5 6
0	1 3 4 5 6 0

-Agora, se usarmos a política de alocação local igualitária, 3 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 foi o processo B que gerou

a falha, somente precisaremos construir a tabela para esse processo. Devido ao processo A ter acessado as páginas 0, 2 e 4, então o processo B acessou, em ordem, as páginas 1, 3, 1, 3, 5 e 6. Logo, como podemos ver pela tabela a seguir, similar à tabela descrita anteriormente, a página a ser substituída caso uma falha de página ocorra será a 3.

Páginas	Ordenação		
1	1		
3	1	3	
1	3	1	
3	1	3	
5	1	3	5
6	3	5	6

5. (2,0) Considere um computador com um disco de 16 blocos de 64KB, numerados de 0 até 15, alocado do modo dado na figura a seguir. Responda:



- (a) (1,0) Se desejarmos alocar um arquivo ad3.pdf 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: o bloco 0, os blocos 2 e 3, os blocos 9 e 10 e o bloco 15. Se a técnica de alocação contígua for usada, então o tamanho máximo do arquivo ad3.pdf será de 2 blocos, isto é,  $2 \times 64\text{KB} = 128\text{KB}$ , se



ele for armazenado nos blocos 2 e 3 ou nos blocos 9 e 10. Agora, se uma das outras técnicas de alocação for usada, então o arquivo ad3.pdf poderá usar todos os 6 blocos livres do disco e, com isso, o seu tamanho será de até  $6 \times 64\text{KB} = 384\text{KB}$ .

- (b) (1,0) 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 ad1.pdf e ad2.pdf, não estão todos consecutivos no disco. Nas outras técnicas, será preciso supor uma ordem para os blocos dos arquivos ad1.pdf e ad2.pdf. Qualquer resposta será considerada correta se estiver de acordo com a ordem dos blocos escolhida.

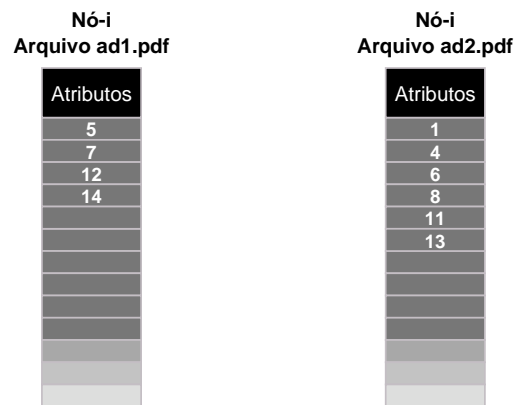
-Se a técnica de alocação por lista encadeada utilizando um índice for usada, vamos obter a tabela dada na figura a seguir. Note que ela tem 16 entradas, referenciadas pelos endereços dos blocos, pois temos 16 blocos no disco. Nesta tabela, um “X” na entrada indica que o bloco associado a ela é o último bloco do arquivo.

Ordem dos blocos:  
Arquivo ad1.pdf: 5, 7, 12 e 14.  
Arquivo ad2.pdf: 1, 4, 6, 8, 11 e 13.

	← 4 bits →	
0	L	
1	4	← Início do arquivo Arquivo ad2.pdf
2	L	
3	L	
4	6	
5	7	← Início do arquivo Arquivo ad1.pdf
6	8	
7	12	
8	11	
9	L	← L indica os blocos livres do disco
10	L	
11	13	
12	14	
13	X	← Fim do arquivo Arquivo ad2.pdf
14	X	← Fim do arquivo Arquivo ad1.pdf
15	L	

-Se a técnica de alocação por nós-i for usada, os nós-i dados na

figura a seguir estarão associados aos arquivos ad1.pdf e ad2.pdf. Nesta figura, as entradas não usadas de um nó-i não foram preenchidas. Como cada arquivo tem menos do que 11 blocos, não foi necessário o uso dos três últimos campos de cada nó-i, usados para indicar os endereços dos blocos indiretos simples, duplo e triplo, pois podemos armazenar todos os blocos de cada arquivo no seu nó-i.



6. (2,0) Suponha que o computador tem um disco com 524288 blocos de 64KB. Suponha ainda que o sistema operacional usa o mecanismo de proteção por lista de controle de acesso ao gerenciar o acesso aos objetos do sistema pelos processos. Responda:
- (a) (0,5) Em relação ao gerenciamento dos blocos livres, qual técnica vista na aula 12 gastaria menos espaço no disco para armazenar a lista de blocos livres do disco se  $x$  blocos estiverem sendo usados por arquivos?

**Resp.:** Se usarmos a lista encadeada para armazenar os  $x$  blocos livres, precisaremos usar  $3x$  bytes do disco, pois como existem  $524288 = 2^{19}$  blocos no disco, cada endereço de um bloco precisará de 3 bytes (24 bits) para ser representado devido a não podermos usar parcialmente um byte. Agora, se usarmos um mapa de bits, sempre precisaremos de 524288 bits, que ocuparão 65536 bytes do disco, pois cada bloco do disco deve sempre ser representado por 1 bit no mapa de bits. Logo, se  $3x < 65536$ , ou seja, se

$x < 21845, 33 \dots$ , então a lista encadeada gastará menos espaço do disco. Ou seja, isso ocorrerá se  $x$  for no máximo 21845. Agora, se  $x > 21845$ , o mapa de bits é que gastará menos espaço do disco.

- (b) (0,7) 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	(so, *, --X)
Arquivo1	(cederj, curso, RW-)
Arquivo2	(so, *, R-X), (professor, staff, RWX), (prova, *, ---)
Arquivo3	(*, curso, -W-)

**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 “so”, pertencente a qualquer grupo (indicado por um “\*”) pode acessar o objeto, e somente para execução.
- Objeto Arquivo1: A única entrada da lista significa que somente o usuário “cederj” do grupo “curso” pode acessar o objeto, podendo tanto lê-lo quanto alterá-lo.
- Objeto Arquivo2: A primeira entrada da lista significa que o usuário “so” de qualquer grupo (indicado por um “\*”) pode ler ou executar o objeto. A segunda entrada significa que o usuário “professor” do grupo “staff” pode ler, alterar ou executar o objeto. Finalmente, a terceira entrada significa que o usuário “prova” de qualquer grupo (indicado por um “\*”), não pode executar nenhuma operação sobre o objeto. Nenhum outro usuário pode acessar o objeto.
- Objeto Arquivo3: A única entrada significa que qualquer usuário (indicado por um “\*”) do grupo “curso” pode acessar o objeto, e somente para alterá-lo.

- (c) (0,8) Suponha que dois blocos livres sejam incorretamente alocados aos objetos Arquivo0 e Arquivo3 da figura do item anterior, após ambos aumentarem o seu tamanho em dois blocos. Isso geraria uma inconsistência no sistema de arquivos? E se os arquivos

aumentarem os seus tamanhos em um bloco?

**Resp.:** -Se o tamanho de ambos os arquivos aumentar em dois blocos, isso gerará uma inconsistência, pois ambos os blocos estarão necessariamente associados aos dois objetos 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.

-Já se cada arquivo aumentar o seu tamanho em um bloco então poderemos ou não ter uma inconsistência. Se cada arquivo receber um bloco diferente, então nenhuma inconsistência ocorrerá. Já se o mesmo bloco for alocado a ambos os arquivos, além do erro de um bloco ser alocado a mais de um arquivo, teremos também o erro de o outro bloco não alocado a nenhum arquivo ser ausente, pois não estará alocado a nenhum arquivo e não estará mais marcado como livre. Finalmente, se ambos os blocos forem alocados a ambos os arquivos teremos, além da inconsistência descrita no item anterior, um erro no tamanho dos arquivos, pois cada tamanho deveria ter aumentando em dois blocos.