



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

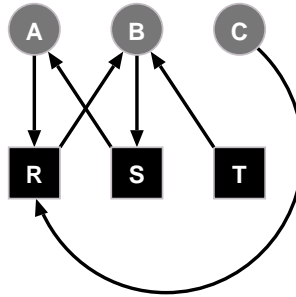
Quarto Período
AD2 - Primeiro Semestre de 2008

Nome -
Assinatura -

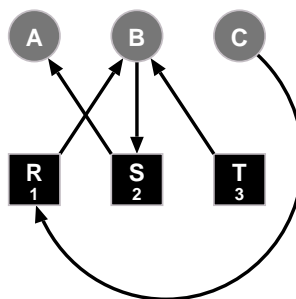
1. (1.0) Suponha que três processos A, B e C compartilham três recursos R, S e T. Considere a seqüência de alocação de recursos dada a seguir. Sempre teremos um impasse, ou isso dependerá dos recursos serem ou não preemptíveis? Se for possível a ocorrência de impasses, eles poderiam ser evitados numerando os recursos? Justifique a sua resposta.
- (a) B requisita e obtém T e R;
 - (b) A requisita e obtém S;
 - (c) C requisita R e é bloqueado;
 - (d) A requisita R e é bloqueado;
 - (e) B requisita S e é bloqueado.

Resp.: -Como podemos ver pela figura dada a seguir, que mostra o grafo de recursos obtido após a seqüência de alocações dada na questão, temos um ciclo neste grafo, envolvendo os processos A e B e os recursos

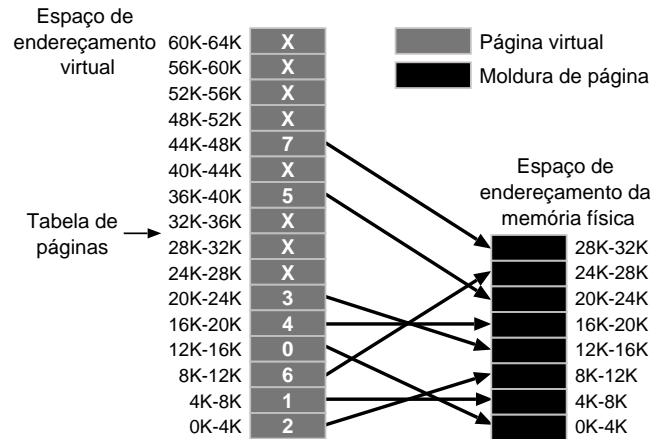
R e S. Para que este ciclo gere um impasse, os recursos R e S devem ambos ser não preemptíveis pois, em caso contrário, poderíamos retirar o recurso preemptível do processo que o possui (por exemplo, o recurso R do processo B), para depois alocá-lo ao processo A, desfazendo assim o ciclo do grafo de recursos.



-Supondo que R e S são não preemptíveis, poderemos evitar os impasses se, como vimos na Aula 7, numerarmos os recursos R, S e T em uma dada ordem como, por exemplo, 1 para o R, 2 para o S e 3 para o T. Depois de numerá-los, um processo somente é bloqueado ao requisitar um recurso se todos os recursos que possuir tiverem número menor do que o recurso que ele deseja (em caso contrário, um erro será gerado). Com isso, na nossa numeração proposta, o processo A não poderia ser bloqueado ao requisitar R, pois A possui S e o número 2 atribuído a S é maior do que o número 1 atribuído a R. Com isso, o grafo de recursos seria como o mostrado a seguir e, como não existem mais ciclos no grafo, não temos mais um impasse.



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 seguintes endereços virtuais:



- (a) (0.5) 19740.
 (b) (0.5) 6456.
 (c) (0.5) 47104.

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 19740 (0100 | 110100011100): para este endereço, vemos que estamos acessando a palavra 3356 da página virtual 4 (com endereços de 16384 até 20479). Agora, como a página 4 está mapeada na moldura de página 4 (com endereços também de 16384 até 20479), o endereço físico também é 19740.

- (b) Endereço virtual 6456 (0001 | 100100111000): neste caso, vemos que estamos acessando a palavra 2360 da página virtual 1 (com endereços de 4096 até 8191). Pela figura, vemos que a página virtual 1 está mapeada na moldura de página 1 (com endereços também de 4096 até 8191). Logo, o endereço físico também é 6456.
 - (c) Endereço virtual 47104 (1011 | 100000000000): este endereço aponta para a palavra 2048 da página virtual 11 (com endereços de 45056 até 49151). Agora, como esta página é mapeada na moldura de página 7 (com endereços de 28672 até 32767), então o endereço físico é 28672 (a primeira palavra da moldura) mais 2048 (o deslocamento), isto é, 30720.
3. (1.5) Suponha que um computador possui um espaço de endereçamento virtual de 24 bits e um espaço de endereçamento físico de 16 bits. Suponha ainda que o tamanho de uma página virtual é de 4KB. Responda:
- (a) (0.5) Quantas molduras de página existem no espaço de endereçamento físico? E 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, 4KB. Logo, precisaremos de 12 bits no endereço físico para representar o deslocamento dentro da moldura, pois 4KB é igual a $4096 = 2^{12}$ bytes. Agora, como um endereço físico tem 16 bits, os 4 bits restantes serão usados para representar o número da moldura de página e, com isso, teremos $2^4 = 16$ molduras de página.

-O deslocamento dentro da página virtual precisará de 12 bits para ser representado, pois o tamanho das páginas virtuais é de 4KB. Com isso, o número da página virtual usará os 12 bits restantes do endereço virtual, o que fará com que o espaço de endereçamento virtual seja dividido em $2^{12} = 4096$ páginas virtuais.

- (b) (0.5) 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 6 níveis na tabela de páginas?

Resp.: Como vimos na resposta do item (a), usamos 12 bits do endereço virtual para representar o número da página virtual. Agora, como desejamos obter tabelas multinível com o mesmo número de entradas, este campo deve ser dividido em partes iguais. Logo, como desejamos uma tabela com até 6 níveis, podemos dividir esse campo em 2, 3, 4 e 6 partes iguais, pois 12 é divisível por 2, 3, 4 e 6.

- (c) (0.5) Suponha que o hardware do computador foi aperfeiçoado, e que uma TLB com 256 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 4096 páginas virtuais. Agora, como todas estas páginas possuem a mesma probabilidade de serem acessadas, então a probabilidade de as informações de uma página estarem na TLB será de $256/4096 = 0,0625$, isto é, será de 6,25%.

4. (1.2) Suponha que um computador possui 64KB de memória. Suponha ainda que temos três processos A, B e C cujos códigos possuem tamanhos de, respectivamente, 1KB, 3KB e 48KB. Se o hardware do computador permitir páginas com tamanhos de 1KB, 2KB, 4KB e 8KB, então:

- (a) (0.6) Para quais tamanhos de página poderíamos armazenar todos os códigos dos processos na memória?

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 das páginas. Na segunda coluna mostramos o número total de páginas para o tamanho da primeira coluna. Nas colunas de três até cinco mostramos

a quantidade de páginas usadas pelos processos. Finalmente, na última coluna, mostramos o número total de páginas necessárias para armazenar os processos A, B e C na memória (igual à soma das quantidades de páginas nas colunas de três até cinco). Como vemos pela tabela, podemos armazenar todos os códigos dos processos para todos os tamanhos de página do enunciado.

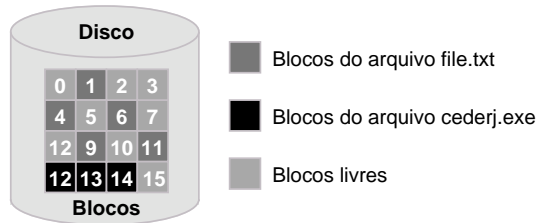
Tamanho da página	Número de páginas	Processos			Total de páginas
		A	B	C	
1KB	64	1	3	48	52
2KB	32	1	2	24	27
4KB	16	1	1	12	14
8KB	8	1	1	6	8

- (b) (0.6) Quais seriam os desperdícios devido à fragmentação interna para cada um dos possíveis tamanhos de página?

Resp.: Na tabela a seguir mostramos, para cada um dos tamanhos de página dados no enunciado da questão, os desperdícios para cada um dos processos. Na primeira coluna da tabela mostramos o tamanho da página. Nas colunas de dois até quatro mostramos o desperdício de espaço na última página para cada um dos processos. Finalmente, na última coluna, mostramos o desperdício total de espaço, isto é, os desperdícios devido à fragmentação interna. Como podemos ver pela tabela, somente não teremos fragmentação interna para o tamanho de página de 1KB.

Tamanho da página	Processos			Total de desperdício
	A	B	C	
1KB	0KB	0KB	0KB	0KB
2KB	1KB	1KB	0KB	2KB
4KB	3KB	1KB	0KB	4KB
8KB	7KB	5KB	0KB	12KB

5. (2.4) Suponha que o computador possui um disco com 16 blocos de 64KB, e que dois arquivos, file.txt e cederj.exe, estão armazenados no disco, como mostrado na figura a seguir. Responda:

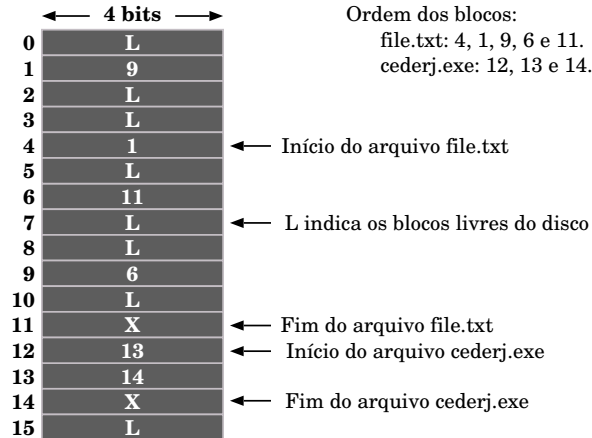


- (a) (0.8) Esta configuração pode ter resultado do uso da alocação contígua? Justifique a sua resposta.

Resp.: Não, pois o arquivo file.txt não pode ter sido armazenado através da alocação contígua, pois os seus blocos não são consecutivos no disco. Note que não importa que o arquivo cederj.exe possa ter sido armazenado usando esta técnica (se supusermos que os seus blocos lógicos são, em ordem, 12, 13 e 14), pois todos os arquivos do disco deveriam ter sido armazenados usando esta técnica.

- (b) (0.8) Suponha que o sistema operacional usa a alocação por lista encadeada utilizando um índice. Mostre a tabela com os blocos físicos para este disco, destacando o início e o fim dos arquivos file.txt e cederj.exe, e as entradas com os blocos livres do disco.

Resp.: Existem diversos modos de preenchermos a tabela, pois não especificamos na questão a ordem em que os blocos lógicos dos arquivos estão distribuídos nos blocos físicos. Vamos considerar como correta a resposta que mostre a tabela correta ao supormos uma ordem em particular para os blocos lógicos dos arquivos. Isso pode ser facilmente obtido da tabela, pois pedimos que fossem destacados os blocos inicial e final do arquivo. A seguir mostramos uma das possibilidades, supondo que os blocos lógicos do arquivo file.txt são armazenados, em ordem, nos blocos 4, 1, 9, 6 e 11 do disco, e os blocos lógicos do arquivo cederj.exe são armazenados, em ordem, nos blocos 12, 13 e 14 do disco.



- (c) (0.8) Suponha que os arquivos file.txt e cederj.exe são os únicos arquivos armazenados no diretório que os contém. Quais seriam as entradas deste diretório se o sistema operacional do computador fosse o MS-DOS? E se o sistema operacional fosse o UNIX? Ao responder, preencha somente, para cada entrada, os campos cujas informações podem ser obtidas pelo enunciado da questão.

Resp.: -Também teremos mais de uma resposta para esta parte da questão. O número de bytes usados de um bloco alocado a um arquivo varia de 1 até 65536 (64KB), pois o bloco deve ser usado. Logo, o tamanho do arquivo file.txt pode variar de 262145 (4 blocos com 64KB mais um byte no quinto bloco) até 327680 (todos os 5 blocos com 64KB). Já o tamanho do arquivo cederj.exe pode variar de 131073 (2 blocos com 64KB mais um byte no terceiro bloco) até 196608 (todos os 3 blocos com 64KB). Além disso, podemos escolher quaisquer um dos blocos dos arquivos como os seus blocos iniciais. Logo, podemos escolher um dentre os blocos 1, 4, 6, 9 e 11 para o arquivo file.txt, e um dentre os blocos 12, 13 e 14 para o arquivo cederj.exe. Portanto, todas as respostas que escolherem, para os campos S (tamanho do arquivo) e B (bloco inicial do arquivo), um dos valores dados anteriormente estarão corretas. A seguir mostramos uma das possíveis respostas. Escolhemos os maiores tamanhos, isto é, 327680 para o arquivo file.txt e 196608 para o arquivo cederj.exe. Os blocos iniciais que escolhemos foram o 4 para o arquivo file.txt, e o 12 para o arquivo cederj.exe.

	8	3	1	10	2	2	2	4	
file	txt						4	327680	
cederj	exe						12	196608	

- Ao contrário da parte anterior da questão, aqui temos somente uma possibilidade para as entradas do diretório dos arquivos file.txt e cederj.txt. Elas são mostradas na figura a seguir.

2	14
	file.txt
	cederj.exe

6. (2.4) Suponha que o sistema operacional usa um domínio de proteção com os objetos e os domínios definidos pela matriz mostrada a seguir, similar à matriz que foi vista na Aula 12. Responda:

Objetos

A: Arquivo1 B: Arquivo2
 C: Arquivo3 D: Arquivo4
 E: Arquivo5 F: Arquivo6
 G: Impressora1 H: Plotter2
 I: Domínio1 J: Domínio2
 K: Domínio3

Objeto

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

- (a) (0.8) Que objetos o domínio 2 pode acessar? Para cada um destes objetos, que operações podem ser executadas?

Resp.: -O domínio 2 poderá acessar cada objeto que não seja um domínio (pois a operação **Enter** é usada para a troca de domínios) cuja entrada na coluna correspondente a este objeto na linha do domínio 2 possua alguma das operações **r**, **w** ou **x**. Logo, poderemos acessar os seguintes objetos: A (Arquivo1), C (Arquivo3), E (Arquivo5), G (Impressora1) e H (Plotter2).

- 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 ler dados do objeto A, ou podemos executá-lo;
- Podemos somente ler dados do objeto C;
- Podemos ler dados do objeto E ou salvar dados neste objeto;
- Podemos somente executar o objeto G. Note que como esta operação é inadequada para uma impressora, possivelmente existe um erro de inconsistência no sistema.
- Podemos somente salvar dados no objeto H (“salvar dados” neste caso significa imprimir no plotter);

- (b) (0.8) Estando no domínio 3, um processo pode passar, usando a operação **Enter**, para quais domínios?

Resp.: Para descobrirmos para quais domínios o processo pode passar, deveremos olhar as colunas da linha do domínio 3 associadas aos objetos que são domínios, e que possuam a operação **Enter**. Pela matriz, vemos que podemos passar do domínio 3 para os domínios 1 e 2.

- (c) (0.8) Qual é o significado de cada uma das entradas vazias da matriz?

Resp.: O significado depende do objeto associado à entrada. Se este objeto for um domínio, isso significará que um processo que está associado ao domínio da linha com a entrada não poderá passar para o domínio da coluna com a entrada. E, em caso contrário, um processo no domínio associado à linha da entrada não terá direito a nenhum tipo de acesso (**r**, **w** ou **x**) ao objeto.