



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 2015

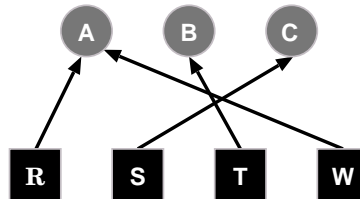
Atenção: Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um, $1/3$ dos pontos daquela questão.

Nome -
Assinatura -

-
1. (1,5) Suponha que existam quatro recursos no sistema operacional, R, S, T e W, e que três processos, A, B e C, já tenham começado as suas execuções. Responda, supondo que A tenha obtido R e W, B tenha obtido T e C tenha obtido S:
 - (a) (0,5) Se todos os recursos forem não-preemptivos, poderão ocorrer impasses se um dos processos tentar obter um novo recurso? Exemplifique um desses impasses, caso possam ocorrer.

Resp.: As alocações de recursos dadas no enunciado geram o grafo de recursos dado a seguir. Pelo grafo, vemos que pelo menos

dois processos precisam tentar obter dois recursos diferentes para que um impasse possa ocorrer (por exemplo, se A tentar obter T e B tentar obter R). Logo, não teremos impasses se somente um processo tentar obter um recurso.



- (b) (1,0) Quantos recursos deverão ser preemptivos para garantir que não teremos impasses, caso algum possa ocorrer no item anterior?

Resp.: Não é necessário que um dos recursos seja preemptivo porque, como vimos, um impasse somente ocorrerá se dois processos tentarem obter dois recursos diferentes.

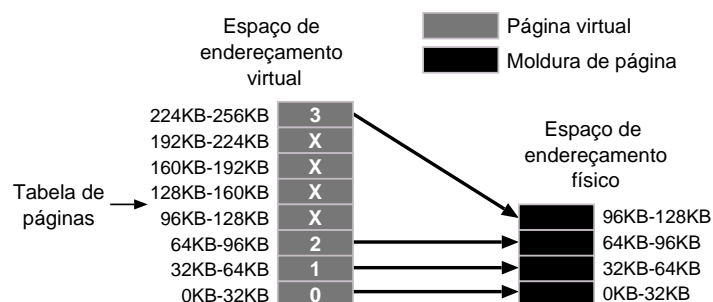
2. (1,5) Suponha que um computador tenha uma memória virtual de 256KB e uma memória física de 128KB. Suponha ainda que as páginas virtuais tenham 32KB de tamanho e que, inicialmente, nenhuma página virtual esteja mapeada em uma moldura de página. Suponha ainda que cada página seja mapeada na moldura de número mais próximo ao número da página, considerando apenas as molduras livres, caso existam, ou todas, caso contrário, neste último caso causando uma substituição. Responda:

- (a) (1,0) Como será o mapeamento das páginas nas molduras após um processo acessar, em ordem, os endereços virtuais 123456, 1045, 67441, 34565, 234432 e 0? Para responder, use uma figura similar à dada na transparência 19 da aula 8.

Resp.: Pelo enunciado da questão, vemos que o espaço de endereçamento virtual possui 256KB, com endereços de 18 bits variando de 0 até 262143. Já o espaço de endereçamento físico possui 128KB, com endereços de 17 bits variando de 0 até 131071. As

páginas virtuais possuem 32KB de tamanho e, com isso, temos 8 páginas virtuais e 4 molduras de página, pois o tamanho de cada moldura é igual ao da página. Logo, um endereço virtual é dividido em 3 bits para o número da página virtual e 15 bits para o deslocamento, e um endereço físico é dividido em 2 bits para o número da moldura de página e 15 bits para o deslocamento. A tabela a seguir mostra, para cada endereço virtual, supondo a ordem de acesso dada no enunciado, como ele é na base binária e como ele é dividido, a página virtual associada a ele, as molduras de página livres quando ele foi acessado e, finalmente, caso o acesso gere uma falha de página, a moldura escolhida segundo o critério dado no enunciado. Baseado nessa tabela, obtemos a figura dada após a tabela, que mostra todos os mapeamentos das páginas virtuais nas molduras de página após todos os acessos dados no enunciado.

Endereço virtual	Endereço em binário (página deslocamento)	Página virtual	Molduras livres	Moldura escolhida
123456	011 110001001000000	3	0-3	3
1045	000 000010000010101	0	0-2	0
67441	010 000011101110001	2	1-2	2
34565	001 000011100000101	1	1	1
234432	111 001001111000000	7		3
0	000 000000000000000	0		



- (b) (0,5) Que faixas de endereços virtuais não estarão mapeadas após o processo acessar, em ordem, as páginas dadas no item anterior?

Resp.: Pela figura, vemos que somente uma faixa de endereços virtuais, de 96KB–224KB, ou seja, os endereços de 98304 até 229375, não terá as suas páginas virtuais mapeadas em molduras de página.

3. (2,0) Suponha que um processo tenha acessado, em ordem, as páginas virtuais 1, 6, 3, 3, 2, 0, 5, 0, 2 e 7, e que tenham sido alocadas 3 molduras, inicialmente vazias, para o processo. Suponha ainda que o bit **referenciada** de uma página seja ligado quando ela é acessada e desligado após uma página diferente dela ter sido acessada. Se o algoritmo de segunda chance for usado, qual será a página acessada que gerará mais falhas? E se o algoritmo de idade for usado, supondo que os contadores tenham 5 bits? Justifique a sua resposta.

Resp.: -Primeiramente, vamos mostrar como as páginas são substituídas de acordo com o algoritmo de segunda chance, usando o critério de atualização do bit **referenciada** dado no enunciado da questão. Como vimos na aula 9, nesse algoritmo, as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo em que elas foram copiadas para a memória. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página copiada há mais tempo, se o seu bit **referenciada** for 0. Se o bit **referenciada** da página copiada há mais tempo na memória for 1, esse bit se tornará 0, a página deixará de ser a escolhida e será colocada no final da ordenação, e a próxima página da ordenação será considerada. 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. Já na segunda coluna mostramos a ordem em que as páginas devem ser escolhidas, sendo que ao lado da página, em parênteses, mostramos o valor atual do bit **referenciada**. Finalmente, na última coluna, mostramos se a página gerou uma falha de página ao ser acessada. Pela tabela, vemos que todas as páginas acessadas geraram somente 1 falha de página.

Página	Ordenação	Falha?
1	1(1)	Sim
6	1(0) 6(1)	Sim
3	1(0) 6(0) 3(1)	Sim
3	1(0) 6(0) 3(1)	Não
2	6(0) 3(0) 2(1)	Sim
0	3(0) 2(0) 0(1)	Sim
5	2(0) 0(0) 5(1)	Sim
0	2(0) 0(1) 5(0)	Não
2	2(1) 0(0) 5(0)	Não
7	0(0) 5(0) 7(1)	Sim

-Agora, como vimos na aula 9, o algoritmo de idade define, para cada página que está na memória, um contador, no nosso caso com 5 bits, que é inicializado em 0 quando a página é carregada na memória. A cada interrupção do temporizador, para cada página que está 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 não foi definida na questão após quantos acessos às páginas a interrupção do temporizador é gerada, serão aceitas as respostas que usem qualquer intervalo para a interrupção, desde que estejam corretas. Vamos supor aqui que a interrupção é gerada a cada acesso a uma página. Na figura dada 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 (d) 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, pois ainda existiam molduras livres. Também dizemos na figura, em cada parte com exceção da (a), se o acesso à página gerou ou não uma falha. Pela figura, vemos que novamente todas as páginas acessadas geraram somente 1 falha de página.

Páginas		Páginas		Páginas	
- -		1 (1) 10000		1 (0) 01000	
- -		- -		6 (1) 10000	
- -		- -		- -	
Falha: -		Falha: Sim		Falha: Sim	
(a)		(b)		(c)	
Páginas		Páginas		Páginas	
1 (0) 00100	⇐	1 (0) 00010	⇐	2 (1) 10000	
6 (0) 01000		6 (0) 00100		6 (0) 00010	⇐
3 (1) 10000		3 (1) 11000		3 (0) 01100	
Falha: Sim		Falha: Não		Falha: Sim	
(d)		(e)		(f)	
Páginas		Páginas		Páginas	
2 (0) 01000		2 (0) 00100	⇐	2 (0) 00010	⇐
0 (1) 10000		0 (0) 01000		0 (1) 10100	
3 (0) 00110	⇐	5 (1) 10000		5 (0) 01000	
Falha: Sim		Falha: Sim		Falha: Não	
(g)		(h)		(i)	
Páginas		Páginas			
2 (1) 10001		2 (0) 01000			
0 (0) 01010		0 (0) 00101	⇐		
5 (0) 00100	⇐	7 (1) 10000			
Falha: Não		Falha: Sim			
(j)		(k)			

4. (1,5) Suponha que as páginas 1, 2, 4, 2, 1, 0, 3 e 0 tenham sido acessadas na ordem dada. Suponha ainda que o algoritmo de substituição usado tenha sido o LRU, e que 4 molduras, inicialmente vazias, estejam disponíveis no sistema operacional. Se o sistema usar a política de alocação global, que página será substituída se a página 1 for novamente acessada? E se agora a política de alocação for local, supondo que as molduras sejam divididas entre dois processos, A e B, usando a política de alocação igualitária, e supondo que A tenha acessado as

páginas 0 e 2? 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, nenhuma página será substituída porque a página 1 ainda está na memória.

Páginas	Ordenação
1	1
2	1 2
4	1 2 4
2	1 4 2
1	4 2 1
0	4 2 1 0
3	2 1 0 3
0	2 1 3 0

-Agora, se usarmos a política de alocação local supondo que as molduras são divididas entre A e B usando a alocação igualitária, então 2 molduras são 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 pois somente ele acessou a página 1, somente precisaremos construir a tabela para B. Devido a A ter acessado as páginas 0 e 2, então B acessou, em ordem, as páginas 1, 4, 1, e 3. Logo, como podemos ver pela tabela a seguir, similar à tabela descrita anteriormente,

novamente nenhuma página será substituída porque a página 1 ainda está na memória.

Páginas	Ordenação
1	1
4	1 4
1	4 1
3	1 3

5. (2,0) Considere um computador com um disco de n bytes e com blocos de m bytes, sendo n divisível por m . Suponha ainda que o disco esteja inicialmente vazio. Responda, justificando a sua resposta:

- (a) (1,0) Se a alocação contígua for usada, e se um arquivo A, com a blocos, for armazenado a partir da posição x , quantos bytes do disco um novo arquivo B poderá usar no mínimo e no máximo?

Resp.: Pelo enunciado da questão, teremos exatamente $\frac{n}{m}$ blocos no disco, pois cada bloco tem m bytes e o tamanho n do disco é divisível por m . Como o arquivo A tem a blocos e começa na posição x , e como a alocação contígua é usada, A é armazenado nos blocos de x até $x + a - 1$, sendo que $x + a - 1 < \frac{n}{m}$, pois os blocos são numerados de 0 até $\frac{n}{m} - 1$ e A foi armazenado com sucesso no disco. Como a alocação é contígua, B pode usar todos os blocos de 0 até $x - 1$, ou pode usar todos os blocos de $x + a$ até $\frac{n}{m} - 1$. Logo, o número máximo de blocos que B pode usar é de $\max\{x, \frac{n}{m} - x - a\}$. Já o número mínimo de blocos que B pode usar é 1 se A não usar todos os blocos do disco, ou seja, se $a < n$, ou B não pode ser armazenado no disco se $a = n$.

- (b) (1,0) Como será o mapa de bits desse disco para o caso do item anterior, supondo agora que o arquivo B, de b blocos, possa ser armazenado no disco a partir da posição y , tal que $x + a < y \leq \frac{n}{m} - b$?

Resp.: Como o arquivo B pode ser armazenado a partir da posição y do disco e tem b bytes, B é armazenado nos blocos

de y até $y + b - 1$ do disco. Como ambos os arquivos puderam ser armazenados no disco, significando que o sistema de arquivos está consistente, A e B não terão blocos em comum. Logo, os bits de x até $x + a - 1$ e de y até $y + b - 1$ do mapa de bits serão iguais a 0, e todos os outros bits restantes serão iguais a 1.

6. (1,5) Suponha que o sistema operacional use o mecanismo de proteção por lista de controle de acesso ao gerenciar o acesso aos objetos do sistema pelos processos. Descreva o significado de cada uma das permissões dadas na figura a seguir:

Objeto	Lista com os dominios e as operações
notas.xls	(professor, cederj, RWX)
ad2.pdf	(aluno, *, R-), (professor, so, RW-)
teste.exe	(aluno, so, R-X), (cederj, cederj, --X), (professor, *, RWX)
slide.ppt	(*, so, -W-), (aluno, *, ---)

Resp.: Como vimos na aula 12, quando uma lista de controle de acesso é usada, cada entrada da lista representa um objeto do sistema, e o seu conjunto de ternos informa o que certos usuários do sistema podem fazer com o objeto. Cada terno é composto por um usuário do sistema, o grupo ao qual esse usuário pertence, e as operações que podem ser executadas sobre o objeto. Quando o usuário for “*”, isso significa que as operações são válidas para todos os usuários do grupo. Já se o grupo for “*”, isso significa que as operações são válidas para todos os grupos do mesmo usuário (note que um “*” no usuário e no grupo indica que as operações são válidas para todos os usuários do sistema). Finalmente, o campo de operações é composto por uma string com três caracteres, sendo que o primeiro caractere indica se o objeto pode (R) ou não (-) ser lido, o segundo caractere indica se o objeto pode (W) ou não (-) ser escrito e, finalmente, o terceiro caractere indica se o objeto pode (X) ou não (-) ser executado. A seguir mostramos o significado das entradas para cada arquivo:

- Arquivo notas.xls: o único terno indica que somente o usuário professor do grupo cederj pode ler, escrever ou executar o arquivo.
- Arquivo ad2.pdf: o primeiro terno diz que o usuário aluno de qualquer grupo pode somente ler o arquivo, enquanto que o segundo

terno diz que o usuário professor do grupo so pode ler e escrever no arquivo. Nenhum outro usuário, além dos descritos, pode ler, escrever ou executar o arquivo.

- Arquivo teste.exe: o primeiro terno diz que usuário aluno do grupo so pode ler e executar o arquivo. Já o segundo terno diz que o usuário cederj do grupo cederj pode somente executar o arquivo. Finalmente, o terceiro terno diz que o usuário professor de qualquer grupo pode ler, escrever e executar o arquivo. Assim como antes, nenhum outro usuário, além dos descritos, pode executar operações no arquivo.
- slide.ppt: o primeiro terno diz que qualquer usuário do grupo so pode somente escrever no arquivo, enquanto que o segundo terno diz que o usuário aluno de qualquer grupo não pode fazer nenhuma operação no arquivo. Finalmente, como no caso dos outros arquivos, nenhum usuário, além dos descritos, pode ler, escrever ou executar o arquivo.