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 2019

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.

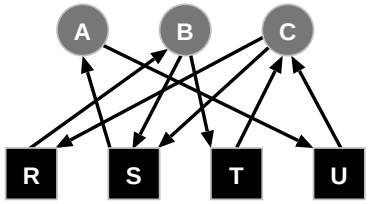
Nome -
Assinatura -

---

1. (1,5) Suponha que três processos, A, B e C, estejam executando no sistema operacional. Suponha ainda que quatro recursos não-preemptivos, R, S, T e U, estejam disponíveis para serem usados pelos processos, e que um processo possa requisitar até dois recursos, sendo bloqueado caso um dos recursos solicitados esteja sendo usado e somente desbloqueado após os recursos solicitados estarem disponíveis. Considere que A tenha obtido S, B tenha obtido R, C tenha obtido T e U e que, após isso, A tenha solicitado U, B tenha solicitado S e T e C tenha solicitado R e S. Responda:

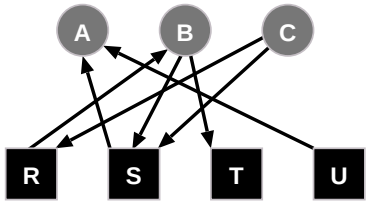
(a) (0,5) Mostre o grafo de recursos após as solicitações dadas no enunciado.

Resp.:



(b) (1,0) Será possível evitar qualquer impasse se somente um dos processos deixar de solicitar os recursos? Justifique a sua resposta.

Resp.: Pelo grafo dado no item anterior, vemos que existem dois impasses, porque existem dois ciclos orientados no grafo. O primeiro ciclo, B→T→C→R→B, define um impasse envolvendo os processos B e C e os recursos R e T. Já o segundo ciclo, A→U→C→S→A, envolve os processos A e C e os recursos S e U. Como C pertence a ambos os ciclos, não teremos mais impasses caso T e U não tenham sido alocados a ele. Somente como ilustração, na figura a seguir mostramos o caso em que, além de C não mais possuir T e U, A obteve U com sucesso (B não pode obter T porque S ainda está alocado a A).



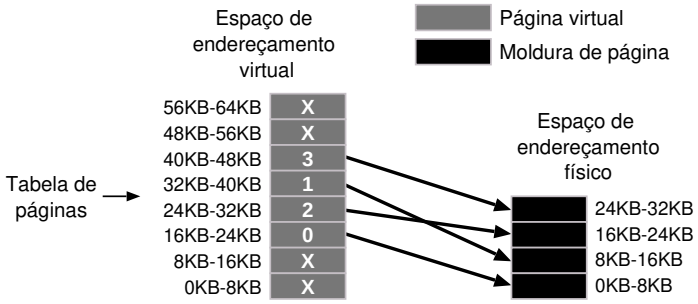
2. (1,5) Suponha que o tamanho da memória física seja de 32KB, que o tamanho da memória virtual seja de 64KB, e que o tamanho da página virtual seja de 8KB. Suponha ainda que as molduras de página inicialmente não estejam mapeando páginas virtuais, e que os endereços virtuais 65534, 12221, 0, 62401, 45678, 24123, 32768 e 32109 tenham

sido acessados nessa ordem. Responda, justificando a sua resposta:

(a) (0,7) Mostre uma figura, similar à dada no slide 19 da aula 8, com os mapeamentos obtidos após todos os endereços virtuais dados no enunciado terem sido acessados. Quando uma falha de página ocorrer, se existirem molduras livres, aquela com o menor número será escolhida para mapear a página. Se não existirem molduras livres, a moldura escolhida será a que mapeou há mais tempo uma página.

Resp.: Como o tamanho da memória virtual é de 64KB e o tamanho da página virtual é de 8KB, então existem 8 páginas virtuais, numeradas de 0 até 7. Como o tamanho da moldura de página também é de 8KB, pois é igual ao tamanho da página virtual, e como o tamanho da memória física é de 32KB, então existem 4 molduras de página, numeradas de 0 até 3. Devido ao campo número da página virtual estar nos bits mais significativos do endereço virtual e o deslocamento estar nos bits menos significativos desse endereço então, para cada endereço virtual dado no enunciado, o número da página virtual que o contém é o quociente da divisão desse endereço pelo tamanho da página virtual. Na tabela a seguir mostramos, na primeira coluna, o endereço virtual acessado. Já na segunda coluna mostramos o número da página virtual com esse endereço, e finalmente, na última coluna, a moldura de página na qual essa página virtual foi mapeada, usando o critério dado no enunciado, se necessário. Devido ao critério usado para mapear as páginas, o mapeamento final, mostrado na figura abaixo da tabela, usará os últimos mapeamentos dados na tabela para cada moldura de página.

Endereço virtual	Página virtual	Moldura mapeando a página
65534	7	0
12221	1	1
0	0	2
62401	7	0
45678	5	3
24123	2	0
32768	4	1
32109	3	2



(b) (0,8) Para cada endereço dado no enunciado para o qual a sua página ainda está mapeada, mostre em binário como esse endereço é dividido nos campos número da página virtual e deslocamento, e como o endereço físico no qual esse endereço foi mapeado é dividido, também em binário, nos campos número da moldura da página e deslocamento.

Resp.: Pelo enunciado, vemos que o endereço virtual tem 16 bits, que o endereço físico tem 15 bits, e que o tamanho da página virtual tem 13 bits. Logo, os 3 bits mais significativos do endereço virtual definem o número da página virtual e os 13 bits menos significativos definem o deslocamento dentro dessa página. Similarmente, os 2 bits mais significativos do endereço físico definem o número da moldura de página e os 13 bits menos significativos definem o deslocamento dentro dessa moldura, porque o tamanho da moldura também é 8KB. Na tabela a seguir, mostramos os endereços pedidos no enunciado em decimal e em binário, usando no caso binário o símbolo "|" para separar os bits mais significativos

dos menos significativos.

Endereço virtual		Endereço físico	
Decimal	Binário	Binário	Decimal
45678	101|1001001101110	11|1001001101110	29294
24123	010|1111000111011	00|1111000111011	7739
32768	100|0000000000000	01|0000000000000	8192
32109	011|1110101101101	10|1110101101101	23917

3. (2,0) Suponha que um processo tenha acessado as páginas virtuais 1, 2, 4, 2, 3, 1, 0, 3, 0 e 4 nessa ordem. Se duas molduras da página, inicialmente vazias, estiverem disponíveis para serem usadas pelo processo, quantas falhas de página serão geradas por cada página diferente acessada pelo processo se o algoritmo LRU for usado pelo sistema operacional? E se agora o algoritmo FIFO passar a ser usado, supondo que três molduras de página, mapeando inicialmente as páginas 4, 3 e 1, copiadas para a memória nessa ordem, estejam disponíveis para o processo? Justifique a sua resposta.

   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 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. Já na segunda coluna mostramos a ordem em que as páginas devem ser escolhidas para substituição. Finalmente, na última coluna, mostramos se o acesso à página gerou uma falha de página. Como podemos ver pela tabela, foram geradas pelas páginas 0, 1, 2, 3 e 4, respectivamente, 1, 2, 1, 2 e 2 falhas.

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

-Como vimos na aula 9, no algoritmo FIFO as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo da sua cópia para a memória. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página copiada há mais tempo para a memória. A tabela a seguir é similar à anterior, mas agora mostra, na segunda coluna, a ordem de escolha segundo o algoritmo FIFO, que agora usa três molduras. Como podemos ver pela tabela, foram geradas pelas páginas 0, 1, 2, 3 e 4, respectivamente, 1, 1, 1, 1 e 2 falhas.

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

4. (1,5) Suponha que cinco processos, A, B, C, D e E, estejam em execução, e que seus tamanhos sejam de, respectivamente, 128KB, 675MB, 1GB,

776MB e 1024KB. Se a segmentação com paginação for usada pelo sistema operacional, se cada processo for armazenado em um segmento, e se os possíveis tamanhos da página virtual forem 32KB, 64MB, 512MB, 1GB e 2GB, qual é o maior conjunto de processos que poderá ser armazenado na memória para cada tamanho de página, supondo uma memória física com 4GB? Se, para um tamanho de página, existirem múltiplos conjuntos, escolha aqueles que minimizarem o desperdício de espaço devido à fragmentação interna dentro das páginas. Justifique a sua resposta.

Resp.: Primeiramente, na tabela a seguir, mostramos o número de molduras de página (igual ao número de páginas virtuais), necessárias para armazenar cada um dos segmentos com os processos. Na primeira coluna mostramos o tamanho da moldura. Na segunda coluna mostramos o número total de molduras para o tamanho dado na primeira coluna. Nas colunas 3 a 7 mostramos a quantidade de molduras usadas pelos segmentos. Finalmente, na última coluna, mostramos o número total de molduras necessárias para armazenar simultaneamente todos os segmentos. Como podemos ver pela tabela, para os tamanhos 32KB, 64MB e 512MB, conseguiremos armazenar todos os cinco segmentos na memória, ou seja, o conjunto com A, B, C, D e E. Porém, para o tamanho de 1GB, poderemos armazenar somente quatro segmentos. Como cada segmento com um processo usa, neste caso, somente uma página, então o espaço desperdiçado devido à fragmentação interna, para cada processo, será 1GB menos o tamanho desse processo. Logo, não deveremos armazenar o processo A com o menor tamanho de 128KB e, com isso, para o tamanho de 1GB, o conjunto será composto pelos processos B, C, D e E. Finalmente, para o tamanho de 2GB, poderemos somente armazenar dois segmentos e, similarmente ao caso de 1GB, escolheremos os processos com os maiores tamanhos (pois cada segmento também usa uma página), implicando em que, neste caso, o conjunto será composto pelos processos C e D.

Tamanho da moldura	Número de molduras	Processos					Total de molduras
		A (128KB)	B (675MB)	C (1GB)	D (776MB)	E (1024KB)	
32KB	131072	4	21600	32768	24832	32	79236
64MB	64	1	11	16	13	1	42
512MB	8	1	2	2	2	1	8
1GB	4	1	1	1	1	1	5
2GB	2	1	1	1	1	1	5

5. (1,5) Suponha que você deseje acessar n posições do arquivo AD2.pdf, numeradas de 1 até n. Se o acesso sequencial for usado, responda, justificando a sua resposta:

   (a) (0,7) Em que ordem os blocos devem ser acessados para que cada posição seja acessada o mínimo de vezes?

   Resp.: Para acessar cada posição o mínimo de vezes (uma vez), basta acessar as posições na ordem crescente, ou seja, na ordem 1, 2, ..., n, já que nessa ordem nenhum **rewind** é necessário.

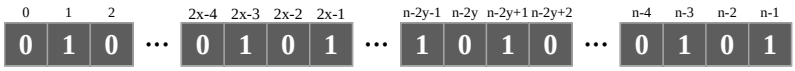
   (b) (0,8) Em que ordem os blocos devem ser acessados para que cada posição seja acessada o máximo de vezes? Qual é o valor do máximo para cada posição?

   Resp.: -Para acessar cada posição o máximo de vezes, basta acessar as posições na ordem decrescente, ou seja, na ordem n, ..., 2, 1, já que ao acessar a posição i, $1 \leq i \leq n$, será necessária a operação **rewind** e depois acessar as posições de 1 até i.
   -Como cada posição i será acessada quando as posições i, $i+1$, ..., n forem acessadas, então a posição i será acessada $n - i + 1$ vezes.

6. (2,0) Considere que o sistema operacional use um mapa de bits para gerenciar os blocos livres de um disco com n blocos de 128KB, n ímpar, numerados de 0 até $n-1$. Suponha que um arquivo A use os x blocos pares iniciais do disco, e que um arquivo B use os y blocos ímpares finais do disco. Responda, justificando a sua resposta:

(a) (0,8) Qual será o mapa de bits após os arquivos A e B serem armazenados no disco?

Resp.: Primeiramente vamos notar que o mapa de bits tem n bits, sendo que o bit i, $0 \leq i \leq n-1$, indica se o bloco i está usado (quando igual a 0) ou livre (quando igual a 1). O arquivo A usa os blocos 0, 2, ..., $2(x-2)$ e $2(x-1)$, porque A usa os x blocos pares iniciais. Finalmente, o arquivo B usa os blocos $n-2y$, $n-2(y-1)$, ..., $n-4$ e $n-2$, porque B usa os y blocos ímpares finais. Logo, no mapa de bits, os bits 0, 2, ..., $2x-4$ e $2x-2$, e os bits $n-2y$, $n-2y+2$, ..., $n-4$ e $n-2$ serão todos iguais a 0. Os bits restantes, ou seja, os bits 1, 3, ..., $2x-3$, os bits de $2x-1$ até $n-2y-1$, e os bits $n-2y+1$, $n-2y+3$, ..., $n-3$ e $n-1$ serão todos iguais a 1. A figura a seguir ilustra o mapa.



(b) (0,4) Quais condições devem ser impostas a x e y para que o sistema de arquivos seja consistente?

Resp.: Como A usa os blocos pares iniciais do disco e B usa os blocos ímpares finais do disco, implicando em que A e B não usam os mesmos blocos então, para o sistema de arquivos ser consistente, basta que x seja menor ou igual ao número de blocos pares no disco, e que y seja menor ou igual ao número de blocos ímpares do disco. Agora, como n é ímpar e os blocos são numerados de 0 até $n-1$, temos $\frac{n+1}{2}$ blocos pares e $\frac{n-1}{2}$ blocos ímpares. Assim, o sistema será consistente se $x \leq \frac{n+1}{2}$ e $y \leq \frac{n-1}{2}$.

(c) (0,8) Qual o tamanho máximo de um novo arquivo C, em KB, se ele for armazenado em blocos consecutivos do disco?

Resp.: Como o arquivo C será armazenado em blocos consecutivos do disco, como antes do final de A teremos somente blocos ímpares livres, e como após o início de B teremos somente blocos pares livres, então C poderá usar todos os blocos de $2(x-1)+1 = 2x-1$ até $n-2y-1$. Agora, como teremos $n-2y-1-(2x-1)+1 =$

$n-2x-2y+1$ blocos entre o final de A e o início de B, como o tamanho de cada bloco é de 128KB, e como para maxizar o tamanho, C deve usar todos os bytes de todos os blocos, então o tamanho máximo será de $128(n-2x-2y+1) = (128n-256x-256y+128)$ KB. A figura a seguir mostra a alocação dos três arquivos no disco.