



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 2011

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

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.

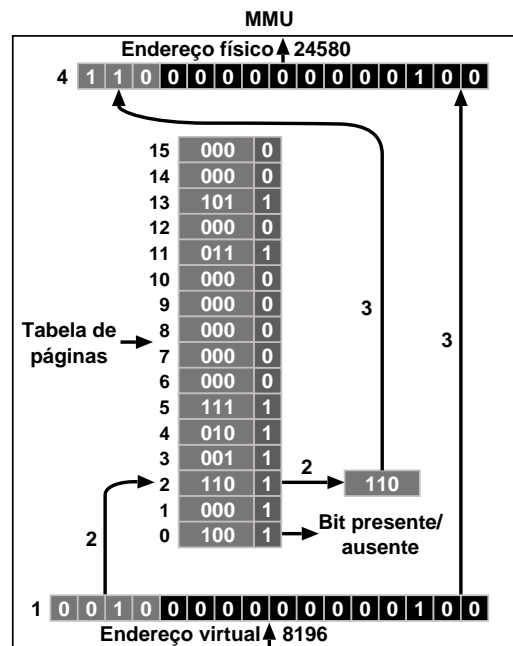
Observação: A questão 5 é opcional. Se você tentar resolvê-la, você poderá ganhar até 2,0 pontos adicionais na nota da prova, mas a sua nota ainda será limitada a 10,0 pontos.

Nome -
Assinatura -

1. (2,0) Suponha que um sistema operacional implemente a seguinte alternativa a de numerar os recursos do sistema, com a intenção de evitar a ocorrência de impasses. Um contador interno é inicializado em 0 sempre que o sistema é inicializado. Além disso, quando o sistema cria um processo, ele primeiramente incrementa o contador em 1 unidade, e depois associa o valor deste contador ao processo criado. Para evitar um impasse, o sistema somente bloqueará um processo que desejar um recurso possuído por um outro processo se este último tiver um valor menor do que o valor desse processo. Este método alternativo também evitará a ocorrência dos impasses? Justifique a sua resposta.

Resp.: Sim, o método alternativo também garante que os impasses sejam evitados. A razão é que, usando esse método, é impossível que se forme um ciclo de espera entre os processos. Note que em um ciclo desses cada processo participante teria que ter um número superior ao do próximo processo no ciclo, o que não pode ser verdade para todos os processos.

2. (3,0) Considerando a figura dada a seguir, que mostra como o endereço virtual 8196 é mapeado no endereço físico 24580, responda as seguintes perguntas:



- (a) (1,0) Como é dividido, em bits, um endereço virtual? E um endereço físico?

Resp.: -Pela figura, vemos que um endereço virtual é composto por 16 bits, sendo que os 4 bits superiores definem o número da página virtual que contém o endereço (esse número é usado como um índice na tabela de páginas), e que os 12 bits inferiores definem o deslocamento desse endereço dentro da página virtual.

-Também pela figura, vemos que um endereço físico é composto por 15 bits, sendo que os 3 bits superiores definem o número da moldura de página que contém o endereço, e que os 12 bits inferiores definem o deslocamento desse endereço dentro da moldura. Note que o deslocamento é o mesmo do endereço virtual, pois os tamanhos das páginas e das molduras são iguais.

- (b) (1,0) Quais endereços virtuais geram uma falha de página ao serem usados em uma instrução do processador?

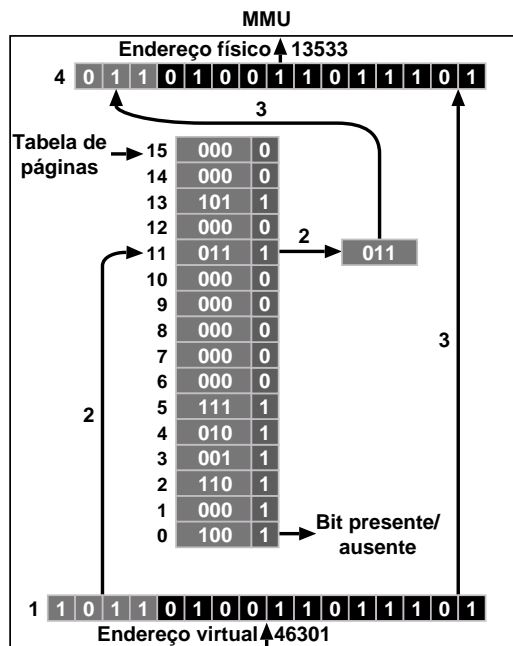
Resp.: Todos os endereços virtuais contidos em páginas virtuais

cujos bits presente/ausente são 0. Esses endereços são dados na tabela a seguir.

Página	Faixa de endereços virtuais
6	24576-28671
7	28672-32767
8	32768-36863
9	36864-40959
10	40960-45055
12	49152-53247
14	57344-61439
15	61440-65535

- (c) (1,0) Mostre, usando uma figura similar à dada no enunciado da questão, o endereço físico para o qual o endereço virtual 46301 é convertido.

Resp.: Para determinar o endereço físico, primeiramente devemos converter o endereço virtual 46301 para binário, obtendo o valor 1011010011011101. Isso significa que o endereço está na página de número decimal 11 e que o deslocamento dentro dessa página é 1245. A seguir damos a figura pedida, onde vemos que o endereço físico é, em binário, 011010011011101, ou seja, o endereço é 13533 na moldura de página 3.



3. (3,0) Suponha que um sistema operacional use o algoritmo LRU ao gerenciar as páginas virtuais. Suponha ainda que quatro molduras, inicialmente vazias, foram alocadas a um processo. Responda as seguintes perguntas, supondo que o processo acessou, em ordem, as páginas, 2, 3, 4, 5, 3, 0, 2, 1, 4 e 6, justificando todas as respostas com base na descrição de como o algoritmo é executado quando cada uma das páginas é acessada.
- (1,0) Para quais páginas o acesso não gerou uma falha de página?
 - (1,0) Se o processo agora acessar a página 10, qual página será substituída?
 - (1,0) Quais páginas das já acessadas pelo processo podem novamente ser acessadas sem gerar uma nova falha de página?

Resp.: Primeiramente vamos mostrar, na tabela a seguir, a sequência de acessos às páginas dadas na questão para depois responder a cada uma das perguntas. 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

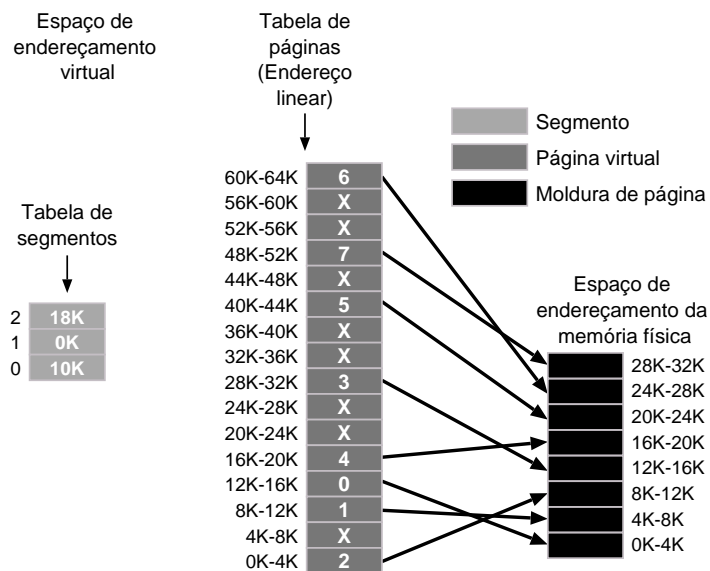
há mais tempo. Na tabela dada a seguir mostramos, em cada linha, o que ocorre ao acessarmos as páginas na ordem dada no enunciado. Para cada uma dessas linhas mostramos na primeira coluna a página que é acessada, na segunda coluna a ordem em que as páginas devem ser escolhidas e finalmente, na terceira coluna, se o acesso gera uma falha de página. A primeira página segundo a ordenação, que é aquela a ser substituída, é mostrada em negrito. Agora, vamos então responder cada uma das perguntas da questão:

Páginas	Ordenação	Ocorreu uma falha?
2	2	Sim
3	2 3	Sim
4	2 3 4	Sim
5	2 3 4 5	Sim
3	2 4 5 3	Não
0	4 5 3 0	Sim
2	5 3 0 2	Sim
1	3 0 2 1	Sim
4	0 2 1 4	Sim
6	2 1 4 6	Sim

- (a) Pela tabela, vemos que somente o segundo acesso à página 3 não gerou uma falha de página.
- (b) Como a página 10 não está na memória, precisamos substituir uma página virtual. Pela última linha da tabela, que mostra as páginas atualmente na memória ordenadas pelo tempo em que foram acessadas pela última vez, após o processo acessar todas as páginas dadas na questão, vemos que a página a ser substituída é a 2.
- (c) Para que um acesso a uma página não gere uma falha, essa página precisa estar na memória. A última linha da tabela mostra quais páginas ficaram na memória após o processo acessar todas as páginas dadas na questão. Logo, as páginas virtuais 1, 2, 4 e

6 mostradas nessa última linha não irão gerar uma falha se forem novamente acessadas pelo processo.

4. (2,0) Suponha que um computador utilize uma técnica de segmentação com paginação similar à do processador Pentium da Intel. Na figura a seguir mostramos, na tabela de segmentos, o endereço linear inicial no qual cada segmento usado por um processo é mapeado. Nesta figura também mostramos, na tabela de páginas do espaço de endereçamento linear, como as páginas virtuais são mapeadas nas molduras de página da memória física. Usando essas tabelas, forneça o endereço linear, e o endereço físico somente se o endereço linear estiver mapeado em uma moldura de página, para cada um dos seguintes endereços virtuais bidimensionais (segmento, endereço dentro do segmento):



- (a) (0,7) (0,2560).
- (b) (0,7) (1,3456).
- (c) (0,6) (2,9584).

Resp.: No Pentium, cada segmento é mapeado a partir do endereço inicial dado na tabela de segmentos. Todos os endereços iniciais dos

segmentos pertencem a um espaço de endereçamento comum, chamado de **espaço de endereçamento linear**, que é paginado do mesmo modo que o espaço de endereçamento virtual visto na aula 9. Logo, o endereço linear correspondente a um endereço de um segmento é igual à soma deste endereço mais o endereço linear a partir do qual o segmento é mapeado. Pela figura vemos que o espaço de endereçamento linear tem 64K, com endereços de 16 bits variando de 0 até 65535. Já o espaço de endereçamento físico tem 32K, com endereços de 15 bits variando de 0 até 32767. Como as páginas possuem 4K de tamanho, o espaço de endereçamento linear é dividido em 16 páginas, e o espaço de endereçamento físico é dividido em 8 molduras de página. Assim, cada endereço linear é dividido do seguinte modo: 4 bits para o número da página e 12 bits para o deslocamento. No que segue, cada endereço linear também é dado em binário, separando o número da página e o deslocamento por “|”.

- (a) Endereço virtual (0, 2560): como o segmento 0 é, segundo a tabela, mapeado para o endereço linear 10K ou 10240, então o endereço linear correspondente ao endereço virtual é 10240 mais 2560, ou seja, é igual a 12800. Agora, para o endereço linear 12800 (0011 | 001000000000), vemos que estamos acessando a palavra 512 da página virtual 3 (com endereços de 12288 até 16383). Pela figura, vemos que a página virtual 3 está mapeada na moldura de página 0 (com endereços de 0 até 4095). Logo, o endereço físico é 0 (o primeiro endereço da moldura) mais 512 (o deslocamento), isto é, é 512.
- (b) Endereço virtual (1, 3456): já para o segmento 1, o endereço linear também é 3456, pois segundo a tabela do segmento, o endereço virtual foi mapeado a partir do endereço linear 0K ou 0. No caso do endereço linear 3456 (0000 | 110110000000), estamos acessando a palavra 3456 da página virtual 0 (com endereços de 0 até 4095). A figura indica que esta página está mapeada na moldura de página 2 (com endereços de 8192 até 12287). Com isso, o endereço físico é 8192 (o endereço inicial da moldura) mais 3456 (o deslocamento), isto é, é 11648.
- (c) Endereço virtual (2, 9584): no caso do segmento 2 que, segundo a tabela de segmentos, é mapeado a partir do endereço linear 18K ou 18432, o endereço linear correspondente ao endereço virtual é

18432 mais 9584, ou seja, é 28016. Esse endereço linear 28016 (0110 | 110101110000) está na palavra 3440 da página virtual 6 (com endereços de 24576 até 28671), a qual não está na memória.

5. (2,0) Suponha que o disco do computador tem n bytes, e que cada um dos seus blocos tem b bytes. Supondo que não é possível usar parcialmente bytes e blocos do disco, qual seria o espaço adicional gasto, em bytes, pelo método de alocação por lista encadeada? E se o método usado agora for o de alocação por lista encadeada utilizando um índice, supondo que a tabela com os índices é armazenada no disco?

Resp.: Como um disco de um computador não pode ter blocos de tamanhos diferentes, então podemos supor que n é múltiplo de b , e que $\frac{n}{b}$ define o número total de blocos do disco. Como um endereço de um bloco varia de 0 até o número de blocos menos 1, ou seja, até $\frac{n}{b} - 1$, então precisamos de $\lceil \log_2(\frac{n}{b}) \rceil$ bits para armazená-lo, ou seja, $\lceil \frac{\lceil \log_2(\frac{n}{b}) \rceil}{8} \rceil = \lceil \frac{\log_2(\frac{n}{b})}{8} \rceil$ bytes, pois não podemos usar bytes parcialmente. Para o método de alocação por lista encadeada, como cada bloco do disco armazena o endereço do próximo bloco do arquivo, então gastaremos $\frac{n}{b} \lceil \frac{\log_2(\frac{n}{b})}{8} \rceil$ bytes do disco. Já para o método de alocação por lista encadeada utilizando um índice o espaço gasto é igual ao da tabela com os índices. Como a tabela tem $\frac{n}{b}$ entradas, e como cada entrada armazena um endereço com tamanho de $\lceil \frac{\log_2(\frac{n}{b})}{8} \rceil$ bytes, então a tabela também gasta $\frac{n}{b} \lceil \frac{\log_2(\frac{n}{b})}{8} \rceil$ bytes. Porém, como essa tabela deve usar completamente os blocos nos quais ela é armazenada, mesmo que o seu tamanho não seja múltiplo do tamanho b de um bloco, então a tabela gasta $\lceil \frac{\frac{n}{b} \lceil \frac{\log_2(\frac{n}{b})}{8} \rceil}{b} \rceil$ blocos do disco ou $b \lceil \frac{\frac{n}{b} \lceil \frac{\log_2(\frac{n}{b})}{8} \rceil}{b} \rceil$ bytes.