

DI/CT/UFES - 3ª. Prova de Sistemas Operacionais I
Período: 2009/2 - Profª. Roberta Lima Gomes
GABARITO

- 1.i) Tamanho da página = 2^c bytes = 2^{10} bytes = 1 kbytes
- 1.ii) Espaço de endereçamento lógico = 2^{16} = 64 kbytes
- 1.iii) Nº de molduras = 2^x , onde $x = n^\circ$ de bits do endereço físico - $c \Rightarrow x = 12 - 10 \Rightarrow n^\circ$ de molduras = 4
- 1.iv) Espaço de endereçamento físico = 2^{12} = 4 kbytes
- 1.ii) Nº de páginas virtuais = $2^{a+b} = 2^6 = 64$ páginas

2.a) **2a. chance** – este algoritmo implementa uma melhoria sobre o FIFO onde, antes de remover a página mais antiga (que está há mais tempo em memória), seu bit R é verificado: se R=0, a página é substituída; se R=1, a página vai para fim da fila, como se houvesse sido carregada agora e seu bit é setado para 0.

A fila FIFO é:

Pag.5(R=1) -> Pag.10 (R=1) -> Pag.7 (R=0)-> Pag.8 (R=0),

mas como a Pag.5 e a Pag.10 têm seus respectivos bits R setados, com o 2ª. chance elas vão para o fim da fila:

Pag.7 (R=0)-> Pag.8 (R=0) -> Pag.5(R=0) -> Pag.10 (R=0),

Desta forma, a página que será removida da memória é a Pag. 7

NRU – neste algoritmo as páginas são classificadas conforme os valores dos bits R e M:

Classe 0: Not referenced, not modified (R=0 , M=0) --- Pag.8

Classe 1: Not referenced, modified (R=0 , M=1) -- Pag.1

Classe 2: referenced, not modified (R=1 , M=0)

Classe 3: referenced, modified (R=1 , M=1) -- Pag.5 e Pag.10

São selecionadas primeiramente as páginas da classe 0, não havendo nenhuma página nesta classe, passa-se para a classe 1 e assim por diante.

Desta forma, a página que será removida da memória é a Pag. 8

LRU – neste algoritmo Páginas são ordenadas conforme o tempo da última referência, onde a primeira página a ser removida da memória deve ser aquela que se encontra há mais tempo sem ser referenciada.

A fila LRU é

Pag.7(230) -> Pag.8 (233) -> Pag.5 (245)-> Pag.10 (250),

Classe 0: Not referenced, not modified (R=0 , M=0) --- Pag.8

Classe 1: Not referenced, modified (R=0 , M=1) -- Pag.1

Classe 2: referenced, not modified (R=1 , M=0)

Classe 3: referenced, modified (R=1 , M=1) -- Pag.5 e Pag.10

Desta forma, a página que será removida da memória é a Pag. 7

2.a) **Processo de Tradução:** Um endereço virtual é gerado durante o processamento da aplicação e encaminhado pela CPU para a MMU (memory management unit), um dispositivo do processador cuja função é transformar o endereço virtual em físico e solicitar este último endereço ao controlador de memória. A conversão de endereços virtuais em físicos baseia-se em tabelas de páginas, que são estruturas de dados mantidas pelo sistema operacional em memória. Cada endereço virtual é dividido em [nº da pag.] [deslocamento]. Cada tabela é indexada pelo número da página de um endereço virtual e contém o número da moldura correspondente ou a indicação de que a página não está em memória (bit de validade). Como o acesso à tabela de páginas é muito lento, pois está em memória, a MMU possui uma memória associativa chamada TLB (Translation Lookaside Buffer -- buffer de tradução de endereços), que consiste em uma pequena tabela contendo as entradas da tabela de página referentes aos últimos endereços virtuais solicitados. Desta forma, usando o número da página do endereço virtual, caso a entrada correspondente não esteja na TLB, a MMU acessa a tabela de páginas do processo na memória usando o número da página como índice. Ela então verifica se a página encontra-se realmente na memória (bit de validade).

- Se sim, usa-se então o número da moldura indicado nessa entrada da tabela de páginas. Esse número é então concatenado ao deslocamento do endereço virtual, gerando o endereço físico
- Se não, a MMU gera um *PageFault*. O SO realiza o tratamento dessa interrupção que consiste em aplicar algum algoritmo de substituição (no caso deste problema será o LRU), trazendo a página requisitada para a memória e então o processo é todo repetido (isto é o endereço virtual é novamente repassado da CPU para a MMU). Mas desta vez não ocorrerá *Page Fault*.

Tradução dos Endereços:

No problema, como a página tem 2^{11} bytes, temos o endereço virtual dividido conforme a seguir:

| 5 bits | 11 bits |
|-------------------|---------------------|
| <i>n° da pag.</i> | <i>deslocamento</i> |

- **Endereço 2000**

| | |
|-------|-------------|
| 00000 | 11111010000 |
|-------|-------------|

Isso implica que ele encontra-se na pag. 0, a MMU verifica que ela não encontra-se na memória e gera um *PageFault*. Como resultado essa página é trazida para a primeira moldura (moldura 00, moldura em que se encontrava a pag.7 que, seguindo o LRU, deve ser removida da memória). Note que o n° da moldura deve conter pelo menos dois bits dado que temos 4 molduras. Então finalmente a MMU fará a tradução substituindo o n° da página pelo n° da moldura, gerando o seguinte endereço físico:

| | |
|----|-------------|
| 00 | 11111010000 |
|----|-------------|

- **Endereço 12000**

| | |
|-------|-------------|
| 00101 | 11011100000 |
|-------|-------------|

Isso implica que ele encontra-se na pag. 5, a MMU verifica que ela encontra-se na memória, na terceira moldura (moldura 10). Então finalmente a MMU fará a tradução substituindo o n° da página pelo n° da moldura, gerando o seguinte endereço físico:

| | |
|----|-------------|
| 10 | 11011100000 |
|----|-------------|

3) No caso da memória virtual, o gerenciamento, para que uma troca entre as páginas de um processo ocorra mais rapidamente, em geral adia a retirada de páginas modificadas para que se evite que se tenha que fazer a sincronização dos dados com o disco antes de retirá-la, desta forma agiliza-se a troca. No caso do sistema de arquivos, o maior interesse é que os dados sempre estejam consistentes. Dessa forma a preocupação maior é gravar os blocos que foram modificados para que em casos de falhas do sistema, haja uma minimização das perdas e, conseqüentemente, da inconsistência. [Resposta de Diego Ribon Lima]

4) 6 acessos (diagrama a fazer...)