

GABARITO DA PROVA 3

1ª. Questão:

Determinado computador fornece a seus usuários um espaço de memória virtual de 232 Kbytes. O computador tem 218 Kbytes de memória física. A memória virtual é implementada por paginação e o tamanho de página é 4096 bytes. Um processo de usuário gera o endereço virtual 123456. Explique como o sistema estabelece a posição física correspondente.

Resp: Em um sistema de memória virtual baseada em paginação, o endereço virtual apresenta o formato (p, d) , onde p = número da página e d = deslocamento dentro da página. A MMU (Memory Management Unit) deve, portanto, transformar primeiramente o endereço virtual "linear" 123456 para este formato (no exemplo da questão, o sistema possui 58 páginas virtuais ($232K/4k = 58$) e cada endereço virtual possui 18 bits, assim divididos: 6 bits para armazenar o número p da página e 12 bits para armazenar o deslocamento d). Para se determinar os valores de página e deslocamento, o endereço virtual 123456 é dividido pelo tamanho da página, que é de 4096 bytes. O resultado desta divisão implica em: página = 30, deslocamento = 576. De posse do número da página, a tabela de páginas do processo é então acessada para se descobrir se esta página está mapeada na memória. Se estiver, o número da moldura correspondente é usado para, juntamente com o deslocamento, calcular o endereço físico a ser acessado (endereço físico = número da moldura * tamanho da moldura + deslocamento). O tamanho da moldura é, obviamente, igual ao tamanho da página. Caso a página acessada não esteja na memória (o que pode ser deduzido observando-se o valor do bit de validade/presença da entrada na tabela de páginas), é gerada uma exceção (*page fault* - falta de página) e um algoritmo de substituição de páginas é executado para se determinar qual das páginas mapeadas na memória deve ser substituída para que a página recém-acessada seja carregada na moldura em que se encontra a página selecionada para expulsão. A tabela de páginas é atualizada. Por exemplo, supondo que a página 30 esteja mapeada na moldura 10 da memória real, a posição física correspondente seria assim calculada: $10 \times 4096 + 576$ (moldura x tamanho da moldura + deslocamento), resultando em acesso ao endereço físico 41.536. Caso o hardware possua um sistema de cache de páginas implementado via TLB - Translation Look Aside Buffer, a página a ser acessada é verificada antes se está na TLB. Caso não esteja mapeada, somente depois é examinada a tabela de páginas do processo na memória. O endereço da tabela de páginas na memória faz parte do contexto do processo.

Palavras-chave: MMU, formato do endereço virtual (página, deslocamento), tabela de páginas, moldura, bit de validade/presença, cálculo do endereço físico (moldura * tamanho da moldura +

deslocamento), falta de página (page fault), algoritmo de substituição de página, atualização da tabela de páginas, TLB.

2ª. Questão:

Considere um sistema com páginas de 4K, endereçamento lógico de 16 páginas, e endereçamento físico de 8 frames. Considere a seguinte tabela de página do processo em execução:

	bit validade	moldura
0	1	2
1	1	1
2	1	6
3	1	0
4	1	4
5	1	3
6	0	-
7	0	-
8	0	-
9	1	5
10	0	-
11	1	7
12	0	-
13	0	-
14	0	-
15	0	-

- Mostre como fica uma Tabela Invertida para este sistema, com o mesmo mapeamento, e como endereço 8300 é traduzido com esta tabela.
- Mostre como fica uma tabela em dois níveis para este sistema com o mesmo mapeamento, se o primeiro e o segundo níveis suportarem 4 entradas cada. Além disso, mostre como o endereço do item a) é traduzido com esta tabela.

Resp: a) A tabela de páginas invertida é indexada pelo número da moldura e não pelo número da página, como acontece na tabela de páginas convencional. O tamanho de uma tabela de páginas invertida é diretamente proporcional ao tamanho da memória física (número de molduras) e não ao tamanho do espaço de endereçamento lógico (número de páginas), o que reduz o tamanho final da tabela de páginas. Na questão, é informado que o sistema possui 8 frames (molduras); com isso, conclui-se que a tabela invertida possui apenas 8 entradas, ao contrário das 16 entradas da tabela convencional mostrada na questão. A figura abaixo ilustra a tabela invertida:

*** Figura ***

Para a tradução do endereço virtual 8300 deve-se, primeiramente, determinar a página lógica que contém esse endereço. Para isso, divide-se 8300 pelo tamanho da página, 4096. O resultado desta divisão indica que o endereço 8300 corresponde à página 2 e deslocamento 128 ($p=2$, $desloc=128$). Consultando-se a tabela de páginas invertida (*existe alguma moldura que contém a página 2 do processo em questão?*), verifica-se que é a moldura 6 que mapeia a página 2. O endereço físico a ser acesso acessado pode ser assim calculado: Calculando: $6 \times 4096 + 128 = 24704$.

b) Um endereço virtual em um sistema de gerência de memória virtual baseada em paginação de dois níveis, com 4 entradas no primeiro nível, 4 no segundo nível e tamanho de página 4096 possui o seguinte formato:

*** Figura ***

onde: $p_1 = 2$ bits (índice da tabela de primeiro nível), $p_2 = 2$ bits (índice da tabela de segundo nível) e deslocamento = 12 bits. Colocando-se o endereço virtual 8300 em formato binário, $(8300_{10} = 00.10.000001101100_2)$, verificamos que $p_1=00_2$ (entrada de índice 0 da tabela de primeiro nível) e $p_2=10_2$ (entrada de índice 2 da tabela de segundo nível). Numa máquina com espaço de endereçamento virtual de 16 páginas de 4K, uma tabela de primeiro nível com 4 entradas divide o espaço de endereçamento em 4 blocos de 16k ($4 \times 16K = 64K$). Assim, a entrada $p_1 = 00_2$ (0_{10}) corresponde a um endereço da primeira entrada desta tabela, ou seja, dos primeiros 16K deste espaço de endereçamento. Como a tabela de segundo nível também possui 4 entradas, ela divide este espaço de endereçamento de 16K em 4 blocos de 4K cada ($4 \times 4K = 16K$). Assim, a entrada $p_2 = 10_2$ (2_{10}) corresponde a um endereço da terceira entrada da tabela de segundo nível, ou seja, localizado entre 8 e 12K. Examinando a tabela, vemos que esta página está mapeada na moldura 6.