

1. Se o algoritmo de substituição FIFO é usado por quatro molduras de página e oito páginas virtuais, quantas faltas de página ocorrerão com a cadeia de referências 0172327103 se os quatro quadros estão inicialmente vazios? Agora repita este problema para LRU.

FIFO: De início, haverá falta para as 4 primeiras:

0 1 7 2  
1 7 2 3 - falta (3), 2, 7 e 1 estão lá  
7 2 3 0 - falta (0), 3 está lá  
6 faltas.

LRU: De início, falta para as 4 primeiras (em ordem de tempo de acesso):

0 1 7 2  
1 7 2 3 - falta (3)  
1 7 3 2 - acesso à 2  
1 3 2 7 - acesso à 7  
3 2 7 1 - acesso à 1  
2 7 1 0 - falta (0)  
7 1 0 3 - falta (3)  
7 faltas.

2. Suponha que os bits R, para as páginas de B a A, sejam 11011011, respectivamente. Qual página será removida pelo algoritmo segunda chance, em caso de page fault?

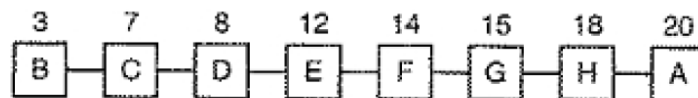


Figura 1: Tabela do exercício

Página D.

3. Um computador pequeno tem quatro molduras de página. No primeiro tique de relógio, os bits R são 0111. Nos tiques subsequentes os valores são 1011, 1010, 1101, 0010, 1010, 1100, 0001. Se o algoritmo do envelhecimento (aging) é usado com um contador de 8 bits, quais os valores dos quatro contadores após o último tique?

Aging: Cada página recebe um contador. A cada tique, o contador é deslocado à direita, e o bit R correspondente adicionado à esquerda

Contador	0111	1011	1010	1101	0010
00000000	00000000	10000000	11000000	11100000	01110000
00000000	10000000	01000000	00100000	10010000	01001000
00000000	10000000	11000000	11100000	01110000	10111000
00000000	10000000	11000000	01100000	10110000	01011000

Figura 2: Tabela do exercício

4. Suponha que o intervalo de tempo do conjunto de trabalho (working set) é  $t = 400$  na figura abaixo. Qual página será removida?

A página mais antiga com  $R = 0$  é a marcada com 1620. Sua idade é  $2204 - 1620 = 584 > 400$ , portanto esta será a removida.

<i>Contador</i>	<i>1010</i>	<i>1100</i>	<i>0001</i>
01110000	10111000	11011100	01101110
01001000	00100100	10010010	01001001
10111000	11011100	01101110	00110111
01011000	00101100	00010110	10001011

Figura 3: Tabela do exercício

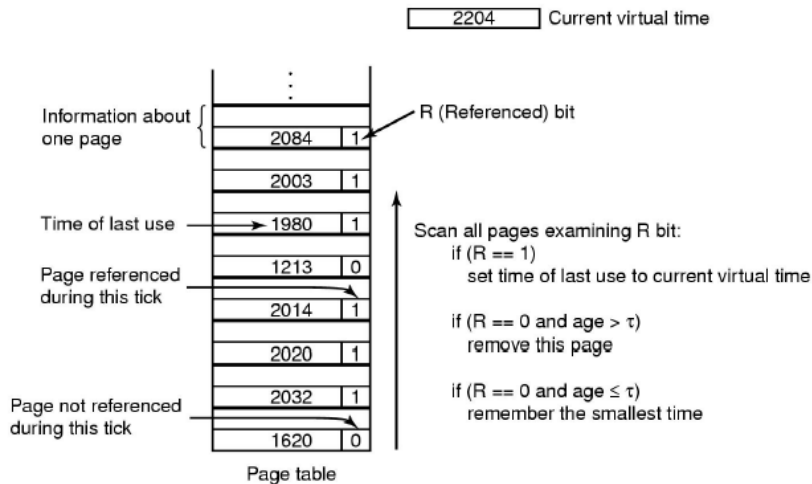


Figura 4: Tabela do exercício

5. o ponteiro do relógio aponta para a página com  $R=0$ . Se  $t = 400$  (working set), essa página será removida? O que acontecerá se  $t = 1000$ ? Suponha que o horário atual é 2204.

Como  $22041213 = 991$ , a página marcada com 1213 será removida, se  $t = 400$ . Se  $t = 1000$ , a página não será removida, de início, pois está dentro do working set. Contudo, ela será marcada, pois é a mais antiga com  $R=0$ . Como todas estão no working set, na próxima volta do algoritmo ela será removida.

6. Um computador tem quatro molduras de página. O tempo de carregamento de página na memória, o instante do último acesso e os bits R e M para cada página são mostrados a seguir (os tempos estão em tiques de relógio):

- Qual página será trocada pelo NRU?  
2, pois  $(R,M) = (0,0)$
- Qual página será trocada pelo FIFO?  
3, é a mais antiga
- Qual página será trocada pelo LRU?  
1, é a menos recentemente usada
- Qual página será trocada pelo Segunda Chance?  
2, é a mais antiga, com  $R=0$

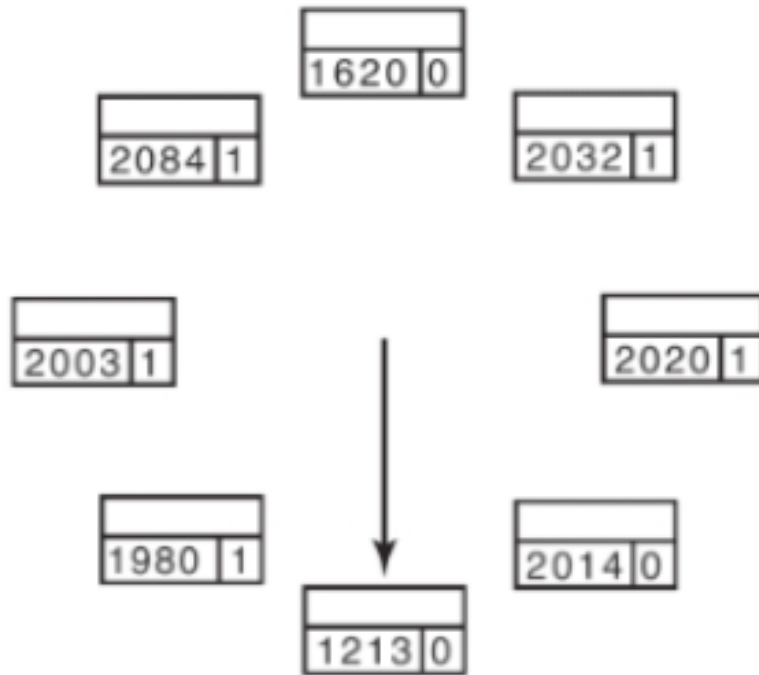


Figura 5: Tabela do exercício

Página	Carregado	Última referência	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Figura 6: Tabela do exercício

7. Considere a matriz

```
int x[64][64]
```

Suponha que um sistema tenha 4 molduras de página e que cada moldura seja de 128 palavras (um inteiro ocupa uma palavra). Os programas que manipulam o arranjo  $x$  se ajustam exatamente a uma página e sempre ocupam a página 0. Os dados sofrem swap nas outras 3 molduras. O arranjo  $x$  é armazenado segundo a ordem da fila maior (isto é,  $x[0][1]$  segue  $x[0][0]$  na memória, com as linhas concatenadas). Qual dos dois fragmentos de código a seguir gerará o menor número de faltas de página? Explique e calcule o número total de faltas de página.

*Fragmento A*

```
for (int j=0; j<64; j++)
    for (int i=0; i<64; i++) x[i][j] = 0;
```

Figura 7: Tabela do exercício

O fragmento B acessará  $64 \times 64 = 4096$  palavras consecutivas na memória. Como o programa ocupa uma página, elas começarão no início da seguinte. Com 128 palavras em cada moldura, teremos  $4096/128 = 32$  page faults no processo.

## *Fragmento B*

```
for (int i=0; i<64; i++)  
    for (int j=0; j<64; j++) x[i][j] = 0;
```

Figura 8: Tabela do exercício

No caso do fragmento A, o laço pula de 64 em 64 palavras. Como cada moldura armazena 128, isso ocasionará um total de  $128/64 = 2$  acessos por página somente (ou seja, serão lidas somente 2 palavras por página). Num universo de 4096 palavras, isso leva a  $4096/2 = 2048$  page faults.

8. Um computador provê, a cada processo, 65536 bytes de espaço de endereço dividido em páginas de 4096 bytes. Um determinado programa tem um segmento de texto de 32768 bytes, um segmento de dados de 16386 bytes e um segmento de pilha de 15870 bytes. Esse programa se encaixa no referido espaço de endereçamento? Se o tamanho da página fosse de 512 bytes ele se encaixaria? (Lembre-se de que uma página não pode conter partes de dois segmentos diferentes)

Com 65536 e páginas de 4096 temos  $65536/4096 = 16$  páginas. O segmento de texto consome  $32768/4096 = 8$  páginas, o de dados  $16386/4096 = 4.0005 = 5$  páginas, e a pilha  $15870/4096 = 3.87 = 4$  páginas, num total de 17. Assim, não caberá na RAM.

Já com páginas de 512 temos  $65536/512 = 128$  páginas. O segmento de texto consome  $32768/512 = 64$  páginas, o de dados  $16386/512 = 32.004 = 33$  páginas, e a pilha  $15870/512 = 30.996 = 31 = 4$  páginas, num total de 128, então caberá na RAM.

9. Uma página pode estar em dois conjuntos de trabalho ao mesmo tempo? Explique

Se o SO permitir compartilhamento de página (ex: texto do programa), sim.

10. Tem-se observado que o número de instruções executadas entre faltas de página é diretamente proporcional ao número de molduras de página alocada para um programa. Se a memória disponível for duplicada, o intervalo médio entre faltas de página também será duplicado. Suponha que uma instrução normal leve um microssegundo, mas, se uma falta de página ocorrer, ela levará 2001 microssegundos (isto é, 2 ms para tratar a falta). Se um programa leva 60 s para executar - período em que ele terá 15 mil faltas de página -, quanto tempo ele levaria para executar se existisse duas vezes mais memória disponível?

Como cada falta leva 2000 micro seg, 15000 faltas tomarão 30s (30.000.000 micro seg). Assim, o programa gastará 30s tratando tão somente de faltas de página e 30s processando instruções. Ao dobrarmos a memória, o número de faltas deve cair pela metade, gastando  $30/2 = 15$ s. Assim, o programa levará 30s (instruções) + 15s (faltas) = 45s.

11. Uma instrução em linguagem de máquina para carregar uma palavra de 32 bits para dentro de um registrador contém o endereço de 32 bits da própria palavra a ser carregada. Qual o número máximo de faltas de página que essa instrução pode causar?

A instrução poderia estar parte em uma página e parte em outra, causando 2 possíveis faltas somente para processar a instrução: uma para achar a primeira parte da instrução, e outra para achar a segunda parte. A palavra de memória poderia também estar parte numa página e parte em outra, causando outras 2 faltas, um total de 4. Se, contudo, o sistema exige que as palavras estejam alinhadas na memória (que não é o caso do Pentium), então serão 2 faltas para a instrução e 1 para a palavra de dados (total de 3).

12. Consideremos um programa que tenha os dois segmentos mostrados a seguir, consistindo de instruções no segmento 0 e de dados de leitura/escrita no segmento 1. O segmento 0 tem permissão leitura/execução e o segmento 1 tem permissão leitura/escrita. O sistema de memória é um sistema de memória virtual paginado por demanda, com endereços virtuais que têm um número de página de 4 bits e um deslocamento de 10 bits. As tabelas de páginas e permissões são as seguintes:

Para cada um dos seguintes casos, dê o endereço de memória real (efetiva) que resulta da tradução de endereço dinâmico ou identifique o tipo de erro que ocorre (seja erro de página ou de proteção)

Segmento 0		Segmento 1	
Leitura/Execução		Leitura/Escrita	
Página Virtual	Moldura	Página Virtual	Moldura
0	2	0	Em disco
1	Em disco	1	14
2	11	2	9
3	5	3	6
4	Em disco	4	Em disco
5	Em disco	5	13
6	4	6	8
7	3	7	12

Figura 9: Tabela do exercício

Com 10b, cada moldura/página conterá 1024B (1KB)

- Busca do segmento 1, página 1, deslocamento 3  
 $(14,3) \rightarrow 14 * 1024 + 3 = 14339$
- Armazenamento no segmento 0, página 0, deslocamento 16  
 Protection fault: não se pode escrever lá
- Busca do segmento 1, página 4, deslocamento 28  
 Page fault: o endereço buscado está em disco
- Salto para localização no segmento 1, página 3, deslocamento 326  
 Protection fault: salto é usado para desvio de fluxo em programas -> irá desviar para uma instrução que será executada. A moldura, contudo, não tem permissão para execução