

Lista de Exercícios — Gerenciamento de Memória — Respostas

1. 2^{20} bytes = 1024 KB = 1 MB
2. Podem ser usados dois raciocínios equivalentes:
 - Raciocínio 1: 2 GB = 2×2^{30} bytes = 2^{31} bytes
 - Raciocínio 2: 2 GB = 2.147.483.648 bytes
 $\log_2 2.147.483.648 = 31$

Portanto, são necessários endereços físicos de 31 bits.

3. Se não há MMU, o espaço de endereçamento físico e o espaço de endereçamento lógico são idênticos. Portanto, o maior tamanho de programa, que é dado pelo espaço de endereçamento lógico, é de 2^{18} bytes = 262.144 bytes = 256 KB.
4. As lacunas ocupadas pelas requisições serão as seguintes:
 - (a) 10 MB (sobram 5 MB), 20 MB (sobram 10 MB), 10 MB (sobram 4 MB)
 - (b) 7 MB (sobram 2 MB), 10 MB, 9 MB (sobram 3 MB)
 - (c) 20 MB (sobram 15 MB), 18 MB (sobram 8 MB), 15 MB (sobram 9 MB)
 - (d) 10 MB (sobram 5 MB), 20 MB (sobram 10 MB), 10 MB (sobram 4 MB)
5. As lacunas ocupadas pelas requisições serão as seguintes:
 - (a) 20 KB (sobram 5 KB), 10 KB (sobram 6 KB), 18 KB (sobram 10 KB)
 - (b) 18 KB (sobram 3 KB), 4 KB, 9 KB (sobra 1 KB)
 - (c) 20 KB (sobram 5 KB), 18 KB (sobram 14 KB), 14 KB (sobram 6 KB)
 - (d) 20 KB (sobram 5 KB), 5 KB (sobra 1 KB), 18 KB (sobram 10 KB)
6.
 - *First-fit*: 500 KB (sobram 288 KB), 600 KB (sobram 183 KB), 288 KB (sobram 176 KB), não cabe
 - *Best-fit*: 300 KB (sobram 88 KB), 500 KB (sobram 83 KB), 200 KB (sobram 88 KB), 600 KB (sobram 174 KB)
 - *Worst-fit*: 600 KB (sobram 388 KB), 500 KB (sobram 83 KB), 388 KB (sobram 276 KB), não cabe

Best-fit é o mais eficiente neste caso, pois é o único que consegue alocar todos os processos.

7.
 - (a) Memória livre = $256 + 512 + 256 = 1.024$ KB = 1 MB
 - (b) O maior processo é limitado à maior lacuna disponível, que é de 512 KB.
 - (c) Com 4.096 KB / 256 KB = 16 unidades de alocação, é necessário um mapa com 16 bits, ou 2 bytes, que ficaria assim:

byte 0	byte 1
11110111	11001110

- (d) Os nós da lista têm 3 campos:

P/L	início	tamanho
-----	--------	---------

P/L indica se é processo ou lacuna. *início* e *tamanho* são expressos em unidades de alocação. Assim, uma região de memória (ou **partição**) com 2 MB alocada a um processo a partir do endereço 524.288 (= 512 KB) seria representada pelo nó

P	2	8
---	---	---

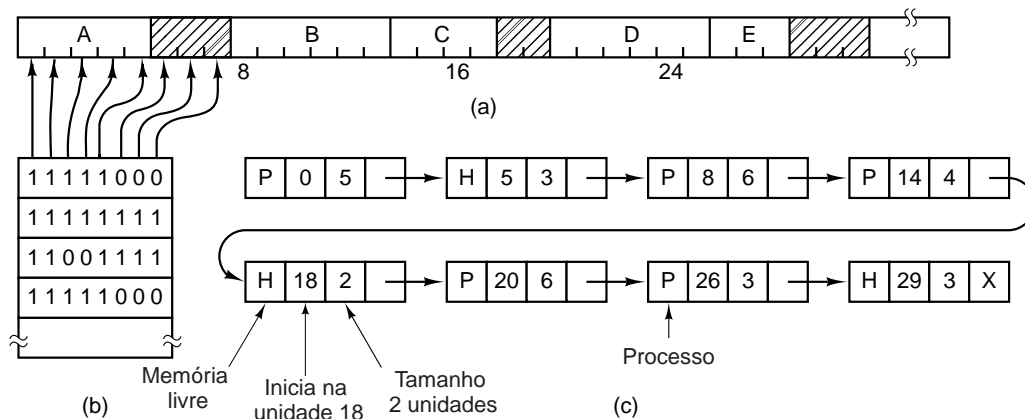
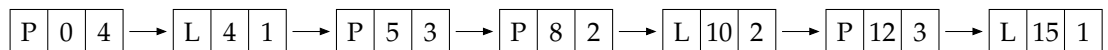
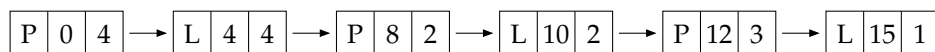


Figura 1: Gerenciamento de espaço livre com mapa de bits e lista encadeada

Assim, para a situação inicial do exercício, a lista seria:



- (e) Com o encerramento do processo A, sua partição deve ser fundida com a lacuna precedente, e a lista fica:



O processo C está alocado entre duas lacunas; quando ele encerra, sua partição deve ser fundida com essas lacunas, e a lista fica:



8. (a) O número de parágrafos na memória física é $2 \text{ GB} / 16 \text{ B} = 2^{31} / 2^4 = 2^{27}$. Portanto, é necessário um mapa com 2^{27} bits, o que corresponde a $2^{27} / 2^3 = 2^{24}$ bytes, ou 16 MB.
- (b) Com endereços de 32 bits, cada nó da lista possui pelo menos 12 bytes (cf. fig. 1): são 31 bits para o endereço inicial, 31 bits para o tamanho, 1 bit para indicar se é uma partição ou lacuna, e 31 bits para o ponteiro para o próximo nó, o que dá 94 bits. Se o tamanho de cada lacuna/partição for $256 \text{ KB} = 2^{18}$ bytes, teremos $2^{31} / 2^{18} = 2^{13}$ lacunas/partições, que correspondem a uma lista com 2^{13} nós. Essa lista irá ocupar $2^{13} \times 12 = 98.304 \text{ bytes} = 96 \text{ KB}$.
- (c) Seguindo o raciocínio do item (b), se o tamanho de cada lacuna/partição for $4 \text{ KB} = 2^{12}$ bytes, teremos $2^{31} / 2^{12} = 2^{19}$ lacunas/partições, que correspondem a uma lista com 2^{19} nós. Essa lista irá ocupar $2^{19} \times 12 = 6.291.456 \text{ bytes} = 6 \text{ MB}$.

9. (a) $EEL = 2^{16} \text{ bytes} = 65.536 \text{ bytes} = 64 \text{ KB}$
 (b) $EEF = 2^{20} \text{ bytes} = 1.048.576 \text{ bytes} = 1 \text{ MB}$
 (c) 10 bits (número da página física)
 (d) Dada uma página de 1 KB = 2^{10} bytes , a tabela de páginas possui $2^{16}/2^{10} = 64$ entradas.
10. (a) memória física = $2^{24} \text{ bytes} = 16 \text{ MB}$
 (b) maior programa = $2^{20} \text{ bytes} = 1 \text{ MB}$
 (c) Dada uma página de 1 KB = 2^{10} bytes , a tabela de páginas possui $2^{20}/2^{10} = 1.024$ entradas.
 (d) Cada entrada na tabela de páginas ocupa $14 + 2 = 16 \text{ bits}$. Portanto, a tabela de páginas ocupa $1.024 \times 16 = 16.384 \text{ bits}$.
11. (a) Para representar 16 segmentos, são necessários $\log_2 16 = 4 \text{ bits}$ para o número de segmento.
 (b) Cada segmento possui até $64 \text{ KB}/512 \text{ B} = 128$ páginas. Portanto, são necessários $\log_2 128 = 7 \text{ bits}$ para o número de página lógica.
 (c) Como cada página tem 512 bytes, são necessários $\log_2 512 = 9 \text{ bits}$ de deslocamento.
 (d) O endereço lógico é formado por número de segmento, número de página lógica e deslocamento, o que dá um total de $4 + 7 + 9 = 20 \text{ bits}$.
12. (a) Como são necessários dois acessos à memória (uma à tabela de páginas e outro à página física), o tempo de acesso é de $200 + 200 = 400 \text{ ns}$.
 (b) $h = 0,75$
 $t_{ac} = h \cdot t_{mem} + (1 - h) \cdot 2t_{mem} = 0,75 \cdot 200 + 0,25 \cdot 400 = 250 \text{ ns}$
13. A expressão usada para obter o endereço físico é $EL = \text{BASE} + EF$. Representando os endereços solicitados por letras de (a) a (e), tem-se:
 (a) $EF = 219 + 430 = 649$
 (b) $EF = 2300 + 10 = 2310$
 (c) erro ($EL > \text{limite}$)
 (d) $EF = 1327 + 400 = 1727$
 (e) erro ($EL > \text{limite}$)
14. O número de página é dado por $p = \left\lfloor \frac{\text{endereço}}{\text{tamanho página}} \right\rfloor$.
 O deslocamento é dado por $d = \text{endereço} - (p \times \text{tamanho página})$.
 Para páginas de 4 KB, os pares (página, deslocamento) são (4, 3616), (8, 0) e (14, 2656). Para páginas de 8 KB, os pares são (2, 3616), (4, 0) e (7, 2656).
15. A tabela de páginas contém $2^{32}/2^{13} = 524.488$ entradas. O carregamento da tabela de páginas leva 52 ms. Se um processo recebe 100 ms, isso corresponde a 52 ms para carregar a tabela de páginas e 48 ms para executar. Logo, 52% do tempo do processo é gasto carregando a tabela de páginas.
16. 20 bits são usados para o número de página virtual, e restam 12 para o deslocamento, o que dá páginas de $2^{12} = 4 \text{ KB}$. Com 20 bits para o número de página existem 2^{20} páginas.
17. Para paginação em um nível, são necessárias $2^{20} = 1 \text{ M}$ entradas. Para paginação em dois níveis, a tabela de páginas principal possui $2^{10} = 1 \text{ K}$ entradas que apontam, cada uma, para uma tabela de páginas de segundo nível. Apenas duas dessas tabelas são usadas (uma para código+dados e outra para pilha), cada uma tendo também $2^{10} = 1 \text{ K}$ entradas. Portanto, no total são necessárias $3 \times 2^{10} = 3072$ entradas.

18. A sobrecarga, que é o tempo necessário para obter o endereço físico correspondente ao endereço virtual fornecido, é dada por $t_{sc} = h \cdot t_{tlb} + (1 - h)t_{mem}$. Usando $t_{tlb} = 1$ ns e $t_{mem} = 5$ ns e fazendo $t_{sc} = 2$ ns, tem-se:

$$2 = h + (1 - h)5$$

$$2 = h + 5 - 5h$$

$$h = 0,75$$

Ou seja, a taxa de acerto necessária é de 75%.

19. FIFO MRU
 (a) 10 FP 10 FP
 (b) 9 FP 8 FP
 (c) 8 FP 7 FP
20. (a) Os números de página correspondentes aos endereços lógicos gerados são
 0, 1, 0, 2, 3, 4, 0, 0, 5, 8
 Sendo assim, tem-se 8 FP tanto para FIFO quanto para MRU.
- (b) Os números de página correspondentes aos endereços lógicos gerados são
 0, 0, 0, 1, 1, 2, 0, 0, 2, 4
 Sendo assim, tem-se 4 FP tanto para FIFO quanto para MRU.
21. (a) 9 faltas de página
 (b) 10 faltas de página
22. (a) 1 (b) 4 (c) 3
23. O sistema deve ter 128 MB, de modo que devem ser adicionados 96 MB.
24. Endereços físicos: 33268, falta de página, 65844, falta de página, 45172
25. Os endereços 3792, 1854 e 4300 têm as entradas de tabela de páginas correspondentes (páginas 3, 1, 4) na TLB, enquanto os endereços 5200 e 6800 (páginas 5, 6) não. Portanto, o tempo de acesso estimado será de $3 t_{hit} + 2 t_{miss} = 3 \cdot 96 + 2 \cdot 186 = 660$ ns.

26. (a)

Processo 1	
	página física
0	4
1	8
2	11
3	15
4	3
5	6
6	
7	

Processo 2	
	página física
0	14
1	1
2	7
3	5
4	
5	
6	
7	

Os números à esquerda representam o número de página lógica, e a coluna “válida” indica o bit de válido de cada entrada.

(b) O deslocamento de cada endereço lógico é dado por

$$d = EL \bmod t_{pag},$$

onde t_{pag} é o tamanho da página. O endereço físico correspondente é dado por

$$EF = f \cdot t_{pag} + d,$$

onde f é o número da página física (*frame*) encontrado na tabela de páginas.

Portanto, os endereços físicos são:

P1: 4608, 5119, 8192, 9168, 15388, 16288, 6224, 7024

P2: 14848, 15359, 1024, 2000, 5148, 6048

(c) Os tempos de acesso são de 51 ns (em caso de acerto na TLB) e 101 ns (em caso de erro). Tanto para P1 como para P2 são feitos pares de acessos consecutivos a uma mesma página, o que dá um erro e um acerto na TLB a cada dois acessos. Portanto, o tempo médio de acesso, em ambos os casos, é de $(101 + 51)/2 = 76$ ns.