

SISTEMAS OPERACIONAIS I  
LISTA DE EXERCÍCIOS III  
PROF. FRIEDRICH

1. Explique a diferença entre endereço lógico e endereço físico.
2. Explique os seguintes algoritmos de alocação de memória:
  - a) First-fit
  - b) Best-fit
  - c) Worst-fit
3. Considere as seguintes partições livres: 10K, 20K, 4K, 7K, 9K, 12K, 15K e 18K. Como os algoritmos First-fit, Best-fit, Worst-fit e Next-fit alocariam partições para as seguintes requisições:
  - a) 12K
  - b) 10K
  - c) 9K
4. Explique a diferença entre fragmentação externa e interna.
5. Considere um computador usando *Buddy System* para o gerenciamento de memória. Inicialmente ele tem um bloco de 256K no endereço 0. Depois das requisições 5K, 25K, 35K e 20K acontecerem, quantos blocos existem e qual o tamanho e endereço?
6. Porque o tamanho das páginas é sempre potência de 2?
7. Considere um espaço de endereçamento lógico de 8 páginas de 1K cada, mapeados em uma memória física de 32 frames.
  - a) Quantos bits tem o endereço lógico?
  - b) Quantos bits tem o endereço físico?
8. Considere um sistema com paginação onde a tabela de páginas está na memória:
  - a) Quanto tempo leva uma referência a memória paginada se uma referência a memória leva 150ns?
  - b) Qual o tempo de uma referência a memória se tivermos uma memória associativa que tem sucesso em 75% das vezes? (o tempo da memória associativa é 15% do tempo normal)
9. Considere um computador com 32-bits de endereço usando tabela de páginas em 2-níveis. Os endereços virtuais são compostos por um campo de 9-bits para o nível 1, 11-bits para o nível 2 e um deslocamento. Qual o tamanho das páginas e quantas páginas podem existir no endereço lógico?

10. Uma máquina tem um endereço lógico de 48-bits, endereço físico de 32-bits e páginas de 8K. Quantas entradas são necessárias para uma tabela de páginas convencional? E para uma tabela invertida?

11. Porque segmentação e paginação são combinadas em um novo esquema?

12. Quando usamos segmentação e paginação, primeiro consultamos o descritor de segmentos e depois o descritor de páginas. No caso da memória associativa, ela funciona da mesma forma? Com dois níveis de consulta?

13. Quando ocorre uma falta de página (*page fault*)? Descreva as ações do SO em uma *page fault*?

14. Considere uma sequência de referências a páginas para um processo com  $m$  frames (vazios inicialmente). A sequência tem tamanho  $p$  com  $n$  páginas distintas ocorrendo. Para um algoritmo de troca de páginas:

- a) o que é um limite baixo de *page fault*?
- b) o que é um limite alto de *page fault*?

15. Se usarmos um algoritmo de troca de páginas FIFO com 4 frames e 8 páginas, quantos *page faults* vão acontecer na seguinte sequência de referências: 0 1 7 2 3 2 7 1 0 3 se os 4 frames estão inicialmente vazios.

16. Repita o problema anterior usando LRU.

17. Considere o seguinte array:

**var A: array[1..100] of integer;**

onde  $A[1][1]$  está na posição 200 em um sistema com memória paginada com páginas de tamanho 200. O processo que manipula a matriz está na página 0 (0-199), desta forma as instruções são buscadas na página 0.

Considerando 3 frames, quantas *page fault* são geradas pelos seguintes códigos de inicialização do array, usando LRU, e sabendo que o frame 1 tem o código de inicialização e os outros dois frames estão vazios inicialmente.

a) **for j:= 1 to 100 do**  
    **for i:= 1 to 100 do**  
         $A[i][j]:=0$ ;

b) **for i:= 1 to 100 do**  
    **for j:= 1 to 100 do**  
         $A[i][j]:=0$ ;

18. Considere a utilização de uma política de troca de páginas que regularmente examina cada página e descarta aquela que não foi usada desde a última vez que foi feita o exame. O que voce ganha e o que voce perde usando esta política ao invés de LRU ou segunda chance?

19. Considere um sistema de computação com paginação por demanda onde o grau de multiprogramação é fixado em 4. O sistema foi recentemente medido para determinar a utilização da CPU e do disco de paginação. Os resultados podem estar entre os listados abaixo. Para cada caso, o que está acontecendo? O grau de multiprogramação pode ser aumentado para aumentar a utilização da CPU?

- a) utilização da CPU 13%; utilização do disco 97%
- b) utilização da CPU 87%; utilização do disco 3%
- c) utilização da CPU 13%; utilização do disco 3%

20. Qual é a causa do *thrashing* ? Como o sistema detecta o *thrashing* ? Uma vez detectado o que o sistema pode fazer para eliminar o problema?

21. Um processo tem 4 *frames* alocados. O quadro a seguir mostra as informações referentes aos mesmos.

No. Pag. Virt.	Frame	Tempo carga	Tempo referenc.	Bit R	Bit M
2	0	60	159	1	1
1	1	130	160	0	1
0	2	26	162	1	0
3	3	30	163	0	0

Um *page fault* para a página virtual 4 ocorre. Qual *frame* será trocado para os seguintes algoritmos de troca:

- a) FIFO (first-in-first-out) \_\_\_\_\_
- b) Clock (segunda chance) \_\_\_\_\_
- c) LRU (least recently used) \_\_\_\_\_
- d) Algoritmo Ótimo \_\_\_\_\_

22. Um SO observa o conjunto de páginas lógicas em uso pelos processos para decidir se pode ou não iniciar a execução de um novo processo. Esse sistema define, como conjunto de trabalho, o conjunto das páginas acessadas no último intervalo completo do *timer*. O sistema dispõe de 20 páginas físicas. A história do sistema no último intervalo completo do *timer* é mostrada abaixo. Quantas páginas físicas o sistema poderá fornecer a um novo processo, mantendo ainda o conjunto de trabalho dos processos atuais?

P0 (4,5,3,6,3,4,5,3,3,6 )

P1( 1,7,8,8,1,3,1,7,8,9 )

P2( 6,7,8,5,8,6,6,7,5,5 )

23. Qual problema pode ocorrer quando um periférico pode acessar diretamente a memória e utiliza-se memória virtual? Como resolve-lo?