



**Lista de Exercícios 5 –
Gerenciamento de Memória**

1) Os primeiros computadores, e consequentemente os primeiros SO's, não possuíam uma unidade para gerenciar a memória principal de um computador. Por que nesse tipo de ambiente não era possível haver mais de um programa ao mesmo tempo na memória se não houvesse algum pré-processamento sobre o código do processo?

2) Explique a diferença entre endereçamento lógico e endereçamento físico.

3) Inicialmente alocava-se uma quantidade de endereços físicos contínuos na memória para um determinado processo. Isto era feito através de dois valores armazenados em registradores lógicos denominados registrador-base e registrador-limite. Responda:

a) O que é armazenado em cada um destes registradores?

b) Como é calculado o endereço físico a partir de um determinado endereço lógico usando esta metodologia?

c) Qual o principal problema da alocação contínua de memória? Explique-o.

4) O gerenciamento de memória através de paginação cria uma tabela que correlaciona cada página da memória lógico (virtual) a um quadro da memória física. Explique verbalmente como um endereço lógico é mapeado para um endereço físico utilizando esta abordagem de gerenciamento de memória.

5) Por que normalmente a quantidade de endereços de uma página ou quadro é uma potência de 2?

6) Que informações são armazenadas para cada página na tabela de páginas comuns a maioria dos Sistemas Operacionais atuais?

7) A paginação evita o problema da fragmentação externa, porém apresenta o problema da fragmentação interna. Explique o que é este problema e onde ele ocorre.

8) Considere um sistema de alocação contínua de memória. O gerenciador tem ciência dos seguintes espaços livres de memória, em KB, nesta sequência: 10, 4, 20, 18, 7, 9, 12 e 15. Cada linha da tabela abaixo representa um processo novo com a quantidade que necessita de memória para ser alocada. Cada coluna da tabela indica o algoritmo de escolha de espaço livre de memória. Preencha a tabela indicando que espaço livre é escolhido para cada processo segundo cada um dos algoritmos

	<u>First-Fit</u>	<u>Best-Fit</u>	<u>Worst-Fit</u>
A (12KB)			
B (10KB)			
C (9 KB)			

9) Em um sistema de alocação contínua, o gerenciamento de memória livre pode ser feito através de uma lista encadeada (Abaixo ilustrada usando uma tabela). Cada elemento da lista indica o id do processo ou um id especial (L) indicando que o espaço é livre, além dos registrador-base e registrador-limite do processo.

<u>Id</u>	<u>Reg. Base</u>	<u>Reg. Limite</u>
P1	0	20
L	20	10
P2	30	15
L	45	16
P3	61	12
L	73	20
P4	93	9
L	102	18
P5	120	32
L	152	48

Mostre o estado da lista após cada uma das ações descritas a seguir, supondo que o algoritmo de alocação é o First-Fit:

I) O processo P6 é iniciado e precisa de 12KB

II) O processo P3 é finalizado

III) O processo P2 é finalizado

IV) O processo P7 é iniciado e precisa de 25KB

10) Repita a questão 9 utilizando como algoritmo de alocação o Best-Fit.

11) Repita a questão 9 utilizando como algoritmo de alocação o Worst-Fit.

12) Para mover um processo dentro do espaço de endereçamento físico, primeiro ele deve ser copiado para uma área grande o suficiente que o caiba e depois sua localização anterior é apagada. O processo de compactação em alocação contínua, consiste em mover os processo para uma das bordas da memória física a fim de unir todos os espaços de endereçamento dos processos e assim unir todas as áreas livres em uma (*processo de desfragmentação*). Usando a lista da questão 9 (antes das alterações), realize o processo de compactação, afirmando cada processo que está sendo movido e mostrando como a lista vai sendo alterada no decorrer da compactação.

13) Usando a lista da questão 9 (antes das alterações) calcule os endereços físicos pedidos abaixo:

- a) Processo P1 pede o endereço 6KB.
- b) Processo P4 pede o endereço 7KB.
- c) Processo P3 pede o endereço 13KB.
- d) Processo P2 pede o endereço 3KB.

Obs: Caso o endereço lógico seja inválido, declare o erro de acesso irregular de memória

14) Considere um espaço de memória lógico com 256 páginas e cada página com 4K endereços. Esse espaço é mapeado para uma memória física com 64 quadros.

- a) Quantos bits são necessários para o endereço lógico?
- b) Quantos bits são necessários para o endereço físico?

15) Suponha que um programa é executado em duas máquinas diferentes. Na primeira, o quadro de memória possui 4K endereços. Na segunda, o quadro possui 8K endereços. Supondo que em ambos os computadores o processo é paginado, determine o número da página dos seguintes endereços lógicos em cada uma das máquinas:

- a) 20000
- b) 32768
- c) 60000

16) Assuma que cada página possua 1K endereços. Quais os números das páginas e do deslocamento dos seguintes endereços virtuais?

- a) 3085
- b) 42095
- c) 215201
- d) 650000
- e) 2000001

17) Considere um processo com a seguinte tabela de páginas, onde cada página possui 2k endereços.

<u>Página</u>	<u>Quadro</u>
0	7
1	8
2	5
3	10

Calcule:

- a) O endereço físico do endereço lógico 256.
- b) O endereço físico do endereço lógico 4256.
- c) O endereço lógico do endereço físico 20994.

18) 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 do relógio):

<u>Página</u>	<u>Carregado</u>	<u>Última ref.</u>	<u>R</u>	<u>M</u>
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- a) Qual página será trocada pelo FIFO?
- b) Qual página será trocada pelo NRU?
- c) Qual página será trocada pelo LRU?
- d) Qual página será trocada pelo SC?

19) Se o algoritmo de substituição FIFO é usado com quatro quadros e oito páginas virtuais, quantas faltas de página (**page faults**) ocorrerão com a cadeia de referências 0-1-7-2-3-2-7-1-0-3 se os quatro quadros estão inicialmente vazios?

20) Repita a questão anterior para LRU usando a implementação:

- a) Pilha dos últimos referenciados
- b) Mapa de bits

21) Um computador tem quatro quadros de memória. No primeiro tique do relógio, os bits R são 0111 (página 0 é 0, as demais são 1). Nos tiques subsequentes os valores são 1011, 1010, 1101, 0010, 1010, 1100 e 0001. Se o algoritmo de envelhecimento (*aging*) é usado com um contador de 8 bits, quais valores dos quatro contadores após o último tique? Se uma página precisasse ser removida após o último tique, qual seria?

22) Alguns processadores possuem uma memória associativa de alta velocidade denominada TLB. Qual a sua finalidade? E por que ela melhora o desempenho de acesso a memória?

23) Considere um sistema de paginação com a tabela de páginas armazenada na memória.

a) Se uma referência a memória leva 50 nanossegundos, quanto tempo leva para um processo fazer uma referência a um endereço lógico?

b) Se adicionarmos uma TLB de forma que 75% de todas as referências a páginas são encontradas nela, qual a média de tempo gasta para um processo fazer uma referência a um endereço lógico?

24) A tabela de páginas de um processo é armazenada na memória principal. O que acontece se essa tabela for maior do que um quadro da memória, supondo que esta utiliza paginação?

25) Explique o processo de paginação por demanda.

26) O que é segmentação? Quais são as vantagens de se utilizar segmentação?

27) Por que segmentação e paginação são normalmente combinadas em um único esquema?

28) Considere a seguinte tabela de paginação:

<u>Segmento</u>	<u>Reg. Base</u>	<u>Reg. Limite</u>
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Quais os endereços físicos para os seguintes endereços lógicos?

- a) 0,430
- b) 1,10
- c) 2,500
- d) 3,400
- e) 4,112

29) Consideremos um programa que tenha dois segmentos mostrados a seguir, consistindo de instruções no segmento 0 e de dados de leitura/escrita no segmento 1. O segmento 0 tem proteção leitura/execução e o segmento 1 tem proteção 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 são as seguintes (todos os números na tabela são decimais):

Segmento 0		Segmento 1	
Leitura/Execução		Leitura/Escrita	
Nº Página	Nº Quadro	Nº Página	Nº Quadro
0	2	0	Disco
1	Disco	1	14
2	11	2	9
3	5	3	6
4	Disco	4	Disco
5	Disco	5	13
6	4	6	8
7	3	7	12

Para cada um dos seguintes casos, diga o endereço físico ou identifique o tipo de erro que ocorre (**page fault** = falta de página ou **protection fault** = falta de proteção).

	Ação	Segmento	Página	Deslocamento
a)	Buscar	1	1	3
b)	Atualizar	0	0	16
c)	Buscar	1	4	28
d)	Atualizar	1	3	32

30) Um sistema possui 4 quadros de memória física, inicialmente não alocados e com SO implementando paginação por demanda. Nas tabelas abaixo, a primeira linha indica a sequência de páginas que os processos estão requisitando à memória. As 8 primeiras colunas mostram como as páginas vão sendo alocadas aos quadros e referenciadas. Entretanto, a partir na 9 coluna não há quadros suficientes para a página **A**. Continue o preenchimento das tabelas segundo os algoritmos mencionados, indicando na última linha se aconteceu um **page fault** naquela rodada.

FIFO (Primeiro a entrar, primeiro a sair)

Quadro	E	D	E	H	B	D	E	D	A	E	B	E	D	E	B	G
Q1	E	E	E	E	E	E	E	E								
Q2		D	D	D	D	D	D	D								
Q3				H	H	H	H	H								
Q4					B	B	B	B								
Fault	X	X		X	X											

LRU (Menos usada recentemente)

Quadro	E	D	E	H	B	D	E	D	A	E	B	E	D	E	B	G
Q1	E	E	E	E	E	E	E	E								
Q2		D	D	D	D	D	D	D								
Q3				H	H	H	H	H								
Q4					B	B	B	B								
Fault	X	X		X	X											

31) A tabela abaixo ilustra os valores dos bits R (referenciada) de cada página de memória a cada interrupção do relógio. Neste SO os algoritmos NFU (*Not Frequently Used* = Não Frequentemente Usado) e Aging (Envelhecimento) usam contadores com 8 bits. Após estas interrupções, acontece um **page fault** e uma das páginas precisa ser substituída. Que página será substituída segundo cada um dos algoritmos? Justifique sua resposta mostrando a ideia dos algoritmos.

Bit R	Interrupções								
	1	2	3	4	5	6	7	8	9
Pág. 0	1	0	1	0	0	1	1	0	0
Pág. 1	1	1	1	1	0	0	0	0	0
Pág. 2	0	1	0	0	0	1	0	0	1
Pág. 3	0	0	0	1	0	0	1	1	0

32) A alocação por segmentação paginada é uma abordagem de gerenciamento que reuni as vantagens da segmentação com a paginação e está presente nos principais sistemas operacionais modernos. Dessa forma, os bits de um endereço lógico de um sistema desses são divididos em três grupos: os bits do segmento, os bits da página e os bits do deslocamento. Suponha que neste sistema, em particular, cada endereço lógico contém 18 bits, sendo que os 2 primeiros bits representam o segmento, os 5 bits seguintes representam a página e os bits restantes representam o deslocamento. Já o endereço físico contém 16 bits, sendo os 5 primeiros bits correspondentes ao número do quadro e os restantes correspondentes ao deslocamento. Calcule o endereço físico do endereço lógico 0100000000000001101. (**Obs:** Anuncie caso o endereço não possa ser obtido por causa de uma **page fault**.)

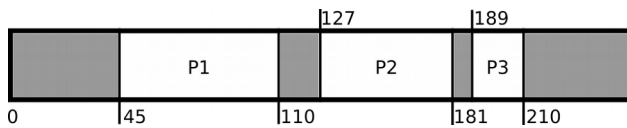
Segmento 0 Código		Segmento 1 Dados		Segmento 2 Pilha	
Nº Pág.	Nº Quad.	Nº Pág.	Nº Quad.	Nº Pág.	Nº Quad.
0	Disco	0	9	0	Disco
1	7	1	Disco	1	Disco
2	Disco	2	3	2	2
3	0	3	Disco	3	Disco
4	Disco	4	8	4	Disco

33) O MMU de um SO aloca os processos na memória continuamente e gerencia as áreas livres e ocupadas usando a lista encadeada a seguir, onde o identificador indica que a área é livre e os demais identificadores indicam o processo alocado naquela região da memória. Os demais números em cada célula da lista encadeada indicam o registrador-base e o registrador limite da região. Suponha que a MMU implementa o algoritmo Worst-Fit para alocar um processo.

$P_a \ 0 \ 5 \rightarrow L \ 5 \ 3 \rightarrow P_b \ 8 \ 6 \rightarrow P_c \ 14 \ 4 \rightarrow$
 $\rightarrow L \ 18 \ 2 \rightarrow P_d \ 20 \ 6 \rightarrow L \ 26 \ 6$

Desenhe a lista encadeada após a remoção do processo **P_b** e da inserção do processo **P_e**, que necessita de um espaço igual a 5, nesta ordem.

34) O MMU de um SO aloca os processos na memória continuamente. Suponha que o sistema possua apenas 256 endereços de memória. Há 3 processos em execução neste sistema alocados segundo a figura abaixo. (**Obs:** Os números de endereços, indicam o primeiro endereço daquela área correspondente)



Utilizando o algoritmo *Best-Fit*, adicione um processo P4 que precisa de 16 endereços, mostrando seu registrador base e registrador limite e re-desenhando o quadro acima com o processo P4 alocado.

35) O endereço lógico de um sistema tem 16 bits de tamanho. Cada página possui 4K endereços. Considere a seguinte tabela de páginas do processo em execução exibida ao lado. Calcule o endereço físico que a MMU traduzirá para uma referência à memória feita pelo processo corrente ao endereço 8292. **Obs:** Anuncie caso o endereço não seja possível de ser obtido por causa de uma *page fault*.

Página	Quadro
0	2
1	1
2	6
3	0
4	4
5	3
6	-
7	-
8	-
9	5
10	-
11	7