Lista de Exercícios 5 – Gerenciamento de Memória

Respostas

8)

	First-Fit	Best-Fit	Worst-Fit
A (12KB)	20	12	20
B (10KB)	10	10	20
C (9 KB)	10	9	20

9) A alocação é contínua e o algoritmo é o First-Fit, ou seja, quando um processo é iniciado e precisa de memória, o primeiro espaço onde ele couber é escolhido e é alocado um espaço para ele.

I				II				III				IV		
Id	RB	RL		Id	RB	RL		Id	RB	RL		Id	RB	RL
P1	0	20		P1	0	20		P1	0	20		P1	0	20
L	20	10		L	20	10		L	20	25		P7	20	25
P2	30	15		P2	30	15		P6	45	12		P6	45	12
P6	45	12		P6	45	12		L	57	36		L	57	36
L	57	4		L	57	36		P4	93	9		P4	93	9
P3	61	12		P4	93	9		L	102	18		L	102	18
L	73	20		L	102	18		P5	120	32		P5	120	32
P4	93	9		P5	120	32		L	152	48		L	152	48
L	102	18		L	152	48								
P5	120	32												
L	152	48												

I) O processo P6 é iniciado e precisa de 12KB. Ele cabe no segundo espaço livre, logo depois de P2. Como havia 16K de espaço livre, ainda sobra 4K livre entre P6 e P3.

II) O processo P3 é finalizado. Sua memória deve ser liberada. Como ele está entre duas áreas livres, as duas áreas livres e a de P3 são mescladas em uma única área livre de 36K de espaço.

III) O processo P2 é finalizado. Como ele está entre uma área livre e o processo P6, sua área é mesclada com a área livre anterior.

IV) O processo P7 é inicado e precisa de 25K. Esse espaço é exatamente do mesmo tamanho da primeira área livre, entre P1 e P6, logo ela é alocada para P7 e nenhuma outra alteração é feita.

Dica: Uma vez que um processo é alocado, seu registrador base e registrador limite não mudam. Quando duas áreas se mesclam, o registrador base menor é mantido e os registradores limite das áreas mescladas são somadas.

10 e 11) A ideia é a mesma da questão 9, mas aplicando os algoritmos Best-Fit (quando um processo é iniciado, ele ocupa a área livre que mais se aproxima da área que ele precisa, desde que o caiba) e Worst-Fit (quando o processo é iniciado, ele ocupa a maior área livre, desde que o caiba).

12) A compactação pode ser feita copiando todos os processos para a outra ponta, deixando uma área livre grande no início. Note que o último endereço dessa memória é 200 (a última área livre tem reg. base 152 e reg. limite 48).

I) Copiando P5 para o final da memória, as áreas livres em torno dele se mesclam antes dele.

II) Copiando P4 para imediatamente antes de P5, mescla-se as áreas livres em torno dele.

III) Aplicando o mesmo raciocínio para P3.

IV) E para P2

V) E para P1.

I			II			III			IV			V		
Id	RB	RL	Id	RB	RL	Id	RB	RL	Id	RB	RL	Id	RB	RL
P1	0	20	P1	0	20	P1	0	20	P1	0	20	L	0	112
L	20	10	L	20	10	L	20	10	L	20	112	P1	112	20
P2	30	15	P2	30	15	P2	30	15	P2	132	15	P2	132	15
L	45	16	L	45	16	L	45	102	P3	147	12	P3	147	12
P3	61	12	P3	61	12	P3	147	12	P4	159	9	P4	159	9
L	73	20	L	73	86	P4	159	9	P5	168	32	P5	168	32
P4	93	9	P4	159	9	P5	168	32						
L	102	66	P5	168	32									
P5	168	32												

Dica: cuidado com os valores dos registradores. Se P5 precisa de 32K e a memória termina em 200, então ele deve começar em 168 para que ele caiba. Se P4 precisa de 9K e P5 começa em 168, ele deve começar em 159. E assim a mesma ideia é feita nos outros processos. Ao fim, todas as áreas livres foram mescladas em uma só.

13) Esse é o caso de calculo do endereço físico (EF) usando alocação contínua. Para cada um destes cálculos, primeiro se verifica se o endereço lógico (EL) é menor do que o registrador limite (RL). Se não for, acontece um erro. Se for menor então soma-se o endereço logico com o reigstrador base (RB), obtendo assim o endereço fisico.

a) (EL = 6, RB = 0, RL = 20) → EL < RL (ok) → EF = RB + EL = 0 + 6 = 6

b) (EL = 7, RB = 93. RL = 9) → EL < RL (ok) → EF = RB + EL = 93 + 7 = 100

c) (EL = 13, RB = 61, RL = 12) → EL < RL (erro)

d) (EL = 3, RB = 30, RL = 15) → EL < RL (ok) → EF = RB + EL = 30 + 3 = 33

14) Cada página (e cada quadro, já que eles devem ter o mesmo tamanho) possuem 4K endereços. Isto significa que são $4*2^{10}$ endereços por página, ou 2^{12}. Ou seja, precisa-se de 12 bits para o deslocamento.

a) O endereçamento lógico possui 256 páginas, ou seja, 2^8 páginas. Logo, precisa-se de 8 bits para indicar a página do endereço logico. Somados aos 12 bits de deslocamento, o endereço lógico deve possuir 20 bits.

b) O endereçamento físico possui 64 quadros, ou seja, 2^6 quadros. Logo, precisa-se de 6 bits para indicar o quadro do endereço físico. Somados aos 12 bits de deslocamento, o endereçamento físico deve possuir 18 bits.

15) Se em uma máquina, o quadro possui 4K endereços, então ele possui 2^{12} endereços por quadro (ou página), ou seja, 12 bits para deslocamento. Na segunda máquina, o quadro possui 8K endereços, ou seja, 2^{13} endereços, logo o deslocamento é de 13 bits.

a) Convertendo 20000 em binário obtemos 100111000100000.

Na primeira máquina (com 12 bits de deslocamento). Removendo os 12 últimos bits, obtem-se o número binário 100, que é igual a 4, em decimal. Este é o número da página deste endereço nesta máquina.

Na segunda máquina (com 13 bits de deslocamento). Removendo os 13 últimos bits, obtem-se o número binário 10, que é igual a 2, em decimal. Este é o número da página deste endereço nesta máquina.

b e c) Similar ao item anterior.

16) Se uma página tem 1K endereços, então ela tem 2^{10} endereços ou 1024 endereços. Para descobrir o número da página e do deslocamento, pode-se dividir o endereço pela quantidade de endereços por página. O quociente será o número da página e o resto será o deslocamente. Outra opção é converter o endereço para binário. Separar os 10 últimos bits para deslocamento e os demais bits para página. Converte-os novamente para decimal e descobrirá o número.

a) 3085 em binário é 110000001101. Página = $11_2 = 3_{10}$. Deslocamento = $0000001101_2 = 13_{10}$

Outra forma de fazer:

3085 dividido por 1024 é igual a 3 com resto igual a 13.

17) Se cada página possui 2k endereços, então são 2048 = 2^{11} endereços, 11 bits para deslocamento.

a) 256 dividido por 2048 é igual a 0 com resto igual a 256. Logo o número da página é 0. Essa página corresponde ao quadro 7. Logo o endereço físico é igual a 7*2048 + 256 = 14592

c) Essa questão tá pedindo o contrário, dado o end. fisico, calcule o end. logico. Entretanto, a ideia é a mesma:

20994 dividido por 2048 é igual a 10 com resto igual a 514. Logo o número do quadro é 10 e o deslocamento é igual a 514. O quadro 10 corresponde a página 3. Logo o endereço logico é igual a 3*2048 + 514 = 6658.

18)

a) O algoritmo FIFO escolhe a página carregada mais antiga para substituir, logo a página escolhida é a página 3.

b) O algoritmo NRU escolhe a página pelos bits R e M. As páginas com R = 0 e M = 0 estão na primeira prioridade para serem escolhidas. Só temos a página 2 nestas condições.

c) O algoritmo LRU escolhe a página menos usada recentemente, ou seja, com menor valor do relógio da última referência. Logo a página escolhida é 1.

d) O algoritmo SC (Segunda Chance) funciona como o FIFO, mas usa o algoritmo R para dar uma segunda chance à página. A página 3 é a carregada mais antiga, mas tem bit R igual a 1, logo, seu bit R torna-se zero, e vamos para a segunda mais antiga: a página 0. Esta também tem bit igual a 1. A próxima mais velha é a página 2. Esta tem bit R igual a zero, então é escolhida.

19) As quatro primeira páginas darão page fault porque a memória não tem ninguém inicialmente, então cada vez que o processo tentar acessar estas páginas, o SO terá que carregá-las do disco e colocar em algum quadro. Quando a página 3 for acessada pela primeira vez, dará um page fault porque ela não está na memória, mas alguma página precisará ser removida da memória. Como o algoritmo FIFO remove a página mais antiga, a página 0 é a escolhida. As três próximas referências acontecem sem problemas, porque estas páginas já estão na memória. Ao acessar a página 0 novamente, acontece mais um page fault. O algoritmo escolhe a página 1, no quadro 2, para ser substituida. E finalmente, o último acesso acontece sem problemas, porque a página 3 já está na memoria.

Pag	0	1	7	2	3	2	7	1	0	3
Q1	**0**	0	0	0	**3**	3	3	3	3	**3**
Q2		**1**	1	1	1	1	1	**1**	**0**	0
Q3			**7**	7	7	7	**7**	7	7	7
Q4				**2**	2	**2**	2	2	2	2

Número de page faults = 6

Página em vermelho significa que aconteceu page fault e a página teve de ser carregada do disco.

Página em azul significa que a página já estava na memória, então não aconteceu page fault.

20)

a) O algoritmo da pilha vai mantendo a página mais recentemente acessada no topo. As que não vão sendo acessadas vai caindo na pilha. Quando não houver mais espaço na memória, a base da pilha é removida da memória e a página recém acessada é carregada e posta no topo da pilha. Nos 5 primeiros acessos, acontece a mesma coisa que o algoritmo FIFO. Mas quando a página 2 é acessada novamente, ela é puxada para o topo e a página 3 cai uma posição. O mesmo acontece com as páginas 7 e 1 acessadas posteriormente. Quando a página 0 é acessada, ela não está na pilha, logo a página 3, que está na base é removida e a página 0 é colocada no topo. Em seguida, a página 3 é acessada, a página 2, que estava na base, é removida e a página 3 inserida no topo.

Pag	0	1	7	2	3	2	7	1	0	3
				2	**3**	**2**	**7**	**1**	**0**	**3**
			7	7	2	3	2	7	1	0

		1	1	1	7	7	3	2	7	1
	0	0	0	0	1	1	1	3	2	7

Número de page faults = 7 (Normalmente o algoritmo LRU é mais eficiente que o algoritmo FIFO, mas para este caso, especificamente, o resultado foi favorável ao FIFO.)

b) O mapa de bits tem nos slides seu funcionamento. Cada vez que uma página for acessada, sua linha recebe uns e sua coluna recebe zeros. Os 4 primeiros acessos acontecerá da mesma forma que os demais (*page faults* estão acontecendo nestes primeiros acessos porque não tem nenhuma página na memória ainda).

21) O algoritmo de envelhecimento usa um contador para cada página com 8 bits (neste caso), todos eles iguais a zero inicialmente. A cada tique, estes contadores são deslocados um bit para a direita e os bits mais a esquerda são preenchidos com os bits R das páginas. Os contadores inicialmente são:

Pag0 = 00000000

Pág1 = 00000000

Pág2 = 00000000

Pág3 = 00000000

No primeiro tique, os bits R são 0111. Estes bits são adicionados no início dos contadores na mesma sequência:

Pág0 = 00000000

Pág1 = 10000000

Pág2 = 10000000

Pág3 = 10000000

No segundo tique, os bits R são 1011. Os contadores têm seus bits deslocados para a direita e adiciona-se esta sequência de bits R no início dos contadores.

Pág0 = 10000000

Pág1 = 01000000

Pág2 = 11000000

Pag3 = 11000000

No terceiro tique, os bits R são 1010, logo:

Pag0 = 11000000

Pag1 = 00100000

Pag2 = 11100000

Pag3 = 01100000

Continue esse procedimento com cada uma das sequências de bits R. No final, veja qual das páginas possui o contador de menor valor. Esta será a página a ser removida.

23)

a) Se a tabela de páginas está na memória, então primeiro o SO precisa acessar a tabela de páginas para descobrir o endereço e depois mais um acesso a memória para acessar o endereço em si. Logo o tempo gasto para cada referência a memória é igual a 50+50 = 100ns

b) Se a TLB está presente e 75% dos endereços são traduzidos por ela, estas referências são solucionadas com um único acesso a memória, ou seja, em 50 ns. As demais 25% das referências continuarão precisando a acessar a tabela na memória para traduzir o endereço e depois outro acesso para o endereço requisitado. Fazendo a média ponderada obtemos:

Tempo de referência médio = 0,75*50 + 0,25*100 = 62,5 ns

28) Esta questão é quase a mesma coisa que a questão da alocação contínua. Primeiro obtem-se os valores dos registradores base e registrador limite através do número do segmento. Depois verifica-se se o endereço é menor do que o reg. limite, em caso afirmativo, o endereço é somado ao reg. base.

a) segmento 0 → RB = 219, RL = 600 → EL = 430 < RL (ok) → EF = RB + EL = 219 + 430 = 649

29) O segmento 0 pode ler ou executar seu conteúdo, mas não alterar. O segmento 1 pode ler ou alterar seu conteúdo. A questão também diz que são 10 bits para deslocamento, ou seja, cada página possui 2^{10} ou 1024 endereços.

a) Buscar é a mesma coisa que ler. O segmento 1 permite a leitura. Deseja-se acessar a página 1. Esta página está armazenada no quadro 14, segundo a tabela. O endereço físico pode ser calculado como 14*1024 + 3 = 14339.

b) Atualizar é a mesma coisa que escrever. O segmento 0 não permite escrita, logo acontece um erro de proteção.

c) O segmento 1 permite a leitura, mas a página 4 não está em nenhum quadro, está em disco. Isso caracteriza uma ***page fault*** (falta de página)

d) O segmento 1 permite a escrita e a página 3 está no quadro 6. Logo o endereço físico é igual a: 6*1024 + 32 = 6176