



Lista de Exercícios - Sistemas Operacionais

Aula 9: Gerenciamento de Memória - Parte 2

Professores: Felipe M. G. França e Valmir C. Barbosa

Assistente: Alexandre H. L. Porto

1. A seguir é apresentada a listagem de um programa curto em linguagem *assembly* para um computador com páginas de 512 bytes. O programa está localizado no endereço 1020 e seu ponteiro de pilha está em 8192 (a pilha cresce em direção a 0). Forneça a cadeia de referências de páginas gerada por esse programa. Cada instrução ocupa 4 bytes (1 palavra) e referências tanto de instruções como de dados contam na cadeia de referências.

```
1020 Carregue a palavra 6144 no registrador 0
1024 Coloque o registrador 0 na pilha
1028 Chame um procedimento em 5120, colocando o endereço de
      retorno na pilha
1032 Subtraia a constante imediata 16 do ponteiro da pilha
1036 Compare o parâmetro real com a constante imediata 4
1040 Salte se igual a 5152
```

Resp.: Como cada página possui 512 bytes, então podemos inicialmente concluir o seguinte: a primeira instrução do programa, do endereço 1020, está na posição 508 da Página 1; o topo da pilha está no início da Página 16 (na posição 0), pois o ponteiro da pilha está apontando para o endereço 8192. Vamos agora ver quais páginas são acessadas ao executarmos cada uma das instruções:

- (a) **Carregue a palavra 6144 no registrador 0:** O processador deve ler o endereço 1020 para obter o código da instrução, acessando para isso a posição 508 da Página 1. Depois de obtê-lo, o

processador deverá agora acessar o endereço 6144, localizado no início da Página 12 (na posição 0), para obter o valor a ser armazenado no registrador 0. Note que o endereço com a próxima instrução será o 1024, localizado no início da Página 2.

- (b) **Coloque o registrador 0 na pilha:** O processador primeiramente lê o endereço 1024 no início da Página 2, para obter o código da instrução. Agora, o ponteiro da pilha é decrementado em pelo menos uma unidade, para armazenarmos o valor do registrador 0 na pilha. Independentemente do valor deste decremento, iremos acessar a Página 15, pois o ponteiro da pilha estava originalmente no início da Página 16. Note que agora o endereço da próxima instrução será o 1028, localizado na posição 4 da Página 2.
- (c) **Chame um procedimento em 5120, colocando o endereço de retorno na pilha:** O processador acessará novamente a Página 2 (na posição 4), ao obter o código da instrução armazenado no endereço 1028. Depois disso, o ponteiro da pilha será novamente decrementado, e vamos novamente acessar a Página 15 ao armazenarmos o endereço de retorno da chamada. O endereço da próxima instrução será agora o 5120, localizado no início da Página 10.
- (d) **Subtraia a constante imediata 16 do ponteiro da pilha:** O processador primeiramente lerá o endereço 5120, no início da Página 10, para obter o código da instrução. Depois subtrairá o valor 16 do ponteiro da pilha, e como este é um registrador, nenhuma referência será feita a uma página. Agora, o endereço da próxima instrução será o 5124, localizado na posição 4 da Página 10.
- (e) **Compare o parâmetro real com a constante imediata 4:** Primeiramente é acessada a posição 4 da Página 10, para obtermos o código da instrução armazenado no endereço 5124. Depois disso, a Página 15 será acessada, para lermos o parâmetro real do topo da pilha. A próxima instrução estará agora no endereço 5128 (localizado na posição 8 da Página 10).
- (f) **Salte se igual a 5152:** O processador acessa, para obter o código da instrução, a posição 8 da Página 10 (o endereço 5128). Note que a próxima instrução a ser executada ainda estará na Página 10, pois o endereço 5134 está na posição 12 desta página, e o endereço 5152 está na posição 32 da página.

Logo, temos a seguinte cadeia de referências, sendo que colocamos, en-

tre parênteses, se a página foi acessada para obter o código da instrução (I) ou um dado (D): 1 (I), 12 (D); 2 (I), 15 (D); 2 (I), 15 (D); 10 (I); 10 (I), 15 (D); 10 (I). Nesta cadeia, separamos as referências de duas instruções consecutivas por um ponto e vírgula, e as referências em uma mesma instrução por uma vírgula.

2. Um computador cujos processos têm 1024 páginas em seus espaços de endereços mantém suas tabelas de páginas na memória. O *overhead* exigido para ler uma palavra da tabela de páginas é 500ns. Para reduzir esse *overhead*, o computador tem uma TLB, a qual armazena 32 pares ⟨página virtual, moldura de página física⟩ e pode fazer uma busca em 100ns. Qual é a taxa de acertos necessária para reduzir o *overhead* médio para 200ns?

Resp.: Quando não usamos a TLB, o tempo médio para ler a palavra será sempre de 500 ns. Agora, se usarmos a TLB, o tempo médio será reduzido para 100ns se a palavra estiver na TLB. Note que o tempo médio para ler a palavra ainda será de 500ns caso esta palavra não esteja na TLB, pois a procura da palavra na TLB é feita em paralelo com a leitura dela da tabela de páginas. Note também que se a palavra estiver na TLB, o tempo será de 100ns, pois podemos abortar a leitura da tabela na memória. Logo, se a taxa de acerto (isto é, a fração das palavras da tabela de páginas referenciadas que estavam na TLB) for de h , o tempo médio de acesso será de $100h + 500(1 - h)$. Agora, como queremos um tempo médio de 200ns, bastará resolvermos a equação anterior para este valor, ou seja, resolver a equação $100h + 500(1 - h) = 200$. Ao resolvermos esta equação ($100h + 500 - 500h = 200 \Rightarrow 400h = 300 \Rightarrow h = 3/4$), vemos que a taxa de acerto deverá ser de 0,75, ou seja, uma taxa de acerto de 75%.

3. Suponha que o sistema operacional aloque três molduras de página para um processo, inicialmente vazias. Suponha ainda que o processo acesse, nesta ordem, as páginas 1, 13, 5, 0, 7, 7, 5, 9, 0 e 1. Para cada um dos algoritmos dados a seguir, diga quantas falhas de página ocorrerão, e qual página será substituída, se o processo posteriormente acessar uma página diferente das que acessou:

- (a) O algoritmo NRU, supondo que o processo altere o conteúdo das páginas 5 e 9, e que o bit **referenciada** de uma página é ligado quando ela é carregada na memória e desligado sempre que o processo acessa, em sequência, duas páginas diferentes dela.

Resp.: Como vimos na Aula 9, no algoritmo NRU, as páginas são primeiramente divididas em 4 classes diferentes, de acordo com o valor dos bits **modificada** (M) e **referenciada** (R): Classe 0: ambos os bits R e M são 0; Classe 1: o bit R é 0 e o bit M é 1; Classe 2: o bit R é 1 e o bit M é 0; e Classe 3: ambos os bits R e M são 1. Depois de dividir as páginas em classes, a página a ser substituída será uma das páginas da classe não vazia com o menor número. Como vimos na Aula 9, o bit R sempre é ligado quando a página é acessada. Na figura dada a seguir mostramos, em cada linha, como as páginas na memória são divididas em cada uma das 4 classes, após acessarmos a página dada na primeira coluna, de acordo com a ordem dada no enunciado. As classes são mostradas nas colunas de 2 até 5 e, na última coluna, é dito se o acesso à página causou uma falha de página. Também destacamos, em negrito, a(s) página(s) a ser(em) substituída(s). Como podemos ver por esta figura, ocorreram 8 falhas de página, e a próxima página a ser substituída seria a 9. Note que somente as páginas 5 e 9 podem estar nas classes 1 ou 3, pois somente para elas, que são alteradas, o bit M é 1.

Página	Classes				Ocorreu uma falha?
	0	1	2	3	
1	-	-	1	-	Sim
13	-	-	1,13	-	Sim
5	1	-	13	5	Sim
0	13	-	0	5	Sim
7	-	5	0,7	-	Sim
7	0	5	7	-	Não
5	0	-	7	5	Não
9	7	-	-	5,9	Sim
0	-	5	0	9	Sim
1	-	9	0,1	-	Sim

(b) O algoritmo LRU.

Resp.: Como vimos na Aula 9, no algoritmo LRU, as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo do seu último acesso. A página a ser substituída será a primeira página segundo esta ordenação, isto é, a página não acessada há mais tempo. Na figura dada a seguir mostramos,

em cada linha, o que ocorre ao acessarmos as páginas na ordem dada no enunciado. Para cada uma destas linhas mostramos, na primeira coluna, a página que é acessada, na segunda coluna, a ordem em que as páginas devem ser escolhidas e finalmente, na terceira coluna, se o acesso gera uma falha de página. A primeira página segundo a ordenação, que será a substituída, é mostrada em negrito. Como podemos ver por esta figura, teremos 8 falhas de página, e a próxima página a ser substituída será a 9.

Páginas	Ordenação	Ocorreu uma falha?
1	1	Sim
13	1 13	Sim
5	1 13 5	Sim
0	13 5 0	Sim
7	5 0 7	Sim
7	5 0 7	Não
5	0 7 5	Não
9	7 5 9	Sim
0	5 9 0	Sim
1	9 0 1	Sim

4. Em um sistema operacional que usa o algoritmo NRU para escolher as páginas a serem substituídas, foi obtida a sequência de substituições dadas na tabela a seguir, quando um processo acessou um conjunto de páginas virtuais. Nesta tabela, a primeira coluna mostra, da primeira para a última linha, a ordem em que as páginas foram acessadas, e a segunda coluna mostra a divisão das páginas nas classes definidas pelo algoritmo NRU. Suponha que o critério usado para o gerenciamento dos bits **referenciada** e **modificada** tenha sido o mesmo para todas as páginas virtuais, e que 3 molduras de página tenham sido alocadas ao processo. Qual será a página virtual a ser substituída após o processo acessar, na mesma ordem, todas as páginas da tabela, se o sistema operacional agora passar a usar, ao substituir as páginas:

Página virtual	Classes			
	0	1	2	3
1	-	-	1	-
13	-	-	1,13	-
5	1	-	13	5
0	13	-	0	5
7	-	5	0,7	-
7	0	5	7	-
5	0	-	7	5
9	7	-	-	5,9
0	-	5	0	9
1	-	9	0,1	-

Resp.: Pela tabela dada na questão, e pela definição do bit **referenciada**, vemos que o critério de atualização deste bit é o seguinte: quando a página é carregada na memória, o seu bit **referenciada** é alterado para 1. Depois disso, ele será alterado para 1 quando a página for novamente acessada, ou para 0 após acessarmos, em sequência, duas páginas diferentes desta página. Baseado neste critério de atualização, descreveremos em cada item a seguir quais páginas serão substituídas para cada algoritmo:

(a) O algoritmo de segunda chance?

Resp.: Ao usarmos o algoritmo de segunda chance, onde as páginas a serem escolhidas são ordenadas, em ordem crescente, de acordo com o tempo em que elas foram carregadas na memória, temos as escolhas dadas na tabela a seguir. Note que neste algoritmo, o bit **referenciada** da página carregada há mais tempo na memória se tornará 0 se ele for igual a 1, pois esta página deixará de ser escolhida e será colocada no final da ordenação. Nesta tabela, na primeira coluna, damos a página acessada, sendo que as páginas são mostradas na mesma ordem dada na tabela do enunciado. Na segunda coluna mostramos a ordem em que as páginas são escolhidas pelo algoritmo, sendo que ao lado de cada página damos, entre parênteses, o valor atual do seu bit **referenciada**. Finalmente, na terceira coluna, dizemos se o acesso à página gerou ou não uma falha de página. Pela última linha da tabela, vemos que a página 9 será a próxima a ser substituída.

Página	Ordem de escolha			Ocorreu uma falha?
1	1(1)			Sim
13	1(1)	13(1)		Sim
5	1(0)	13(1)	5(1)	Sim
0	13(0)	5(1)	0(1)	Sim
7	5(0)	0(1)	7(1)	Sim
7	5(0)	0(0)	7(1)	Não
5	5(1)	0(0)	7(1)	Não
9	0(0)	7(0)	5(0)	Sim
	7(0)	5(0)	9(1)	
0	5(0)	9(1)	0(1)	Sim
1	9(0)	0(1)	1(1)	Sim

- (b) O algoritmo de idade, sendo que cada contador tem 5 bits e que uma interrupção do temporizador é sempre gerada após o acesso a cada três páginas?

Resp.: Como vimos na Aula 9, o algoritmo de idade define, para cada página que está na memória, um contador, no nosso caso com 5 bits, que é inicializado em 0 quando a página é carregada na memória. A cada interrupção do temporizador, para cada página que está na memória, o seu contador é deslocado um bit para a direita e o seu bit **referenciada** é copiado para o bit mais significativo do seu contador. Quando precisarmos substituir uma página, a página com o menor contador será a escolhida para ser substituída. Na figura dada a seguir mostramos, na parte (a), o estado inicial antes de acessarmos qualquer página e, nas partes de (b) até (k), os valores dos contadores após acessarmos as páginas na ordem dada na tabela do enunciado (a linha da página acessada é destacada em negrito). Também damos, entre parênteses, após o número de cada página, o valor atual do seu bit **referenciada**. Também, nas partes de (d) até (k) da figura, as páginas escolhidas para serem substituídas são indicadas por uma seta. Note que nos três primeiros acessos não indicamos esta página, pois ainda existiam molduras não usadas. Também dizemos na figura, em cada parte com exceção da (a), se o acesso à página gerou ou não uma falha. Como uma interrupção do temporizador ocorrerá a cada três acessos, os contadores de todas as páginas serão atualizados somente nas partes (d), (g) e (j) da figura. Pelo item (k), vemos que a página 1 será a próxima a ser substituída. Note que na parte (h) poderíamos ter escolhido a página 5, mas como as páginas 13

e 5 não são mais refenciadas, a única diferença seria que a página 13, ao invés da 5, possuiria o contador com o valor 01000 no passo (i), e seria a escolhida no passo (j), pois o seu contador também seria igual a 00100.

Páginas - - - - - -	Páginas 1 (1) 00000 - - - -	Páginas 1 (1) 00000 13 (1) 00000 - -
Falha: - (a)	Falha: Sim (b)	Falha: Sim (c)
Páginas 1 (0) 00000 13 (1) 10000 5 (1) 10000	Páginas 0 (1) 00000 13 (0) 10000 5 (1) 10000	Páginas 7 (1) 00000 13 (0) 10000 5 (0) 10000
Falha: Sim (d)	Falha: Sim (e)	Falha: Sim (f)
Páginas 7 (1) 10000 13 (0) 01000 5 (0) 01000	Páginas 7 (1) 10000 13 (0) 01000 5 (1) 01000	Páginas 7 (0) 10000 9 (1) 00000 5 (1) 01000
Falha: Não (g)	Falha: Não (h)	Falha: Sim (i)
Páginas 7 (0) 01000 0 (1) 10000 5 (0) 00100	Páginas 7 (0) 01000 0 (1) 10000 1 (1) 00000	
Falha: Sim (j)	Falha: Sim (k)	