



Lista de Exercícios de Gerenciamento de Memória
Prof. Carlos A. Astudillo

Instruções. A continuação se apresenta um conjunto de problemas que lhe serão de ajuda para exercitar-se e estudar a matéria. **Estes problemas não têm uma nota associada. Porém, se recomenda fortemente resolvê-los.**

1. Um processo num sistema que utiliza memória virtual tem a seguinte tabela de paginação

Página	Quadro
0	0
1	4
2	1

Assuma que o endereço virtual está representado com 6 bit, onde os primeiros 2 bit são usados para determinar a página.

- (a) Qual é o tamanho de página e de quadro?
(b) Qual é o número máximo de páginas do processo?
(c) Qual é o endereço de memória física de uma chamada à memória **0b010110**?
2. Considere um sistema de administração de memória multinível onde o primeiro nível se define em base a segmentos em memória. Cada segmento armazena as páginas de maneira sequencial e as acessa através do número de página. Adicionalmente, este sistema usa a seguinte estrutura de endereços lógicos

| # Segmento virtual | # Página | Offset |

Cada endereço lógico tem 2 bit para o número de segmento, 8 bit para o número de página e 12 bit para o offset. (Note que os endereços estão representados usando seis dígitos hexadecimais. Portanto, temos 24 bit na representação, enquanto o endereço tem 22 bit. Então, todos os endereços têm um padding pela esquerda para ter o tamanho correto.) E cada endereço físico tem um tamanho de 20 bit.

As entradas da tabela de página têm 8 bit. Os segmentos deste sistema estão definidos pela seguinte tabela:

Tabela de segmento			
#	Início	Tamanho	Bandeiras
0	0x02004	64	Válido, leitura
1	0x00000	16	Válido, leitura/escrita
2	0x02040	64	Válido, leitura/escrita
3	0x01010	16	Inválido

onde # é o número do segmento, **início** denota o endereço de memória física onde o segmento começa, **tamanho** é a quantidade de páginas do segmento, e as bandeiras são as características do segmento. Esta tabela mapeia os segmentos em memória onde estão as páginas direcionadas diretamente pelo número de página em seu segmento respectivo.

O mapa da memória física é o seguinte:

Memoria física

Endereço	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x00000	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
0x00010	1E	1F	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D
...																
0x01010	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
...																
0x02000	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
...																
0x02040	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51
...																

O mapa representa os dois dígitos mais significativos dos endereços físicos. Por exemplo, o segmento 1 começa no endereço físico 0x00000; a primeira página esta na posição +0, então o endereço físico dela é 0x0E000; e a quinta página esta na posição +4, então o endereço físico dela é 0x12000.

- Qual é o endereço físico do endereço virtual 0x204ABC?
 - O que erro ocorre quando se traduz o endereço virtual 0x1103DB?
 - O que erro ocorre quando se traduz o endereço virtual 0x304F51?
 - Qual é o quantidade máxima de quadros que o sistema pode referenciar na RAM?
 - Qual é o tamanho de página do sistema?
3. Quais instruções podem invalidar o TLB? Justifique sua resposta para cada.
4. Considere um arranjo bidimensional A declarado como `char A[][] = new char[100][100]`; onde A[0][0] se encontra na posição 200, num sistema com memoria paginada com quadros de tamanho 200 B (suponha que o tipo de dato `char` tem um tamanho de 1 B) e com espaço em memoria para três quadros. Suponha que um processo pequeno se encontra manipulando a matriz e seu código reside na página 0 (com instruções nas localidades 0 a 199). E suponha que o código ha sido compilado utilizando uma linguagem que produz matrizes de tal forma que as colunas estão contíguas na memoria (*column-major*).
- Dado o seguinte código

```

1  for (int j=0; j<100; j++)
2    for (int i=0; i<100; i++)
3      A[i][j] = ' ';

```

Quantos falhos de página (*page-fault*) teremos?
 - Dado o seguinte código

```

1  for (int i=0; i<100; i++)
2    for (int j=0; j<100; j++)
3      A[i][j] = ' ';

```

Quantos falhos de página (*page-fault*) teremos?
5. Suponha que tem um sistema de 32 bit com memoria virtual com páginas de tamanho 2 KiB (2^{11} B). E um processo que usa 1024 (2^{10}) páginas do espaço de memoria virtual esta se executando.
- Quanto espaço ocupará a tabela de páginas para esse processo se é implementada como uma tabela densa de páginas de um nível? Em caso de ser impossível calculá-lo explique por que. Suponha que cada entrada na tabela de páginas ocupa 4 B. Pode deixar sua resposta em função de potencias de 2.

- (b) Suponha que o sistema utiliza uma tabela de páginas de dois níveis, de tal maneira que o tamanho do primeiro nível é 8 vezes o tamanho de uma tabela do segundo nível. E onde o primeiro nível é implementado de maneira esparsa. Calcule o tamanho **exato** da tabela antes descrita, em caso de ser impossível explique por que.
- Suponha que cada entrada na tabela de páginas ocupa 4 B. Pode deixar sua resposta em função de potências de 2.
- (c) Para a pergunta anterior, qual é a cota inferior do tamanho da tabela?
6. Considere a seguinte configuração de um sistema **x86** de 32 bit (todos os endereços estão em hexadecimal). Os endereços lógicos estão divididos por:

| 10 bit PDE | 10 bit PTE | 12 bit offset |

E o conteúdo do diretório de páginas (DP) e das tabelas de páginas (TP) do sistema é o seguinte. (Os endereços nos encabeçados das tabelas indicam onde estão armazenadas.)

Diretório Págs.		Tabela pág.		Tabela pág.	
0x0000 1		0x0000 2		0x0000 3	
0	0x0000 2	0	0x0000 5	0	0x0000 5
1	0x0000 3	1	0x0000 6	1	0x0000 5
2	0x0000 2	x	Inválido	x	Inválido
x	Inválido				

- (a) Qual é o endereço virtual do endereço físico **0x0000 5555**?
- (b) A qual endereço deve redirecionar o kernel quando pedimos uma leitura ou escrita ao endereço **0x0201 0000**?
- (c) Suponha que tem o código
- ```

1 int a = 0;
2 int b = 1;
3 a = b + 1;

```
- onde **a** se encontra armazenada no endereço **0x0080 1ABC** e **b** no endereço **0x0080 1AA0**. Suponha, além das condições anteriores, que o sistema tem um TLB com uma entrada. Explique como o sistema operacional resolve os endereços e opera o código anterior.
7. Suponha um sistema que tem um buffer de páginas de tamanho 4. Considere que o sistema tem a seguinte sequência de páginas 1, 2, 3, 4, 2, 5, 7, 2, 3, 2, 1, 7, e 8 como pedidos de página que deve atender.
- Para as seguintes políticas, calcule quais páginas geraram um falho de página, e detalhe o estado do buffer durante a sequência das páginas.
- (a) First in, first out
- (b) Last in, last out
- (c) Least recently used
- (d) Most recently used
8. (a) Descreva a diferença entre fragmentação interna e externa.
- (b) Indique qual das duas é mais provável de criar problemas em
- Uma máquina de memória simples usando registradores base e limite, e particionamento estático de memória.
  - Uma máquina similar a anterior com particionamento dinâmico de memória.

9. A anomalia de Bélády é a situação onde adicionar mais quadros aumenta o número de falhos de página para uma padrão de acesso particular.
- Suponha a sequência de acessos de página: 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4. Calcule a quantidade de falhos para um buffer de páginas de tamanho 3 e 4 usando uma política FIFO.
  - Explique porque acontece a anomalia de Bélády para o padrão de acesso anterior.
  - Poderia uma política *Least Recently Used* apresentar uma anomalia similar?
10. Para os seguintes esquemas de alocação de memória, explique se apresentam fragmentação interna ou externa e por que.
- Contígua
  - Segmentos
  - Paginada
11. Você tem que projetar um novo processador. Depois de selecionar todos os componentes, tem R\$ 460 000 para gastar nos seguintes componentes

| Item         | Latência      | Tamanho mínimo | Custo           |
|--------------|---------------|----------------|-----------------|
| TLB          | 10 ns         | 256 entradas   | R\$ 100/entrada |
| Memória      | 180 ns        | 2 GiB          | R\$ 10 000/GiB  |
| Disco rígido | 8 ms (8 M ns) | 300 GiB        | R\$ 100/GiB     |

O tamanho de página é fixo em 64 KiB. Suponha que você quer executar até 20 aplicativos simultaneamente. Cada aplicativo tem, em geral, um tamanho máximo de 1 GiB e um grupo de trabalho de 256 MiB. As entradas do TLB não têm identificador de processo.

Discuta como dividiria os recursos que tem disponíveis entre os distintos elementos para otimizar o rendimento do computador. (*Lembre que suas justificações devem ser técnicas.*)

12. Suponha que tem um sistema com memória paginada que usa uma tabela de páginas baseada numa *hash table* e um TLB. Neste sistema, o TLB tem um tempo de acesso  $\epsilon$  de 0.001 referências a memória, e as buscas da tabela de páginas toma, na média, 2 referências a memória (não incluindo a leitura do endereço solicitado). E, em promédio, uma referência a memória tem um hit no TLB de 70% do tempo. Se você tem que projetar o hardware e tem uma escolha entre aumentar o tamanho do TLB para melhorar o hit rate para 80%, ou para usar um algoritmo mais inteligente de hashing que reduz o tempo de busca na tabela de páginas a 1.8 referências, qual escolha você faz para melhorar o tempo de acesso efetivo no sistema? Justifique sua resposta.
13. Um sistema que tem endereços de 16 bit e páginas de 512 B, tem um TLB com os seguintes conteúdos

| V | R | M | VP   | PF   |
|---|---|---|------|------|
| 1 | 0 | 1 | 0x0C | 0x38 |
| 1 | 1 | 1 | 0x45 | 0x1D |
| 1 | 0 | 0 | 0x19 | 0x22 |
| 0 | 1 | 1 | 0x37 | 0x66 |

onde as primeiras três colunas correspondem aos bit válido, referenciado, e modificado da página, e as últimas duas colunas são o número de página virtual e o número de quadro.

- Qual é o endereço físico do endereço virtual 0x194A?
- Se o TLB usa o algoritmo *Not Recently Used* para evacuar as entradas, qual é a primeira e a segunda entrada a ser evacuadas do TLB se usamos o NRU?