

SISTEMAS OPERACIONAIS

INF09344 - Sistemas Operacionais / INF02780 - Sistemas Operacionais / INF02828 - Sistemas de Programação II

Prof^a. Roberta Lima Gomes (soufes@gmail.com)

2^a. Lista de Exercícios – Parte 2

Data de Entrega: não há. O objetivo da lista é ajudar no estudo individual dos alunos. Soluções de questões específicas poderão ser discutidas em sala de aula, conforme interesse dos alunos.

GERÊNCIA DE MEMÓRIA

- 1) Cite duas diferenças entre endereços lógicos e físicos.
- 2) Explique a diferença entre fragmentação interna e externa.
- 3) Compare as 2 estruturas de dados para controlar os blocos de memória livre, mostrando vantagens e desvantagens de cada uma delas.
- 4) Considere um espaço de endereçamento lógico de oito páginas de 1024 palavras de 2 bytes cada, mapeado em uma memória física de 32 quadros. Quantos bits existem no endereço lógico? E no endereço físico?
- 5) Determinado computador fornece a seus usuários um espaço de memória virtual de 256 Kbytes. O computador tem 140 Kbytes de memória física. A memória virtual é implementada por paginação e o tamanho de página é 4096 bytes. Um processo de usuário gera o endereço virtual 123456. Explique como o sistema estabelece a posição física correspondente. Faça a distinção entre as operações de software e hardware.
- 6) Considere o vetor bidimensional A:

```
int A[][] = new int[100][100];
```

onde A[0][0] está na posição 200, em um sistema paginado com páginas de tamanho 200 palavras. Observe que o vetor é armazenado em linha. Ou seja, o vetor é armazenado na ordem: A[0][0], A[0][1], ..., A[0][99], A[1][0], ..., A[99][99]. No sistema paginado em questão, cada página pode armazenar até 200 elementos de A (isso porque cada elemento do vetor tem o tamanho de uma palavra). Um processo pequeno está na página 0 (posições de 0 a 199, travado em memória) para manipular a matriz; assim, toda a busca de instruções será a partir da página 0.

Para uma memória com apenas três quadros de página, indique em que situação, código a) ou código b), ocorre mais faltas de páginas nos seguintes laços de inicialização do vetor, usando o algoritmo de substituição de página LRU em Software (Least Recently Used) e assumindo que a moldura 0 tem a página 0 do processo e que os outros dois estão inicialmente vazios. Justifique sua resposta.

- (a)

```
for (int j=0; j<100; j++)  
    for (int i=0; i<100; i++)  
        A[i][j]=0;
```
- (b)

```
for (int i=0; i<100; i++)  
    for (int j=0; j<100; j++)  
        A[i][j]=0;
```

- 7) Um pequeno computador tem quatro molduras de página. Na primeira interrupção de tempo, os bits R são 0111 (o da página 0 em 0 e os demais em 1). Nas interrupções seguintes tais valores são 1011, 1010, 1101, 0010, 1010, 1100, 0001. Se o algoritmo do envelhecimento (*aging*) for usado, com um contador de oito bits, dê o valor dos quatro contadores após a última interrupção.
- 8) Considere um sistema com páginas de 4K, endereçamento lógico de 16 páginas, e endereçamento físico de 8 frames. Considere a seguinte tabela de página do processo em execução:

	bit validade	moldura
0	1	2
1	1	1
2	1	6
3	1	0
4	1	4
5	1	3
6	0	-
7	0	-
8	0	-
9	1	5
10	0	-
11	1	7
12	0	-
13	0	-
14	0	-
15	0	-

- a) Mostre em quais endereços físicos a MMU traduz cada uma das seguintes referências à memória feitas pelo processo corrente: 8292 e 28094.
- b) Mostre como fica uma Tabela Invertida para este sistema, com o mesmo mapeamento, e como os endereços do item anterior são traduzidos com esta tabela.
- c) Mostre como fica uma Tabela em dois níveis para este sistema com o mesmo mapeamento, se o primeiro e segundo níveis suportarem 4 entradas cada. Além disso, mostre como os endereços do item a) são traduzidos com esta tabela.

- 9) Considere um sistema com páginas de 4K, endereçamento de 16 bits e memória física de 16K, cujo Sistema Operacional trabalhe com Memória Virtual Paginada. Considere que nenhum programa esteja em execução e que todas as molduras estejam inicialmente livres. Por simplicidade, suponha que todas as molduras possam ser ocupadas pelo programa (embora isto seja irreal, já que pelo menos o Sistema Operacional deve estar em execução e ocupando espaço de memória). Suponha que um programa entre em execução e faça referência aos seguintes endereços virtuais, nesta ordem: 200, 8191, 4196, 16383, 8192, 65535. Se é usado o algoritmo de troca de página LRU, responda:

- a) Construa uma Tabela de Páginas (Convencional, ou Invertida ou Multinível) para este sistema, indicando os campos existente na Tabela de Páginas.
- b) Mostre para cada endereço de memória virtual referenciado pelo programa, em qual endereço físico ele é traduzido, e como este processo é feito pela MMU e pelo Sistema Operacional. Atualize a Tabela de Páginas, quando necessário.

- 10) Considere um sistema de paginação em que haja 4 molduras e seis páginas virtuais. Na sequência de eventos listada abaixo, “Read x” significa que a informação contida em um endereço na página virtual ‘x’ deve ser lida; “Write x” significa que uma informação será escrita em um endereço na página virtual ‘x’; “TICK” é um *tick* de relógio. (Lembre-se que os argumentos de “Read” e “Write” são números de páginas virtuais, e não de

molduras). Liste as ocorrências de *Page Faults* e os números das molduras escolhidas para receber as páginas virtuais caso o S.O. utilize a) NRU (*Not Recently Used*); b) *Second Chance*; c) *Aging*. Se o algoritmo gerar um empate na decisão de páginas a serem substituídas, escolha a que estiver posicionada na moldura de menor número. Assuma que, inicialmente, a memória encontra-se vazia.

Time: Events:	NRU	Second Chance	Aging
1 Write 1			
2 Read 2			
3 Read 3			
4 TICK			
5 Write 2			
6 Read 4			
7 Read 5			
8 TICK			
9 Read 1			
10 Write 6			
11 Write 3			
12 TICK			
13 Read 2			
14 Write 1			
15 Write 4			
16 TICK			
17 Write 5			
18 Write 2			

11) Considere um sistema de memória composto de 4 molduras de páginas de 2k cada em um espaço de endereçamento total de 16 bits. Se no instante 233 os seguintes dados estão disponíveis:

Moldura	Página Virtual	Carga	Última Referencia	R	M
1	7	100	231	0	1
2	8	112	233	1	1
3	5	187	230	0	1
4	10	200	232	0	0

Mostre qual página será removida da memória para se atender a requisição de acesso ao endereço virtual 4095, para cada um dos seguintes algoritmos: FIFO, NRU, LRU e 2a. chance.

SISTEMAS DE ARQUIVOS

- 1) Apresente, pelo menos, uma vantagem e uma desvantagem de cada uma das implementações para alocação de blocos de arquivos: contígua, lista encadeada, e lista encadeada usando tabela em memória (FAT), i-nodes.
- 2) Fale sobre fragmentação interna e externa em sistemas de arquivos.

- 3) Quantas operações em disco são necessárias para acessar o primeiro bloco do arquivo `/home/joao/so/projeto.c`, considerando i-nodes que só têm endereços diretos de blocos? E o seu correspondente na FAT (`C:\home\joao\so\projeto.c`)? Suponha que o diretório raiz esteja na memória, mas nenhum outro componente ao longo do caminho se encontre na memória. Suponha também que todos os diretórios caibam em um único bloco de disco. Explique suas respostas.
- 4) Considere um sistema de arquivos em um disco que tem tamanhos de bloco físico e lógico de 512 bytes e endereçamento de blocos de 32 bits. Suponha que as únicas informações dos arquivos mantidos em memória sejam as entradas de diretório dos arquivos abertos. Para cada uma das três estratégias de alocação (contígua, lista encadeada, e lista encadeada usando tabela em memória), responda as seguintes questões:
- Dado um deslocamento em bytes a partir do início do arquivo, como é obtido o número do Bloco físico em que este byte se encontra?
 - Se o último acesso a um bloco de um arquivo foi o bloco lógico 10, quantos blocos de discos devem ser lidos para acessar o bloco lógico 4?
- 5) Imagine um sistema de arquivos Unix sobre um disco de 160 Gigabytes, com blocos de 1K. Suponha que sejam usados 8 bytes para descrever o endereço de cada bloco.
- Explique qual seria a dimensão máxima de um arquivo neste sistema considerando que o i-node possui 10 entradas diretas de endereços para blocos, 1 entrada indireta simples (para um bloco de ponteiros), 1 entrada indireta dupla (para um bloco de ponteiros para blocos de ponteiros), e 1 entrada indireta tripla (para um bloco de ponteiros ...)
 - Suponha que o arquivo texto `/tmp/test` seja aberto e que seja feita uma chamada ao sistema para avançar a posição no arquivo em 266Kbytes a partir do início, e em seguida seja feita uma leitura. Descreva com o auxílio de um diagrama como será encontrada a localização em disco da posição desejada, partindo do i-node do arquivo.
 - Supondo que apenas a entrada do diretório `/tmp` encontrava-se na memória, explique quantos acessos a disco foram necessários para a leitura da posição 266Kbytes do arquivo `/tmp/test`.
- 6) Acessar um arquivo pelo seu nome relativo é mais rápido do que fazê-lo pelo seu nome absoluto? Por quê?
- 7) Qual a importância das chamadas ao sistema `open` e `close`?
- 8) Explique como o tamanho do bloco pode influenciar no desempenho do sistema de arquivos.
- 9) Sobre gerência de espaço livre em disco, responda:
- Para um sistema de arquivos que opera com blocos de 1Kb, quantos blocos são necessários para armazenar o mapa de bits a ser utilizado na gerência de blocos livres de um disco rígido de 4Gb? Assuma que vai estar disponível, em algum outro lugar, a lista ligada que indica a seqüência de blocos utilizados para armazenar o mapa de bits.
 - Assumindo que o mesmo disco esteja vazio (0% de ocupação), quantos blocos seriam necessários para armazenar a lista ligada de blocos livres, caso esta viesse a ser utilizada em lugar do mapa de bits? Nota: Os blocos são endereçados usando 32 bits.
 - ... E se o disco estivesse com 50% de ocupação?
 - ... E se o disco estivesse com 97% de ocupação?

e) Comente os resultados obtidos.

10) Cite duas vantagens do mapa de bits em relação à lista de blocos livres.

11) Diferencie soft link de hard link.

12) Explique um possível uso para o campo de um inode que armazena o número de referências que são feitas por diretórios junto ao arquivo associado àquele inode.

13) Fale sobre o uso de buffers nas implementações de sistemas de arquivos do UNIX.