

## Lista de Exercícios para a P2/PF

### INF1316

A segunda. parte da matéria trata de Gerenciamento de Memória, Sistemas de Arquivos, e Entrada e Saida, além de todas as aulas de laboratório.

#### Gerenciamento de Memória

M1-Explique a diferença entre fragmentação interna e externa.

M2- 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?

M3- 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 física 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;
```

M4- Considere o seguinte padrão de referência de página:

**1; 2; 3; 4; 2; 1; 5; 6; 2; 1; 2; 3; 7; 6; 3; 2; 1; 2; 3; 6**

Qual o menor número de quadros que evita faltas de páginas em, pelo menos, 50% destas referências, considerando a substituição LRU ? Considere que todos os quadros de páginas estão inicialmente vazios.

M5- 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.

Construa uma Tabela de Páginas (Convencional e Multinível) para este sistema, indicando os campos existente na Tabela de Páginas.

M6- Analise a veracidade das afirmativas abaixo, justificando sua resposta:

- a) Quando se aumenta a quantidade de memória diminui-se a taxa de falta de página.
- b) Um esquema de segmentação pura causa fragmentação interna e externa.
- c) Os motivos que levam o processador a ficar muito ocioso num sistema multitarefa, podem ser pouca quantidade de memória quanto tamanho de página muito pequeno.
- d) Programas mal projetados podem causar muitas faltas de páginas
- e) MMU é um dispositivo responsável pelo tratamento de falta de página.

## Sistemas de Arquivos

F1- 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:

- a. a. 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?
- b. b. 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?

F2- 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.

a. 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 ... )

b. 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. c. 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.

F3- Explique como funciona um *Journaling File System*, e qual é o principal benefício. Diga se eles são geralmente mais eficientes ou menos eficientes do que Sistemas de Arquivos convencionais.

F4- Como as operações `open()` e `close()` em arquivos afetam o cache do SA na memória principal. Explique. E o `opendir()` e o `closedir()`?

F5- Sobre gerência de espaço livre em disco, responda:

a) 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. (Resp:512 blocos)

b) 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. (Resp: 16449 blocos)

F6- Para agilizar o acesso a arquivos e diretórios recentemente acessados o Sistema de Arquivos no UNIX mantém em cache (na memória RAM) os blocos de diretórios lidos, os inodes de arquivos abertos e os blocos de disco acessados recentemente, e só copia estes de volta para o disco com uma certa frequência pré-determinada. Indique pelo menos uma vantagem e uma desvantagem de proceder dessa forma, (já que outros SA fazem a copia do disco a cada modificação no bloco ou inode)

F7- Sabendo que um inode possui D endereços para blocos diretos, 1 indireto, 1 duplamente indireto e 1 triplamente indireto, e cada bloco de índice comporta N endereços para blocos, escreva as expressões algébricas que indicam o número máximo e o número mínimo de blocos ocupados por um arquivo A quando se sabe que o endereços duplamente indireto e um triplamente direto em um inode não estão usados.

Min\_blocks (A) =

Max\_blocks (A) =

## Entrada & Saída

ES1- Explique o que é um DMA (Direct Memory Access) e qual é o seu papel na E/S de bloco, em particular, no swapping de páginas na paginação. O que aconteceria com a CPU nesses momentos.

ES2- Qual é o benefício de identificar e tratar todo dispositivo de E/S como um arquivo, como acontece no Unix?

ES3- Explique por que para alguns periféricos voce precisa reinicializar o seu computador antes de poder usar o novo dispositivo.

ES4- Qual é o papel da controladora de um dispositivo de E/S? Pense em dispositivos mecânicos ou eletrônicos mais simples.