



Curso de Tecnologia em Sistemas de Computação  
Disciplina de Sistemas Operacionais  
**Professores:** Valmir C. Barbosa e Felipe M. G. França  
**Assistente:** Alexandre H. L. Porto

Quarto Período  
AP2 - Segundo Semestre de 2007

Nome -

Assinatura -

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (1.25) Descreva sucintamente como as operações de E/S sobre um dispositivo físico são executadas pelo sistema operacional em um hardware que não possui uma DMA.

**Resp.:** Em um hardware sem uma DMA, o sistema operacional será o responsável pelas transferências de dados entre a controladora do dispositivo físico e a memória, ao executarmos operações de E/S. Neste caso, ao lermos dados do dispositivo, o sistema enviará os comandos necessários à controladora para que ela inicie a leitura no dispositivo. Dependendo do modo de como a operação de E/S é tratada, o sistema pode esperar pelo término da operação (E/S síncrona), ou a controladora pode gerar uma interrupção para avisar ao sistema que a operação terminou (E/S assíncrona). Quando o sistema souber do término da operação de E/S, ele copiará os dados da controladora (a partir, por exemplo, de um buffer interno) para a memória, se a operação foi executada com sucesso. A escrita de dados no dispositivo é similar à leitura mas, neste caso, o sistema primeiro copia os dados da memória para a controladora, e somente depois envia os comandos à controladora para que ela inicie a operação de escrita no dispositivo. Assim como antes, o sistema pode esperar pelo término da operação, ou pode ser interrompido quando a operação for concluída.

2. (1.25) Suponha que um computador possui 1MB de memória. Se o sistema operacional usar o gerenciamento de memória por troca, e se cada unidade de alocação possuir 64KB, qual seria o tamanho de um mapa de bits? E qual seria a fração de memória usada pelo mapa de bits?

**Resp.:-** Como temos uma memória de 1MB, ou seja, de 1024KB, e como cada unidade de alocação tem 64KB de tamanho, então temos 16 unidades de alocação na memória. Agora, como cada unidade de alocação ocupa um bit no mapa de bits, então o mapa de bits terá 16 bits, ou seja, 2 bytes.

-Como o mapa de bits ocupa 2 bytes, e como a memória tem 1MB, ou seja,  $2^{20}$  bytes, a fração gasta da memória será de  $2/2^{20} = 1/2^{19}$  (ou de aproximadamente 0,000002, o que equivale a aproximadamente 0,0002% da memória).

3. (3.0) Um sistema operacional gerencia a memória do computador através do uso da técnica de paginação. Suponha que o tamanho de cada página é de 64KB. Responda:

- (a) (1.0) Qual seria o tamanho da memória virtual se existirem 128 páginas virtuais? Quantos bits teria o endereço virtual, e como ele seria dividido?

**Resp.:**-Como temos 128 páginas virtuais de 64KB, então o tamanho da memória virtual será de 8192KB, isto é, 8MB.

-Como temos  $128 = 2^7$  páginas virtuais, o campo do endereço virtual que define a página virtual deverá ter 7 bits. Já o campo que define o deslocamento dentro da página virtual neste endereço deverá ter 16 bits, pois cada página tem 64KB de tamanho, isto é,  $65536 = 2^{16}$  bytes. Logo, o endereço virtual será composto de 23 bits, sendo que os 7 bits mais significativos definirão a página virtual, e os 16 bits menos significativos definirão o deslocamento dentro desta página.

- (b) (1.0) Qual seria o tamanho da memória física se pudermos armazenar no máximo 1/4 das 128 páginas virtuais na memória física? Quantos bits teria o endereço físico, e como ele seria dividido?

**Resp.:** Como podemos armazenar 1/4 das 128 páginas virtuais na memória física, então isso significa que ela possui  $128/4 = 32$  molduras de página. Agora, como cada moldura de página tem 64KB de tamanho (pois o seu tamanho é igual ao tamanho da página virtual), então o tamanho da memória física será de 2048KB, isto é, 2MB.

-Como temos  $32 = 2^5$  molduras de página, então o campo que define a moldura de página no endereço físico deverá ter 5 bits. Com isso, o endereço físico deverá ter 21 bits, pois o campo que define o deslocamento deverá ter 16 bits, porque cada moldura de página tem  $65536 = 2^{16}$  bytes (64KB) de tamanho. O endereço então deverá ser dividido do seguinte modo: os 5 bits superiores indicarão a moldura de página, e os 16 bits inferiores restantes

indicarão o deslocamento dentro desta moldura.

- (c) (1.0) Suponha que 3 molduras de página foram alocadas ao processo. Na tabela dada a seguir mostramos como as páginas são alocadas às molduras de página, ao acessarmos, em ordem, as páginas virtuais 0, 27, 74, 0, 6 e 0. Supondo que em cada linha as páginas são mostradas na ordem em que serão escolhidas para serem substituídas pelo algoritmo de substituição de páginas, o algoritmo usado poderia ser o LRU? Justifique a sua resposta.

Página	Molduras	Falha de página?
0	0	Sim
27	0 27	Sim
74	0 27 74	Sim
0	0 27 74	Não
6	27 74 6	Sim
0	74 27 0	Sim

**Resp.:** Como vimos na Aula 9, o algoritmo LRU ordena as páginas a serem substituídas em uma ordem crescente de acordo com a última vez em que elas foram acessadas. Como na tabela anterior as páginas estão ordenadas em uma ordem crescente de acordo com o tempo em que elas foram copiadas para a memória, o algoritmo usado não pode ter sido o LRU. Note que o algoritmo usado poderia ser sido o FIFO, e que se o algoritmo usado fosse o LRU, as escolhas deveriam ter sido como as dadas na tabela a seguir. Além disso, note que também deveríamos ter tido uma falha de página a menos.

Página	Molduras	Falha de página?
0	0	Sim
27	0 27	Sim
74	0 27 74	Sim
0	27 74 0	Não
6	74 0 6	Sim
0	74 6 0	Não

4. (1.25) Quais são as vantagens de usarmos a técnica de segmentação em relação à técnica de paginação? Se existirem, quais seriam as desvan-

tagens?

**Resp.:-** Como a técnica de segmentação disponibiliza aos processos vários espaços de endereçamento virtual com tamanho e proteção distintos (os segmentos), temos várias vantagens em usá-la. Uma vantagem é a de facilitar a proteção dos dados e do código do processo, pois poderemos usar segmentos diferentes com proteções diferentes para os dados e o código (por exemplo, o segmento de dados não pode ser executável, e não é desejável que o segmento de código seja alterável). Uma outra vantagem é a da maior facilidade em compartilhar dados e procedimentos entre vários processos, pois se os colocarmos em segmentos diferentes, isso facilitará tanto o compartilhamento (pois bastará compartilhar o segmento com os dados ou o procedimento) como a realocação (pois os dados ou o procedimento podem sempre estar no início do segmento). A programação também pode ser facilitada se um programa usar estruturas com tamanho variável, pois o programador poderá usar um segmento diferente para cada estrutura, cujo tamanho será sempre igual ao tamanho da estrutura.

-Existem também várias desvantagens ao usarmos a técnica de segmentação. Uma delas é a de que a técnica de segmentação pura (quando a técnica de paginação não é usada em cada segmento) não permite segmentos maiores do que o tamanho da memória física. Outra desvantagem é que a técnica gera uma fragmentação externa da memória, conforme os segmentos são movidos entre a memória e o disco, removidos quando não são mais usados, ou movidos na memória após o seu tamanho ser alterado. Finalmente, uma outra desvantagem é que a técnica é visível ao programador, pois ele deve explicitamente definir o conteúdo e o tipo de proteção de cada um dos segmentos usados pelo programa, mesmo que isso não seja necessário.

5. (1.25) O número máximo de blocos do disco que podem ser alocados a um arquivo sempre será igual ao número de blocos livres do disco? Justifique a sua resposta.

**Resp.:** Depende da técnica usada para alocar os blocos do disco a um arquivo. Se a técnica usada não for a alocação contígua, então o número

máximo de blocos que podem ser alocados a um arquivo será de fato igual ao número de blocos livres do disco pois, como vimos na Aula 11, não existirá uma relação entre os blocos alocados ao arquivo. Agora, se usarmos a alocação contígua, somente será possível alocar todos os blocos livres a um arquivo se estes forem consecutivos ao último bloco do arquivo no disco, pois agora os blocos alocados ao arquivo deverão ser consecutivos. Isso será possível se usarmos a compactação do disco, pois bastará armazenar todos os blocos livres ao final do disco, imediatamente depois do último bloco do arquivo. Logo, somente poderemos garantir a alocação de todos os blocos livres a um arquivo ao usarmos a alocação contígua, se a compactação do disco estiver disponível no sistema operacional.

6. (2.0) Suponha que depois de uma verificação de consistência do sistema de arquivos, obtivemos as seguintes tabelas que, como vimos na Aula 12, contabilizam os blocos em utilização e os blocos livres, para um disco com 16 blocos. Responda:

		Blocos do disco															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Blocos em utilização		1	0	0	1	1	0	2	0	0	1	0	0	1	1	0	1
		0	1	0	0	0	1	0	1	1	0	2	1	0	0	1	0
		Blocos livres															

- (a) (1.0) Existem problemas de consistência neste sistema de arquivos. Quais são eles?

**Resp.:** Sim, existem três problemas de consistência. O bloco 2 do disco é um bloco ausente (um bloco do disco que não pode ser usado) pois, em ambas as tabelas, o seu contador é igual a zero. O bloco 6 está sendo usado por mais de um arquivo pois, como podemos ver pela tabela Blocos em utilização, o seu contador é igual a 2, o que significa que ele está associado a dois arquivos. Note que este é um problema sério de consistência, pois certamente um

dos arquivos está corrompido. Finalmente, o bloco 10 está duplicado na lista de blocos livres, pois o seu contador na tabela Blocos livres é igual a 2 (como veremos na resposta da próxima questão, deveremos necessariamente ter uma lista neste caso).

- (b) (1.0) Quais dos problemas de consistência não poderia ocorrer se usarmos um mapa de bits para sabermos quais blocos estão livres no disco?

**Resp.:** O único problema que não poderia ocorrer se usarmos um mapa de bits para gerenciar os blocos livres do disco seria o de o bloco 10 estar duplicado (com o contador igual a 2) na tabela Blocos livres. Isso é devido a nunca podermos ter os contadores da tabela Blocos livres com valores maiores do que 1 se usarmos um mapa de bits, porque temos somente um bit associado a cada bloco do disco, que pode ser igual a 0 ou a 1.