



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 2010

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. (2,5) Diga se as seguintes afirmativas são falsas ou verdadeiras. Para responder, escreva apenas F ou V para cada item em seu caderno de respostas.

- (a) (0,5) Quando um espaço de endereçamento especial é usado para acessar os registradores da controladora de um dispositivo físico, eles são mapeados em uma região da memória.

**Resp.:** F (Falsa, porque o que foi descrito neste item ocorre na E/S mapeada em memória, e não em um espaço de endereçamento especial, no qual as instruções especiais de E/S do processador são usadas para acessar os registradores da controladora).

- (b) (0,5) Quando um hardware possui uma DMA, ela é a responsável pela transferência de dados entre cada dispositivo de E/S e a memória. Neste caso, o processador somente precisa inicializar os registradores da controladora de cada dispositivo para definir que uma transferência de dados deve ser inicializada.

**Resp.:** V (Verdadeira).

- (c) (0,5) A diferença entre um dispositivo compartilhável e um dispositivo dedicado é que o primeiro pode ser usado simultaneamente por diversos processos enquanto o último somente pode ser usado por um único processo em um dado intervalo de tempo.

**Resp.:** V (Verdadeira).

- (d) (0,5) Um impasse poderá ocorrer quando um processo, que possui somente recursos não-preemptivos, tentar obter algum recurso preemptivo.

**Resp.:** F (Falsa, pois o impasse pode ser evitado se o recurso preemptivo for retirado do processo que o possui, para depois ser entregue ao processo que o deseja).

- (e) (0,5) O grafo de recursos é um modelo que define as relações de dependência entre todos os recursos do sistema. Cada vértice do grafo está associado a um recurso, e uma aresta orientada indica que o recurso associado ao vértice origem da aresta depende do recurso associado ao vértice destino da aresta.

**Resp.:** F (Falsa, pois o grafo de recursos modela o estado da alocação dos recursos do sistema operacional. Neste grafo, existe um vértice para cada recurso e um vértice para cada processo do sistema. Uma aresta orientada de um vértice que representa um recurso para um vértice que representa um processo indica que o processo possui o recurso. Finalmente, uma aresta orientada de um vértice que representa um processo para um vértice que representa um recurso indica que o processo deseja o recurso).

2. (1,5) Qual é a vantagem do gerenciamento por memória virtual sobre o gerenciamento por *overlays*?

**Resp.:** Como vimos na aula 8, os dois métodos de gerenciamento são usados para permitir a execução de programas cujos tamanhos são maiores do que a memória física disponível no computador. No gerenciamento por *overlays*, a programação é dificultada, porque cabe ao programador dividir o programa em partes diferentes, chamadas de *overlays*, que podem ser carregadas na memória pelo sistema operacional quando forem necessárias. A vantagem do gerenciamento por memória virtual sobre o gerenciamento por *overlays* é exatamente que a programação é facilitada, porque agora o sistema operacional é o responsável por dividir cada processo em partes diferentes. Estas partes são chamadas de páginas virtuais, e também podem ser carregadas na memória somente quando foram necessárias.

3. (1,5) Descreva a utilidade de uma TLB.

**Resp.:** A TLB (Translation Lookaside Buffer) é uma memória associativa interna à MMU que é usada para acelerar a conversão de alguns endereços virtuais em endereços físicos, ao permitir que esta conversão

seja feita sem o uso da tabela de páginas. Para poder fazer isso, a TLB armazena as informações mais importantes (como o número da moldura de página com a página virtual) de um pequeno número de entradas da tabela de páginas. Quando a TLB é usada, ao mapear um endereço virtual em um físico, primeiramente a MMU, após obter a página virtual associada a este endereço, verifica (de modo eficiente, pois a memória é associativa) se as informações mais importantes desta página estão na TLB. Se as informações estiverem na TLB, então o número da moldura de página será obtido diretamente da entrada associada à página na TLB, sem precisar acessar a tabela de páginas. Caso as informações não estejam na TLB, então o número da moldura deverá ser obtido acessando a entrada da página na tabela de páginas. Depois de obtido o número da moldura de página, o endereço físico será computado do modo como foi visto na aula 9, usando este número e o deslocamento dentro da página virtual. Como uma TLB armazena números de molduras de página, somente as páginas mapeadas em molduras terão as suas informações armazenadas na TLB. Além disso, para garantir o maior ganho em termos de velocidade de conversão, a TLB sempre deverá armazenar as informações associadas às páginas virtuais mais acessadas pelo processo em execução no processador.

4. (1,5) Suponha que um sistema operacional use o gerenciamento de memória por segmentação com paginação. Se cada segmento puder ser dividido em 64 páginas de 8KB, quantos segmentos existirão se um programa puder acessar até 32GB de memória?

**Resp:.** Como cada segmento é dividido em  $64 = 2^6$  páginas de 8KB, ou  $2^{13}$  bytes, então o tamanho de cada segmento é de  $2^6 \times 2^{13} = 2^{19}$  bytes. Agora, como o programa não pode acessar mais de 32GB, isto é,  $2^{35}$  bytes, então teremos  $2^{35}/2^{19} = 2^{16} = 65536$  segmentos.

5. (1,5) Quais são as funções das operações **retroceder** e **seek** quando acessamos bytes (ou registros) de um arquivo? Ambas as operações podem ser aplicadas a arquivos com acesso sequencial? E em relação a arquivos com acesso aleatório?

**Resp.:** -A operação **retroceder** faz com que a próxima posição a ser acessada do arquivo seja a sua primeira posição. Já a operação **seek** faz com que a posição passada como parâmetro à operação seja a próxima posição a ser acessada.

-Em relação ao acesso sequencial, a operação **seek** não pode ser usada porque, neste tipo de acesso, somente podemos acessar uma posição do arquivo após termos acessado, em sequência, todas as posições anteriores a ela no mesmo arquivo. Já a operação **retroceder** pode ser usada sem problemas, pois a próxima posição a ser acessada será a primeira posição do arquivo.

-Já em relação ao acesso aleatório, que permite acessar as posições do arquivo independentemente umas das outras, ambas as operações podem ser usadas. Note porém que a operação **retroceder** é desnecessária, porque ela pode ser simulada usando uma operação **seek** para a qual é passada, como parâmetro, a primeira posição do arquivo.

6. (1,5) Suponha que, depois de uma verificação de consistência do sistema de arquivos, tenhamos obtido as seguintes tabelas que, como vimos na Aula 12, contabilizam os blocos em utilização e os blocos livres, para um disco com 16 blocos. Um aluno de sistemas operacionais afirmou que não existem inconsistências neste sistema de arquivos. Se você acha que o aluno está correto, basta dizer que sim mas, se você acha que ele está errado, diga somente quais são os blocos que fizeram o sistema de arquivos ser inconsistente.

		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	0	1	2	1	0	0	1	0	1	0	1	0	4
Blocos livres		0	1	0	1	0	0	0	1	3	0	1	0	0	0	1	0

**Resp.:** O aluno está errado, pois as tabelas da questão mostram que existem cinco inconsistências no sistema de arquivos, devidas aos blocos 2, 5, 8, 12 e 15 do disco. Na sua resposta, você não precisa dar as explicações dadas a seguir sobre os blocos, mas somente dizer os blocos dados anteriormente. Os blocos 2 e 12 são blocos ausentes, pois não estão sendo usados por nenhum arquivo (as suas entradas são ambas

iguais a 0 na primeira tabela) e nem estão marcados como livres (as entradas para estes blocos também são ambas iguais a 0 na segunda tabela). Já os blocos 5 e 15 foram incorretamente alocados a mais de um arquivo (o bloco 5 a dois arquivos e o bloco 15 a quatro arquivos), o que pode ter feito com que os conteúdos dos arquivos que os compartilham estejam corrompidos. Finalmente, o bloco 8 aparece três vezes na lista de blocos livres (note que este erro nunca poderia ocorrer se o sistema operacional usasse um mapa de bits, pois nele existe um único bit para cada bloco do disco), o que significa que ele pode futuramente ser incorretamente alocado a até três arquivos diferentes, podendo tornar o conteúdo destes arquivos corrompido.