



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 - Primeiro Semestre de 2007

Nome -

Assinatura -

---

Observações:

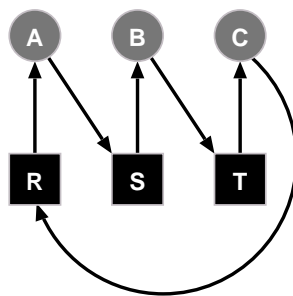
1. Prova sem consulta e sem uso de máquina de calcular.
  2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
  3. Você pode usar lápis para responder as questões.
  4. Ao final da prova devolva as folhas de questões e as de respostas.
  5. Todas as respostas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

1. (1.0) Como é feita a transferência de dados entre a memória e os dispositivos físicos sem a existência de DMA? Existe alguma vantagem em usarmos DMA?

**Resp.:-**Quando não existe DMA no hardware, a transferência de dados entre os dispositivos físicos e a memória é feita pelo processador. Logo, parte do tempo de processamento, que poderia ser usada para executar processos, será agora gasta com estas transferências de dados.

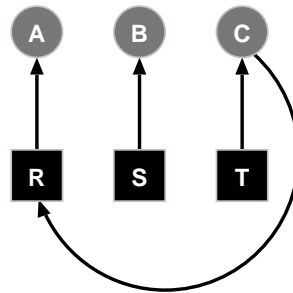
-A vantagem de usarmos DMA é a de que não perderemos mais tempo de processamento nas transferências de dados entre os dispositivos físicos e a memória, porque elas serão agora feitas pela DMA. Com isso, todo o tempo de processamento será usado para executar processos ou para executar tarefas do sistema operacional (como, por exemplo, o escalonador, muito embora o tempo gasto nessa tarefa não seja, em geral, significativo).

2. (2.0) Suponha que temos três processos, A, B e C, em execução no sistema, e que temos três recursos não-preemptivos, R, S e T. Depois de uma dada sequência de pedidos de recursos ao sistema pelos processos, obtivemos o grafo de recursos dado a seguir, em que podemos notar a existência de um impasse. Se numerarmos os recursos R, S e T, com, respectivamente, 1, 2 e 3, ainda teremos um impasse se um processo não conseguir alocar um recurso cujo número é maior do que o recurso que ele possui? Justifique a sua resposta.



**Resp.:** Na Aula 7 vimos que a numeração dos recursos e a definição de uma regra que proíbe um processo de alocar um recurso cujo número é menor do que os de todos os recursos que ele possui evita os impasses,

pois a condição de espera circular não ocorrerá mais. Note que esta regra obriga cada processo a alocar os recursos em ordem crescente, de acordo com os números atribuídos aos recursos. Nesta questão definimos uma regra similar à anterior, no tocante ao uso de uma ordem global associada aos recursos, mas agora os processos deverão alocar os recursos em ordem decrescente. Neste novo modo, um processo é proibido de alocar um recurso cujo número é maior do que os dos recursos que ele possui. É possível concluir que não teremos um impasse definido pois não existirá mais um ciclo no grafo, como podemos ver pela figura dada a seguir. O processo A falhará ao tentar alocar S numerado com 2 e não será bloqueado, pois ele possui R, numerado com 1, e  $2 \not< 1$ . O processo B também falhará e não será bloqueado ao tentar obter T numerado com 3, pois o recurso que ele possui, S, foi numerado com 2, e  $3 \not< 2$ . Porém, o processo C será bloqueado ao tentar obter o recurso R, pois ele possui T, que foi numerado com 3, R foi numerado com 1, e  $1 < 3$ .



3. (2.0) Suponha que um computador possui uma memória física de 128KB e uma memória virtual de 256KB. Suponha que o tamanho da página virtual é de 16KB. Responda:

- (a) (1.0) Como seriam divididos os endereços virtuais? E os endereços físicos?

**Resp.:-**Como a memória virtual possui 256KB, isto é,  $2^{18}$  bytes, e como o tamanho da página virtual é de 16KB, ou seja,  $2^{14}$  bytes, então precisaremos de 14 bits para representar um deslocamento dentro da página, e de 4 bits para representar a página (temos 16

páginas virtuais). Logo, os 4 bits mais significativos do endereço virtual, com 18 bits, definirão a página, e os 14 bits menos significativos definirão o deslocamento.

-Precisaremos de 14 bits para representar o deslocamento dentro da moldura de página, pois a moldura tem o mesmo tamanho da página virtual, ou seja,  $2^{14}$  bytes (16KB). Porém, agora precisaremos de 3 bits para representar as molduras de página (temos  $2^3 = 8$  molduras), pois a memória física possui  $2^{17}$  bytes (128KB). Logo, o endereço físico, com 17 bits, será dividido do seguinte modo: os 3 bits mais significativos indicarão a moldura de página, e os 14 bits menos significativos indicarão o deslocamento.

- (b) (1.0) Suponha que o sistema operacional alocou 3 molduras de página para um processo, e que somente uma delas está não vazia, contendo a Página Virtual 3. Quantas falhas de página serão geradas se o processo acessar as páginas virtuais na ordem 0, 1, 3, 4, 2, 3, 1, 0, 3?

**Resp.:** Na Aula 9 vimos que a quantidade de falhas de página dependerá do algoritmo de substituição de páginas usado. Como na questão não foi especificado o algoritmo, não podemos determinar o número exato de falhas de página que foram geradas. Note que teremos pelo menos seis falhas de página, como veremos a seguir. Duas falhas serão geradas ao acessarmos inicialmente as Páginas 0 e 1, pois somente a Página 3 está na memória. As quatro falhas restantes serão geradas após acessarmos as Páginas 4, 2, 1 e 0 (as duas últimas pela segunda vez), pois o processo terá, em todos os casos, acessado, em sequência, quatro páginas virtuais diferentes. Note também que o primeiro acesso à Página 3 não poderá gerar falhas de página, pois tínhamos duas molduras vazias antes de acessarmos pela primeira vez as Páginas 0 e 1.

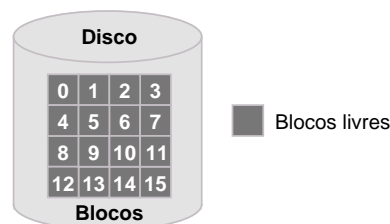
4. (1.0) O que é um conjunto funcional de um processo? Como ele pode ser usado para reduzir as falhas de páginas?

**Resp.:-**O conjunto funcional de um processo são as páginas virtuais

usadas pelo processo em um dado momento da sua execução, isto é, ele pode variar conforme o processo executa no processador. Cada uma das variações do conjunto funcional pode ser vista como uma fase diferente da execução de um processo.

-A quantidade de falhas de página que um processo gera será significativamente reduzida se sempre garantirmos que o conjunto funcional esteja na memória, pois as falhas somente ocorrerão quando o processo entrar em uma nova fase de sua execução. Note que todos os algoritmos de substituição de páginas vistos na Aula 9 tentam estimar os conjuntos funcionais dos processos, mas eles nem sempre terão sucesso em suas estimativas. Isso ocorre porque o conjunto funcional de um processo depende das páginas acessadas durante um dado momento da sua execução, e estes acessos não seguem necessariamente uma regra pré-determinada.

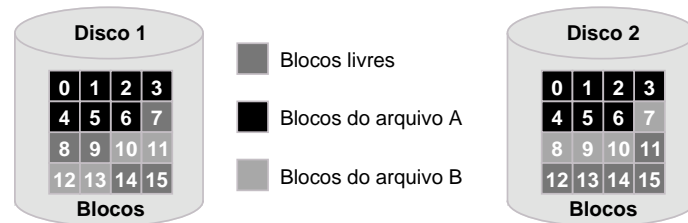
5. (2.0) Suponha que temos um disco com 16 blocos, numerados de 0 até 15, como o da figura a seguir, com inicialmente todos os seus blocos livres. Responda:



- (a) (1.0) Se usarmos a alocação contígua, qual seria o tamanho máximo que um arquivo poderia ter após criarmos dois arquivos com 7 e 4 blocos, e os salvarmos a partir dos blocos 0 e 10 do disco?

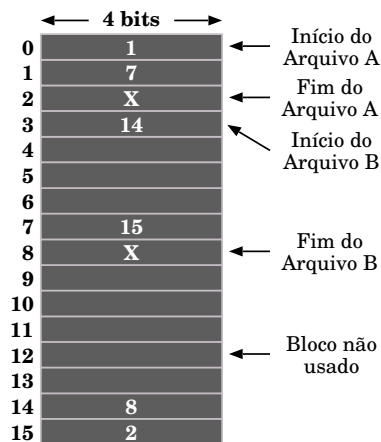
**Resp.:** O tamanho máximo dependerá de o sistema operacional usar ou não a compactação do disco. Se o sistema não usar a compactação, a distribuição dos blocos será como a dada no Disco 1 da figura a seguir. Como na alocação contígua um arquivo deve ser armazenado em blocos consecutivos do disco, então o tamanho do maior arquivo será de 3 blocos (armazenado nos blocos de 7 até 9). Agora, se o sistema usar a compactação, a distribuição

dos blocos será como a dada no Disco 2 da figura a seguir. Neste caso, o tamanho do maior arquivo será de 5 blocos (armazenado nos blocos de 11 até 15).



- (b) (1.0) Suponha que dois arquivos, A e B, foram criados no disco, e que usamos os blocos 0, 1, 7, 15 e 2 para o arquivo A, e os blocos 3, 14 e 8 para o arquivo B. Mostre a tabela obtida se usarmos a alocação por lista encadeada baseada em um índice, supondo que os blocos foram alocados na ordem dada na questão.

**Resp.:** Quando usamos a alocação por lista encadeada baseada em um índice, temos uma tabela com uma entrada cada bloco do disco. A entrada de um bloco conterá, se este não estiver livre ou não for o último bloco lógico do arquivo, o próximo bloco lógico do arquivo. Agora, como o disco possui 16 blocos, então a tabela terá 16 entradas. Como os 5 blocos lógicos do Arquivo A estão armazenados nos blocos 0, 1, 7, 15 e 2, e os 3 blocos lógicos do Arquivo B nos blocos 3, 14 e 8, então a tabela será como a dada a seguir. Nesta tabela, assim como na Aula 11, as entradas dos blocos livres estão vazias, e as entradas dos últimos blocos dos Arquivos A e B estão preenchidas com um “X”.



6. (2.0) Em relação à consistência de um sistema de arquivos, o que ocorrerá se:
- (a) (1.0) Usarmos uma lista encadeada para gerenciar os blocos livres do disco? E se um bloco aparecer duas ou mais vezes nesta lista?

**Resp.:** Se usarmos uma lista encadeada para gerenciar os blocos livres do disco, deverá existir uma e somente uma entrada para cada bloco livre, com o número deste bloco no disco. Logo, poderemos ter dois problemas de consistência: um bloco livre cujo número não está na lista, e um bloco cujo número aparece mais de uma vez na lista. Se o número de um bloco não estiver na lista e o bloco não estiver livre, este bloco não poderá ser usado até que o erro de consistência seja corrigido, e o seu número seja adicionado na lista. Se um bloco aparecer mais de uma vez na lista, este poderá ser alocado a mais de um arquivo, o que poderá causar a perda dos dados deste bloco, dado que os diferentes arquivos possam ser modificados independentemente. Vide próximo item da questão.

- (b) (1.0) Um bloco estiver alocado simultaneamente a mais de um arquivo?

**Resp.:** Se um bloco estiver alocado a mais de um arquivo, quando alterarmos um dos arquivos, poderemos ter perda de dados em to-

dos os outros arquivos. Isso ocorrerá quando, ao alterarmos este arquivo, alterarmos os dados do bloco que estes arquivos possuem em comum. Neste caso, o sistema operacional não poderá recuperar os dados perdidos, mas poderá evitar futuras perdas se associarmos este bloco a um dos arquivos, alocarmos novos blocos para os outros arquivos, e copiarmos o conteúdo deste bloco nos blocos novos.