



## Lista de Exercícios - Sistemas Operacionais

### Aula 12: *Sistemas de Arquivos - Parte 2*

**Professores:** Felipe M. G. França e Valmir C. Barbosa

**Assistente:** Alexandre H. L. Porto

1. Um disco tem 4000 cilindros, cada um com 8 trilhas de 512 blocos. Uma busca leva 1 milissegundo por cilindro movido. Se nenhuma tentativa for feita para colocar os blocos de um arquivo próximos uns aos outros, dois blocos que sejam logicamente consecutivos (isto é, um segue o outro no arquivo) irão requerer um tempo médio de busca de 5 milissegundos. Se, entretanto, o sistema operacional fizer uma tentativa de agrupar os blocos relacionados, a distância média entre blocos poderá ser reduzida a 2 cilindros e o tempo de busca reduzido para 100 microssegundos. Quanto tempo leva para ler um arquivo de 100 blocos em ambos os casos, se a latência rotacional é de 10 milissegundos e o tempo de transferência é de 20 microssegundos por bloco?

**Resp.:** Sempre que transferimos um bloco do disco, perdemos um certo tempo para buscá-lo no disco, mais um outro tempo para esperar que ele passe por uma das cabeças magnéticas do disco, e finalmente mais o tempo necessário para transferi-lo. Os dois últimos tempos são independentes da posição do bloco no disco. No caso de um bloco aleatório, o tempo para que o bloco passe por uma cabeça do disco será de 5ms em média (a metade do tempo de latência rotacional do disco), e o tempo de transferência será de 20 microssegundos, ou seja, 0.02ms. Já o tempo médio de busca dependerá de o sistema operacional colocar ou não os blocos relacionados próximos entre si no disco. Se o sistema operacional não colocar estes blocos próximos uns dos outros, o tempo médio de busca de um bloco aleatório será de 5ms. Logo, neste caso, o tempo total decorrido até lermos um bloco será de 10.02ms, e com

isso, o tempo para ler 100 blocos será de 1002ms. Agora, se o sistema colocar os blocos relacionados próximos uns dos outros, o tempo médio de busca será reduzido para 100 microssegundos, isto é, 0.1ms. Logo, o tempo total para ler um bloco será agora de 5.12ms, e com isso, o tempo para ler 100 blocos será de 512ms.

**Obs.:** A resposta acima vale para o caso em que o tempo de latência rotacional é o maior tempo de espera até que um bloco alcance uma cabeça. Entretanto, caso alguma resposta tenha sido baseada em tomar 10ms como o tempo médio de latência rotacional, ela será também correta. Nesse caso, seriam necessários 1502ms para ler os 100 blocos se o sistema operacional não colocasse os blocos relacionados próximos uns aos outros, ou 1012ms se o fizesse.

2. Duas alunas de ciência da computação, Neusa e Severina, estão tendo uma discussão sobre nós-i. Neusa sustenta que as memórias ficaram tão grandes e tão baratas que, quando um arquivo é aberto, é mais simples e mais rápido simplesmente colocar uma nova cópia do nó-i na tabela de nós-i, em vez de pesquisar a tabela inteira para ver se ele já está lá. Severina discorda. Quem está certa?

**Resp.:** Serevira está correta, pois manter mais de uma cópia do nó-i na tabela de nós-i pode gerar inconsistências nos atributos do arquivo ou no sistema de arquivos, a não ser que todas estas cópias possam somente ser lidas da tabela. Se todas as cópias puderem ser livremente alteradas, quando salvarmos os nós-i no disco, somente as alterações da última cópia serão salvas no disco, pois para cada cópia salva, com exceção desta última, suas alterações serão perdidas ao salvarmos a próxima cópia no disco. Note que não podemos atualizar todas as cópias, pois isso significaria fazer pesquisas na tabela de nós-i. No caso de inconsistências nos atributos do arquivo, os atributos alterados pelas outras cópias do nó-i serão perdidos. Já o problema de inconsistência no sistema de arquivos ocorrerá devido a perda de nós-i e/ou blocos do disco, que poderão nunca mais ser disponibilizados para o uso. Por exemplo, poderemos perder um nó-i e blocos do disco se removermos uma das referências a ele e se esta informação não for salva no disco. Isso ocorrerá porque ao removermos as referências restantes a este nó-i, ele e os blocos usados por ele não serão liberados e somente poderão ser usados novamente após corrigirmos, se possível, a inconsistência no sistema de arquivos.

3. Suponha que o computador tenha um disco com 65536 blocos de 32KB. Suponha ainda que o sistema operacional use o mecanismo de proteção por lista de controle de acesso ao gerenciar o acesso aos objetos do sistema pelos processos. Responda:

- (a) Em relação ao gerenciamento dos blocos livres, qual das técnicas vistas na Aula 12 gastaria menos espaço no disco para armazenar a lista de blocos livres do disco se  $x$  blocos estivessem livres?

**Resp.:** Se usarmos a lista encadeada para armazenar os  $x$  blocos livres, precisaremos usar  $2x$  bytes do disco, pois existem 65536 blocos no disco, o que significa que cada endereço de um bloco precisará de 2 bytes para ser representado. Agora, se usarmos um mapa de bits, sempre precisaremos de 65536 bits, que ocuparão 8192 bytes do disco, pois cada bloco do disco deve sempre ser representado por 1 bit no mapa de bits. Logo, se  $2x < 8192$ , ou seja, se  $x < 4096$ , então a lista encadeada gastará menos espaço do disco. Agora, se  $x > 4096$ , o mapa de bits é que gastará menos espaço do disco. Finalmente, se  $x = 4096$ , tanto a lista encadeada como o mapa de bits gastarão 8192 bytes do disco.

- (b) Quais seriam os significados das permissões de cada um dos objetos dados na figura a seguir?

Objeto	Lista com os domínios e as operações
Arquivo0	(hal, *, R-X)
Arquivo1	(vod, land, RWX)
Arquivo2	(hal, *, RW-), (lam, staff, RW-), (felipe, *, R-X)
Arquivo3	(*, cederj, --X)
Arquivo4	(mp, *, ---), (*, cederj, R--)

**Resp.:** A seguir descrevemos o significado das permissões para cada um dos objetos dados na figura, lembrando que um “R” significa que o objeto pode ser lido, que um “W” significa que o objeto pode ser alterado, e que um “X” significa que o objeto pode ser executado:

- Objeto Arquivo0: A única entrada da lista significa que somente o usuário “hal”, pertencente a qualquer grupo (indicado por um “\*”) pode ler ou executar o objeto.
- Objeto Arquivo1: A única entrada da lista significa que somente o usuário “vod” do grupo “land” pode ler, alterar ou executar o objeto.

- Objeto Arquivo2: A primeira entrada da lista significa que o usuário “hal” de qualquer grupo (indicado por um “\*”) pode ler ou alterar o objeto. A segunda entrada significa que o usuário “lam” do grupo “staff” pode também ler ou alterar o objeto. Finalmente, a terceira entrada significa que o usuário “felipe” de qualquer grupo (indicado por um “\*”), pode ler ou executar o objeto.
- Objeto Arquivo3: A única entrada significa que somente qualquer usuário (indicado por um “\*”) do grupo “cederj” pode executar o objeto.
- Objeto Arquivo4: Estas duas permissões são interessantes: a primeira entrada significa que o acesso ao objeto é negado (indicado por um “---”) ao usuário “rnp” de qualquer grupo (indicado por um “\*”). Logo, a segunda entrada significa que somente qualquer usuário (indicado por um “\*”) do grupo “cederj”, com exceção do usuário “rnp” (devido à primeira entrada), pode ler o objeto.

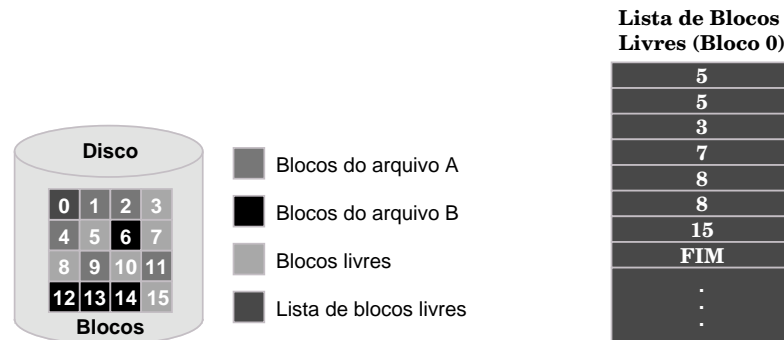
(c) Suponha que um bloco livre seja incorretamente alocado aos objetos Arquivo0 e Arquivo3 da figura do item anterior, após ambos aumentarem o seu tamanho em um bloco. Isso gera uma inconsistência no sistema de arquivos? Como ela afeta os dados desses arquivos?

**Resp.:** -Sim, isso geraria uma inconsistência, pois um mesmo bloco estaria associado a dois objetos (Arquivo0 e Arquivo1) e, quando o sistema de arquivos está consistente, cada um dos blocos do disco está livre ou associado a somente um dos objetos armazenados no sistema de arquivos.

-Os dados destes arquivos (note que os objetos são arquivos) seriam afetados como descrito a seguir. Se alterássemos os dados armazenados no bloco compartilhado em um dos arquivos, parte dos dados do outro arquivo, também armazenados neste bloco, seriam indiretamente alterados. Logo, a alteração dos dados de um dos arquivos pode alterar parte dos dados do outro arquivo, comprometendo o conteúdo deste arquivo.

4. Considere um computador com um disco de 16 blocos e a alocação representada pela figura a seguir. Supondo que o bloco 0 seja reservado e armazene a lista de blocos livres, e que os arquivos A e B sejam os

únicos arquivos armazenados no disco, responda às seguintes perguntas:



- (a) O sistema de arquivos está consistente? Justifique a sua resposta.

**Resp.:** Não, pois temos dois tipos de inconsistências no sistema de arquivos dado na figura. A primeira delas é que os blocos 5 e 8 estão duplicados na lista de blocos livres. A segunda é que o bloco 10 é um bloco ausente, porque não está na lista de blocos livres e nem está alocado aos arquivos A e B. Note que o bloco 0 não pode estar nesta lista, pois ele está reservado para armazená-la.

- (b) Suponha que os blocos sejam sempre escolhidos na ordem definida, de cima para baixo, na lista de blocos livres dada na figura. O que ocorrerá se o arquivo A aumentar o seu tamanho em um bloco, e depois o arquivo B também aumentar o seu tamanho em um bloco? Justifique a sua resposta.

**Resp.:** Como o bloco 5 está duplicado na lista de blocos livres, e como ele está ocupando as duas primeiras posições desta lista, então ele será alocado, ao mesmo tempo, aos arquivos A e B. Com isso, o conteúdo do arquivo A será comprometido após o arquivo B salvar os seus dados no bloco 5, porque os dados salvos pelo arquivo A neste bloco serão perdidos. Além disso, se futuramente alterarmos um dos arquivos e salvarmos novos dados neste bloco, o conteúdo do outro arquivo também será comprometido. Logo, pelos problemas descritos anteriormente, o que ocorrerá é que não será mais confiável armazenar dados nos arquivos A e B.