Sistema de Arquivos

Prof. Edwar Saliba Júnior
Março de 2007

Objetivos

- **Facilitar o acesso dos usuários ao conteúdo dos arquivos;**
- **Prover uma forma uniforme de manipulação de arquivos, independente dos dispositivos de armazenamento;**
- **Aspectos a serem considerados: identificação, organização, compartilhamento, métodos de acesso, proteção e operações de I/O;**
- **Atributos: nome, tipo, posição, tamanho, proteção, hora, data e dono.**

Arquivo

- **Conjunto de instruções ou dados relacionados:**
  - **Instruções: .exe, .com, .bat;**
  - **Dados: .doc, .dat, .txt ...**
- **Pode ser armazenado em diferentes dispositivos (discos magnéticos, discos ópticos, fitas magnéticas...);**
- **É identificado por seu nome;**
- **As regras para nomenclatura de arquivos, variam de acordo com o sistema de arquivos:**
  - ***Case sensitive* X *case insensitive*: distinção entre maiúsculas e minúsculas:**
    - ***Sensitive*: Unix e seus derivados (Ex.: GNU/Linux e FreeBSD);**
    - ***Insensitive*: Windows;**
  - **Tamanho máximo do nome.**

Nome de Arquivo no Windows

"O Windows geralmente limita os nomes de arquivos em 260 caracteres. Mas o nome do arquivo, na verdade, deve ser mais curto que isso, já que o caminho completo (como C:\Arquivos de Programa\nome_do_arquivo.txt) está incluído nessa contagem de caracteres."

Fonte: <http://windows.microsoft.com/pt-BR/windows-vista/File-names-and-file-name-extensions-frequently-asked-questions> Acesso em: 19 abr. 2011.

Métodos de Acesso

- **Sequencial:**
  - **Um processo só pode ler os bytes de um arquivo na ordem em que eles estão armazenados;**
  - **Primeiros sistemas operacionais;**
  - **Arquivos sequenciais podem ser “rebobinados” para serem lidos novamente;**
  - **Utilizado em fitas magnéticas.**

Métodos de Acesso

- **Direto:**
  - **Surgiu com o aparecimento dos discos magnéticos;**
  - **Possibilidade de acesso aos *bytes* fora de sua ordem de armazenamento;**
  - **Fundamentais para algumas aplicações. Ex: bancos de dados que podem armazenar grande quantidade de dados em um único arquivo físico;**
  - **Utilizado nos sistemas operacionais atuais.**

Operações com Arquivos

- **Interface que o sistema operacional disponibiliza para que o usuário utilize os arquivos de maneira uniforme e simples (*system calls*):**
  - ***CREATE*;**
  - ***OPEN*;**
  - ***READ*;**
  - ***WRITE*;**
  - ***CLOSE*;**
  - ***DELETE*.**

Diretórios

- **Organização lógica dos arquivos;**
- **Cada diretório de dispositivo possui informações sobre os arquivos nele contidos (nome, posição, tamanho, tipo);**

- **Nível Único:**
  - **Todos os arquivos em um mesmo diretório;**
  - **Não pode haver mais de um arquivo com mesmo nome no mesmo dispositivo;**
  - **Ex: CP/M (Sistema Operacinal do processador Z-80).**

Diretórios

- **Diretório de dois níveis:**
  - **Cada usuário possui um diretório;**
  - **No momento em que o usuário efetua *logon*, o sistema busca num diretório de arquivos mestre (MFD) e identifica em qual dos diretórios de arquivos de usuário (UFD) ele deverá trabalhar;**
  - **Pode existir arquivos com mesmo nome, desde que para usuários distintos;**
  - **Um programa específico gerencia a criação de novos usuários;**
  - **Soluciona parcialmente a colisão de nomes mas não permite ao usuário uma organização adequada.**

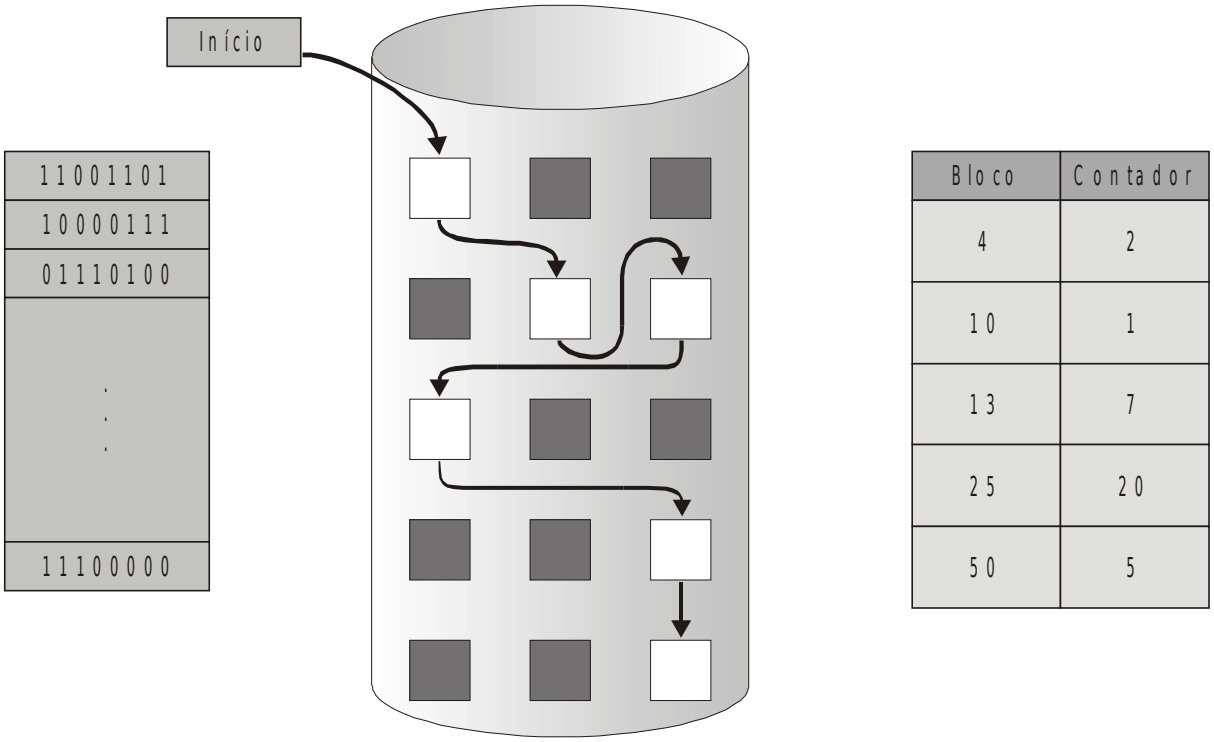
Diretórios

- **Múltiplos níveis (árvore):**
  - **Adotado pela maioria dos sistemas operacionais;**
  - **Cada usuário pode criar diversos níveis de diretórios (subdiretórios);**
  - **O arquivo passa a ser especificado por um *path* (caminho) e o seu nome. Ex: /guilherme/downloads/teste.txt**
  - **Em alguns sistemas, como o MS-DOS, um diretório só pode ser excluído se estiver vazio.**

Gerência do espaço livre em disco

- **Mapa de bits:**
  - **Registro de todos os blocos do disco, indicando se estão livres ou não;**
    - **Gasto de memória;**
- **Lista encadeada de blocos livres:**
  - **Cada bloco livre possui um apontador para o próximo bloco livre:**
    - **Espaço gasto com o apontador;**
    - **A busca de espaço livre sempre precisa percorrer toda a lista;**
- **Tabela de blocos livres:**
  - **Mantém uma tabela onde cada registro contém o endereço do primeiro bloco e o número de blocos livres contíguos que seguem.**

Gerência do espaço livre em disco



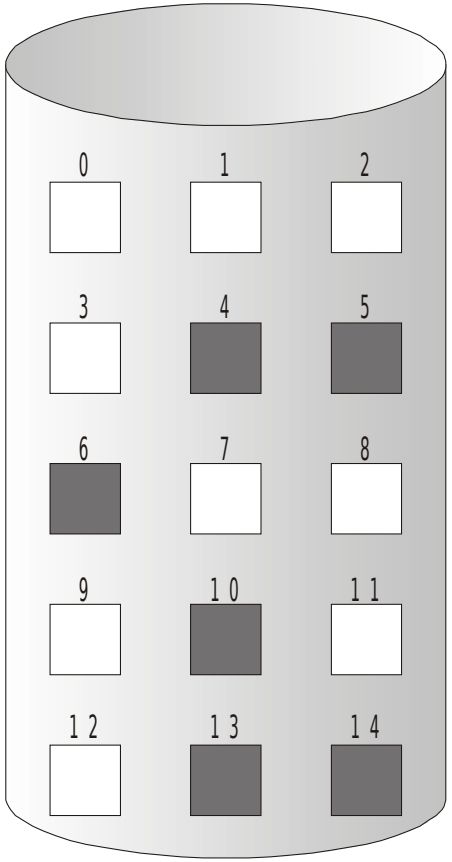
Bloco	Contador
4	2
10	1
13	7
25	20
50	5

(a) Mapa de bits (b) Lista encadeada © Tabela de blocos livres

Alocação de espaço em disco

- **Alocação contígua:**
  - **Um arquivo é armazenado em blocos sequenciais;**
  - **A localização do arquivo é feita com o endereço do primeiro bloco e a quantidade de blocos que ele ocupa;**
  - **Vantagem: o acesso a arquivos é simples e rápido (não exige movimento da cabeça do disco);**
  - **Desvantagem: a criação e eliminação frequente de arquivos causa fragmentação dos espaços livres, independente da estratégia de alocação utilizada (*best-fit*, *first-fit*, *worst-fit*);**
  - **Solução: desfragmentação do disco;**
  - **Outro problema: e se o arquivo precisar aumentar de tamanho?**
  - **Ex: VM/CMS da IBM;**

Alocação de espaço em disco



Arquivo	Bloco	Extensão
A.TXT	4	3
B.TXT	10	1
C.TXT	13	2

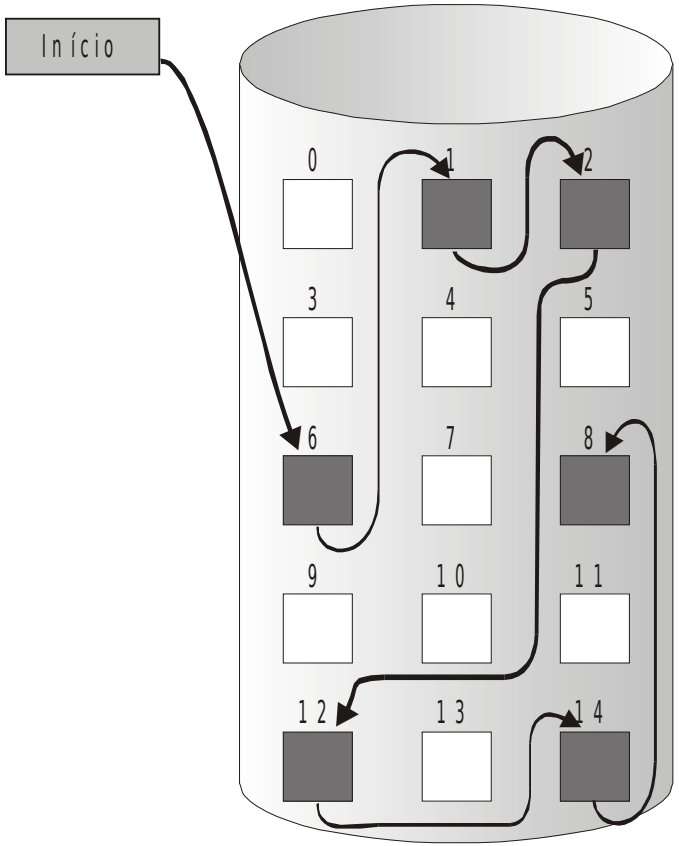
Alocação de espaço em disco

- **Alocação encadeada:**
  - **Um arquivo pode ser organizado em blocos não sequenciais, ligados logicamente por ponteiros;**
  - **A fragmentação dos espaços livres não é problema;**
  - **Fragmentação de arquivos: arquivos divididos em blocos distantes fisicamente -> maior tempo de acesso;**
  - **A desfragmentação tenta unir o arquivo em blocos menos distantes;**
  - **Gasto de espaços nos blocos para os ponteiros.**

Alocação de espaço em disco

- **Para se chegar a uma informação no final do arquivo, é preciso percorrer todos os blocos;**
- **Alguns sistemas como o MS-DOS e algumas versões do Windows implementam a alocação encadeada com uma tabela de alocação e arquivos (FAT - *File Allocation Table*);**
- **A FAT possibilita o acesso direto a partes no final do arquivo.**

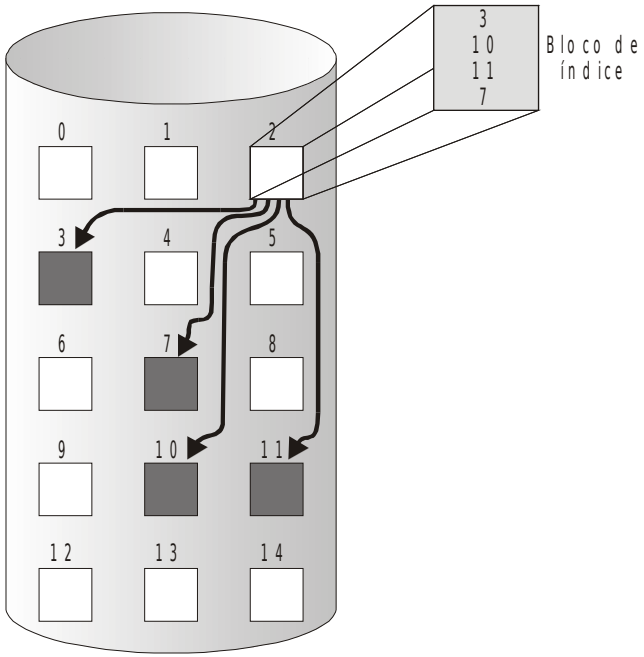
Alocação de espaço em disco



Arquivo	Bloco
A.TXT	6
...	...
...	...
...	...
...	...

Alocação de espaço em disco

- **Alocação indexada:**
  - **Os ponteiros são centralizados no bloco de índice;**



Proteção

- **Permitir o compartilhamento de arquivos entre diferentes usuários somente quando desejado;**
- **Concessão ou não de leitura, escrita, execução e eliminação.**

Mecanismos de Proteção

- **Senha de acesso:**
  - **Cada arquivo possui uma senha que libera todas as operações sobre ele;**
- **Grupos de usuários:**
  - **Usado no UNIX e GNU/Linux, dentre outros;**
  - **Três grupos: dono, grupo e todos;**
  - **Para cada grupo, três permissões: *Read*, *Write*, *eXecute*;**
- **Lista de acesso:**
  - **Para cada arquivo, tem-se uma lista de usuários e os tipos de acesso permitidos;**
  - **Em alguns casos, o tamanho da lista pode ser grande;**
  - **A verificação da permissão pode ser lenta.**

Compressão de Dados

- Compressão de dados é uma técnica utilizada para economizar espaço nos arquivos;
- Os três métodos de compressão utilizados nas bases de dados são:
  - caracteres repetidos,
  - termos repetidos e
  - compressão de *front end*.

Compressão de Dados

- Os registros com caracteres repetidos podem ser abreviados. Por exemplo, dados em um campo de tamanho fixo podem incluir um pequeno nome seguido de vários caracteres vazios;
- Valores repetidos podem ser trocados por representações menores sem perda de conteúdo. Exemplo:
  - Digamos que a cadeia de caracteres ADAMS tenha o seguinte aspecto quando é armazenada em um campo de 15 caracteres (v representa um caractere vazio): ADAMSvvvvvvvvvv
- Ao aplicarmos a compressão, ela toma o seguinte aspecto:
  - ADAMSv10
- Números com muitos zeros também podem ser encurtados:
  - 300000000
- Após a aplicação da compressão:
  - 3#8

Compressão de Dados

- Termos repetidos também podem ser comprimidos;
- Um método possível é a utilização de símbolos para representar as palavras na base de dados;
- Palavras com diversos caracteres que se repetem num texto/arquivo, podem ser substituídas por um único caractere;
- O sistema deverá ser inteligente para distinguir dados comprimidos e não comprimidos.

Compressão de Dados

- A compressão *Front End* é utilizada nos sistemas de gerenciamento de base de dados para compressão de índices;
- Exemplo:
  - Vejamos a tabela no próximo *slide*.

Compressão de Dados

Lista original	Lista comprimida
Smith, Betty	Smith, Betty
Smith, Gino	7Gino
Smith, Donald	7Donald
Smithberger, John	5berger, John
Smithbren, Ali	6ren, Ali
Smithco, Rachel	5co, Rachel
Smither, Kevin	5er, Kevin
Smithers, Renny	7s, Renny
Snyder, Katherine	1nyder, Katherine

Compressão de Dados

- Em qualquer tipo de compressão de dados há sempre uma compensação:
  - ganha-se espaço de armazenagem
  - mas perde-se tempo de processamento.

Implementação de Caches

- O acesso a disco é bastante lento se comparado ao acesso à memória principal, devido à arquitetura dos discos magnéticos;
- Este é o principal motivo das operações de E/S com discos serem um problema para o desempenho do sistema operacional.

Implementação de *Caches*

- Com o objetivo de minimizar este o problema a maioria dos sistemas implementam um técnica denominada: *buffer cache*;
- O sistema operacional reserva uma área da memória para que se tornem disponíveis *cache* utilizados em operações de acesso ao disco;
- Quando uma operação é realizada, seja leitura ou gravação, o sistema verifica se a informação desejada se encontra no *buffer cache* antes de realizar a operação;
- Em caso positivo, não é necessário o acesso ao disco.
- Caso o bloco requisitado não se encontre no *cache*, a operação de E/S é realizada e o *cache* é atualizado.

Implementação de *Caches*

- Como existe uma limitação no tamanho do *cache*, cada sistema adota políticas para substituição de blocos como a FIFO (*first in*, *first out*) ou a LRU (*Least Recently Used*).

Implementação de *Caches*

- Apesar de esta implementação melhorar o desempenho do sistema, aspectos de segurança devem ser levados em consideração;
- No caso de blocos de dados permanecerem por um longo período de tempo na memória principal, a ocorrência de problemas de energia pode ocasionar:
  - a perda de tarefas já realizadas e consideradas já salvas em disco.

Implementação de *Caches*

- Existem duas maneiras distintas de tratar este problema:
  - pode ser implementada uma rotina que executa periodicamente em um intervalo de tempo, atualizando em disco todos os blocos modificados do *cache*;
  - toda vez que um bloco do *cache* for modificado, que seja realizada imediatamente uma atualização no disco (*write-through caches*).

Bibliografia

- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais,** 3ª Ed., Rio de Janeiro: LTC Editora, 2002.
- SILVA, Guilherme Baião S. *Slides* da disciplina de Sistemas Operacionais de Arquitetura Fechada. Faculdade INED, 2005.