

Sistema de Arquivos

Ciclo 5 – AT1

**Prof. Hermes Senger /
Hélio Crestana Guardia**

Referência:

Deitel – Cap. 13

Nota

O presente material foi elaborado com base no material didático do livro
*Sistemas Operacionais, 3ª edição, de H.M.Deitel, P.J. Deitel, D.R.
Choffnes, Editora Pearson Prentice Hall, São Paulo, 2005,*
disponibilizado pela editora.

Objetivos

■ Este ciclo apresenta:

- O papel dos sistemas de arquivos.
- Operações que podem ser realizadas com arquivos e diretórios.
- A ocupação das áreas do disco (organização do espaço)
- A organização do espaço livre de um disco.
- O controle de acesso aos dados
- Mecanismos de cópia de segurança (backup), recuperação e integridade de um sistema de arquivos.

Introdução

■ Arquivos

- **Coleção nomeada** de dados manipulada como uma **unidade**.
 - Ex: cadastro dos funcionários de uma empresa, etc
 - Residem em dispositivos de armazenamento secundário.
 - Ex: Disco, fita, disco ótico, etc
- Os sistemas operacionais fornecem uma **interface** que facilita a **manipulação** dos arquivos dos usuários
 - Os sistemas de arquivo podem proteger esses dados contra corrupção ou perda decorrente de acidentes

Arquivos

- **Itens de dados individuais dentro de um arquivo podem ser manipulados por operações como:**
 - Ler
 - Escrever (gravar)
 - Atualizar
 - Inserir
 - Apagar
- **Os arquivos podem ser caracterizados por atributos como:**
 - Localização
 - Acessibilidade
 - Tipo
 - Volatilidade
 - Atividade
- **Os arquivos podem conter um ou mais registros**

Sistemas de arquivo

■ Sistema de arquivo

- É **parte integrante** do Sistema Operacional
- **Organiza** arquivos e gerenciam o acesso aos dados.
- Provê **gerenciamento** dos **arquivos**
- Provê gerenciamento do **armazenamento** auxiliar
- Provê mecanismos de integridade do arquivo
- Implementa métodos de acesso
- Gerencia o espaço de armazenamento secundário, em especial o armazenamento em disco.

Sistemas de arquivo

■ Características do sistema de arquivo:

- Devem exibir **independência de dispositivos**:
 - Os usuários devem poder referir-se a seus arquivos por **nomes simbólicos** em vez de ter de utilizar nomes de dispositivos **físicos**.
- Devem também oferecer o recurso de **cópia de segurança** e recuperação para evitar perdas acidentais ou danos maliciosos às informações
- Podem ainda oferecer o recurso de **criptografia** e **decriptação** para tornar as informações úteis apenas ao público a que se destina

Diretórios

■ **Diretórios:**

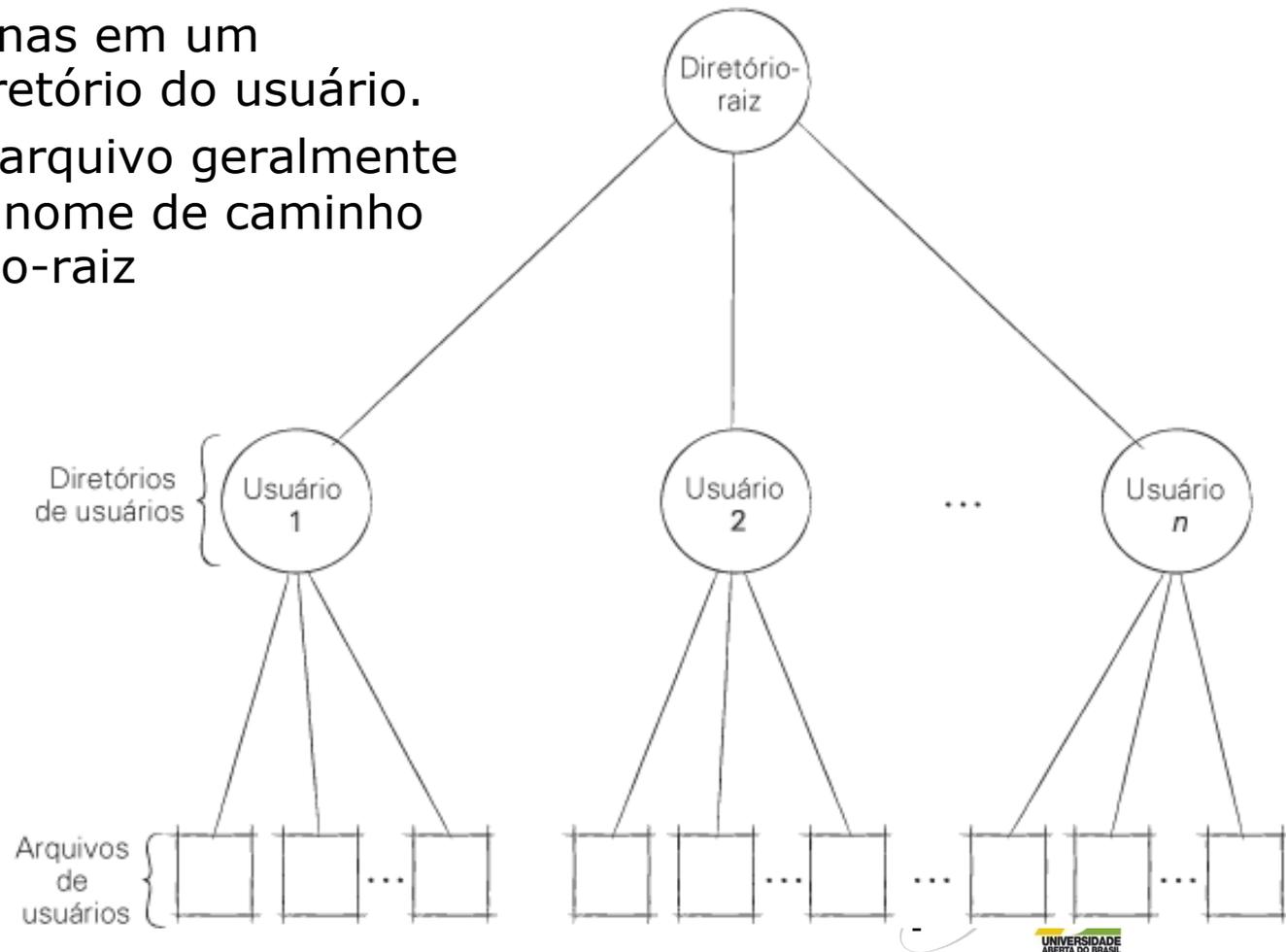
- São arquivos que contêm o **nome** e o **local** de outros arquivos
- Permitem organizar e localizar arquivos rapidamente.

■ **A entrada de diretório armazena informações como:**

- Nome do arquivo
- Local
- Tamanho
- Tipo
- Horário de acesso
- Horário de modificação e criação

Sistema de arquivo hierárquico

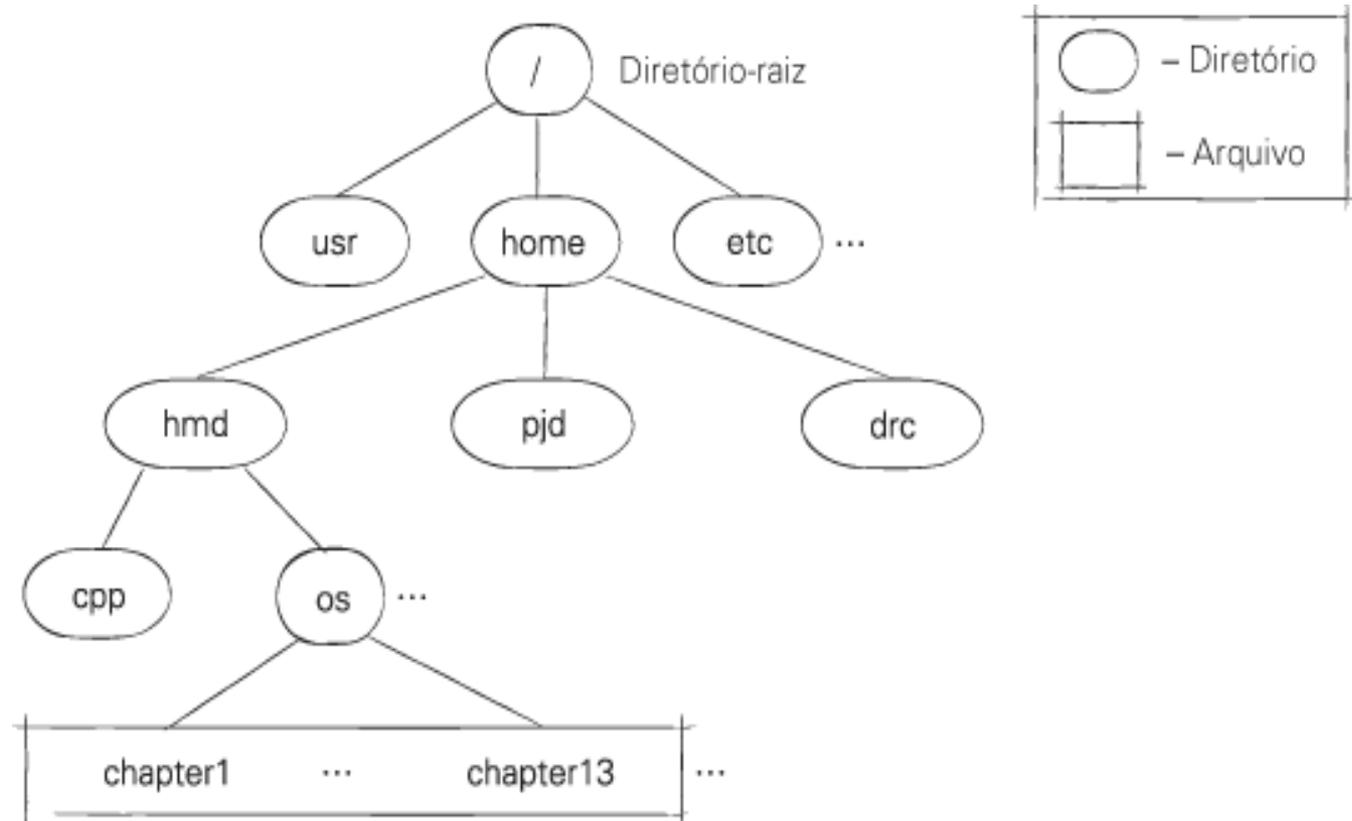
- O **diretório-raiz** aponta para os vários diretórios, cada um dos quais contém uma entrada para cada um de seus arquivos.
- O **nome** do arquivo tem de ser **exclusivo**, apenas em um determinado diretório do usuário.
- O nome de um arquivo geralmente é formado pelo nome de caminho desde o diretório-raiz até o arquivo.



Diretórios

■ Diretório de trabalho

- Simplifica a navegação usando nomes de caminho.
- Permite que os usuários especifiquem um nome de caminho que não comece no diretório-raiz (isto é, um caminho relativo).
- Caminho absoluto (isto é, o caminho que começa na raiz) = diretório de trabalho + caminho relativo.



Metadados

- **Metadados** (informações sobre os dados)
 - Protegem a integridade do sistema de arquivo.
 - Normalmente não podem ser modificados pelos usuários.
- **Superbloco**: Vários sistemas de arquivo criam um superbloco
 - Armazena **informações críticas** que protegem a integridade do sistema de arquivo. Ex.:
 - O **identificador** de sistema de arquivo.
 - A **localização** dos **blocos livres** do disco ou dispositivo de armazenamento.
 - Normalmente os sistemas de arquivo distribuem **cópias redundantes** do superbloco pelo dispositivo de armazenamento
 - Isso diminui o **risco** de **perda** de dados

Operação:

Abertura de arquivos

- A operação de **abertura** de um arquivo retorna um **descriptor de arquivo**.
 - Um **índice** (número) inteiro não negativo que aponta para dentro da **tabela de arquivos** abertos.
- Após a abertura, todo acesso ao arquivo será feito através do descriptor de arquivo.

■ Exemplo:

```
FILE *fd;
char buffer[11];

fd = fopen("arq_dados.txt", "r");

if (fd == NULL) {
    // Erro, arquivo não encontrado
} else {
    // Lê dados (10 bytes) e fecha arquivo
    fread(buffer, sizeof(char), 10, fd);
    fclose(fd);
}
```

Operação: Montagem

- Operação de montagem cria uma **associação** entre um Sistema de Arquivos e um diretório:
- Cria **referência** a **vários** sistemas de arquivo em **um único espaço de nomes**, para que possam ser referenciados a partir de um único diretório-raiz.
- O **ponto de montagem** é um diretório no sistema de arquivo nativo
 - Ele aponta para a **raiz** do sistema de arquivo montado.
- As **tabelas de montagem**:
 - Usadas pelos sistemas de arquivo para gerenciar os diretórios montados
 - Contém informações sobre **localização** dos pontos de montagem + **dispositivos** para os quais apontam.
- Os usuários podem criar **ligações flexíveis (*symbolic links*)** para sistemas de arquivo montados, mas não para **ligações estritas (*hard links*)** entre sistemas de arquivo.

- Exemplo:

```
$ mount
```

```
/dev/sdb2 on / type ext3 (rw)
```

```
/dev/sdb3 on /home type ext3 (rw)
```

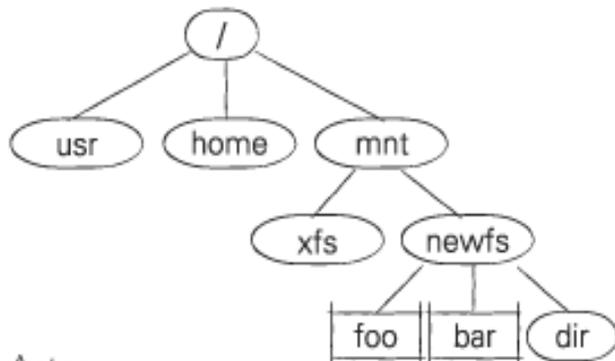
```
/dev/sdb4 on /tmp type ext3 (rw)
```

```
/dev/sdb5 on /var type ext3 (rw)
```

```
nfs_server:/home/helio on /home/helio type nfs (rw,soft,bg,intr)
```

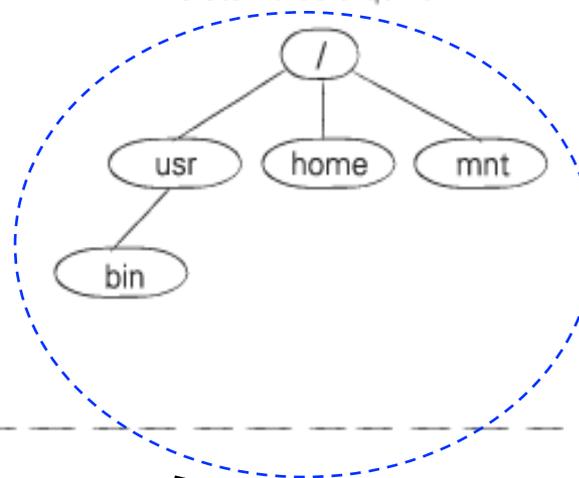
Operação: Montagem

Sistema de arquivo A

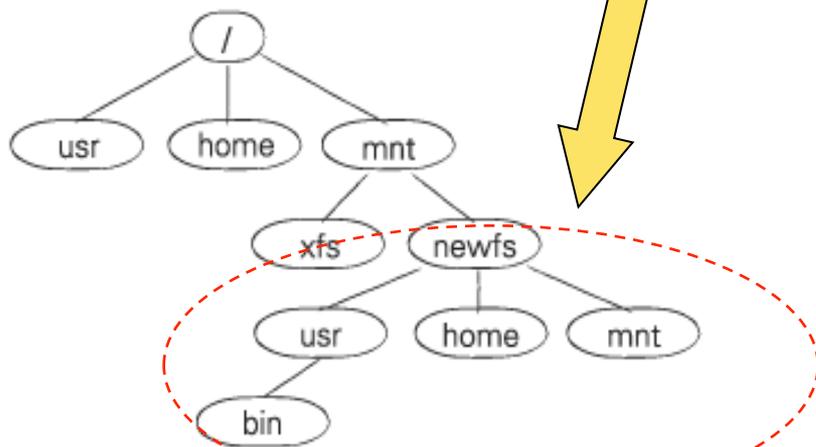


Antes

Sistema de arquivo B



Sistema de arquivo B montado no diretório /mnt/newfs no sistema de arquivo A



Depois

Alocação contígua de arquivos

■ Alocação contígua

- Coloca os dados dos arquivos em endereços contíguos no dispositivo de armazenamento.
- **Vantagem**
 - Registros lógicos sucessivos em geral estão fisicamente adjacentes uns aos outros.

■ Desvantagens

- **Fragmentação externa.**
- Desempenho ruim se os arquivos aumentarem ou diminuïrem ao longo do tempo.:

Se um arquivo exceder o tamanho originalmente especificado e não houver nenhum bloco contíguo disponível, deverá ser transferido para uma nova área de tamanho adequado, o que exige operações adicionais de E/S.

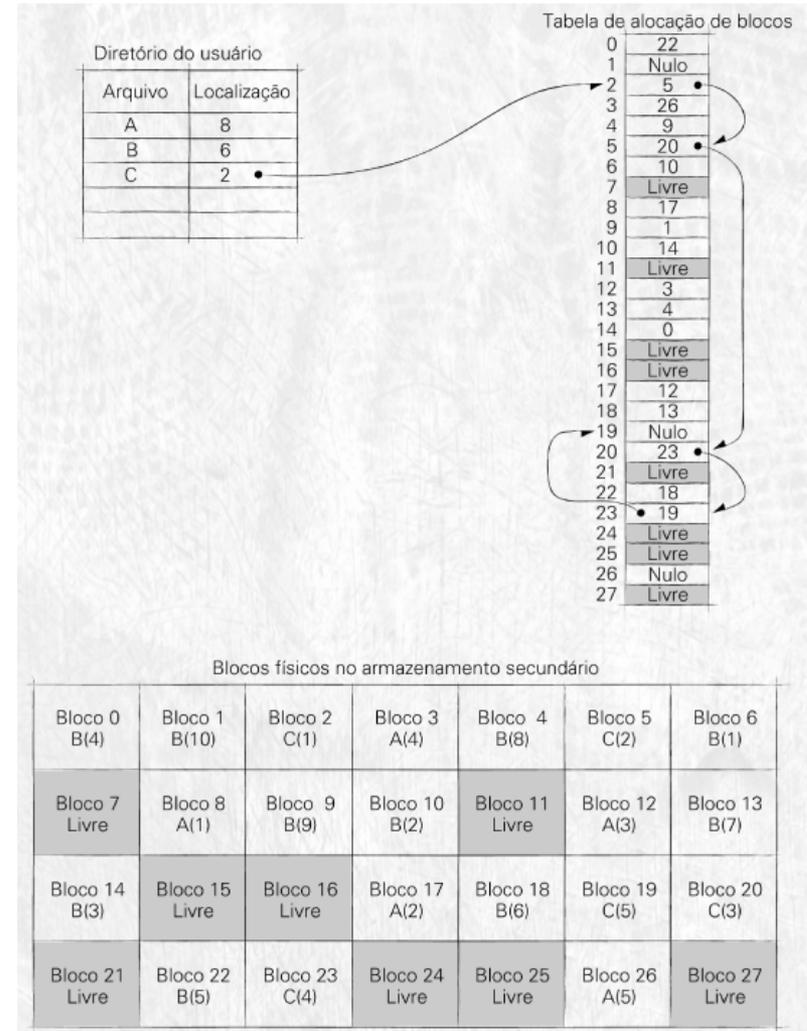
Alocação de espaços de arquivos

- **Discos e unidades de armazenamento são dispositivos de bloco:** leem e escrevem blocos de dados, e não byte-a-byte
- **Alocação de espaços nos discos (sistemas de arquivo) ocorre em blocos**
- **Tamanhos dos blocos definidos pelo administrador do sistema na criação do sistema de arquivos**
- **Tamanhos típicos são múltiplos de 512 bytes:**
 - 1024, 2048, 4096
- **Blocos associados a um arquivo devem ser indicados numa estrutura de dados (arquivo!)**

Alocação de arquivo tabular não contígua

■ Alocação de arquivo tabular não contígua

- Usa **tabelas** que armazenam **ponteiros para blocos** de arquivo.
 - Permite buscas mais rápidas do que na alocação por lista encadeada.
- Entradas de diretório indicam o primeiro bloco de um arquivo.
- **Tabela de alocação:**
 - contém informações sobre o **encadeamento** dos blocos
 - Mas não sobre a sua localização física
 - Reside em memória principal depois que o arquivo é "aberto"
 - A localização física é armazenada em outra tabela

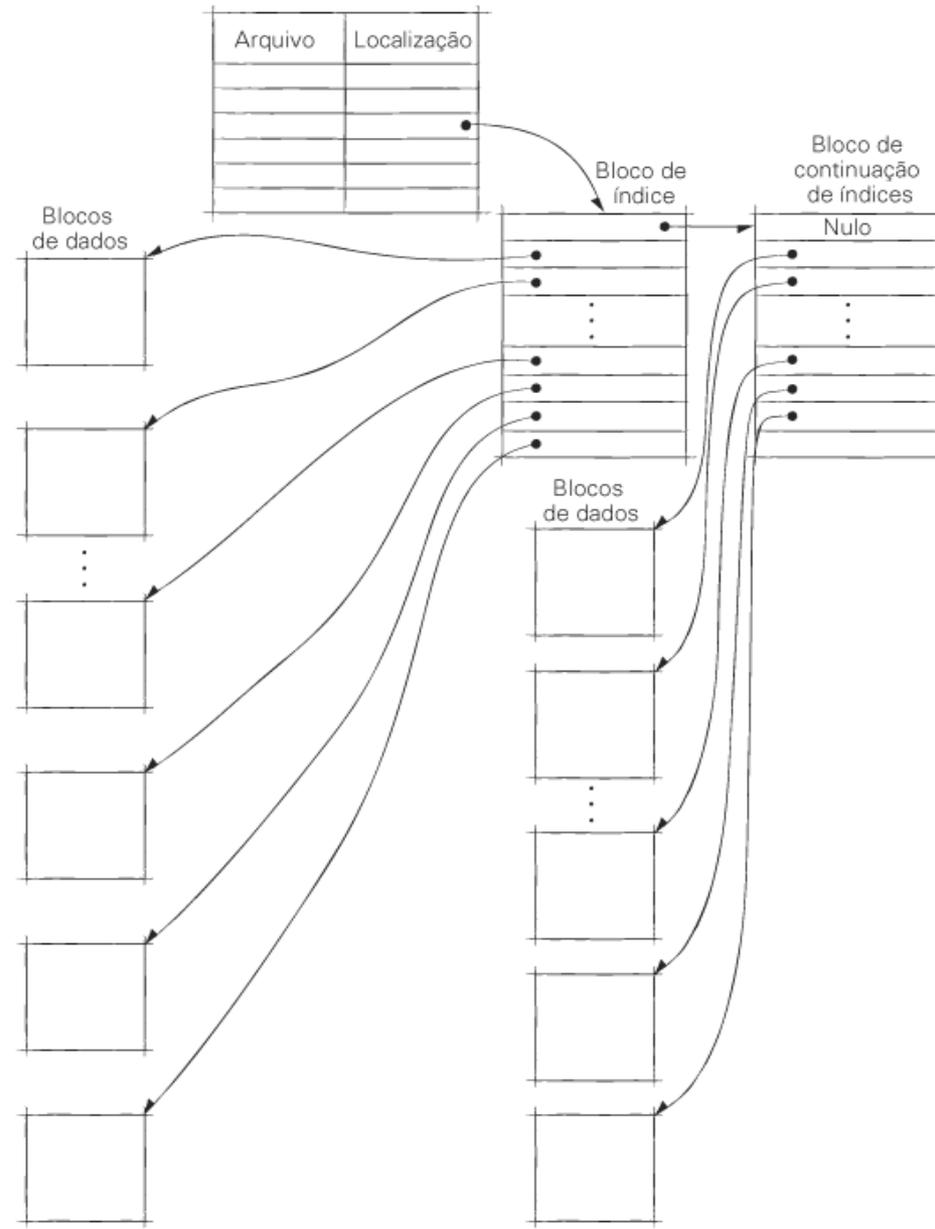


Alocação de arquivo tabular não contígua

- **Quando um dispositivo de armazenamento for muito grande (contém muitos blocos):**
 - A tabela de alocação de blocos pode ficar muito grande
 - Pode haver fragmentação da tabela.
 - Isso diminui o desempenho do sistema de arquivo.
- **Exemplo de implementação**
 - Sistema de arquivos **FAT** da Microsoft.

Alocação de arquivos indexada

- **Alocação de arquivos não contígua indexada:**
 - Todo arquivo possui um bloco (ou vários blocos) de índice.
- **Bloco de índice:** contém uma lista de ponteiros que apontam para os blocos de dados do arquivo.
 - A entrada do diretório do arquivo aponta para seu bloco de índice
 - Bloco de índice pode reservar as últimas poucas entradas para armazenar ponteiros para mais blocos de índice
 - Técnica denominada encadeamento.



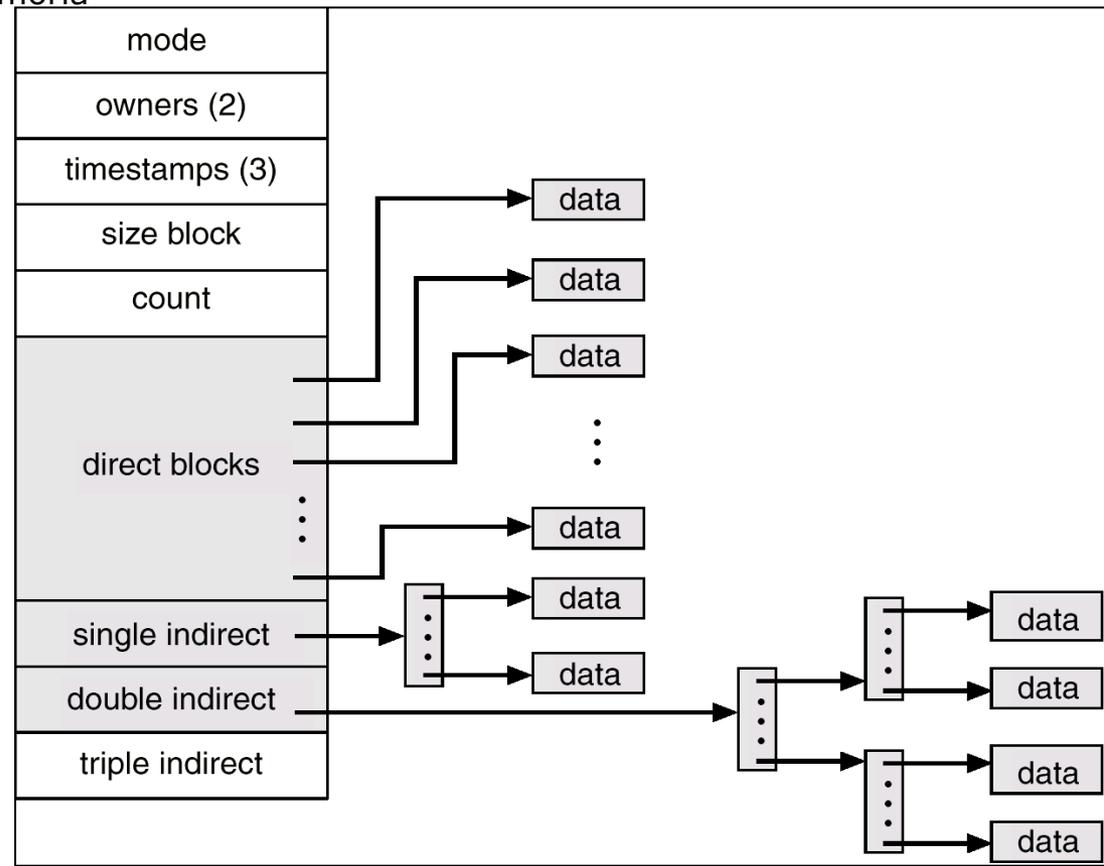
Alocação de arquivos indexada

■ A principal vantagem

- A busca para localização dos blocos de dados pode ocorrer **nos próprios blocos** de índice.
- Geralmente os sistemas de arquivo colocam os blocos de índice perto dos blocos de dados a que se referem
 - Assim, os blocos de dados podem ser acessados rapidamente depois que seu bloco de índice for carregado para a memória

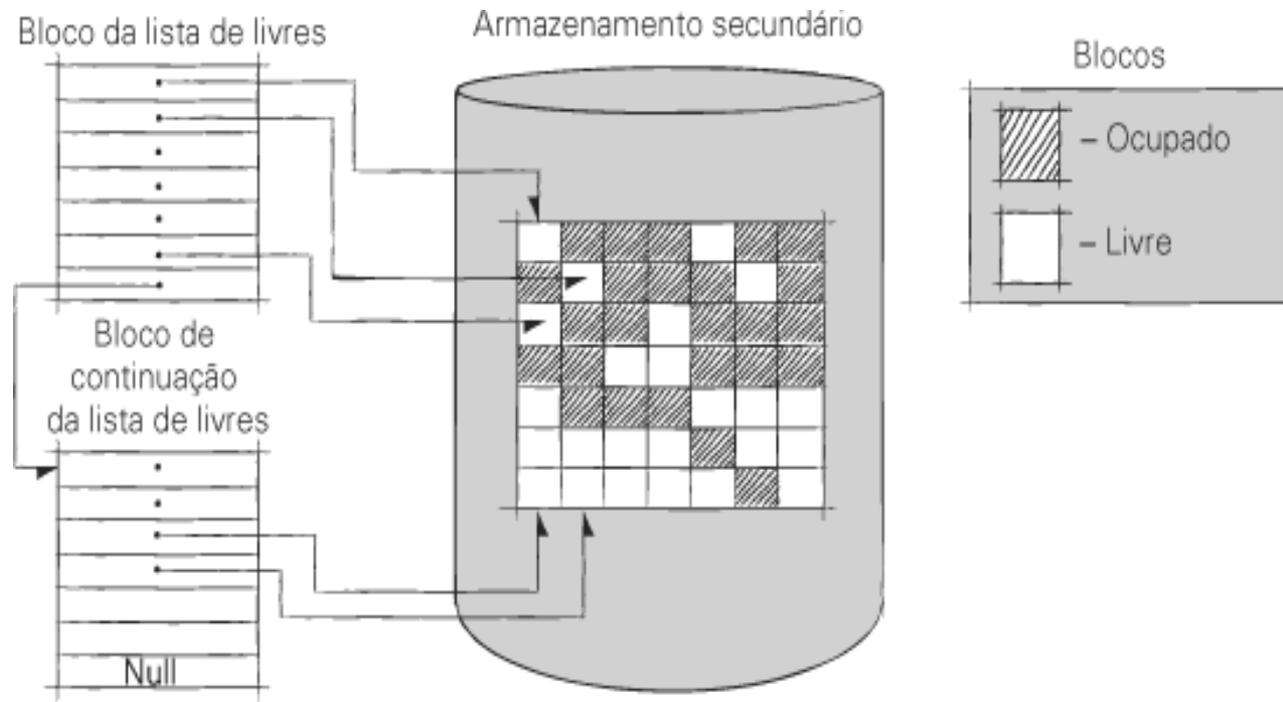
■ Exemplo: UNIX

- Os blocos de índice são chamados ***i-nodes*** (isto é, *index nodes*, ou nós de índice) em sistemas operacionais baseados no UNIX.



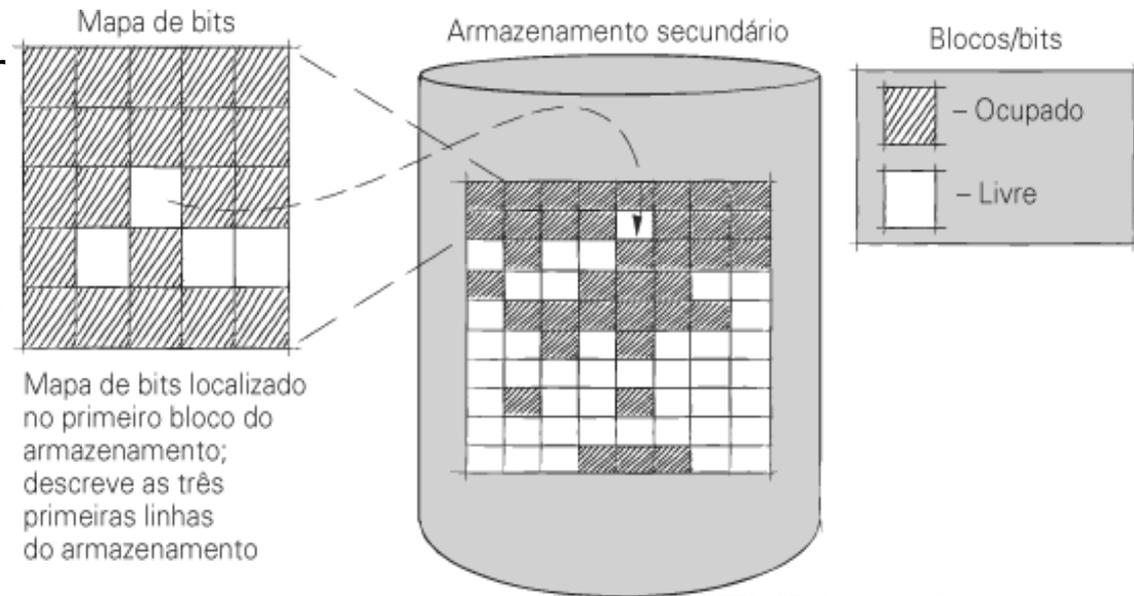
Gerenciamento de espaço livre

- Alguns sistemas usam uma **lista** para gerenciar o espaço de livres do dispositivo de armazenamento.
 - **Lista de blocos livres**: lista encadeada **que contém a localização dos blocos livres**.
 - Na alocação, os blocos são alocados a partir do começo da lista de livres.
 - Os blocos recém-liberados são anexados ao fim da lista.
- Essa forma de organização gera **baixa sobrecarga** para executar operações de manutenção da lista de livres.
- Os arquivos tendem a ser alocados em blocos **não contíguos**.
 - Isso aumenta o tempo de acesso ao arquivo.



Gerenciamento de espaço por mapa de bits

- Outra forma de organizar os blocos na unidade de armazenamento externo consiste em usar um **mapa de bits**
- Mapa (*bitmap*) contém **um bit para cada bloco na memória**.
 - o i -ésimo bit corresponde ao estado (livre ou ocupado) do i -ésimo bloco no dispositivo de armazenamento.
- **Vantagem do mapa de bits sobre a lista de livres:**
 - O sistema de arquivo consegue determinar rapidamente se existem blocos contíguos disponíveis **em localizações específicas** no disco.
- **Desvantagem do mapa de bits:**
 - O sistema de arquivo pode precisar pesquisar todo o mapa de bits para encontrar um bloco livre, caso em que a sobrecarga de execução é substancial.



Controle de acesso a arquivos

- **Os arquivos em geral também são usados para armazenar dados sensíveis, como:**
 - Número de cartão de crédito
 - Senha
 - Número da previdência social
- **Portanto, devem contar com mecanismos para controlar o acesso dos usuários aos dados.**
 - Matriz de controle de acesso
 - Controle de acesso por classe de usuários

Matriz de controle de acesso

- **Matriz bidimensional de controle de acesso:**
 - A entrada a_{ij} é 1 se o usuário i tiver permissão de acesso ao arquivo j .
 - Do contrário, $a_{ij} = 0$.

- Quando o número de usuários for grande e a quantidade de arquivos também
 - Matriz geralmente seria extensa e **esparsa**.
- **Não é adequada para a maioria dos sistemas.**

Usuário \ Arquivo	Arquivo									
	1	2	3	4	5	6	7	8	9	10
1	1	1	0	0	0	0	0	0	0	0
2	0	0	1	0	1	0	0	0	0	0
3	0	1	0	1	0	1	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1
6	0	0	0	0	0	1	1	0	0	0
7	1	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	0
9	1	1	1	1	0	0	0	0	1	1
10	1	1	0	0	1	1	0	0	0	1

Controle de acesso por classes de usuário

- **Requer um espaço consideravelmente menor**
- **Controla o acesso de várias classes de usuários**
- **Dentre as classes de usuários encontram-se:**
 - O proprietário do arquivo
 - Um usuário específico
 - Grupo
 - Projeto
 - Público
- **Dados de controle de acesso**
 - Podem ser armazenados como parte do bloco de controle de arquivo.
 - Em geral consomem uma quantidade insignificativa de espaço.

Proteção da integridade dos dados

- **Os sistemas de computador normalmente armazenam informações críticas, como:**
 - Inventários
 - Registros financeiros
 - Informações pessoais
- **Quedas de sistema, acidentes naturais e programas mal-intencionados podem destruir essas informações.**
 - Os resultados desses eventos podem ser catastróficos.
- **Os sistemas operacionais e os sistemas de armazenamento de dados devem ser tolerantes a falhas:**
 - Devem estar preparados para a possibilidade de acidentes e oferecer técnicas que permitam voltar à condição normal.

Cópia de segurança e recuperação

- **Técnicas de cópia de segurança (backup)**
 - Armazenam cópias extras das informações.
- **Técnicas de recuperação**
 - Permitem que o sistema restaure os dados após uma falha de sistema.
- **Proteções físicas como travas e alarmes contra incêndio são o nível mais baixo de proteção de dados.**
- **A execução periódica de cópias de segurança é a técnica mais comum usada para garantir a disponibilidade contínua de dados.**

Cópia de segurança e recuperação

- **Cópias de segurança físicas**
 - Duplicam os dados do dispositivo de armazenamento no nível do bit.
- **Cópias de segurança lógicas**
 - Armazenam dos dados de um sistema de arquivo e sua estrutura lógica.
 - Inspeccionam a estrutura do diretório para determinar quais arquivos precisam de cópia de segurança e, em seguida, gravam esses arquivos em um dispositivo de cópia de segurança em um formato de arquivamento comum, em geral, compactado.
- **As cópias de segurança **incrementais** são cópias de segurança lógicas que armazenam apenas os dados do sistema de arquivo que mudaram desde a última cópia de segurança.**

Servidores de arquivos e sistemas distribuídos

- Uma maneira de controlar referências a arquivos não locais em uma rede de computadores é encaminhar as essas requisições a um **servidor de arquivos remoto**
 - Sistema de computador dedicado a resolver referências a arquivos feitas a partir de outros computadores
 - Centraliza o controle dessas referências
 - O servidor de arquivos pode facilmente se tornar um gargalo, porque todos os computadores clientes enviam todas as requisições ao servidor.
- **O melhor é permitir que os computadores comuniquem-se diretamente uns com os outros**

Windows

- Sistemas de arquivos suportados pelo Windows
 - CDFS (Superset ISO-9660)
 - UDF (DVD-ROM)
 - FAT12, FAT16, FAT32
 - NTFS: Sistema de arquivos nativo do Windows
 - Indexadores de 64 bits
 - Tamanho máximo do volume 256TB
 - Características avançadas, como: proteção de arquivos, quotas, compressão, criptografia

Linux

- Exemplos de sistemas de arquivos suportados pelo Linux
 - VFS (Virtual File System)
 - ext2fs
 - ext3fs
 - proc (sistema de arquivos de processos)
 - zfs
 - etc

Fim

