

SISTEMAS OPERACIONAIS

Sistema de Arquivos

Andreza Leite
andrea.leite@univasf.edu.br

Plano de Aula



- Arquivos
- Organização de arquivos
- Diretórios
- Métodos de acesso
- Métodos de alocação em disco
- Fragmentação
- Tamanho de bloco
- Proteção de acesso

Introdução

3

- Condições essenciais para armazenamento de informações por um longo prazo:
 - ▣ Deve ser possível armazenar uma grande quantidade de informação.
 - ▣ A informação deve sobreviver ao término do processo que esta usando a mesma.
 - ▣ Múltiplos processos devem ser capaz de acessar a informação simultaneamente.

Introdução

4

- Pense em um disco como uma sequência linear de blocos de tamanho fixo e que apoiam a leitura e registro dos blocos. As questões a seguir surgem rapidamente:
 - ▣ Como encontrar a informação?
 - ▣ Como manter um usuário de uma outra leitura de dados?
 - ▣ Como saber se os blocos são livres?

Introdução



- Os arquivos são gerenciados pelo sistema operacional e é mediante a implementação de arquivos que o sistema operacional estrutura e organiza as informações.
- A parte do sistema responsável pela gerência é denominada **sistema de arquivo** que é a parte mais visível do sistema operacional pois é uma **atividade freqüentemente realizada pelos usuários**.
- Deve ocorrer de maneira uniforme independente dos diferentes dispositivos de armazenamento.

Arquivos



- É um conjunto de registros definidos pelo sistema de arquivos e podem ser armazenados em diferentes dispositivos físicos.
- É constituído de informações logicamente relacionadas, podendo representar programas ou dados.

Arquivos



- É identificado por meio de um nome, formado por uma seqüência de caracteres. Em alguns sistemas operacionais, a identificação de um arquivo é composta por duas partes separadas por um ponto, a parte após o ponto é chamada extensão do arquivo e serve para identificar o conteúdo.

Extensão de Arquivos



Extensão	Descrição
arquivo.c	Arquivo fonte em C
arquivo.cob	Arquivo fonte em Cobol
arquivo.exe	Arquivo executável
arquivo.obj	Arquivo objeto
arquivo.pas	Arquivo fonte em Pascal
arquivo.txt	Arquivo texto
arquivo.mp3	Arquivo de música
arquivo.dll	Arquivo de biblioteca dinâmica

Organização de Arquivos



- Consiste no modo como seus dados estão internamente armazenados.
- Quando o arquivo é criado pode-se definir que organização será adotada, que pode ser uma estrutura suportada pelo sistema operacional ou definida pela própria aplicação.
- A forma mais simples de organização de arquivos é através de uma seqüência não-estruturada de bytes, onde o sistema de arquivos não impõe nenhuma estrutura lógica para os dados, a aplicação deve definir toda a organização.

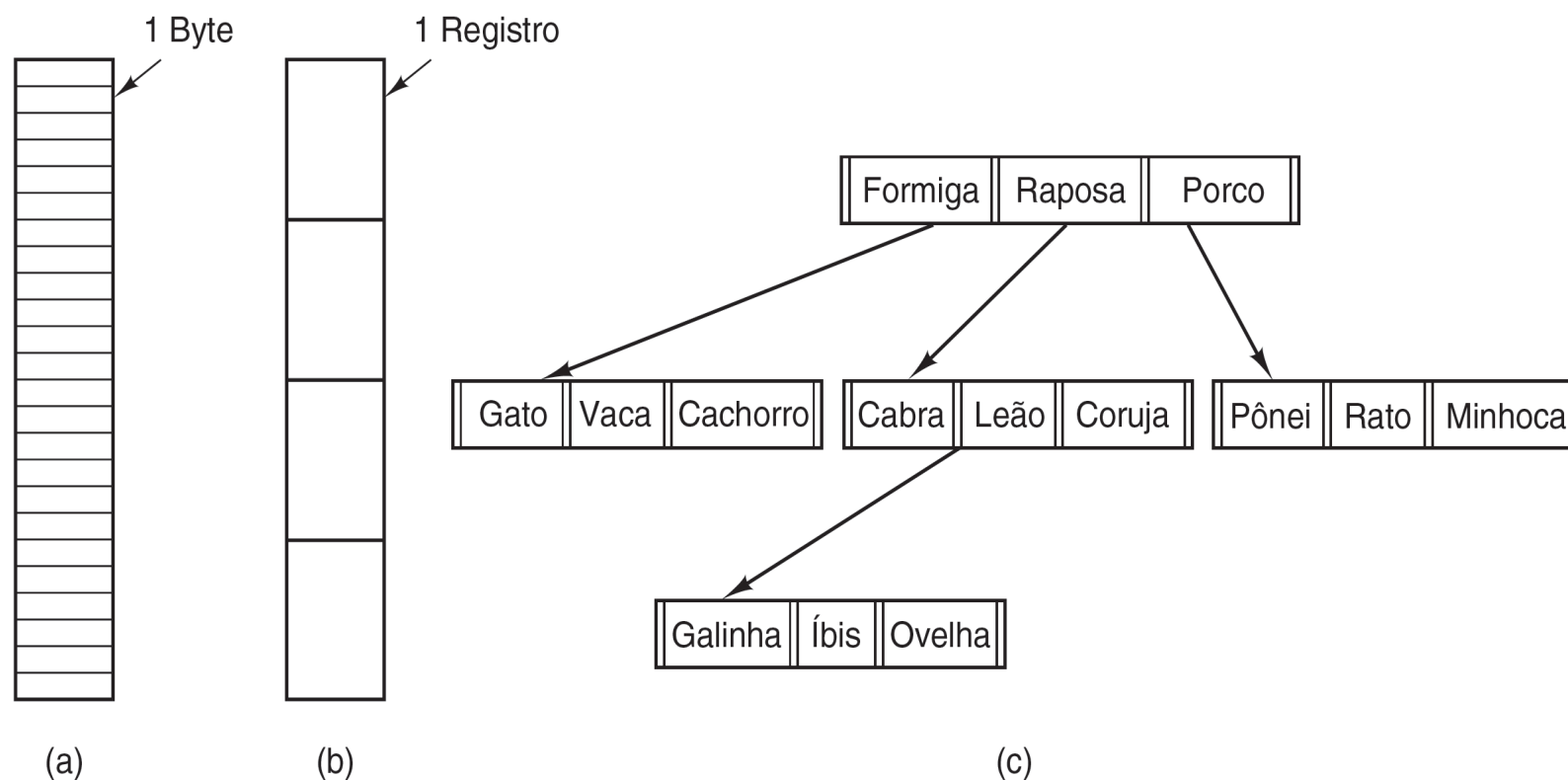
Organização de Arquivos



- A grande vantagem deste modelo é a flexibilidade para criar estruturas de dados, porém todo o controle de dados é de responsabilidade da aplicação.
- Alguns sistemas operacionais estabelecem diferentes organizações de arquivos e cada um deve seguir um modelo suportado pelo sistema de arquivos.

Estrutura de Arquivos

11



■ **Figura 4.1** Três tipos de arquivos. (a) Seqüência de bytes. (b) Seqüência de registros. (c) Árvore.

Estrutura de Arquivos

12

- Na **sequencia** desestruturada **de bytes** o SO não sabe o que o arquivo contém. Tudo o que ele vê são bytes. Qualquer significado deve ser imposto pelos programas em nível de usuário.
 - UNIX, MS-DOS e Windows utilizam essa estratégia.
 - Oferece máxima flexibilidade
 - Os programas de usuarios podem colocar qualquer coisa em seus arquivos e nomea-los como lhes convier.

Estrutura de Arquivos

13

- No tipo **sequencia de registros**, um arquivo é uma sequencia registros de tamanho fixo, cada um com alguma estrutura interna.
 - ▣ A idéia central é que a operação de leitura retorna um registro e a de escrita sobrepõe ou anexa um registro
 - ▣ Nenhum sistema atual usa esse modelo como sistema primário de arquivo, mas ele era comum nos computadores de grande porte na época dos cartões perfurados.

Estrutura de Arquivos

14

- No último tipo, um arquivo é constituído de uma **árvore de registros**, não necessariamente todos de mesmo tamanho, cada um contendo um campo-chave em uma posição fixa no registro.
 - ▣ A árvore é ordenada pelo campo chave pra que se busque mais rapidamente por uma chave específica. A operação básica não é obter o próximo registro mas obter o registro com a chave específica.
 - ▣ Esse tipo é bem diferente do fluxo de bytes desestruturado usados no UNIX e Windows mas é aplicado em computadores de grande porte ainda usados para alguns processamentos de dados comerciais.

Idéias Básicas



- A representação da informação dentro de um computador *digital* é feita na forma de *números*.
- No Linux ou no Windows, arquivos são concebidos e tratados como STREAMS de BYTES.
- O que é um BYTE ? Um BYTE é um número entre 0 e 255.
 - ▣ Um BYTE é um conjunto de 8 bits (dígitos binários)

Decifrando os números: o código

A	:	65	a	:	97	0	:	48
B	:	66	b	:	98	1	:	49
...				
Z	:	90	z	:	122	9	:	57

Caracteres especiais:

0	:	NULO
8	:	BACKSPACE
9	:	TAB
10	:	mudança de linha
12	:	mudança de página
13	:	ENTER

Representação de um arquivo

Representação interna do conteúdo de um arquivo

65	66	67	10	49	50	51	10	68	69	10	97	98	10
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Convertendo a representação interna de um arquivo que eu *considero* como um arquivo texto:

A	B	C	10	1	2	3	10	D	E	10	a	b	10
---	---	---	----	---	---	---	----	---	---	----	---	---	----

*mudança
de linha*

*mudança
de linha*

*mudança
de linha*

*mudança
de linha*

Representação de um arquivo

Visualizando o arquivo no meu Terminal de Vídeo:

A	B	C	10	1	2	3	10	D	E	10	a	b	10
---	---	---	----	---	---	---	----	---	---	----	---	---	----

```
$ cat arquivo
```

```
ABC
```

```
123
```

```
DE
```

```
ab
```

```
$
```

Métodos de Acesso



- Em função de como o arquivo está organizado o sistema de arquivos pode recuperar registros de diferentes maneiras:
 - ▣ **Acesso seqüencial:** arquivos armazenados em fitas magnéticas, o acesso era restrito à leitura na ordem em que eram gravados, sendo a gravação de arquivos possível apenas no final do arquivo. Pode-se combinar o acesso seqüencial com o direto e com isso acessar diretamente um arquivo e os demais em forma seqüencial.

Métodos de Acesso



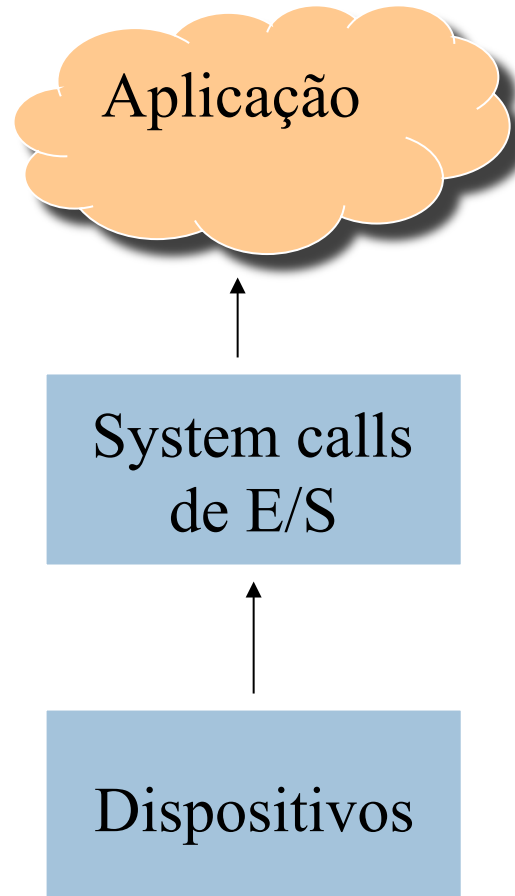
- **Acesso direto:** permite a leitura/gravação de um registro diretamente na sua posição. É realizado através do número de registro. Não existe restrição a ordem em que os registros são lidos ou gravados. Somente é possível quando é definido com registros de tamanho fixo.
- **Acesso indexado ou por chave:** o arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros e a partir desta informação realiza-se um acesso direto.

Operações de Entrada / Saída



- O sistema de arquivos oferece um conjunto de system calls que permite às aplicações realizar operações de E/S, como tradução de nomes em endereços, leitura e gravação de dados e criação/eliminação de arquivos.
- As system calls de E/S tem como função oferecer uma interface simples e uniforme entre a aplicação e os diversos dispositivos.

Operações de Entrada / Saída



Operações de Entrada/Saída



Comando	Descrição
<i>create</i>	Criação de um arquivo
<i>open</i>	Abertura de um arquivo
<i>read</i>	Leitura de dados de um arquivo
<i>write</i>	Gravação de dados de um arquivo
<i>close</i>	Fechamento de um arquivo
<i>rename</i>	Alteração de nome de um arquivo
<i>erase</i>	Eliminação de um arquivo

Atributos



- São informações de controle que dependendo do sistema de arquivos variam, porém alguns como tamanho, criador, proteção e data estão presentes em quase todos.
- Alguns atributos especificados na criação do arquivo não podem ser mudados e outros são modificados pelo próprio sistema operacional. E ainda existe alguns que podem ser alterados pelo usuário tais como proteção, tamanho e senha.

Exemplo de Atributos

```
$ ls -l
```

```
total 9552  
-rw-r--r--@ 1 andreza  staff  141029 Feb  6 11:54 AboutUFPE.docx  
-rw-r--r--@ 1 andreza  staff   23959 Feb  5 10:55 Anexo II - versão Ricardo Ramos.xlsx  
-rw-r--r--@ 1 andreza  staff   48938 Feb  7 09:19 Anexo II v3.xlsx  
-rw-r--r--@ 1 andreza  staff   40671 Feb  4 15:59 Anexo II.xlsx  
-rw-r--r--@ 1 andreza  staff   46169 Feb  5 10:29 Anexo II_andreza.xlsx  
-rwxrwxrwx  1 andreza  staff 48394215 Jan 30 12:23 Arquitetura de Sistemas Operacionais.pdf  
drwxr-xr-x  3 andreza  staff    102 Jan 30 09:23 Docs Posse  
drwxr-xr-x  5 andreza  staff    170 Nov 29 19:00 FileZilla.app  
-rw-r--r--  1 andreza  staff  113180 Feb  7 15:19 NORMAS PARA SEGUNDA CHAMADA.pdf  
drwxrwxrwx 35 andreza  staff    1190 Feb  6 14:56 backupPC  
-rw-r--r--@ 1 andreza  staff  103090 Feb  6 19:18 textoDOCUMENTOProjetoEn.docx  
Andrezas-MacBook-Pro:UNIVASF andreza$
```

Exemplo de Atributos

```
$ ls -l
```

```
drwxr-xr-x  5 laureano prof      4096 Abr 14 11:34 a
drwxr-xr-x  3 laureano prof      4096 Mar 31 12:25 abc
-rw-r--r--  1 laureano prof        647 Abr 28 12:24 arqcut1
-rw-r--r--  1 laureano prof      2335 Abr 28 12:24 arqgrep
drwxr-xr-x 11 laureano prof      4096 Mai 30  2005 arquivos
drwxr-xr-x  2 laureano prof      4096 Mai 18  2005 asu
drwx----- 11 laureano prof      4096 Jul  5  2004 bash-2.05b
-rw-r--r--  1 laureano prof 8437760 Jul  5  2004 bash5.tar
drwxr-xr-x  2 laureano prof      4096 Mai 22  2005 c
-rw-----  1 laureano prof      1470 Set 28  2004 client.c
```

Diretórios



- Modo como o sistema organiza os diferentes arquivos contidos num disco. É a estrutura de dados que contém entradas associadas aos arquivos onde estão informações como localização física, nome, organização e demais atributos.
- Quando um arquivo é aberto, o sistema operacional procura a sua entrada na estrutura de diretórios, armazenando as informações do arquivo em uma tabela mantida na memória principal. Esta tabela contém todos os arquivos abertos, sendo fundamental para aumentar o desempenho das operações com arquivos.

Diretórios



- **Nível Único (single-level directory):** é a implementação mais simples, existe apenas um único diretório contendo todos os arquivos do disco. É bastante limitado já que não permite que usuários criem arquivos com o mesmo nome o que ocasionaria um conflito no acesso aos arquivos.

Estrutura de diretórios - Nível Único

29

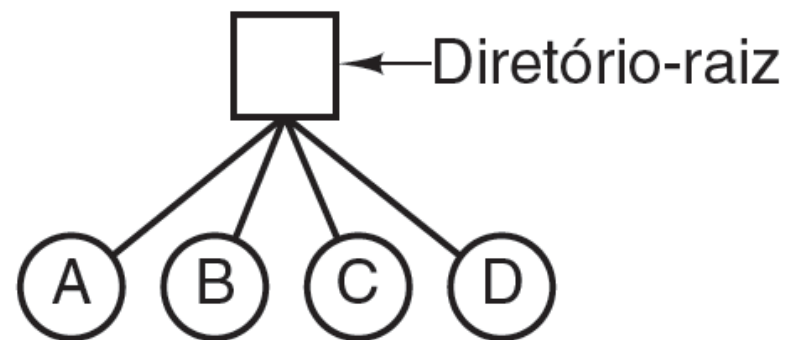


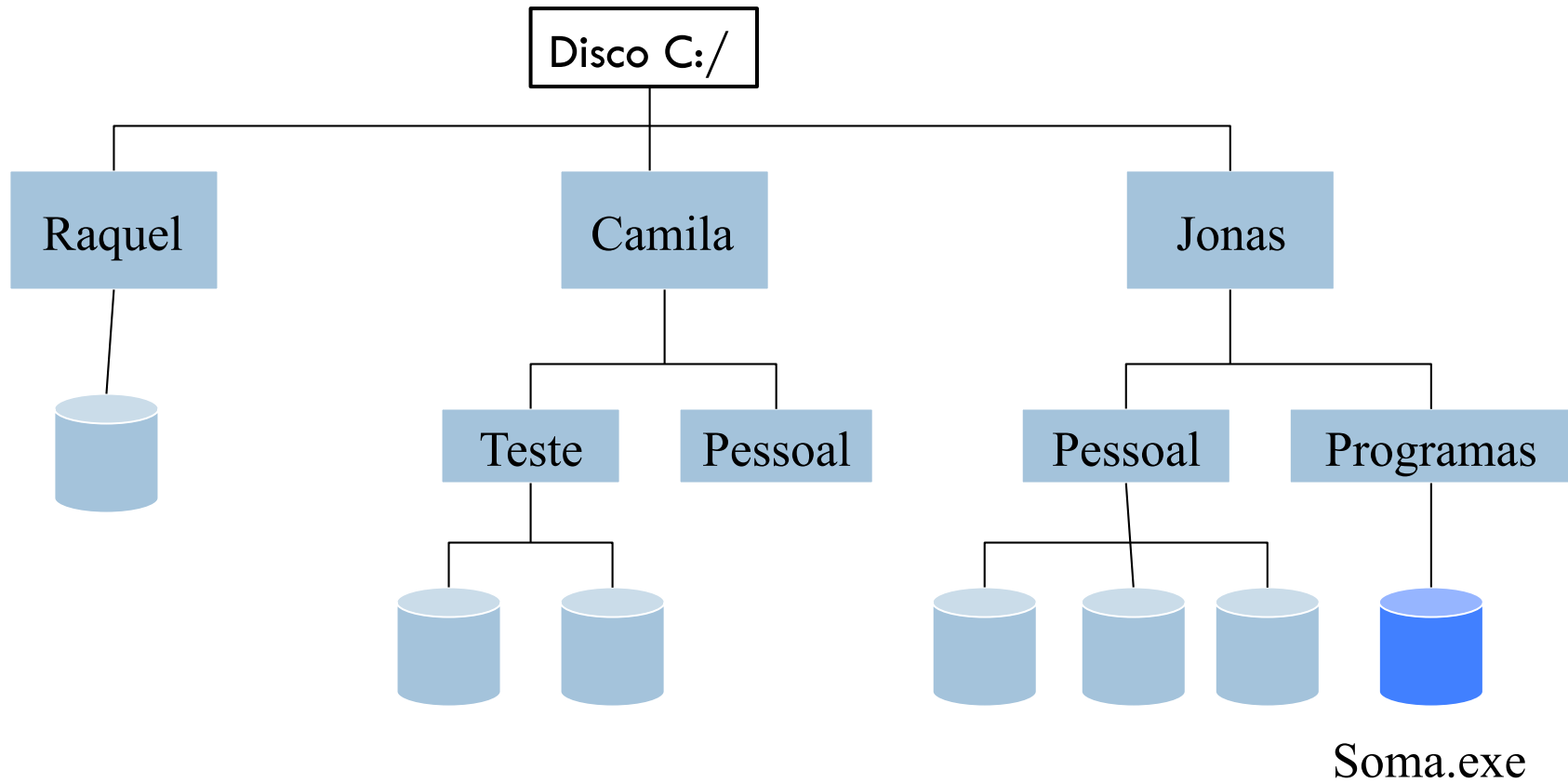
Figura 4.4 Um sistema de diretórios em nível único contendo quatro arquivos.

Diretórios

- *User File Directory* (UFD): para cada usuário existe um diretório particular e assim poderia criar arquivos com qualquer nome.
- Deve haver um nível de diretório adicional para controle chamado de *Master File Directory* (MFD) que é indexado pelo nome do usuário e cada entrada aponta para o diretório pessoal.
- É análoga a uma estrutura de dados em árvore onde o MFD é a raiz, os galhos são a UFD e os arquivos são as folhas.
- Quando se referencia a um arquivo é necessário especificar seu nome e seu diretório isto é chamado de *path* (caminho).

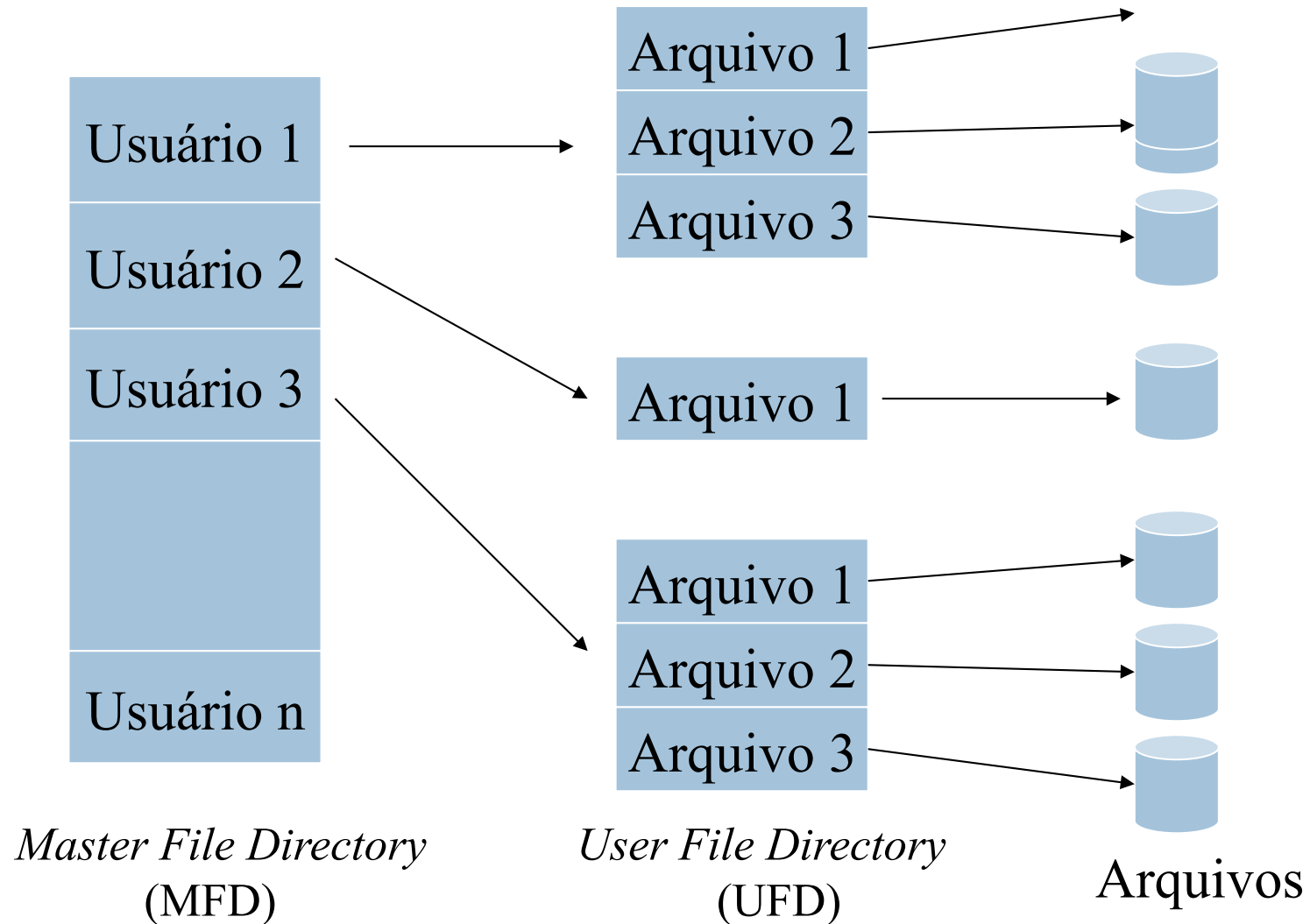
Path de um Arquivo

31



Estrutura de diretórios - Dois níveis

32

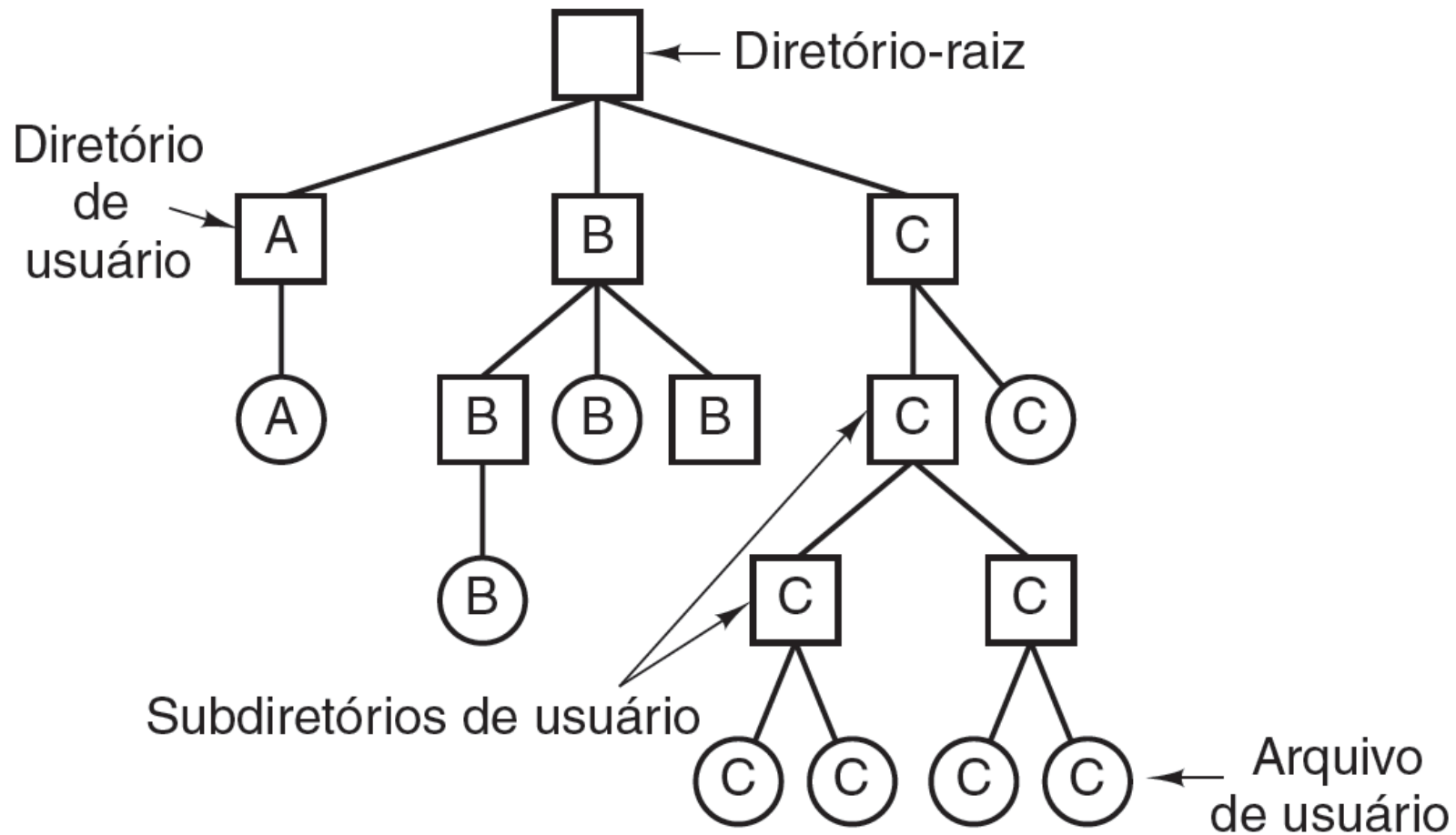


Diretórios

- Estrutura de diretórios em *Árvore (Tree Structured Directory)*
 - ▣ Adotado pela maioria dos sistemas operacionais e é logicamente melhor organizado.
- É possível criar quantos diretórios quiser, podendo um diretório conter arquivos e outros diretórios (chamados subdiretórios).
- Cada arquivo possui um path único que descreve todos os diretórios da raiz (MFD) até o diretório onde o arquivo está ligado e na maioria dos sistemas os diretórios são tratados como arquivos tendo atributos e identificação.

Estrutura de Diretórios - Árvore

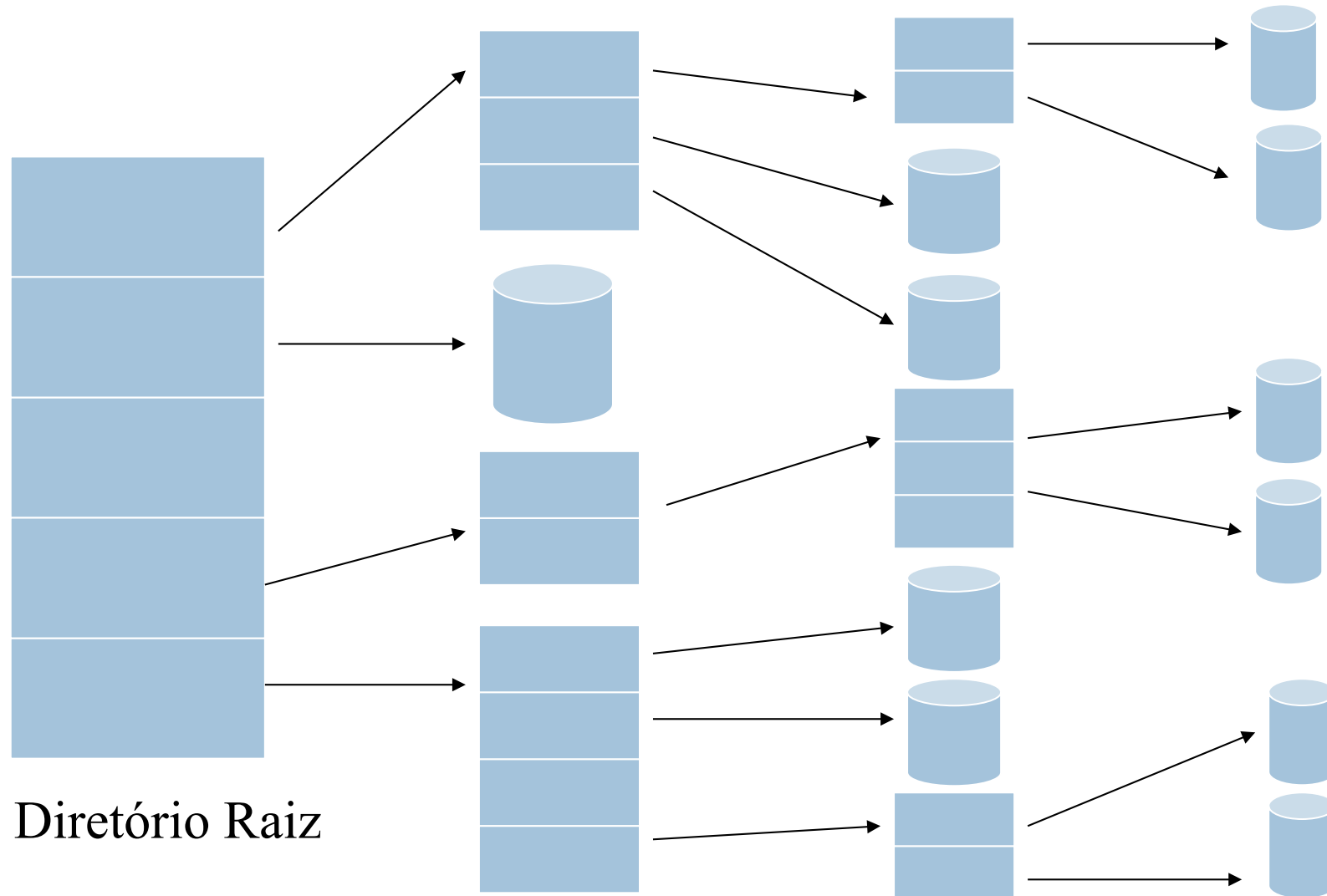
34



■ **Figura 4.5** Um sistema hierárquico de diretórios.

Estrutura de Diretórios - Árvore

35



Monitoramento de Espaço em Disco



- A criação de arquivos exige que o sistema operacional tenha controle de quais áreas ou blocos no disco estão livres e este controle é realizado através de uma estrutura (geralmente lista ou tabela) de dados que armazenam informações e possibilitam ao sistema de arquivos gerenciar o espaço livre.

Monitoramento de Espaço em Disco



- A forma mais simples de implementar uma estrutura de espaços livres é através de uma tabela chamada mapa de bits (*bit map*) onde cada entrada da tabela é associada a um bloco e representado por um bit, que pode assumir valor igual a 0 (bloco livre) ou 1 (bloco alocado).
- Esta estrutura gera um gasto excessivo de memória já que para cada bloco deve existir uma entrada na tabela.

Monitoramento de Espaço em Disco

- Outra forma é realizar o controle por meio da ligação encadeada de todos os blocos livres e cada bloco deve possuir uma área reservada para armazenamento do endereço do próximo.
- A partir do primeiro bloco pode-se ter acesso seqüencial aos demais de forma encadeada.
- Apresenta restrições se considerarmos que o algoritmo de busca de espaço livre sempre deve realizar uma pesquisa seqüencial na lista.

Monitoramento de Espaço em Disco

39

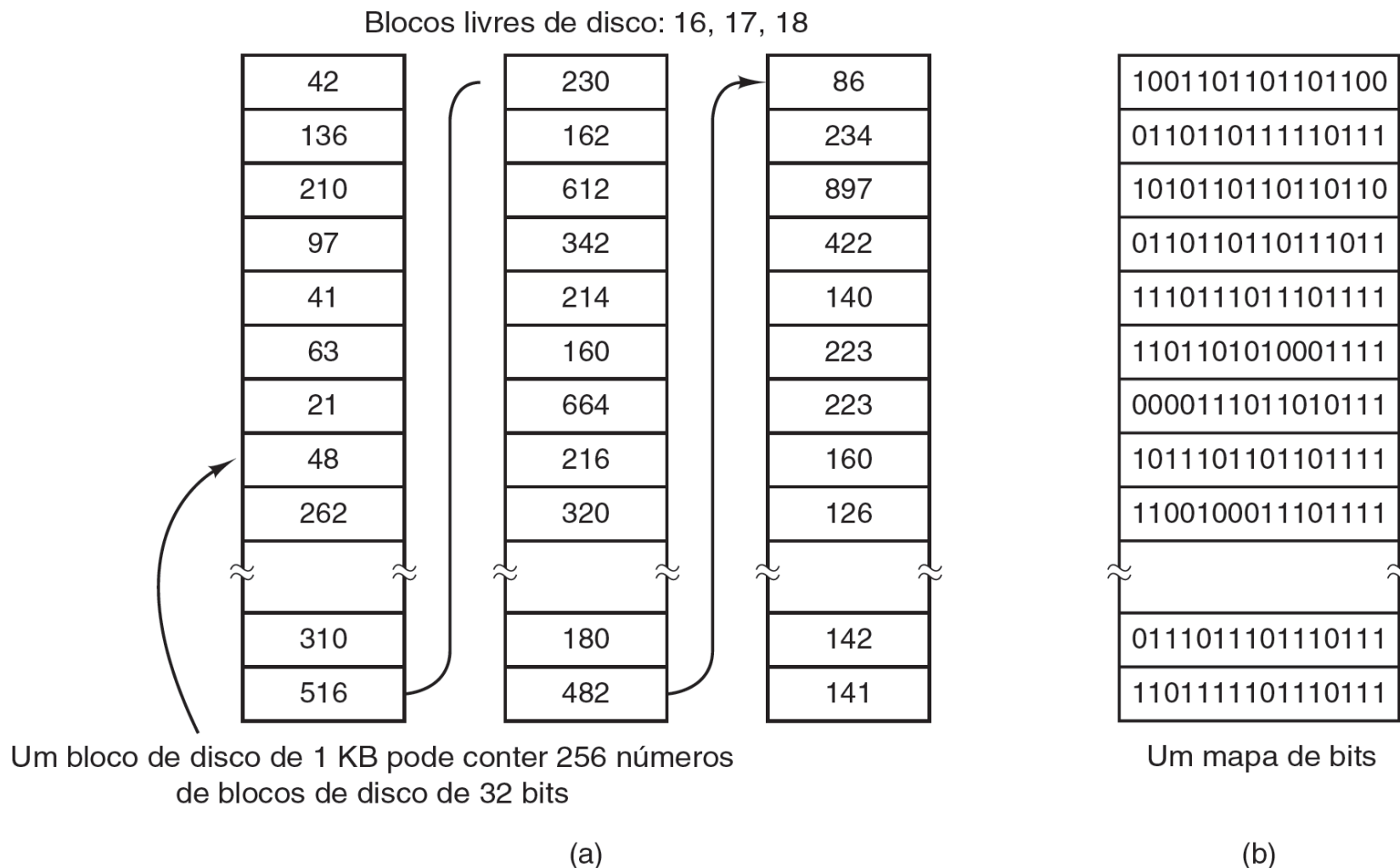


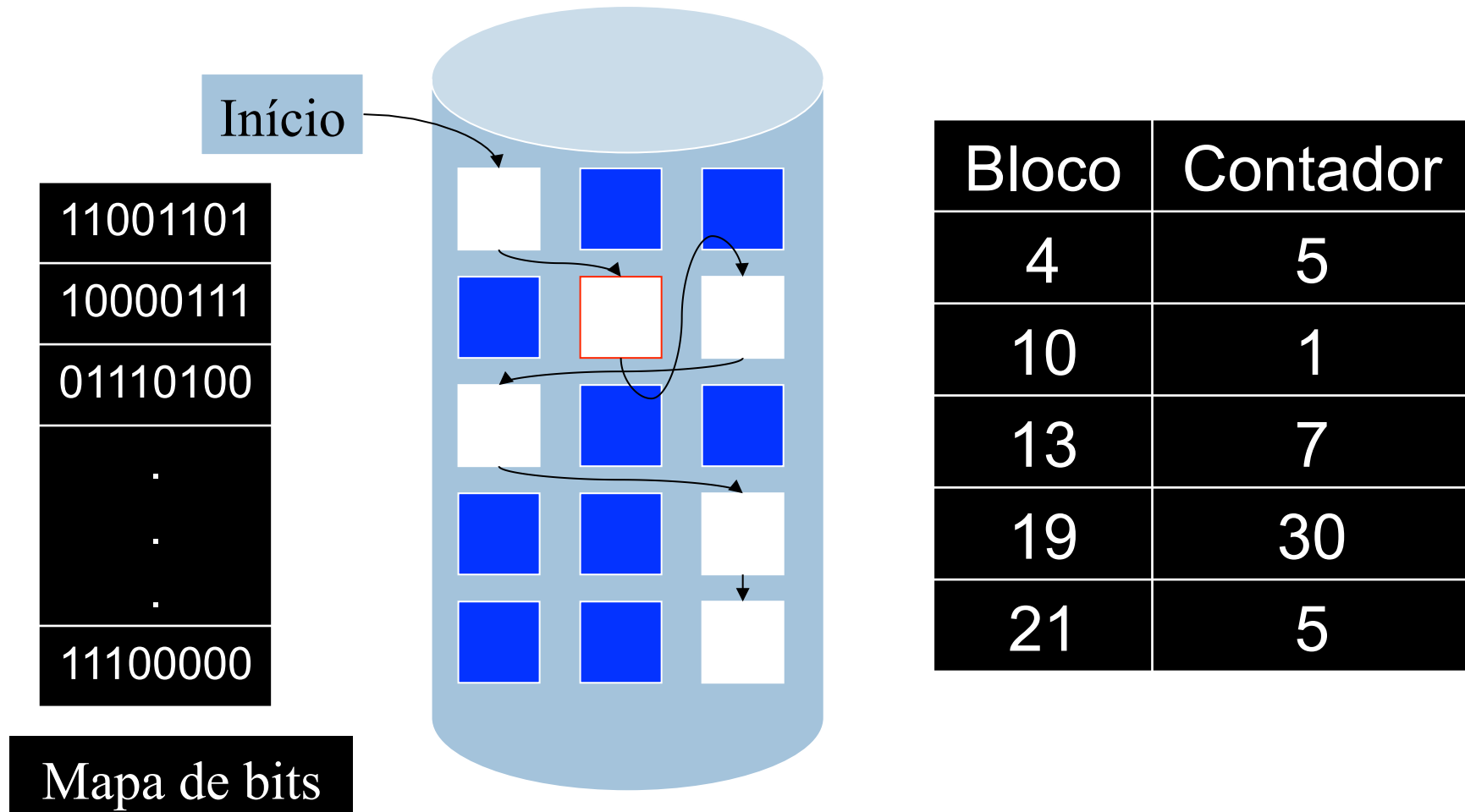
Figura 4.19 (a) Armazenamento da lista de blocos livres em uma lista encadeada. (b) Um mapa de bits.

Monitoramento de Espaço em Disco



- Outra solução leva em conta que blocos contíguos são geralmente alocados ou liberados simultaneamente, com base neste conceito é possível manter uma tabela com o endereço do primeiro bloco de cada segmento e o número de blocos livres contíguos que se seguem.

Alocação de Espaço em Disco



Alocação Contígua

- Consiste em armazenar um arquivo em blocos seqüencialmente dispostos. Neste tipo, o sistema localiza um arquivo através do endereço do primeiro bloco e da sua extensão em blocos.
- O acesso é bastante simples tanto para a forma seqüencial quanto para a direta
 - ▣ seu principal problema é a alocação de novos arquivos nos espaços livres, pois para colocar n blocos é necessário que se tenha uma cadeia com n blocos dispostos seqüencialmente no disco.

Alocação contínua

43

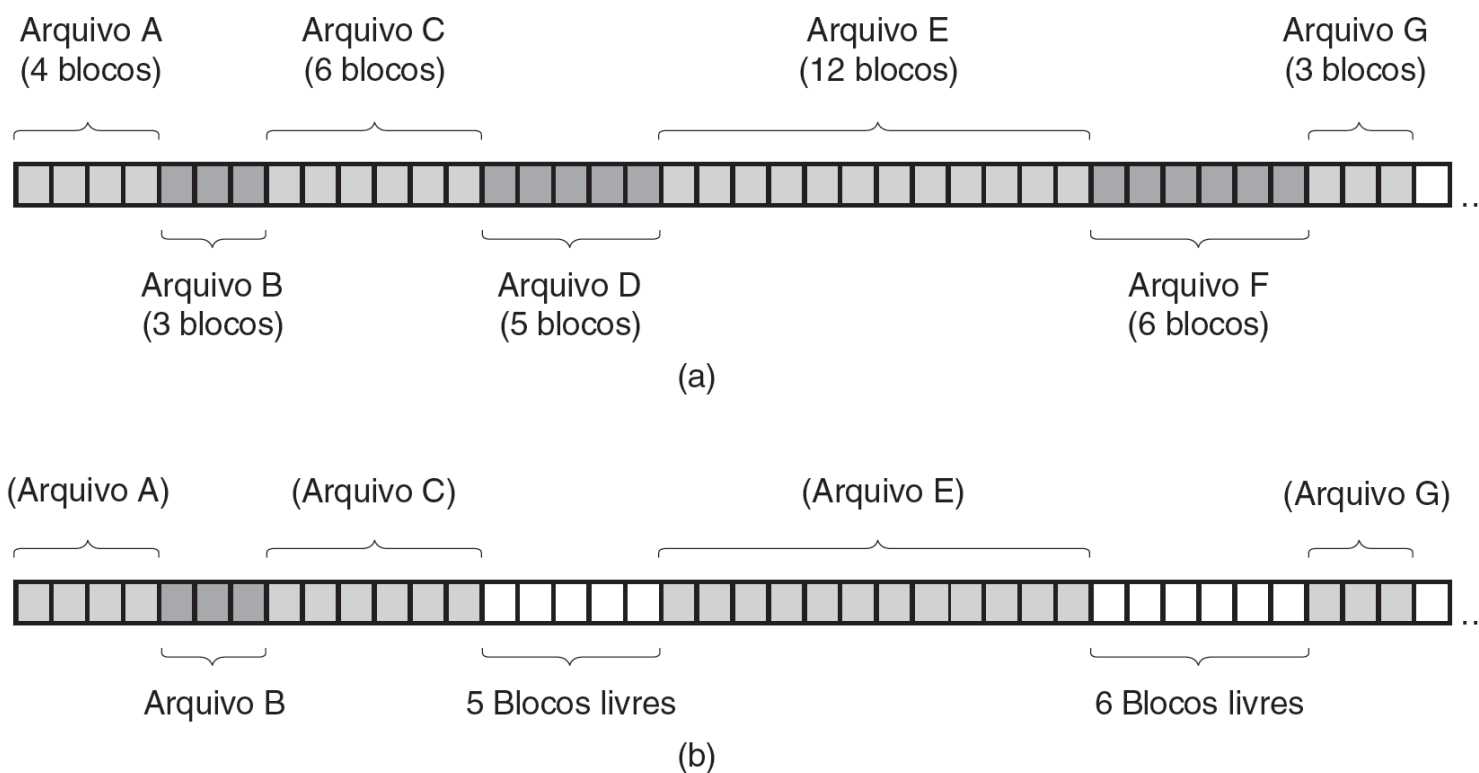


Figura 4.8 (a) A alocação contígua do espaço em disco para sete arquivos. (b) O estado do disco depois de os arquivos *D* e *F* terem sido removidos.

Alocação Contígua

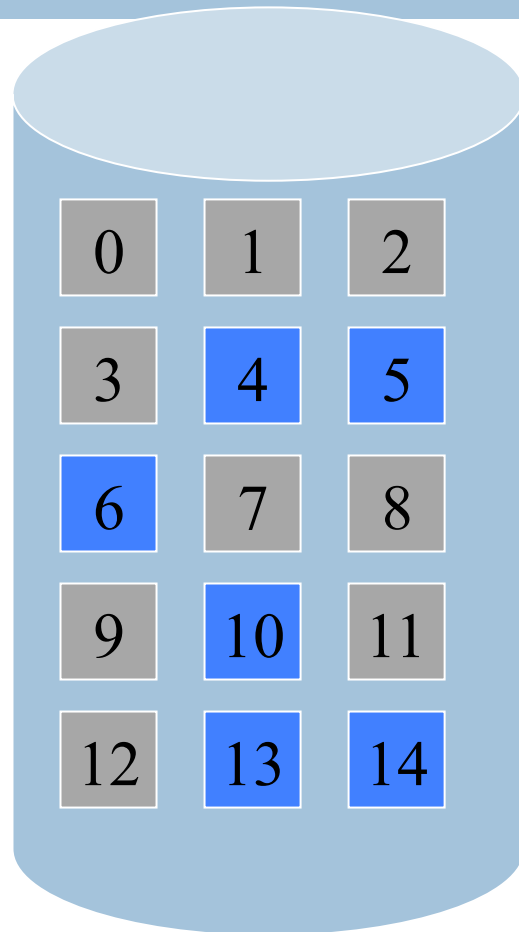
- Existe alguns problemas como:
 - **determinar o espaço necessário a um arquivo quando é criado e depois pode existir a necessidade de extensão e esta é uma operação complexa, sendo assim a pré-alocação seria uma solução mas pode ocasionar que parte do espaço alocado permaneça ocioso por um logo período de tempo.**
 - **Quando o sistema operacional deseja alocar espaço para um novo arquivo, pode existir mais de um segmento livre disponível com o tamanho exigido e é necessário alguma estratégia de alocação seja adotada para selecionar qual segmento deve ser escolhido.**

Alocação Contígua

arquivo	inicio	#blocos
readme.txt	008	003
prova.doc	000	008
Aula.pdf	017	005



Alocação Contígua



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

Alocação Contígua

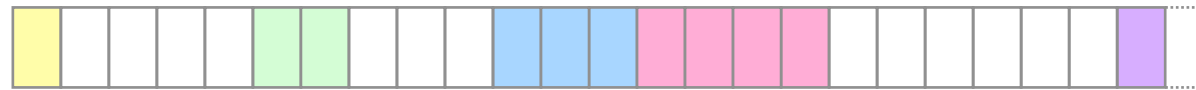


- Analisaremos as três principais estratégias:
 - First-fit: o primeiro segmento livre com tamanho suficiente para alocar o arquivo é selecionado. A busca na lista é seqüencial, sendo interrompida tão logo se encontre um segmento adequado.
 - Best-fit: seleciona o menor segmento livre disponível com tamanho suficiente para armazenar o arquivo. A busca em toda a lista se faz necessária para a seleção do segmento, a não ser que a lista esteja ordenada por tamanho.
 - Worst-fit: o maior segmento é alocado e a busca por toda a lista se faz necessária, a menos que exista uma ordenação por tamanho.

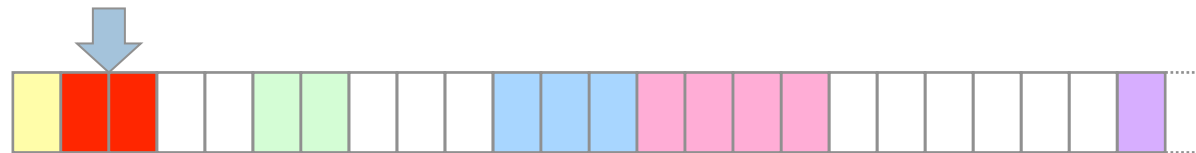
Alocando um arquivo c/ 2 blocos

48

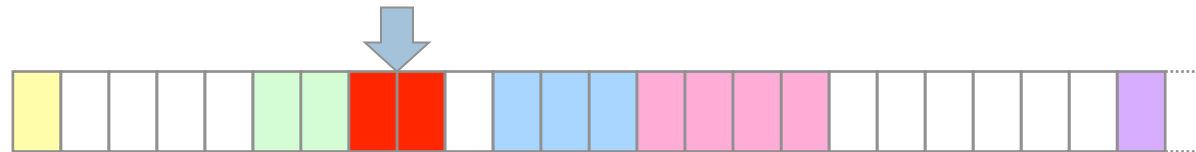
Situação inicial



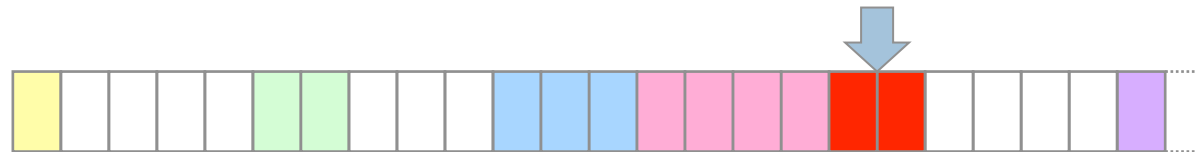
First-fit



Best-fit



Worst-fit



Alocação Contígua

- Independente da estratégia utilizada, a alocação apresenta um problema chamado **fragmentação de espaços livres**. O problema pode se tornar crítico quando um disco possui blocos livres disponíveis, porém sem um segmento contíguo onde o arquivo possa ser alocado.
- Deve ser feito a **desfragmentação periodicamente** (visando que este problema seja resolvido) para reorganizar os arquivos no disco a fim de que exista um único segmento de blocos livres. Há um grande consumo de tempo neste processo e **tem efeito temporário**.

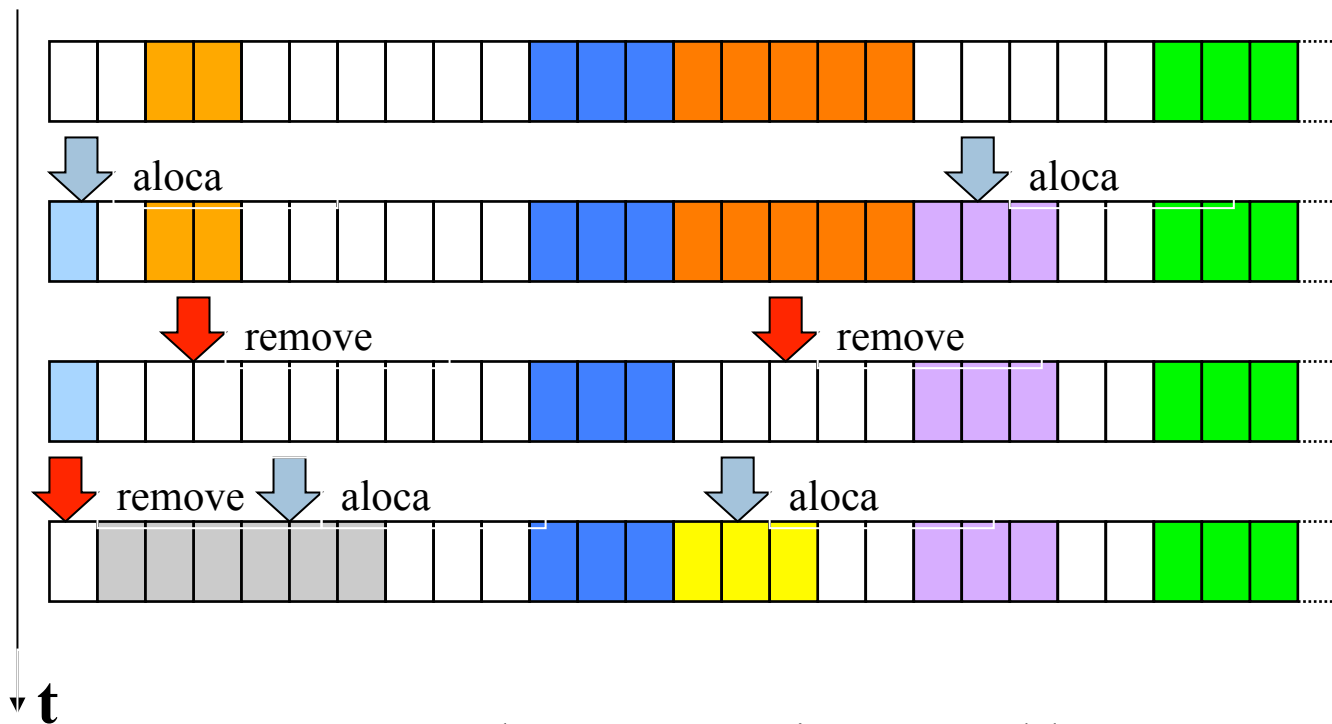
Fragmentação externa



- Espaços vazios **entre** blocos de arquivos.
 - À medida que o sistema evolui:
 - ▣ arquivos são criados e removidos
 - ▣ mais espaços vazios aparecem.
 - ▣ os espaços vazios ficam menores.
- ➔ **Alocar novos arquivos torna-se difícil !**

Evolução da fragmentação

51



Agora, como alocar um arquivo com 4 blocos ?

Desfragmentação



- ❑ Mover arquivos para reagrupar os fragmentos em espaços maiores
- ❑ Visa permitir alocar arquivos maiores
- ❑ Deve ser feita periodicamente
- ❑ Uso de algoritmos para minimizar movimentação de arquivos

Estratégias de desfragmentação

53

Situação inicial



Moveu 6 blocos



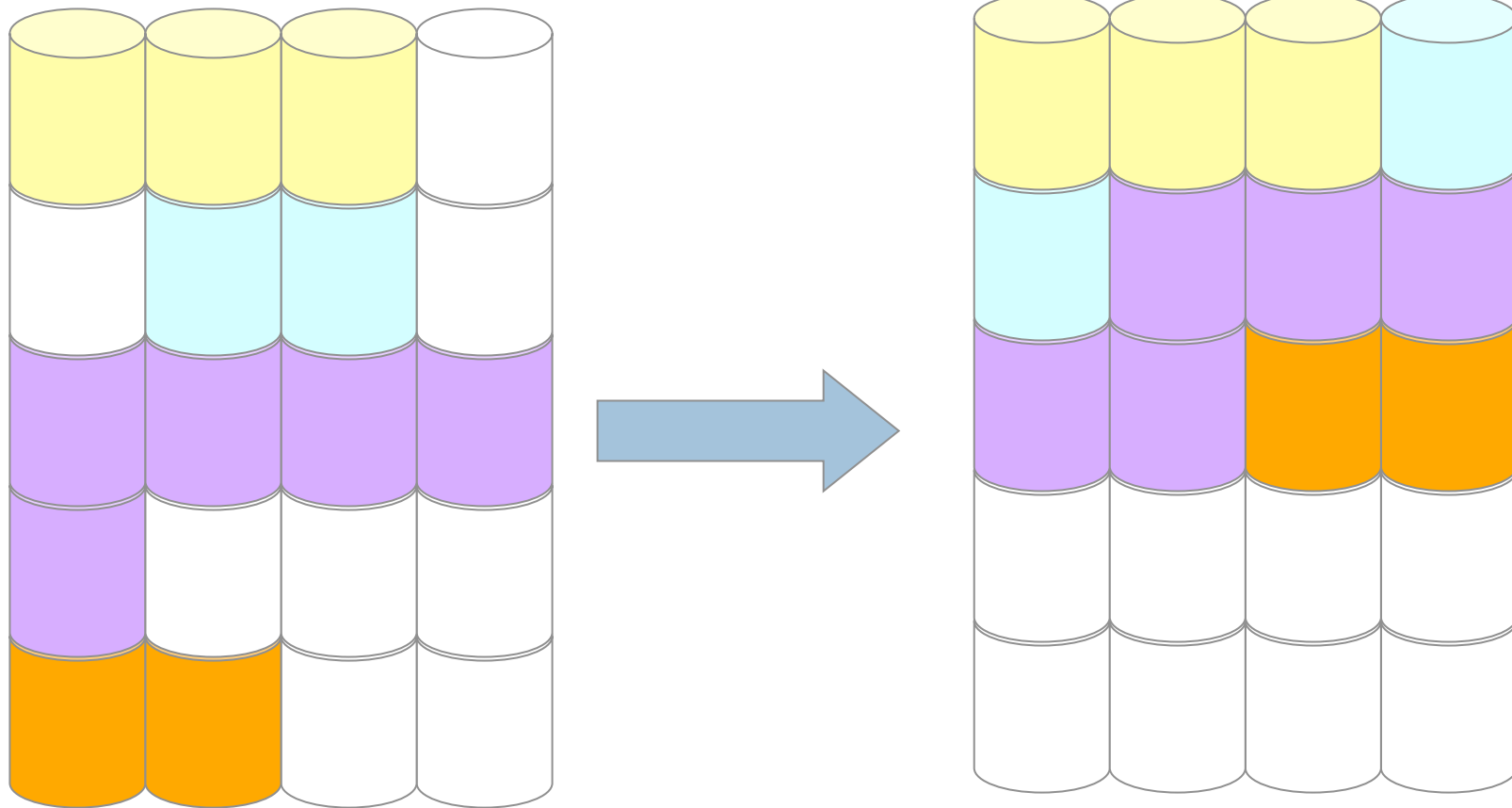
Moveu 4 blocos



Moveu 2 blocos



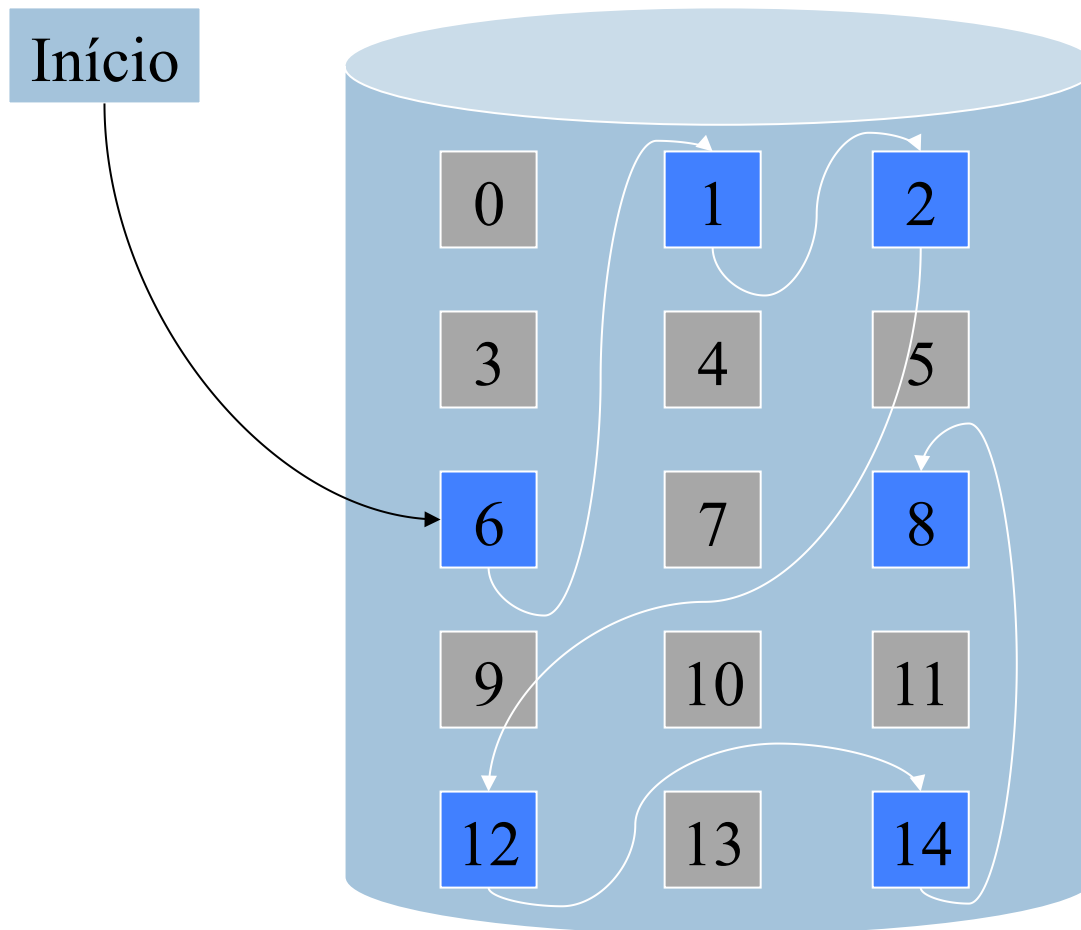
Estratégias de desfragmentação



Alocação Encadeada

- ❑ O arquivo é organizado como um conjunto de blocos ligados no disco, independente de sua localização física e cada um deve possuir um ponteiro para o bloco seguinte.
- ❑ O que ocorre neste método é a fragmentação de arquivos (quebra do arquivo em diversos pedaços denominados extents) o que aumenta o tempo de acesso ao arquivo, pois o disco deve deslocar-se diversas vezes para acessar todas as extents.
- ❑ É necessário que o disco seja desfragmentado periodicamente, esta alocação só permite acesso seqüencial e desperdiça espaço nos blocos com armazenamento de ponteiros.

Alocação Encadeada



Alocação por lista encadeada

57

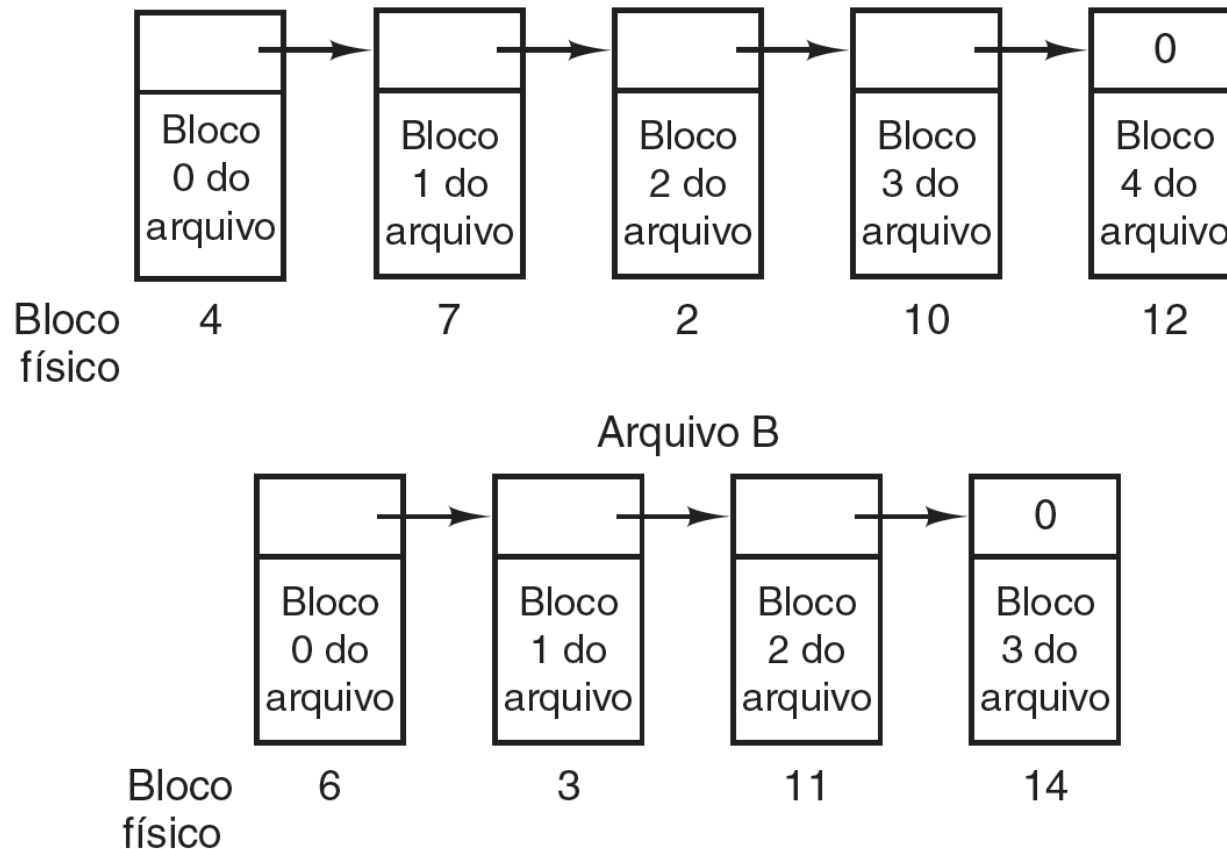
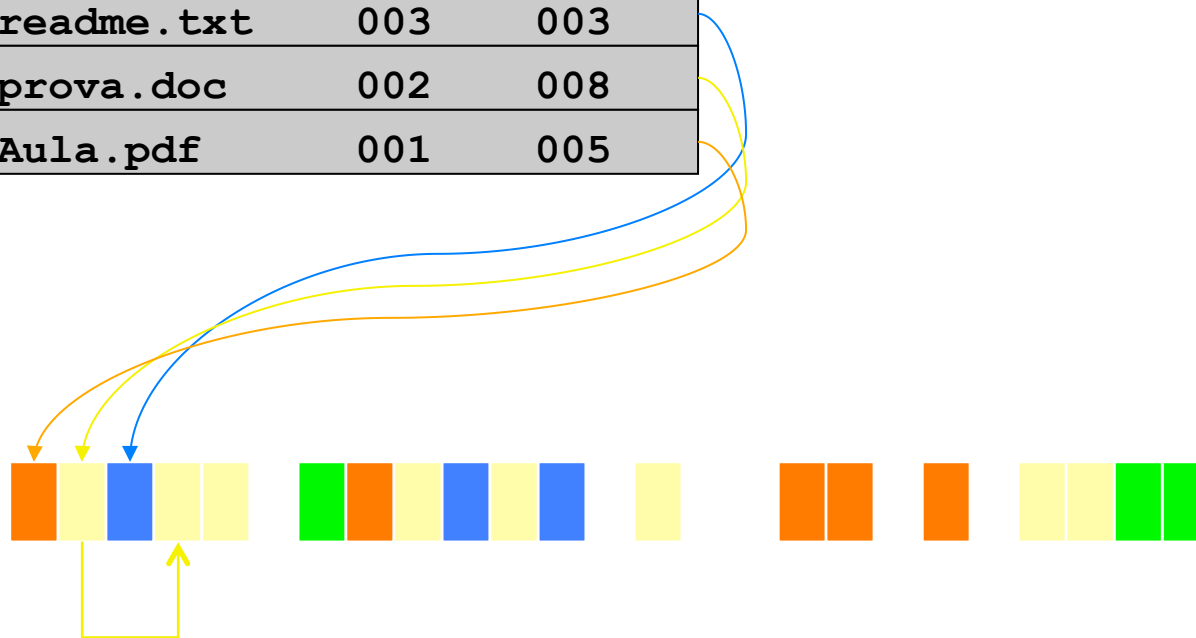


Figura 4.9 Armazenamento de um arquivo como uma lista encadeada de blocos de disco.

Alocação por lista encadeada

arquivo	inicio	#blocos
readme.txt	003	003
prova.doc	002	008
Aula.pdf	001	005



Alocação por lista encadeada

59

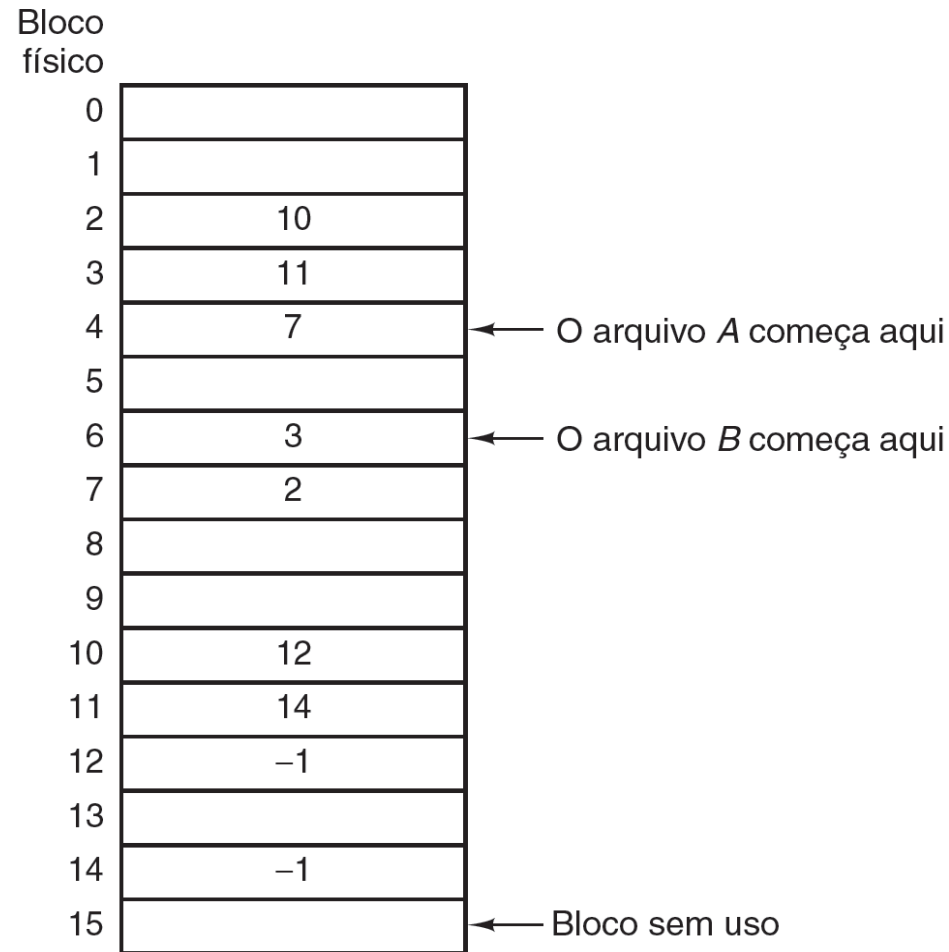


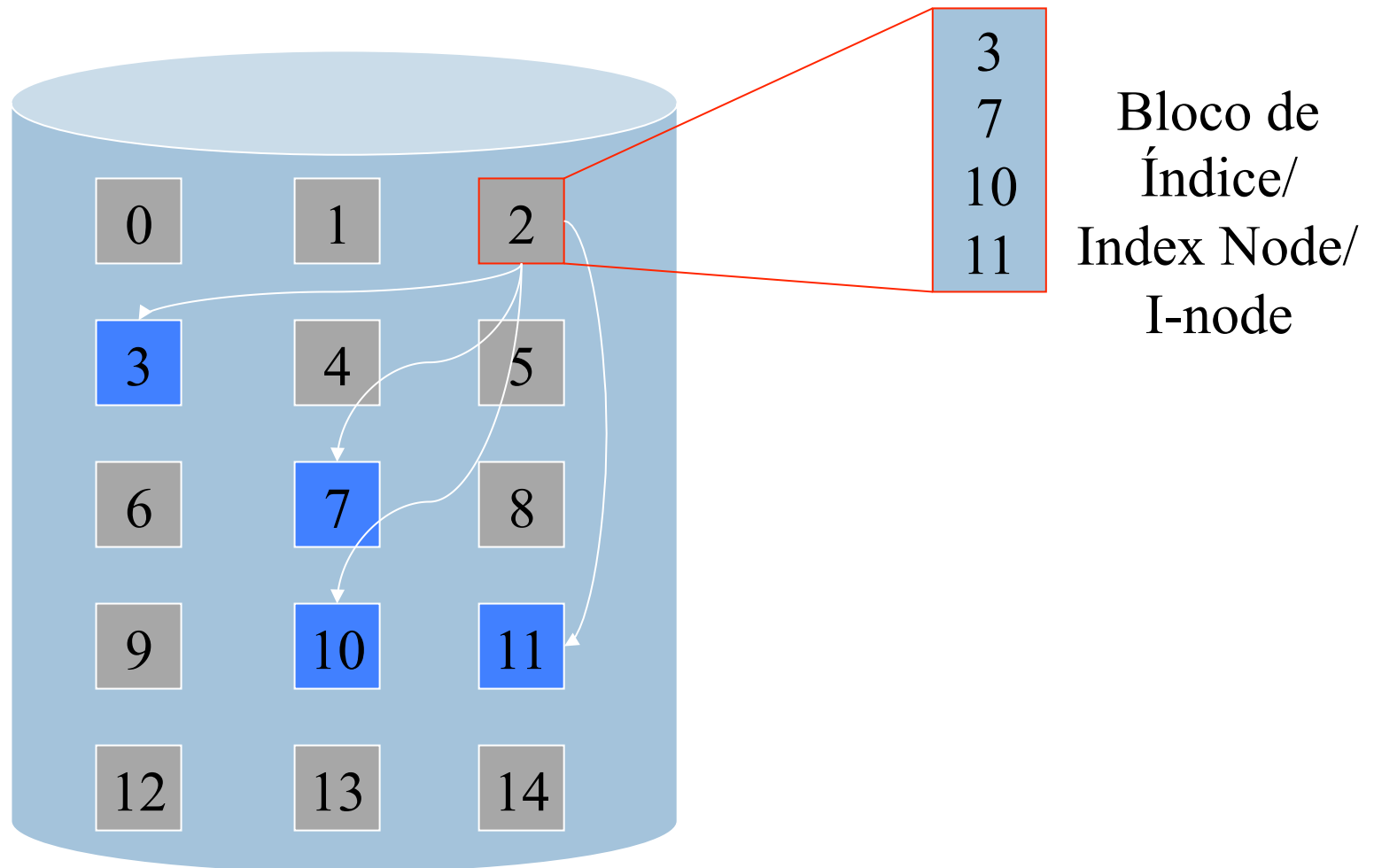
Figura 4.10 Alocação por lista encadeada usando uma tabela de alocação de arquivos na memória principal.

Alocação Indexada



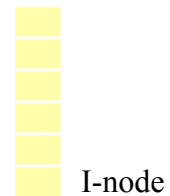
- O princípio desta técnica é manter os ponteiros de todos os blocos de arquivos em uma única estrutura denominada bloco de índice.
- Além de permitir o acesso direto aos blocos do arquivo, não utiliza informações de controle nos blocos de dados como existe na alocação encadeada.

Alocação Indexada



Alocação Indexada

arquivo	inicio	#blocos
readme.txt	010	003
prova.doc	002	008
Aula.pdf	017	005



Fragmentação interna



- Arquivos são alocados em blocos:
 - Os blocos têm tamanho fixo.
 - Entre 512 bytes e 8 Kbytes.
 - Um bloco não pode ser alocado parcialmente.
- Se usarmos blocos de 4096 bytes:
 - um arquivo de 5700 bytes ocupará 2 blocos.
 - 2492 bytes serão perdidos no último bloco.
- Em média, perde-se 1/2 bloco por arquivo.

Tamanho dos blocos



- A escolha do tamanho dos blocos é importante para a eficiência do sistema.

- **Blocos pequenos:**
 - menor perda por fragmentação interna
 - mais blocos por arquivo: maior custo de gerência

- **Blocos grandes:**
 - maior perda por fragmentação interna
 - menos blocos por arquivo: menor custo de gerência

Proteção de Acesso

- Considerando que os meios de armazenamento são **compartilhados** é necessário ter mecanismos de proteção para garantir a proteção de arquivos e diretórios.
- Qualquer sistema de arquivos deve possuir mecanismos próprios para proteger o acesso as informações gravadas.
- O tipo de acesso é mediante **concessão ou não** de acessos que podem ser realizados como a **leitura** (read), **gravação** (write), **execução** (execute) e **eliminação** (delete).

Proteção de Acesso



- Há diferenças entre o controle de acesso a diretórios e arquivos.
- O controle da criação/eliminação de arquivos nos diretórios, visualização do seu conteúdo e eliminação do próprio diretório são operações que também devem ser protegidas.
- Existem diferentes mecanismos e níveis de proteção e para cada tipo de sistema um modelo é mais adequado do que o outro.

Tipos de Acesso

Acesso	Descrição
Leitura	Qualquer tipo de operação em que o arquivo possa ser visualizado, como a exibição de seu conteúdo, edição ou cópia de um novo arquivo
Gravação	Alteração no conteúdo do arquivo, como inclusão ou alteração de registros.
Execução	Associado a arquivos executáveis ou arquivos de comandos, indicando o direito de execução do arquivo.
Eliminação	Permissão para se eliminar um arquivo.

Senha de Acesso



- É bastante simples e se resume ao usuário ter conhecimento da senha e a liberação do acesso ao arquivo concedida pelo sistema.
- Cada arquivo possui apenas uma senha, o acesso é liberado ou não na sua totalidade.
 - ▣ Não é possível determinar quais tipos de operações podem ou não ser concedidas.
 - ▣ Outra desvantagem é a dificuldade de compartilhamento já que todos os demais usuários deveriam ter conhecimento da senha.

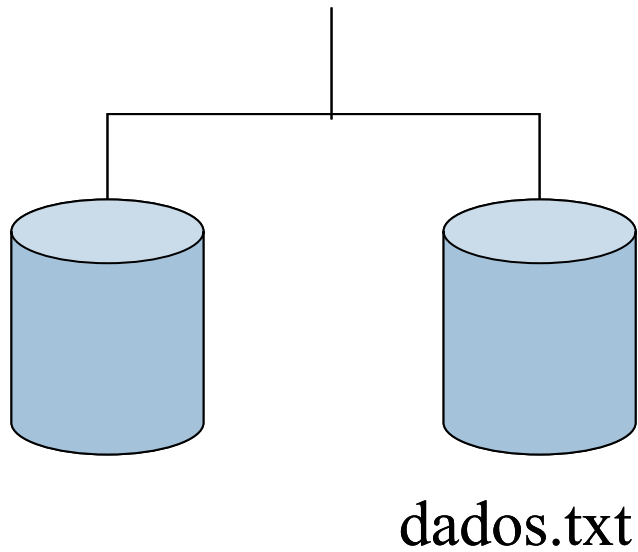
Grupos de Usuários



- Tem como princípio a associação de cada usuário do sistema a um grupo. Os usuários são organizados com o objetivo de compartilhar arquivos entre si.
- Implementa três tipos de proteção: **owner** (dono), **group** (grupo) e **all** (todos) e na criação do arquivo é especificado quem e o tipo de acesso aos três níveis de proteção.
- Em geral, somente o dono ou usuários privilegiados é que podem modificar a proteção dos arquivos.

Grupos de Usuários

70



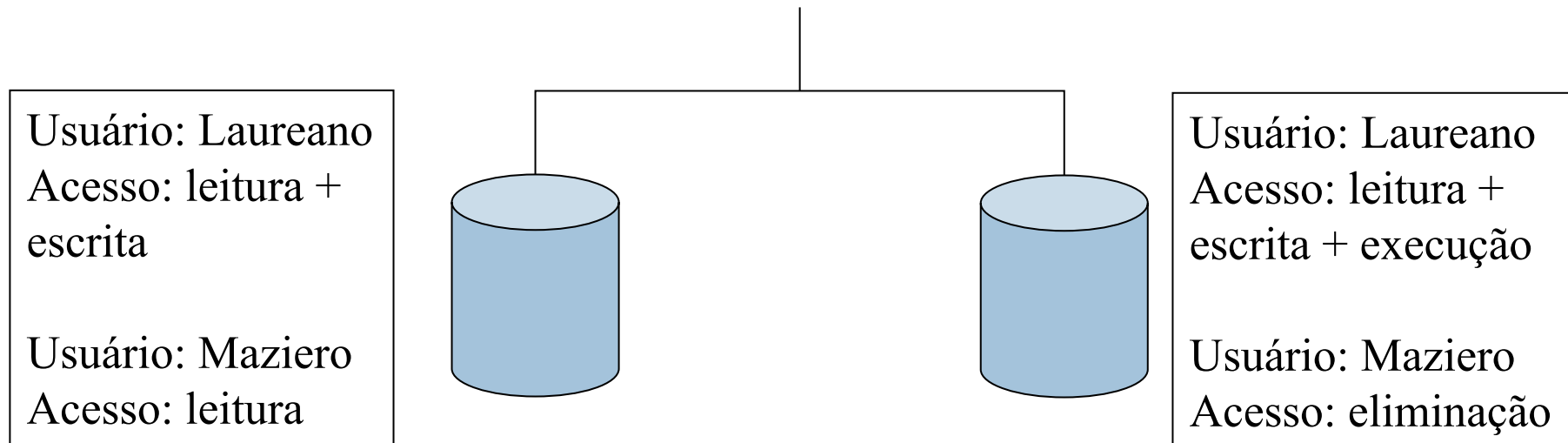
Nível de proteção	Tipo de Acesso
Owner	Leitura Escrita Execução Eliminação
Group	Leitura
All	--

Lista de Controle de Acesso

- **Access Control List** – ACL consiste em uma lista associada a cada arquivo onde são especificados quais os usuários e os tipos de acesso permitidos.
- O tamanho desta estrutura pode ser bastante extenso se um arquivo tiver seu acesso compartilhado por diversos usuários.
- Existe um overhead adicional devido a pesquisa seqüencial que o sistema deverá realizar na lista sempre que solicitado.
- É possível encontrar tanto a proteção por grupos de usuários quanto pela lista de acesso oferecendo uma maior flexibilidade ao mecanismo de proteção.

Lista de Controle de Acesso

72



Implementação de Caches

- ❑ O **acesso a disco** é bastante **lento** ao comparado a memória principal e este é o fator para que as operações de E/S serem um problema ao desempenho do sistema.
- ❑ Com o objetivo de minimizar este problema, a maioria dos sistemas operacionais implementa a técnica de **buffer cache** onde o sistema reserva uma **área na memória** para que se tornem disponíveis caches utilizados em operações de acesso a disco.
- ❑ Quando uma operação é realizada o sistema **procura no cache** a **informação** e caso não encontre, ele **busca no disco** e depois **atualiza a buffer cache**.

Implementação de Caches

- Como existe limite para o tamanho do cache o sistema adota políticas de substituição como o FIFO (First in First out) ou a LRU (Least Recently Used).
- No caso de dados permanecerem por um longo tempo na memória a ocorrência com problemas de energia pode resultar na perda de tarefas já executadas e consideradas salvas em disco.
- Existem duas maneiras de tratar deste problema:
 - ▣ O sistema pode possuir uma **rotina** que **executa**, em intervalos de tempo, **atualizações em disco** de todos os blocos modificados no cache.
 - ▣ Uma segunda alternativa é que **toda vez que** um bloco do cache for **modificado**, realizar uma **atualização** no disco (*write-through caches*).

Implementação de Caches

- Podemos concluir que a primeira técnica implica em menor quantidade de operações de E/S porém o risco de perda de dados é maior:
 - Pois pode ocorrer que dados atualizados de um arquivo ainda no cache sejam perdidos na falta de energia.
- Isso já não acontece nos caches tipo *write-through* porém existe um aumento considerável nas operações de E/S o que o torna menos eficiente.
- A maioria dos sistemas utiliza a primeira técnica.