

SISTEMAS  
OPERACIONAIS  
MODERNOS

3ª EDIÇÃO

# Sistema de Arquivos

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Sistemas de arquivos

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 está usando a mesma
- Múltiplos processos devem ser capazes de acessar a informação **simultaneamente**

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

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 prevenir que um usuário não leia os dados de outro?
- Como saber quais blocos estão livres?

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

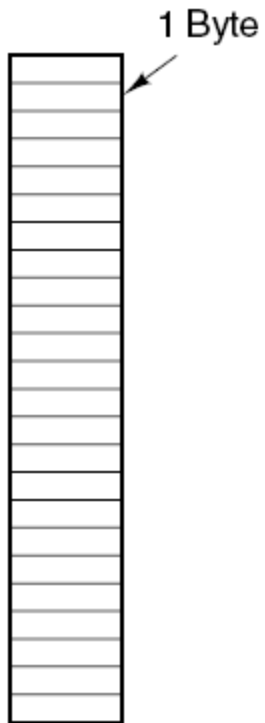
## Nomeação de arquivos

Extensão	Significado
.bak	Cópia de segurança
.c	Código-fonte de programa em C
.gif	Imagem no formato <i>Graphical Interchange Format</i>
.hlp	Arquivo de ajuda
.html	Documento em HTML
.jpg	Imagem codificada segundo padrões JPEG
.mp3	Música codificada no formato MPEG (camada 3)
.mpg	Filme codificado no padrão MPEG
.o	Arquivo objeto (gerado por compilador, ainda não ligado)
.pdf	Arquivo no formato PDF ( <i>Portable Document File</i> )
.ps	Arquivo PostScript
.tex	Entrada para o programa de formatação TEX
.txt	Arquivo de texto
.zip	Arquivo compactado

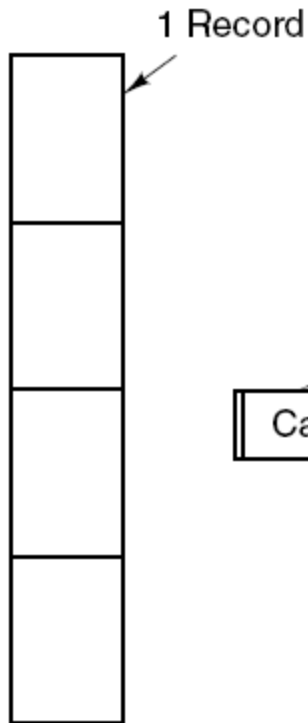
**Tabela 4.1** Algumas extensões comuns de arquivos.

# SISTEMAS OPERACIONAIS MODERNOS

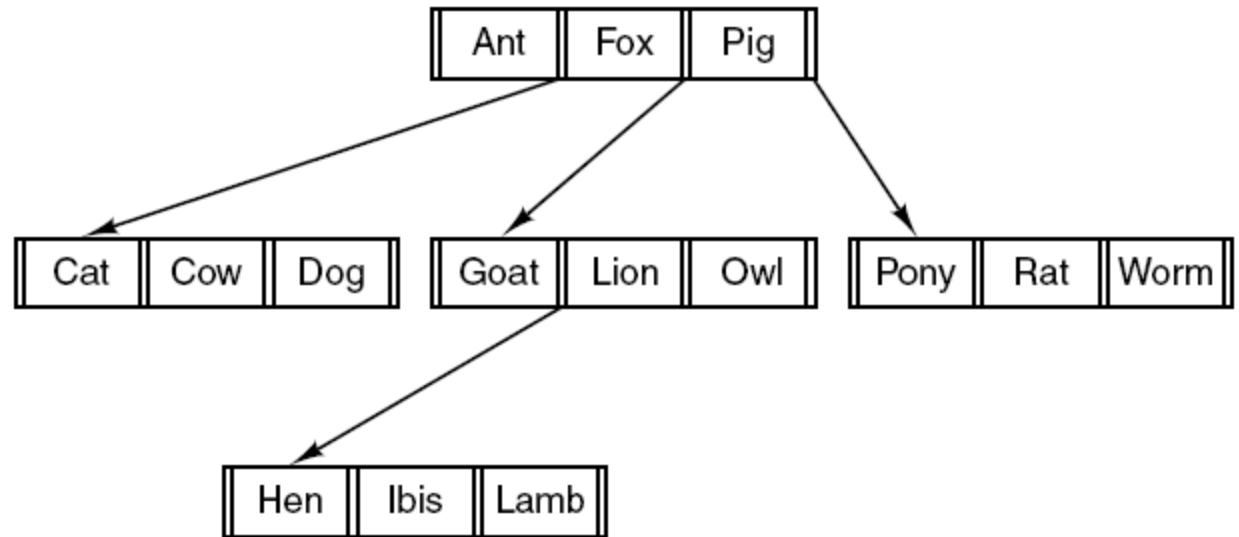
3ª EDIÇÃO



(a)



(b)



(c)

Três tipos de arquivos. (a) Sequência de bytes.  
(b) Sequência de registros. (c) Árvore.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Tipos de arquivo

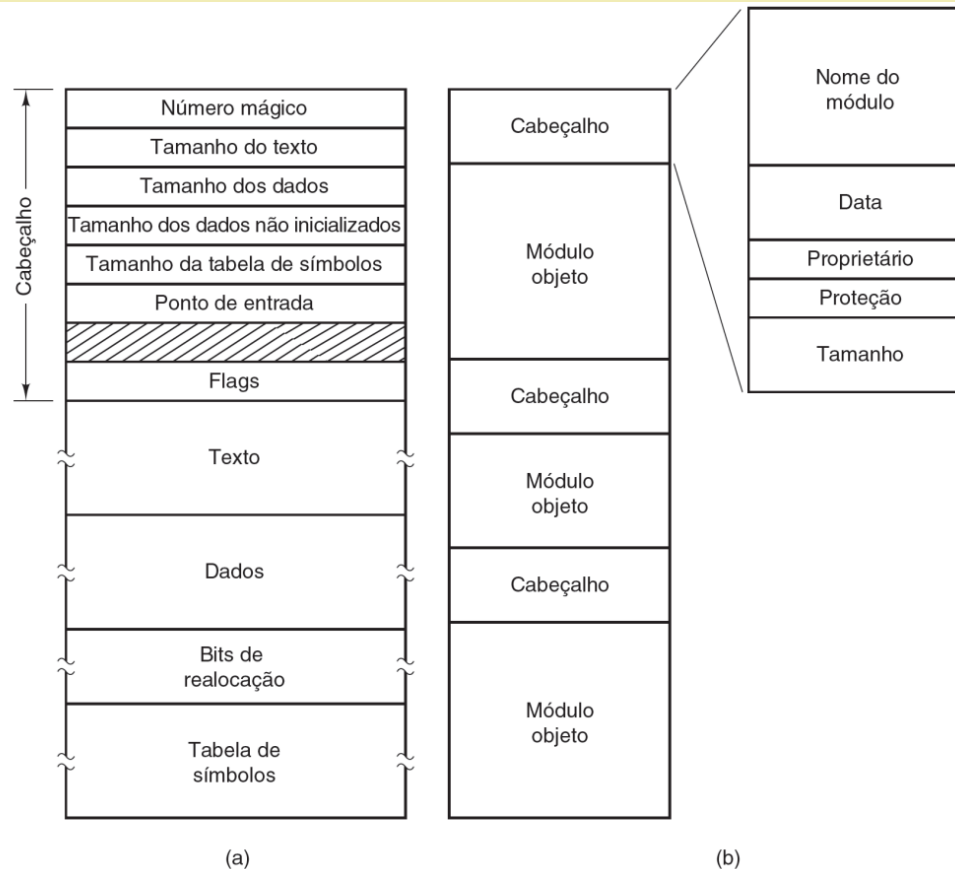


Figura 4.2 (a) Um arquivo executável. (b) Um arquivo.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Atributos de arquivos

Atributo	Significado
Proteção	Quem tem acesso ao arquivo e de que modo
Senha	Necessidade de senha para acesso ao arquivo
Criador	ID do criador do arquivo
Proprietário	Proprietário atual
Flag de somente leitura	0 para leitura/escrita; 1 para somente leitura
Flag de oculto	0 para normal; 1 para não exibir o arquivo
Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
Flag de arquivamento	0 para arquivos com backup; 1 para arquivos sem backup
Flag de ASCII/binário	0 para arquivos ASCII; 1 para arquivos binários
Flag de acesso aleatório	0 para acesso somente sequencial; 1 para acesso aleatório
Flag de temporário	0 para normal; 1 para apagar o arquivo ao sair do processo
Flag de travamento	0 para destravados; diferente de 0 para travados
Tamanho do registro	Número de bytes em um registro
Posição da chave	Posição da chave em cada registro
Tamanho do campo-chave	Número de bytes no campo-chave
Momento de criação	Data e hora de criação do arquivo
Momento do último acesso	Data e hora do último acesso do arquivo
Momento da última alteração	Data e hora da última modificação do arquivo
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número máximo de bytes no arquivo

■ Tabela 4.2 Alguns atributos possíveis de arquivos.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Operações com arquivos

Chamadas de sistemas mais comuns relacionadas aos arquivos:

- Create (criar)
- Delete (apagar)
- Open (abrir)
- Close (fechar)
- Read (ler)
- Write (escrever)
- Append (anexar)
- Seek (procurar)
- Get Attributes (conseguir atributos)
- Set Attributes (configurar atributos)
- Rename (renomear)



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Exemplo de um programa usando chamadas de sistemas para arquivos

```
/* Programa que copia arquivos. Verificação e relato de erros é mínimo.*/

#include <sys/types.h>           /* inclui os arquivos de cabeçalho necessários */
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]); /* protótipo ANS */

#define BUF_SIZE 4096           /* usa um tamanho de buffer de 4096 bytes */
#define OUTPUT_MODE 0700       /* bits de proteção para o arquivo de saída */

int main(int argc, char *argv[])
{
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if (argc != 3) exit(1);      /* erro de sintaxe se argc não for 3 */

    /* Abre o arquivo de entrada e cria o arquivo de saída */
    in_fd = open(argv[1], O_RDONLY); /* abre o arquivo de origem */
    if (in_fd < 0) exit(2);         /* se não puder ser aberto, saia */
    out_fd = creat(argv[2], OUTPUT_MODE); /* cria o arquivo de destino */
    if (out_fd < 0) exit(3);       /* se não puder ser criado, saia */
```

**(Continua)**

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Exemplo de um programa usando chamadas de sistemas para arquivos

**(Continuação)**

```
/* Laço de cópia */
while (TRUE) {
    rd_count = read(in_fd, buffer, BUF_SIZE); /* lê um bloco de dados */
    if (rd_count <= 0) break                    /* se fim de arquivo ou erro, sai do laço */
    wt_count = write(out_fd, buffer, rd_count); /* escreve dados */
    if (wt_count <= 0) exit(4);                 /* wt_count <= 0 é um erro */
}

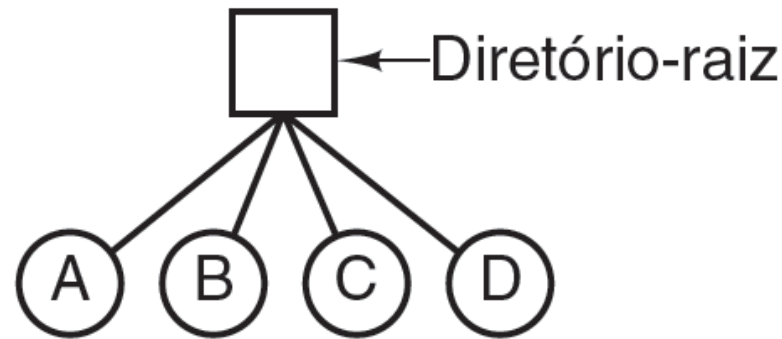
/* Fecha os arquivos */
close(in_fd);
close(out_fd);
if (rd_count == 0)                                /* nenhum erro na última leitura */
    exit(0);
else
    exit(5);                                       /* erro na última leitura */
}
```

**Figura 4.3** Um programa simples para copiar um arquivo.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

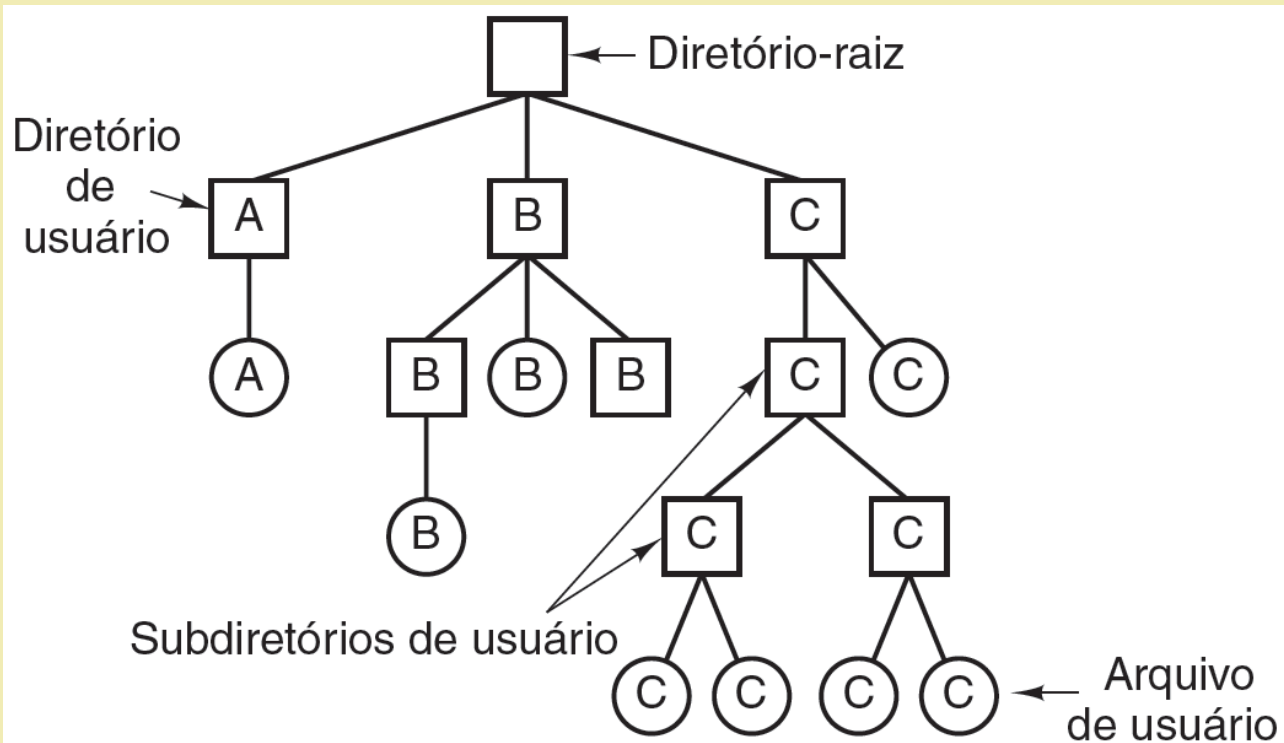
## Sistemas de diretórios hierárquicos



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

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO



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

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Operações com diretórios

Chamadas de sistema para gerenciamento de diretórios:

- Create (criar)
- Delete (apagar)
- Opendir (abrir diretório)
- Closedir (fechar diretório)
- Readdir (ler diretório)
- Rename (renomear)
- Link (ligar)
- Uplink

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Esquema do sistema de arquivos

Master Boot Record:  
Registro Principal do  
Boot, usado para  
iniciar o computador

Principais parâmetros  
do sistema de arquivo  
– ex. tipo do sistema  
de arquivos, número  
de blocos do sistema

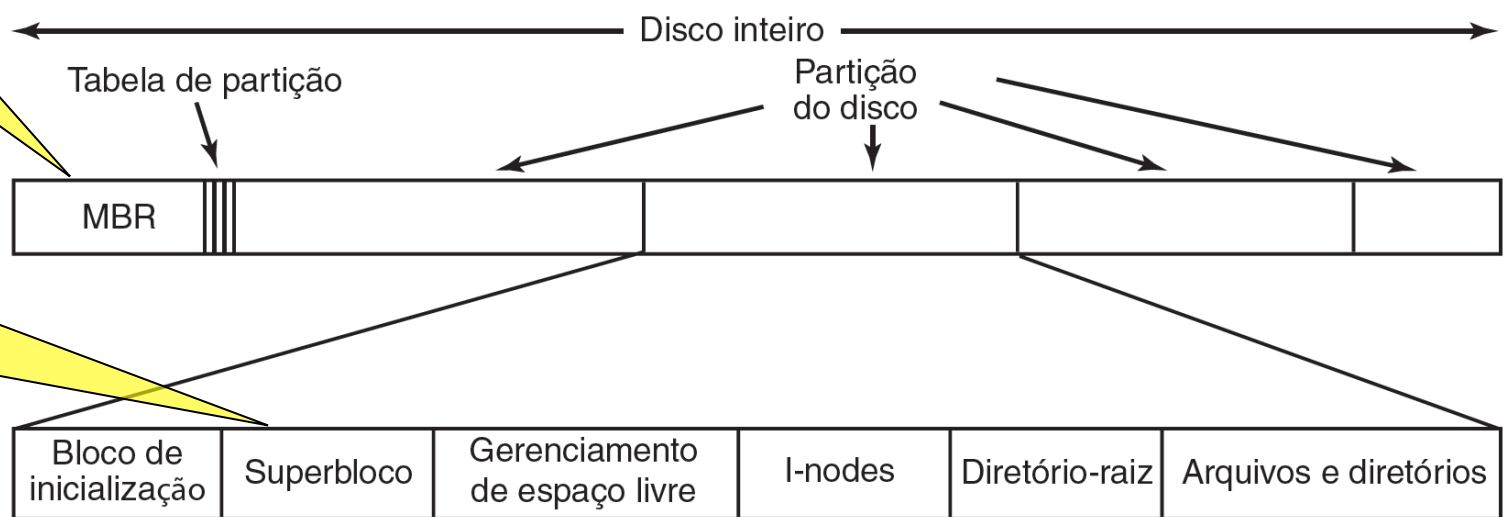
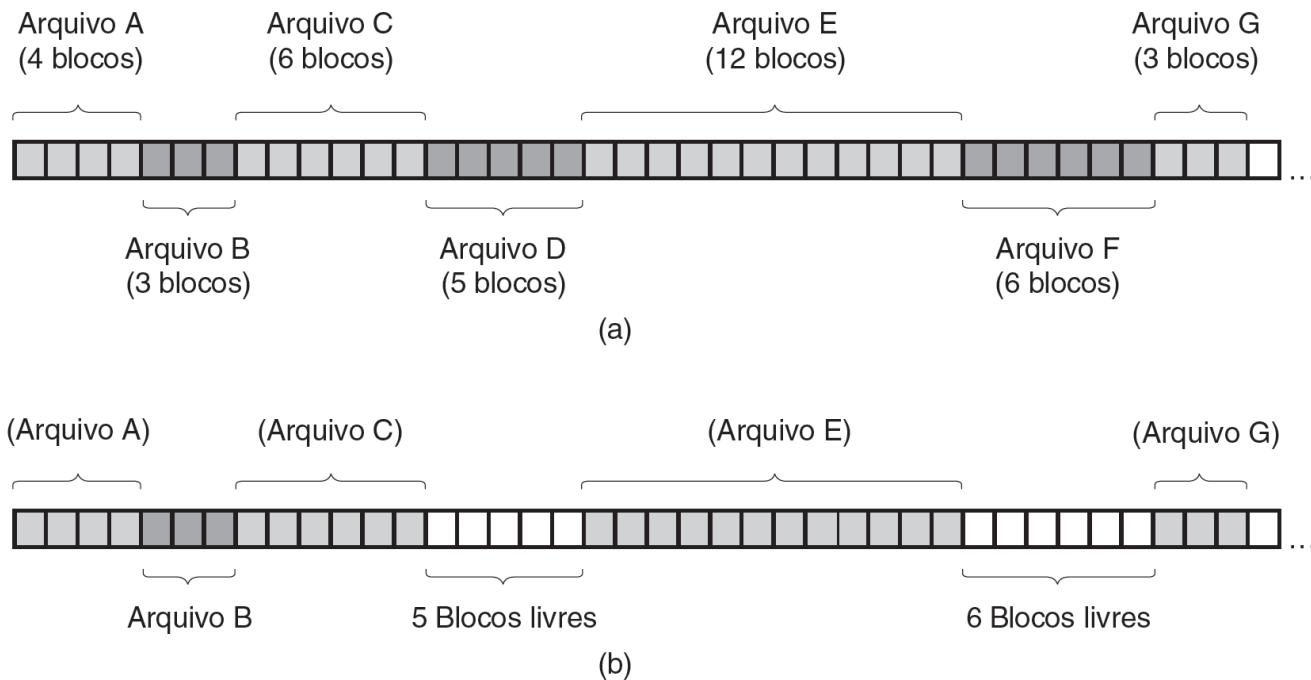


Figura 4.7 Uma organização possível para um sistema de arquivos.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Alocação contígua



**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

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.

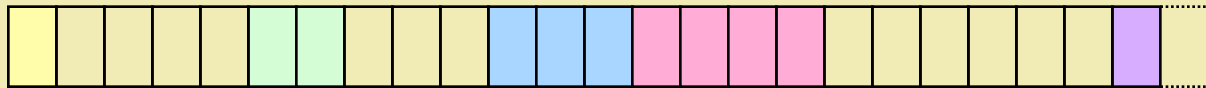


# SISTEMAS OPERACIONAIS MODERNOS

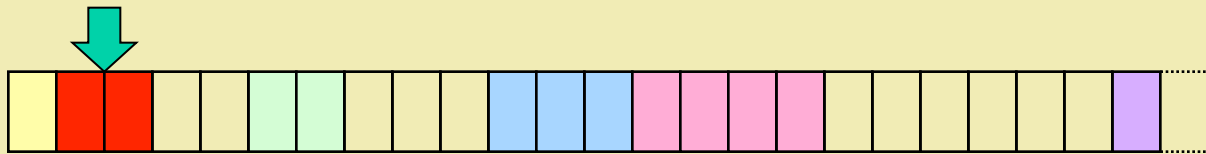
3ª EDIÇÃO

## Alocando um arquivo com 2 blocos

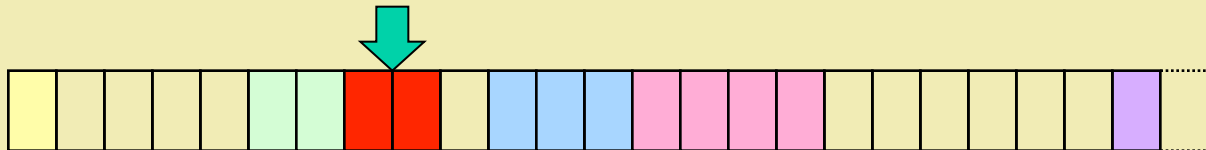
Situação inicial



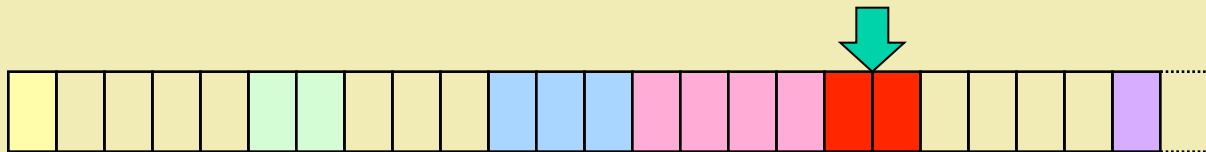
First-fit



Best-fit



Worst-fit



*Mais rápido*

*Menos desperdício*

*Mais expansível*

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

- Independente da estratégia utilizada, a alocação apresenta um problema chamado **fragmentação de espaços livres**
  - o problema pode se tornar crítico quando um disco possuir blocos livres disponíveis, porém sem um segmento contíguo onde o arquivo possa ser alocado
- Deve ser feita a **defragmentação** periodicamente 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

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

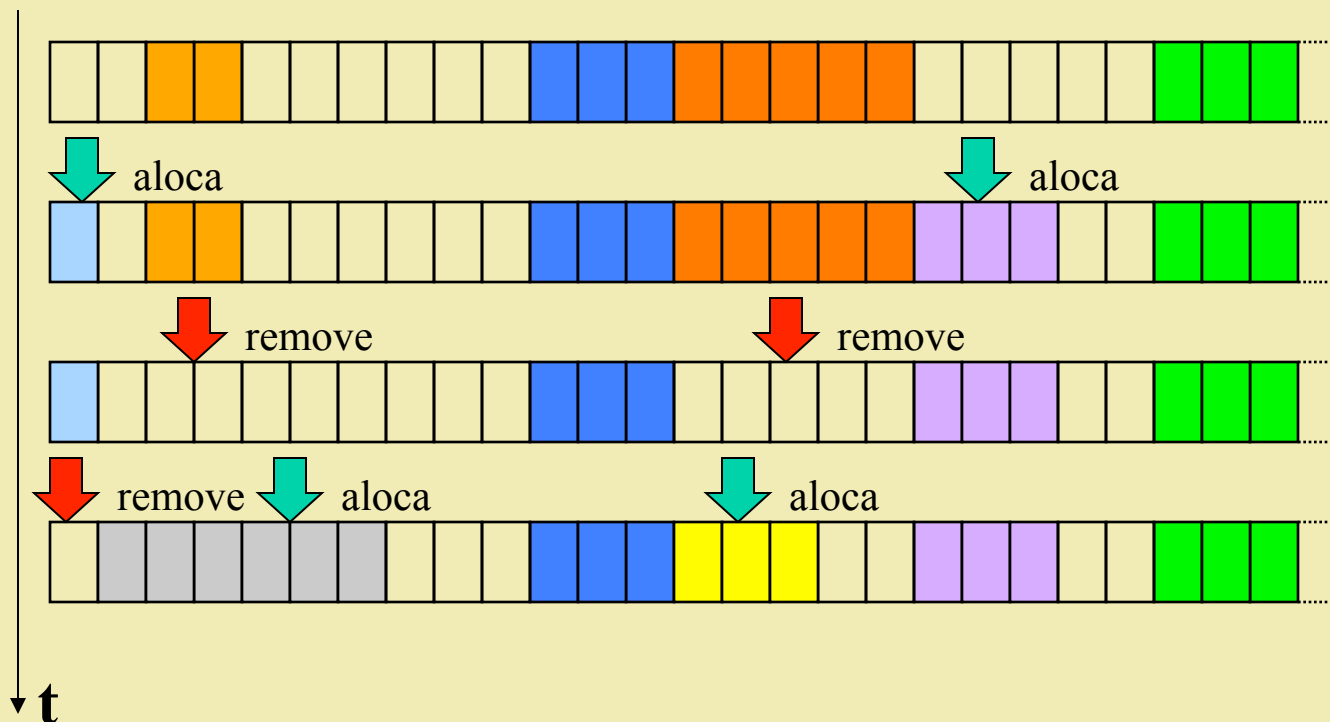
## Fragmentação externa

- 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 se torna difícil!!!**

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Evolução da fragmentação



Agora, como alocar um arquivo com 4 blocos ?

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Defragmentação

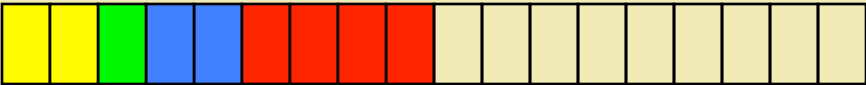
- Mover arquivos para **reagrupar os fragmentos** em espaços maiores
- Visa permitir alocar arquivos maiores
- Deve ser feita periodicamente
- Uso de **algoritmos para minimizar movimentação** de arquivos (eficiência)

# Estratégias de defragmentação

Situação inicial



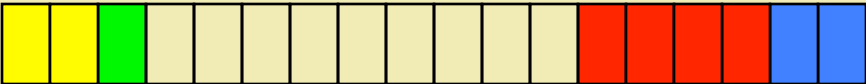
Moveu 6 blocos



Moveu 4 blocos



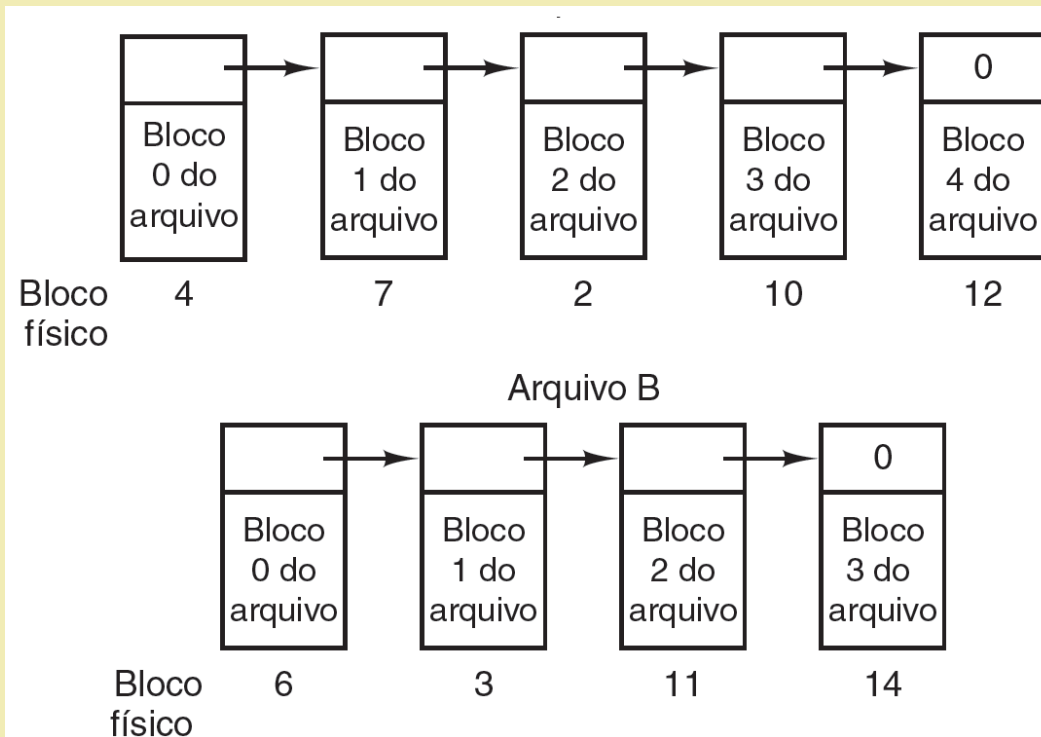
Moveu 2 blocos



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Alocação por lista encadeada

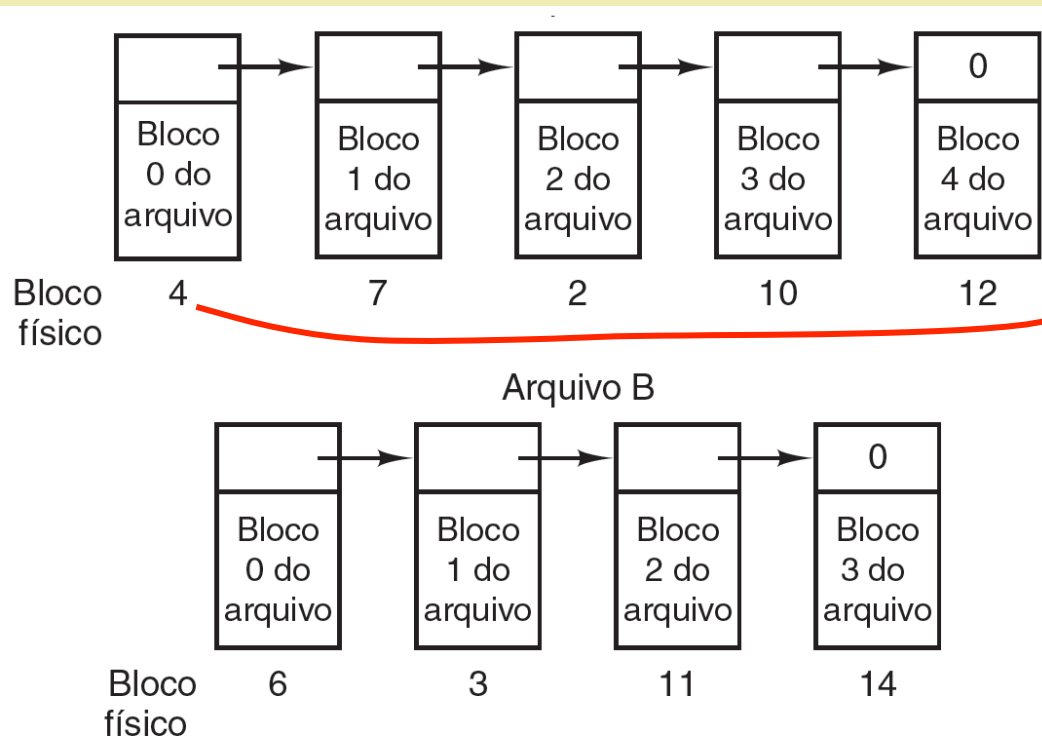


**Figura 4.9** Armazenamento de um arquivo como uma lista encadeada de blocos de disco.

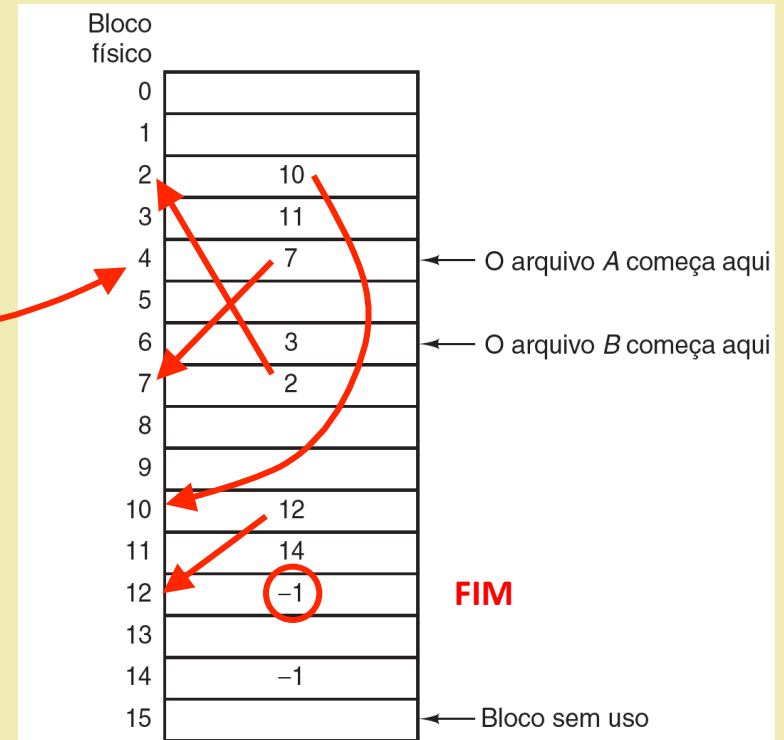
# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Alocação por lista encadeada usando uma tabela na memória



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



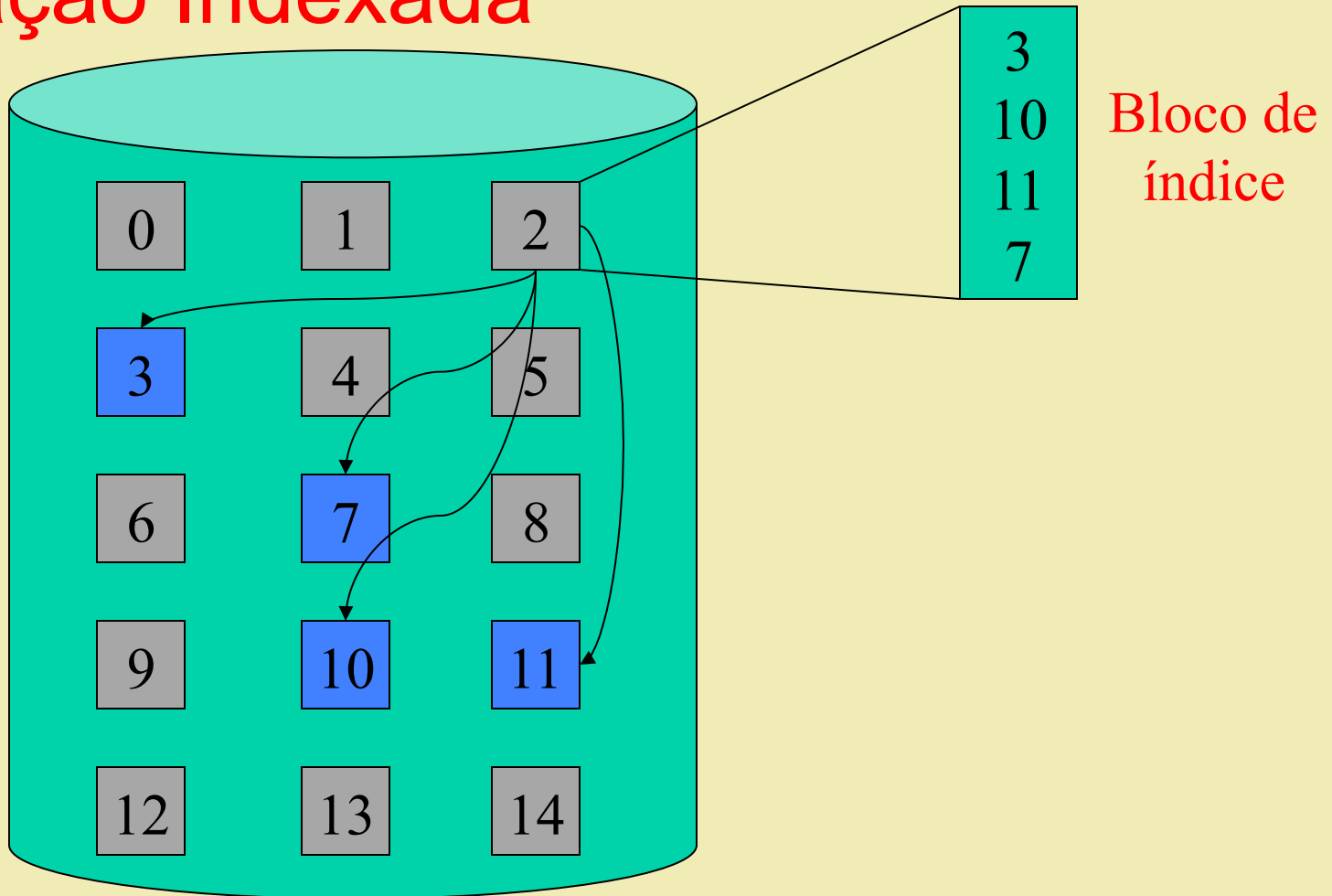
**Figura 4.10** Alocação por lista encadeada usando uma tabela de alocação de arquivos na memória principal.



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

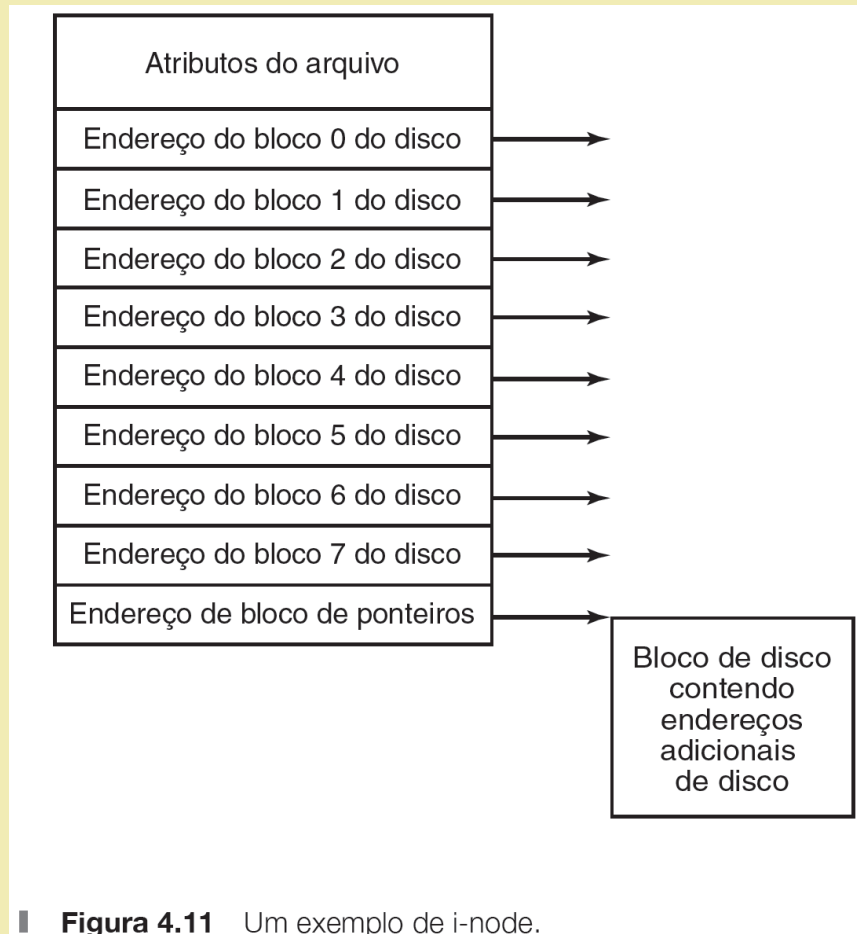
## Alocação Indexada



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## I-nodes



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## 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

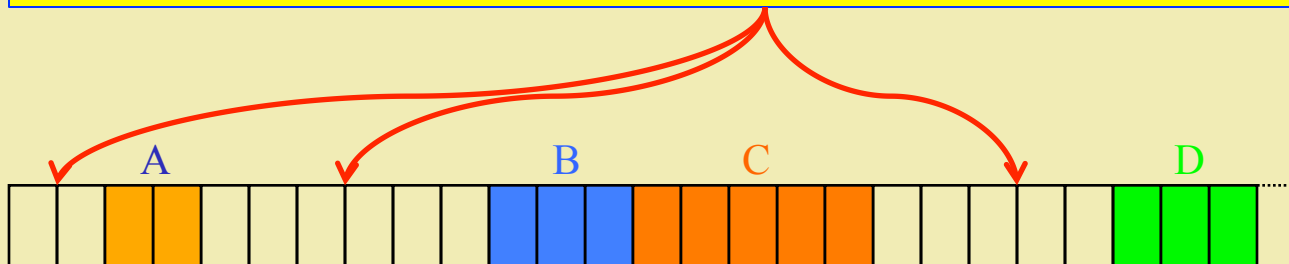
**Fragmentação externa:** espaços vazios entre blocos de arquivos

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

**Fragmentação externa: espaços vazios entre blocos de arquivos**

arquivos:



um arquivo de 5700 bytes ocupando 2 blocos de 4096 bytes – 2492 bytes são perdidos no último bloco

**Fragmentação interna: uso incompleto do espaço do último bloco de um arquivo**

# Tamanho dos blocos

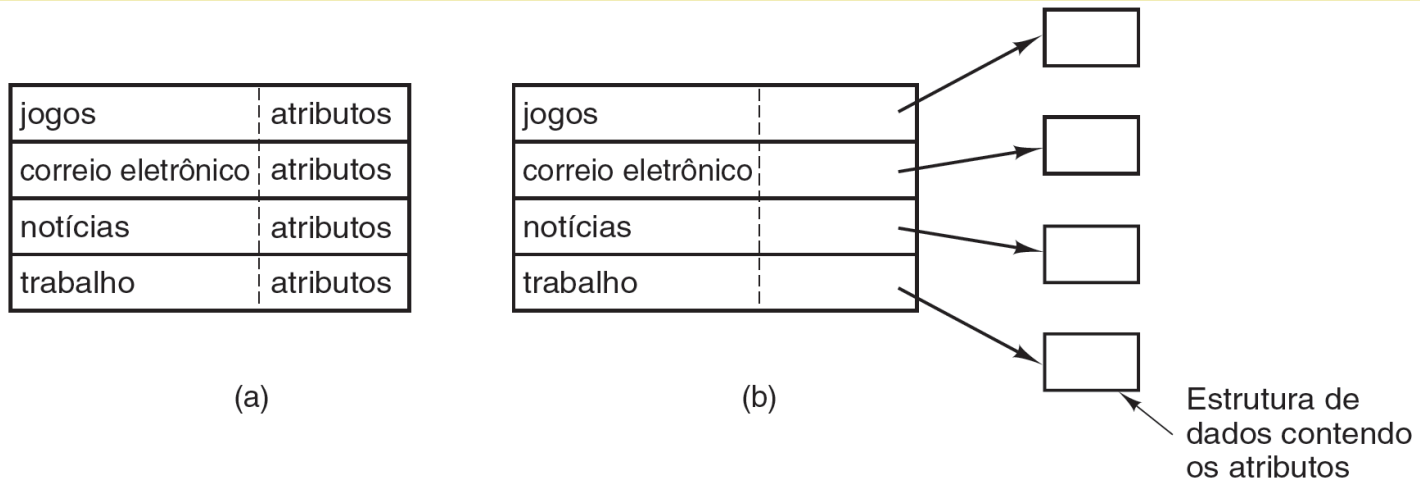
A escolha do tamanho dos blocos é importante para a eficiência do sistema

- **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

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

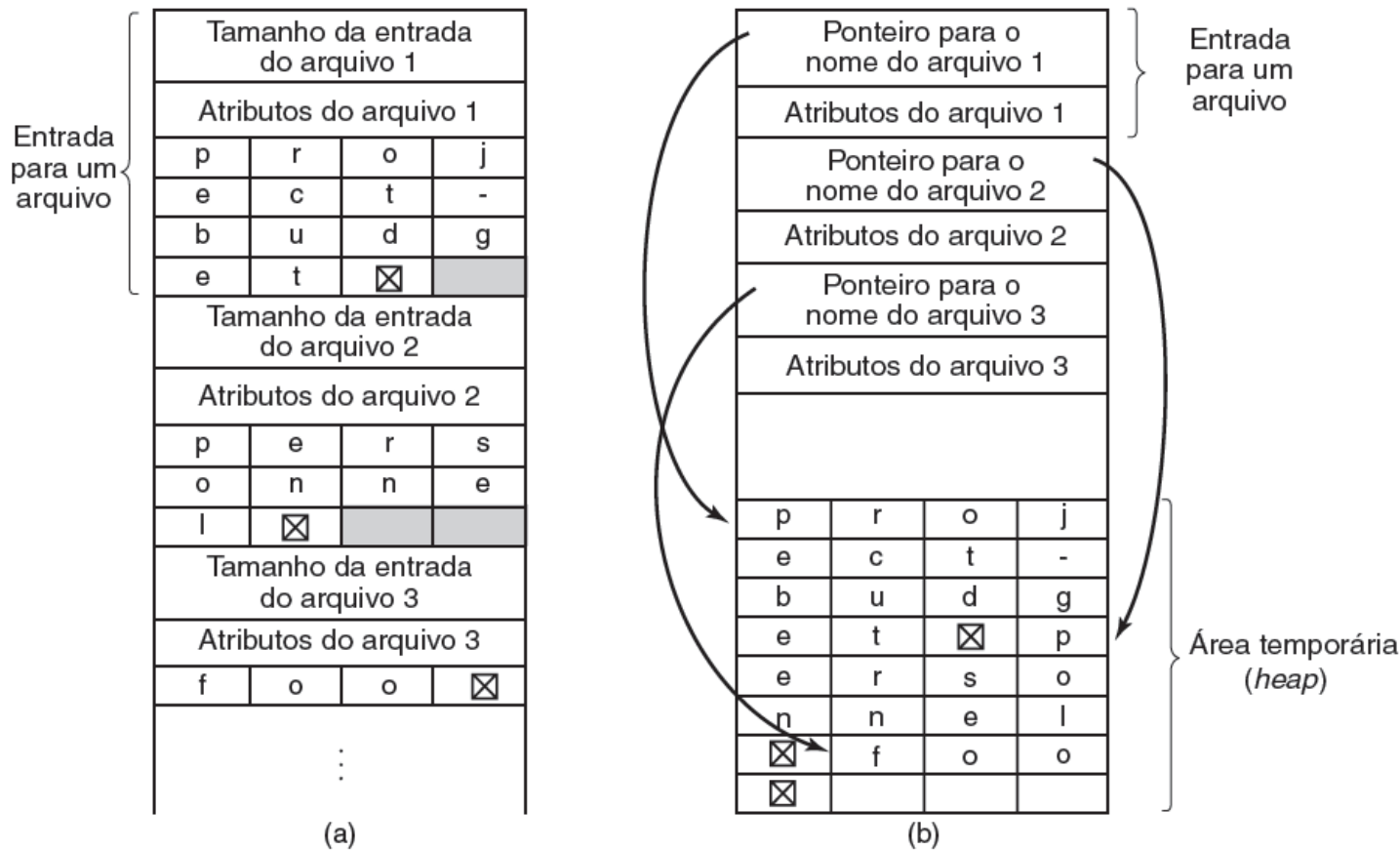
## Implementação de diretórios



**Figura 4.12** (a) Um diretório simples com entradas de tamanho fixo com os endereços de disco e atributos na entrada de diretório.  
(b) Um diretório no qual cada entrada se refere apenas a um i-node.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

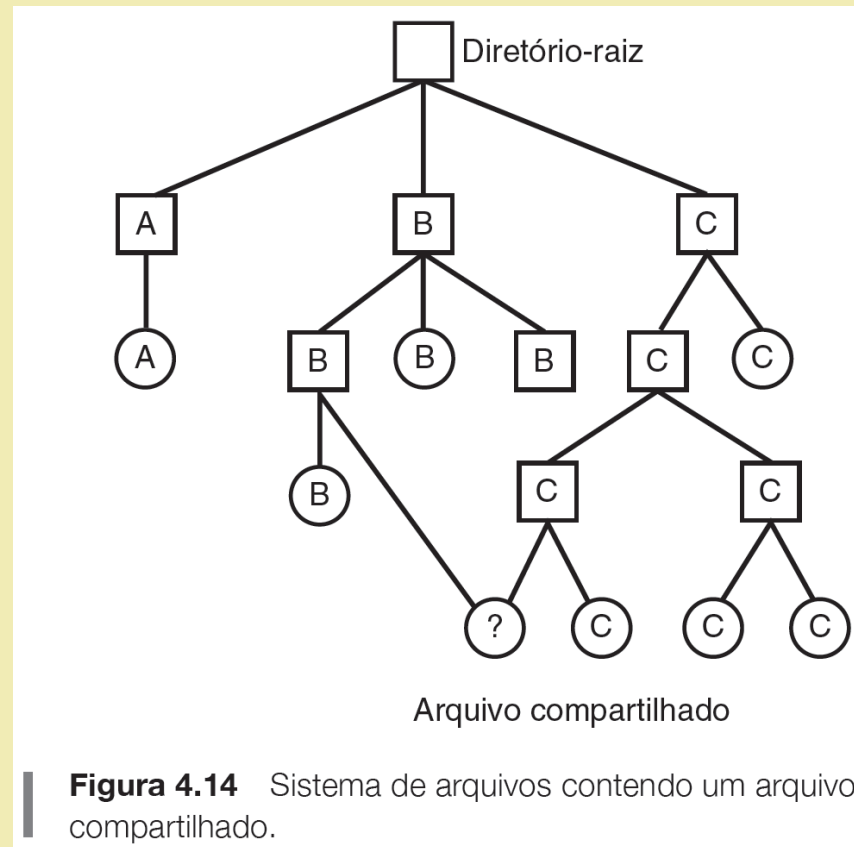


**Figura 4.13** Duas maneiras de gerenciar nomes de arquivos longos em um diretório. (a) Sequencialmente. (b) Em uma área temporária.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

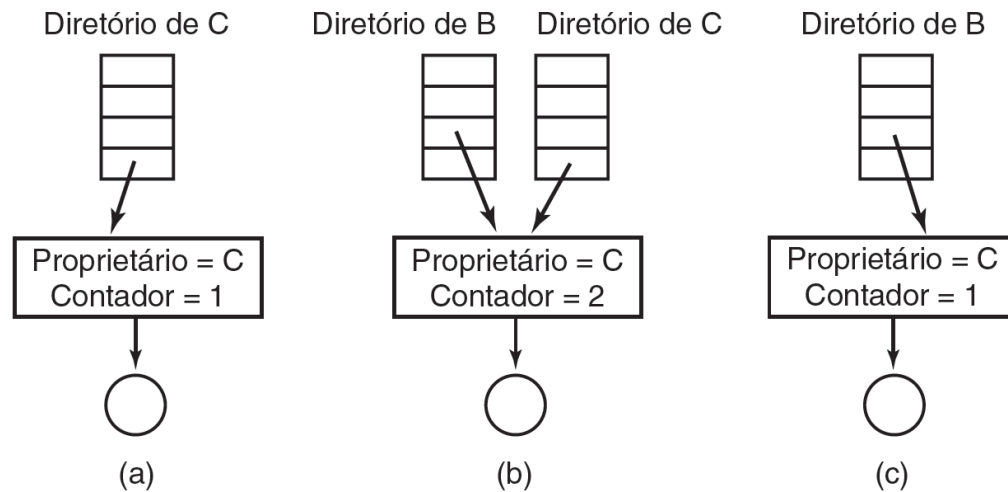
## Arquivos compartilhados





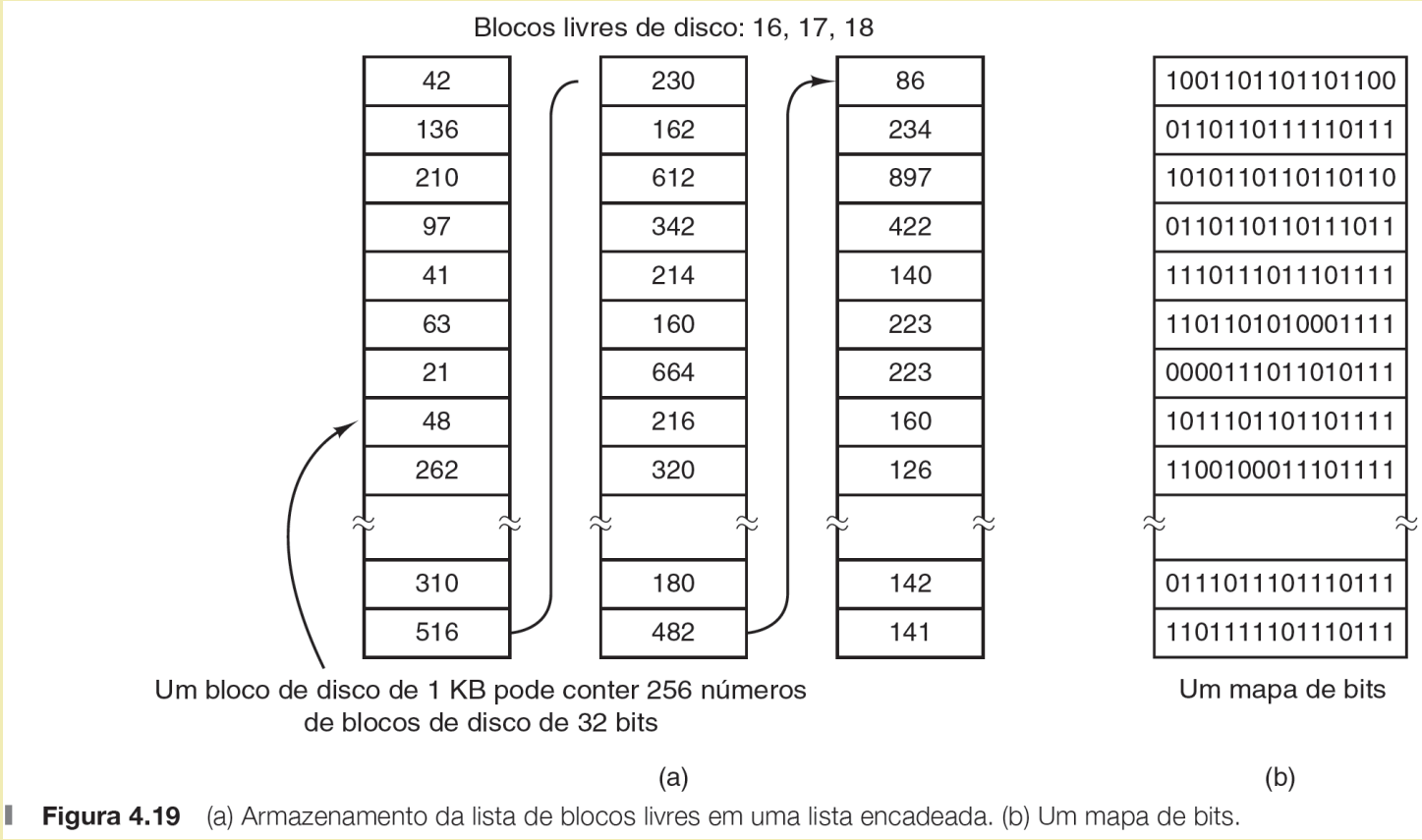
# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO



**Figura 4.15** (a) Situação antes da ligação. (b) Depois da criação da ligação. (c) Depois que o proprietário original remove o arquivo.

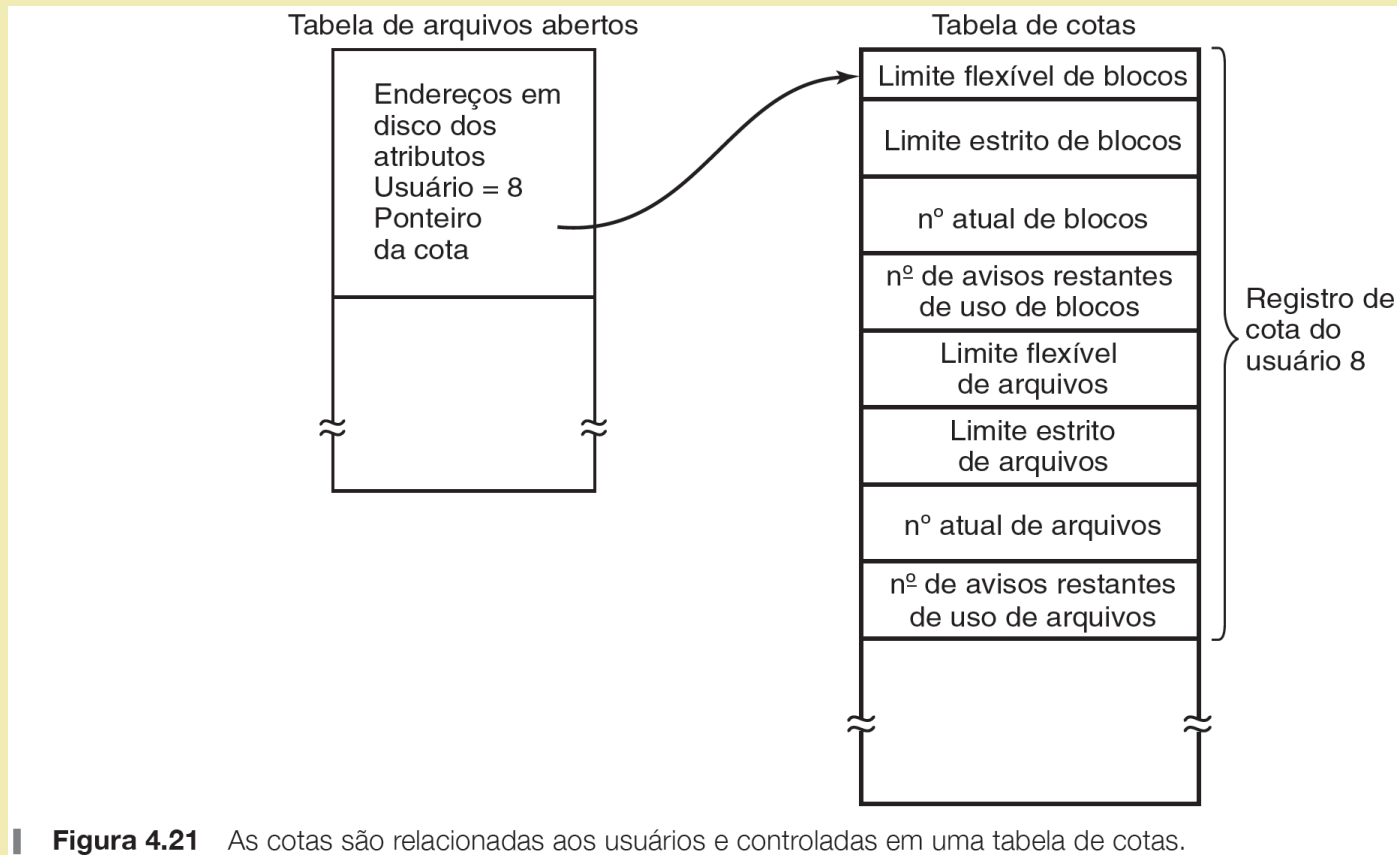
# Monitoramento dos blocos livres



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Cotas de disco



# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## O sistema de arquivos do MS-DOS

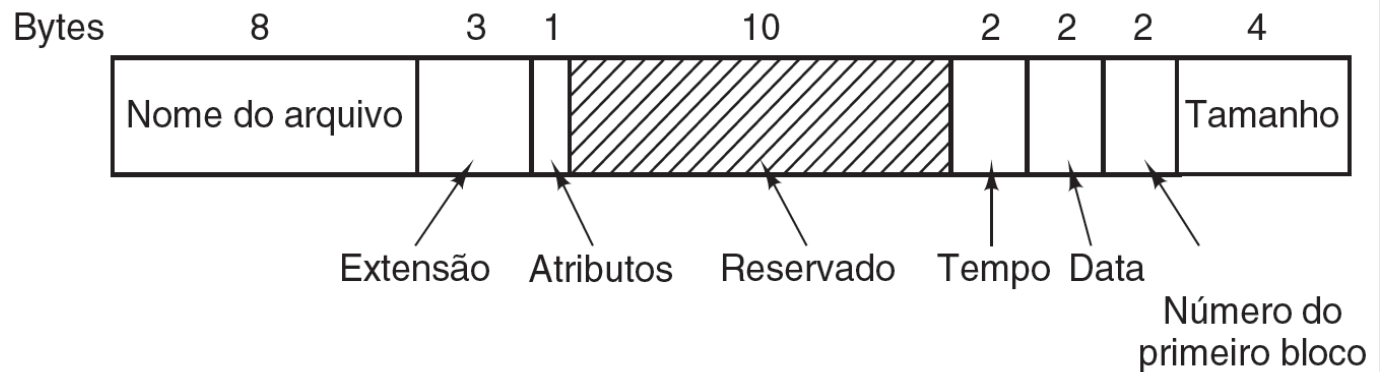
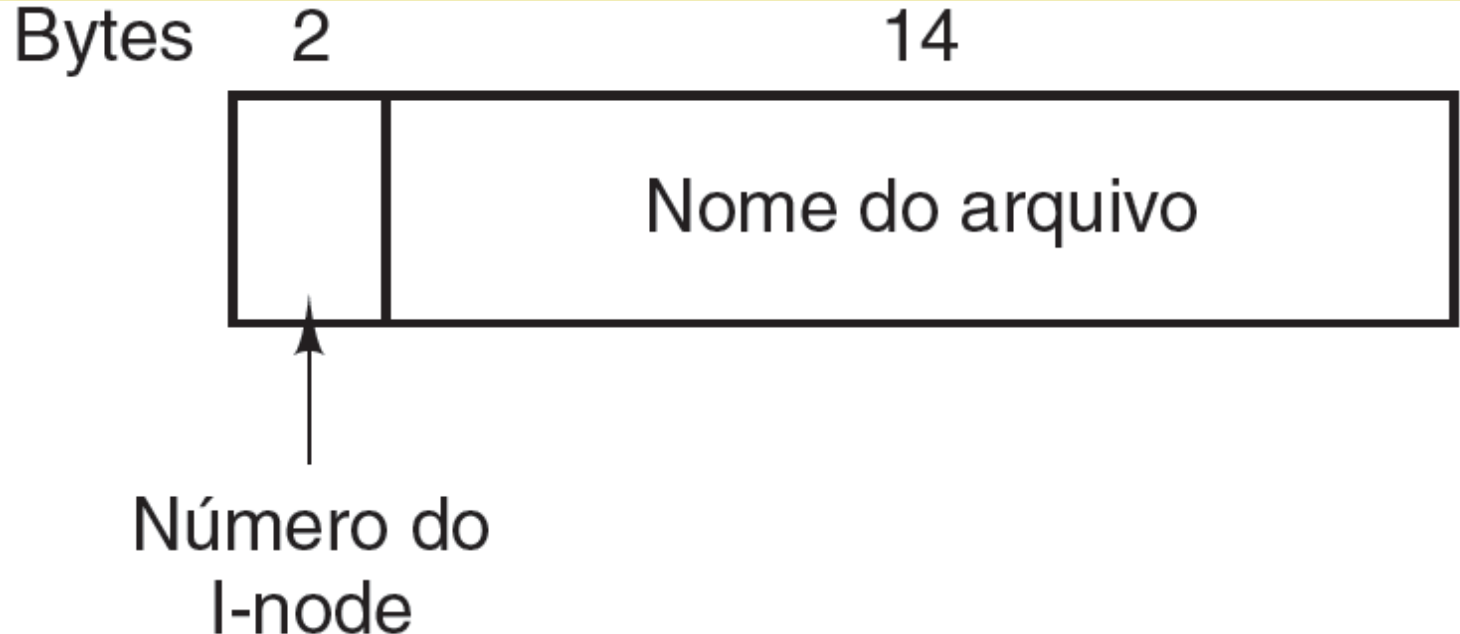


Figura 4.28 A entrada de diretório do MS-DOS.

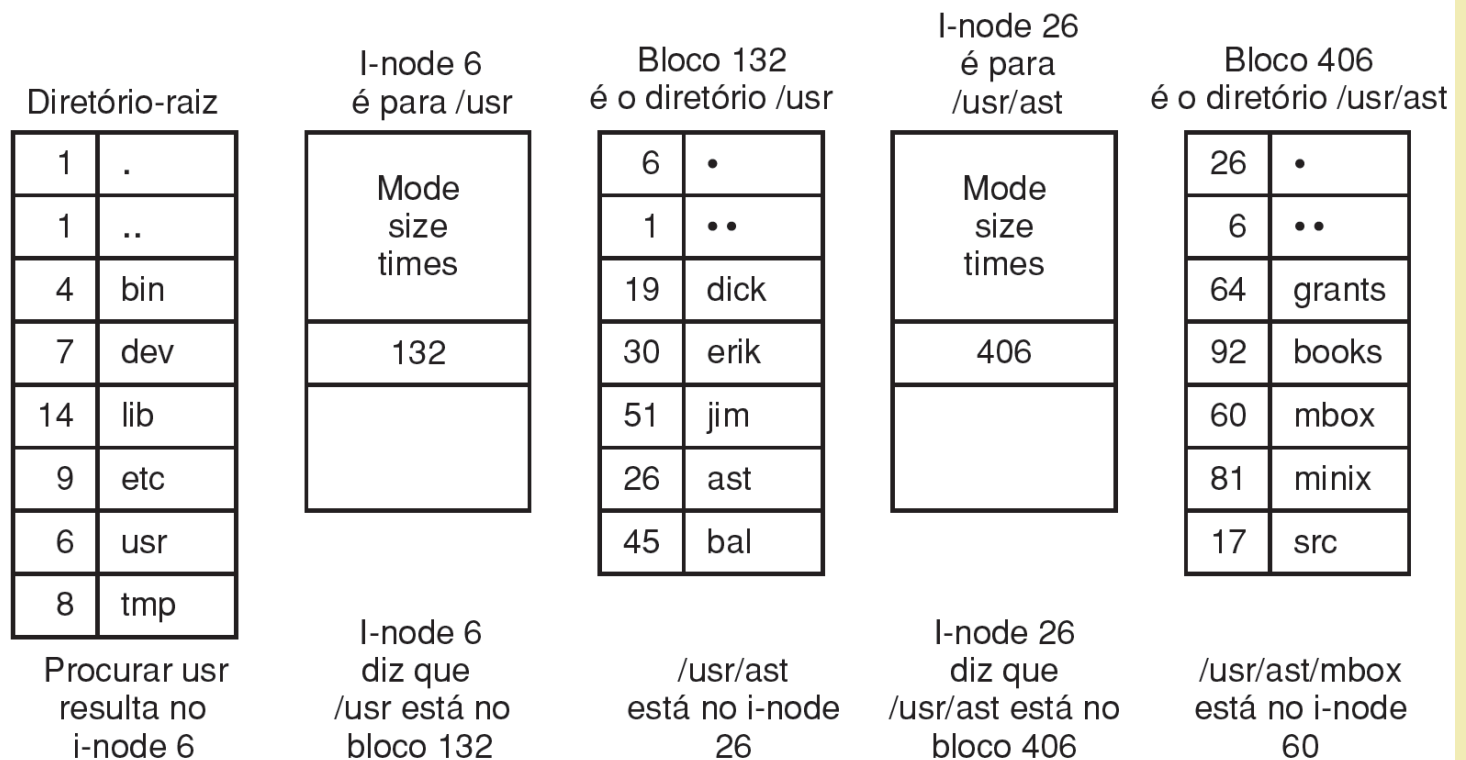
## O sistema de arquivos do UNIX V7



**Figura 4.29** Uma entrada de diretório do UNIX V7.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO



■ **Figura 4.31** Os passos para pesquisar em `/usr/ast/mbox`.

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Gerenciamento do Espaço em Disco

### **Considerações relevantes:**

- Tamanho do bloco: eficiência
- Monitoramento de blocos livres (ex. mapas de bits)
- Cotas de usuários

# SISTEMAS OPERACIONAIS MODERNOS

3ª EDIÇÃO

## Sistemas de arquivos

- Armazenamento persistente
- Grandes quantidades de informação e compartilhamento
- Arquivos e diretórios
- Blocos
- Alocação: contígua, lista encadeada e indexada (ex. i-nodes)
- Monitoramento de blocos livres: mapa de bits e lista encadeada