

**Universidade Federal do Rio Grande do Norte  
Unidade Acadêmica Especializada em Ciências Agrárias  
Escola Agrícola de Jundiaí  
Curso de Análise e Desenvolvimento de Sistemas  
TAD0006 - Sistemas Operacionais - Turma 01**

# Sistemas de Arquivos

Antonino Feitosa  
antonino.feitosa@ufrn.br

Macaíba, junho de 2025



# Aula Passada

---

- Recursos
- Impasses
  - Condições
  - Modelagem
- Detecção e Recuperação de Impasses
- Evitando Impasses
- Prevenindo Impasses
- Livelock e Inanição

# Roteiro

---

- Arquivos
- Diretórios
- Esquemas de Sistemas de Arquivos
- Implementação de Arquivos

# Sistemas de Arquivos



# Sistemas de Arquivos

---

- O espaço de endereçamento pode não ser suficiente para armazenar todos os dados de um processo.
- Alguns dados precisam persistir, muitas vezes por dias ou meses.
  - Uma falha do sistema pode causar perdas dos dados.
- Pode ser necessário compartilhar informações entre diferentes processos.

# Sistemas de Arquivos

Requisitos para armazenamento de informações por um longo prazo:

1. Deve ser possível armazenar uma quantidade muito grande de informações.
2. As informações devem sobreviver ao término do processo que as está utilizando.
3. Múltiplos processos têm de ser capazes de acessá-las ao mesmo tempo.

# Sistemas de Arquivos

Principais questões ao tratar de informações:

1. Como você encontra informações?
2. Como impedir que um usuário leia os dados de outro?
3. Como saber quais espaços estão livres?

# Sistemas de Arquivos

---

- Abstrações para o armazenamento de informação a longo prazo: Arquivos.
- Principais abstrações do projeto de sistemas operacionais: processos, espaço de endereçamento e arquivos.



# Sistemas de Arquivos

---

- Arquivos são unidades lógicas de informação criadas por processos.
  - Uma espécie de espaço de endereçamento usados para modelar o armazenamento secundário (persistente).
    - HS, SSD, entre outros dispositivos.
  - Processos podem executar várias operações sobre arquivos.
    - Criação, remoção, leitura, gravação, renomeação, etc.
      - Informações armazenadas em arquivos devem ser persistentes.
        - São independentes da criação e término de um processo.

# Sistemas de Arquivos

---

- Arquivos são gerenciados pelo sistema operacional.
  - Como são estruturados, nomeados, acessados, usados, protegidos, implementados, etc.
- Sistema de Arquivos: parte do sistema operacional lidando com arquivos.
  - Para o usuário: quais operações podem ser efetuadas.
  - Projetista: estruturação dos arquivos, controle de acesso, etc.

# Arquivos



# Arquivos

---

- Arquivo: abstração que fornece uma maneira para armazenar informações sobre o dispositivo de armazenamento secundário e lê-las depois.
  - Detalhes de como o dispositivo funciona e qual dispositivo está sendo utilizado devem ser abstraídos para o usuário.

# Arquivos

---

- Nomeação dos arquivos: característica mais importante na abstração de arquivos.
  - Cada sistema adota as suas regras.
  - Em geral, cadeias de uma a oito letras.
  - Distinção entre letras maiúsculas e minúsculas.

# Arquivos

---

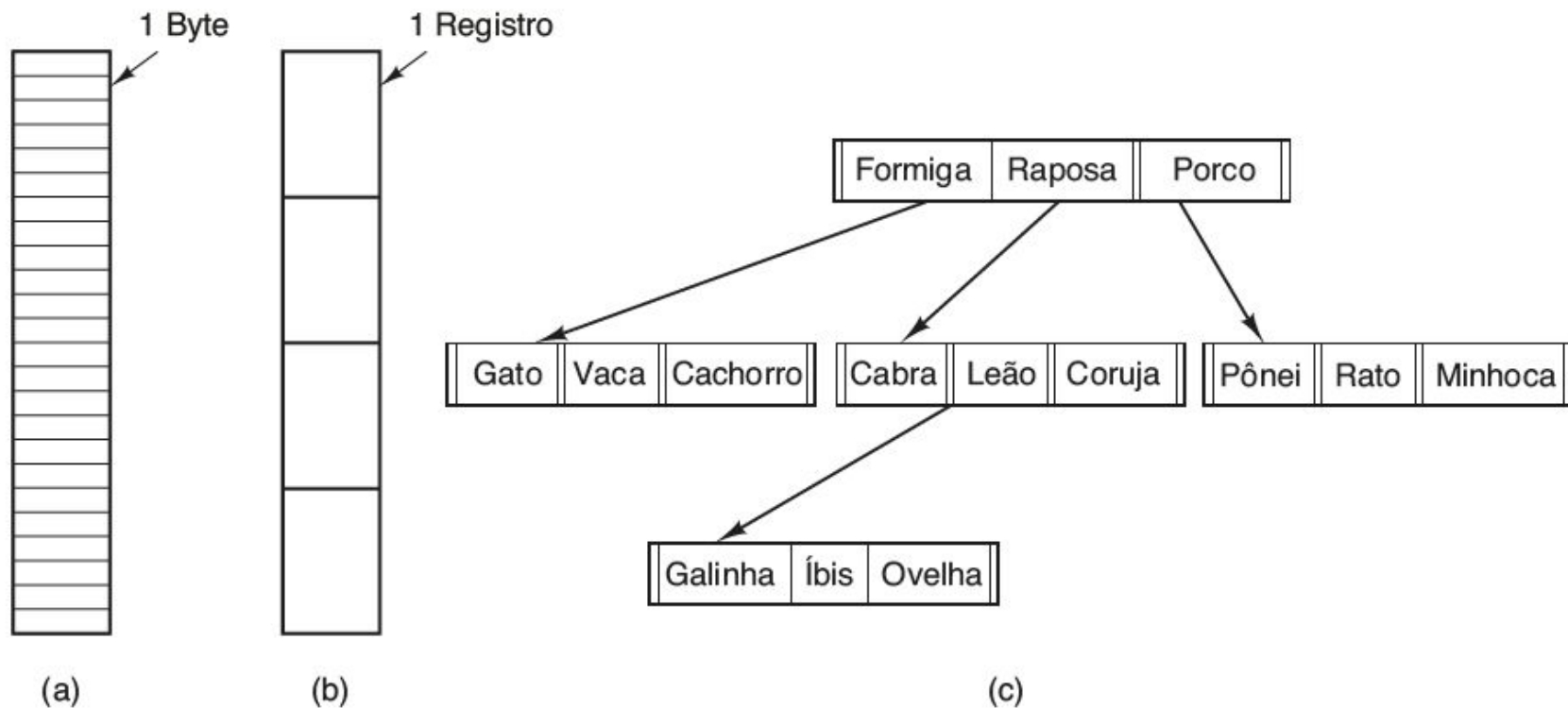
- Extensão de arquivos: separação dos nomes em duas partes.
  - Parte final após o ponto é chamada de extensão.
  - Apenas convenções que podem ser designados significado.
    - Podemos alterar a extensão do arquivo por meio de nomeação, no entanto, isso não altera a **estrutura** interna do arquivo.

# Estrutura de Arquivos

- Arquivos podem ser estruturados como:
  - Sequência desestruturada de bytes.
    - O sistema operacional não sabe ou não se importa sobre o que há no arquivo.
  - Uma sequência de registros de tamanho fixo, cada um com alguma estrutura interna.
  - Uma árvore de registros, não necessariamente todos do mesmo tamanho, cada um contendo um campo chave em uma posição fixa no registro.

# Estrutura de Arquivos

**FIGURA 4.2** Três tipos de arquivos. (a) Sequência de bytes. (b) Sequência de registros. (c) Árvore.





# Tipos de Arquivos

---

- **Arquivos regulares:** são aqueles que contêm informações do usuário.
- **Diretórios:** são arquivos do sistema para manter a estrutura do sistema de arquivos.
- **Arquivos especiais de caracteres:** são relacionados com entrada/saída e usados para modelar dispositivos de E/S seriais
- como terminais, impressoras e redes.
- **Arquivos especiais de blocos:** são usados para modelar discos.

# Arquivos Regulares

- Geralmente são arquivos ASCII ou binários.
- Todo sistema operacional deve reconhecer pelo menos um tipo de arquivo: o seu próprio arquivo executável.

# Arquivos Regulares

---

- Arquivos ASCII consistem de linhas de texto.
  - Término de linha pode variar entre sistemas.
    - Carriage return (CR): `\r`.
    - Line feed: (LF): `\n` (Linux e macOS).
    - Windows usa ambos.
  - Linhas podem ter tamanhos diferentes.
  - Podem ser exibidos e impressos como são e editados com qualquer editor de texto.

# Arquivos Regulares

- Geralmente são arquivos ASCII ou binários.
- Arquivos binários são sequências de bytes.
  - Possuem alguma estrutura interna conhecida pelos programas que os usam.
    - Devem satisfazer a estrutura estabelecida.
  - Listá-los em uma impressora ou editor de texto resulta em algo completamente incompreensível.

# Acesso aos Arquivos

- Acesso Sequencial: leitura em ordem.
  - Origem em dispositivos de fita magnética.
- Acesso Aleatório: arquivos ou registros que podem ser lidos em qualquer ordem.
  - Operação para determinar a posição a ser lida (seek).

# Metadados

---

- Fornecem informações adicionais sobre o arquivo que não fazem parte dos dados, com nome, data de modificação, acesso, etc.

**FIGURA 4.4** Alguns possíveis 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 de 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

# Metadados

---

Posição da chave	Posição da chave em cada registro
Tamanho da chave	Número de bytes na 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



# Operações Básicas

---

- Create: O arquivo é criado sem dados
- Delete: Quando o arquivo não é mais necessário, ele tem de ser removido para liberar espaço para o disco.
  - Há sempre uma chamada de sistema para essa finalidade.
- Open: Antes de usar um arquivo, um processo precisa abri-lo.
  - Recurso
- Close: Quando todos os acessos são concluídos, os atributos e endereços de disco não são mais necessários, então o arquivo deve ser fechado para liberar espaço da tabela interna.
  - Imposição um número máxím de arquivos abertos em processos.

# Operações Básicas

- Read: Dados são lidos do arquivo.
  - Os bytes vêm da posição atual.
  - Deve especificar a quantidade de dados necessária.
  - Deve fornecer um buffer para colocá-los.
- Write: Dados são escritos para o arquivo de novo, normalmente na posição atual.
  - Se a posição atual for o final do arquivo, seu tamanho aumentará.
  - Se estiver no meio do arquivo, os dados existentes serão sobrescritos e perdidos para sempre.

# Operações Básicas

- Append: Acrescenta dados somente para o final do arquivo.
- Seek: método para especificar de onde tirar os dados.
  - Para arquivos de acesso aleatório.
- Get attributes/Set attributes: altera os metadados.
- Rename: renomeia um arquivo.
  - Pode ser substituído por uma operação de cópia.

# Diretórios

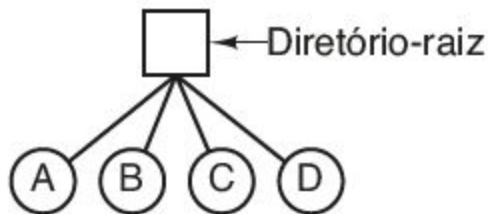
---

# Diretórios

- Arquivos utilizados para controlar outros arquivos.
- Sistemas de diretório em nível único: único diretório contendo todos os arquivos.
  - A forma mais simples de um sistema de diretório.
  - Como ele é o único, o nome não importa.
  - Adequado para aplicações dedicadas muito simples.

# Diretórios

**FIGURA 4.6** Um sistema de diretório em nível único contendo quatro arquivos.

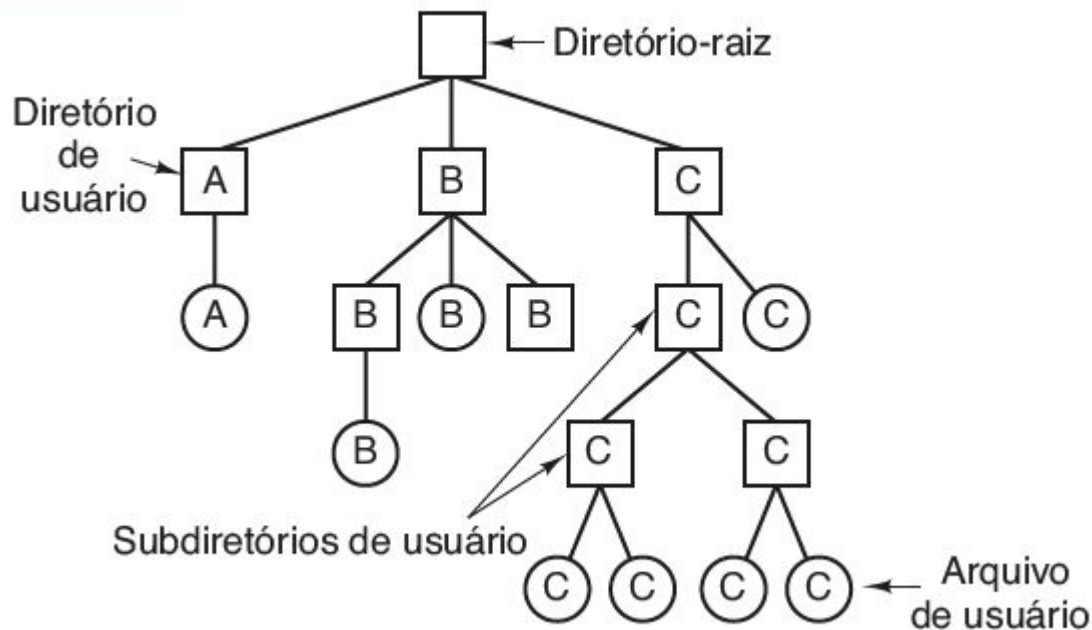


# Diretórios: Hierárquicos

- Sistemas de diretórios hierárquicos
- Permite agrupar os arquivos, facilitando a gerência e busca de arquivos em sistemas com centenas ou milhares de arquivos.
- Organização hierárquica: árvore de diretórios.

# Diretórios: Hierárquicos

**FIGURA 4.7** Um sistema hierárquico de diretórios.





# Diretórios: Hierárquicos

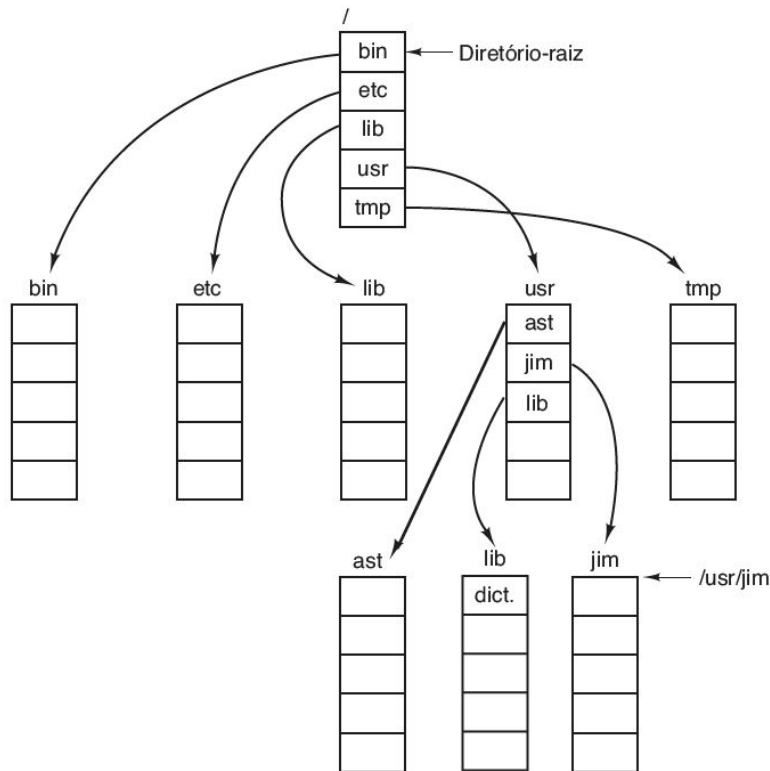
- Nomes de caminhos: determina como especificar os nomes dos arquivos considerando a estrutura hierárquica.
  - Caminhos absolutos: consiste no caminho do diretório-raiz para o arquivo.
    - Os componentes do caminho são separados por um separador.
    - Windows \usr\ast\caixapostal
    - UNIX /usr/ast/caixapostal
    - O primeiro caractere do nome do caminho é o separador.
  - Caminhos relativos: relativos ao diretório de trabalho (atual).

# Diretórios: Hierárquicos

- Entradas especiais.
  - ponto (.): refere-se ao diretório atual.
  - pontoponto (..): refere-se ao diretório pai.
    - Refere-se a si mesmo no diretório raiz.

# Diretórios: Hierárquicos

**FIGURA 4.8** Uma árvore de diretórios UNIX.



# Diretórios: Hierárquicos - Operações

---

- Create: Um diretório é criado.
- Delete: Um diretório é removido.
  - Deve estar vazio.
- Opendir: Diretórios podem ser lidos.
- Closedir: Quando um diretório tiver sido lido, ele será fechado para liberar espaço de tabela interno.
- Readdir: Essa chamada retorna a próxima entrada em um diretório aberto.
- Rename: Renomeia o diretório.
- Link: A ligação (linking) é uma técnica que permite que um arquivo apareça em mais de um diretório.
  - Ligação estrita (hard link): dois nomes apontando para a mesma estrutura de dados interna.
- Unlink: Uma entrada de diretório é removida.

# Diretórios: Hierárquicos

- Ligação Simbólica: nome pode ser criado que aponte para um arquivo minúsculo que contém o nome de outro arquivo.
  - Quando o primeiro é usado — aberto, por exemplo — o sistema de arquivos segue o caminho e encontra o nome no fim.
  - Conseguem nomear arquivos em computadores remotos.
  - Implementação menos eficiente.

# Implementação



# Esquema do Sistema de Arquivos: MBR

- Sistemas de arquivos são armazenados em discos (dispositivos de armazenamento secundário).
  - A maioria dos discos pode ser dividida em uma ou mais partições.
  - Cada partição pode ter um sistema de arquivos independente.
  - O Setor 0 do disco é chamado de MBR (Master Boot Record – registro mestre de inicialização)
    - É usado para inicializar o computador.
    - O fim do MBR contém a tabela de partição.
      - Fornece os endereços de início e fim de cada partição.

# Esquema do Sistema de Arquivos: MBR

- Uma das partições da tabela é marcada como ativa.
  - Bloco de inicialização.
  - Deve conter um sistema operacional.
- Quando o computador é inicializado, a BIOS lê e executa o MBR.
  - Lê e executa a partição ativa.
- Toda partição possui um bloco de inicialização, mesmo que não tenha um SO.
  - Por questões de uniformidade.



# Esquema do Sistema de Arquivos: MBR

- Principal esquema para sistemas legados baseados em BIOS.
- Suporta dispositivos de até 2TB.
- Suporta apenas 4 partições.
- Não possui backup ou sistema de recuperação.
  - Caso seja corrompido, não é possível iniciar o SO.

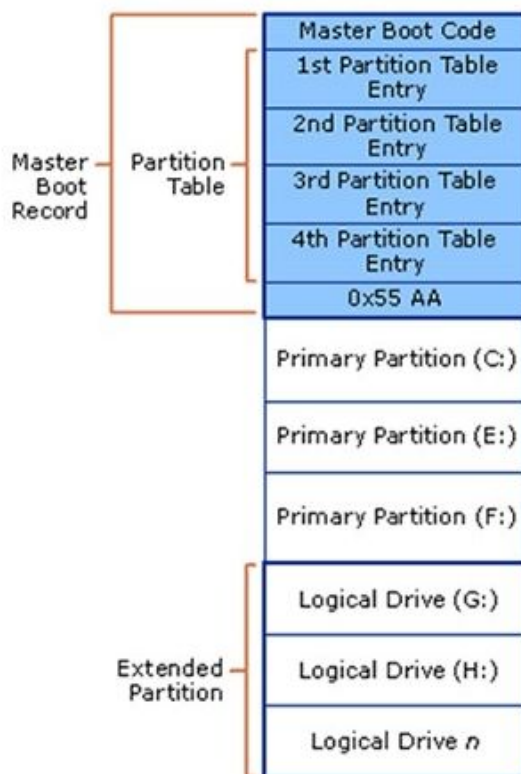
# Esquema do Sistema de Arquivos: GPT

- GUID Partition Table (GPT).
  - Cada partição possui um GUID (identificador globalmente exclusivo).
- Alternativa moderna ao MBR.
- Desenvolvido como parte do Unified Extensible Firmware Interface (UEFI).
- Mantém um MBR para compatibilidade com sistemas legados.

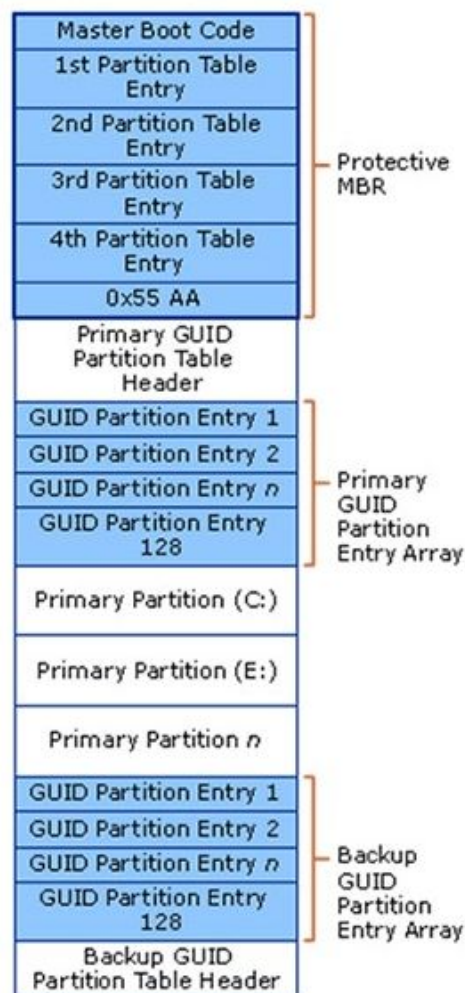
# Esquema do Sistema de Arquivos: GPT

- MBR: o particionamento e a inicialização os dados são armazenados em um só lugar.
- GPT: armazena várias cópias dos dados de particionamento e a inicialização.
  - Recuperação caso os dados sejam corrompidos.

## MBR Partition Table Scheme



## GPT Partition Table Scheme



# Esquema do Sistema de Arquivos

---

Feature	MBR (Master Boot Record)	GPT (GUID Partition Table)
Drive Size Support	Up to 2TB	More than 2TB (exceeds 18TB)
Partition Limit	4 primary partitions	Virtually unlimited (128 partitions in practice)
Compatibility	Works with older BIOS systems	Requires UEFI (but compatible with most modern systems)
Partition Data Backup	No automatic backups	Stores multiple copies of partition data
Corruption Recovery	Difficult to recover	Built-in redundancy, easier to recover
Boot Mode Support	BIOS only	UEFI (but supports booting in legacy mode)
Usage	Older systems, smaller drives	Modern systems, larger drives

# Implementação de Arquivos

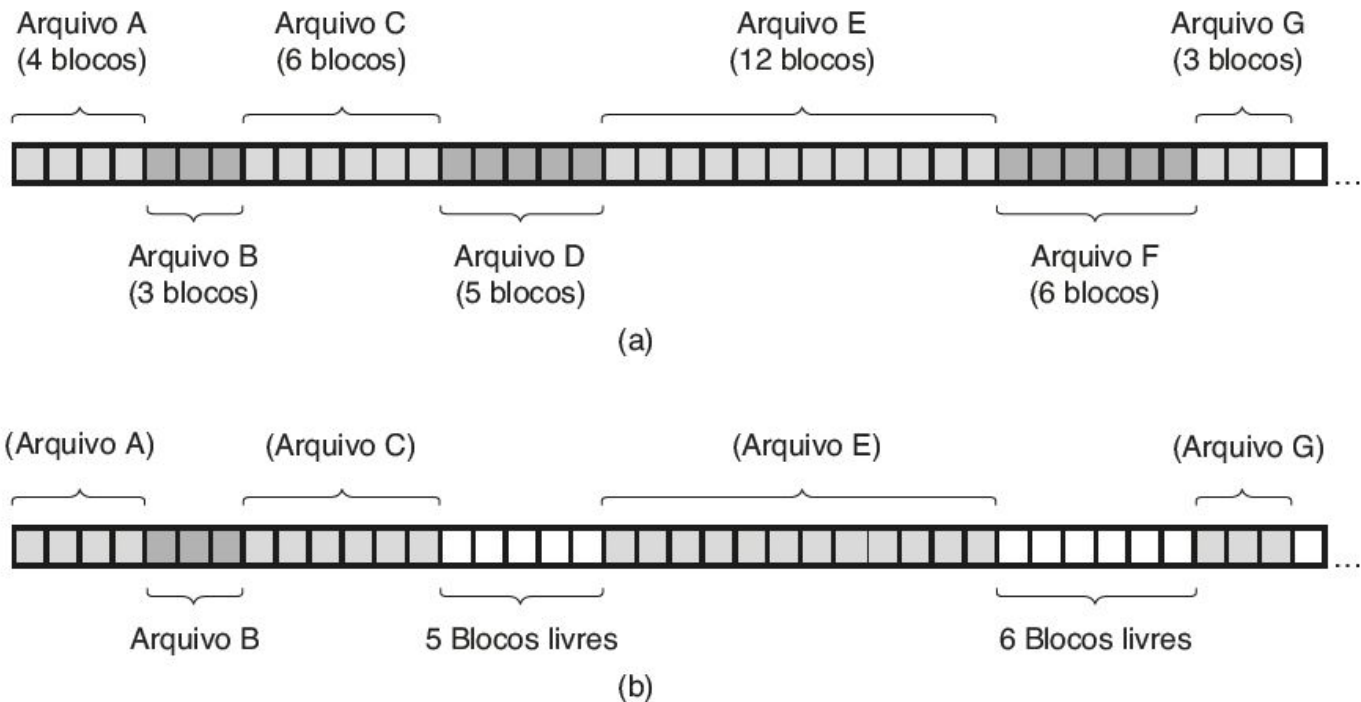
---

# Implementação de Arquivos

- Alocação Contígua: cada arquivo é uma sequência contínua de blocos de armazenamento.
  - Simples de implementar: bloco inicial e quantidade de blocos.
  - Desempenho de leitura: apenas uma busca para leitura sequencial.
  - Gera fragmentação do disco.

# Implementação de Arquivos

**FIGURA 4.10** (a) Alocação contígua de espaço de disco para sete arquivos. (b) O estado do disco após os arquivos D e F terem sido removidos.





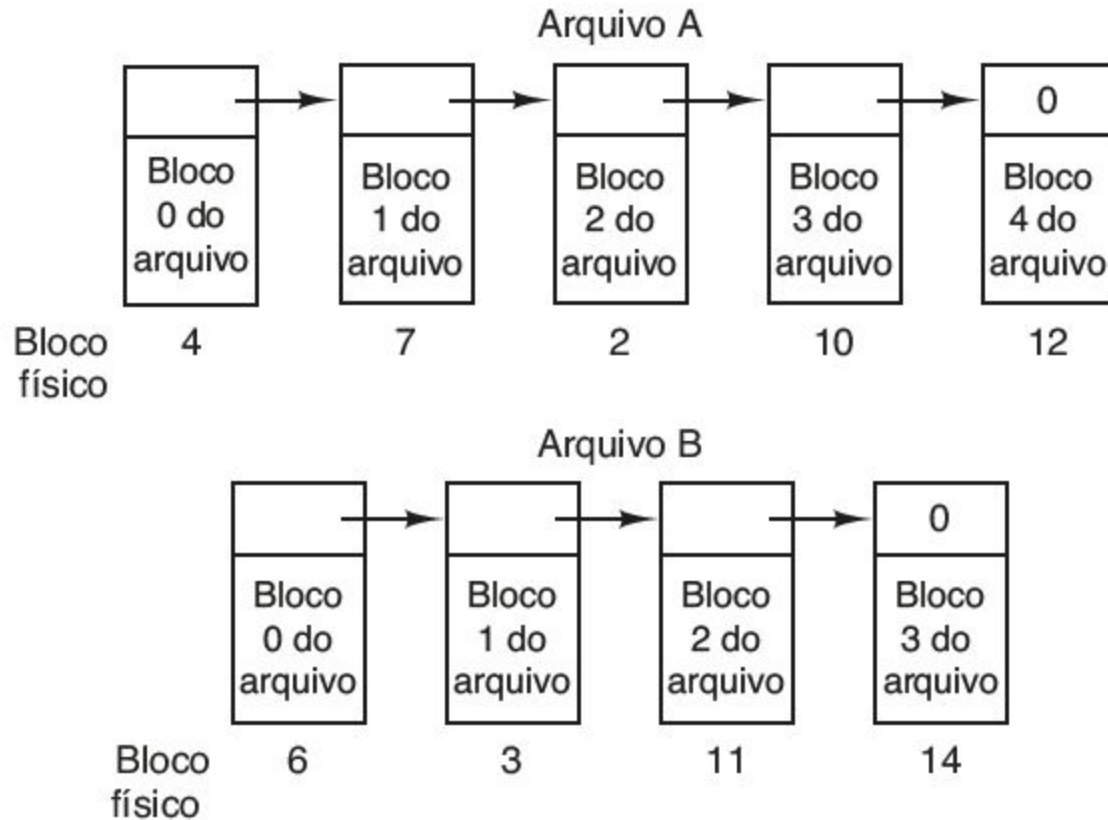
# Alocação por Lista Encadeada

- O arquivos é armazenado em blocos de modo que cada bloco possui uma referência para o próximo.
- Não gera fragmentação.
- Basta armazenar o endereço do primeiro bloco.
  - Útil para acesso sequencial.
  - Extremamente lento para acesso aleatório.
    - É necessário efetuar a leitura dos  $n-1$  blocos anteriores antes de acessar o bloco  $n$ .

# Alocaç

**FIGURA 4.11**

Armazenando um arquivo como uma lista encadeada de blocos de disco.



# Alocação por Lista Encadeada com Tabela

- Alocação por lista encadeada usando uma tabela na memória principal: consiste em manter uma tabela com os endereços de cada bloco, permitindo o acesso aleatório.
  - A tabela é para o sistema, independente dos arquivos.
- Essa tabela na memória principal é chamada de FAT (File Allocation Table — tabela de alocação de arquivos).
  - O acesso ainda é sequencial, no entanto, a leitura não é necessária.
- A desvantagem é que a tabela precisa estar na memória.

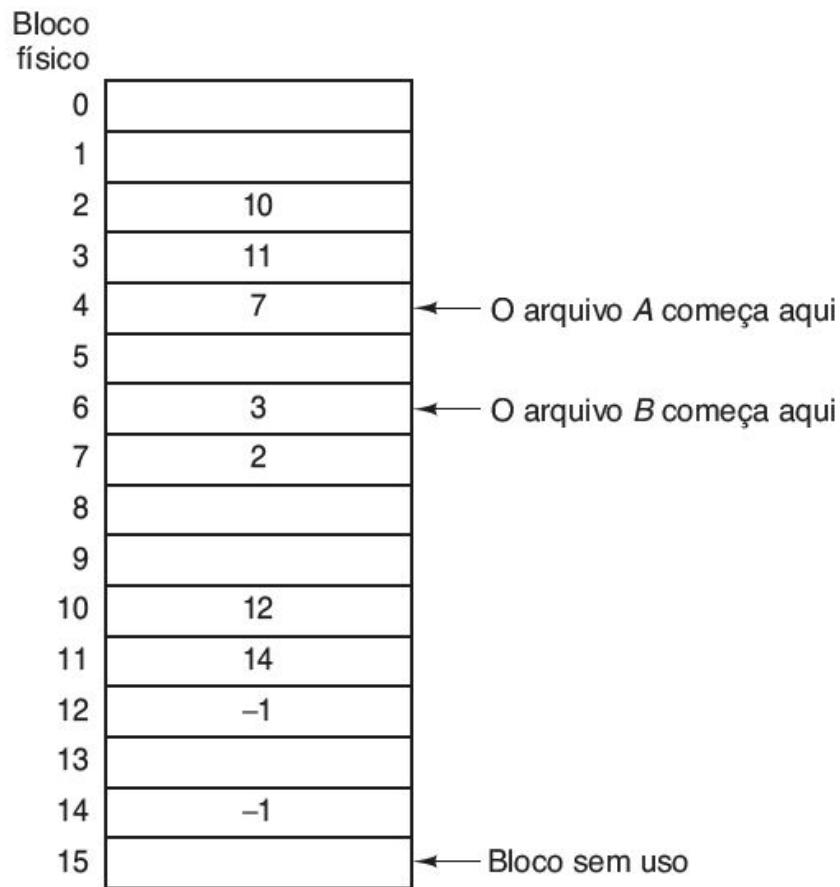
# Alocação por Lista Encadeada com Tabela

- Com um disco de 1 TB e um tamanho de bloco de 1 KB, a tabela precisa de 1 bilhão de entradas, uma para cada um dos 1 bilhão de blocos de disco.
- Desse modo, a tabela ocupará 3 GB da memória principal o tempo inteiro.

# Alocação

**FIGURA 4.12**

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



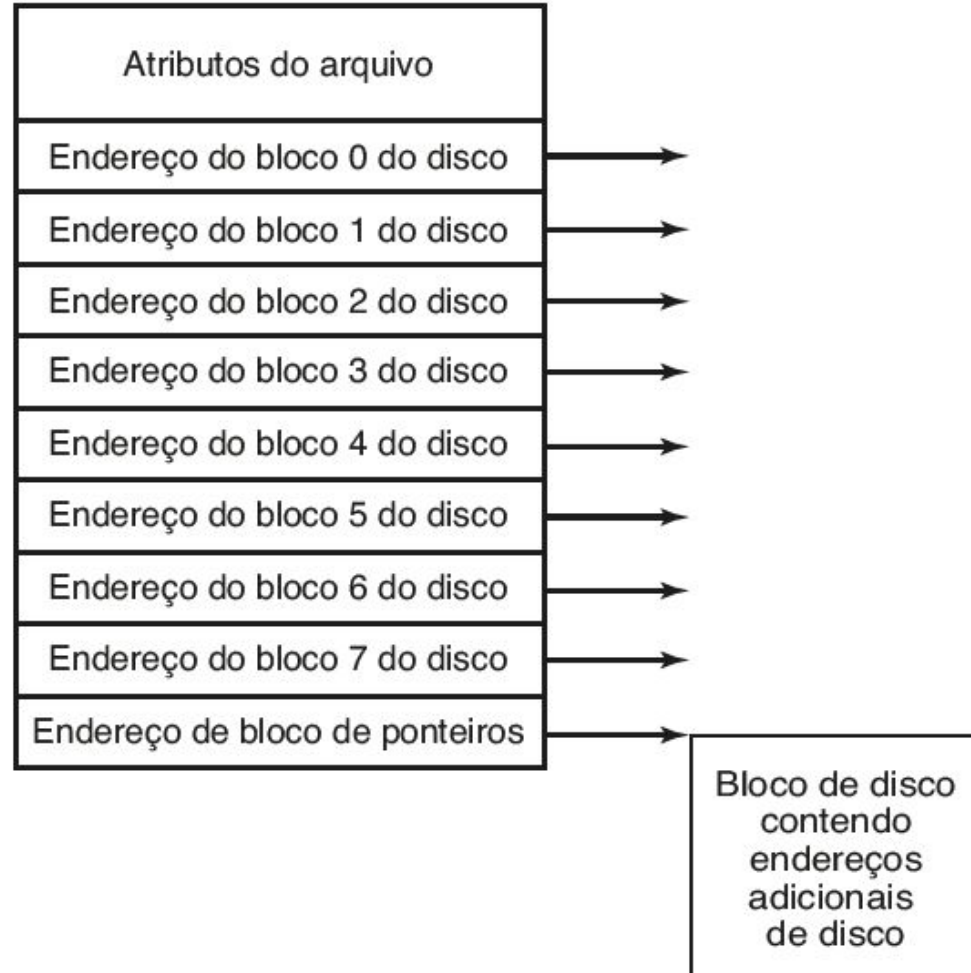
# Estrutura i-node

- Estrutura de dados chamada de i-node: lista os atributos e os endereços de disco dos blocos do disco.
- O i-node precisa estar na memória apenas quando o arquivo correspondente estiver aberto.

**FIGURA 4.13** Um exemplo de i-node.

# Estrutura

---



# Resumo

---



# Resumo

---

- Arquivos
- Diretórios
- Esquemas de Sistemas de Arquivos
- Implementação de Arquivos

# Leitura Complementar

- Entendendo os sistemas de arquivos

<https://www.kingston.com/br/blog/personal-storage/understanding-file-systems>

# Dúvidas?

