

Nota: Este material complementar, disponível em <https://prettore.github.io/lectures.html> representa uma cópia resumida de conteúdos bibliográficos disponíveis gratuitamente na Internet.

## Sistemas de Arquivos

<b>Introdução</b>	<b>1</b>
Arquivos	2
Atributos de Arquivo	2
Operações sobre Arquivos	2
Tipos de Arquivo	3
Diretórios	3
Estruturas de Diretório	3
Operações sobre Diretórios	4
<b>Implementação de Sistemas de Arquivos</b>	<b>4</b>
Estrutura do Sistema de Arquivos	4
Métodos de Alocação de Espaço em Disco	4
Gerenciamento de Espaço Livre	5
<b>Otimização e Gerenciamento de Sistemas de Arquivos</b>	<b>5</b>
Cache de Disco (Buffer Cache)	5
Recuperação	5
Desempenho	6
<b>Conclusão</b>	<b>6</b>
<b>Referências</b>	<b>6</b>

## Introdução

Os sistemas de arquivos são um componente fundamental de qualquer sistema operacional, fornecendo o mecanismo pelo qual os dados são armazenados, organizados e acessados em dispositivos de armazenamento secundário, como discos rígidos, SSDs e pen drives. Do ponto de vista do usuário, um sistema de arquivos oferece uma visão lógica e hierárquica dos dados, geralmente na forma de arquivos e diretórios, abstraindo a complexidade física do hardware de armazenamento. Para o sistema operacional, o sistema de arquivos é responsável por gerenciar o espaço em disco, garantir a integridade dos dados, controlar o acesso e fornecer uma interface eficiente para que os programas possam ler e escrever informações. Este capítulo explora os conceitos essenciais dos sistemas de arquivos, incluindo a natureza dos arquivos e diretórios, os métodos de implementação de sistemas de arquivos, e as técnicas de otimização e gerenciamento empregadas para garantir desempenho e confiabilidade.

## Arquivos

Um **arquivo** é uma coleção nomeada de informações relacionadas, registrada em armazenamento secundário. Do ponto de vista do usuário, um arquivo é a menor unidade lógica de armazenamento; ou seja, os dados não podem ser escritos em armazenamento secundário a menos que estejam dentro de um arquivo. Os arquivos podem representar programas (código fonte e executável), dados (numéricos, alfabéticos, alfanuméricos, binários) ou documentos (texto, imagens, áudio, vídeo).

### Atributos de Arquivo

Cada arquivo possui um conjunto de atributos que descrevem suas propriedades. Os atributos comuns incluem:

- **Nome:** O nome simbólico do arquivo, a única informação mantida em formato legível por humanos.
- **Identificador:** Um número único (geralmente não legível por humanos) que identifica o arquivo dentro do sistema de arquivos.
- **Tipo:** Necessário para sistemas que suportam diferentes tipos de arquivos (e.g., executável, texto, imagem).
- **Localização:** Um ponteiro para a localização do arquivo no dispositivo de armazenamento.
- **Tamanho:** O tamanho atual do arquivo (em bytes, palavras ou blocos) e, possivelmente, o tamanho máximo permitido.
- **Proteção:** Controles de acesso que especificam quem pode ler, escrever, executar, etc., o arquivo.
- **Datas, Horários e Identificação do Usuário:** Informações para criação, última modificação e último acesso, além do proprietário e grupo do arquivo.

Esses atributos são armazenados na estrutura de diretório, que também reside no armazenamento secundário.

### Operações sobre Arquivos

Os sistemas operacionais fornecem chamadas de sistema para realizar operações básicas em arquivos:

- **Criar (Create):** Encontrar espaço no sistema de arquivos e fazer uma entrada para o novo arquivo no diretório.
- **Escrever (Write):** Localizar o arquivo e escrever dados nele, geralmente mantendo um ponteiro de escrita para a próxima posição.
- **Ler (Read):** Localizar o arquivo e ler dados dele, geralmente mantendo um ponteiro de leitura para a próxima posição.
- **Reposicionar (Reposition / Seek):** Mover o ponteiro de posição atual do arquivo para um local específico dentro do arquivo.
- **Excluir (Delete):** Remover o arquivo do diretório e liberar o espaço em disco alocado a ele.

- **Truncar (Truncate):** Manter os atributos do arquivo, mas zerar seu conteúdo, liberando o espaço em disco.
- **Abrir (Open):** Antes de usar um arquivo, um processo deve abri-lo. O SO busca as entradas do diretório, move o conteúdo para a memória e retorna um pequeno inteiro chamado descritor de arquivo (ou handle) para o processo. Operações subsequentes usam esse descritor.
- **Fechar (Close):** Quando o acesso ao arquivo não é mais necessário, o processo o fecha. Isso libera as estruturas de dados internas do SO associadas ao arquivo e garante que quaisquer dados em buffer sejam escritos no disco.

## Tipos de Arquivo

Muitos sistemas operacionais reconhecem e suportam diferentes tipos de arquivo, frequentemente indicados pela extensão do nome do arquivo (e.g., .exe, .txt, .jpg). O tipo pode indicar a estrutura interna do arquivo ou o programa que pode operá-lo.

## Diretórios

Os dispositivos de armazenamento podem conter um grande número de arquivos. Para organizar esses arquivos, os sistemas de arquivos usam **diretórios** (ou pastas). Um diretório é, em si, um arquivo que contém informações sobre outros arquivos e diretórios.

### Estruturas de Diretório

- **Diretório de Nível Único (Single-Level Directory):** Todos os arquivos estão no mesmo diretório. Simples, mas inadequado para muitos arquivos ou múltiplos usuários devido a conflitos de nomes e dificuldades de agrupamento.
- **Diretório de Dois Níveis (Two-Level Directory):** Cada usuário tem seu próprio diretório de arquivos (User File Directory - UFD). O sistema mantém um diretório mestre (Master File Directory - MFD) que aponta para os UFDs. Resolve conflitos de nome entre usuários, mas não permite que um usuário organize seus próprios arquivos em subgrupos.
- **Diretório em Árvore (Tree-Structured Directory):** A estrutura mais comum. Permite que os usuários criem seus próprios subdiretórios e organizem seus arquivos hierarquicamente. Cada diretório pode conter arquivos e outros diretórios. O conceito de caminho (path name) é usado para especificar a localização de um arquivo (e.g., /usr/local/bin/programa).
- **Diretório Acíclico-Grafo (Acyclic-Graph Directory):** Permite que diretórios e arquivos sejam compartilhados, tendo múltiplas entradas de diretório apontando para eles. Isso é alcançado através de links ou atalhos. A estrutura deve ser acíclica para evitar ciclos na busca e problemas com travessias.
- **Diretório de Grafo Geral (General Graph Directory):** Permite ciclos. Requer algoritmos mais complexos para evitar problemas, como a coleta de lixo para recuperar espaço quando um arquivo não é mais referenciado, ou a limitação do número de referências.

## Operações sobre Diretórios

- **Criar (Create):** Criar um novo diretório.
- **Excluir (Delete):** Excluir um diretório (geralmente deve estar vazio).
- **Listar (List):** Listar os arquivos e subdiretórios contidos em um diretório.
- **Renomear (Rename):** Alterar o nome de um diretório.
- **Atravessar (Traverse):** Navegar pela estrutura de diretórios para localizar um arquivo ou diretório específico.

## Implementação de Sistemas de Arquivos

A implementação de um sistema de arquivos envolve várias camadas e estruturas de dados, tanto na memória principal quanto no disco.

### Estrutura do Sistema de Arquivos

Os sistemas de arquivos são tipicamente armazenados em discos. Um disco é dividido em uma ou mais partições, e cada partição pode conter um sistema de arquivos. A estrutura de um sistema de arquivos em disco geralmente inclui:

- **Bloco de Boot (Boot Control Block):** Contém informações necessárias para inicializar o sistema operacional a partir daquela partição (se for inicializável).
- **Bloco de Controle de Volume/Partição (Volume Control Block / Superblock):** Contém detalhes sobre a partição, como o número de blocos, tamanho dos blocos, contagem de blocos livres, ponteiros para blocos livres, etc.
- **Estrutura de Diretório:** Usada para organizar os arquivos. Contém informações sobre os arquivos, como nome, atributos e localização dos blocos de dados.
- **Bloco de Controle de Arquivo (File Control Block - FCB / inode no Unix):** Contém muitos dos atributos do arquivo, incluindo permissões, datas, proprietário, tamanho e ponteiros para os blocos de dados do arquivo.

### Métodos de Alocação de Espaço em Disco

Descrevem como os blocos de disco são alocados aos arquivos:

1. **Alocação Contígua:** Cada arquivo ocupa um conjunto contíguo de blocos no disco. Simples e rápido para acesso sequencial e direto. Sofre de fragmentação externa e dificuldade em encontrar espaço para arquivos que crescem.
2. **Alocação Encadeada (Linked Allocation):** Cada arquivo é uma lista encadeada de blocos de disco; os blocos podem estar dispersos em qualquer lugar do disco. Cada bloco contém um ponteiro para o próximo bloco. Resolve a fragmentação externa. Ineficiente para acesso direto (requer travessia da lista). Se um ponteiro for perdido ou corrompido, os dados subsequentes podem ser perdidos. Uma variação é a Tabela de Alocação de Arquivos (File Allocation Table - FAT), usada pelo MS-DOS e versões antigas do Windows, onde os ponteiros são armazenados em uma tabela separada na memória.

3. **Alocação Indexada (Indexed Allocation):** Traz todos os ponteiros para os blocos de dados de um arquivo juntos em um bloco especial chamado bloco de índice (index block). Cada arquivo tem seu próprio bloco de índice. Suporta acesso direto eficientemente e não sofre de fragmentação externa. O problema é o tamanho do bloco de índice; para arquivos grandes, um único bloco de índice pode não ser suficiente. Soluções incluem encadear blocos de índice, usar um esquema de índice multinível ou uma combinação.

## Gerenciamento de Espaço Livre

O sistema de arquivos deve rastrear os blocos de disco livres para poder alocá-los quando novos arquivos são criados ou arquivos existentes crescem.

- **Vetor de Bits (Bit Vector / Bitmap):** Uma coleção de bits, um para cada bloco do disco. Se o  $\text{bit}[i] = 0$ , o bloco  $i$  está livre; se  $\text{bit}[i] = 1$ , o bloco  $i$  está ocupado. Simples e eficiente para encontrar blocos livres contíguos.
- **Lista Encadeada:** Os blocos livres são encadeados. O cabeçalho da lista aponta para o primeiro bloco livre, que contém um ponteiro para o próximo, e assim por diante. Ineficiente para encontrar múltiplos blocos contíguos.
- **Agrupamento (Grouping):** Armazena os endereços de  $N$  blocos livres no primeiro bloco livre. Os  $N-1$  primeiros são realmente livres. O último bloco contém os endereços de outros  $N$  blocos livres.
- **Contagem (Counting):** Mantém o endereço do primeiro bloco livre e o número ( $N$ ) de blocos livres contíguos que se seguem a ele.

## Otimização e Gerenciamento de Sistemas de Arquivos

### Cache de Disco (Buffer Cache)

Para melhorar o desempenho, os sistemas operacionais mantêm um cache na memória principal (RAM) contendo cópias de blocos de disco recentemente acessados. Quando um processo solicita um bloco, o SO primeiro verifica o cache. Se o bloco estiver lá (cache hit), ele é usado diretamente da memória. Se não (cache miss), o bloco é lido do disco para o cache e depois para o processo. Técnicas de escrita como write-through (escrita síncrona) ou write-back (escrita atrasada) são usadas.

### Recuperação

Falhas no sistema (e.g., queda de energia) podem deixar o sistema de arquivos em um estado inconsistente. Mecanismos de recuperação são necessários:

- **Backup e Restauração:** Copiar dados do sistema de arquivos para um dispositivo de armazenamento offline e restaurá-los em caso de perda.
- **Verificadores de Consistência (e.g., fsck no Unix, chkdsk no Windows):** Programas que varrem as estruturas do sistema de arquivos (como tabelas de diretórios e FCBs/inodes) para verificar inconsistências e corrigi-las, se possível.

- **Sistemas de Arquivos com Journaling (Journaling File Systems):** Antes de realizar qualquer modificação nas estruturas principais do sistema de arquivos, uma descrição da operação (transação) é escrita em um log sequencial chamado journal. Se o sistema falhar, o journal pode ser relido para trazer o sistema de arquivos de volta a um estado consistente, refazendo ou desfazendo operações incompletas. Exemplos: ext3, ext4, NTFS, XFS.
- **Sistemas de Arquivos Log-Structured (LFS):** Todas as modificações são escritas sequencialmente em um log. Isso pode melhorar o desempenho de escrita, mas a leitura pode ser mais complexa.

## Desempenho

O desempenho do sistema de arquivos é crucial. Fatores que afetam o desempenho incluem:

- **Organização dos blocos no disco:** Colocar blocos de um mesmo arquivo próximos uns dos outros (localidade espacial) pode reduzir o tempo de busca e latência rotacional.
- **Tamanho do bloco:** Um tamanho de bloco maior pode melhorar a taxa de transferência para arquivos grandes, mas aumentar a fragmentação interna para arquivos pequenos.
- **Eficiência do cache de disco.**
- **Algoritmos de alocação e gerenciamento de espaço livre.**

## Conclusão

Os sistemas de arquivos são uma abstração essencial que permite aos usuários e aplicações interagir com dispositivos de armazenamento de forma organizada e eficiente. Eles fornecem a noção de arquivos e diretórios, gerenciando o armazenamento físico subjacente através de métodos de alocação de espaço, estruturas de diretório e mecanismos de controle de acesso. A implementação de um sistema de arquivos envolve um complexo conjunto de estruturas de dados e algoritmos para garantir a integridade, o desempenho e a confiabilidade. Técnicas como caching, journaling e verificadores de consistência são cruciais para otimizar o acesso e proteger contra perdas de dados. A compreensão dos princípios dos sistemas de arquivos é vital para qualquer pessoa que trabalhe com sistemas computacionais, desde o usuário final até o desenvolvedor de sistemas operacionais.

## Referências

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson Education.

- Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.
- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books. (Disponível online)

#### Isenção de Responsabilidade:

Os autores deste documento não reivindicam a autoria do conteúdo original compilado das fontes mencionadas. Este documento foi elaborado para fins educativos e de referência, e todos os créditos foram devidamente atribuídos aos respectivos autores e fontes originais.

Qualquer utilização comercial ou distribuição do conteúdo aqui compilado deve ser feita com a devida autorização dos detentores dos direitos autorais originais. Os compiladores deste documento não assumem qualquer responsabilidade por eventuais violações de direitos autorais ou por quaisquer danos decorrentes do uso indevido das informações contidas neste documento.

Ao utilizar este documento, o usuário concorda em respeitar os direitos autorais dos autores originais e isenta os compiladores de qualquer responsabilidade relacionada ao conteúdo aqui apresentado.