

## Sistemas de Arquivos

Em computação, sistema de ficheiros ou sistema de arquivos (também conhecido por sistema de gestão de ficheiros) é a forma de organização de dados em algum meio de armazenamento de dados em massa, frequentemente feito em discos magnéticos.

Ele controla como os dados são armazenados e recuperados, possibilitando ao sistema operacional decodificar os dados armazenados e lê-los ou gravá-los. Sem um sistema de arquivos, as informações colocadas em um meio de armazenamento seriam um grande corpo de dados, sem nenhuma maneira de dizer onde uma parte da informação termina e a próxima começa. Ao separar os dados em pedaços e dar um nome a cada peça, a informação é facilmente isolada e identificada. Tomando seu nome a partir do modo como os sistemas de informação baseados em papel são nomeados, cada grupo de dados é chamado de "arquivo".

A estrutura e as regras lógicas usadas para gerenciar os grupos de informações e seus nomes são chamadas de "sistema de arquivos".

Existem muitos tipos diferentes de sistemas de arquivos. Cada um tem estrutura e lógica diferentes, propriedades de velocidade, flexibilidade, segurança, tamanho e muito mais. Alguns sistemas de arquivos foram projetados para serem usados em aplicativos específicos. Por exemplo, o sistema de arquivos ISO 9660 foi projetado especificamente para discos ópticos.

Os sistemas de arquivos podem ser usados em vários tipos diferentes de dispositivos de armazenamento que usam diferentes tipos de mídia. O dispositivo de armazenamento mais comum em uso hoje é uma unidade de disco rígido. Outros tipos de mídia usados incluem memória flash, fitas magnéticas e discos ópticos. Em alguns casos, como no tmpfs, a memória principal do computador (memória de acesso aleatório, RAM) é usada para criar um sistema de arquivos temporário para uso a curto prazo.

Alguns sistemas de arquivos são usados em dispositivos locais de armazenamento de dados,<sup>[2]</sup> outros fornecem acesso a arquivos por meio de um protocolo de rede (por exemplo, clientes NFS,<sup>[3]</sup> SMB ou 9P). Alguns sistemas de arquivos são "virtuais", o que significa que os "arquivos" fornecidos (chamados de arquivos virtuais) são computados mediante solicitação (como procfs e sysfs) ou são apenas um mapeamento em um sistema de arquivos diferente usado como armazenamento de backup. O sistema de arquivos gerencia o acesso ao conteúdo dos arquivos e aos metadados sobre esses arquivos. É responsável por organizar o espaço de armazenamento. A confiabilidade, eficiência e ajuste com relação ao meio físico de armazenamento são importantes considerações de projeto.

## Origem Do Termo

Antes do advento dos computadores, o termo *sistema de arquivos* era usado para descrever um método de armazenamento e recuperação de documentos em papel.<sup>[4]</sup> Em 1961, o termo estava sendo aplicado ao arquivamento computadorizado ao lado do significado original.<sup>[5]</sup> Em 1964, já era de uso geral.

Fazendo analogias, tal organização assemelha-se a uma biblioteca escolar. O bibliotecário organiza os livros conforme um padrão, cuja busca, convenientemente, procura deixar mais fácil, sem ocupar muitas prateleiras e assegurando a integridade deste. Ainda, certamente, organiza os livros segundo suas características (assunto, censura, etc.). Depois de organizados, ou durante a organização, o bibliotecário cria uma lista com todos os livros da biblioteca, com seus assuntos, localizações e códigos respectivos.

Aplicando a analogia à informática, o sistema operacional seria o bibliotecário da "biblioteca de dados" do computador, o disco de armazenamento. Exatamente igual à organização de uma biblioteca, o sistema operacional guarda os dados nos espaços vazios do disco, rotulando-os com um FCB (*File Control Block*, Bloco de Controle de Arquivo) e ainda criando uma lista com a posição deste dado, chamada de MFT (*Master File Table*, Tabela de Arquivos Mestre). Sabendo a posição do arquivo a ser aberto/gravado, o sistema operacional solicita a leitura desta, decodifica/codifica e realiza a abertura/gravação do dado.

Um sistema de ficheiro é assim: uma forma de criar uma estrutura lógica de acesso a dados numa partição. Sendo assim, também é importante referir que nunca poderá ter dois ou mais tipos de sistemas de ficheiros (formatos) numa mesma partição.

O MBR (*Master Boot Record*) é um arquivo de dados interligado com a BIOS cuja importância é o reconhecimento do sistema de arquivos, como também na inicialização do sistema operacional.

Para a maioria dos usuários, o sistema de arquivos é o aspecto mais visível de um sistema operacional. Ele fornece o mecanismo para o armazenamento online e o acesso relacionado tanto aos dados como aos programas do sistema operacional e de todos os usuários do sistema de computação. O sistema de arquivos consiste em duas partes distintas: uma coleção de arquivos, cada um deles armazenando dados relacionados, e uma estrutura de diretórios, que organiza e fornece informação sobre todos os arquivos do sistema. Alguns sistemas de arquivos têm uma terceira parte, as partições, utilizadas para separar física ou logicamente grandes coleções de diretórios.

### Arquitetura

Um sistema de arquivos consiste em duas ou três camadas. Às vezes, as camadas são explicitamente separadas e, às vezes, as funções são combinadas.

O *sistema de arquivos lógico* é responsável pela interação com o aplicativo do usuário. Ele fornece a interface de programação de aplicações (API) para operações de arquivo - OPEN, CLOSE, READ, etc. e passa a operação solicitada para a camada abaixo para processamento. O sistema de arquivos lógico "gerencia as entradas da tabela de arquivos abertos e os descritores de arquivos por processo." Esta camada fornece "acesso a arquivos, operações de diretório, segurança e proteção".

A segunda camada opcional é o *sistema de arquivos virtual*. "Esta interface permite o suporte para múltiplas instâncias simultâneas de sistemas de arquivos físicos, cada um dos quais é chamado de implementação de sistema de arquivos."

A terceira camada é o *sistema de arquivos físico*. Esta camada está relacionada com o funcionamento físico do dispositivo de armazenamento (por exemplo, disco). Ele processa blocos físicos que estão sendo lidos ou gravados. Ele lida com o gerenciamento de buffer e de memória e é responsável pelo posicionamento físico de blocos em locais específicos no meio de armazenamento. O sistema de arquivos físico interage com os drivers de dispositivo ou com o canal para controlar o dispositivo de armazenamento.

### Gerenciamento de espaço

*Observação: isto aplica-se somente a sistemas de arquivos usados em dispositivos de armazenamento.*

Os sistemas de arquivos alocam o espaço de maneira granular, geralmente várias unidades físicas no dispositivo. O sistema de arquivos é responsável por organizar os arquivos e diretórios e manter o controle de quais áreas da mídia pertencem a qual arquivo e quais não estão sendo usadas. Por exemplo, no Apple DOS do início dos anos 80, os setores de 256 bytes em disquetes de 140 kilobytes usavam um *mapa de trilha/setor*.

Isso resulta em espaço não utilizado quando um arquivo não é um múltiplo exato da unidade de alocação, às vezes referido como espaço de folga. Para uma alocação de 512 bytes, o espaço médio não utilizado é de 256 bytes. Para clusters de 64 KB, o espaço médio não utilizado é de 32 KB. O tamanho da unidade de alocação é escolhido quando o sistema de arquivos é criado. Escolher o tamanho da alocação com base no tamanho médio dos arquivos que se espera estar no sistema de arquivos pode minimizar a quantidade de espaço inutilizável. Frequentemente, a alocação padrão pode fornecer uso razoável. Escolher um tamanho de alocação que seja muito pequeno resulta em sobrecarga excessiva se o sistema de arquivos contiver arquivos na maioria muito grandes.

A fragmentação de sistema de arquivos ocorre quando o espaço não utilizado ou os arquivos individuais não são contíguos. Como um sistema de arquivos é usado, os arquivos são criados, modificados e excluídos. Quando um arquivo é criado, o sistema de arquivos aloca espaço para os dados. Alguns sistemas de arquivos permitem ou exigem a especificação de uma alocação de espaço inicial e alocações incrementais subsequentes à medida que o arquivo cresce. À medida que os arquivos são excluídos, o espaço que eles foram alocados eventualmente é considerado disponível para uso por outros arquivos. Isso cria áreas utilizadas e não utilizadas alternadas de vários tamanhos. Esta é a fragmentação do espaço livre. Quando um arquivo é criado e não há uma área de espaço contíguo disponível para sua alocação inicial, o espaço deve ser atribuído em fragmentos.

Quando um arquivo é modificado de modo a ficar maior, ele pode exceder o espaço inicialmente alocado a ele, outra alocação deve ser atribuída em outro lugar e o arquivo se torna fragmentado.

### Nomes de arquivo

Um **nome de arquivo** é usado para identificar uma localização de armazenamento no sistema de arquivos. A maioria dos sistemas de arquivos possuem restrições de comprimento de nomes de arquivo.

Em alguns sistemas de arquivos, nomes de arquivos não são sensíveis a caixa (isto é, os nomes MEUARQUIVO e meuarquivo referem-se ao mesmo arquivo), em outros, os nomes de arquivos são sensíveis a caixa (isto é, os nomes MEUARQUIVO, MeuArquivo e meuarquivo referem-se a três arquivos separados).

A maioria dos sistemas de arquivos modernos permitem que nomes de arquivos contendam uma grande variedade de caracteres do conjunto de caracteres Unicode. Entretanto, eles podem possuir restrições no uso de certos caracteres especiais, impedindo-os de serem usados dentro de nomes de arquivos. Esses caracteres podem ser usados para indicar um dispositivo, tipo de dispositivo, prefixo de diretório, separador de caminho de arquivo ou tipo de arquivo.

### Diretórios

Os sistemas de arquivos geralmente possuem **diretórios** (também chamados de **pastas**) que permitem ao usuário agrupar arquivos em coleções separadas. Isso pode ser implementado associando o nome do arquivo a um índice em uma tabela de conteúdos ou a um inode em um sistema de arquivos do tipo Unix.

As estruturas de diretórios podem ser planas (ou seja, lineares) ou permitir hierarquias nas quais os diretórios podem conter subdiretórios. O primeiro sistema de arquivos a suportar hierarquias arbitrárias de diretórios foi usado no sistema operacional Multics. Os sistemas de arquivos nativos de sistemas semelhantes ao Unix também suportam hierarquias de diretórios arbitrários, como, por exemplo, o Hierarchical File System da Apple e seu sucessor HFS+ no Mac OS clássico, o sistema de arquivos FAT no MS-DOS 2.0, e versões posteriores, e no Microsoft Windows, o sistema de arquivos NTFS na família Windows NT de sistemas operacionais e o ODS-2 (On-Disk Structure-2) e níveis mais altos do sistema de arquivos Files-11 no OpenVMS.

### Metadados

Outras informações de escrituração contábil normalmente são associadas a cada arquivo dentro de um sistema de arquivos. O comprimento dos dados contidos em um arquivo pode ser armazenado como o número de blocos alocados para o arquivo ou como uma contagem de bytes. A hora em que o arquivo foi modificado pela última vez pode ser armazenada como o registro de data e hora do arquivo. Os sistemas de arquivos podem armazenar o horário de criação do arquivo, a hora em que ele foi acessado pela última vez, a hora em que os metadados do arquivo foram alterados ou a hora em que o arquivo foi copiado pela última vez. Outras informações podem incluir o tipo de dispositivo do arquivo (por exemplo, bloco, caractere, soquete, subdiretório etc.), seu ID de usuário e ID de grupo de proprietário, suas permissões de acesso e outros atributos de arquivo (por exemplo, se o arquivo é somente leitura, executável, etc.).

Um sistema de arquivos armazena todos os metadados associados ao arquivo - incluindo o nome do arquivo, o comprimento do conteúdo de um arquivo e o local do arquivo na hierarquia de pastas - separado do conteúdo do arquivo.

A maioria dos sistemas de arquivos armazena os nomes de todos os arquivos em um diretório em um único local - a tabela de diretórios desse diretório - que geralmente é armazenado como qualquer outro arquivo. Muitos sistemas de arquivos colocam apenas alguns dos metadados para um arquivo na tabela de diretórios e o restante dos metadados para esse arquivo em uma estrutura completamente separada, como o inode.

A maioria dos sistemas de arquivos também armazena metadados não associados a nenhum arquivo específico. Esses metadados incluem informações sobre regiões não utilizadas - bitmap de espaço livre, mapa de disponibilidade de blocos - e informações sobre setores defeituosos. Geralmente, essas informações sobre um grupo de alocação são armazenadas dentro do próprio grupo de alocação.

Atributos adicionais podem ser associados em sistemas de arquivos, como NTFS, XFS, ext2, ext3, algumas versões do UFS e HFS+, usando atributos estendidos de arquivo. Alguns sistemas de arquivos fornecem atributos definidos pelo usuário, como o autor do documento, a codificação de caracteres de um documento ou o tamanho de uma imagem.

Alguns sistemas de arquivos permitem que diferentes coleções de dados sejam associadas a um nome de arquivo. Essas coleções separadas podem ser chamadas de *fluxos* ou *garfos*. A Apple usa há muito tempo um sistema de arquivos bifurcado no Macintosh, e a Microsoft suporta fluxos no NTFS.

Alguns sistemas de arquivos mantêm várias revisões anteriores de um arquivo sob um único nome de arquivo. O nome do arquivo por si só recupera a versão mais recente, enquanto a versão salva anteriormente pode ser acessada usando uma convenção de nomenclatura especial, como "nome\_do\_arquivo;4" ou "nome\_do\_arquivo(-4)" para acessar a versão quatro salva.

Veja comparação de sistemas de arquivos para obter detalhes sobre quais sistemas de arquivos suportam quais tipos de metadados.

### **Sistema de arquivos como uma interface abstrata de usuário**

Em alguns casos, um sistema de arquivos pode não fazer uso de um dispositivo de armazenamento, mas pode ser usado para organizar e representar o acesso a qualquer dado, seja armazenado ou gerado dinamicamente (por exemplo, procs).

#### **Utilitários**

Os sistemas de arquivos incluem utilitários para inicializar, alterar parâmetros e remover uma instância do sistema de arquivos. Alguns incluem a capacidade de estender ou truncar o espaço alocado para o sistema de arquivos.

Os utilitários de diretório podem ser usados para criar, renomear e excluir *entradas de diretório*, que também são conhecidas como *dentries* (singular: *dentry*), e para alterar metadados associados a um diretório.

Os utilitários de diretório também podem incluir recursos para criar links adicionais para um diretório (hard links no Unix), para renomear links pai (".." em sistemas operacionais tipo Unix), e para criar links bidirecionais para arquivos.

Os utilitários de arquivos criam, listam, copiam, movem e excluem arquivos e alteram os metadados. Eles podem truncar dados, truncar ou estender a alocação de espaço, anexar, mover e modificar arquivos no local.

Dependendo da estrutura subjacente do sistema de arquivos, eles podem fornecer um mecanismo para preceder ou truncar o início de um arquivo, inserir entradas no meio de um arquivo ou excluir entradas de um arquivo.

Utilitários para liberar espaço para arquivos excluídos, se o sistema de arquivos fornecer uma função de exclusão, também pertence a essa categoria.

Alguns sistemas de arquivos adiam operações como a reorganização do espaço livre, o apagamento seguro de espaço livre e a reconstrução de estruturas hierárquicas, fornecendo utilitários para executar essas funções em momentos de atividade mínima. Um exemplo são os utilitários de desfragmentação do sistema de arquivos.

Algumas das características mais importantes dos utilitários do sistema de arquivos envolvem atividades de supervisão que podem envolver ignorar a propriedade ou o acesso direto ao dispositivo subjacente. Isso inclui backup e recuperação de alto desempenho, replicação de dados e reorganização de várias estruturas de dados e tabelas de alocação dentro do sistema de arquivos.

### **Tipos e sistemas de arquivos**

Os tipos de sistemas de arquivos podem ser classificados em sistemas de arquivos de disco/fita, sistemas de arquivos de rede e sistemas de arquivos de propósito especial.

## Sistemas de arquivos de disco

Um *sistema de arquivos de disco* aproveita a capacidade da mídia de armazenamento em disco para endereçar dados aleatoriamente em um curto período de tempo. Considerações adicionais incluem a velocidade de acesso aos dados após a solicitação inicial e a antecipação de que os dados seguintes também podem ser solicitados.

Isso permite que vários usuários (ou processos) acessem vários dados no disco sem considerar a localização sequencial dos dados. Exemplos incluem FAT (FAT12, FAT16, FAT32), exFAT, NTFS, HFS e HFS+, HPFS, APFS, UFS, ext2, ext3, XFS, btrfs, ISO 9660, Files-11, Veritas File System, VMFS, ZFS, ReiserFS e UDF. Alguns sistemas de arquivos em disco são sistemas de arquivos de registro ou sistemas de arquivos de controle de versão.

## Discos óticos

O ISO 9660 e o Universal Disk Format (UDF) são dois formatos comuns destinados a discos compactos, DVDs e discos Blu-ray. O Mount Rainier é uma extensão do suporte a UDF desde a série 2.6 do kernel do Linux e desde o Windows Vista que facilita a reescrita para DVDs.

## Sistemas de arquivos de flash

Um *sistema de arquivos de flash* considera as habilidades especiais, o desempenho e as restrições dos dispositivos de memória flash. Frequentemente, um sistema de arquivos em disco pode usar um dispositivo de memória flash como a mídia de armazenamento subjacente, mas é muito melhor usar um sistema de arquivos projetado especificamente para um dispositivo flash.

## Sistemas de arquivos de fita

Um *sistema de arquivos de fita* é um sistema de arquivos e um formato de fita projetado para armazenar arquivos em fita em uma forma autodescritiva. As fitas magnéticas são mídias de armazenamento sequenciais com tempos de acesso a dados aleatórios significativamente maiores do que os discos, o que representa desafios para a criação e gerenciamento eficiente de um sistema de arquivos de uso geral.

Em um sistema de arquivos em disco, geralmente há um diretório de arquivos mestre e um mapa de regiões de dados usadas e livres. Quaisquer adições, alterações ou remoções de arquivos exigem a atualização do diretório e dos mapas usados/livres. O acesso aleatório às regiões de dados é medido em milissegundos, portanto, esse sistema funciona bem para discos.

A fita requer movimento linear para enrolar e desenrolar potencialmente bobinas de mídia muito longas. Esse movimento da fita pode levar de alguns segundos a vários minutos para mover a cabeça de leitura/gravação de uma extremidade da fita para a outra.

Consequentemente, um diretório de arquivos mestre e um mapa de uso podem ser extremamente lentos e ineficientes com fita. Escrever normalmente envolve ler o mapa de uso do bloco para encontrar blocos livres para gravação, atualizar o mapa e o diretório de uso para adicionar os dados e avançar a fita para gravar os dados no local correto. Cada gravação de arquivo adicional requer a atualização do mapa e do diretório e a gravação dos dados, o que pode levar vários segundos para ocorrer em cada arquivo.

Em vez disso, os sistemas de arquivos de fitas normalmente permitem que o diretório de arquivos seja distribuído pela fita intercalada com os dados, chamados de *fluxo contínuo*, de modo que os movimentos repetitivos e demorados da fita não sejam necessários para gravar novos dados.

No entanto, um efeito colateral desse design é que a leitura do diretório de arquivos de uma fita geralmente requer a verificação da fita inteira para ler todas as entradas de diretório espalhadas.

A maioria dos softwares de arquivamento de dados que funciona com armazenamento em fita armazenará uma cópia local do catálogo de fitas em um sistema de arquivos em disco, para que a adição de arquivos a uma fita possa ser feita rapidamente sem necessidade de redigitalizar a mídia de fita. Geralmente, a cópia do catálogo de fitas local é descartada se não for usada por um período de tempo especificado. Nesse ponto, a fita deve ser verificada novamente se for usada no futuro.



A IBM desenvolveu um sistema de arquivos para fita chamado Linear Tape File System (em português: Sistema de Arquivos de Fita Linear). A implementação IBM desse sistema de arquivos foi lançada como o produto de código aberto IBM Linear Tape File - Single Drive Edition (LTFS-SDE). O Sistema de Arquivos de Fita Linear usa uma partição separada na fita para registrar os metadados de índice, evitando assim os problemas associados às entradas de diretório de dispersão em toda a fita.

### Formatação de fita

Escrever dados em uma fita, apagar ou formatar uma fita é um processo demorado e que pode levar várias horas em fitas grandes.<sup>[a]</sup> Com muitas tecnologias de fita de dados, não é necessário formatar a fita antes de sobrescrever novos dados nela. Isso se deve à natureza inerentemente destrutiva de sobrescrever dados em mídia sequencial.

Devido ao tempo que pode ser necessário para formatar uma fita, normalmente as fitas são pré-formatadas para que o seu usuário não precise gastar tempo preparando cada nova unidade para uso. Tudo o que é normalmente necessário é escrever um rótulo de mídia de identificação para a fita antes do uso, e mesmo isso pode ser escrito automaticamente pelo software quando uma nova fita é usada pela primeira vez.

### Sistemas de arquivos de banco de dados

Outro conceito para gerenciamento de arquivos é a ideia de um sistema de arquivos baseado em banco de dados. Em vez de, ou além do gerenciamento estruturado hierárquico, os arquivos são identificados por suas características, como tipo de arquivo, tópico, autor ou metadados ricos semelhantes.<sup>[11]</sup>

O IBM DB2 para i<sup>[12]</sup> (anteriormente conhecido como DB2/400 e DB2 para i5/OS) é um sistema de arquivos de banco de dados como parte do sistema operacional IBM i<sup>[13]</sup> baseado em objetos (anteriormente conhecido como OS/400 e i5/OS), incorporando um armazenamento de nível único e rodando no IBM Power Systems (anteriormente conhecido como AS/400 e iSeries), projetado por Frank G. Soltis, ex-cientista-chefe da IBM para o IBM i. Por volta de 1978 a 1988, Frank G. Soltis e sua equipe da IBM Rochester projetaram e aplicaram com sucesso tecnologias como o sistema de arquivos de banco de dados, onde outros, como a Microsoft, mais tarde não conseguiram realizar.<sup>[14]</sup> Essas tecnologias são informalmente conhecidas como 'Fortress Rochester' e eram, em alguns aspectos básicos, estendidas desde as primeiras tecnologias de mainframe, mas de muitas maneiras, mais avançadas do ponto de vista tecnológico.

Alguns outros projetos que não são sistemas de arquivos de banco de dados "puros", mas que usam alguns aspectos de um sistema de arquivos de banco de dados:

Muitos sistemas de gerenciamento de conteúdo da Web usam um SGBD relacional para armazenar e recuperar arquivos. Por exemplo, os arquivos XHTML são armazenados como campos XML ou de texto, enquanto os arquivos de imagem são armazenados como campos blob. As instruções SQL SELECT (com XPath opcional) recuperam os arquivos e permitem o uso de uma lógica sofisticada e de associações de informações mais ricas do que os "sistemas de arquivos comuns". Muitos CMSs também têm a opção de armazenar apenas metadados no banco de dados, com o sistema de arquivos padrão usado para armazenar o conteúdo dos arquivos.

Sistemas de arquivos muito grandes, incorporados por aplicativos como o Apache Hadoop e o Google File System, usam alguns conceitos de *sistema de arquivos de banco de dados*.

### Sistemas e arquivos transacionais

Alguns programas precisam atualizar vários arquivos "todos de uma vez". Por exemplo, uma instalação de software pode escrever binários de programas, bibliotecas e arquivos de configuração. Se a instalação do software falhar, o programa poderá ficar inutilizável. Se a instalação estiver atualizando um utilitário de sistema principal, como o shell de comando, todo o sistema poderá ficar inutilizável.

O processamento de transações introduz a garantia de isolamento, que afirma que as operações dentro de uma transação estão ocultas de outros encadeamentos no sistema até que a transação seja confirmada e que as operações interferentes no sistema sejam serializadas corretamente com a transação.

As transações também fornecem a garantia de atomicidade, garantindo que as operações dentro de uma transação sejam todas comprometidas ou que a transação possa ser abortada e o sistema descarte todos os seus resultados parciais.

Isso significa que, se houver um travamento ou falha de energia, após a recuperação, o estado armazenado será consistente. O software será completamente instalado ou a instalação com falha será completamente revertida, mas uma instalação parcial não utilizável não será deixada no sistema.

O Windows, começando com o Vista, adicionou suporte a transações para o NTFS, em um recurso chamado de NTFS Transacional, mas seu uso hoje é desencorajado.<sup>[15]</sup> Há vários protótipos de pesquisa de sistemas de arquivos transacionais para sistemas UNIX, incluindo o sistema de arquivos Valor,<sup>[16]</sup> Amino,<sup>[17]</sup> LFS<sup>[18]</sup> e um sistema de arquivos ext3 transacional no kernel do TxOS,<sup>[19]</sup> bem como sistemas de arquivos transacionais destinados a sistemas incorporados, como o TFFS.<sup>[20]</sup>

Garantir a consistência em várias operações do sistema de arquivos é difícil, se não impossível, sem transações do sistema de arquivos.

O bloqueio de arquivos pode ser usado como um mecanismo de controle de simultaneidade para arquivos individuais, mas normalmente não protege a estrutura de diretórios ou os metadados de arquivos. Por exemplo, o bloqueio de arquivos não pode impedir as condições de corrida do TOCTTOU em ligações simbólicas. O bloqueio de arquivos também não pode reverter automaticamente uma operação com falha, como uma atualização de software, pois isso requer atomicidade.

Os sistemas de arquivos de registro são uma técnica usada para introduzir a consistência no nível da transação nas estruturas do sistema de arquivos. Transações de registro não são expostas a programas como parte da API do SO, eles são usados apenas internamente para garantir consistência na granularidade de uma única chamada de sistema.

Os sistemas de backup de dados normalmente não fornecem suporte para backup direto de dados armazenados de maneira transacional, o que dificulta a recuperação de conjuntos de dados confiáveis e consistentes. A maioria dos softwares de backup simplesmente observa quais arquivos foram alterados desde um determinado momento, independentemente do estado transacional compartilhado entre vários arquivos no conjunto de dados geral. Como uma solução alternativa, alguns sistemas de banco de dados simplesmente produzem um arquivo de estado arquivado contendo todos os dados até aquele ponto, e o software de backup apenas faz backup e não interage diretamente com os bancos de dados transacionais ativos. A recuperação requer recriação separada do banco de dados do arquivo de estado, depois que o arquivo foi restaurado pelo software de backup.

### **Sistemas de arquivos de rede**

Um *sistema de arquivos de rede* é um sistema de arquivos que atua como um cliente para um protocolo de acesso remoto a arquivos, fornecendo acesso a arquivos em um servidor. Os programas que usam interfaces locais podem criar, gerenciar e acessar de forma transparente diretórios e arquivos hierárquicos em computadores conectados em rede remota. Exemplos de sistemas de arquivos de rede incluem clientes para os protocolos NFS, AFS, SMB e clientes semelhantes a sistemas de arquivos para FTP e WebDAV.

### **Sistemas de arquivos compartilhados**

Um *sistema de arquivos compartilhado* é aquele no qual uma quantidade de máquinas (normalmente servidores) possuem acesso ao mesmo subsistema de disco externo (normalmente um SAN). O sistema de arquivos arbitra acesso a este subsistema, prevenindo colisões de escrita. Exemplos incluem o GFS2 da Red Hat, o GPFS da IBM, o SFS da DataPlow, o CXFS da SGI e o StorNext da Quantum Corporation.

### **Sistemas de arquivos especiais**

Um *sistema de arquivos especial* apresenta elementos não-arquivos de um sistema operacional como se fossem arquivos, para que possam ser acionados usando as APIs do sistema de arquivos. Isso é mais comumente feito em sistemas operacionais do tipo Unix, mas os dispositivos recebem nomes de arquivos em alguns sistemas operacionais não baseados em Unix também.

### Sistemas de arquivos de dispositivo

Um *sistema de arquivos de dispositivo* representa dispositivos de E/S e pseudo-dispositivos como arquivos, chamados de arquivos de dispositivo. Exemplos em sistemas do tipo Unix incluem o devfs e, em sistemas Linux 2.6, o udev. Em sistemas não-Unix, como o TOPS-10 e outros sistemas operacionais influenciados por ele, nos quais o nome de arquivo ou nome de caminho completo de um arquivo pode incluir um prefixo de dispositivo, dispositivos diferentes daqueles que contêm sistemas de arquivos são referenciados por um prefixo de dispositivo especificando o dispositivo, sem nada seguindo-o.

### Outros sistemas de arquivos especiais

No kernel do Linux, o configfs e o sysfs fornecem arquivos que podem ser usados para consultar o kernel em busca de informações e configurar entidades no kernel.

O procfs mapeia processos e, no Linux, outras estruturas de sistemas operacionais em um espaço no arquivo.

### Sistema de arquivos mínimo / armazenamento de áudio-cassete

Nos anos 70, os dispositivos de fita digital e de disco eram muito caros para alguns dos primeiros usuários de microcomputadores. Foi desenvolvido um sistema básico de armazenamento de dados barato que usava fita cassete de áudio comum.

Quando o sistema precisou escrever dados, o usuário foi notificado para pressionar "RECORD" ("GRAVAR") no gravador de cassetes, depois pressionar "RETURN" ("ENTER") no teclado para notificar o sistema que o gravador estava gravando. O sistema escreveu um som para fornecer sincronização de horário, modulou sons que codificavam um prefixo, os dados, uma soma de verificação e um sufixo. Quando o sistema precisava ler os dados, o usuário era instruído a pressionar "PLAY" ("TOCAR") no gravador de cassetes. O sistema ouvia os sons na fita esperando até que uma explosão de som pudesse ser reconhecida como a sincronização. O sistema então interpretaria os sons subsequentes como dados. Quando a leitura dos dados fosse concluída, o sistema notificaria o usuário para pressionar "STOP" ("PARAR") no gravador. Era primitivo, mas funcionava (na maior parte do tempo). Os dados eram armazenados sequencialmente, geralmente em um formato sem nome, embora alguns sistemas (como a série de computadores Commodore PET) permitissem que os arquivos fossem nomeados. Vários conjuntos de dados podem ser gravados e localizados ao avançar rapidamente a fita e observar no contador de fita para encontrar o início aproximado da próxima região de dados na fita. O usuário pode ter que ouvir os sons para encontrar o ponto certo para começar a tocar a próxima região de dados. Algumas implementações incluíam até mesmo sons audíveis intercalados com os dados.

### Sistemas de arquivos planos

Em um sistema de arquivos plano, não há subdiretórios. Entradas de diretório para todos os arquivos são armazenadas em um único diretório.

Quando a mídia de disquete esteve disponível pela primeira vez, esse tipo de sistema de arquivos era adequado devido à quantidade relativamente pequena de espaço de dados disponível. As máquinas CP/M apresentavam um sistema de arquivos planos, onde os arquivos podiam ser atribuídos a uma das 16 *áreas de usuário* e as operações de arquivos genéricos estreitavam para funcionar em uma delas, em vez de padronizar para trabalhar em todas elas. Essas áreas de usuário não eram mais do que atributos especiais associados aos arquivos, ou seja, não era necessário definir uma cota específica para cada uma dessas áreas e os arquivos poderiam ser adicionados a grupos enquanto ainda houvesse espaço de armazenamento livre no disco. O antigo Apple Macintosh também apresentava um sistema de arquivos plano, o Macintosh File System. Não era comum que o programa de gerenciamento de arquivos (Macintosh Finder) criasse a ilusão de um sistema de arquivamento parcialmente hierárquico em cima do EMFS. Essa estrutura exigia que cada arquivo tivesse um nome exclusivo, mesmo que parecesse estar em uma pasta separada. O IBM DOS/360 e o OS/360 armazenam entradas para todos os arquivos em um pacote de disco (*volume*) em um diretório no pacote chamado de *Tabela de Conteúdo do Volume* (VTOC).

Enquanto simples sistemas de arquivos planos tornam-se desajeitados à medida que o número de arquivos cresce e dificultava a organização de dados em grupos de arquivos relacionados.



Uma adição recente à família de sistemas de arquivos simples é o S3 da Amazon, um serviço de armazenamento remoto, que é intencionalmente simplista para permitir que os usuários personalizem como seus dados são armazenados. As únicas construções são os "buckets" (imaginem uma unidade de disco de tamanho ilimitado) e objetos (semelhantes, mas não idênticos ao conceito padrão de um arquivo). O gerenciamento avançado de arquivos é permitido pela capacidade de usar praticamente qualquer caractere (incluindo '/') no nome do objeto e a capacidade de selecionar subconjuntos do conteúdo do bloco com base em prefixos idênticos.

### 1. HFS (Hierarchical File System):

**Características:** Desenvolvido pela Apple para o sistema operacional Mac OS, o HFS foi projetado para gerenciar arquivos e diretórios hierarquicamente.

**Metadados:** Armazena metadados como o tamanho do arquivo, permissões, timestamps (criação, modificação e acesso), e IDs de arquivos.

**Organização física:** Utiliza uma estrutura de árvore B\* para organizar arquivos e diretórios, o que otimiza o acesso e a recuperação dos dados.

### 2. EXT2 (Second Extended File System):

**Características:** Projetado para sistemas Linux, o EXT2 não suporta journaling, tornando-o mais rápido, porém menos seguro em caso de falhas de energia ou interrupções no sistema.

**Metadados:** Armazena informações como tipo de arquivo, permissões, timestamps, número de links e IDs de usuários e grupos. Também mantém registros de blocos alocados a um arquivo.

**Organização física:** Divide o disco em blocos e grupos de blocos. Cada grupo contém tabelas de inodes e blocos de dados. Usa tabelas de blocos e inodes para gerenciar arquivos.

### 3. EXT3 (Third Extended File System):

**Características:** Similar ao EXT2, mas com suporte a journaling, o que adiciona uma camada de segurança ao sistema de arquivos, garantindo a integridade após falhas abruptas.

**Metadados:** Os metadados são geridos de forma semelhante ao EXT2, mas com o adicional de um diário (journal) que rastreia alterações antes de serem aplicadas, garantindo maior recuperação de dados em caso de falhas.

**Organização física:** Estrutura semelhante ao EXT2, com a diferença de incluir o journal para registrar as operações pendentes e completar transações antes de as mudanças serem gravadas no disco.

### 4. REISERFS:

**Características:** Famoso por sua eficiência em lidar com muitos arquivos pequenos, o ReiserFS também inclui journaling, como o EXT3, mas com melhor desempenho em certos cenários.

**Metadados:** Gerencia informações de arquivos de maneira altamente compactada, otimizando o armazenamento de metadados.

**Organização física:** Usa árvores B+ para organizar os arquivos, otimizando a busca e a inserção de dados. Suporta alocação dinâmica, o que permite a compactação de pequenos arquivos no próprio sistema de arquivos.

### 5. NTFS (New Technology File System):

**Características:** Sistema de arquivos usado pelo Windows a partir do Windows NT, projetado para ser robusto e escalável, com suporte a grandes volumes e longos nomes de arquivos.

**Metadados:** Além de armazenar dados convencionais como permissões, timestamps e IDs, o NTFS também gerencia fluxos de dados alternativos e listas de controle de acesso (ACLs), o que permite um controle granular de permissões.

**Organização física:** Utiliza a Master File Table (MFT), onde cada arquivo ou diretório é descrito por uma entrada. O journaling no NTFS é chamado de sistema de recuperação de falhas, registrando operações antes que sejam completadas.

#### 6. FAT16 (File Allocation Table 16-bit):

**Características:** Um sistema de arquivos antigo usado em DOS e versões anteriores do Windows. Sua principal limitação é o suporte a volumes de até 2 GB e arquivos com nomes curtos (8.3).

**Metadados:** Armazena o nome do arquivo, atributos de leitura/escrita, data e hora de criação e modificação, além do endereço inicial do arquivo na FAT.

**Organização física:** Utiliza uma tabela de alocação de arquivos (FAT) que mapeia os blocos de disco usados por um arquivo. Cada entrada na FAT contém o próximo bloco de disco que compõe o arquivo.

#### 7. FAT32 (File Allocation Table 32-bit):

**Características:** Uma evolução do FAT16, o FAT32 suporta volumes maiores (até 2 TB) e arquivos de maior tamanho. Ele é amplamente compatível, sendo utilizado em muitos dispositivos externos, como pendrives.

**Metadados:** Similar ao FAT16, mas com suporte a nomes de arquivos mais longos e timestamps mais precisos.

**Organização física:** Usa uma tabela de alocação de arquivos (FAT) maior, com endereços de bloco de 32 bits, permitindo a alocação de volumes maiores.

#### 8. VFAT (Virtual File Allocation Table):

**Características:** Introduzido no Windows 95, o VFAT é uma extensão do FAT16, projetado para suportar longos nomes de arquivos (até 255 caracteres) enquanto mantém compatibilidade com FAT16.

**Metadados:** Armazena metadados adicionais para gerenciar nomes longos e mantém compatibilidade com os atributos de arquivos tradicionais do FAT16.

**Organização física:** Similar ao FAT16, mas utiliza tabelas auxiliares para gerenciar longos nomes de arquivos.

#### 9. ISO9660:

**Características:** Sistema de arquivos utilizado principalmente em mídias de CD-ROM, projetado para garantir a compatibilidade entre diferentes sistemas operacionais.

**Metadados:** Armazena informações básicas, como permissões, nomes de arquivos (com limite de 32 caracteres) e datas de criação e modificação.

**Organização física:** Os arquivos são organizados em um formato hierárquico. Tem um limite de profundidade de diretórios e tamanho de arquivos.

#### 10. Joliet:

**Características:** Uma extensão do ISO9660, desenvolvida pela Microsoft para permitir o uso de nomes de arquivos longos e compatibilidade com Unicode, aumentando a interoperabilidade entre sistemas.

**Metadados:** Similar ao ISO9660, mas com suporte a caracteres Unicode e nomes de arquivos mais longos (até 64 caracteres).

**Organização física:** Usa uma estrutura hierárquica semelhante ao ISO9660, mas permite maior profundidade de diretórios e maior compatibilidade.

#### 11. UDF (Universal Disk Format):

