



# SISTEMAS OPERATIVOS

## Sistemas de ficheiros

António Godinho

# SISTEMA DE FICHEIROS: ORGANIZAÇÃO DE DADOS(1)

## **Ficheiro:**

- Colecção de dados identificada por um nome e manipulada como unidade.
- Reside em dispositivos de memória secundária (ex. discos magnéticos).

**Sistema de ficheiros: interface criada pelo SO que facilita a navegação e acesso aos ficheiros dos utilizadores e do próprio SO.**

**Organização de dados: a informação é armazenada nos sistemas computacionais de acordo com uma hierarquia de dados:**

- Nível mais baixo: padrões de bits que representam os itens de dados manipuláveis pelo sistema computacional.
- Nível seguinte na hierarquia: organização em padrões de tamanho fixo como bytes, caracteres e palavras:
  - Byte: 8 bits.
  - Palavra (word): número de bits que a CPU pode manipular em cada instrução.
  - Mapa de caracteres (byte ou grupo de bytes) que representam símbolos como letras, números, sinais de pontuação, etc (ASCII, EBCDIC, Unicode).
  - Campo (field): grupo de caracteres/bytes.
  - Registo (record): grupo de campos.
  - Ficheiro (file): grupo de registos relacionados.

# SISTEMA DE FICHEIROS: ORGANIZAÇÃO DE DADOS(1)

- Organização de dados: a informação é armazenada nos sistemas computacionais de acordo com uma hierarquia de dados (continuação):
  - Último nível:
    - Sistema de ficheiros ou base de dados.
    - Volume ou partições: unidade de armazenamento com múltiplos ficheiros (múltiplos sistemas de ficheiros).
    - Memória secundária (ex. disco magnético) : unidade de armazenamento com múltiplos volumes/partições.

# SISTEMA DE FICHEIROS: ORGANIZAÇÃO DE DADOS(2)

## Ficheiro:

- colecção de dados identificada por um nome e manipulada como unidade através das operações: open, close, create, destroy, copy, rename, list, etc.
- Os itens individuais de dados de um ficheiro podem ser manipulados através das operações: read, write, update, insert, delete, etc.
- As características dos ficheiros incluem: localização, acessibilidade, volatilidade, tipo, actividade, etc.

# SISTEMA DE FICHEIROS: ORGANIZAÇÃO DE DADOS(2)

## Sistema de ficheiros:

- Organiza os ficheiros e gere o acesso aos dados desses ficheiros.
- Responsável pela gestão de ficheiros, gestão de acesso a ficheiros, mecanismos que garantem a consistência e integridade dos ficheiros e mecanismos que gerem o acesso aos dados.
- Tem, como principal objectivo, a gestão do espaço de memória de armazenamento secundário, particularmente os discos magnéticos.
- Principais características:
  - Exibir independência relativamente ao dispositivo periférico.
  - Os utilizadores devem poder referir-se aos ficheiros através de nomes simbólicos em vez de utilizar nomes de dispositivos físicos.
  - Deve proporcionar capacidades de salvaguarda (backup) e recuperação (recovery) de modo a prevenir perdas acidentais ou destruição maliciosa de informação (segurança).
  - Pode, adicionalmente, disponibilizar mecanismos de encriptação e desencriptação de modo a tornar a informação unicamente utilizável por quem de direito.

# SISTEMA DE FICHEIROS: DIRETÓRIOS

## Directórios:

- ficheiros que armazenam o nome e a localização de outros ficheiros no sistema de ficheiros.
- permitem organizar e localizar rapidamente ficheiros.
- cada item do ficheiro de directório está associado a um ficheiro e armazena informações tais como: nome do ficheiro, localização, tamanho, tipo, data e hora da última actualização, criação e acesso.

## Sistema de ficheiros com um único nível(flat file system):

- Organização mais simples de um sistema de ficheiros.
- Armazena todos os ficheiros num só directório.
- Dois ficheiros não podem ter o mesmo nome.
- O sistema de ficheiros tem de efectuar uma pesquisa linear no directório de modo a localizar um ficheiro, o que conduz a um desempenho fraco.

# SISTEMA DE FICHEIROS: DIRETÓRIOS

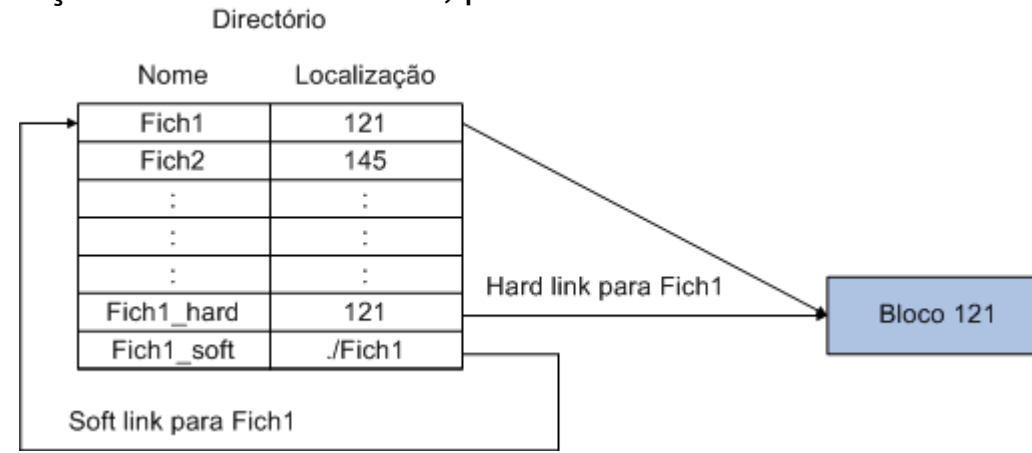
## Sistema hierárquico de ficheiros:

- Cada ficheiro de directório armazena itens relativos a ficheiros e itens relativos a outros directórios.
- Um primeiro directório (raiz) indica o ponto inicial de referência da estrutura
- hierarquizada de ficheiros.
- Os nomes de ficheiros apenas devem ser únicos em cada directório.
- O nome de um ficheiro é constituído pelo conjunto de nomes (caminho - pathname) de directórios que são percorridos desde a raiz até ao directório que contém esse ficheiro.
- Directório de trabalho ou directório corrente:
  - simplifica a navegação recorrendo a caminhos relativos.
  - permite ao utilizador especificar um ficheiro recorrendo a um caminho relativo ao directório de trabalho.
  - caminho absoluto: directório corrente + caminho relativo.

# SISTEMA DE FICHEIROS: LINKS

## Links:

- Item do directório que referencia um ficheiro ou directório localizado num outro directório.
- Facilita a partilha de ficheiros e permite um fácil acesso a ficheiros dispersos por
- toda a estrutura de directórios do sistema de ficheiros.
- Soft link: Item do directório que armazena um caminho (localização lógica) para outro ficheiro. Deste modo, se a localização física for alterada, não é necessário actualizar o link. No entanto, se o ficheiro for transferido para outro directório, o link deixa de ser válido (broken link).
- Hard link: Item do directório que especifica a localização física do ficheiro (tipicamente um número de bloco) no dispositivo de armazenamento. Se a localização física for alterada, pode referenciar dados inválidos(o SO garante que isto não acontece!!).





# SISTEMA DE FICHEIROS: METADADOS

## Metadados:

- Informação que protege a integridade do sistema de ficheiros.
- Não directamente acessível pelos utilizadores.
- Muitos sistemas de ficheiros criam um superbloco que armazena a informação crítica à integridade do sistema de ficheiros, nomeadamente: a identificação do sistema de ficheiros, o tipo, a localização do directório raiz, a lista de blocos livres, etc. Para reduzir o risco de perda de dados, este superbloco é muitas vezes replicado por cópias redundantes salvaguardadas e dispersas ao longo do dispositivo de armazenamento.
- As operações de abertura ou criação de ficheiros devolvem um identificador numérico (número inteiro não negativo, designado por descritor de ficheiro) que é usado em operações futuras de manipulação do conteúdo do ficheiro. O SO mantém uma tabela de ficheiros abertos por cada um dos processos em execução. Em cada elemento desta tabela é guardada uma estrutura de metadados (file control block) que inclui, entre outras informações, dados de controlo de acesso, localização física do ficheiro no dispositivo de armazenamento secundário, ponteiro para a posição de leitura/escrita dentro do ficheiro, etc.

# SISTEMA DE FICHEIROS: METADADOS

## Montagem de sistemas de ficheiros (mounting)

Combina múltiplos sistemas de ficheiros num único espaço de nomes de modo a que os ficheiros possam ser referenciados a partir de um único directório raiz.

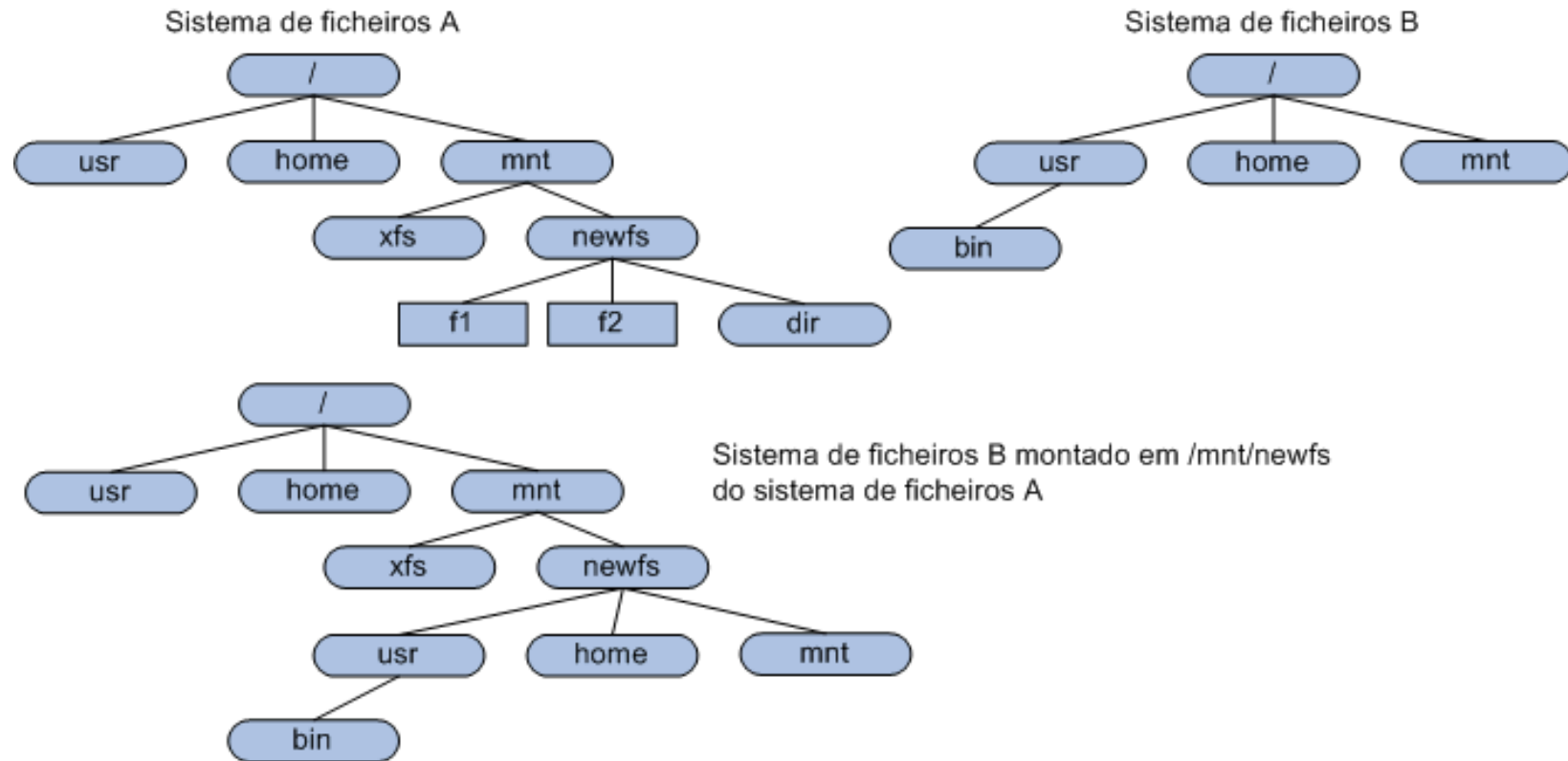
Associa um directório do sistema nativo de ficheiros (ponto de montagem) à raiz do sistema de ficheiros a montar.

Quando o sistema nativo de ficheiros encontra o ponto de montagem, recorre a uma tabela para determinar o tipo de sistema de ficheiros que foi montado naquele ponto.

Os utilizadores podem apenas criar soft links (ligações lógicas) entre ficheiros dos dois sistemas

# SISTEMA DE FICHEIROS: MONTAGEM

## Exemplo:

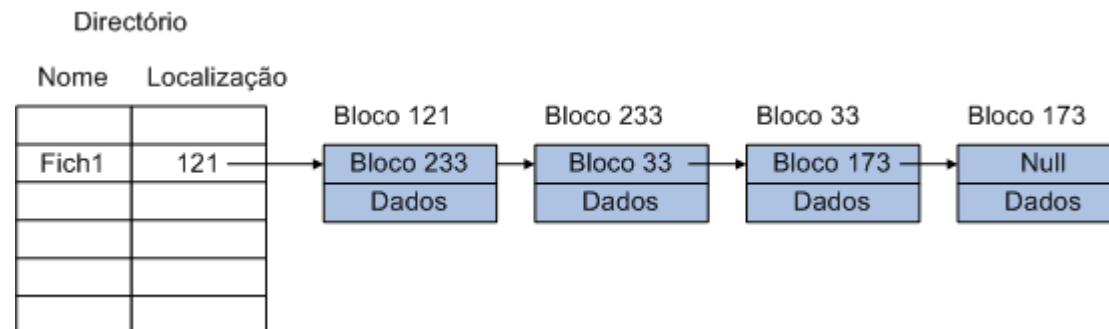


# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

- Memória secundária organizada em blocos de memória de tamanho fixo que constituem a unidade mínima endereçável (dispositivo orientado ao bloco).
- Alocação de ficheiros: afectação ou libertação dos blocos da memória secundária associados a um ficheiro.
- Alocação contígua:
  - Coloca os dados de um ficheiro em blocos contíguos da memória secundária.
  - Vantagem: acesso rápido quando o ficheiro é acedido de forma sequencial.
  - Desvantagens:
    - Gestão difícil quando o tamanho do ficheiro aumenta e diminui ao longo do tempo (fraco desempenho).
    - É necessário saber à partida o tamanho do ficheiro a ser criado (na maioria dos casos não é possível conhecer este tamanho).
- Alocação não contígua:
  - Implementada na maioria dos sistemas de ficheiros dos SO.
  - Principais soluções:
    - lista ligada;
    - baseada numa tabela;
    - por indexação.

# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

- Alocação não contígua baseada numa lista ligada:
  - Para um ficheiro, o respectivo item do directório aponta para o primeiro bloco do ficheiro.
  - Cada bloco contém os dados e um ponteiro para o bloco seguinte.
  - Para a localização de um registo, a lista ligada deve ser percorrida desde o início. Se os blocos estiverem dispersos pelo dispositivo de armazenamento, os tempos de acesso podem ser bastante elevados.
  - A inserção ou eliminação de registos é feita por manipulação dos ponteiros da lista ligada.



# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

Alocação não contígua baseada numa lista ligada:

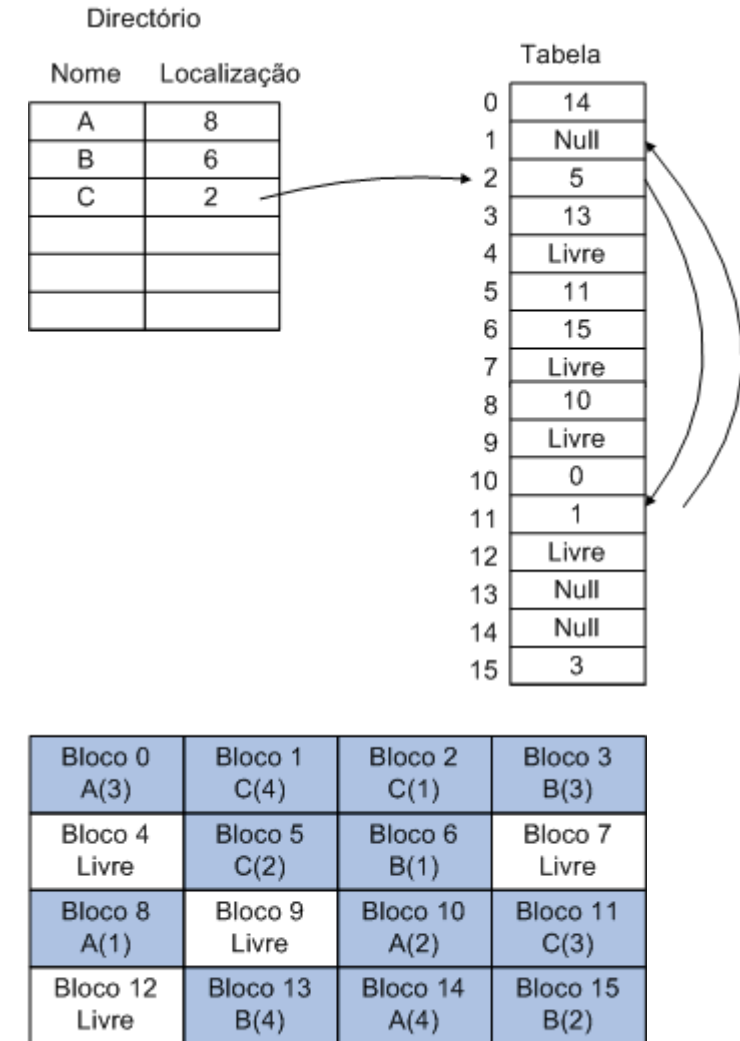
- Recorre a uma tabela (FAT-File Allocation Table) para guardar os ponteiros (números) dos blocos de memória que estão alocados a um ficheiro.
- Para um ficheiro, o respectivo item do directório aponta para o primeiro bloco do ficheiro.
- O número do bloco corrente é usado como índice na tabela onde se encontra o número do bloco seguinte. Se o bloco corrente é o último, o elemento da tabela armazena o valor Null
- A tabela é normalmente mantida em memória principal para melhorar os tempos de acesso.
- A pesquisa do último registo do ficheiro pode obrigar ao sistema de ficheiros a percorrer vários elementos da tabela, podendo ser um processo moroso.
- Solução adoptada pelo sistema de ficheiros FAT da Microsoft.

# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

O NTFS utiliza a MFT (Master File Table), onde são armazenados os atributos dos ficheiros. Cada ficheiro tem uma entrada na MFT. Cada entrada é composta por um cabeçalho, que contém informações necessárias ao funcionamento interno do NTFS (como indicadores que apontam para a localização dos atributos daquele ficheiro).

Os atributos principais são:

- Standard Information Attribute (quando o ficheiro foi criado, modificado, acedido ... read-only, hidden ...)
- File Name Attribute
- Data attribute (armazena o conteúdo do ficheiro – pode ser todo ou parte)
- Security Descriptor Attribute (guarda as ACL's)



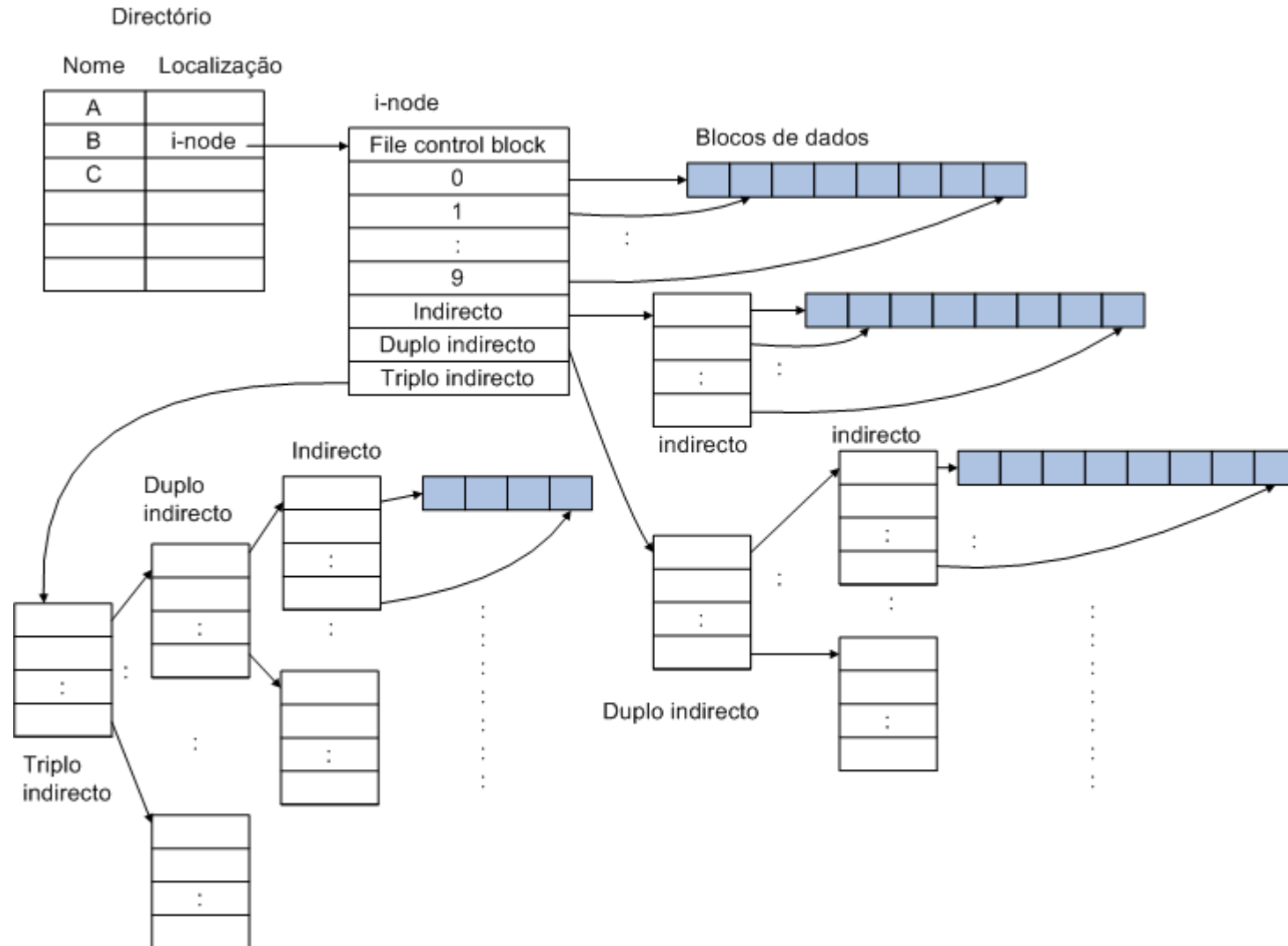
# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

Alocação não contígua por indexação:

- Cada ficheiro contém um bloco de índices ou múltiplos blocos de índices. Cada bloco de índices aponta para uma lista de blocos de dados ou de outros blocos de índices.
- O item do directório aponta para o primeiro bloco de índices. Este bloco armazena ponteiros para os blocos de dados e ponteiros para blocos de índices.
- Solução adoptada pelo sistemas de ficheiros Linux/UNIX. O primeiro bloco do ficheiro é designado por **i-node**.
- Os i-nodes (index nodes) contêm informações a respeito de ficheiros e directórios no sistema de ficheiros. A única coisa que o i-node não contém é o nome do ficheiro. O nome do ficheiro é mantido no directório que também é um tipo de ficheiro.



# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS



# SISTEMA DE FICHEIROS: ALOCAÇÃO DE FICHEIROS

O i-node contém informações sobre as permissões do ficheiro, número de ligações (links), timestamp, blocos duplicados, blocos estragados, associações de tamanho, e ponteiros para os blocos de dados.

O i-node mantém uma contagem de links, que é o número total de referências ao i-node. Blocos duplicados são blocos apontados por dois i-nodes. Blocos estragados ocorrem quando o número de um bloco está além do limite aceitável. O i-node mantém um registo do número de bytes num ficheiro e o número de blocos a ele associados.

O i-node possui também apontadores para os blocos de dados dos ficheiros. Cada i-node possui 16 destes apontadores. Os apontadores de 0 a 11 apontam para os 12 primeiros blocos de dados do ficheiro (blocos directos). O apontador 12 aponta para um bloco indirecto que aponta para mais 1024 blocos de dados. O apontador de número 13 é um bloco indirecto duplo que aponta para 1024 blocos indirectos que por sua vez apontam para 1024 blocos de dados.

Exemplo: Se cada bloco de dados ocupar 8KB

12 ponteiros directos =  $12 * 8 = 96$  KB

Indirecto simples =  $1024 * 8 = 8\,192$  KB

Indirecto duplo =  $1024^2 * 8 = 8\,388\,608$  KB

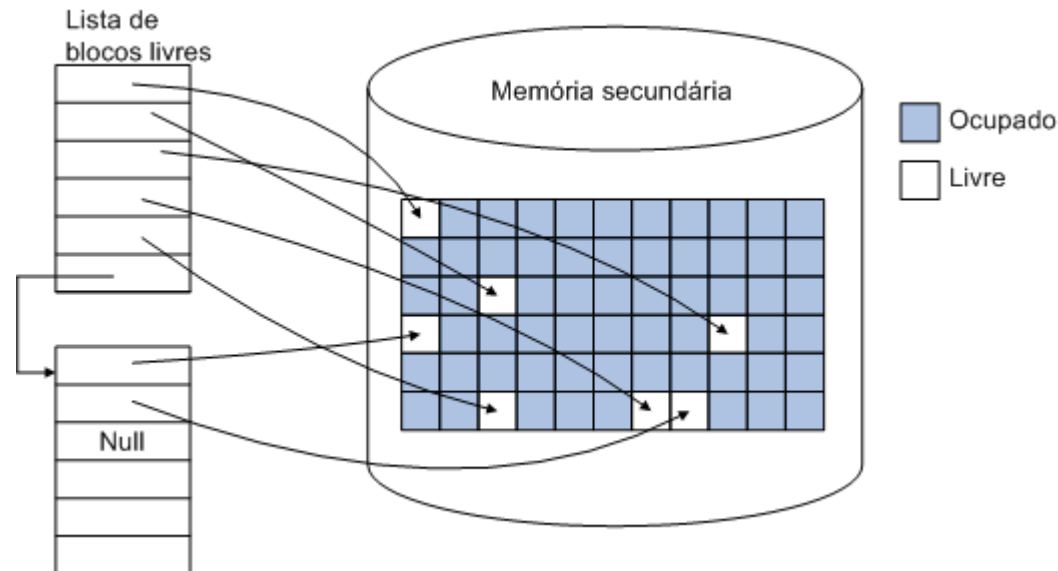
Indirecto triplo será  $1024^3 * 8 = 8\,589\,934\,592$  KB

Para um ficheiro de 6 GB seriam necessários 12 ponteiros directos + 1 indirecto simples + 1 indirecto duplo

# SISTEMA DE FICHEIROS: GESTÃO DO ESPAÇO LIVRE

## Gestão do espaço livre por lista ligada:

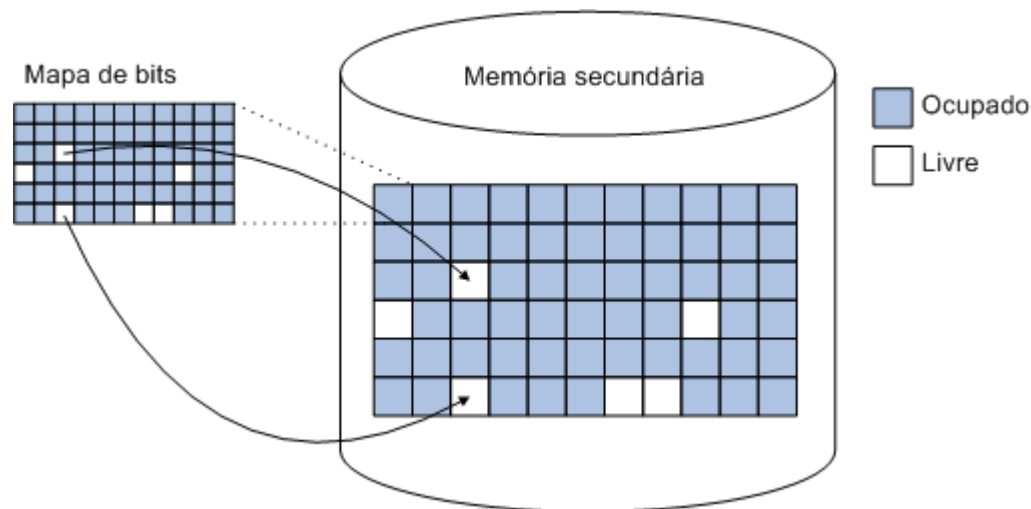
- Alguns sistemas utilizam uma lista ligada de blocos que referenciam os blocos livres da memória secundária.
- Os blocos são alocados do início da lista.
- Os blocos libertados são colocados no fim da lista.
- Devido a este mecanismo de alocação, os blocos de um ficheiro tendem a ficar dispersos, em posições não contíguas (fragmentação), o que tende a aumentar o tempo de acesso.



# SISTEMA DE FICHEIROS: GESTÃO DO ESPAÇO LIVRE

## Gestão do espaço livre por mapa de bits:

- É criado um mapa de bits em que cada bit corresponde a um bloco no dispositivo de memória secundária. O mapa de bits é armazenado no primeiro bloco da memória secundária.
- O valor do  $i$ -ésimo bit (0 ou 1) indica se o  $i$ -ésimo bloco está ou não ocupado.
- Permite ao sistema de ficheiros localizar um conjunto de blocos contíguos, reduzindo desta forma a fragmentação dos ficheiros (menor tempo de acesso).
- O sistema de ficheiros pode ter de percorrer todo o mapa de bits para encontrar um bloco livre.



# TIPOS DE SISTEMAS DE FICHEIROS

Ao instalar uma distribuição de Linux é possível escolher uma grande variedade de tipos de sistemas de ficheiros.

Ext4 é por omissão o tipo de sistema de ficheiros para a maioria das distribuições de Linux, sendo uma evolução do antigo Ext3.

Sendo bastante estável e com provas dadas, é um boa opção em caso de dúvida. Btrfs começa a ser a opção por omissão de muitas das distribuições

Mas porquê Ext4? Porque não Ext3? E o que é *journaling* e a sua importância?

# TIPOS DE SISTEMAS DE FICHEIROS

FAT16 (File Allocation Table 16) e FAT32 (File Allocation Table 32) são sistemas de ficheiros desenvolvidos pela Microsoft para sistemas operativos MS-DOS e Windows.

São utilizados usados para organizar e gerir dados em dispositivos de armazenamento, como discos rígidos, cartões de memória e pen drives.

# FAT16 VS FAT32

## 1.Capacidade de Armazenamento:

FAT16: Suporta tamanhos de partição de até 2 GB em Windows antigos e até 4 GB em sistemas mais recentes.

FAT32: Permite tamanhos de partição muito maiores, com suporte para até 2 TB em sistemas Windows antigos e até 16 TB em sistemas mais recentes.

## 2.Tamanho do Cluster:

FAT16: O tamanho do cluster em FAT16 varia dependendo do tamanho da partição, mas geralmente é maior do que em FAT32, o que pode levar a um desperdício de espaço em discos com muitos ficheiros pequenos.

FAT32: O tamanho do cluster em FAT32 é menor em comparação com o FAT16, permitindo um uso mais eficiente do espaço em disco, especialmente para discos grandes.

# FAT16 VS FAT32

## 3. Número Máximo de ficheiros e subdiretórios:

FAT16: Tem um limite no número máximo de ficheiros e subdiretórios que pode armazenar em uma única partição.

FAT32: Aumentou significativamente o número máximo de ficheiros e subdiretórios que podem ser armazenados em uma única partição em comparação com o FAT16.

## 4. Compatibilidade:

FAT16: É amplamente suportado por uma variedade de sistemas operativos e dispositivos, mas pode ter algumas limitações em relação ao tamanho do disco e número de ficheiros.

FAT32: Também é amplamente suportado e oferece maior compatibilidade com tamanhos de disco maiores e número de ficheiros.



# FAT16 VS FAT32

## 5. Fragmentação:

FAT16: Tende a fragmentar mais rapidamente em comparação com o FAT32, especialmente em partições maiores.

FAT32: Devido ao tamanho menor do cluster e à capacidade de manipular partições maiores, a fragmentação tende a ser menos significativa em comparação com o FAT16.

Em geral, o FAT32 é uma escolha mais versátil e eficiente em termos de espaço de armazenamento para dispositivos modernos, devido ao suporte a tamanhos de disco maiores e menor desperdício de espaço. No entanto, o FAT16 ainda é útil em certos contextos, especialmente para dispositivos mais antigos ou para compatibilidade com sistemas que não suportam FAT32.

# PROBLEMAS: FRAGMENTAÇÃO

- A fragmentação em sistemas de ficheiros ocorre quando os dados são armazenados de forma não contígua no disco.
- Isso acontece quando um ficheiros é dividido em partes e armazenado em diferentes partes do disco.
- Por exemplo, se um ficheiros é grande e o sistema de ficheiros não pode alocar um bloco contíguo grande o suficiente para armazená-lo, ele será dividido em fragmentos menores e armazenado em locais diferentes. Isso pode levar a tempos de acesso mais lentos, já que o disco precisa procurar em vários locais para ler o ficheiro.
- Isso pode dificultar a alocação de espaço contíguo para novos ficheiros, resultando em desperdício de espaço e **diminuindo o desempenho do sistema e o tempos de acessos ao disco.**

# TIPOS DE SISTEMAS DE FICHEIROS

## Journaling

Desenhado para prevenir corrupção de dados quer por crash do Sistema operativo, quer por falta de energia.

Sem o *journaling*, o sistema de ficheiros não sabe se o ficheiro foi completamente escrito no disco.

Em caso de erro fica em disco mas corrompido.

<https://www.ibm.com/developerworks/library/l-journaling-filesystems/index.html>

# JOURNALING

## Journaling

Funciona mantendo um registro (ou "journal") das operações que serão realizadas no sistema de ficheiros antes que elas sejam efetivamente executadas.

Essas operações são registradas em um local seguro no disco antes de serem aplicadas aos metadados do sistema de ficheiros propriamente ditos.

Deste modo se existir uma falha de energia (por exemplo) a meio da escrita, quando o sistema volta à normalidade, tenta completar as tarefas não terminadas, prevenindo deste modo perdas ou corrupção de dados.

### Procedimento:

- o sistema regista que vai escrever um determinado ficheiro no jornal;

- escreve o ficheiro para o disco

- Remove a tarefa de escrita do jornal;

# TIPOS DE SISTEMAS DE FICHEIROS

**Ext** - é o acrónimo de “Extended file system”. Criado em 1992 foi considerado um desenvolvimento muito importante na altura sobre os sistemas de ficheiros existentes. Já não é suportado.

**Ext2** - versão 2 do *ext* e foi introduzido de modo a conseguir suportar discos com capacidade até 2 TB e atributos de ficheiros.

No entanto não suportando *journaling*, tal como **exFAT**, **FAT16** e **FAT32** que também não suportam, deixou de ser utilizado.

# TIPOS DE SISTEMAS DE FICHEIROS

**Ext3** - É a evolução do **Ext2** com suporte para *Journaling*.

Foi desenvolvido com suporte para Ext2 de modo a permitir a conversão de partições entre Ext2 e Ext3 sem necessidade de formatação.

Caiu em desuso a partir do momento que Ext4 se tornou suficientemente testado e confiável.

# TIPOS DE SISTEMAS DE FICHEIROS – EXT4

**Ext4** - Também desenhado para ser compatível com as versões anteriores, sendo possível montar uma partição **Ext4** como **Ext2** ou **Ext3**.

Tem novas funcionalidades em relação às versões anteriores, reduzindo a fragmentação de ficheiros, permite a utilização de discos até 16TB.

O seu mecanismo interno de “delayed allocation”, com a utilização de memórias do tipo Flash, permite menos escritas e ajuda a prolongar o seu tempo de vida

# EXT4 VS EXT3

**Desempenho:** O ext4 foi projetado para oferecer melhor desempenho em relação ao ext3, especialmente em termos de velocidade de leitura e gravação. Ele é otimizado para operações de leitura e gravação rápidas, o que é particularmente útil em ambientes onde o desempenho é crítico, como em servidores.

**Recuperação mais rápida:** Em caso de falhas ou corrupção no sistema de ficheiros, o ext4 é capaz de se recuperar mais rapidamente do que o ext3. Ele utiliza um sistema de journaling mais eficiente, o que reduz o tempo necessário para a verificação e reparação do sistema de ficheiros após uma falha.



# TIPOS DE SISTEMAS DE FICHEIROS

**BtrFS - chamado “Butter” ou “Better FS” desenvolvido pela Oracle, que significa “B-Tree File System”**

Snapshots e Rollbacks: Suporta reversões rápidas do sistema.

Data Integrity: Utiliza a soma de verificação para detetar e corrigir a corrupção de dados.

Compression e Deduplication: Pode economizar espaço de armazenamento.

Suporte RAID: Funcionalidade RAID incorporada. Redimensionamento dinâmico: Pode redimensionar sistemas de ficheiros mais facilmente.

# TIPOS DE SISTEMAS DE FICHEIROS - XFS

## **XFS (Sistema de Ficheiros X) é um sistema de ficheiros de alto desempenho e escalonável**

Conhecido por seu desempenho consistente, mesmo em volumes grandes. Ele é otimizado para lidar com grandes ficheiros e grandes volumes de armazenamento.

Utiliza journaling para garantir a integridade dos dados em caso de falha do sistema. Isso ajuda a evitar a corrupção de dados e a minimizar o tempo de recuperação após uma falha.

Projetado para minimizar a fragmentação de ficheiros e fragmentação de espaço livre, resultando em um desempenho mais consistente ao longo do tempo.

# TIPOS DE SISTEMAS DE FICHEIROS - XFS

XFS (Sistema de Ficheiros X) é um sistema de ficheiros de alto desempenho e escalonável

Oferece suporte a cotas de disco e controle de acesso baseado em ACLs (Access Control Lists), permitindo aos administradores controlar o uso de espaço em disco e definir permissões de acesso granulares.

**NOTA:** algumas operações podem ser complexas para utilizadores menos experientes, sobretudo quando as coisa não correm bem.

# TIPOS DE SISTEMAS DE FICHEIROS - ZFS

## **ZFS pode ser considerado uma opção robusta em comparação com o Btrfs**

Integridade de Dados Avançada: ZFS é conhecido por sua ênfase na integridade de dados. Ele utiliza checksums para verificar a integridade dos dados armazenados, detetando e corrigindo erros silenciosos (bit rot) automaticamente.

Gerenciamento de Armazenamento Avançado: ZFS oferece um conjunto abrangente de recursos para gerenciamento de armazenamento, incluindo pooling de armazenamento (ZFS Pool), instantâneos (snapshots), clones, deduplicação e compressão de dados. Esses recursos são altamente flexíveis e podem ser úteis em uma variedade de cenários de uso, desde armazenamento pessoal até ambientes empresariais.

# TIPOS DE SISTEMAS DE FICHEIROS - ZFS

Suporte a RAID Integrado: ZFS oferece suporte a várias configurações de RAID (como RAID-Z) diretamente no sistema de ficheiros, o que pode simplificar a configuração e manutenção de arrays de armazenamento redundantes.

Escalabilidade: ZFS é altamente escalável e pode lidar com grandes quantidades de armazenamento de forma eficiente. Ele suporta volumes de armazenamento massivos e pode ser dimensionado horizontalmente para atender às necessidades crescentes de armazenamento.

# TIPOS DE SISTEMAS DE FICHEIROS - ZFS

Compressão e Deduplicação: ZFS oferece suporte embutido para compressão de dados, o que pode economizar espaço em disco e melhorar o desempenho de E/S. Além disso, sua capacidade de deduplicação pode reduzir ainda mais o uso de armazenamento, eliminando cópias redundantes de dados idênticos.

Documentação e Suporte: ZFS é bem documentado e tem uma comunidade de usuários ativa que fornece suporte e recursos adicionais. Além disso, muitas distribuições Linux e sistemas operacionais Unix suportam ZFS, facilitando sua integração em diferentes ambientes.

# TIPOS DE SISTEMAS DE FICHEIROS

**Swap** - Embora apareça como uma opção de tipo de Sistema de ficheiros, não o é de facto

É utilizado como memória virtual e não tem uma estrutura de Sistema. Não é possível montar ou verificar o seu conteúdo. É utilizado pelo Kernel para colocar dados que não cabem na RAM.

O Windows tem um mecanismo semelhante, guarda um ficheiro de paging na partição de Sistema. O Linux utiliza uma partição separada para Swap.