

# Sistemas Operativos

LEI - 2021/2022

**:: Sistemas de ficheiros ::**

**Escola Superior de Tecnologia de Setúbal - IPS**

# Conteúdos

- Estrutura dos dispositivos de armazenamento secundário
- Funcionamento e interfaces dos sistemas de ficheiros
- Detalhes de implementação de sistemas de ficheiros locais
- Algoritmos e trade-offs

# Porquê sistemas de ficheiros?

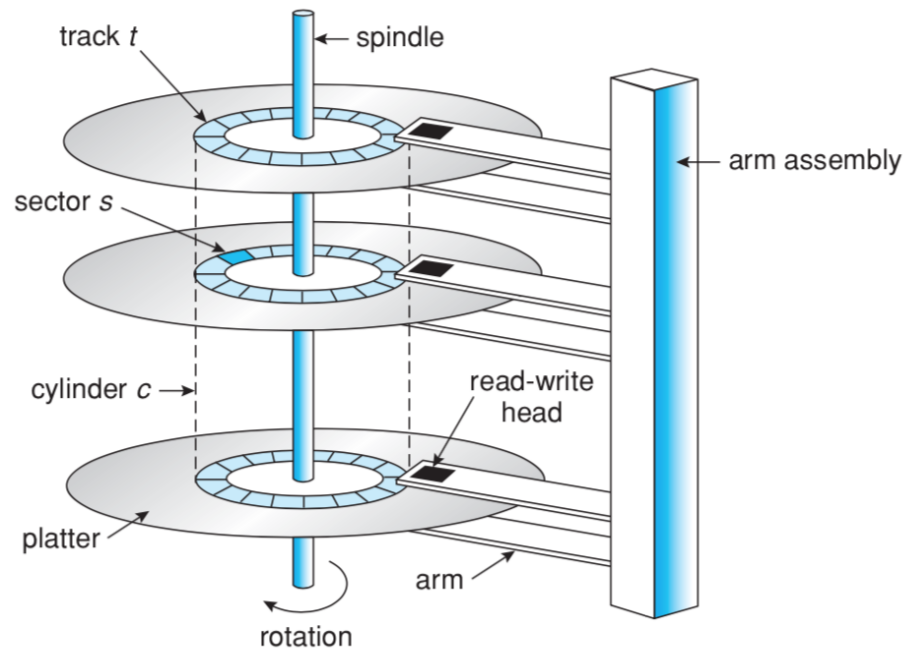
- Memória é insuficiente para guardar todos os dados e programas
- Computadores usam discos (ou outros) para guardar os dados
- Sistemas organizam informação em sistemas de ficheiros



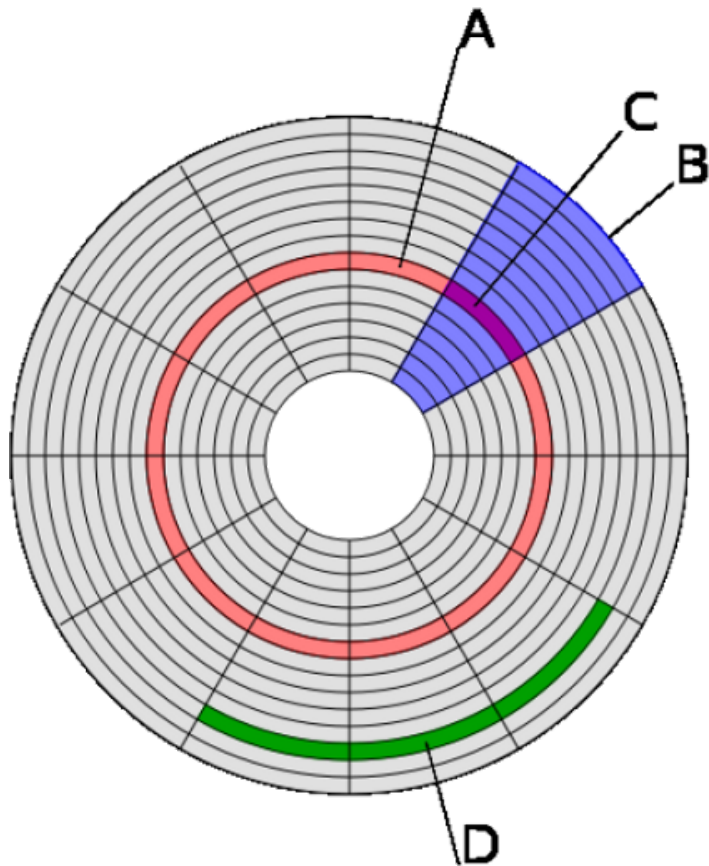
É necessário otimizar I/O para máxima concorrência

# Discos magnéticos

Armazenamento secundário mais utilizado...



# Estrutura de HDD



## Hard Drive Structure:

A = track

B = sector

C = sector of a track

D = cluster



Normalmente um sector tem 512 bytes...

# Performance de HDD

- **Seek time:** tempo que a cabeça de leitura demora a posicionar-se numa determinada pista
  - 70's: 600 ms (track-to-track)
  - Hoje: 0.2-0.6 ms (vs 0.08-0.12 mms SSDs)
- **Latência:** tempo relacionado com a rotação do disco de modo a posicionar a cabeça de leitura num sector.
  - 4200 RPM -> 7.14 ms
  - 7200 RPM -> 4.17 ms
  - 15000 RPM -> 2.00 ms

# Ficheiro

- Blocos contíguos de dados num endereço lógico
- Vários tipos:
  - texto
  - binário
  - programas
  - etc..

# Operações em ficheiros

Um ficheiro é um tipo abstracto de dados...

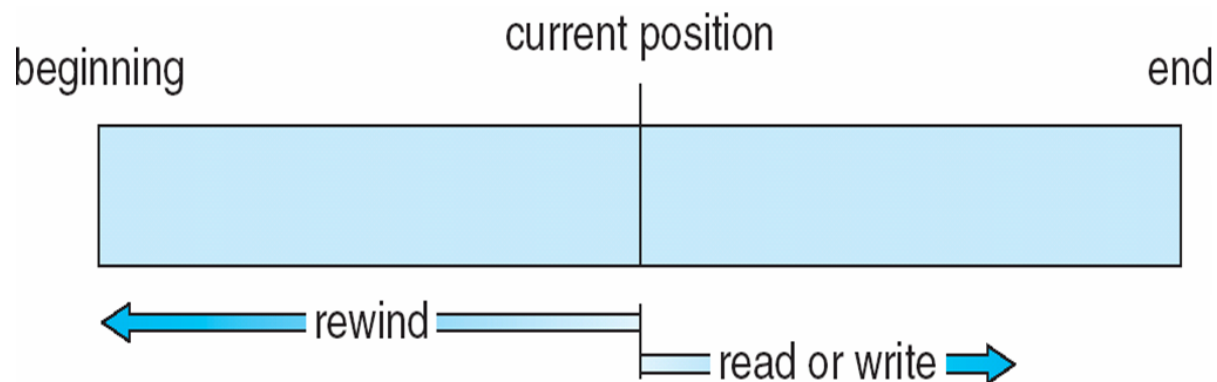
- Criação
- Escrita (num local dado por um ponteiro de escrita)
- Leitura (a partir de um local dado por um ponteiro de leitura)
- Reposicionamento no ficheiro (seek)
- Remoção
- Truncar
- Abrir / Fechar



# Acesso sequencial vs directo

## Sequencial

- Informação processada por ordem
- read\_next | write\_next | reset



# Acesso sequencial vs directo

## Directo

- Informação separada em blocos lógicos de tamanho fixo
- `read(n)` | `write(n)` | `seek(n)`

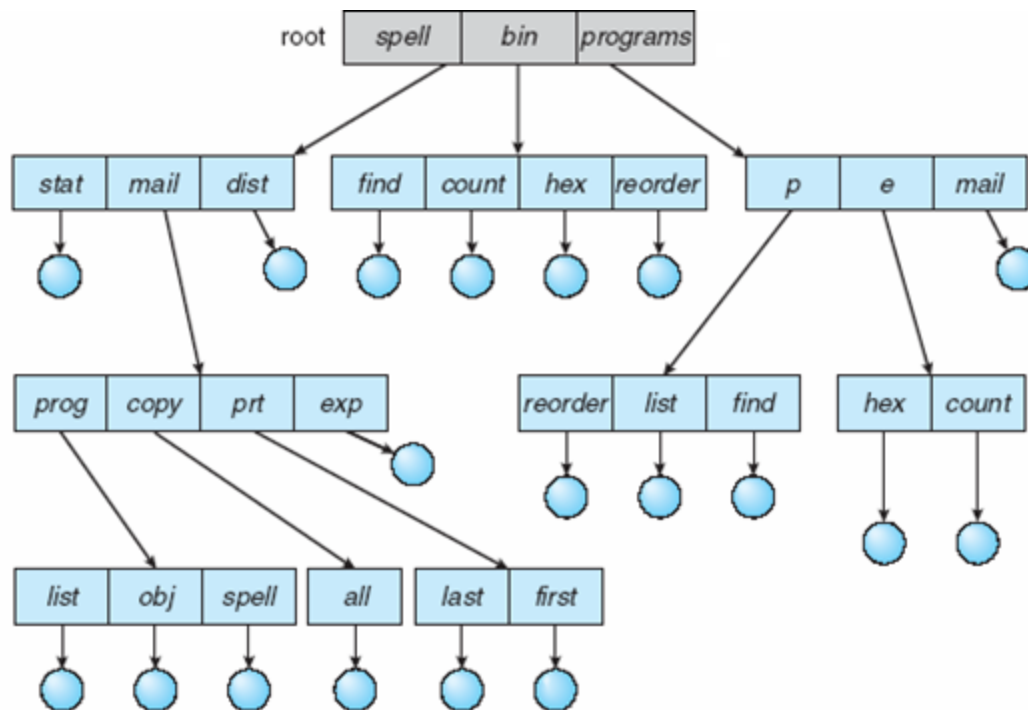
| sequential access | implementation for direct access        |
|-------------------|---|
| <i>reset</i>      | <i>cp = 0;</i>                          |
| <i>read next</i>  | <i>read cp;</i><br><i>cp = cp + 1;</i>  |
| <i>write next</i> | <i>write cp;</i><br><i>cp = cp + 1;</i> |

# C API

```
FILE *fopen( const char *filename, const char *mode );  
  
int fclose( FILE *stream );  
  
size_t fread( void *buffer, size_t size, size_t count,  
              FILE *stream );  
  
size_t fwrite( const void *buffer, size_t size, size_t count,  
              FILE *stream );  
  
int fseek( FILE *stream, long offset, int origin );
```

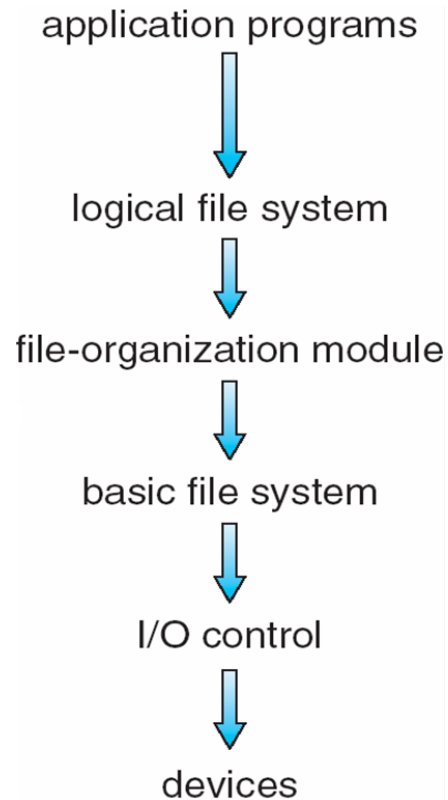
# Estruturas de directórios

Colecções de nós que contém informação sobre ficheiros contidos no directório.



# Sistema de ficheiros

Representação eficiente e lógica de dados num dispositivo..



 Estruturado em camadas...

# Camadas

## Dispositivos

- Ex: read drive 1, cylinder 72, track 2, sector 10 to memory 0x1060..

## Sistema de ficheiros básico

- Ex: read physical block 123

## Módulo de organização de ficheiros

Compreende ficheiros, endereços lógicos e blocos físicos

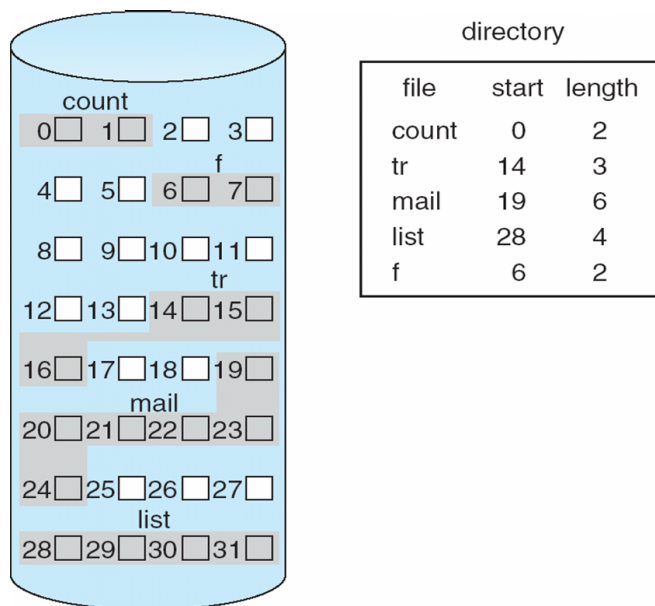
- Ex: read block #1 of file 5

## Sistema de ficheiros lógicos

- Metadados.. Traduz nomes de ficheiros para número, file handler, localização, etc.

# Alocação contígua

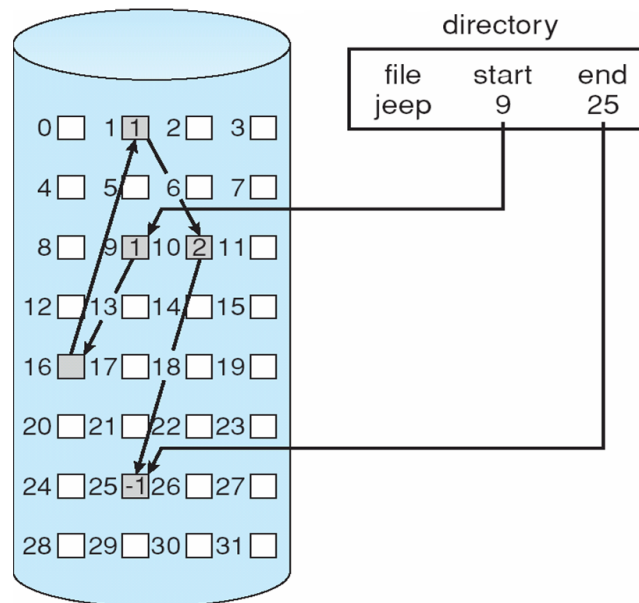
Forma como os dados de um ficheiro estão organizados no disco..



! Fragmentação..

# Alocação por listas ligadas

Resolve problemas de fragmentação da alocação contígua..

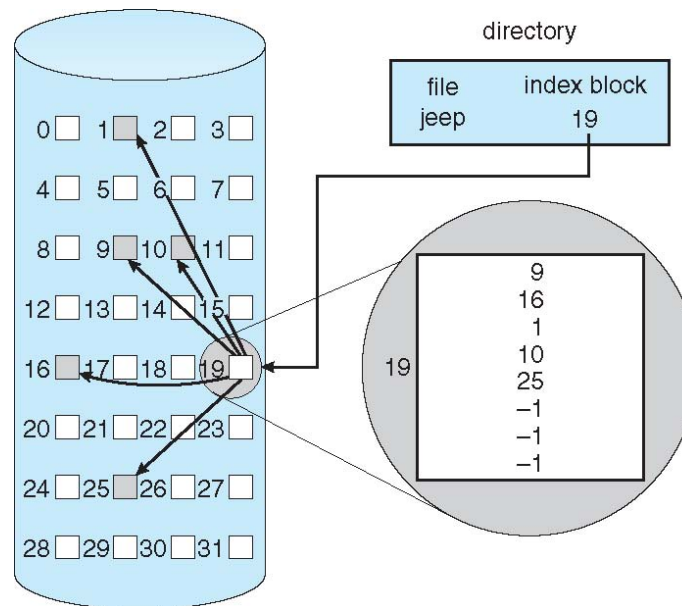


- ! Necessita-se atravessar todos os nós de modo a obter o nó  $i$ ..
- ! Espaço para os ponteiros (ex: 8 bytes em sectores de 512 bytes)



# Alocação por indexação

Resolve problemas dos ponteiros das listas ligadas..



⚠ O bloco de índices poderá ocupar mais espaço que os vários ponteiros em cada sector..

# Ex: FAT (File Allocation Table)

Sistema de ficheiros da Microsoft para MS-DOS

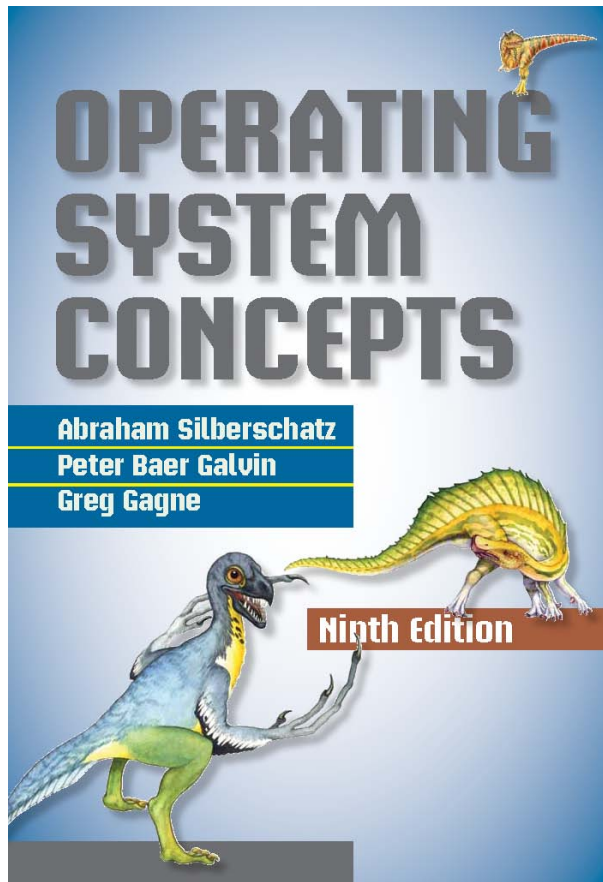
<https://www.youtube.com/watch?v=V2Gxqv3bJCK>

| Nome  | # blocos   | Tamanhos máx                     |
|-------|------------|----------------------------------|
| FAT12 | 4096       | 16MB, 32MB (clusters 4KB, 8KB)   |
| FAT16 | 65536      | 2GB, 4GB (clusters 32BK, 64KB)   |
| FAT32 | 4294967296 | 2TB (sectores de 512K) - 8/16 TB |

**Quiz...**

# Sumário

- Discos rígidos são o tipo de armazenamento secundário mais utilizado
- Ficheiros são tipos abstractos de dados, geridos pelo sistema operativo
- O sistema de ficheiros reside no armazenamento secundário
- Discos podem ser segmentados e os ficheiros pode ser alocados através de várias técnicas (contígua, listas ligadas, índices)
- Dados no disco são referenciados através de endereços lógicos (blocos) no sistema de ficheiros
- Performance e confiabilidade são extremamente relevantes



Ler capítulos 10, 11 e 12...