

# Sistemas Operativos

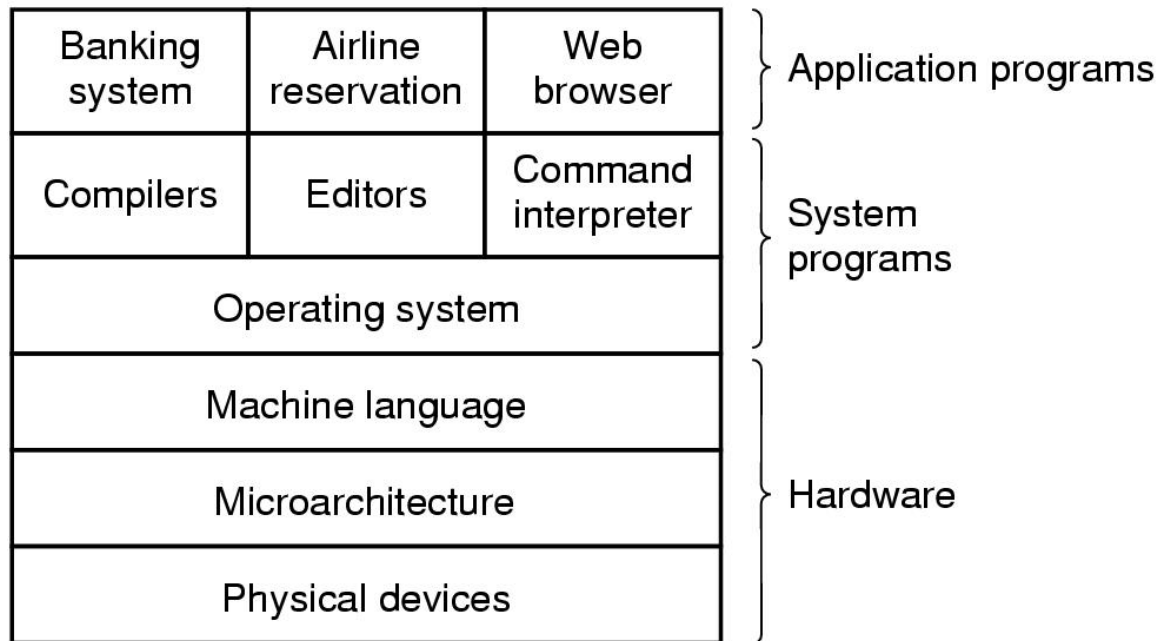
Licenciatura em Engenharia Informática  
Licenciatura em Engenharia Computacional  
Licenciatura em Física (opcional)

Ano letivo 2025/2026

Pedro Azevedo Fernandes (paf@ua.pt)

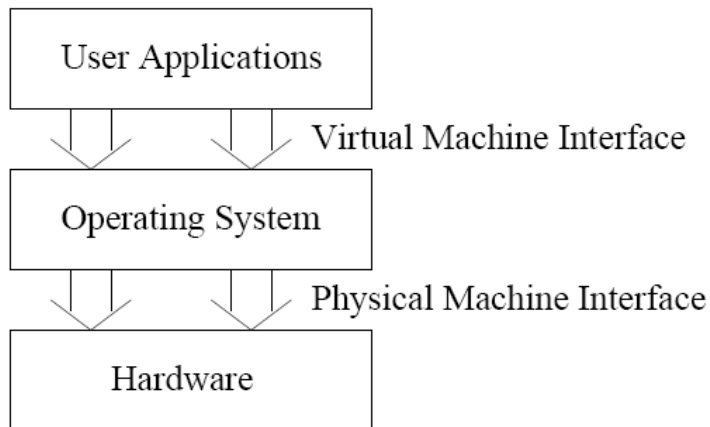
# O que é um Sistema Operativo?

- O Sistema Operativo é o programa base que estabelece a interface entre os programas de aplicação e o hardware.



# Objectivos do Sistema Operativo

- Executar os programas de aplicação
- Tornar o hardware mais fácil de usar
  - O SO cria um nível de abstracção que esconde muitos dos pormenores da utilização de dispositivos específicos (usando *device drivers*)
- Usar o hardware de forma eficiente
  - O SO gere os recursos de hardware do sistema de forma a tornar a sua utilização mais eficiente, justa e segura

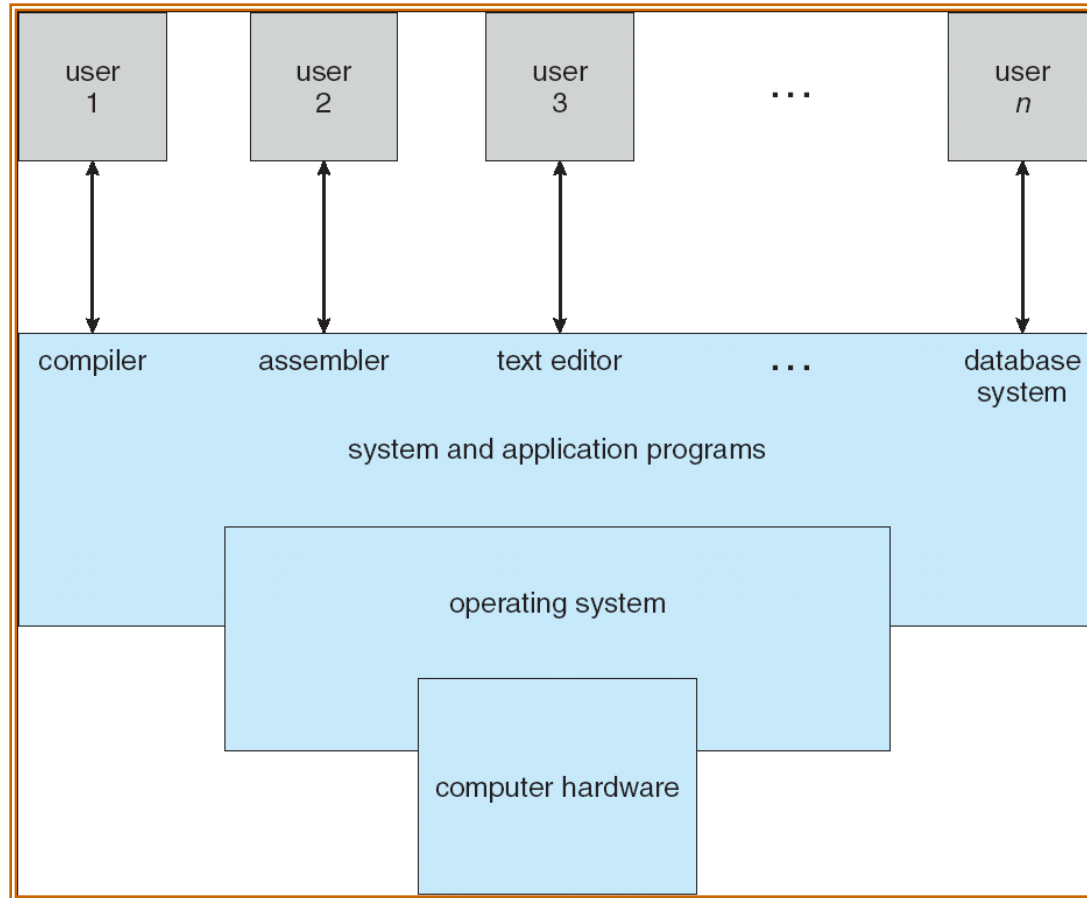


Os 2 últimos objectivos podem facilmente entrar em conflito

Um sistema computacional pode ser dividido em 4 componentes:

- Hardware
  - CPU, Memória, Dispositivos I/O
- Sistema Operativo
  - Controla e coordena o uso de hardware entre as várias aplicações e utilizadores
- Programas de aplicação
  - Processadores de texto, compiladores, browsers, bases de dados, jogos, etc
- Utilizadores
  - Pessoas, máquinas, outros computadores

# Sistema Computacional



- **Serviços:** O SO cria serviços standard que são implementados pelo hardware
  - Exemplos: Sistema de ficheiros, memória virtual, redes, etc
  - Sistema operativo como criador de máquina virtual
- **Coordenação:** O SO coordena várias aplicações e utilizadores de modo a garantir segurança, eficiência e justiça na utilização dos recursos
  - Exemplos: concorrência, protecção da memória, segurança
  - Sistema operativo como gestor de recursos
- **Controlo:** O SO controla a execução dos programas prevenindo erros e uso impróprio do computador
  - Exemplos: escalonamento do CPU, criação de novos processos, seg fault, etc
- **Objectivo:** Criar um SO que é simultaneamente fácil de usar e eficiente

- **Árbitro**
  - Gere recursos partilhados: CPU, memória, discos, impressoras, etc.
- **Ilusionista**
  - Fornece às aplicações/programador abstrações de recursos com capacidades superiores às existentes: memória infinita; uso exclusivo do CPU; etc.
- **Adaptador**
  - Serviços comuns: sistema de ficheiros; rotinas da UI
  - Separa as aplicações dos dispositivos de entrada/saída

- Estabelecimento do ambiente de base de interação com o utilizador
- Mecanismos de execução controlada de programas
- Mecanismos de comunicação entre programas e respetiva sincronização
- Disponibilização de facilidades para o desenvolvimento, teste e depuração de programas
- Espaço de endereçamento virtual dos programas é independente das limitações da memória física
- Sistemas de ficheiros
- Modelo geral de acesso a dispositivos de I/O
- Detecção de situações de erro



- Concorrência
  - Permite que vários programas sejam executados em simultâneo
  - Também pode permitir vários utilizadores em simultâneo
- Dispositivos de I/O
  - CPU continua a trabalhar enquanto I/O não responde
  - Mecanismos comuns para acesso a vários tipos de dispositivos
- Gestão da memória
  - SO gere as alocações de memória e transferências de dados entre memória e disco
- Ficheiros
  - Espaço em disco é organizado num sistema de ficheiros capaz de armazenar vários ficheiros de tamanho variável
- Sistemas distribuídos e redes
  - Permite que um grupo de computadores trabalhem de forma conjunta para resolver um problema

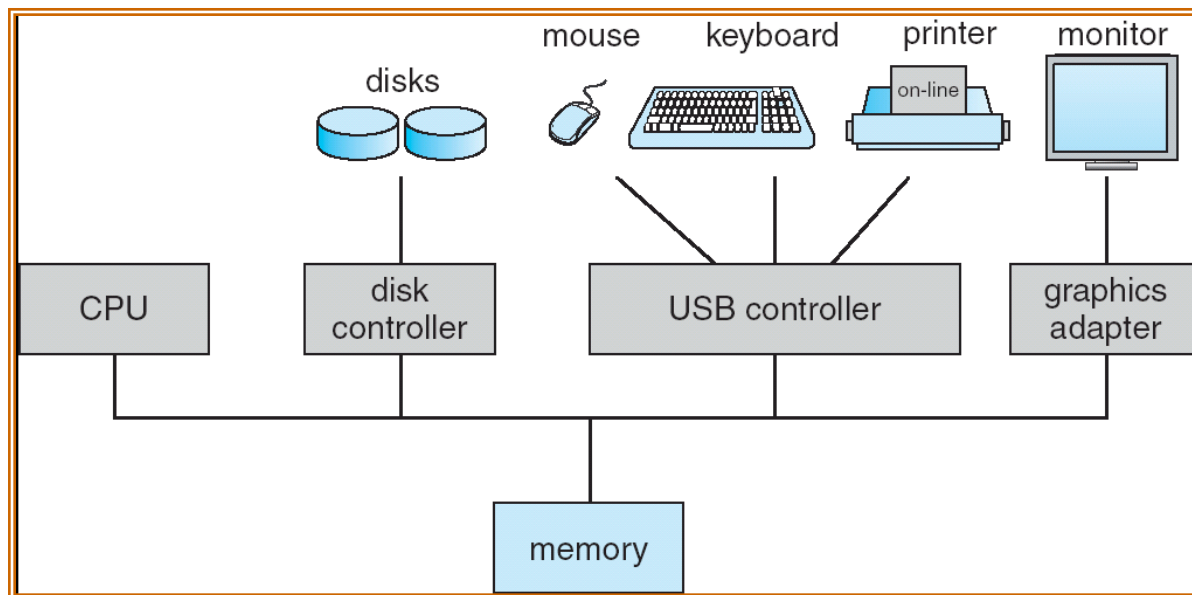
# Tipos de Sistemas Operativos

- Sistemas Operativos para Mainframes
  - Serviços: Batch, Transações e Timesharing
  - Ex: OS/390
- Sistemas Operativos para Servidores
  - Partilha de recursos de hardware e software
  - Ex: Solaris, FreeBSD, Linux, Windows Server 201x
- Sistemas Operativos para Multiprocessadores
  - Ex: Windows, Linux
- Sistemas Operativos para PCs
  - Ex: Windows, Linux
- Sistemas Operativos para Dispositivos Móveis
  - Ex: iOS, Android
- Sistemas Operativos para Sistemas Embebidos
  - Ex: QNX, VxWorks
- Sistemas Operativos para Nós Sensoriais
  - Ex: TinyOS
- Sistemas Operativos de Tempo Real
- Sistemas Operativos para *Smart Cards*

- Programa de *bootstrap* é carregado quando o computador arranca ou é reinicializado
  - Tipicamente armazenado em ROM ou EPROM (*firmware* - *BIOS*)
  - Inicializa vários dispositivos do sistema (teclado, etc.)
  - Carrega o núcleo (*kernel*) do sistema operativo, ou um *bootloader*, e começa a sua execução

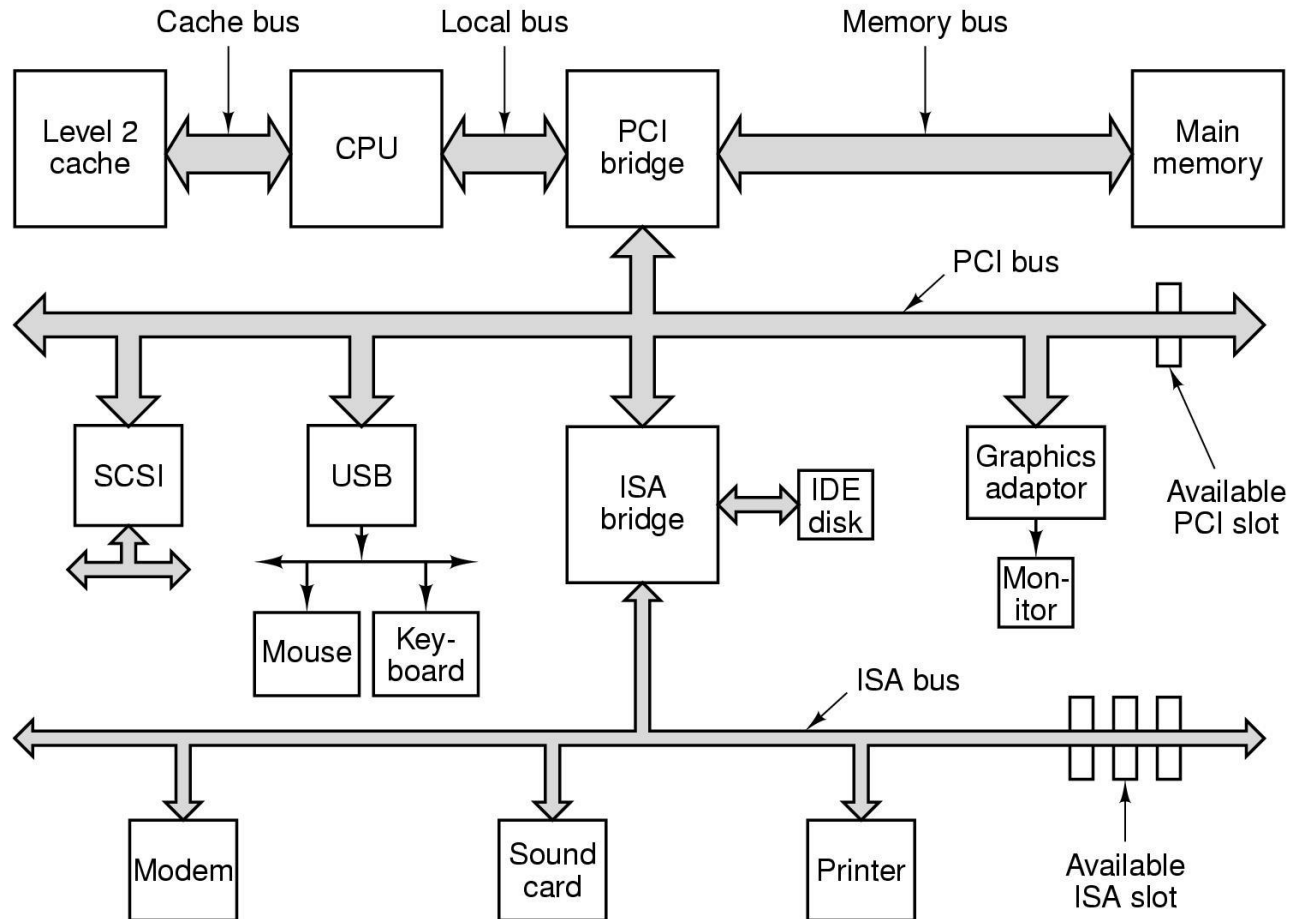
# Organização do computador

- Um ou mais CPUs e controladores de dispositivos ligados à memória através de barramento
- Execução concorrente de CPU e dispositivos origina conflitos no acesso à memória



- CPUs e controladores de dispositivos de I/O executam em paralelo
- Cada controlador de dispositivo trata um tipo particular
- Controladores de dispositivo têm *buffer* local
- CPU move dados de/para memória e de/para *buffers* locais
- Transferências de I/O são do dispositivo para o *buffer* local do respectivo controlador e depois para a memória
- Controlador do dispositivo informa CPU que terminou a operação através do envio de uma interrupção

# Organização do computador



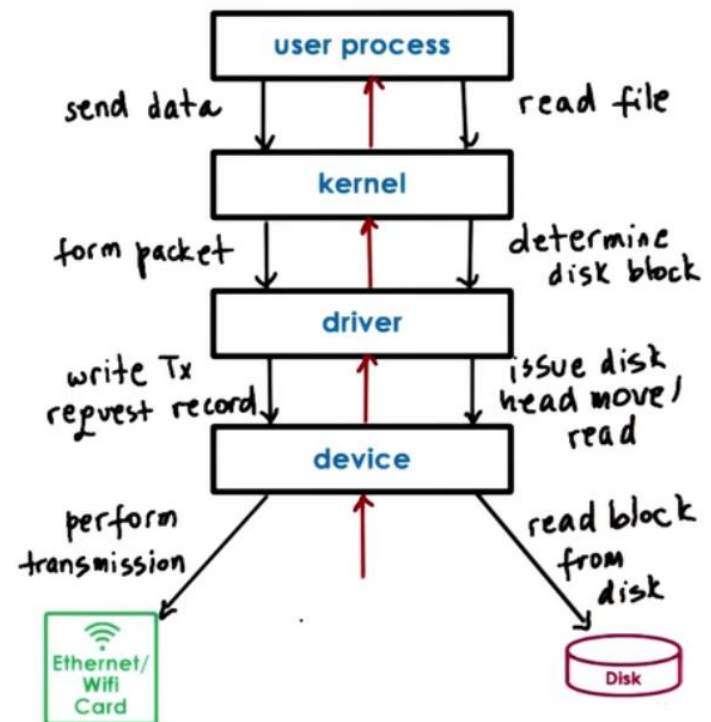
- Um processo é um programa em execução
  - Programa é uma entidade passiva, enquanto que o processo é activo
- Processo necessita de recursos
  - CPU, memória, I/O, ficheiros, etc.
  - Dados de inicialização
- Finalização de um processo deve libertar os recursos que forem reutilizáveis
- Processo com 1 *thread* tem um *Program Counter* (PC) que indica qual a próxima instrução a executar
  - Processo executa as instruções sequencialmente até terminar
- Processo *Multi-threaded* tem 1 PC por *thread*
- Tipicamente o sistema tem muitos processos, alguns do utilizador outros do SO que correm em concorrência em 1 ou mais CPUs
- Concorrência obtêm-se através da multiplexagem no tempo dos CPUs entre os vários processos/*threads*

- Memória física é um recurso escasso (e caro)
- Todos os dados em memória antes e depois do processamento
- Todas as instruções em memória para poderem ser executadas
- Gestão da memória determina o que deve estar em memória em cada altura
  - Tentando maximizar a utilização do CPU
  - Disponibiliza uma utilização da memória mais transparente
  - Garante a segurança na utilização da memória pelos vários processos
  - Permite a partilha de memória entre processos
- Actividades de gestão de memória
  - Conhecer quais as zonas de memória livres e ocupadas
  - Decidir que processos e dados mover para ou para fora da memória
  - Alocar e desalocar espaço de memória



- SO disponibiliza uma vista lógica uniforme do espaço de armazenamento
  - Abstração das propriedades físicas da unidade de armazenamento – **ficheiro**
  - Controladores distintos para tipos de unidades diferentes (disco, tape, etc)
    - Propriedades muito diferentes (velocidade, capacidade, etc)
- Sistemas de Ficheiros
  - Ficheiros organizados em diretorias/pastas
  - Permissões garantem acessos com segurança aos dados
  - Atividades do SO
    - Criar e apagar ficheiros e diretorias
    - Gerir diferentes Sistemas de Ficheiros de forma integrada
    - Primitivas de manipulação de ficheiros e diretorias
    - Mapeamento dos ficheiros na unidade de armazenamento
    - *Backups*

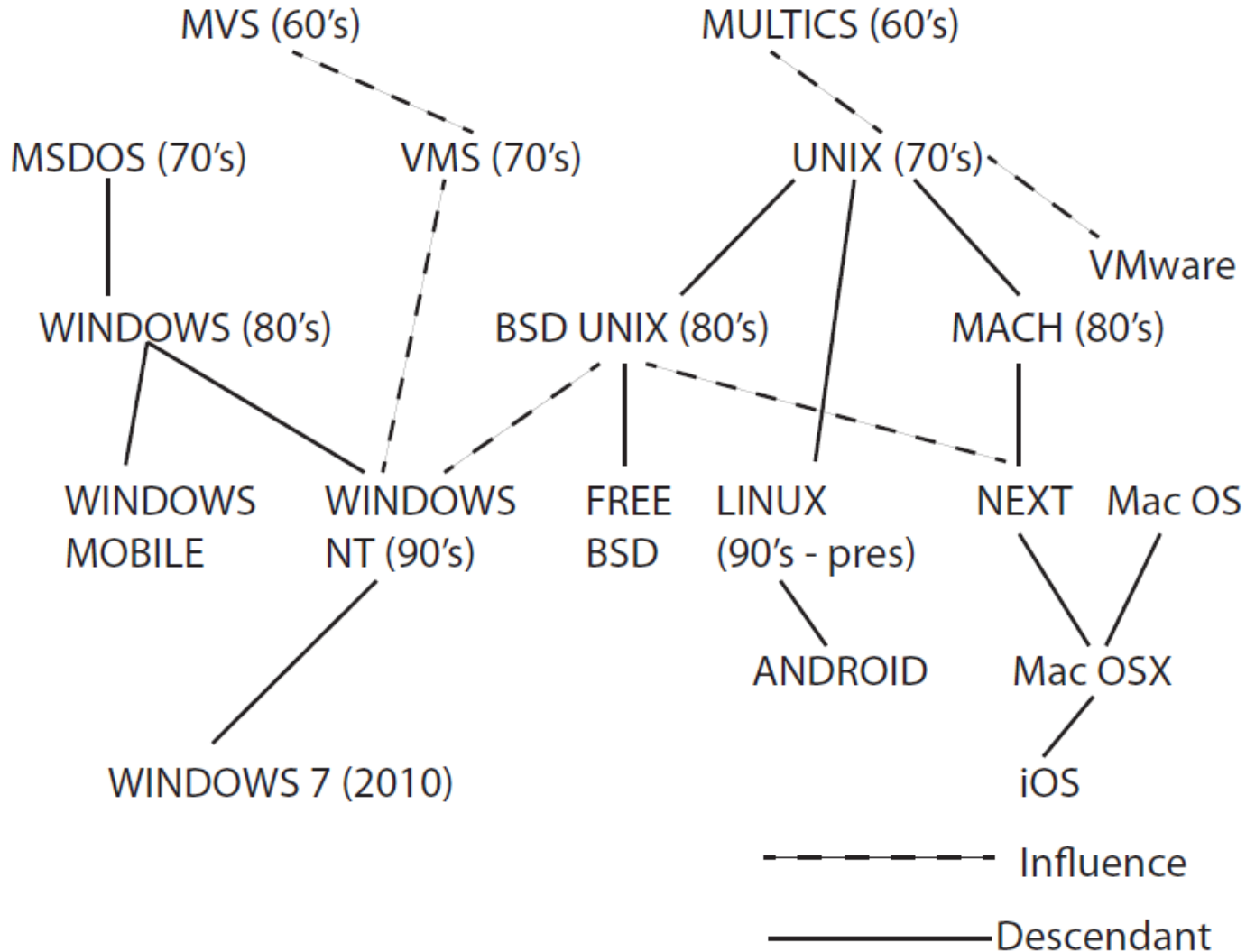
- Um dos objetivos do SO é esconder os pormenores dos dispositivos de hardware do utilizador
- SO é responsável por
  - Gestão da memória de I/O
  - Interface geral dos *device drivers*
  - *Drivers* para dispositivos específicos de hardware



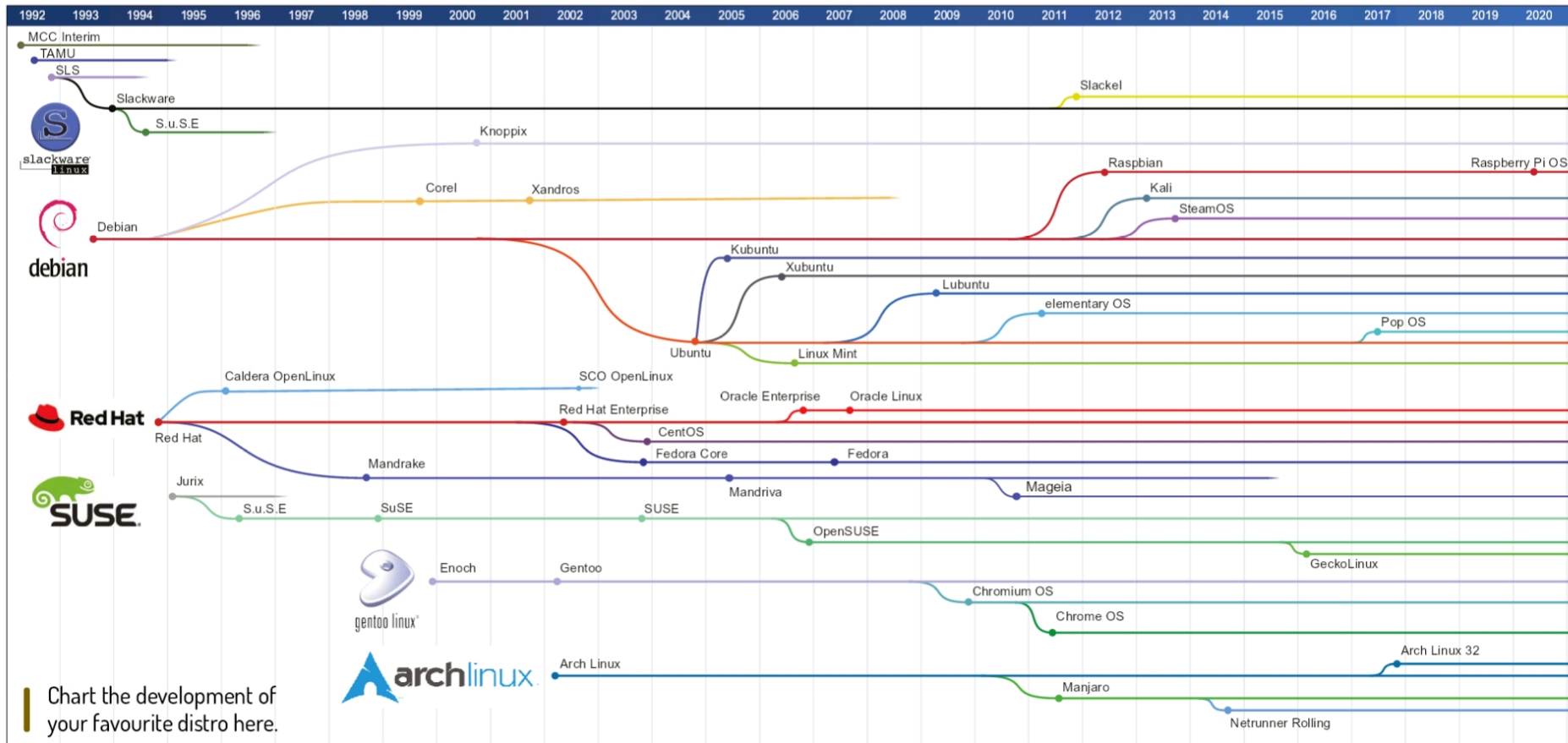
- Proteção – mecanismo do SO para controlar o acesso de processos e utilizadores aos recursos
- Segurança – Defesa do sistema contra ataques internos e externos
  - Vasta gama: *denial-of-service*, vírus, roubo de identidade, etc.
- Sistemas permitem distinguir os utilizadores, definindo assim permissões diferentes
  - Identificação do utilizador (uid, gid, etc)
  - Associação da identificação a ficheiros e processos que o utilizador controla
  - Identificadores de grupo permitem a definição de políticas mais gerais
  - Utilizadores podem alterar o ID de forma controlada
    - Comando `su`
    - Comando `chown`
    - `SUID` bit permission

- Ambiente gráfico com o utilizador
- Multiutilizador e Multitarefa
- Memória virtual
- Sistemas de operação de rede
  - Acesso indistinto a ficheiros/dispositivos locais ou de rede
  - Aplicações de login remoto, correio electrónico, navegação internet, etc
- Enorme gama de *device drivers*
- Ligação dinâmica de dispositivos (*plug and play*)

# Relações entre SOs

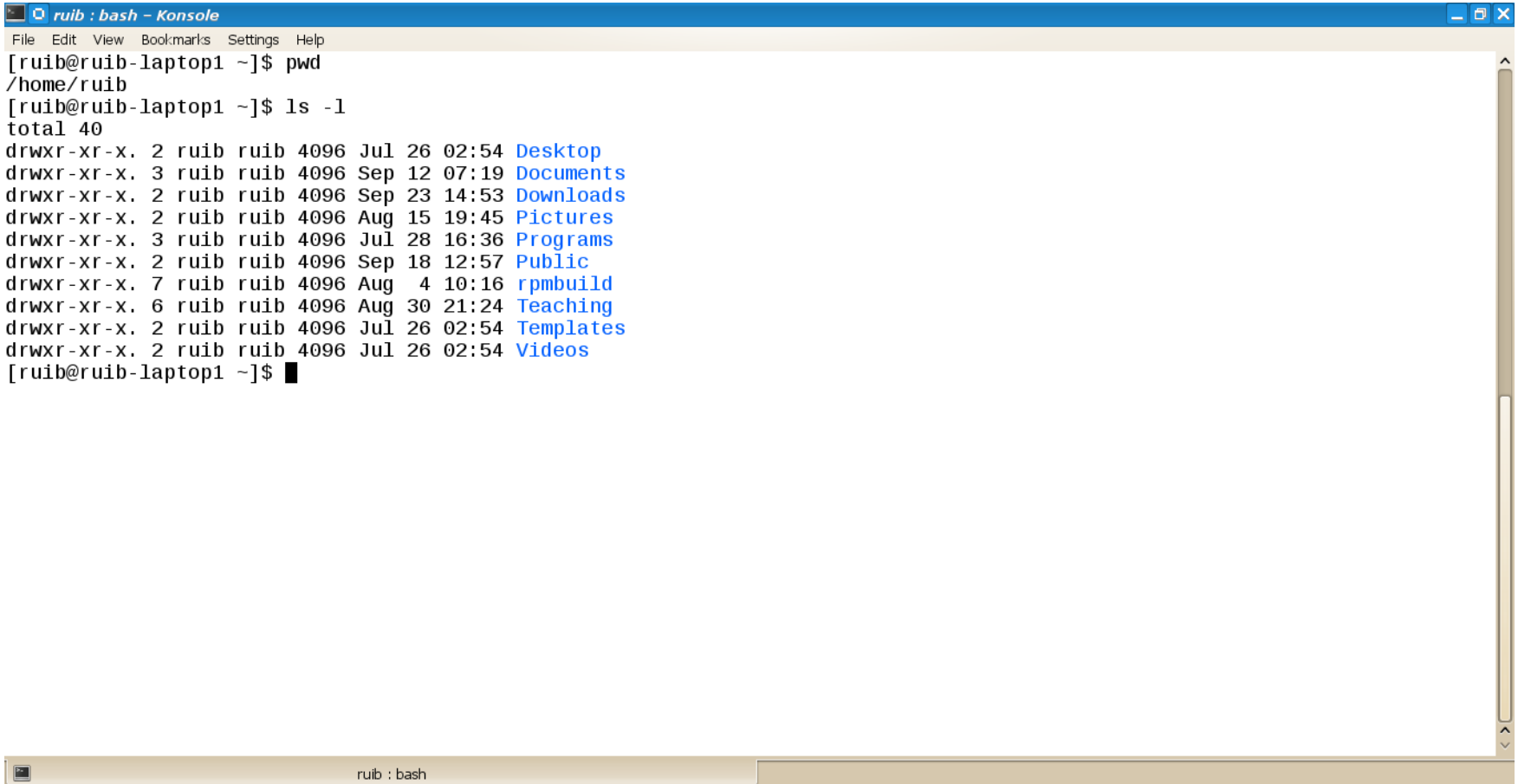


# Linux Chart



**CREDIT:** Based on the LinuxTimeLine, by fabiololix, GNU Free Documentation License v1.3, <https://github.com/FabioLolix/LinuxTimeline/tree/master>

- Interpretador de comandos (CLI)

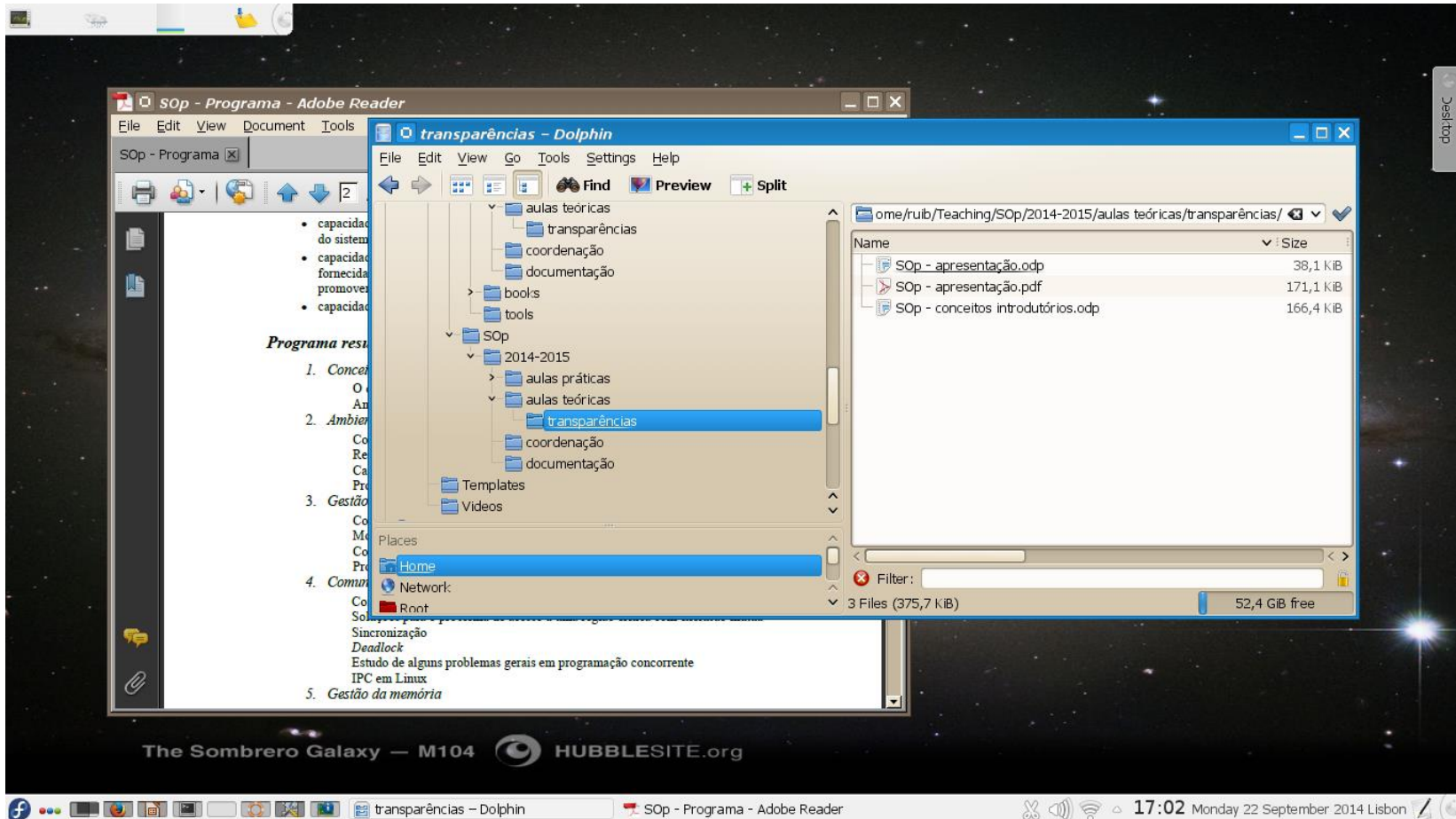


```
ruib : bash - Konsole
File Edit View Bookmarks Settings Help
[ruib@ruib-laptop1 ~]$ pwd
/home/ruib
[ruib@ruib-laptop1 ~]$ ls -l
total 40
drwxr-xr-x. 2 ruib ruib 4096 Jul 26 02:54 Desktop
drwxr-xr-x. 3 ruib ruib 4096 Sep 12 07:19 Documents
drwxr-xr-x. 2 ruib ruib 4096 Sep 23 14:53 Downloads
drwxr-xr-x. 2 ruib ruib 4096 Aug 15 19:45 Pictures
drwxr-xr-x. 3 ruib ruib 4096 Jul 28 16:36 Programs
drwxr-xr-x. 2 ruib ruib 4096 Sep 18 12:57 Public
drwxr-xr-x. 7 ruib ruib 4096 Aug  4 10:16 rpmbuild
drwxr-xr-x. 6 ruib ruib 4096 Aug 30 21:24 Teaching
drwxr-xr-x. 2 ruib ruib 4096 Jul 26 02:54 Templates
drwxr-xr-x. 2 ruib ruib 4096 Jul 26 02:54 Videos
[ruib@ruib-laptop1 ~]$
```

- Interpretador de comandos (CLI)
  - Pode estar incluído no *kernel* ou funcionar como um programa de sistema
  - Se programa independente designado de *shell*
    - *Bourne shell*, *C shell*, *Korn shell*, etc
  - Função principal: ler e executar comandos do utilizador
  - Muitos destes comandos estão relacionados com a gestão de ficheiros
  - O código que efetivamente manipula os ficheiros pode estar integrado no interpretador de comandos ou usar programas independentes deste
    - Comandos internos e externos
      - Comando `type`
    - No caso de serem programas independentes a *shell* não necessita de perceber quais os efeitos do comando executado



- Interface gráfica (GUI)

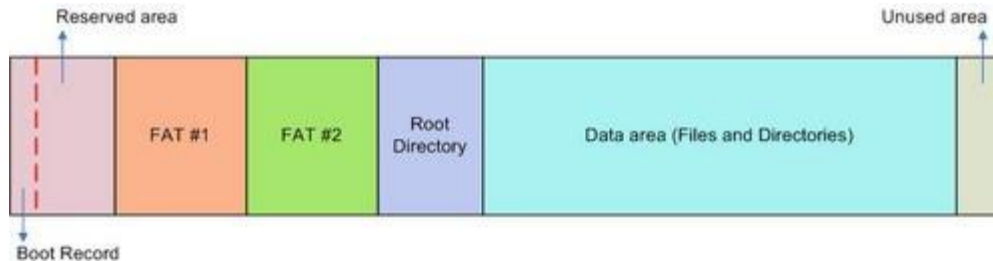


- Interface gráfica (GUI)
  - Sistema baseado em janelas e menus e preparado para ser manipulado através do rato
  - Usando o rato o utilizador pode seleccionar ficheiros, executar programas e abrir vários tipos de menus
  - Metáfora do “ambiente de trabalho”
  - A primeira interface gráfica apareceu em 1973, tendo sido desenvolvida no laboratório Xerox PARC (Silicon Valley)
  - Vários sistemas disponibilizam GUI e CLI
    - Windows é GUI, mas tem CLI como *shell* de comandos
    - Apple Mac OS X tem Aqua como GUI e várias *shells* disponíveis
    - Sistemas UNIX são em geral baseados em CLI, mas com várias GUI disponíveis (KDE, Gnome, etc)

- Sistema de Ficheiros FAT32
  - Organização do disco
    - Boot sector (boot loader, partition table, BIOS parameter block, etc.)
    - 2 cópias da FAT
    - Zona de dados (diretoria raiz no início da zona de dados)
  - Como encontrar conteúdo de um ficheiro?
    - Cluster inicial indicado na entrada de diretoria
    - Clusters seguintes **numa lista ligada** através da FAT
  - FAT
    - Array de números de clusters (cada um com 32 bits)
    - Para cada cluster indica:
      - Cluster seguinte; Final ou Livre
  - Diretorias
    - Constituídas por entradas de diretoria de tamanho fixo
    - Entradas definem nome e metadados de ficheiros e diretorias
    - Entradas definem cluster inicial

- Sistema de Ficheiros FAT32

## Disco



## FAT

XXXXXXXX	XXXXXXXX	00000009	00000004
00000005	00000007	00000000	00000008
FFFFFFFF	0000000A	0000000B	00000011
0000000D	0000000E	FFFFFFFF	00000010
00000012	FFFFFFFF	00000013	00000014
00000015	00000016	FFFFFFFF	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

**Root Directory:**  
2, 9, A, B, 11

**File #1:**  
3, 4, 5, 7, 8

**File #2:**  
C, D, E

**File #3:**  
F, 10, 12, 13, 14, 15, 16

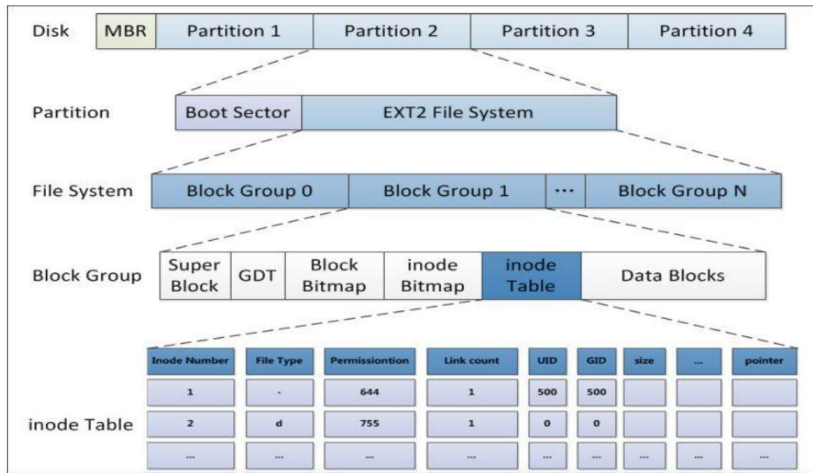
## Entrada de diretoria

### Root Directory SFN Entry Data Structure

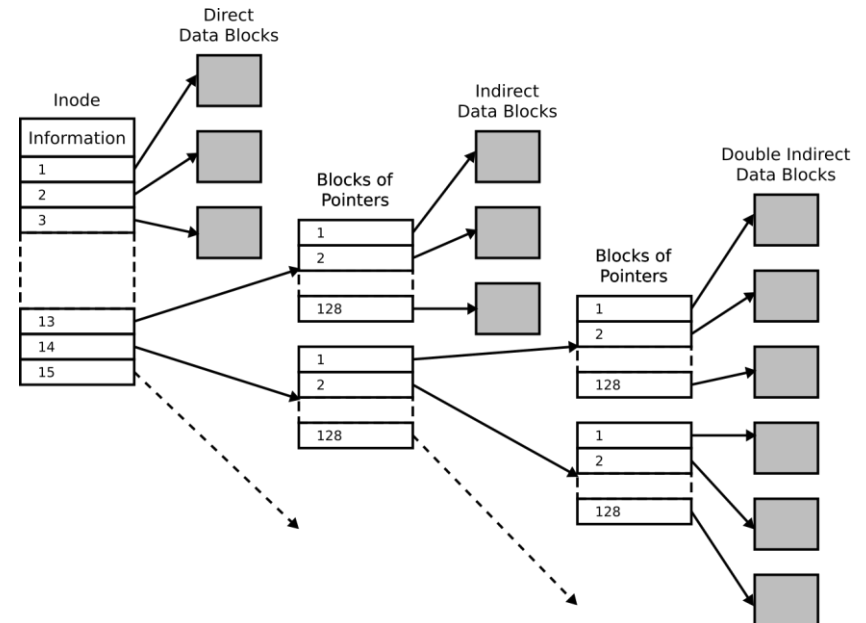
Bytes	Purpose
0	First character of file name (ASCII) or allocation status (0x00=unallocated, 0xe5=deleted)
1-10	Characters 2-11 of the file name (ASCII); the "." is implied between bytes 7 and 8
11	File attributes (see File Attributes table)
12	Reserved
13	File creation time (in tenths of seconds)*
14-15	Creation time (hours, minutes, seconds)*
16-17	Creation date*
18-19	Access date*
20-21	High-order 2 bytes of address of first cluster (0 for FAT12/16)*
22-23	Modified time (hours, minutes, seconds)
24-25	Modified date
26-27	Low-order 2 bytes of address of first cluster
28-31	File size (0 for directories)

# Sistema de ficheiros ext2

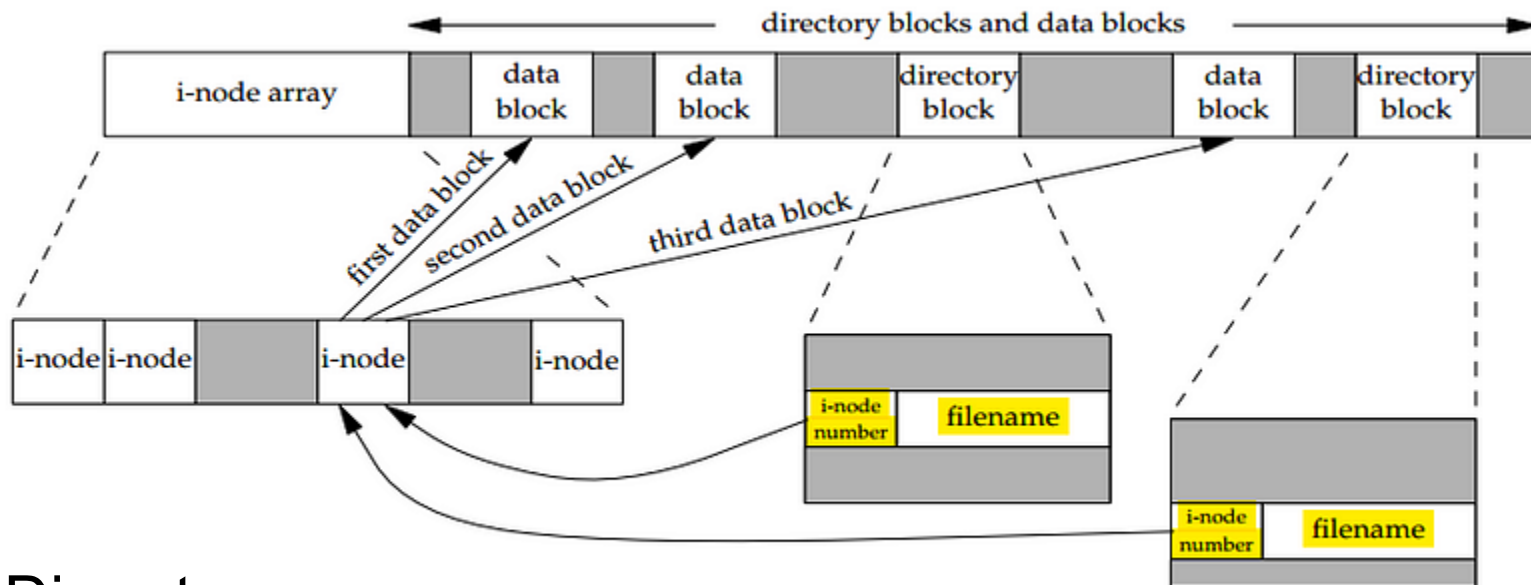
## Disco



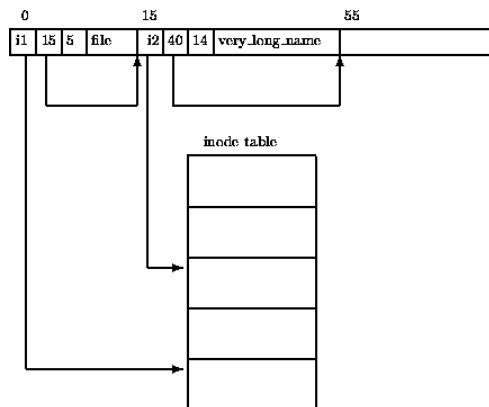
## inode



# Sistema de ficheiros ext2



## Dir entry



# Alguns Sistemas de Ficheiros

File System	Operating System
NTFS	Windows NT, Windows 2000, Windows XP, Windows 2003 Server, Windows 2008, Windows Vista, Windows 7
NTFS5	Windows 2000, Windows XP, Windows 2003 Server, Windows 2008, Windows Vista, Windows 7, 8, 10, 11
exFAT	Windows CE 6.0, Windows Vista SP1, Windows 7, WinXP+KB955704
FAT32	DOS 7.0 and higher, Windows 98, Windows ME, Windows 2000, Windows XP, Windows Server 2003, 2008, 2012, 2016, 2019, 2022, Windows Vista, Windows 7, 8, 10, 11
FAT16	DOS, All versions of Microsoft Windows
FAT12	DOS, All versions of Microsoft Windows

[http://www.ntfs.com/ntfs\\_vs\\_fat.htm](http://www.ntfs.com/ntfs_vs_fat.htm) (com modificações)

# Limitações de Sistemas de Ficheiros

	NTFS5	NTFS	exFAT	FAT32	FAT16	FAT12
Max Volume Size	$2^{64}$ clusters – 1 cluster	$2^{32}$ clusters – 1 cluster	128PB	32GB for all OS, 2TB for some OS	2GB for all OS, 4GB for some OS	16MB
Max Files on Volume	4,294,967,295 ( $2^{32} - 1$ )	4,294,967,295 ( $2^{32} - 1$ )	Nearly Unlimited	4194304	65536	
Max File Size	$2^{64}$ bytes (16 ExaBytes) minus 1KB	$2^{44}$ bytes (16 TeraBytes) minus 64KB	16EB	4GB minus 2 Bytes	2GB (Limit Only by Volume Size)	16MB (Limit Only by Volume Size)
Max Clusters Number	$2^{64}$ clusters – 1 cluster	$2^{32}$ clusters – 1 cluster	4294967295	4177918	65520	4080
Max File Name Length	Up to 255	Up to 255	Up to 255	Up to 255	Standard - 8.3 Extended - up to 255	Up to 254

[http://www.ntfs.com/ntfs\\_vs\\_fat.htm](http://www.ntfs.com/ntfs_vs_fat.htm) (com modificações)



# Características de Sistemas de Ficheiros

	NTFS5	NTFS	exFAT	FAT32	FAT16	FAT12
Unicode File Names	Unicode Character Set	Unicode Character Set	Unicode Character Set	System Character Set	System Character Set	System Character Set
System Records Mirror	MFT Mirror File	MFT Mirror File	No	Second Copy of FAT	Second Copy of FAT	Second Copy of FAT
Boot Sector Location	First and Last Sectors	First and Last Sectors	Sectors 0 to 11 Copy in 12 to 23	First Sector and Copy in Sector #6	First Sector	First Sector
File Attributes	Standard and Custom	Standard and Custom	Standard Set	Standard Set	Standard Set	Standard Set
Alternate Streams	Yes	Yes	No	No	No	No
Compression	Yes	Yes	No	No	No	No
Encryption	Yes	No	No	No	No	No
Object Permissions	Yes	Yes	Yes	No	No	No
Disk Quotas	Yes	No	No	No	No	No
Sparse Files	Yes	No	No	No	No	No
Reparse Points	Yes	No	No	No	No	No
Volume Mount Points	Yes	No	No	No	No	No

# Desempenho de Sistemas de Ficheiros

	NTFS5	NTFS	exFAT	FAT32	FAT16	FAT12
Built-In Security	Yes	Yes	Yes minimal ACL only	No	No	No
Recoverability	Yes	Yes	Yes if TFAT activated	No	No	No
Performance	Low on small volumes High on Large	Low on small volumes High on Large	High	High on small volumes Low on large	Highest on small volumes Low on large	High
Disk Space Economy	Max	Max	Max	Average	Minimal on large volumes	Max
Fault Tolerance	Max	Max	Yes if TFAT activated	Minimal	Average	Average

# Limitações de Sistemas de Ficheiros

Features	Ext2	Ext3	Ext4
Individual file size	16GB-2TB	16GB-2TB	16GB-16TB
Volume file system size	4TB-32TB	4TB-32TB	4TB-1EB
Default inode size	128 bytes	128 bytes	256 bytes
Time Stamp	No support	Second	Nanosecond
Defragmentation	No	No	Yes
Directory Indexing	Disabled	Disabled	Enabled
Multiple Block Allocation	Basic	Basic	Advanced
Preallocation	No	In-core reservation	For extent file
Delayed Allocation	No	No	Yes

- Visualizar aplicações em execução
  - Windows
    - Task Manager/Gestor de Tarefas
  - Linux
    - Comandos: `ps`, `top`, `htop`