Introdução

Objetivos de um SO

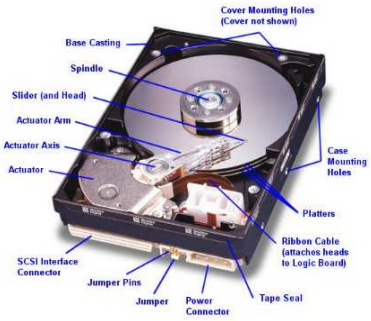
- Transformar o hardware numa máquina simples de usar
- Obter o máximo rendimento do Hardware
  - os computadores são dispositivos dispendiosos
  - cedo se viu que poderiam fazer várias coisas em simultâneo

A vida sem um Sistema Operativo

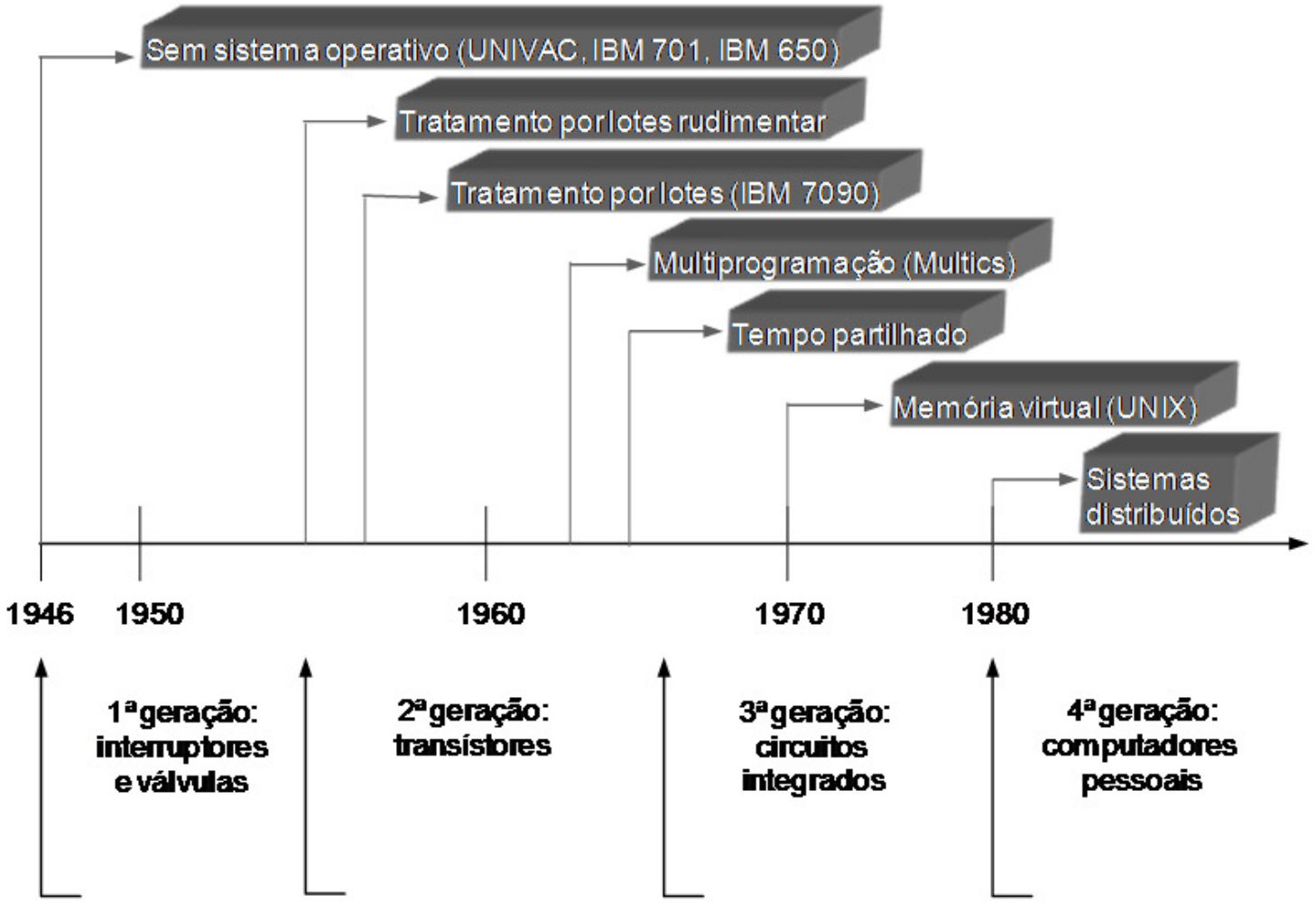
- Onde está o ficheiro? qual o disco, pista, setor? qts blocos?
- Se o hardware entretanto mudar o código tem de mudar…

A vida com um Sistema Operativo

```
file = open ("test.txt", O_WRONLY);
write (file, "test", 1);
close (file);
```

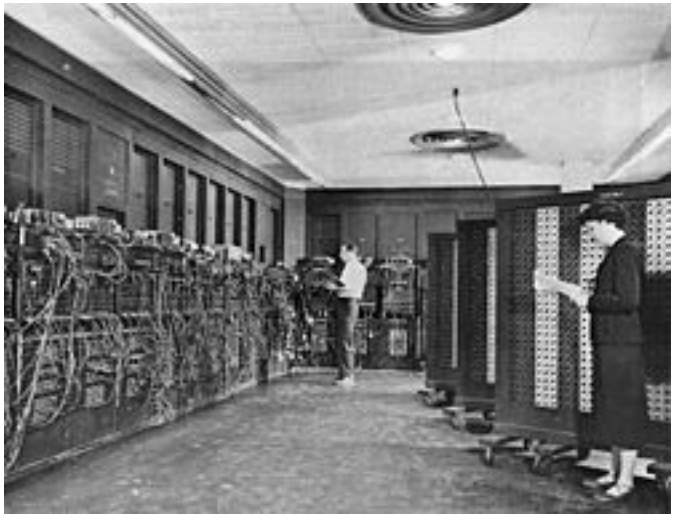


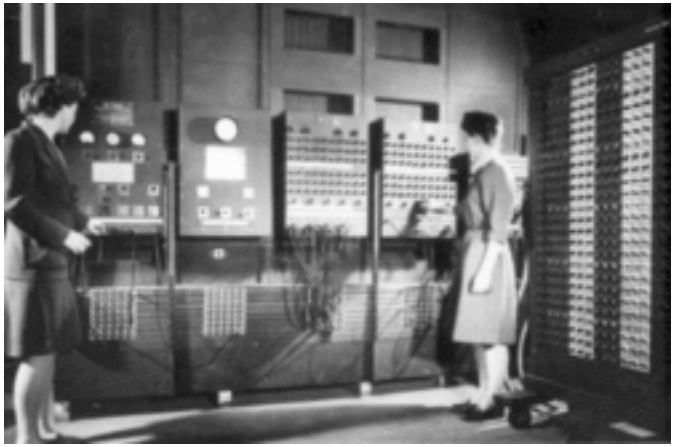
Evolução Histórica



História: 1ª Geração (1945-55)

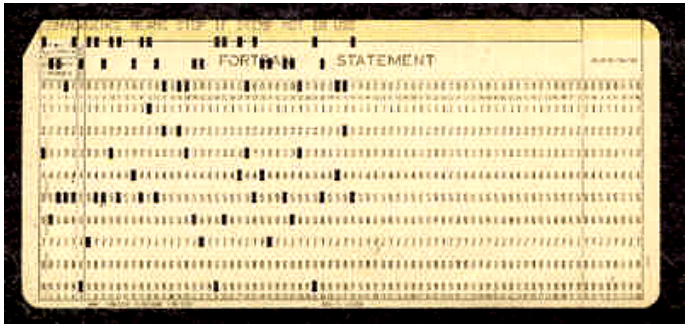
- Inexistência do conceito de sistema operativo
- Computadores a válvulas, grandes e lentos
- A programação das máquinas era feita através da ligação de “plugboards” (placas com componentes eléctricos)
- ENIAC (1946) - Desenvolvido pelo exército norte-americano para cálculo balístico



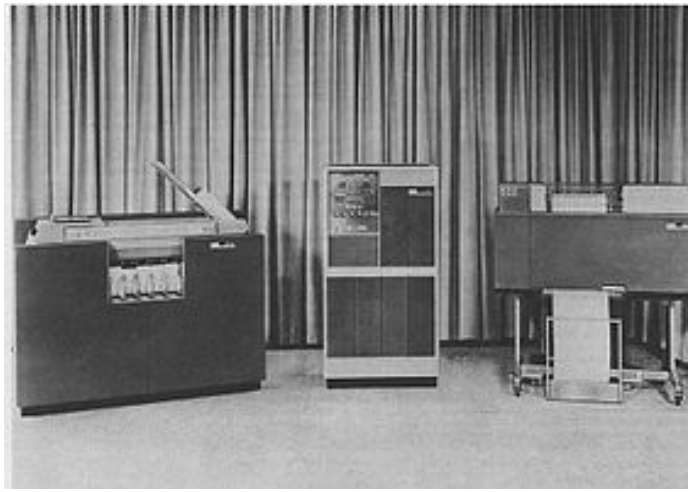


História: 2ª Geração (1955-65)

- Introdução do sistema de processamento por lotes (batch systems)
  - Input, processamento e output feitos em máquinas diferentes para rentabilizar o tempo de CPU

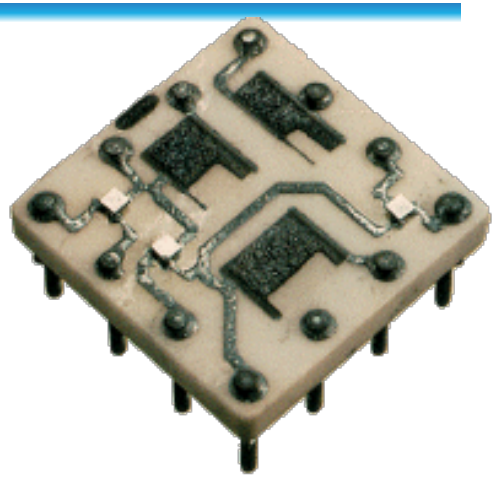


- Computadores a transístores
- Primeiros sistemas operativos:
  - FMS (Fortran Monitor System)
  - IBSYS (Bell Labs)
- Exemplo: IBM 1401 e seus periféricos

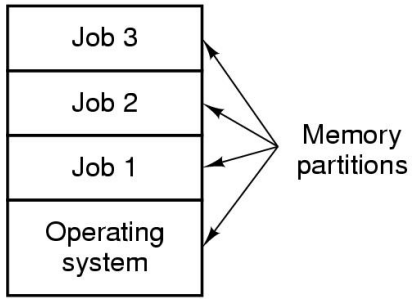


História: 3ª Geração (1965-80)

- Multi-programação e Multi-utilização
  - Multi-programação (multiprogramming): Capacidade de executar vários programas simultaneamente
  - Multi-utilização (timesharing): Capacidade de vários utilizadores poderem utilizar simultaneamente o mesmo computador
- O mecanismo das interrupções permite multiplexar o processador entre várias atividades concorrentes.
- Computadores com CIs
- Primeiras linhas de computadores comerciais
- Aparecimento dos mini-computadores



CI com 3 transístors



3 trabalhos em memória

História: 3ª Geração (1965-80)

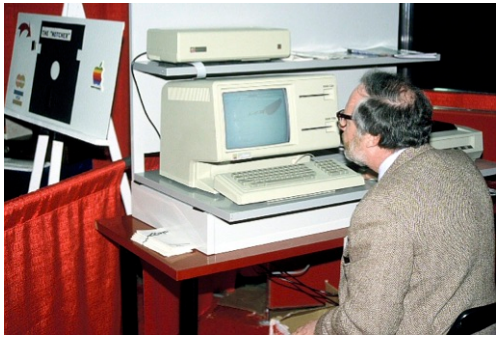
- Sistemas Operativos:
  - OS/360, CTSS, MULTICS, UNIX (System V e BSD)

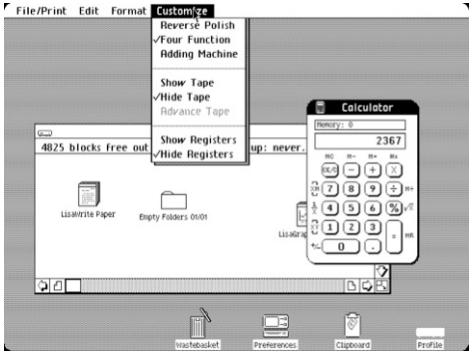
IBM System/360

DEC PDP-11

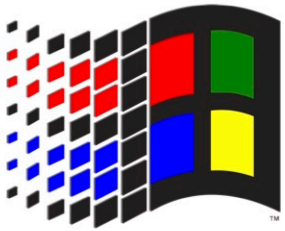
História: 4ª Geração (1980-)

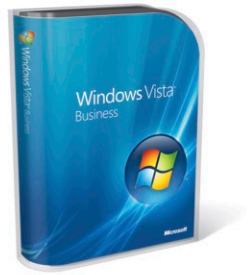
- Computadores pessoais
- LSI e Micro-computadores
- Memória virtual
- Interfaces gráficas (GUI)
- Sistemas em rede
- Sistemas distribuídos
- Sistemas Operativos:
  - VMS, Xenix, Minix, Solaris, Linux, MAC-OS, CP/M, DOS, Windows, etc...





História: 1980-presente





Microsoft Windows xp

Linux

Caixa Mágica Software

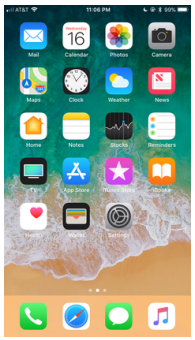
android

História mais recente

- portáteis
- telefones
- tablets
- carros
- máquinas fotográficas (firmware)
- TV
- eletrodomésticos

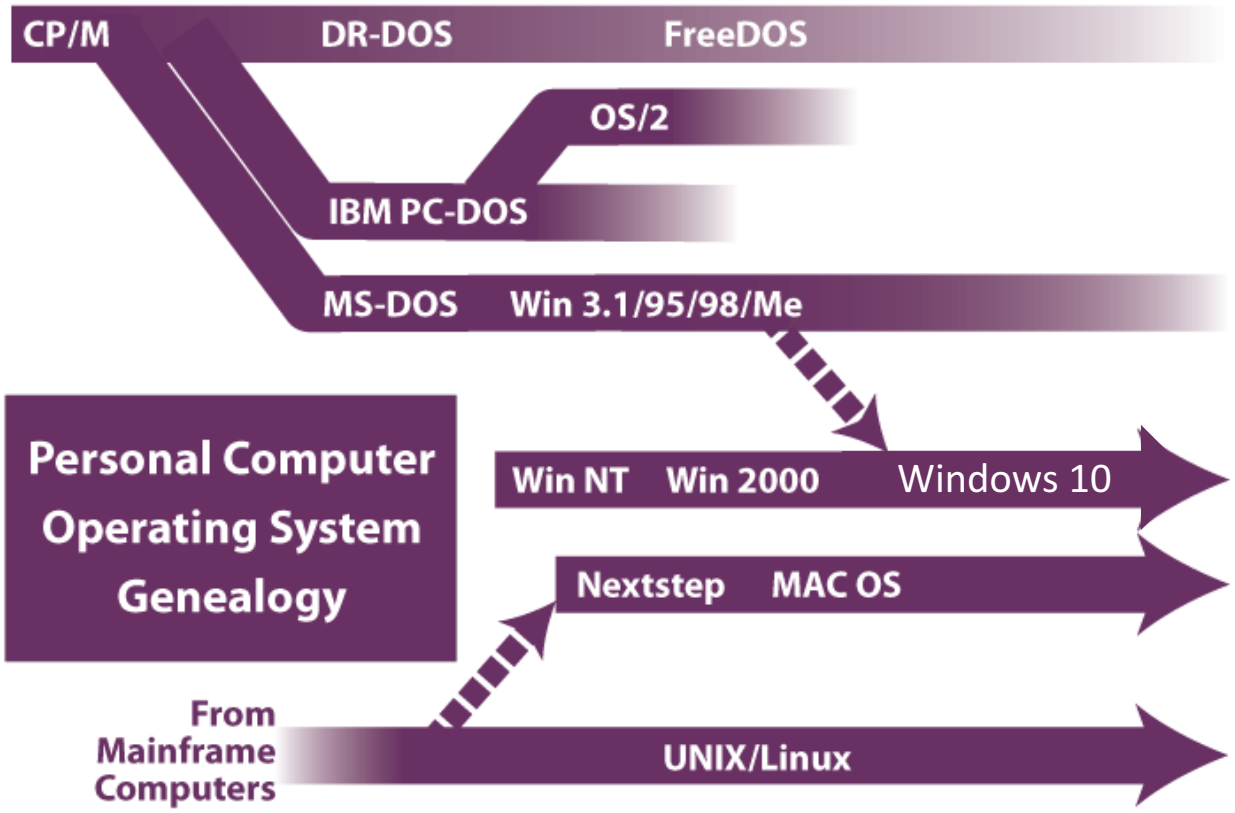
Windows 10

android



macOS Sierra

Proveniência dos atuais SO para PC



extraído de: http://ed-informatics.org/healthcare-it/

O que faz um sistema operativo ?

Há vários pontos de vista

- Os utilizadores querem conveniência, facilidade de utilização, bom desempenho e não se importam com os recursos utilizados
- Computadores partilhados, tais como mainframes ou microcomputadores têm de manter os utilizadores satisfeitos
- Computadores de bolso têm menos recursos e são otimizados para usabilidade e vida da bateria
- Computadores, tais como sistemas embebidos em dispositivos e automóveis, têm pouco ou nenhum interface com o utilizador

Tipos de Sistemas Operativos

- SOs para Mainframes
  - Capacidade para lidar com muitas transações de dados
  - Acesso remoto de milhares de utilizadores
  - Desempenho de múltiplas operações de I/O
  - Utilizados em companhias aéreas, bancos, seguradoras, etc.
  - Exemplo: IBM OS/390
  - Têm vindo a ser gradualmente substituídos por Linux ou outras variantes do Unix.
- SOs para Computadores Pessoais
  - Oferecem um bom conjunto de serviços a um único utilizador (em geral) e uma interface gráfica amigável (ex: Windows, MacOS, Linux)

Tipos de Sistemas Operativos

- SOs para Servidores
  - Oferecem um conjunto de serviços numa rede de computadores local (LAN - Local Area Network)
    - Partilha de ficheiros e periféricos (e.g. impressoras)
    - Acesso à internet
  - Exemplos:
    - Solaris, FreeBSD, Linux, Windows Server 201x
- SOs para Multiprocessadores
  - Capacidade de permitirem a ligação entre diversos CPUs de um sistema
  - Muitos sistemas operativos recentes correm em multiprocessadores.
  - Exemplo: Unix, Windows, Linux, Android, IOS...

Tipos de Sistemas Operativos

- SOs tempo virtual
  - O tempo de execução dos programas não tem relação com o tempo exterior ao computador
  - Sistemas mais utilizados na maioria dos computadores, quer clientes, que servidores (ex: Windows, MacOS, Linux)
- SOs tempo real
  - Utilizados para controlo de processos nos quais o tempo é o fator mais importante. Dá resposta a um acontecimento num intervalo determinado, caso contrário não cumpre a especificação e falha.
  - Inicialmente usados em processos industriais e hoje também para jogos, sistemas de controlo em automóveis, aviões, etc
  - Oferta extensa de SOs de tempo real, muitos para sistemas embebidos

Tipos de Sistemas Operativos

- SOs para computadores de bolso
  - Tablets, smartphones, etc. Hoje em dia bastante sofisticados e lidam com vários CPU, muita memória e sensores, tais como: GPS, cameras, etc.
  - Exemplos: Android, iOS, ...
- SOs para sistemas embebidos
  - Controlo de electrodomésticos e pequenos dispositivos que não se olham como computadores (e.g., televisão, carro, telefones antigos)
  - Software integrado com o hardware.
  - Sistema desenvolvido para um conjunto de operações e não oferece interface para desenvolver aplicações
  - Exemplos: ITRON, Embedded Linux, LynxOS, Vxworks, etc.

Classificações de SOs

- Mono-utilizador vs. Multi-utilizador
- Mono-programação vs. Multi-programação
- Dedicado vs. uso geral
- Centralizado vs. distribuído
- Proprietário vs. aberto

Classificação de SOs

- Mono-utilizador
  - O CPU só pode estar dedicado de forma interativa a um conjunto de processos do mesmo utilizador
    - MS-DOS, todos os Windows
- Multi-utilizador
  - O tempo de processamento do CPU de um computador pode ser partilhado por mais do que um utilizador de forma interativa.
    - Unix, Linux

Classificação de SOs

- Mono-programação/ processamento por lotes
  - Cada programa monopoliza o processador até terminar
    - Spectrum
    - DOS (à parte dos programas residentes)
- Multi-programação
  - Capacidade de correr vários programas simultaneamente (em concorrência)
    - Unix, Linux
    - todos os Windows

Classificação de SOs

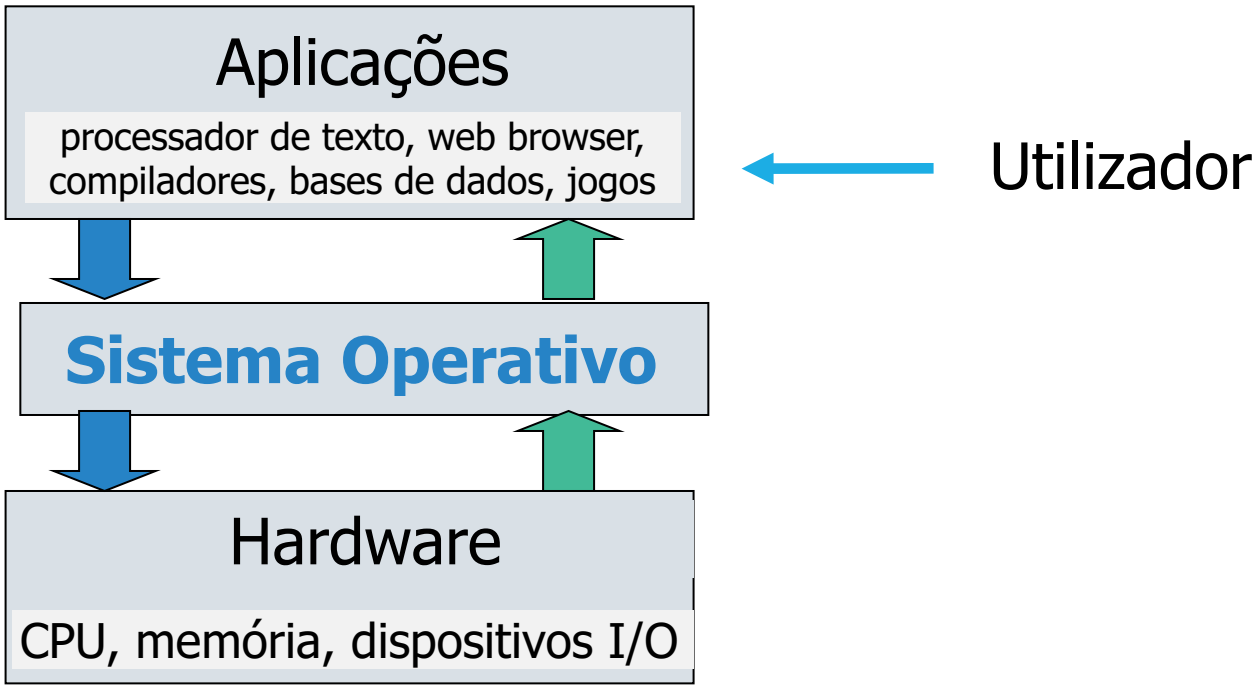
- Dedicado
  - Sistema Operativo projetado para aplicações específicas
  - Exemplos:
    - Controlo de uma linha de montagem - SOs em tempo real
    - Gestão de transações numa companhia aérea - SOs para Mainframes
    - Interface para um telemóvel – SO embebido (embedded)
- Uso geral
  - Projetados para uma fácil utilização
  - Permitem a execução de uma grande variedade de programas
  - Reconhecem uma grande diversidade de periféricos

Classificação de SOs

- Centralizado
  - O Sistema Operativo cria uma máquina virtual sobre um único computador
- Distribuído
  - O Sistema Operativo que corre sobre um conjunto de computadores, dando a ilusão de que este conjunto é uma entidade única
    - Sistemas distribuídos puros
    - Sistemas em rede

Conceito de Sistema Operativo

- O Sistema Operativo pode ser definido como um conjunto de programas que permitem uma interação simplificada entre o utilizador e a máquina



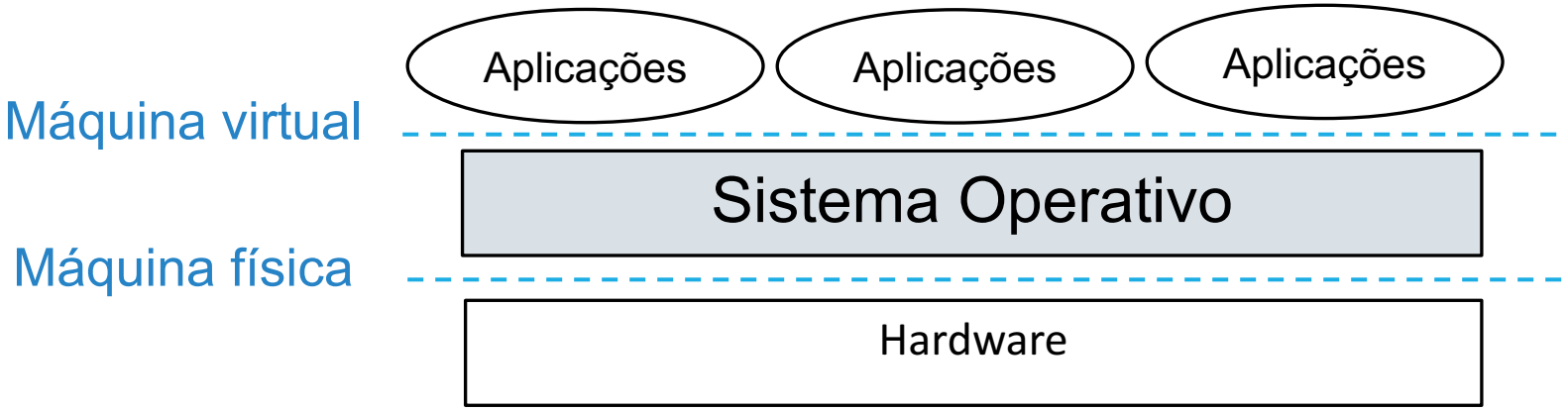
SO visto como *Gestor de Recursos*

- O Sistema Operativo pode ser visto como um Gestor de recursos
  - Efetua a gestão dos diversos componentes da arquitetura de um computador, impondo ordem na atribuição de recursos aos programas.
  - Tira máximo partido dos recursos disponíveis
    - Tempo de CPU, memória, disco, periféricos, etc.
  - Aspetos da máquina física (interrupções, memória, dispositivos, ...) dificilmente poderiam ser controlados pelas aplicações diretamente
  - Permite abstrair os recursos físicos, oferecendo às aplicações um conjunto de recursos lógicos.

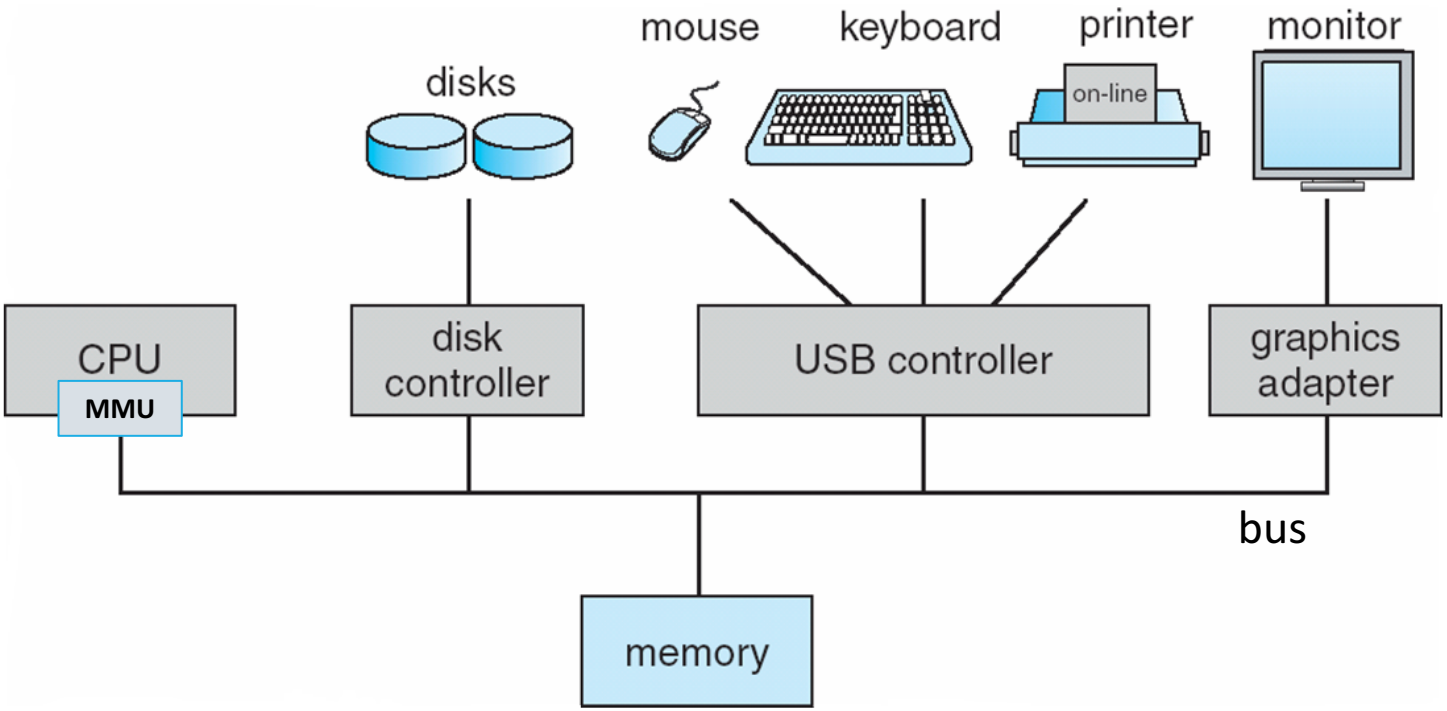
Recursos lógicos	Recursos físicos virtualizados
Processos	CPU
Espaços de endereçamento virtuais	Memória RAM, unidade de gestão de memória
Ficheiros	Dispositivos de memória de massa
Periféricos virtuais	Periféricos físicos
Canais de comunicação	Redes de dados
Utilizadores	Utilizadores humanos

SO visto como *Máquina Virtual*

- O SO também pode ser visto como uma Máquina Virtual
  - Dá ao utilizador a ilusão de dispor de uma máquina muito mais fácil de utilizar e programar do que o hardware.
  - Cria uma máquina virtual sobre a máquina física, que oferece os recursos lógicos básicos para o desenvolvimento de aplicações, Independente do hardware onde executa



Conceitos e Revisões



Modo utilizador e modo núcleo

- Processador (CPU)
  - Elemento ativo do sistema que executa processos
- Os processadores mais recentes suportam
  - Modo utilizador (User Mode / protected mode)
    - Disponível um subconjunto das instruções do CPU. É neste modo que correm as aplicações
  - Modo núcleo (Kernel Mode / supervisor mode)
    - Modo privilegiado do processador, para o qual todas as instruções estão disponíveis.
      Só o Sistema Operativo é que tem acesso a este modo

Funções do Sistema Operativo

- Gestão de processos
- Gestão de memória
- Input/output com periféricos
- Sistema de ficheiros
- Comunicação pela rede
- Autenticação e segurança
- ...

Processos

- Um **processo** é basicamente um programa em execução
  - Num sistema multi-programado, vários processos podem estar a correr simultaneamente
  - Contudo, quando existe um só processador, apenas um processo pode utilizá-lo em cada instante temporal
  - Os processos concorrem pelo processador e cooperam entre si para realizar tarefas mais complexas

Memória

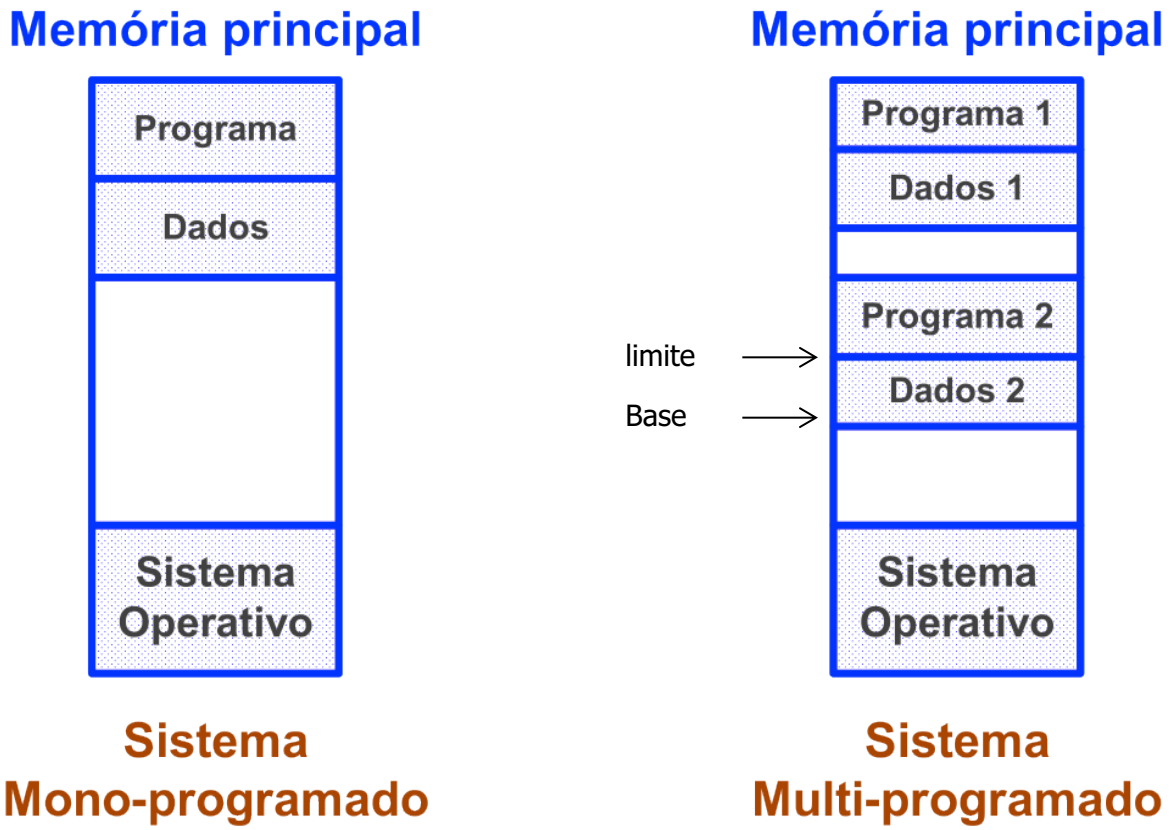
- Memória e a sua hierarquia típica

Gestão de memória

- Gestão de memória
  - Divisão estruturada da memória de modo a ser possível o carregamento de diversos programas na memória principal
    - Colocação e Proteção
  - Existência de mecanismos que permitam o crescimento da memória de dados de um programa
    - Reserva de memória
  - Gestão do espaço de endereçamento de modo a que se possa ter uma capacidade de memória superior à da memória principal (a RAM) – Memória Virtual

Gestão de memória

- Sistemas mono-programados e multi-programados



Input/output com periféricos

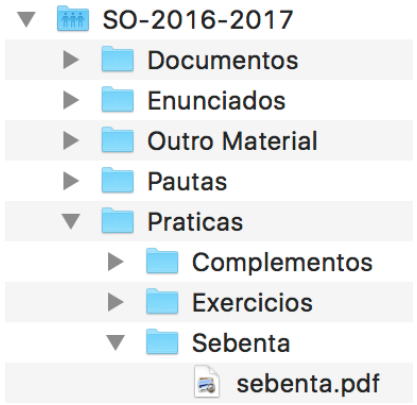
- Periféricos e I/O (input/output)
  - Gestão das operações de escrita e leitura nos diversos periféricos
    - Teclado, impressora, terminais de texto e gráficos, discos, etc.
  - Tratamento de interrupções e de erros
  - Device drivers
    - Programas para gestão de periféricos específicos

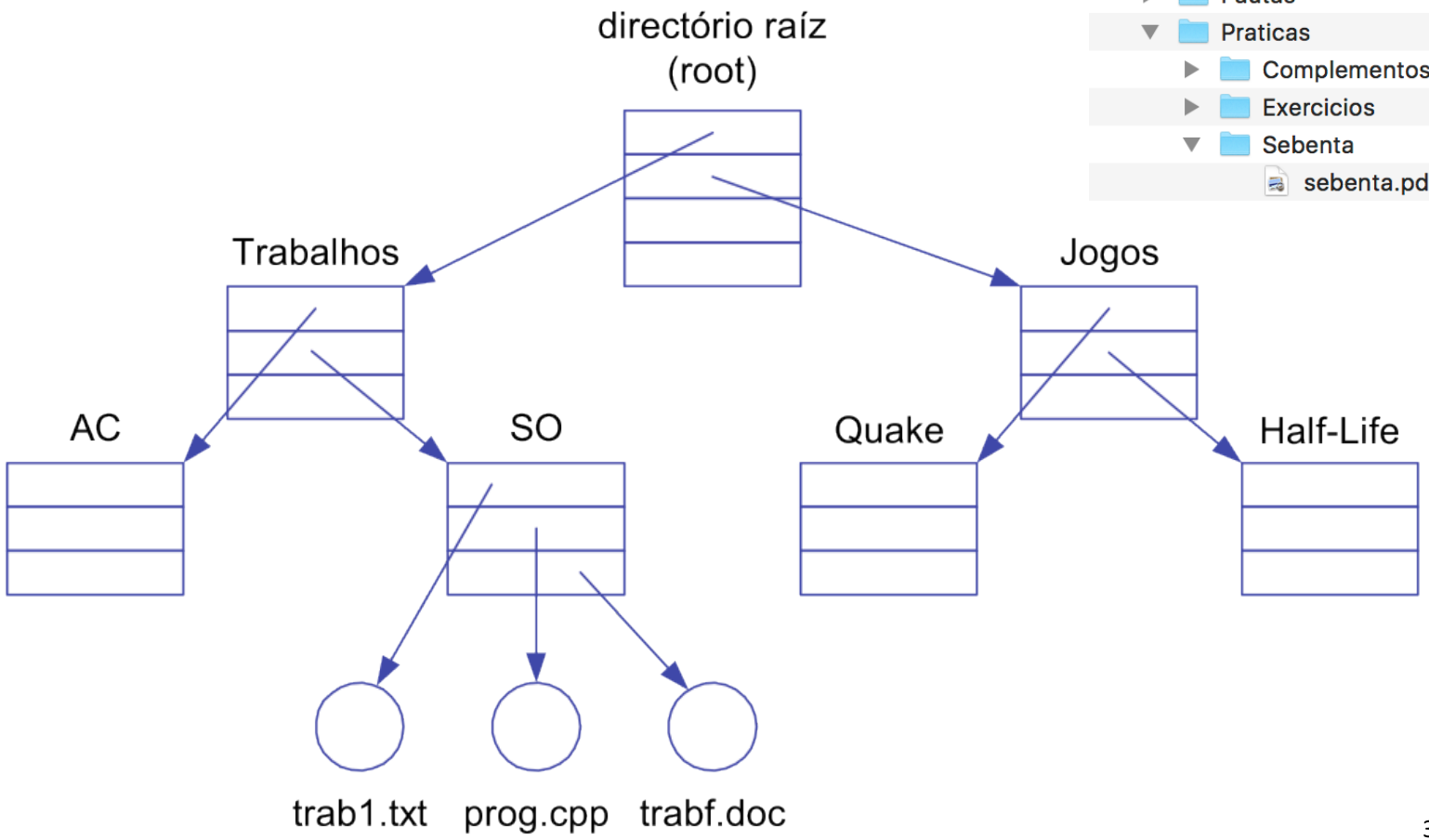
Sistema de ficheiros

- Sistemas de ficheiros
  - Gestão da informação não-volátil armazenada em memória secundária (discos, tapes)
  - Providenciar um nível de abstração para que o utilizador não se preocupe com os detalhes da utilização de discos, disquetes, etc.
  - Chamadas ao sistema:
    - Criação, remoção, cópia, escrita e leitura de ficheiros

Sistema de ficheiros

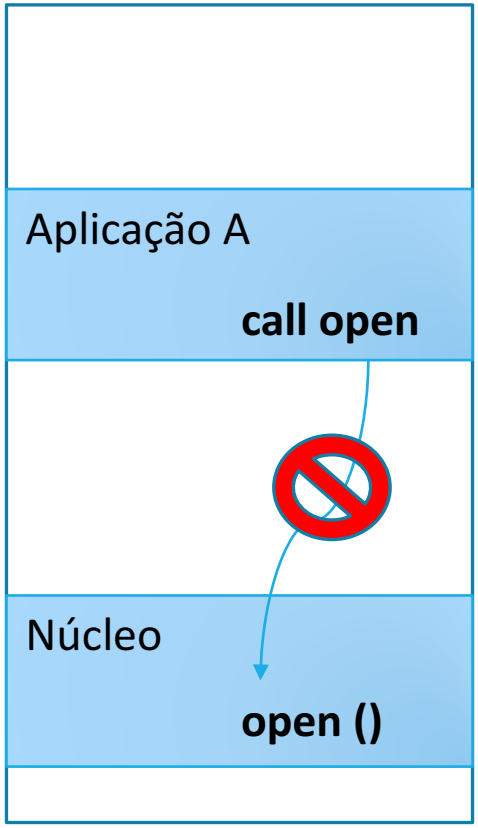
- Estrutura hierárquica (em árvore) - directórios





SO e aplicações, como coexistem?

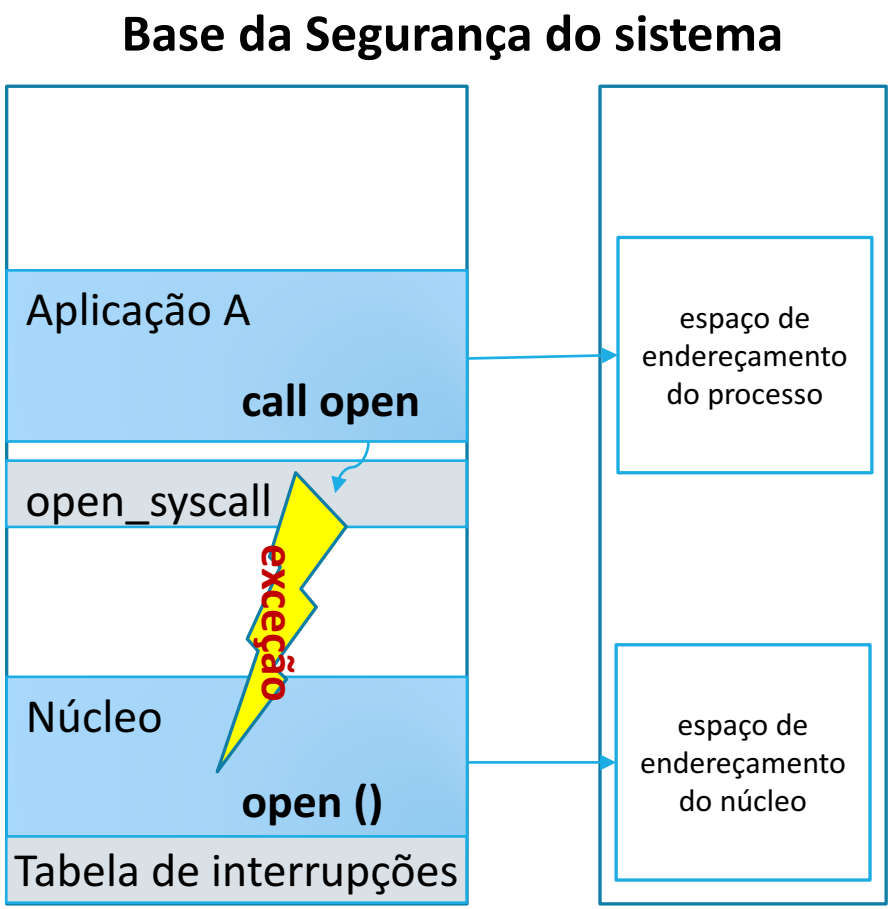
- Problema



SO e aplicações, como coexistem?

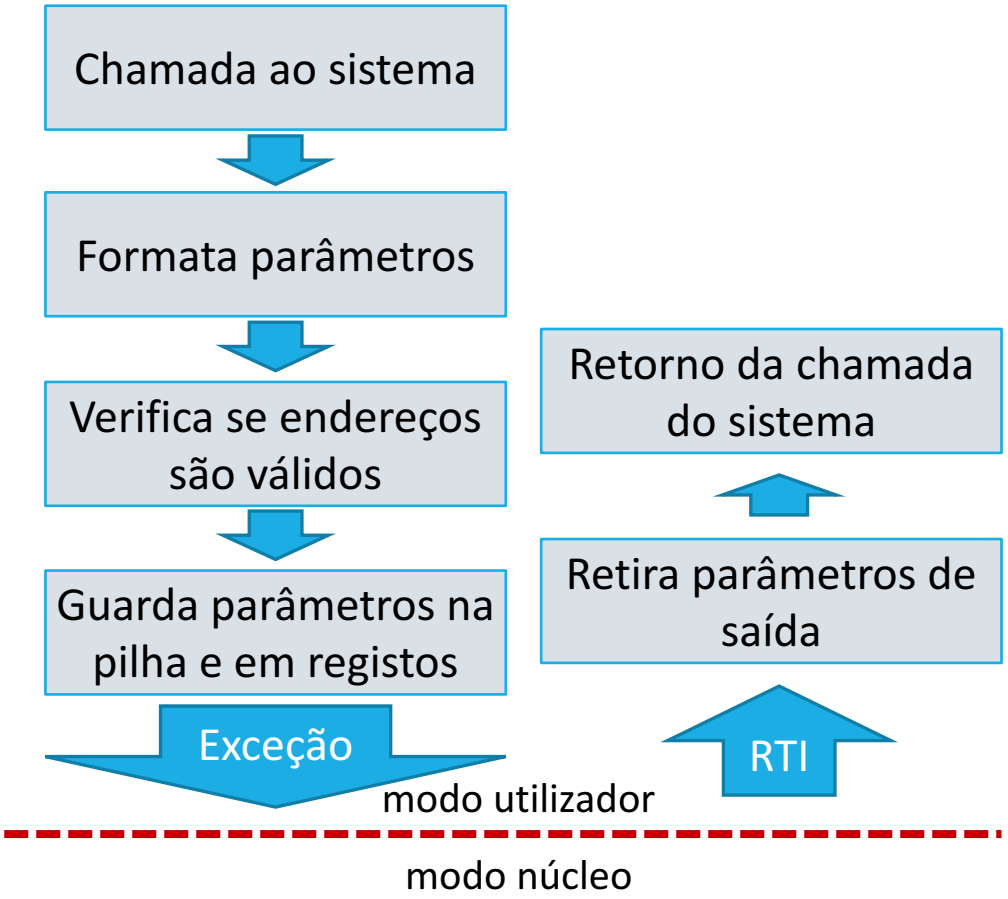
É possível devido à existência de

- 2 modos de execução
- Instruções privilegiadas
- Tradução de memória
- Exceções
- Interrupções
- Chamadas ao Sistema



Razões para passar ao modo núcleo

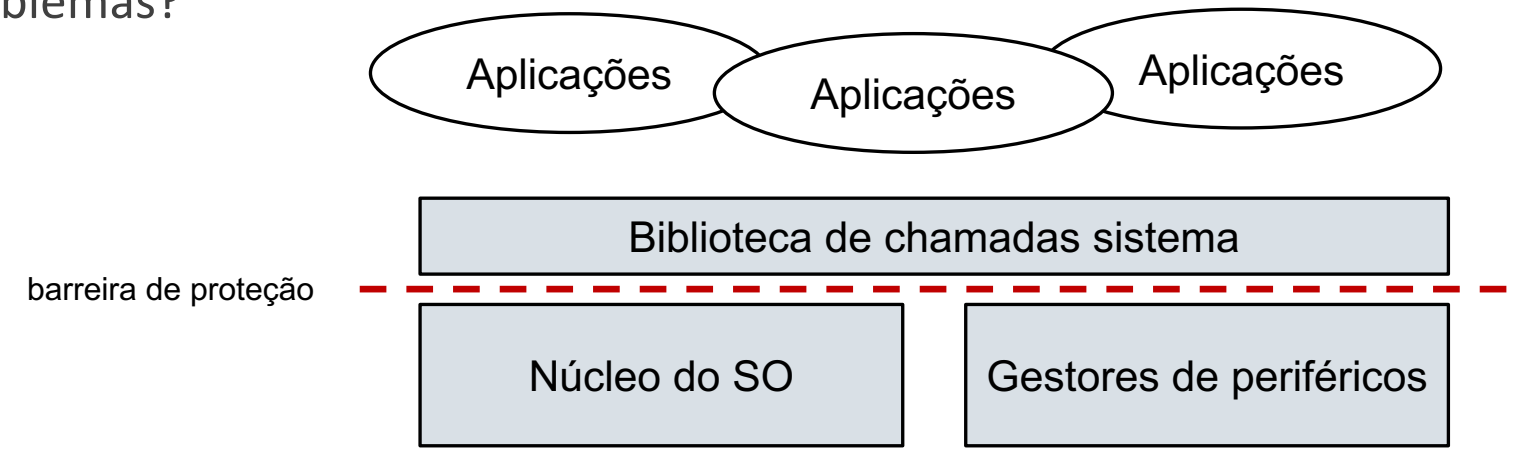
- Exceções
  - causadas pelo processo
  - acesso a endereço inválido, divisão por zero, etc...
- Interrupções
  - originadas em periféricos ...
  - de temporização
- Chamadas ao Sistema



Organização do Sistema Operativo

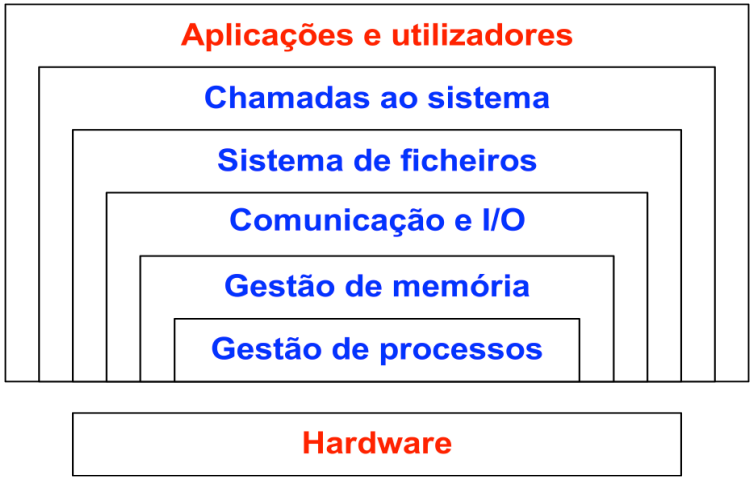
SO: como pode estar organizado?

- Estrutura Monolítica
  - Um único sistema, internamente organizado em módulos
  - Qualquer função pode comunicar com qualquer uma das outras
  - Estruturas de dados globais
  - Como dar suporte à evolução e em especial a novos periféricos ?
    - Solução: gestores de dispositivos (*device drivers*)
  - problemas?



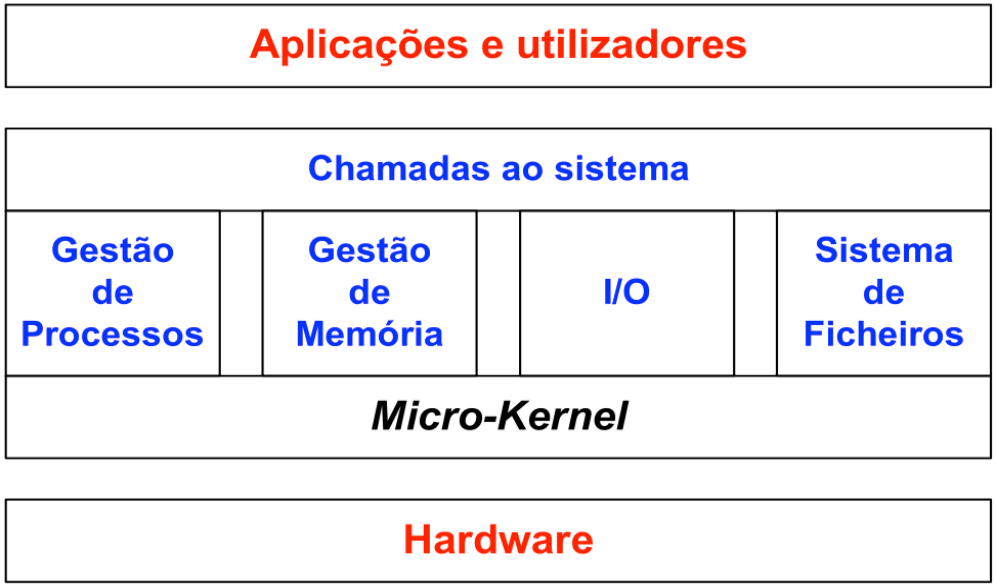
SO: como pode estar organizado?

- Sistema em camadas (Layers)
  - Núcleo estruturado em camadas funcionais
    - Cada camada utiliza serviços de camadas que lhe são interiores
  - Cada camada é uma máquina virtual com uma interface bem definida
    - Torna-se fácil modificar o código de uma camada
  - Maior segurança e robustez
  - Influenciou sistemas como Intel
  - Desvantagem principal?



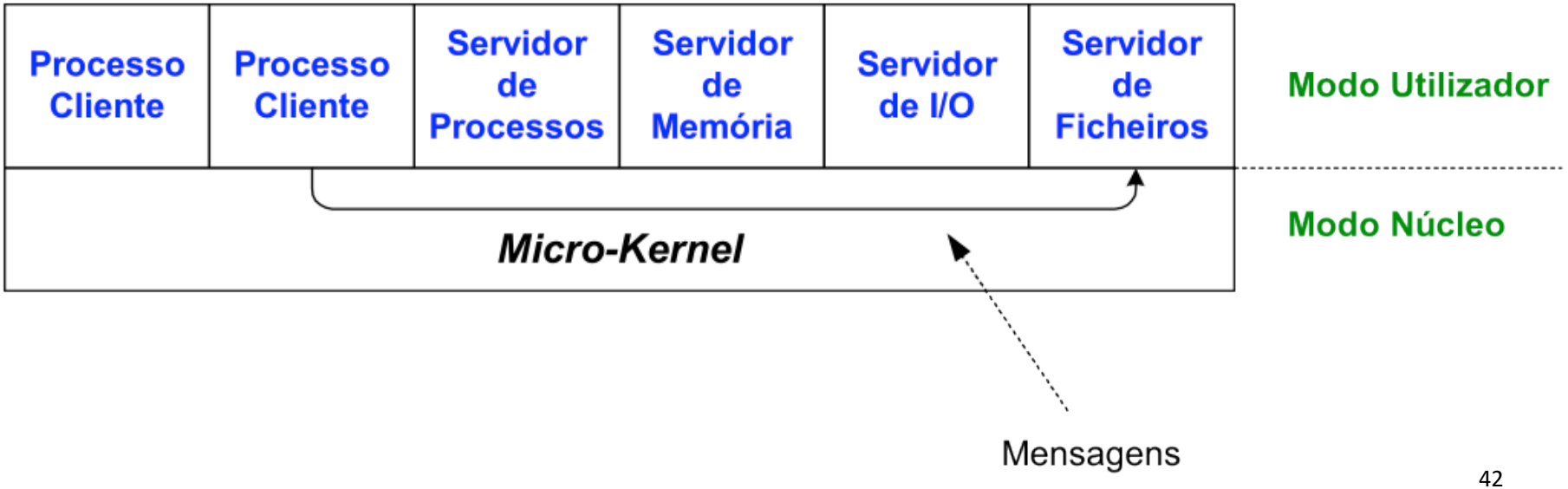
SO: como pode estar organizado?

- Micro-núcleo
  - O SO encontra-se organizado segundo módulos à volta de um micronúcleo (Micro-kernel), geralmente pequeno e que oferece apenas serviços básicos
    - gestão de processos, memória, comunica com o hardware e estabelece a comunicação entre os diversos módulos

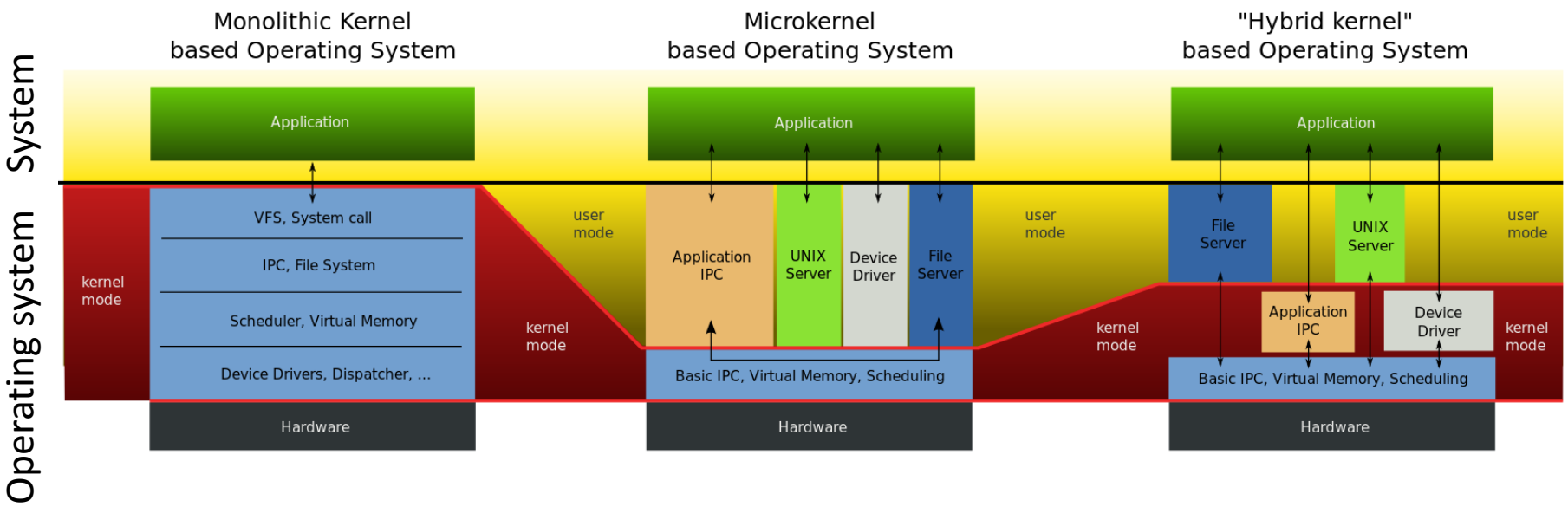


Organização do núcleo

- Modular (Cliente-Servidor)
  - Pode introduzir-se o conceito de processo cliente e de processo servidor que correm em modo utilizador
  - Facilmente adaptável a sistemas distribuídos
  - Estrutura mais estável (teoricamente...)



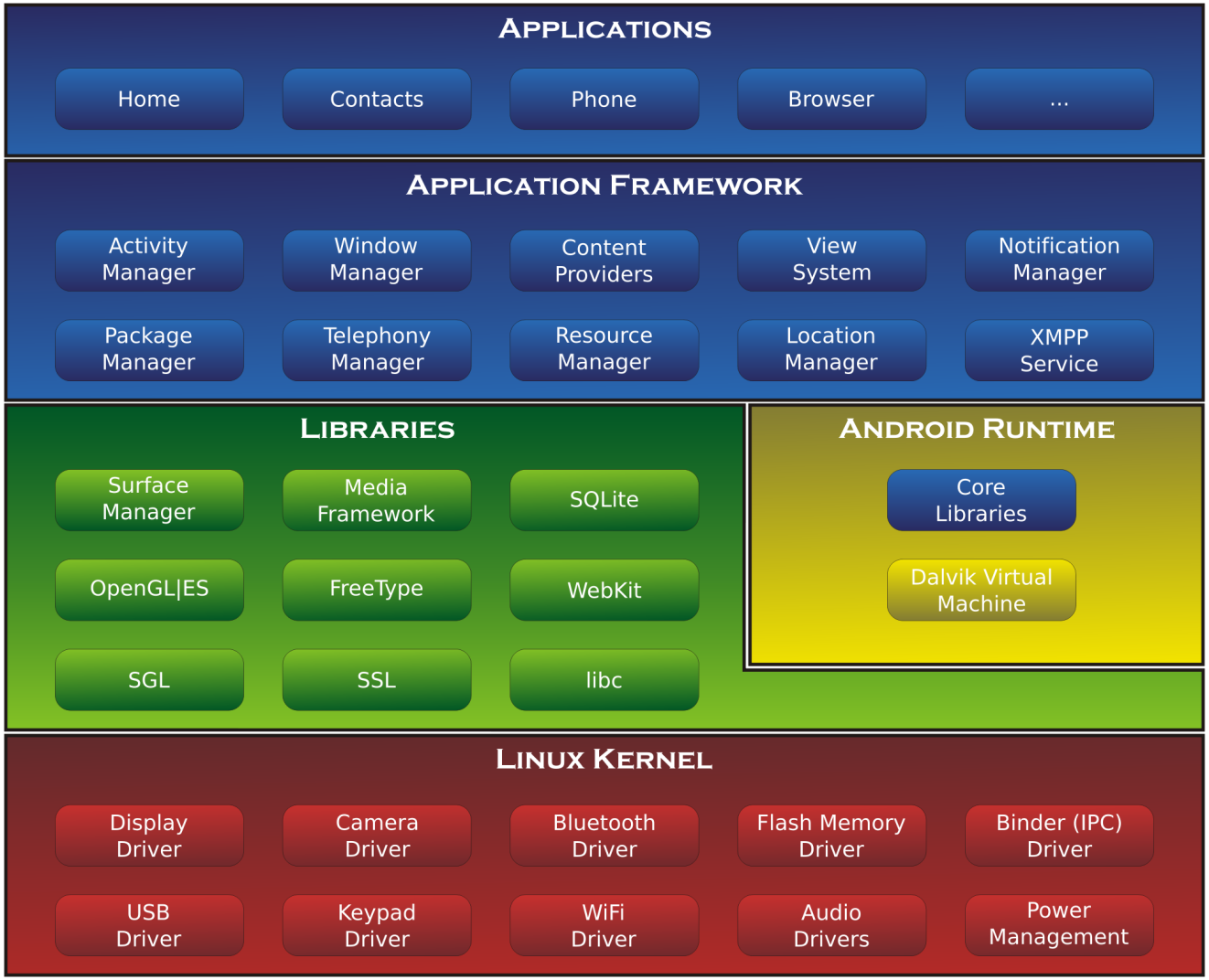
Monolítico vs Micro-kernel



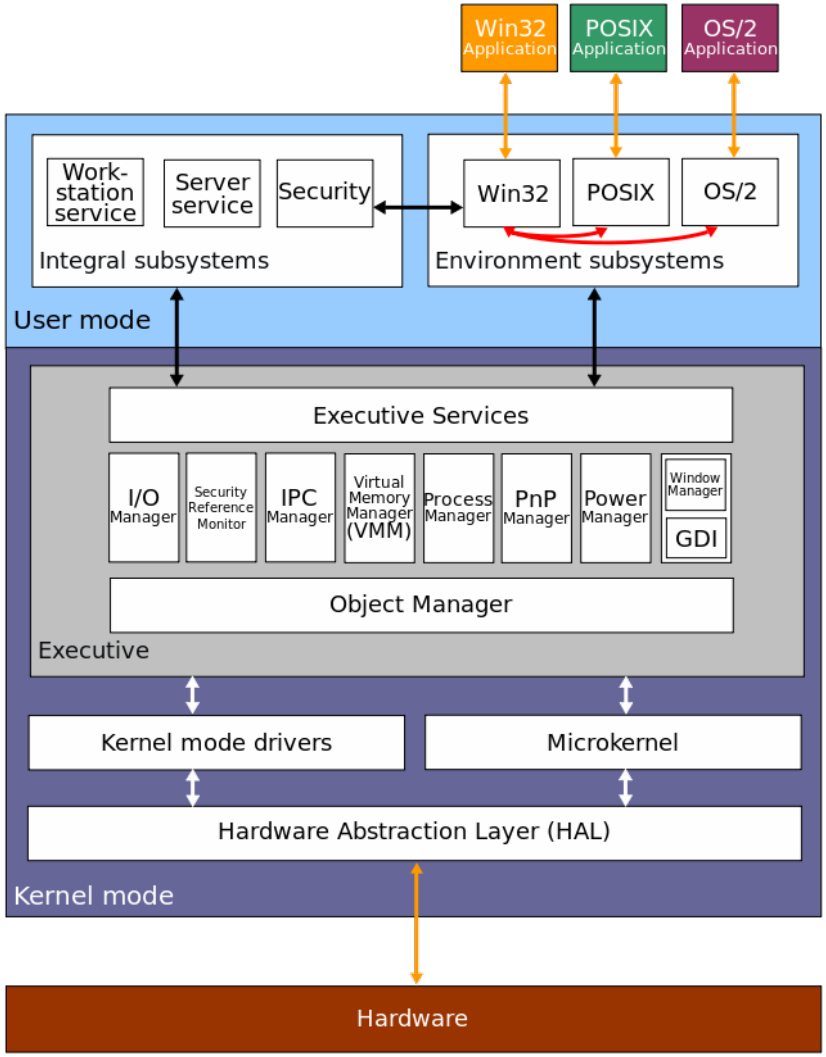
Linux Kernel

<table>
<tr><td rowspan="3">User mode</td><td>User applications</td><td colspan="5">For example, bash, LibreOffice, Apache OpenOffice, Blender, 0 A.D., Mozilla Firefox, etc.</td></tr>
<tr><td>Low-level system components:</td><td>System daemons: systemd, runit, logind, networkd, soundd, ...</td><td>Windowing system: X11, Wayland, Mir, SurfaceFlinger (Android)</td><td colspan="2">Other libraries: GTK+, Qt, EFL, SDL, SFML, FLTK, GNUstep, etc.</td><td>Graphics: Mesa, AMD Catalyst, ...</td></tr>
<tr><td>C standard library</td><td colspan="5"><code>open()</code>, <code>exec()</code>, <code>sbrk()</code>, <code>socket()</code>, <code>fopen()</code>, <code>calloc()</code>, ... (up to 2000 subroutines)
glibc aims to be POSIX/SUS-compatible, uClibc targets embedded systems, bionic written for Android, etc.</td></tr>
<tr><td rowspan="3">Kernel mode</td><td rowspan="3">Linux kernel</td><td colspan="5"><code>stat</code>, <code>splice</code>, <code>dup</code>, <code>read</code>, <code>open</code>, <code>ioctl</code>, <code>write</code>, <code>mmap</code>, <code>close</code>, <code>exit</code>, etc. (about 380 system calls)
The Linux kernel System Call Interface (SCI, aims to be POSIX/SUS-compatible)</td></tr>
<tr><td>Process scheduling subsystem</td><td>IPC subsystem</td><td>Memory management subsystem</td><td>Virtual files subsystem</td><td>Network subsystem</td></tr>
<tr><td colspan="5">Other components: ALSA, DRI, evdev, LVM, device mapper, Linux Network Scheduler, Netfilter
Linux Security Modules: SELinux, TOMOYO, AppArmor, Smack</td></tr>
<tr><td colspan="7">Hardware (CPU, main memory, data storage devices, etc.)</td></tr>
</table>

Android System Architecture



Windows NT (=7, 8, 8.1, 10)



Para os curiosos ...

- Extrato de uma discussão entre o *Andy Tanenbaum* e o *Linus Torvalds* acerca do design do kernel, software livre, etc.
  - **“O linux está obsoleto”, 1992**
    https://root.cern.ch/root/Linus_vs_Tanenbaum.html