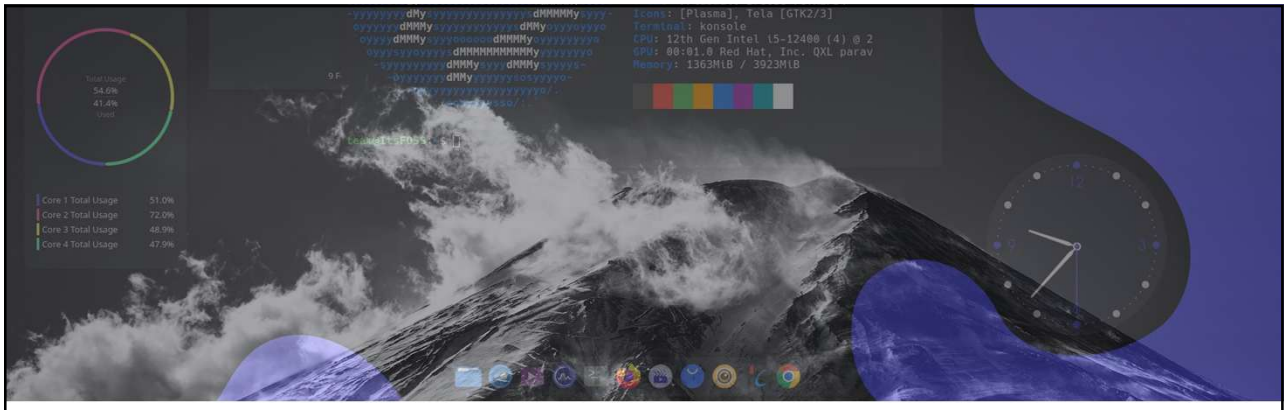


SISTEMAS OPERATIVOS

Arquitecturas do SO

António Godinho

1

ARQUITECTURAS DO SO

Estrutura monolítica

Núcleo monolítico (kernel): onde reside a maior parte da funcionalidade do SO.

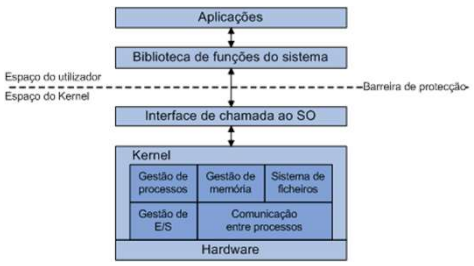
Biblioteca de funções de sistema: interligadas ao código das aplicações e que permitem invocar os serviços do SO.

Interface de chamada ao SO: invocação dos serviços do SO.

Barreira de protecção: separa o espaço de endereçamento do kernel do espaço de endereçamento das aplicações.

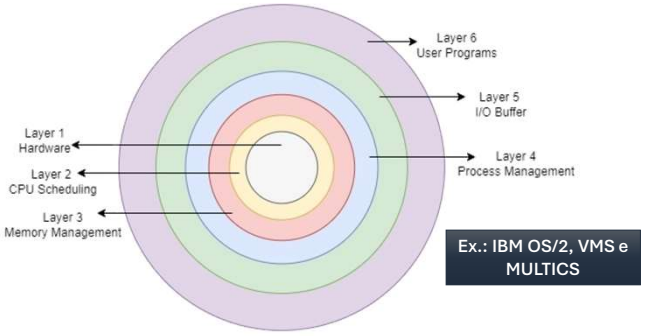
Desvantagem: a alteração de código obriga à reconstrução de todo o kernel.

Ex.: Sistemas UNIX antigos e o MS-DOS. O núcleo do Linux nasceu monolítico



2

ARQUITECTURAS DO SO



Ex.: IBM OS/2, VMS e MULTICS

Estrutura em camadas

Camada: grupo de componentes que realizam tarefas similares. Cada camada comunica apenas com as camadas imediatamente superior e inferior. As camadas inferiores disponibilizam serviços às camadas superiores.

Modularidade: a implementação de cada camada pode ser alterada sem afectar as restantes camadas. Cada camada esconde a sua implementação e apresenta uma interface bem definida às camadas imediatamente superior e inferior.

Desvantagem: desempenho algo condicionado, dada a necessidade de invocar sucessivos métodos, e consequente passagem de parâmetros, entre camadas contíguas.

3

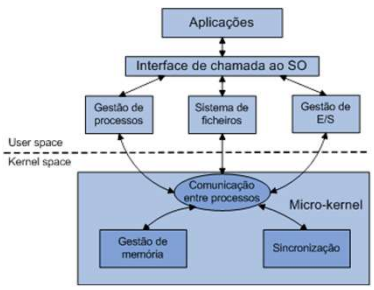
ARQUITECTURAS DO SO

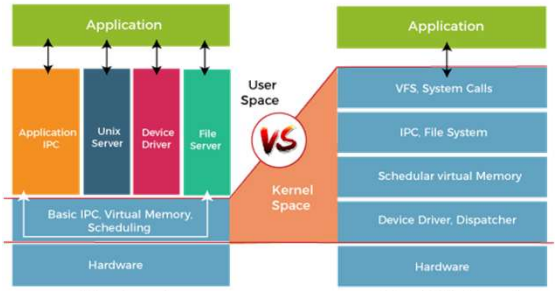
Estrutura em micro-núcleo

Micro-núcleo: pequeno grupo de componentes básicas, numa tentativa de manter um núcleo de dimensão reduzida e escalável. Tipicamente, inclui a gestão de baixo nível da memória e os mecanismos de comunicação e sincronização entre processos.

Os restantes componentes do SO (gestão de processos, gestão de E/S e sistema de ficheiros) executam no espaço do utilizador, com menores privilégios.

Exibe um elevado grau de modularidade, o que os torna extensíveis, portáveis e escaláveis.





Microkernel Monolithic Kernel

4

ARQUITECTURAS DO SO

Ex.: Windows NT, macOS, IBM AIX, Solaris

Arquitetura híbrida

É uma abordagem combinada das arquitecturas do microkernel e do kernel monolítico. Tem um design semelhante ao do kernel monolítico, mas utiliza a modularidade e a estabilidade da arquitetura do microkernel.

Vantagens:

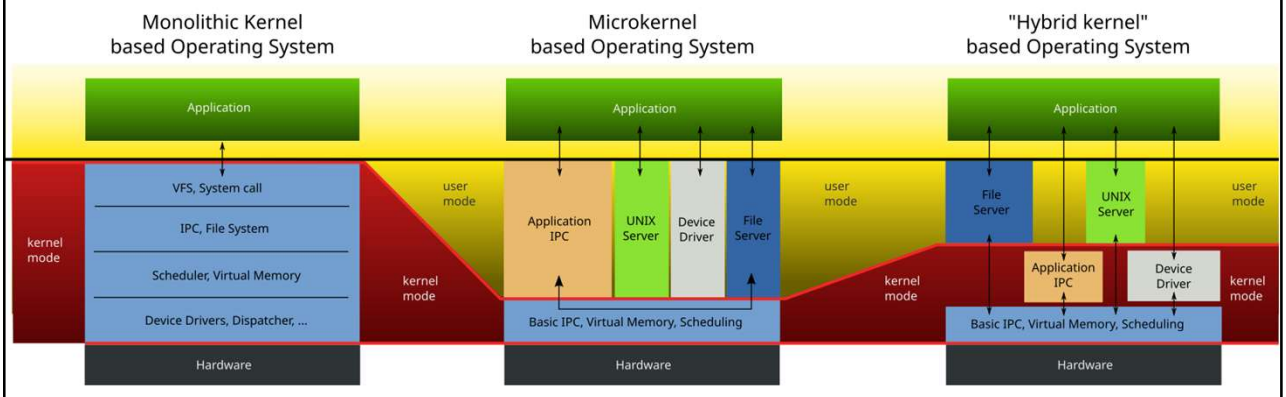
- Flexibilidade: É flexível conceber uma arquitetura híbrida.
- Melhor desempenho: A arquitetura híbrida proporciona um melhor desempenho, uma vez que possui as caraterísticas da arquitetura monolítica e do microkernel.
- Segurança melhorada: A arquitetura híbrida melhora a segurança do sistema ao isolar os serviços críticos.
- Maior suporte de hardware: A arquitetura híbrida pode fornecer um maior suporte de hardware do que uma arquitetura pura de microkernel.

Desvantagens:

- Complexidade: A inclusão de uma arquitetura monolítica e de um microkernel aumenta a complexidade.
- Custo mais elevado: O desenvolvimento da arquitetura híbrida pode ser mais dispendioso do que o de uma arquitetura microkernel pura, devido à necessidade de mecanismos de comunicação inter-processos mais complexos.

5

ARQUITECTURAS DO SO



Fonte da imagem: [Wikipedia]

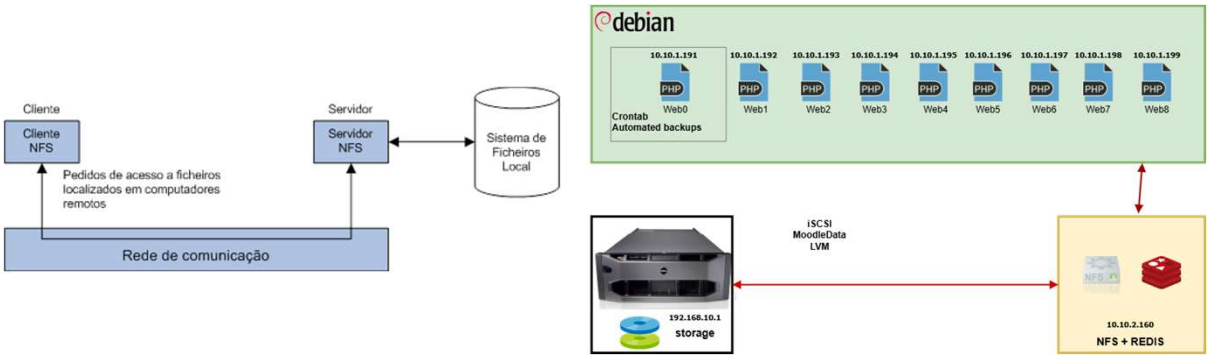
6

ARQUITECTURAS DO SO

Sistemas Operativos Distribuídos e de Rede

SO de Rede (Network Operating System): permite que os processos acedam a recursos que residam em outros computadores da rede, como por exemplo:

- NFS (Sun Microsystem's Network File System) que permite a partilha de ficheiros na rede.

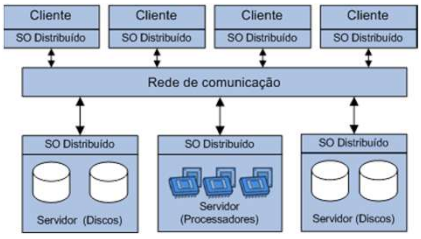


7

ARQUITECTURAS DO SO

Sistemas Operativos Distribuídos e de Rede

SO Distribuído: SO único que gere todos os recursos que se encontram distribuídos por vários computadores da rede. Cria a ilusão de que vários computadores são um único e poderoso sistema computacional.

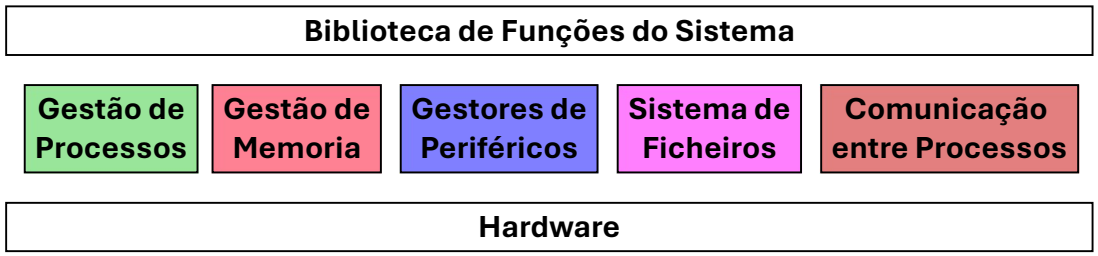


Exemplos de computação distribuida: peer-to-peer, projecto Seti@home, Boinc

8

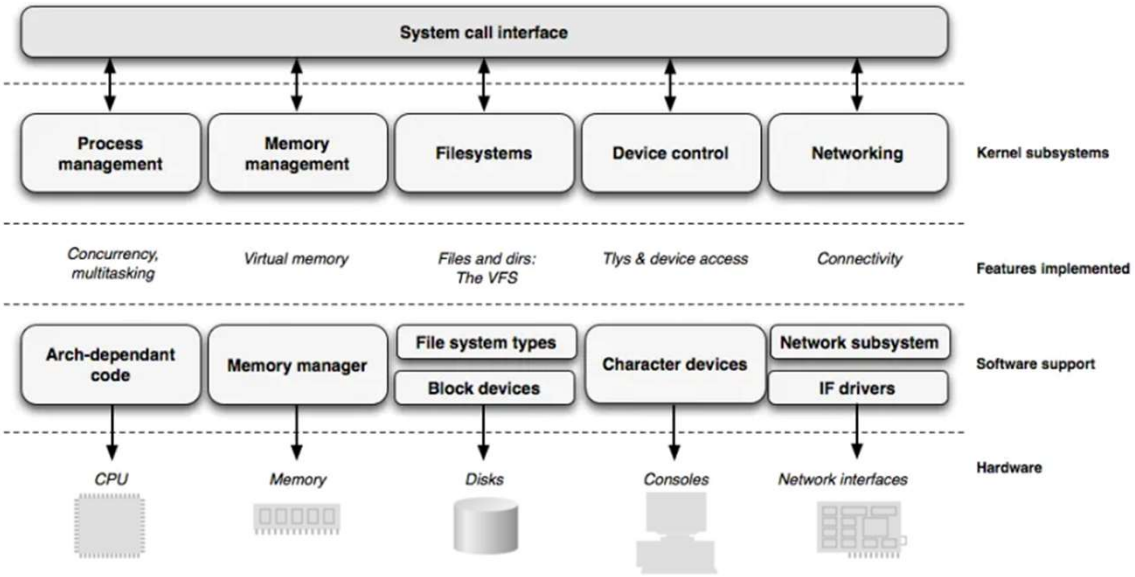
ARQUITECTURA TÍPICA

Para simplificar a exposição podemos usar um modelo de arquitectura conceptual que realça os principais módulos



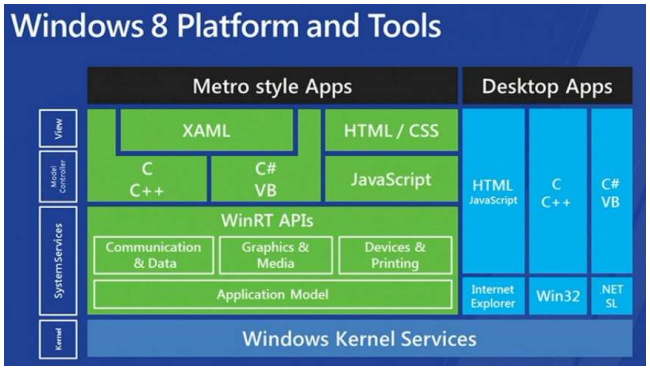
9

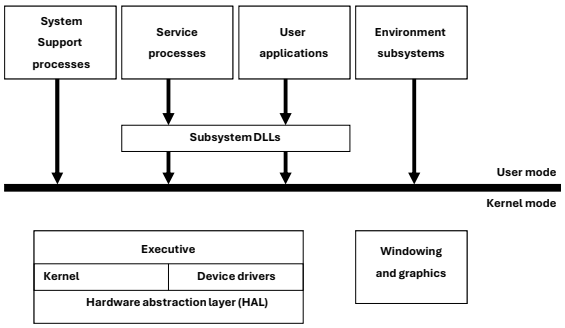
ARQUITECTURA TÍPICA – LINUX



10

ARQUITECTURA TÍPICA – WINDOWS





11

ARQUITECTURA TÍPICA - WINDOWS

Para além de diferenças de âmbito em relação ao diagrama que apresentamos para o Unix convém realçar as seguintes diferenças:

- No Windows existe muita funcionalidade que é executada fora do núcleo do sistema por processos do sistema. Estes processos têm a mesma estrutura que os processos que suportam os utilizadores mas têm privilégios que lhes permitem executar funções de sistema vedadas aos utilizadores normais.
- A estrutura interna em camadas ilustra as principais subdivisões lógicas:
  - Executive
  - Kernel
  - Device drivers
  - Gestor de janelas e Gráficos
  - HAL – Hardware Abstraction Layer
- A forte ligação do sistema operativo ao PC faz com que o sistema de janelas e gráfico faça parte integrante do sistema operativo.
- A estrutura de camadas não corresponde a domínios de proteção diferentes.

12

ARQUITECTURA TÍPICA - WINDOWS

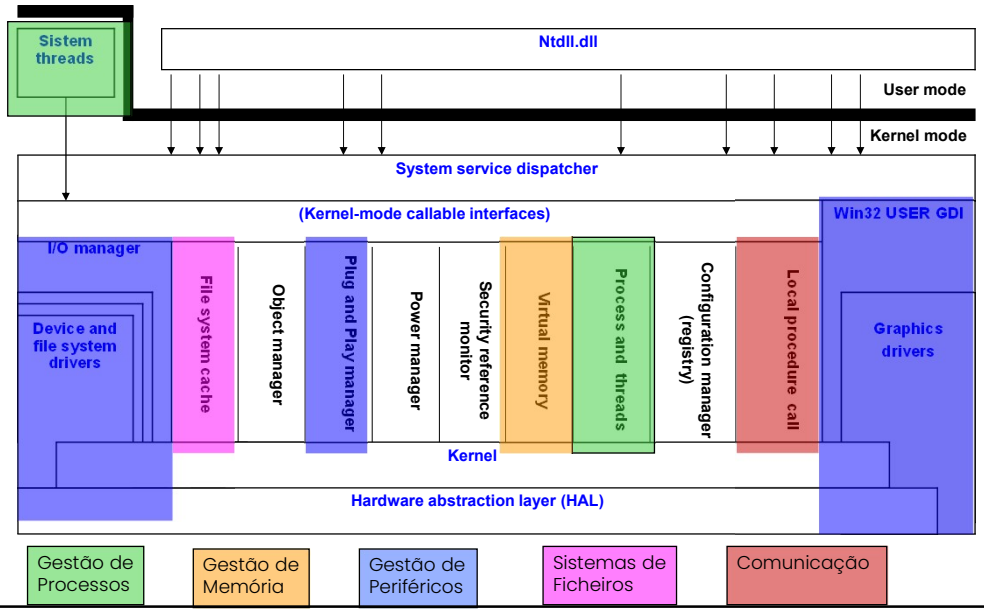
- HAL – Camada que permite isolar o sistema de alguns detalhes de configuração do hardware como por exemplo, configurações da motherboard, ligações dos barramentos.

- Kernel – o kernel tem a funcionalidade de base do sistema, semelhante à que consideramos nos micro-núcleos. Apesar desta decomposição funcional o Windows 2000 não tem uma arquitetura de micro-núcleo. Funcionalidade do kernel:
  - Escalonamento dos fluxos de execução – threads
  - Sincronização de multiprocessadores
  - Tratamento de exceções e interrupções.

- Executive – Nesta camada estão as funções habituais do sistema. Na figura seguinte descreve-se o detalhe interno. É interessante compará-la com a do Unix. A maioria dos módulos tem correspondência direta. Contudo, outros correspondem a funções genéricas que no Unix estão distribuídas por vários módulos como o gestor de objetos do núcleo, o módulo de segurança, o módulo de inicialização (registry).

13

ARQUITECTURA TÍPICA - WINDOWS



14

ARQUITECTURA TÍPICA - WINDOWS

Enviroments Subsystems – Os subsistemas têm por objectivo poder adaptar o sistema de base a diferentes personalizações de sistema ou seja a diferentes modelos computacionais

- No Windows 2000 existem três ambientes standard:
  - WIN32 – ambiente nativo que tem de estar sempre presente
  - POSIX – interface Posix versão normalizada da interface Unix
  - OS/2 – sistema desenvolvido pela IBM e a Microsoft e antecessor do Windows – NT
- Cada aplicação está ligada a um ambiente sendo registado qual dos ambientes vai ser usado quando se inicia a execução
- É interessante notar que este conceito de personalização do sistema é um dos conceitos introduzidos pelos micro-núcleos.

Subsystem DLL – Ntdll.dll – As DLL (Dynamic Loadable Libraries) são as funções do sistema organizadas em bibliotecas que são dinamicamente ligadas às aplicações.

- A vantagem desta organização é permitir que as bibliotecas sejam partilhadas por diversas aplicações.
- Nas DLL existem as funções sistema que chamam os módulos do executive (**chamadas sistema clássicas**) e que são essencialmente as funções da WIN32
- Interface para os subsistemas e funções auxiliares. Estas funções são invocadas através dos environments.

15

ARQUITECTURA TÍPICA - WINDOWS

System Support Processes – processos auxiliares do sistema operativo.

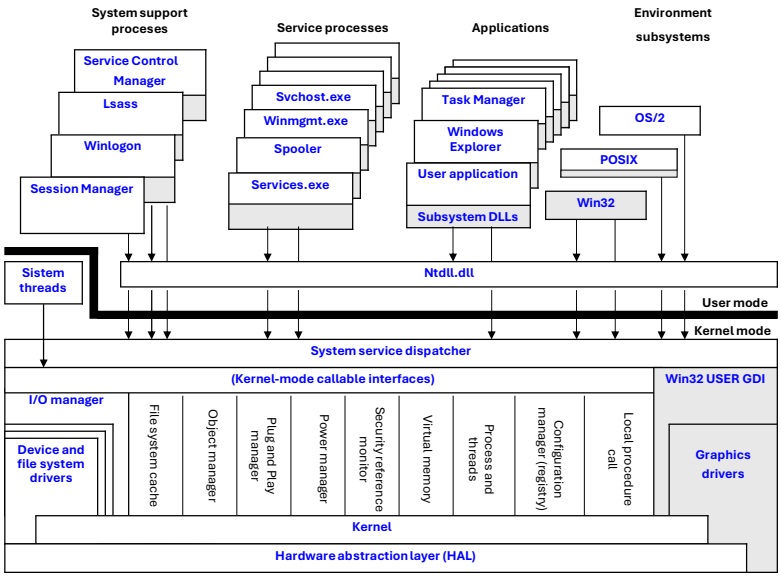
- por exemplo o processo de login (efectua o login e o logout do sistema), o session manager (gestor da sessão do utilizador), o gestor de serviços (service control manager- serviços lançados automaticamente), o idle process, etc.
- As funções executadas pelo processo sistema são complexas como veremos mais adiante e correspondem a funcionalidade indispensável no sistema mas que não precisa de estar no núcleo podendo executar-se em processos independentes.

Service Processes – Servidores ou gestores de periférico semelhantes aos processos daemon do Unix.

- São aplicações Win32 com código adicional para interactuarem com o Service Control Manager.
- São também utilizados para executarem parte de aplicações complexas. Ex.: Exchange, SQL server.

16

ARQUITECTURA TÍPICA - WINDOWS



17

HISTÓRIA (MUITO) RESUMIDA

✓Anos 40 : cada programa executava sozinho e tinha total controle do computador.

✓O carregamento do programa em memória, a varredura dos periféricos de entrada para busca de dados, a computação propriamente dita e o envio dos resultados para os periféricos de saída, byte a byte, tudo devia ser programado detalhadamente pelo programador da aplicação.

✓Anos 50 : os sistemas de computação fornecem "bibliotecas de sistema" (system libraries) que encapsulam o acesso aos periféricos, para facilitar a programação de aplicações.

✓Algumas vezes um programa "monitor" (system monitor) auxilia a carga e descarga de aplicações e/ou dados entre a memória e periféricos (geralmente leitores de cartão perfurado, fitas magnéticas e impressoras de caracteres).

✓1954: MIT's OS

✓1957: BESYS da Bell Labs

✓1961 : o MIT anuncia o desenvolvimento do CTSS – Compatible Time-Sharing System, o primeiro sistema operativo com tempo partilhado.

✓1965 : a IBM lança o BOS/360, um sistema operativo avançado, com partilha de tempo e excelente suporte a discos.

✓1965 : um projeto conjunto entre MIT, GE e Bell Labs define o sistema operativo Multics, cujas ideias inovadoras irão influenciar novos sistemas durante décadas.

✓1966: IBM OS/360 e DOS/360

✓1969 : AT&T cria o Unics (em 1970 muda o nome para Unix). Ken Thompson e Dennis Ritchie, pesquisadores do Bell Labs, criam a primeira versão do UNIX.

✓1975: CP/M (criado para os processadores 8080). Nasce a Microsoft. Compra QDOS

✓1978: Apple DOS

✓1981 : PC-DOS e MS-DOS 1.0

✓1982: SUN OS 1.0

18

HISTÓRIA (MUITO) RESUMIDA

✓1983: SCO XENIX System V para processadores Intel 8086 e 8088 e Novell Netware

✓1984 : a Apple lança o sistema operativo OS System 1.0, o primeiro a ter uma interface gráfica totalmente incorporada no sistema.

✓1985 : primeira tentativa da Microsoft no campo dos sistemas operativos com interface gráfica, através do MS-Windows 1.0.

✓1987 : Andrew Tanenbaum, um professor de computação holandês, desenvolve um sistema operativo didáctico simplificado, mas respeitando a API do UNIX, que foi baptizado como Minix.

✓1987 : IBM e Microsoft apresentam a primeira versão do OS/2, um sistema multitarefa destinado a substituir o MS-DOS e o Windows. Mais tarde, as duas empresas rompem a parceria; a IBM continua no OS/2 e a Microsoft investe no ambiente Windows.

✓1991 : Linus Torvalds, um estudante finlandês, inicia o desenvolvimento do Linux, lançando na rede Usenet o kernel 0.01, logo abraçado por centenas de programadores de todo o mundo.

✓1993 : a Microsoft lança o Windows NT, o primeiro sistema de 32 bits da empresa.

✓1993 : lançamento dos UNIX de código aberto FreeBSD e NetBSD.

✓1995: Windows 95, OpenBSD, Ultrix

✓2000: Windows 2000

✓2001 : a Apple lança o MacOS X, um sistema operativo derivado da família UNIX BSD.

✓2001 : Windows XP

✓2006 : Windows Vista.

✓2008: Windows Server 2008, Android, RedHat Linux 5

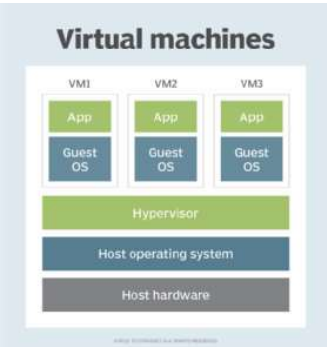
✓2009: Windows 7, FreeBSD 8, Fedora 12, Ubuntu 9.1, Solaris 10, Android 2.0, OpenSuse 11.2, OpenBSD 4.6, Mac OS X 10.6, …

19

ARQUITECTURAS DO SO

Máquinas virtuais

Máquina virtual: é um software de aplicação que cria uma camada virtual de hardware sobre o qual corre outro sistema operativo. O sistema operativo da máquina virtual gere os recursos disponibilizados pela camada virtual de hardware. Cria componentes de software que virtualizam componentes do sistema físico (processador, memória, sistema de ficheiros). A máquina virtual interage com o hardware através do SO nativo.



https://www.techtarget.com/searchitoperations/definition/virtual-machine-VM

20

ARQUITECTURAS DO SO

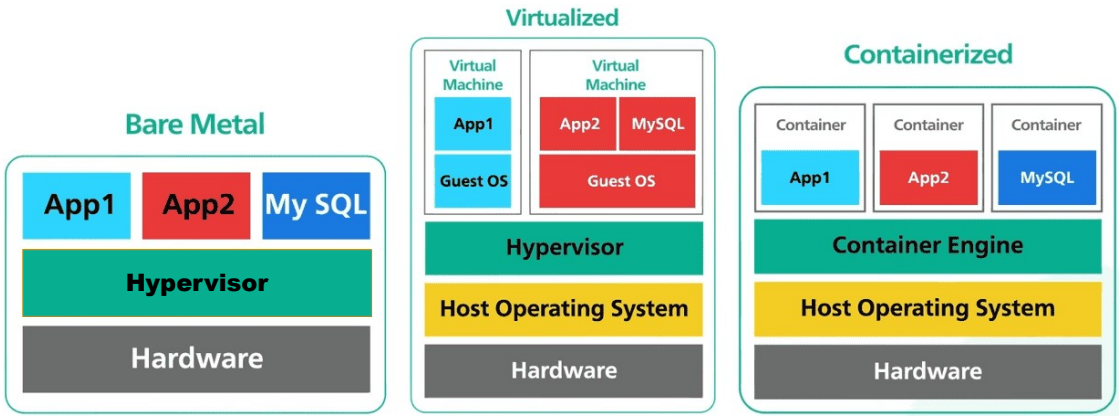
Máquinas virtuais

- O que é uma VM: Uma [máquina virtual (VM)](w) é uma emulação virtual de um computador físico, usando software para executar aplicativos como se fosse um hardware real.
- Isolamento e segurança: VMs permitem rodar um [sistema operacional (SO)](w) de forma isolada do computador hospedeiro, sendo úteis para consolidar recursos e realizar tarefas de risco com mais segurança.
- Recursos virtuais e independência: Os recursos utilizados pela VM são alocados virtualmente do host, permitindo que múltiplas VMs funcionem de forma independente em um mesmo equipamento físico.

21

ARQUITECTURAS DO SO

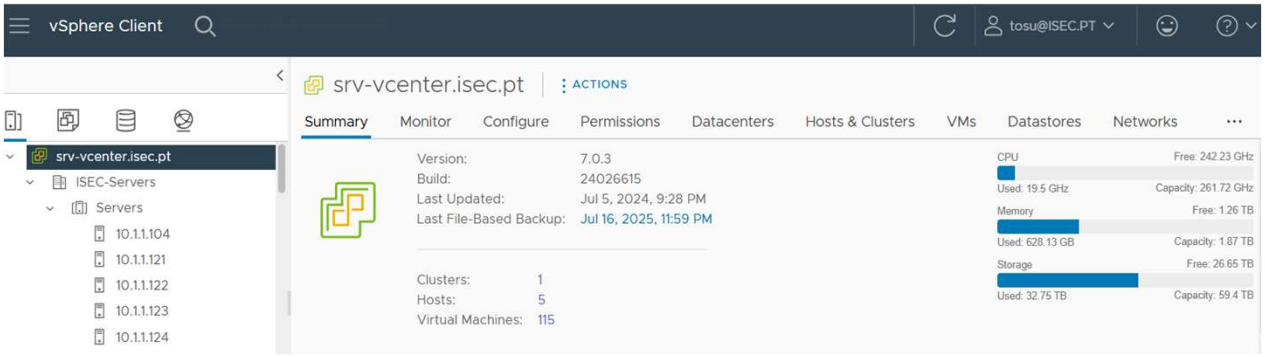
Máquinas virtuais – Bare Metal vs Virtualização vs Containers



22

ARQUITECTURAS DO SO

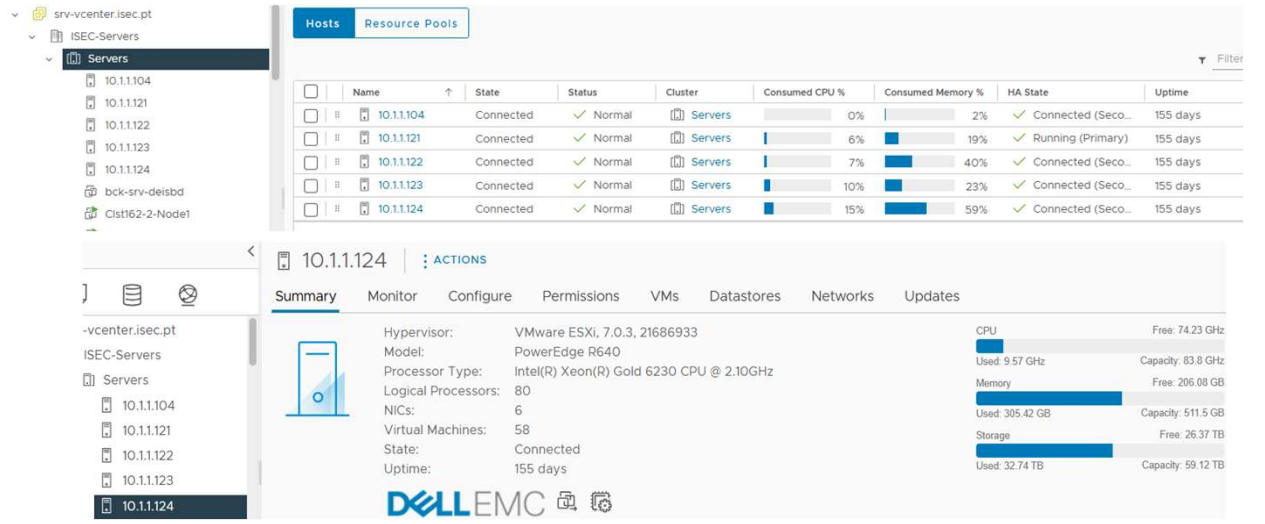
Máquinas virtuais – Exemplo prático – VMWARE



23

ARQUITECTURAS DO SO

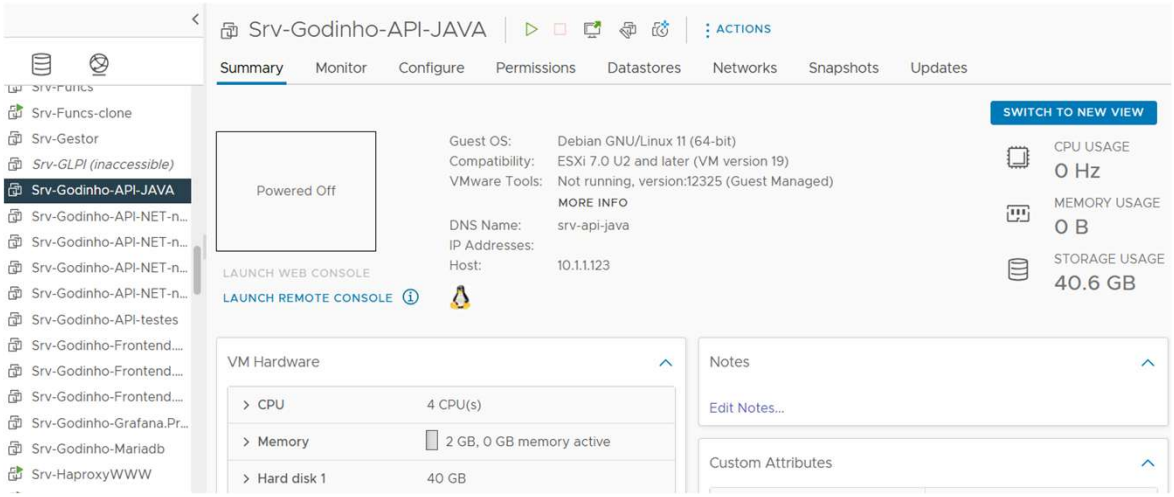
Máquinas virtuais – Exemplo prático – VMWARE



24

ARQUITECTURAS DO SO

Máquinas virtuais – Exemplo prático – VMWARE



25

HISTÓRIA

A primeira versão do UNIX foi desenvolvida em 1969 por Ken Thompson e Dennis Ritchie nos Laboratórios Bell da AT&T.

A Versão 6, lançada em 1976, foi a primeira com uma utilização significativa fora dos laboratórios Bell e de algumas universidades.

Durante os anos 80 e 90 foram desenvolvidas várias versões comerciais do UNIX.

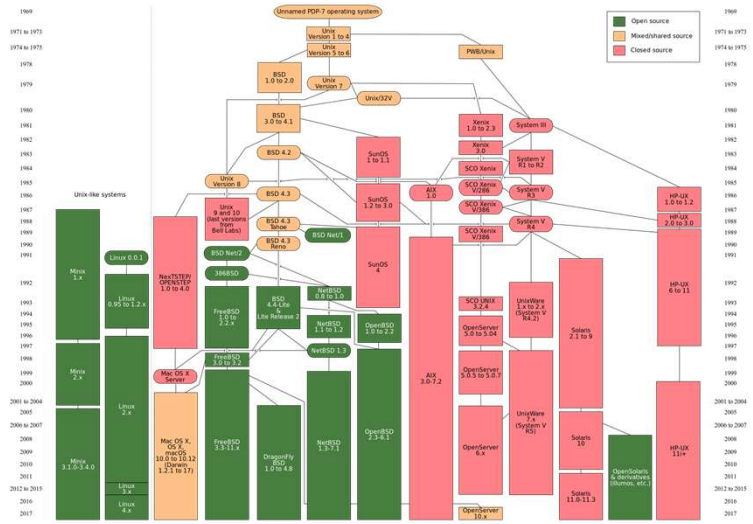
Em 1991, Linus Torvalds desenvolve o Linux – UNIX nos computadores pessoais.

26

26

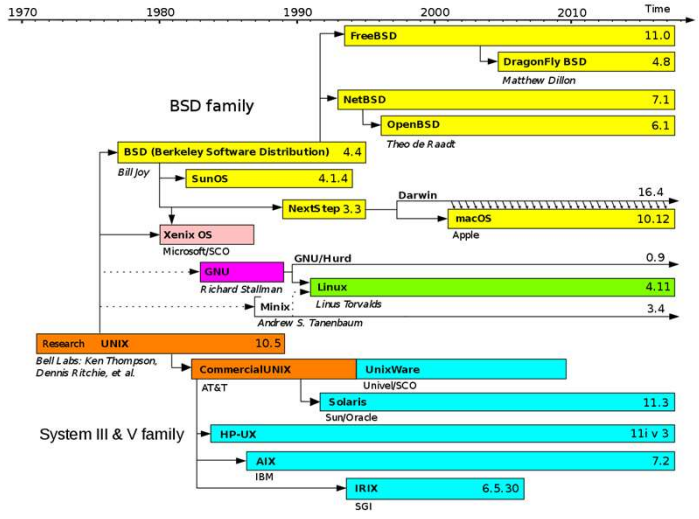
HISTÓRIA

HTTPS://EN.WIKIPEDIA.ORG/WIKI/HISTORY_OF_UNIX#/MEDIA/FILE:UNIX_HISTORY-SIMPLE.SVG



27

HISTÓRIA



28

ARQUITECTURAS DO SO

GNU/Linux

Richard Stallman quanto Linus Torvalds tiveram papéis fundamentais na criação do que hoje conhecemos como Linux:

1. Richard Stallman:
   - Fundador do projeto GNU (em 1983) e da Free Software Foundation.
   - Defendeu a filosofia do software livre (liberdade de usar, estudar, modificar e redistribuir software).
   - Criou licenças como a GPL (GNU General Public License), que garante essas liberdades.
   - Desenvolveu ferramentas essenciais (compiladores, bibliotecas, utilitários) que hoje formam grande parte do "sistema operacional GNU", usado em conjunto com o kernel Linux.
2. Linus Torvalds:
   - Em 1991, criou o kernel Linux, o núcleo do sistema operacional.
   - Esse kernel, combinado com as ferramentas GNU, resultou no que chamamos de GNU/Linux (embora popularmente só se diga "Linux").
   - Torvalds também estabeleceu um modelo de desenvolvimento colaborativo aberto, que fez o Linux crescer e se consolidar como base de servidores, supercomputadores, dispositivos móveis (Android) e até desktops.

29

ARQUITECTURAS DO SO

30