

SISTEMAS OPERATIVOS



ANDROID



INFORMÁTICA DE GESTÃO

Professor Doutor Ricardo Vardasca, PhD, ASIS, FRPS
ricardo.vardasca@islasantarem.pt

ISLA

Santarém

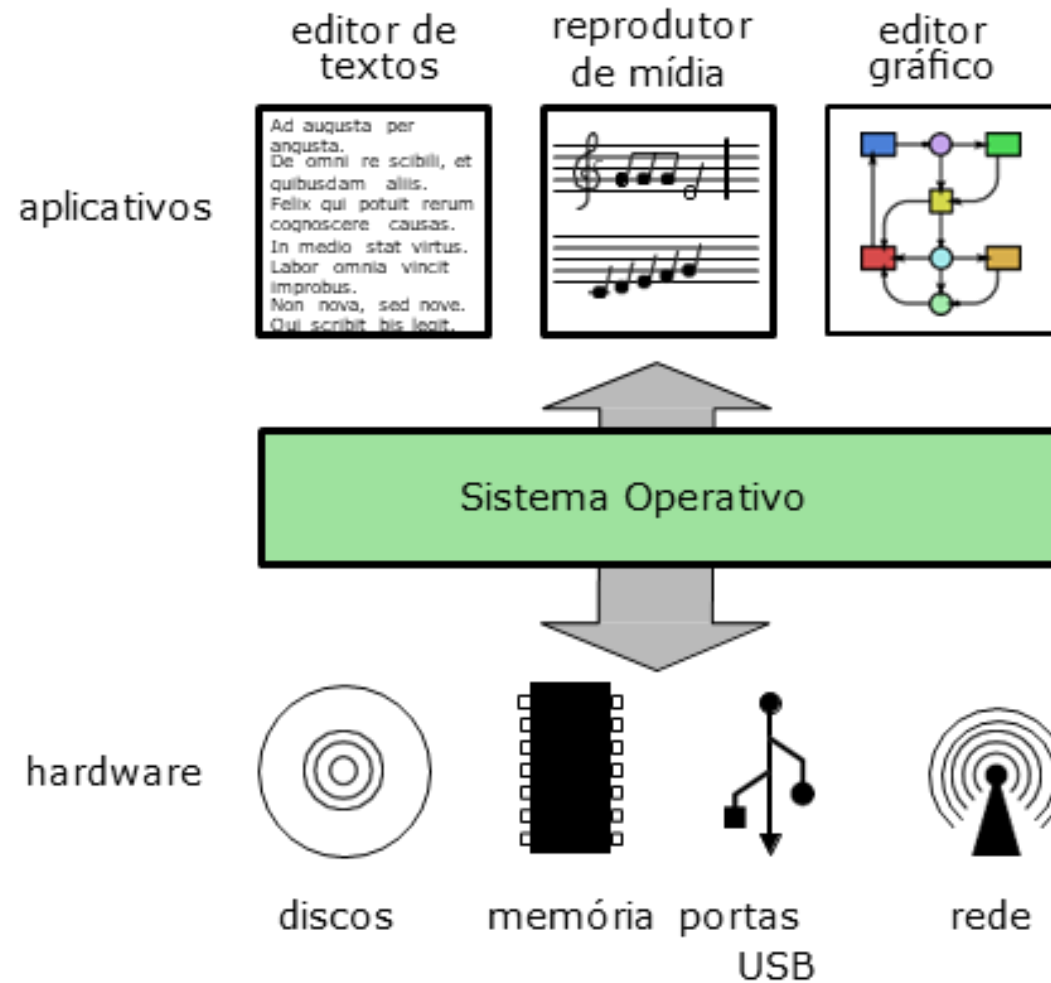
01 - Introdução aos Sistemas Operativos

- Introdução aos Sistemas Operativos:
 - Definição
 - Objectivos
 - Camadas
 - Serviços fornecidos
 - Conceito de kernel
- Evolução dos sistemas operativos

Camada de software que opera entre o hardware e os programas aplicativos.

Estrutura de software ampla e complexa:

- aspetos de baixo nível (drivers)
- aspetos de alto nível (utilitários e interface gráfica)



Abstrair: fornecer interfaces simples e homogêneas

- Simplificar o uso das interfaces de baixo nível
- Tornar os aplicativos independentes do hardware
- Acesso homogêneo a dispositivos com tecnologias distintas

Gerir: coordenar o uso dos recursos pelos programas

- Permitir o uso compartilhado do processador
- Sequenciar acesso a certos recursos (como a impressora)
- Impedir ataques de negação de serviço

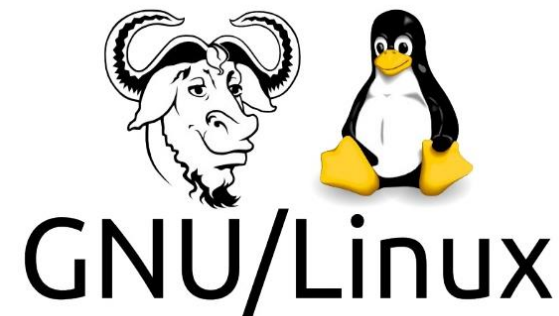
- **Processador**: executar as tarefas dos utilizadores e do sistema
- **Memória**: fornecer áreas de memória isoladas para as aplicações
- **Dispositivos**: configurar e criar abstrações de dispositivos físicos
- **Ficheiros**: criar e manter ficheiros e diretórios (pastas)
- **Proteção**: definir e garantir regras de acesso aos recursos

- Outras áreas: interface gráfica, suporte de rede, multimédia, energia, localização, etc.

- **Batch**: executa tarefas sequenciais (transações, etc.)
- **De rede**: acede a recursos em outros computadores
- **Distribuído**: acede a recursos de forma transparente
- **Multiutilizador** : cada recurso tem um “dono” e regras de acesso
- **Servidor**: gestão eficiente de grandes volumes de recursos
- **Desktop**: interface gráfica e suporte à interatividade
- **Móvel**: gestão de energia, conectividade e sensores
- **Embarcado**: hardware com poucos recursos e energia
- **Tempo real** : tem comportamento temporal previsível (de execução contínua e de resposta a estímulo); pode ser *soft real-time* ou *hard real-time*

Exemplos de sistema operativos

- Sistemas operativos “genéricos”



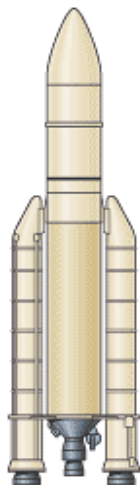
Exemplos de sistema operativos

Mas há mais, muito mais...

Proton
Russia
180 feet
(55 meters)



Ariane 5
European
Space Agency
167 feet
(51 meters)



A Class
(Soyuz-U)
Russia
165 feet
(50 meters)



- Tornar acessível o poder computacional aos utilizadores, controlando o hardware
- Simplificar a utilização dos recursos do sistema
- Criar a ilusão no utilizador que tem a máquina só para si (conceito de máquina virtual)
 - Vários utilizadores podem estar a aceder a uma mesma máquina, mas cada um deles pensará que está sozinho

- Fornecer um conjunto de serviços que simplifique a utilização de um sistema computacional
- Permitir que os recursos do sistema computacional sejam utilizados de forma eficiente
- Permitir o desenvolvimento, teste e introdução de novas funcionalidades sem que estas interfiram com serviços existentes

Camadas de um sistema computacional

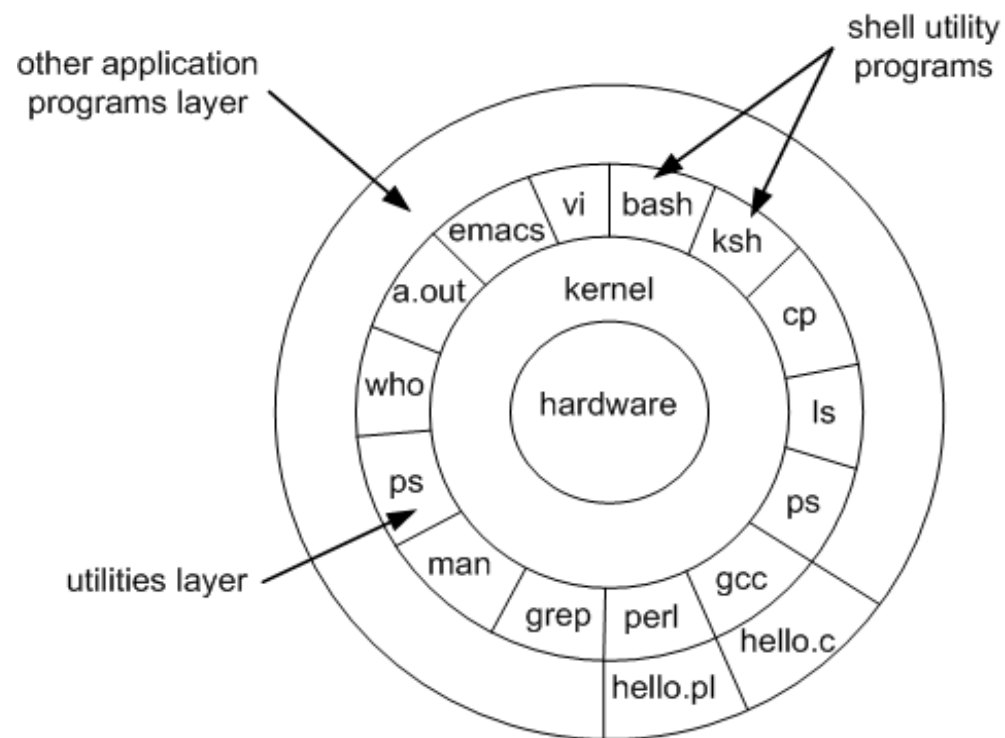
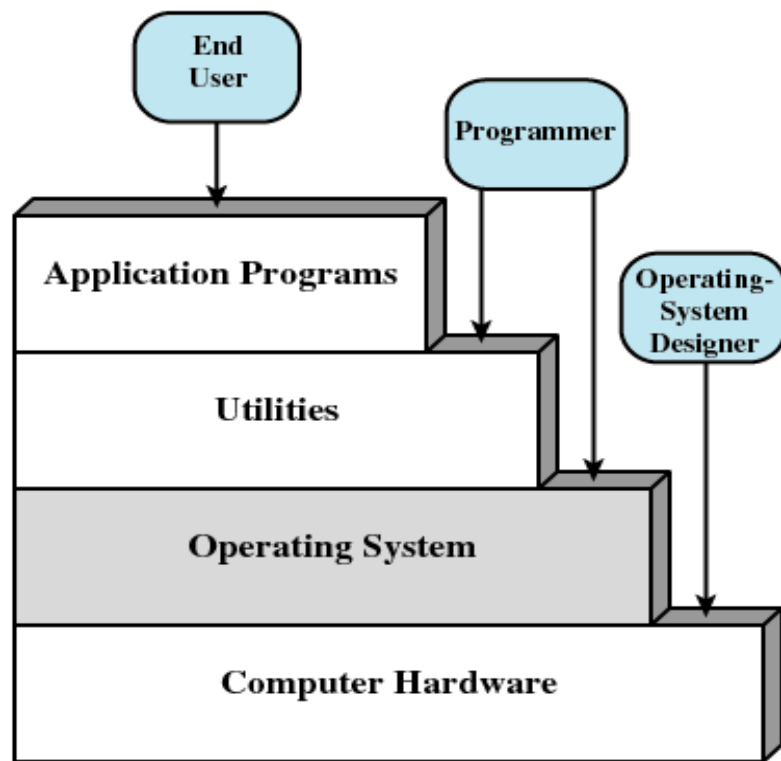
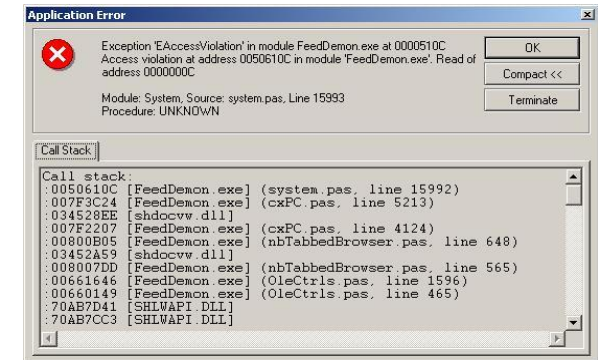


Figure 2.1 Layers and Views of a Computer System

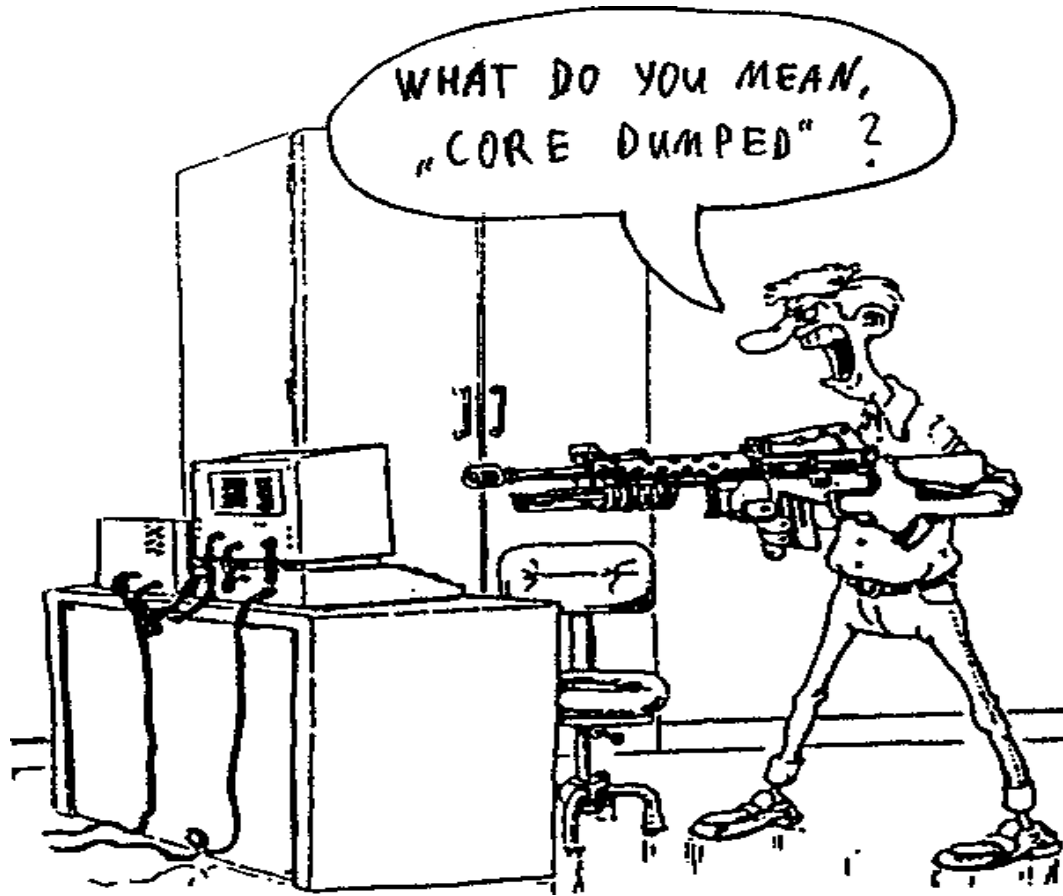
- Desenvolvimento de aplicações
 - Conjunto de ferramentas de desenvolvimento (editores, depuradores, etc)
- Execução de aplicações
 - Carregamento das aplicações e dados para memória e iniciar todo os recursos necessários para executar a aplicação
- Acesso aos dispositivos E/S
 - Interface uniforme para acesso aos diferentes tipos de dispositivos de E/S (disco, placa de rede, rato, teclado, etc)

- Acesso controlado a ficheiros
 - Organização dos dados em disco
 - Controlar o acesso em ambientes multi-utilizador
- Acesso controlado ao sistema
 - Controlar o acesso ao sistema (*login e password*)
 - Protecção de recursos de acessos não autorizados

- Detecção e resposta a erros
 - Erros de *hardware* internos ou externos
 - Erro de memória, falha de um dispositivo E/S
 - Erros de software
 - Transbordos (overflow) aritméticos (divisão por zero, etc)
 - Acesso a endereços proibidos de memória (*segmentation fault - core dumped*)
- Contabilização de recursos
 - Recolher estatísticas sobre utilização
 - Monitorizar o desempenho para possíveis optimizações



O temido “core dump” do Unix...



```
arpum - PuTTY
berkes@arpum:~$ ulimit -c unlimited
berkes@arpum:~$ gcc -o hello -g -Wall hello.c
berkes@arpum:~$ ./hello
Hello world
Segmentation fault (core dumped)
berkes@arpum:~$ ls -l core*
-rw----- 1 berkes  users      65536 Jan 25 17:38 core
berkes@arpum:~$
```

- Comando de cópia de um ficheiro
 - Utilizador: escreve a linha de comando (ou interage com o sistema através de uma interface gráfica)
 - Sistema operativo
 - Valida, interpreta e lança a execução do comando
 - Gestão do sistema de ficheiros
 - Gestão do espaço físico em disco
 - Leitura e escrita de sectores em disco
 - Interligação periférico-sistema
- Concluindo: uma simples acção do utilizador dá origem a muitas (milhentas) acções do sistema operativo

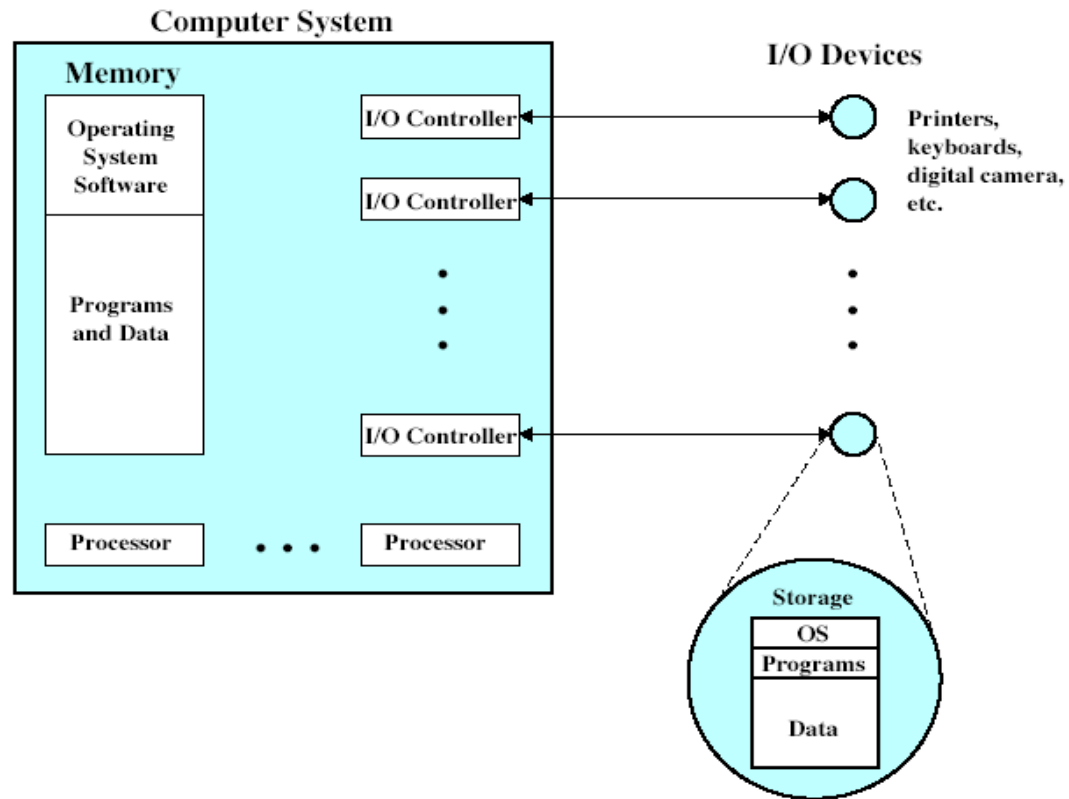


Figure 2.2 The Operating System as Resource Manager

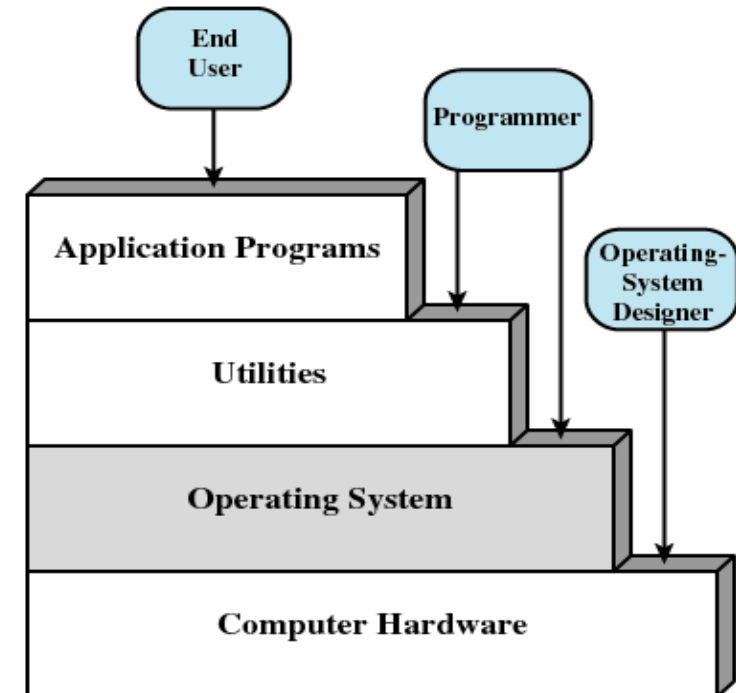


Figure 2.1 Layers and Views of a Computer System

Algumas particularidades:

- O sistema operativo funciona como uma aplicação normal, i.e., é um programa ou conjunto de programas executado pelo processador
- O sistema operativo liberta frequentemente o processador e depende deste para voltar a recuperar o controlo

O que faz um sistema operativo (resumo)?

- Gestor de recursos
 - Gere a alocação de recursos, resolvendo conflitos (e.g. dois “processos” a querer aceder ao disco) de forma **justa** e **eficiente**
- Controlador de programas/processos
 - O sistema operativo controla a execução de programas/processos, gerindo erros e evitando o uso impróprio dos recursos.

- Sistemas iniciais (*Serial processing*)
- Sistemas *batch* simples
- Sistemas *batch* multi-programados
- Sistemas *time sharing*
- Sistemas para computadores pessoais
- Sistemas distribuídos
- Sistemas paralelos
- Sistemas de tempo real

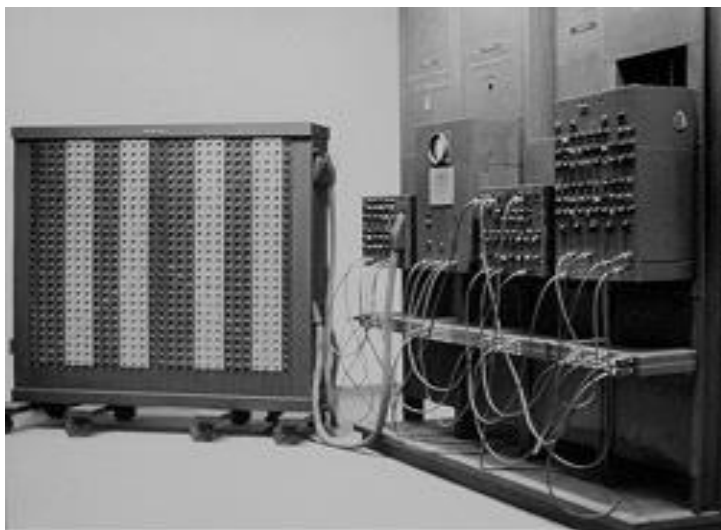
1ª Geração - Era das válvulas (1945-1955)

- Máquinas de grande porte físico, sem sistema operativo
- Operadas a partir de uma consola por um programador especializado
- A programação era feita pelo operador/programador
- A preparação da execução de um programa era complexa e lenta
- Paragem/depuração manual
- Impressão da saída ou armazenamento em fita

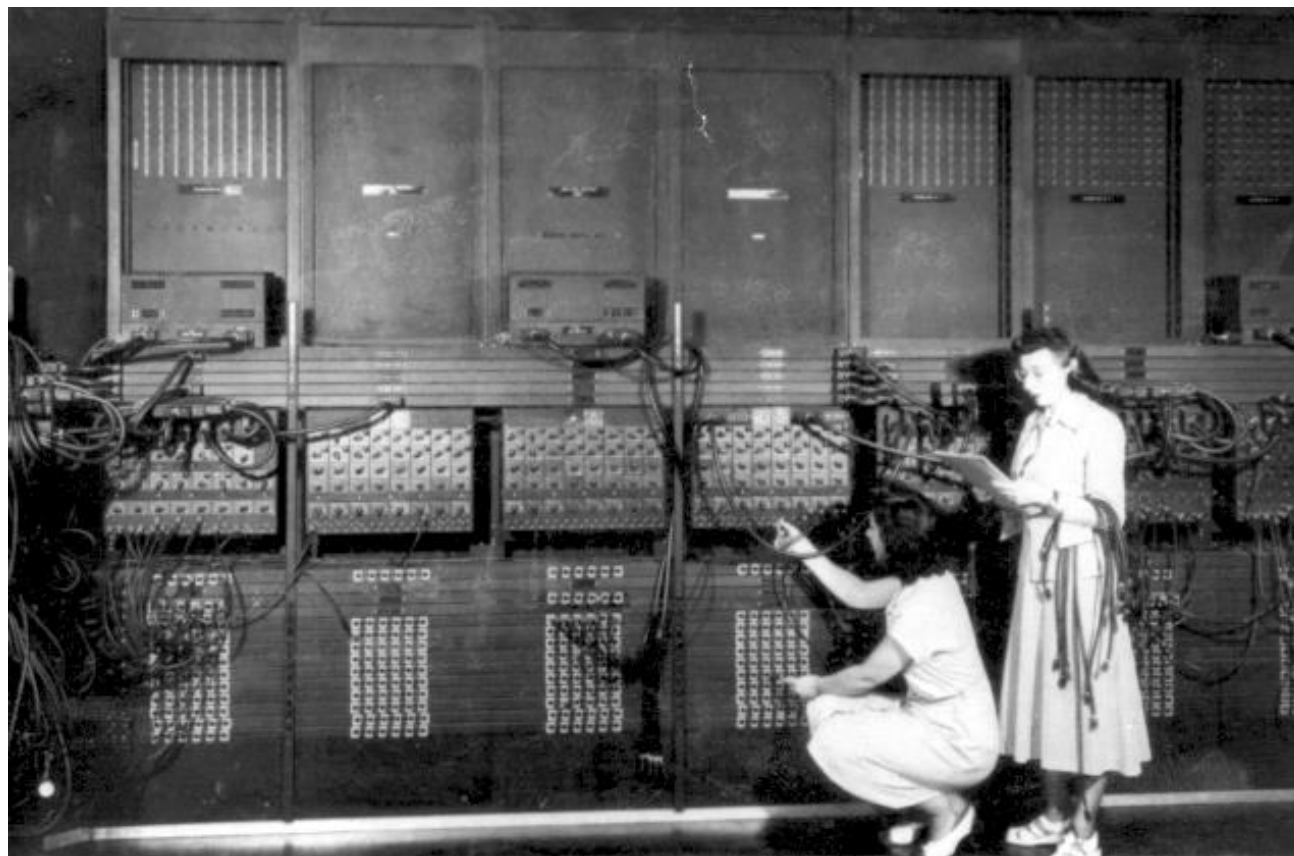
O primeiro computador [1946]

- ENIAC (*Electronic Numerical Integrator and Computer*)
 - Conferência Imprensa: 1 Fevereiro de 1946
 - 5000 adições por segundo, 50 multiplicações
 - 18800 válvulas de 16 tipos, 6000 comutadores
 - 10000 condensadores, 1500 relais e 50000 resistências
 - 3 salas, num total de 72 m²
 - Refrigerado por dois ventiladores movidos por motores 12cv
 - Massa aproximada de 30 toneladas
 - Uma avaria em média todas as 6 horas...

ENIAC (1946-1955)



The ENIAC Today



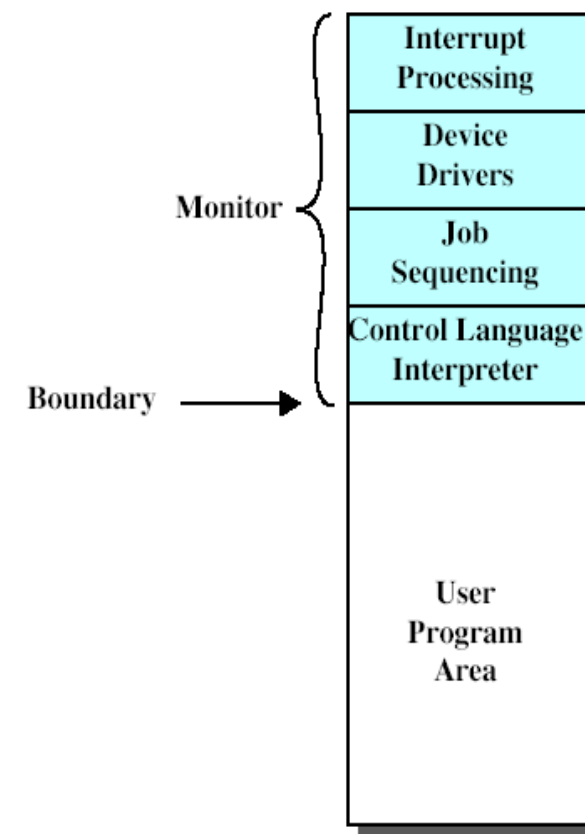
- Principais problemas:
 - Escalonamento: uma folha de papel onde os utilizadores reservavam tempo de CPU (em blocos de 30m)
 - Tempo de *setup*: o tempo de preparação para a execução de um programa (JOB) era demasiado elevado
- Desenvolvimentos posteriores
 - Ferramentas: *Assemblers, loaders e linkers*
 - Impressoras de linhas
 - Unidades de fita magnética
 - Biblioteca de funções
 - Compiladores (FORTRAN, COBOL, etc.)

2ª Geração - Era do transístor (1955-1965)

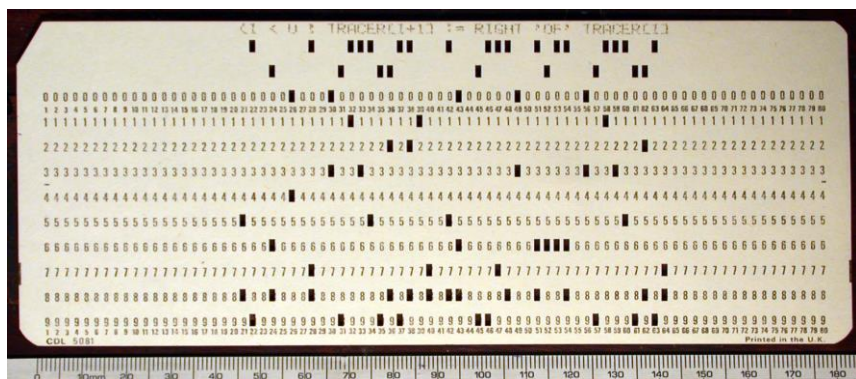
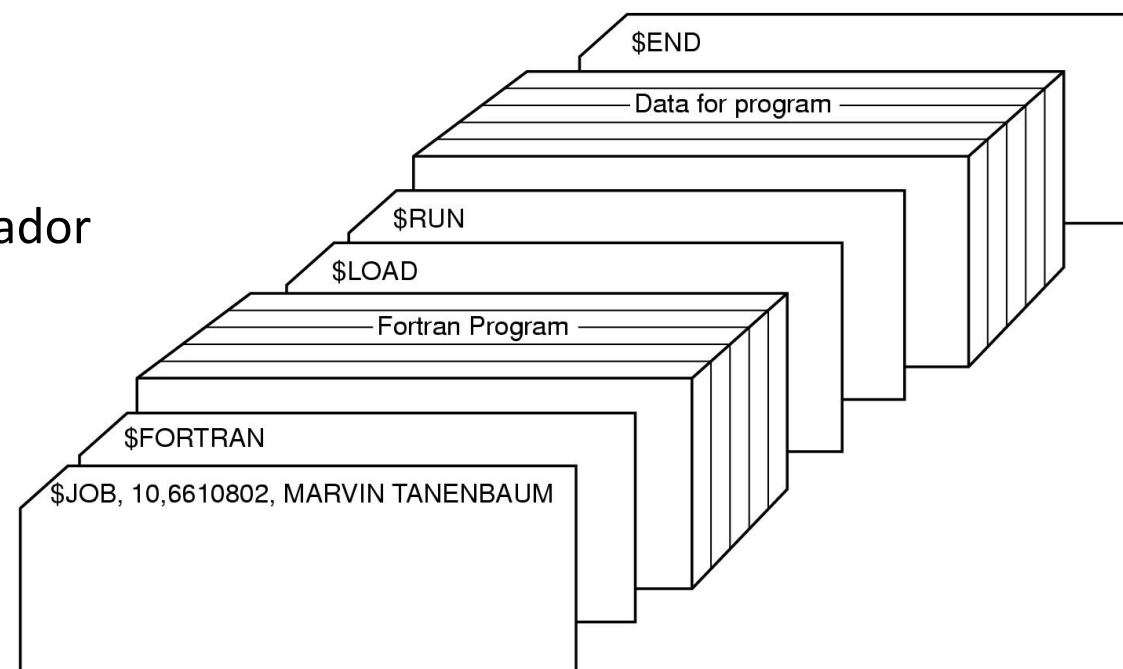
- Tratamento dos programas em grupo (ou em lote), de modo a rentabilizar o sistema
- Principais objectivos:
 - Minimizar o tempo gasto pelo operador em operações de carregamento e controlo
 - Optimizar as operações de entrada/saída
 - Maximizar a utilização do processador
- Conceito chave: monitor

Um pequeno programa (monitor), residente em memória, estava encarregado de:

- Controlar a execução de cada programa
 - Carregar módulos a pedido
 - Transferir o controlo de um programa para o seguinte
-
- O operador deixa de aceder directamente ao hardware



- \$JOB início do trabalho
- \$ASM executar *assembler*
- \$RUN executar o programa do utilizador
- \$END fim do trabalho



Operações de E/S

- As operações de E/S são, normalmente, muito demoradas (em comparação com tempo de ciclo do CPU)
- Se as operações de E/S forem realizadas em série com a execução dos programas isso conduz a um elevado desaproveitamento do CPU
- Alternativas:
 - Processamento *off-line* das entradas/saídas
 - Spooling (Simultaneous Peripheral Operation On-Line)

Processamento *off-line* das entradas/saídas

- Os programas (JOBS) não são lidos directamente dos cartões pelo CPU, mas utilizando periféricos autónomos que os lêem para fita magnética (cartões -> fita magnética)
- A leitura/escrita na fita magnética é relativamente rápida, sendo feita pelo CPU

SPOOL - *Simultaneous Peripheral Operation On-Line*

- Substituição das fitas magnéticas por discos magnéticos, passando-se de acesso sequencial para acesso aleatório
- O disco é usado como buffer de grande dimensões para armazenar a maior quantidade possível de dados de entrada e/ou saída
- Exemplo: *spooler* da impressora (fila de espera)
 - Impressão de um ficheiro leva à cópia do ficheiro para o directório de *spooler* da impressora, sendo a impressão efectuada logo que a impressora esteja livre.

Necessidades de *hardware* desejáveis

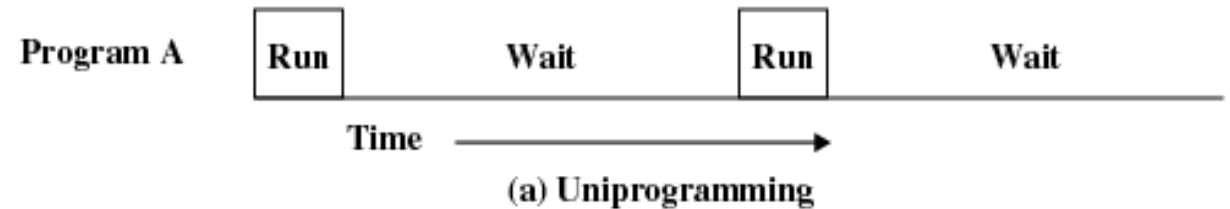
- Protecção de memória
 - Impedir que um programa altere o espaço de memória que contém o monitor
- Temporizador (timer)
 - Evitar que programas monopolizem o processador
- Instruções privilegiadas
 - Permitir que apenas o monitor execute determinado tipo de instruções (e.g. instruções de I/O)
- Interrupções
 - Possibilitar que o monitor (sistema operativo) liberte o processador

Protecção de memória + instruções privilegiadas permitem diferenciar 2 modos de operação:

- System/kernel mode: modo em que o sistema operativo é executado
- User mode: modo onde são executados os programas do utilizador
 - Não permite a execução de instruções privilegiadas
 - Não permite o acesso a determinadas zonas de memória (protegidas)

- Problemas resolvidos:
 - Execução sequencial de programas é automática
- Problemas por resolver:
 - As operações de E/S são muito lentas comparadas com a velocidade do processador
 - A maior parte do tempo o processador não está a realizar trabalho útil (*idle*)

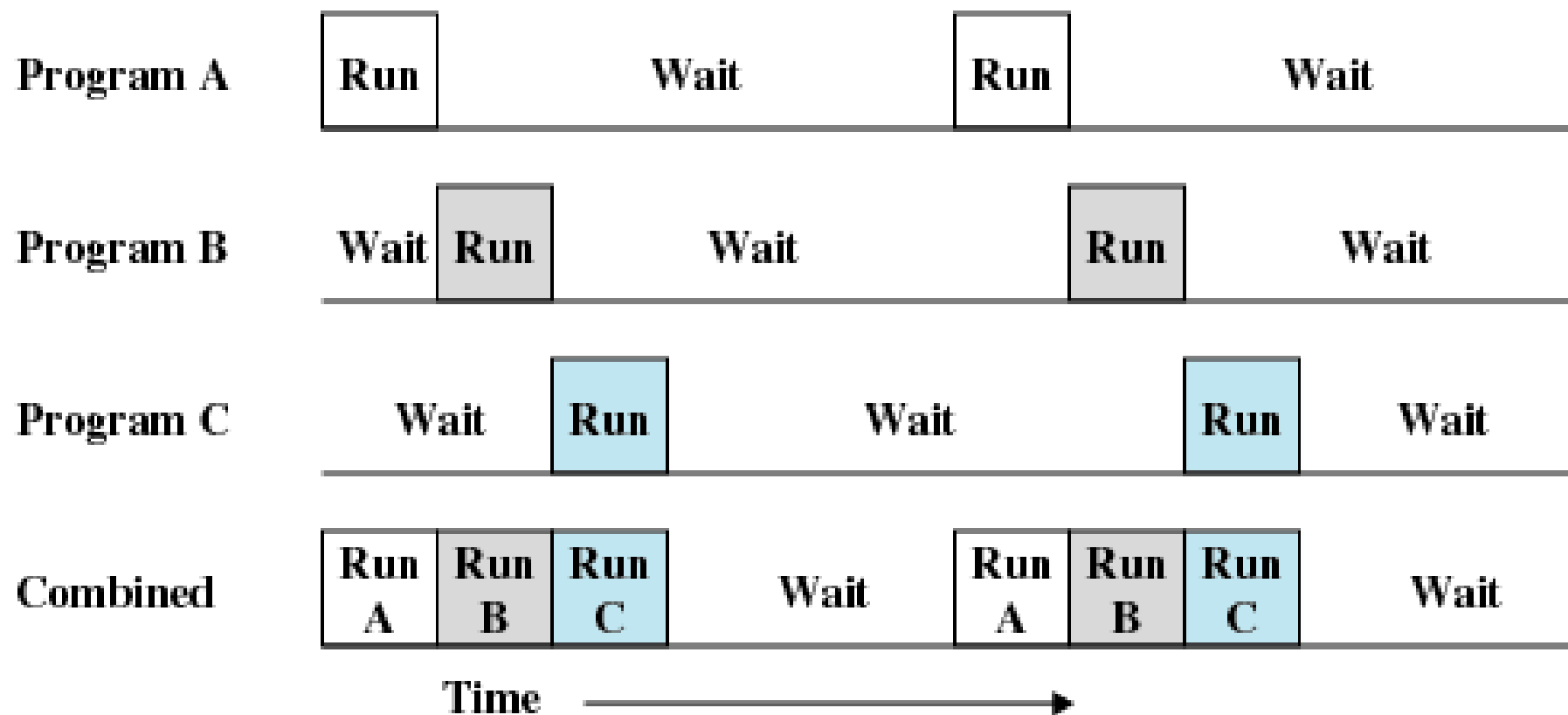
Read one record from file	15 μ s
Execute 100 instructions	1 μ s
Write one record to file	15 μ s
TOTAL	31 μ s
Percent CPU Utilization = $\frac{1}{31} = 0.032 = 3.2\%$	



3ª Geração - Era do circuito integrado (1965-1980)

- A maior parte do tempo as tarefas estão bloqueadas à espera que a operação de E/S termine ou que um recurso fique disponível
- Os sistemas *batch* multi-programados tiram partido da existência de vários programas em disco, carregando vários deles em memória e comutando a execução entre eles
- O SO toma decisões de escalonamento dos programas em função de vários factores:
 - Recursos pretendidos, disponíveis, utilizados, etc.

Sistemas *batch* multi-programados



(c) Multiprogramming with three programs

- Definição mais abrangente:
 - A multi-programação (*multiprogramming*) ou multi-tarefa (*multitasking*) consiste na execução concorrente de vários programas a fim de otimizar a utilização do CPU
- Necessidades de hardware desejáveis
 - Interrupções E/S e possivelmente DMA
 - Permitir a execução de instruções quando o dispositivo de E/S está ocupado
 - Gestão de memória
 - Várias tarefas prontas a serem executadas tem que ser mantidas em memória

- Utilização da multi-programação para suportar várias tarefas interactivas (sistemas *batch* não suportam tarefas interactivas)
 - Ex: sistemas de reserva de bilhetes nas companhias de aviação
- O tempo de CPU é partilhado (*time sharing*) entre os vários utilizadores
 - O escalonamento entre tarefas tem de ser suficientemente rápido para que o utilizador tenha a noção de ter a máquina só para si
- Vários utilizadores acedem ao sistema através de terminais (texto ou gráficos)

Multi-programados vs. *time sharing*

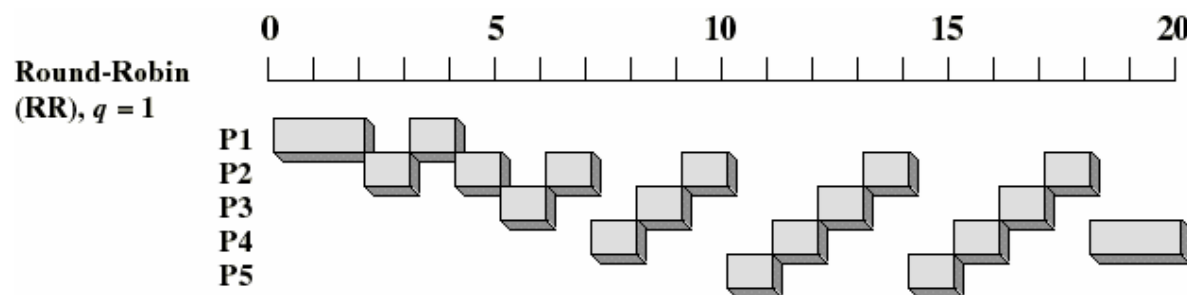
Time sharing – extensão da multi-programação em que o tempo de CPU é dividido (*time sliced*) por cada programa do utilizador

- Se existirem n programas, cada programa recebe $1/n$ de tempo CPU

	Batch Multiprogramming	Time Sharing
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

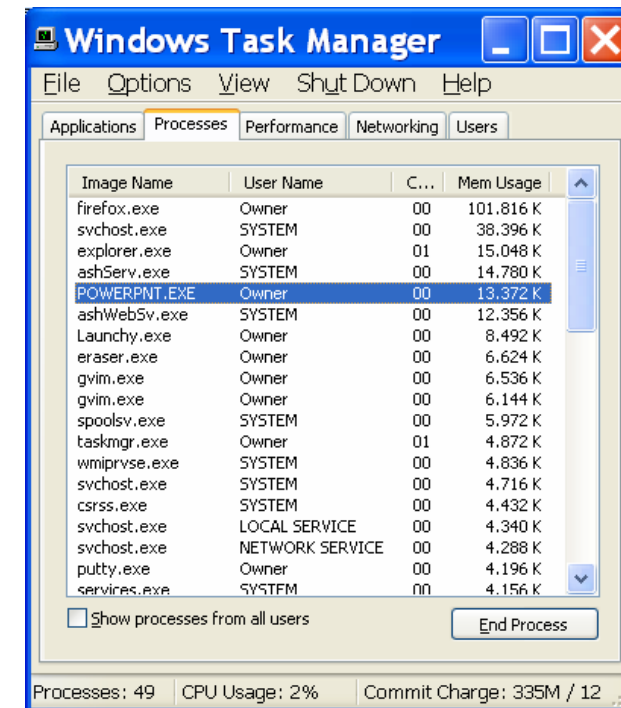
Multitarefa é uma extensão da multiprogramação

- A execução do processo realiza-se em “fatias temporais” (time-slicing)
- Cada processo executa por um curto período de tempo (e.g. 100 ms), é suspenso e outro é executado
- Quando o processo em execução é suspenso (temporariamente) pelo S.O. para que outro possa executar, diz-se que houve “preempção”
- Os S.O. com capacidade de “preempção” são ditos “preemptivos”



Implementação do “multitarefa preemptivo”

- Interrupção do relógio de tempo real (RTC – real time clock)
 - Interrompe o CPU todos os 10 milissegundos
 - Como é o S.O. que processa a interrupção, o próprio S.O. pode proceder à comutação de tarefa/processo



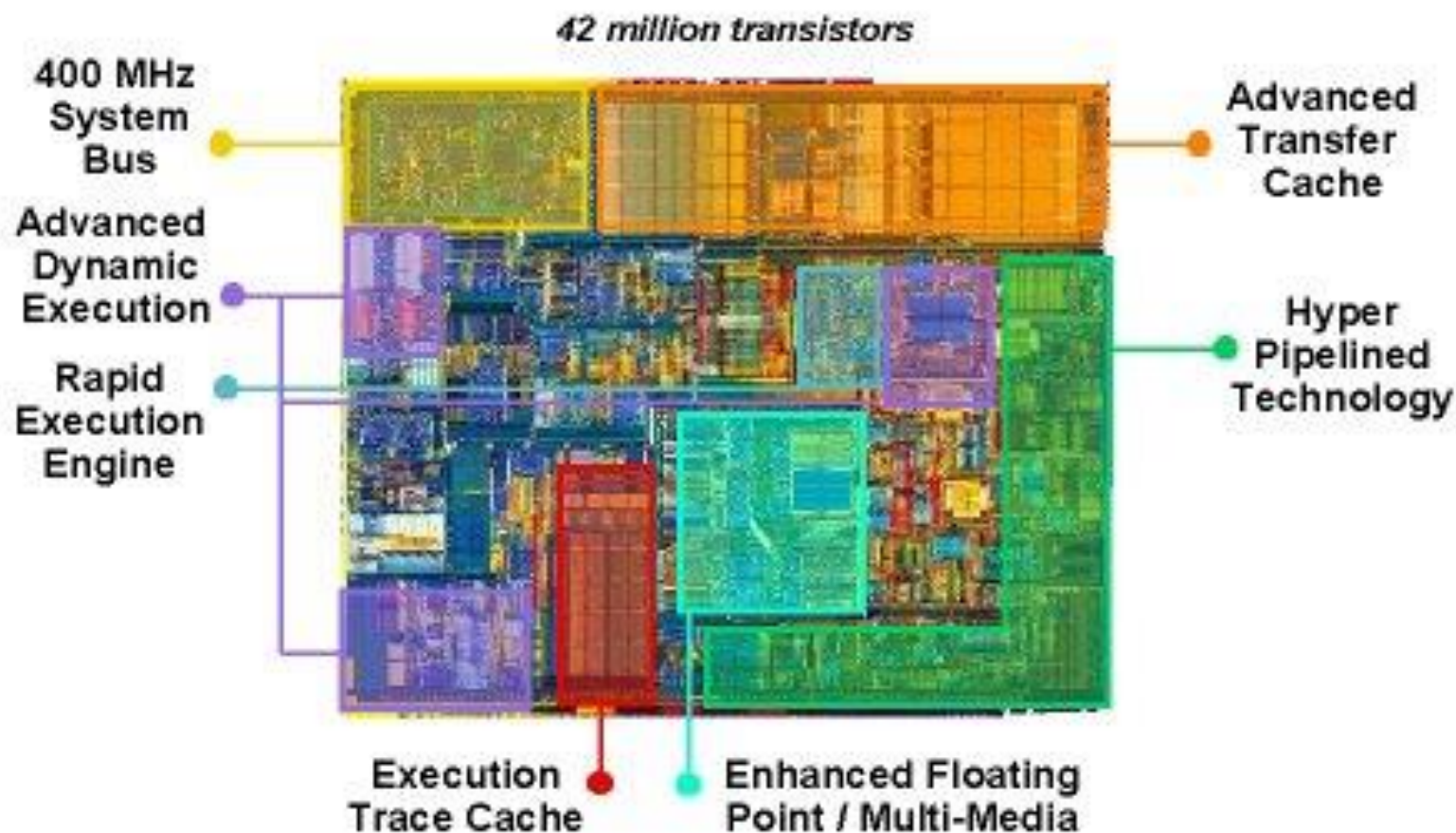
```
1 [ 0.0%] Tasks: 4, 0 thr; 1 running
2 [ 0.0%] Load average: 0.07 0.10 0.14
3 [ 0.0%] Uptime: 01:09:57
4 [ 0.0%]
Mem[|||||] 320M/7.79G
Swp[ ] 0K/1024M
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	208	4	0	S	0.0	0.0	0:00.03	/usr/local/bin/dumb-init -- php server_preemptive.php
8	root	20	0	218M	35540	30072	S	0.0	0.4	0:00.03	php server_preemptive.php
9	root	20	0	18132	3200	2712	S	0.0	0.0	0:00.03	/bin/bash
394	root	20	0	23320	3832	2732	R	0.7	0.0	0:00.02	htop

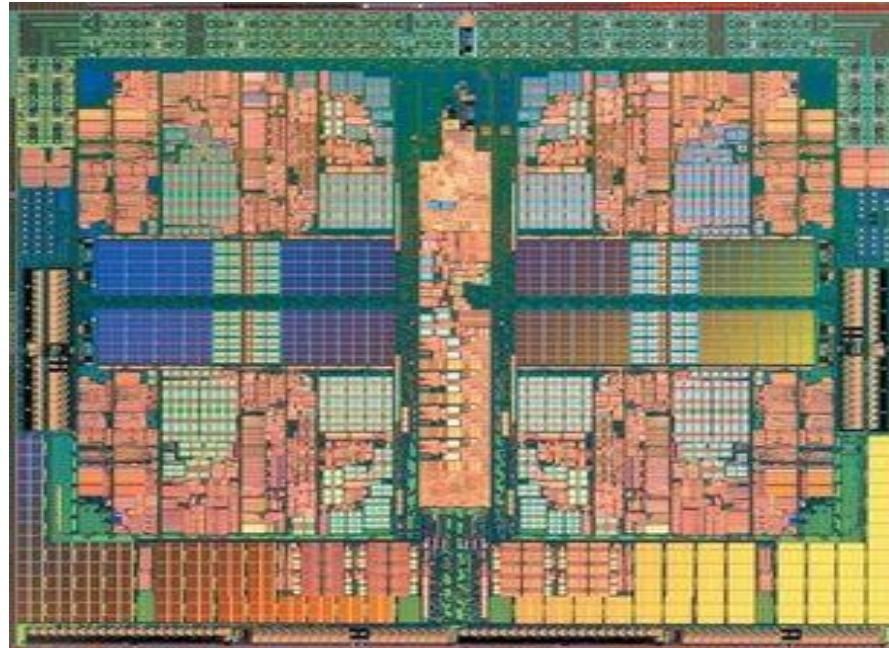
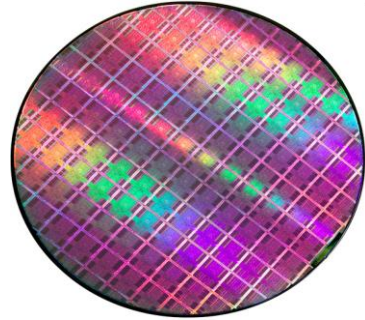
Era dos micro-processadores (1980 – Presente)

- Sistemas para computadores pessoais
 - Decréscimo do custo do hardware
 - Sistemas dedicados a um único utilizador (PCs)
 - Migração dos conceitos e características dos S. O. das máquinas de grande porte para os PCs
 - Desenvolvimento de pequenos computadores com grandes capacidades de processamento, rapidez e capacidade de armazenamento
 - Estações de trabalho (*Workstations*)

Pentium IV: 42 M transístores, até 3.6 GHz



- AMD quad-core: Phenom
 - 463 M transistores (10x do que o Pentium 4!)
 - Cache nível 3 (L3) de 2MB partilhada entre os cores

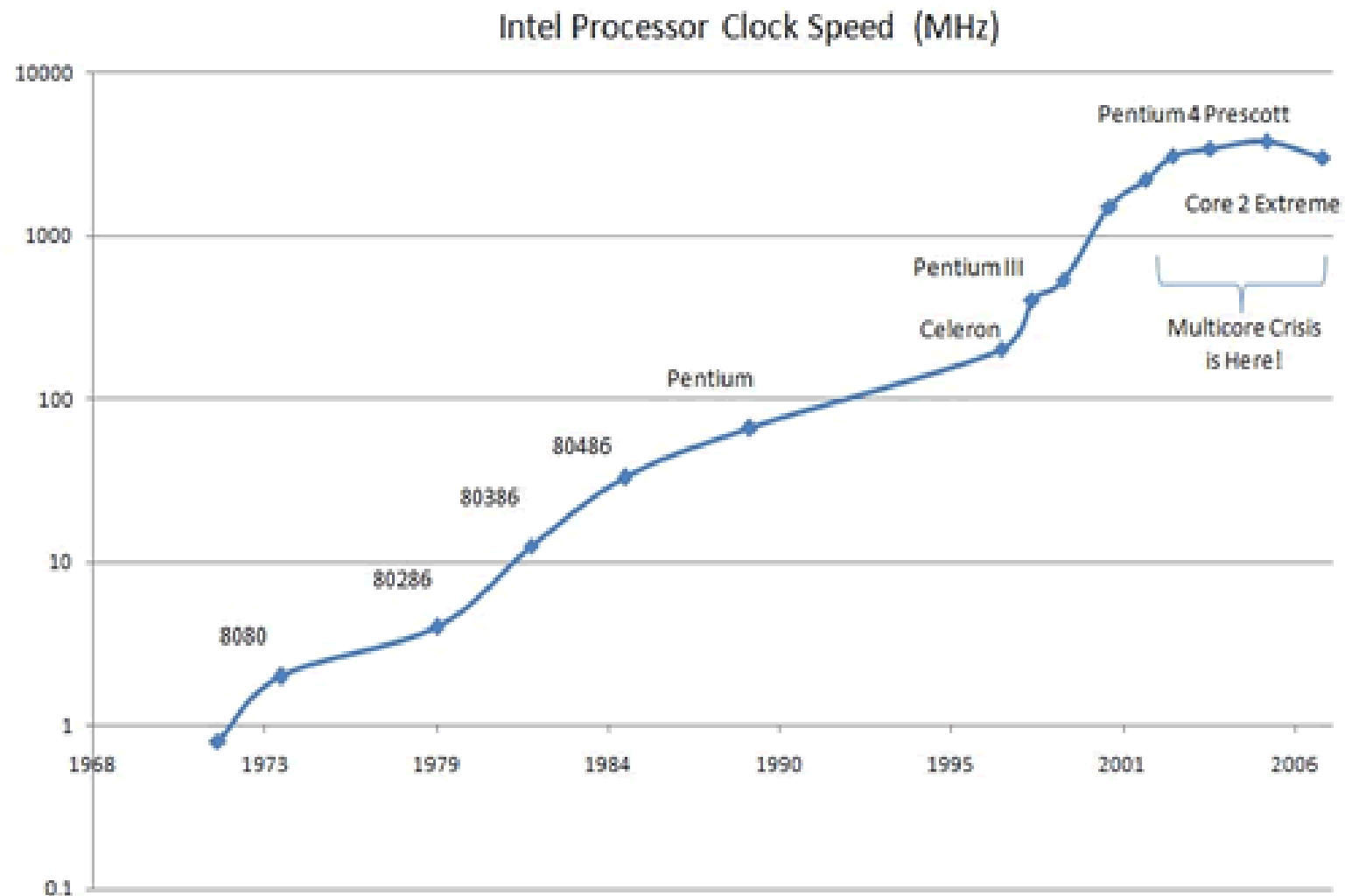


A corrida ao Ghz... o fim?

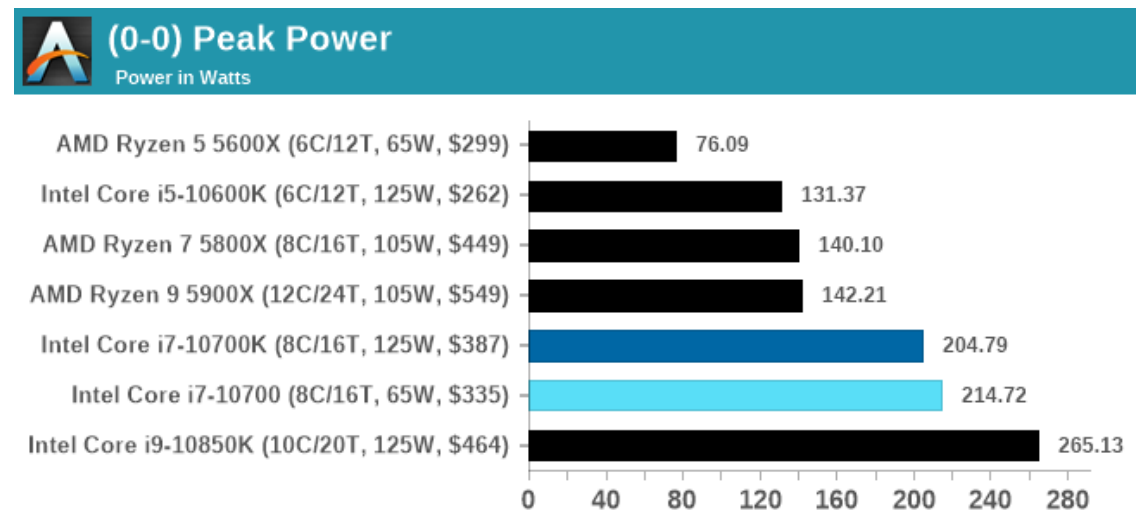
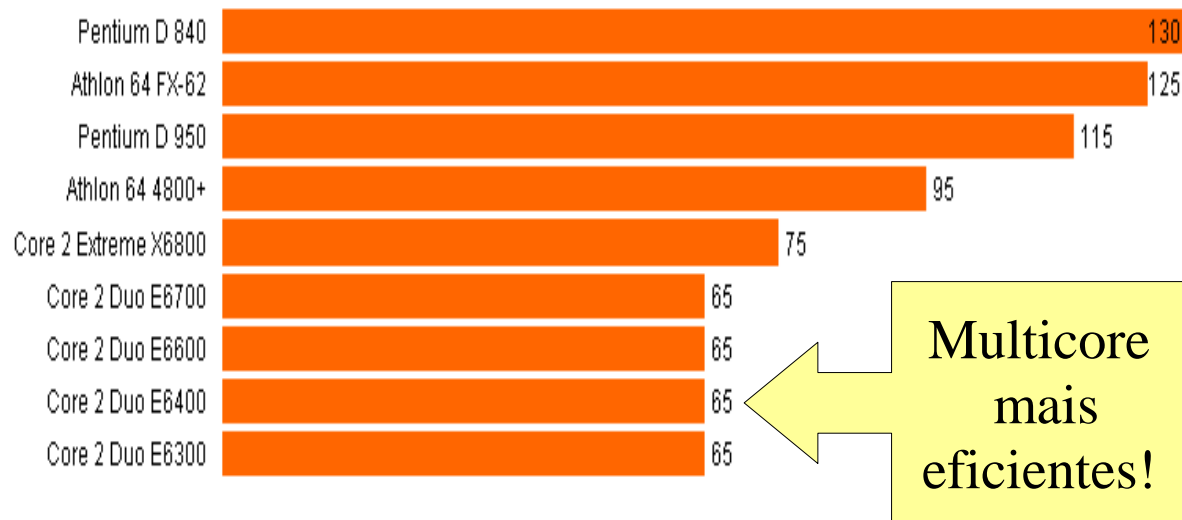
- Desde +/- 2004 a velocidade de relógio dos CPU X86 parou de subir...
 - PII até 350 Mhz (1999)
 - PIII até 1.0 GHz (2002)
 - P4 até 3.6 Ghz (2004), mas ainda se mantém em 3.6GHz
- Velocidade de relógio deixou de trazer dividendos
 - Velocidade da luz: 30 cm/nanosegundo ($1 \text{ ns} \leftrightarrow 1 \text{ GHz}$)
 - Velocidade da luz no cobre: 9 cm/nanosegundo
- Consumo eléctrico CPU dispara...
- ...necessidade de arrefecimento



A corrida ao GHz...



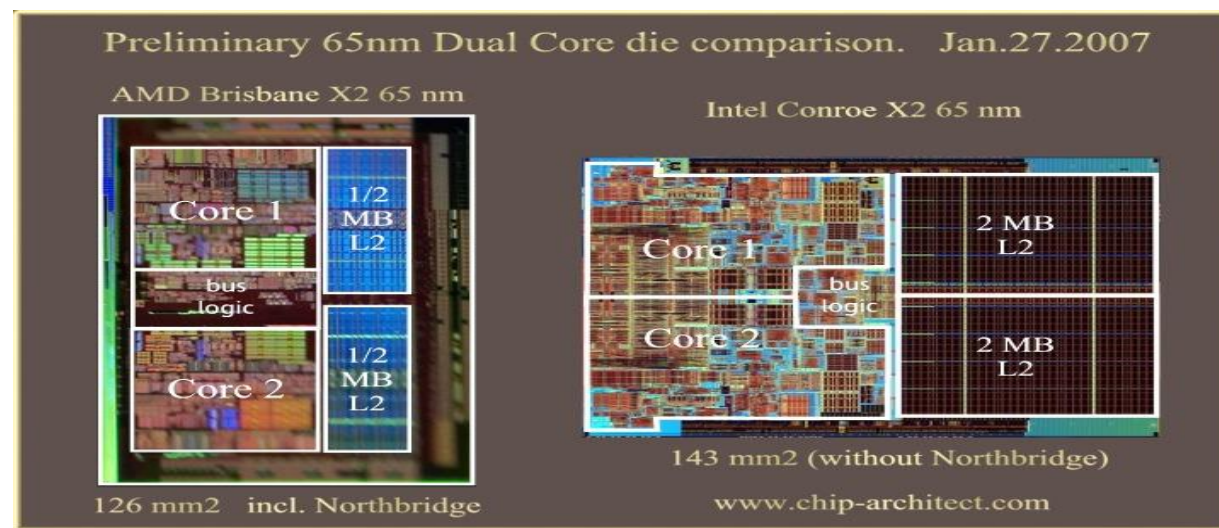
Consumo eléctrico máximo (watt)



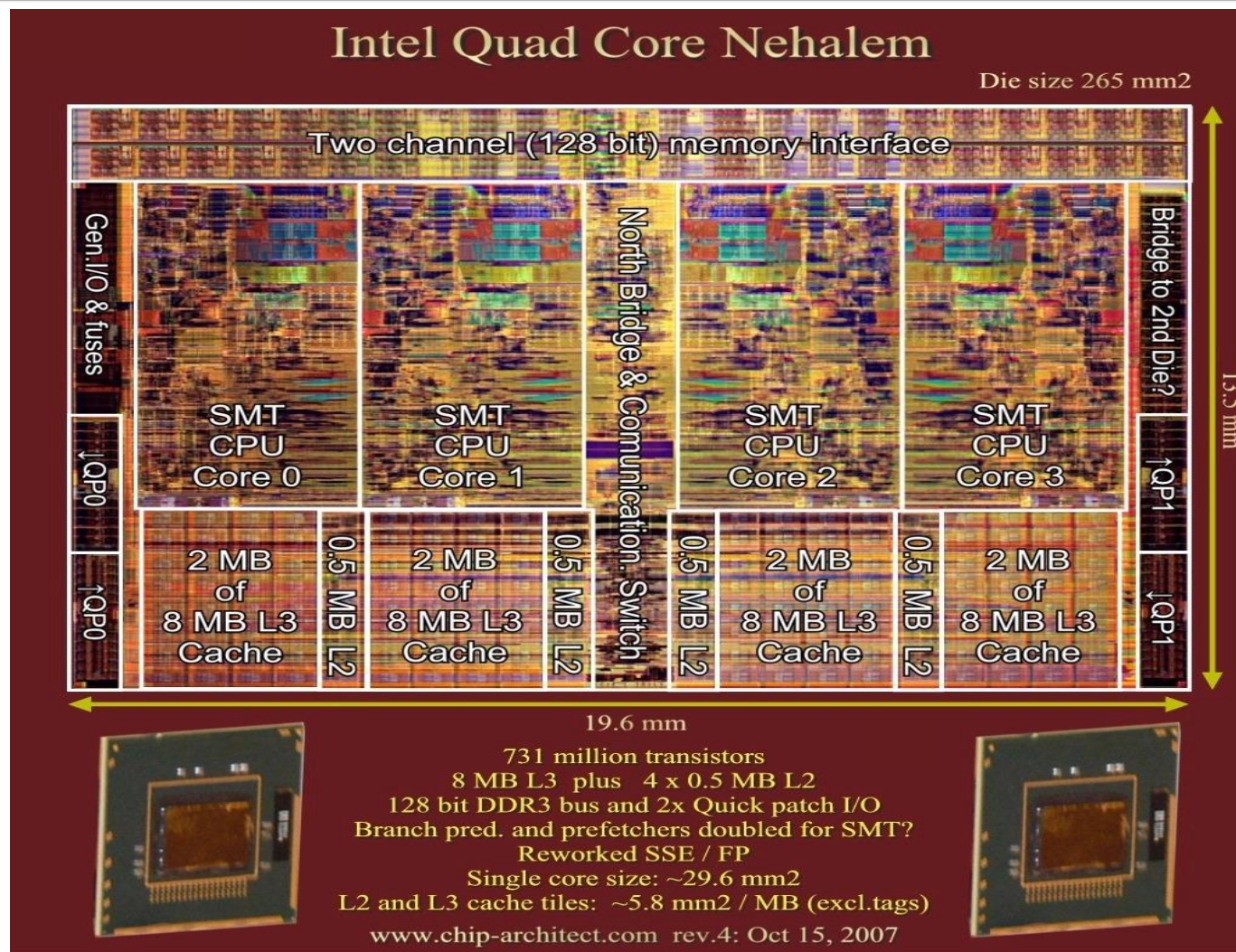
A corrida ao Ghz – possível solução

- Vários CPUs dentro do CPU: multicore
- Maior necessidade de programação concorrente!
- A ler: “The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software”, Herb Sutter

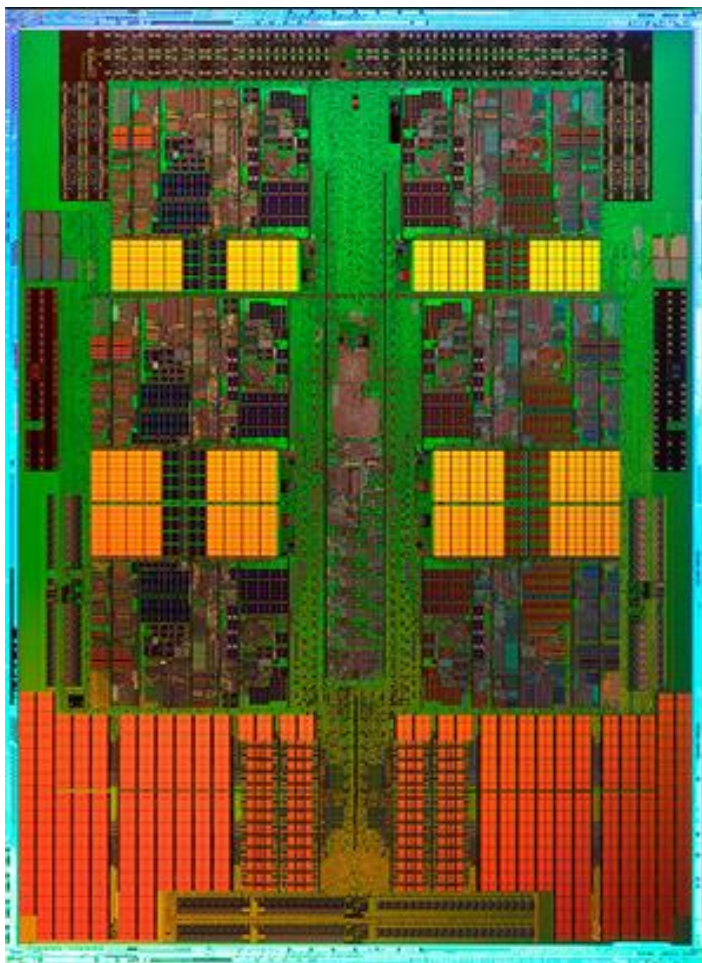
(<http://www.gotw.ca/publications/concurrency-ddj.htm>)



A corrida ao GHz – possível solução



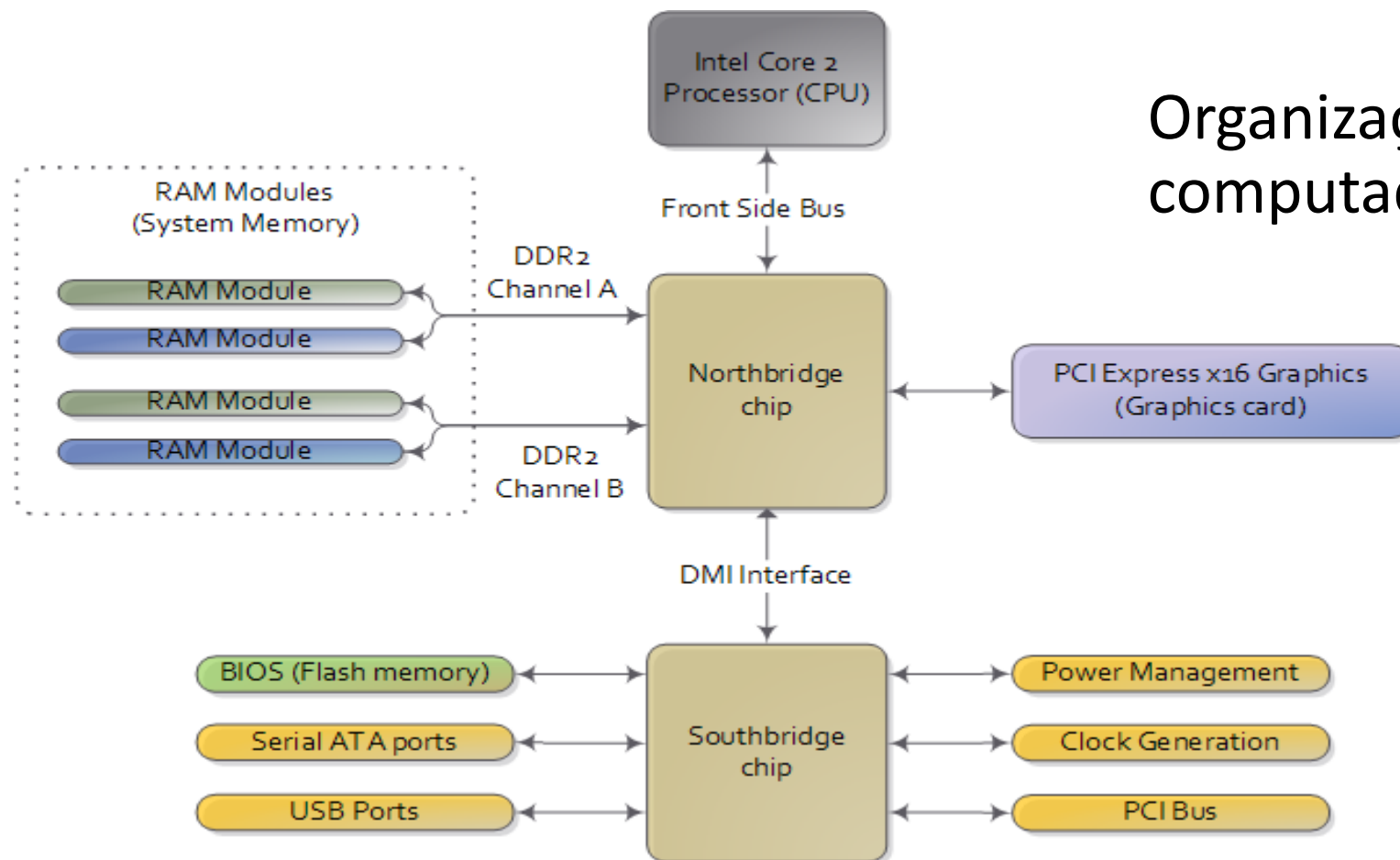
AMD sixcore Istanbul



- L3-Cache: 6 MB, shared
- Clockrate: 2200–2800 MHz
- HyperTransport 3.0
- HT-Assist

Computador pessoal (PC)

Organização interna de um computador pessoal



<http://static.duartes.org/img/blogPosts/motherboardDiagram.png>

- A computação é distribuída entre vários processadores residentes em computadores distintos (*sites, nodes*)
- A comunicação entre processos é feita usando canais ou redes de alta velocidade
- Os processadores não partilham o mesmo relógio
- Razões para o surgimento de sistemas distribuídos
 - Partilha de recursos
 - Aumento da capacidade de computação
 - Fiabilidade
 - Comunicação entre utilizadores/programas
 - Elevado rácio desempenho/preço


Exemplos

- Internet (*World Wide Web*) – maior sistema distribuído!
- Jogos cooperativos: Doom, Quake 😊
- Sistema Multibanco
- *Tracking* de encomendas via *Web* (e.g. Federal Express, etc.)
- Cadeia de supermercados (vendas → abastecimento de stocks, etc.)
- Telemóveis
- Reserva de bilhetes (aviões, comboios, etc.)
- ...

Sistemas distribuído

- Vários computadores “independentes” ligados por rede
 - Usualmente redes dedicadas de alta velocidade
 - Clusters
 - Ultimamente, surgiu conceito de desktop grid
 - Muitos (vária centena de milhares de computadores ligados através da Internet)
 - Exemplo: [SETI@HOME](#), [Einstein@Home](#), ...
 - Uso da plataforma de computação “BOINC”



	Total	Active
Users	191,323	39,345
Hosts	453,915	73,020
Teams	7,795	2,909
Countries	204	154
 The Einstein@Home logo, featuring a colorful square (blue, green, yellow, red) and the text "Einstein@Home".		
Total Credit		5,957,553,117
Recent average credit RAC		8,726,811
Average floating point operations per second	87,268.1 GigaFLOPS / 87.268 TeraFLOPS	

Ainda sobre desktop grids...



Folding@home
distributed computing

Client statistics by OS

OS Type	Native TFLOPS*	x86 TFLOPS*	Active CPUs	Total CPUs
Windows	300	300	315451	3134901
Mac OS X/PowerPC	4	4	5014	134463
Mac OS X/Intel	30	30	9692	113276
Linux	79	79	46508	474253
ATI GPU	1187	1252	11637	120955
NVIDIA GPU	1287	2716	10813	172637
PLAYSTATION®3	1062	2241	37657	930041
Total	3949	6622	436772	5080526



GPUs

<http://fah-web.stanford.edu/cgi-bin/main.py?qtype=osstats>

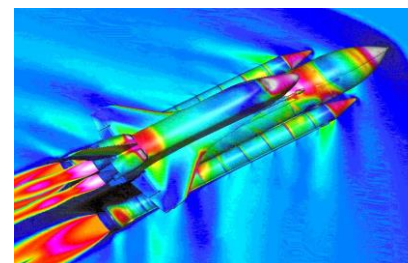
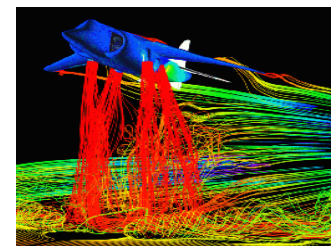
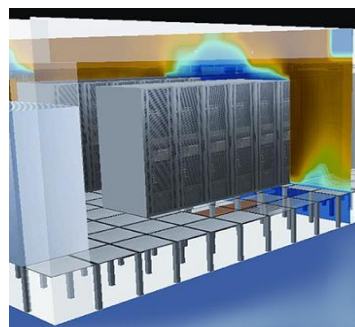
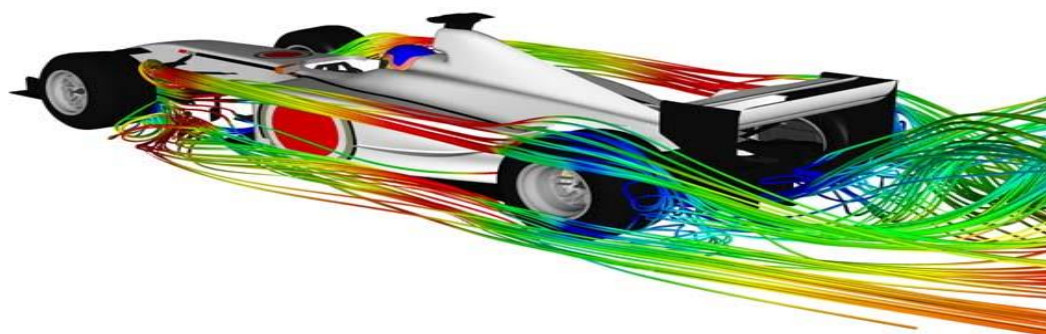
Sistemas distribuído

- Clusters



Aplicação prática da computação distribuída (ou paralela)

- Sistema de pesquisa de impressões digitais do FBI emprega *clusters beowulf-linux* para a identificação de impressões digitais
 - O mesmo sucede para base de dados de DNA (CODIS, etc.)
- *Rendering* de imagens (e.g. Imagens do filme Titanic, especialmente as imagens envolvendo água)
- Mapeamento do genoma humano
- Previsões meteorológicas
- etc



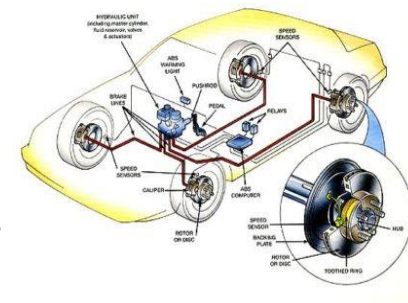
- Usados quando é necessário garantir uma relação entre o tempo cronológico exterior e o tempo de execução dos programas
 - O sistema tem que produzir resultados certos no tempo certo!
 - Exemplo: “Engine Controll Unit” (ECU” dos motores de automóveis)
 - Controla, entre outros elementos, a quantidade de combustível injectada nos cilindros, a temporização da ignição e das válvulas (abertura/fecho)
- Tarefas ou processos tentam controlar ou reagir a eventos que ocorrem no “exterior”. Esses eventos ocorrem em tempo real...



- “A real-time system is one in which the correctness of the computations not only depends upon the logical correctness of the computation, but also upon the time at which the result is produced. If the timing constraints are not met, system failure is said to have occurred.”
 - Fonte: Real-time Computing FAQ
- Conceito de “latência”:
 - Na computação, designa-se por latência o tempo que medeia entre um evento e a resposta a esse mesmo evento

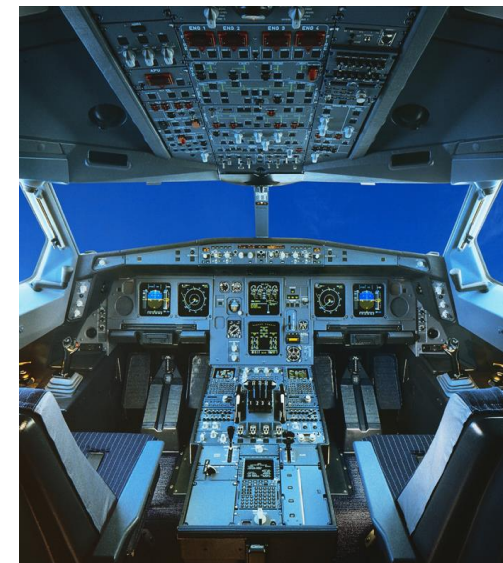
Sistemas de tempo real dividem-se em:

- Sistemas de *hard realtime* – garantem a execução das tarefas críticas dentro dos limites de tempo estabelecidos
 - Exemplo: sistema ABS de uma viatura
 - O sistema deve actuar por forma a que as rodas não bloqueiem
 - Se as rodas bloquear é porque o sistema não actuou no prazo estabelecido
- Sistemas de *soft realtime* – apenas é dada prioridade às tarefas críticas, não sendo garantida a sua execução dentro de limites de tempo estritos
 - Muitas vezes, o “soft” realtime é especificado através de probabilidades
 - Exemplo: o sistema garante com probabilidade p que a resposta a determinado evento ocorrerá num prazo T



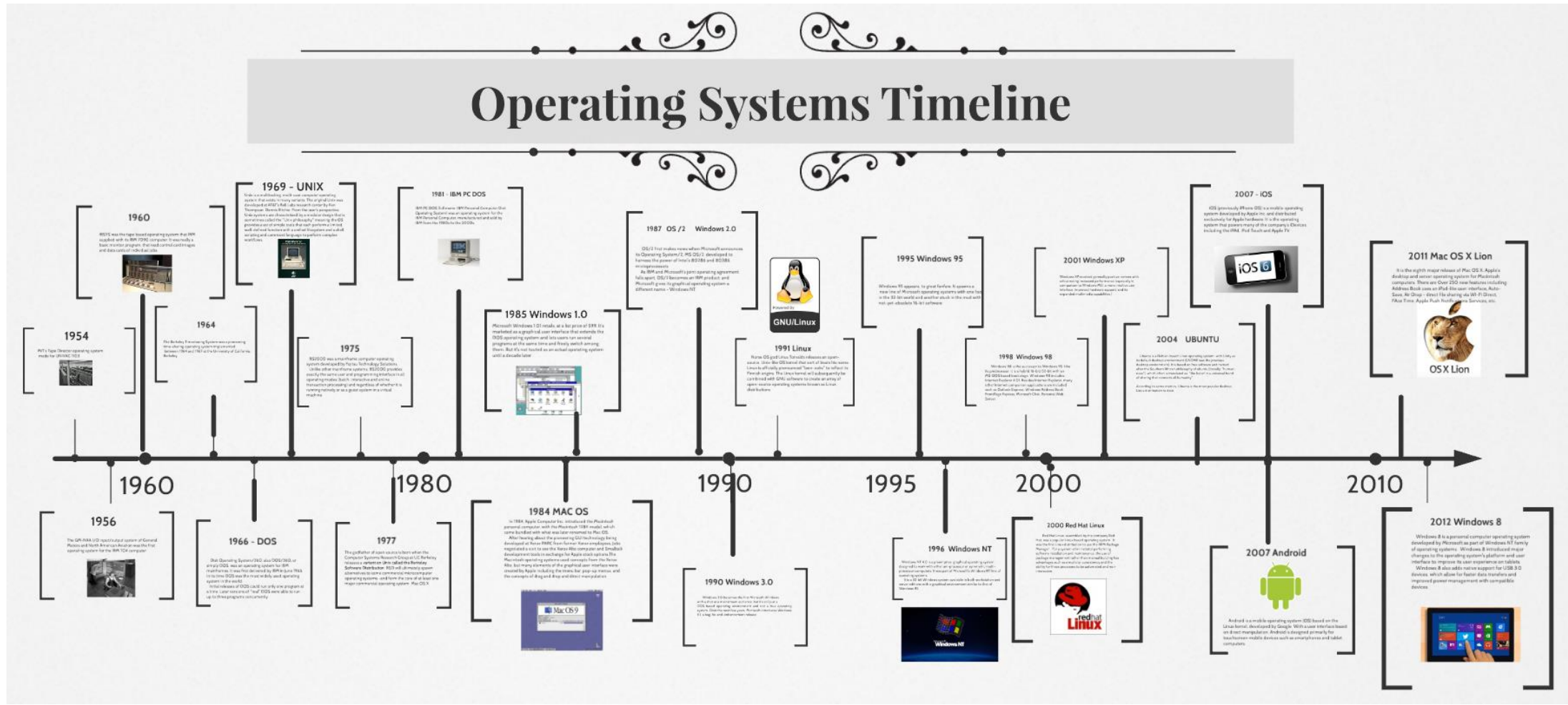
Áreas de aplicação

- Sistemas de suporte de vida (“e-care”)
- Sistemas de controlo industrial
- Sistemas de comunicações
- Sistemas de controlo de tráfego aéreo, aeronaves
- Sistemas de controlo de motores
- Robótica, exploração espacial
- Etc.

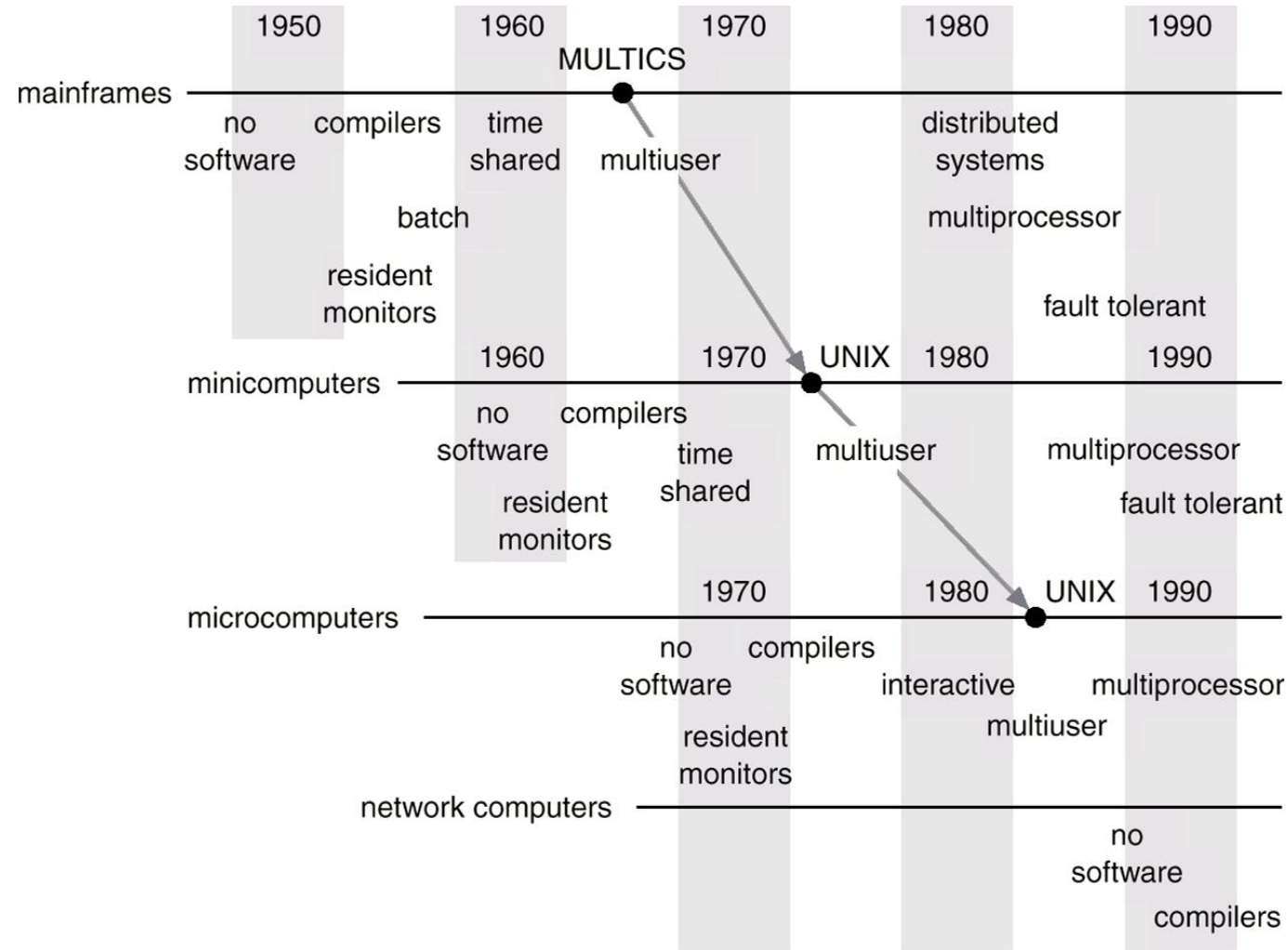


- Anos 40: programa executavam sozinhos na máquina
- Anos 50: “bibliotecas de sistema” e monitores de execução
- 1961: CTSS – *Compatible Time-Sharing System*, no MIT
- 1965: IBM OS/360, com *time sharing* e suporte a discos
- 1965: *Multics*, por MIT, GE e Bell Labs
- 1969: UNIX, por Thompson e Ritchie, Bell Labs
- 1981: MS-DOS, Microsoft
- 1984: Mac OS 1.0, com interface gráfica
- 1985: MS-Windows 1.0
- 1987: Andrew Tanenbaum desenvolve o *Minix*.

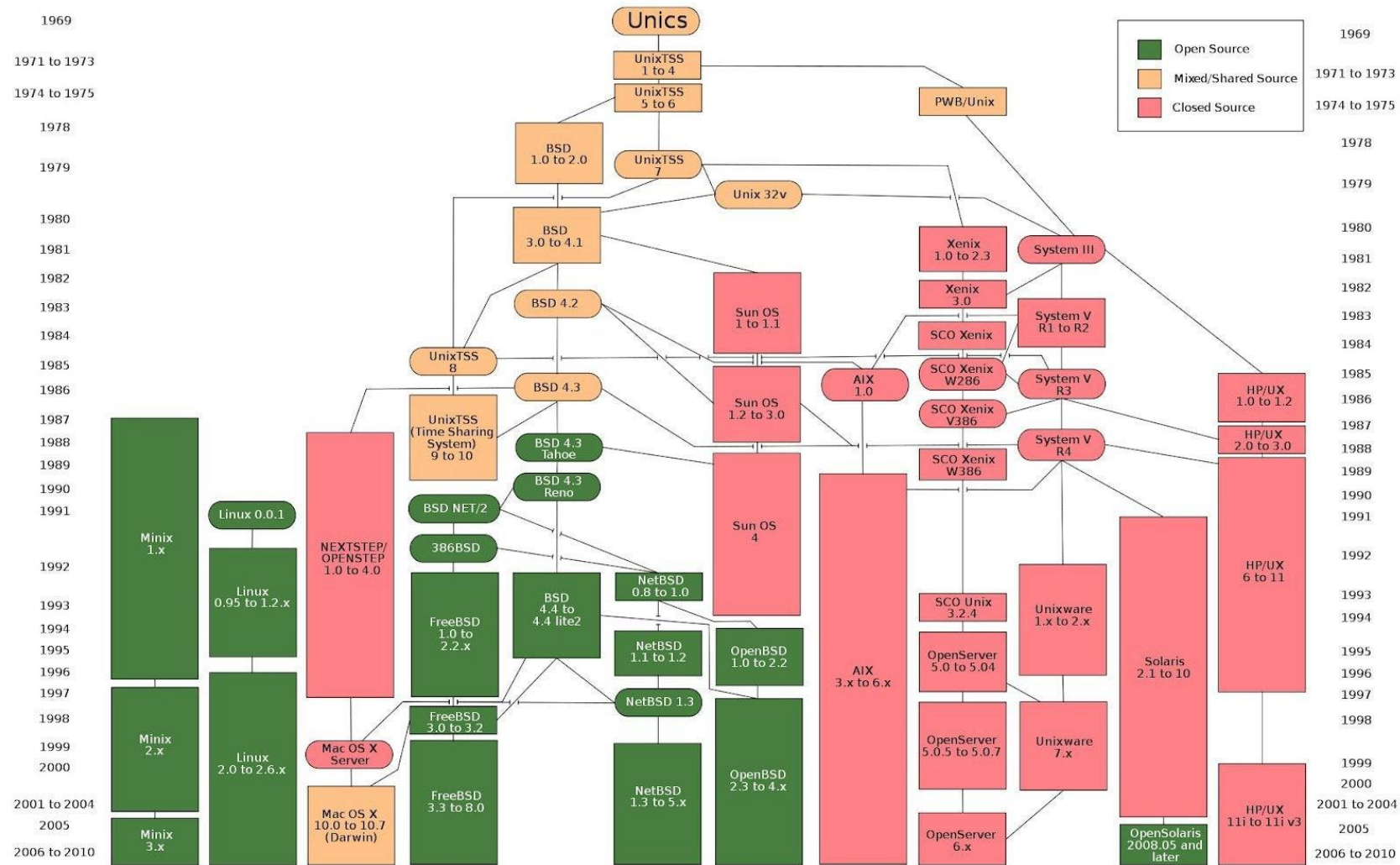
- 1991: Linux, kernel 0.01
- 1993: Windows NT 32 bits, Microsoft
- 1993: FreeBSD e NetBSD
- 1993: Newton OS, com gestão de energia e ecrã tátil
- 1995: Plan 9, um SO distribuído da AT&T
- 2001: MacOS X
- 2007: *iPhone* e iOS
- 2007: Android, com núcleo Linux



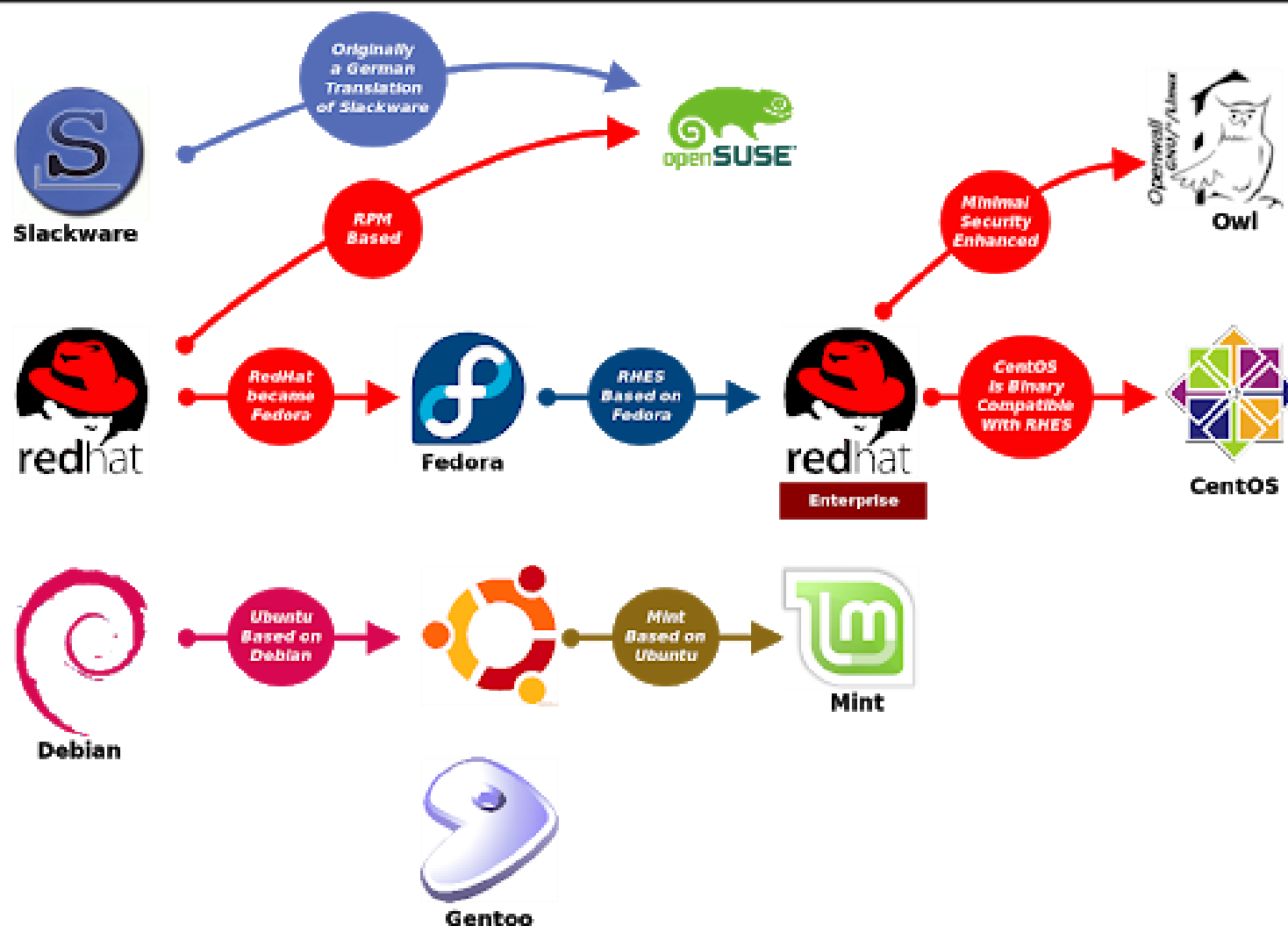
Evolução histórica dos Sistemas Operativos



Evolução histórica dos Sistemas Operativos



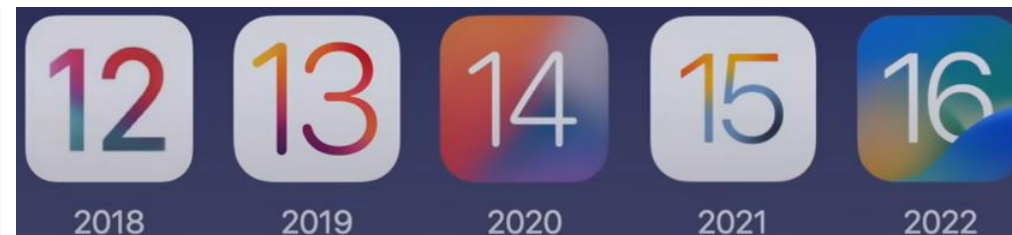
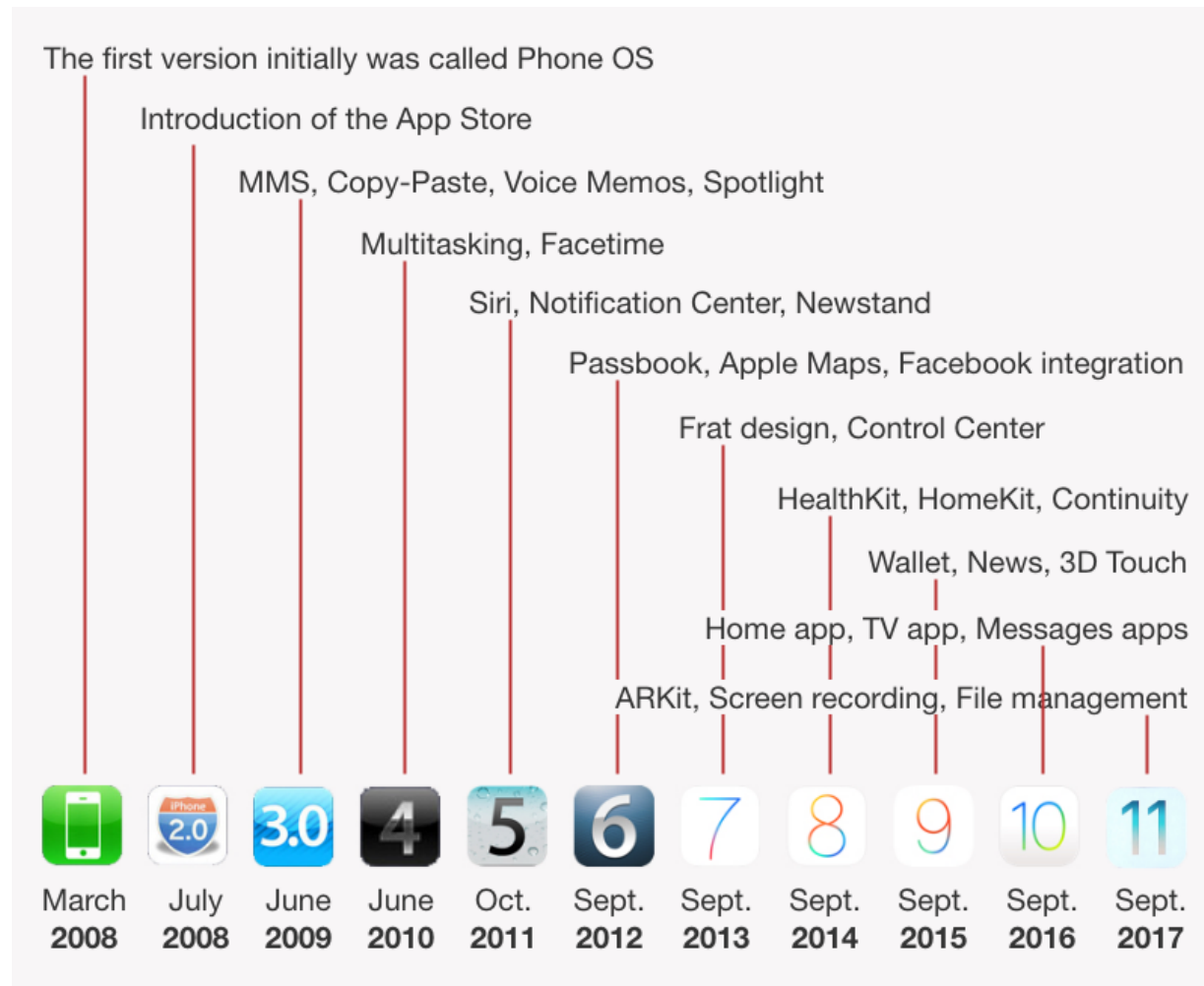
Evolução histórica dos Sistemas Operativos



Evolução histórica dos Sistemas Operativos



Evolução histórica dos Sistemas Operativos

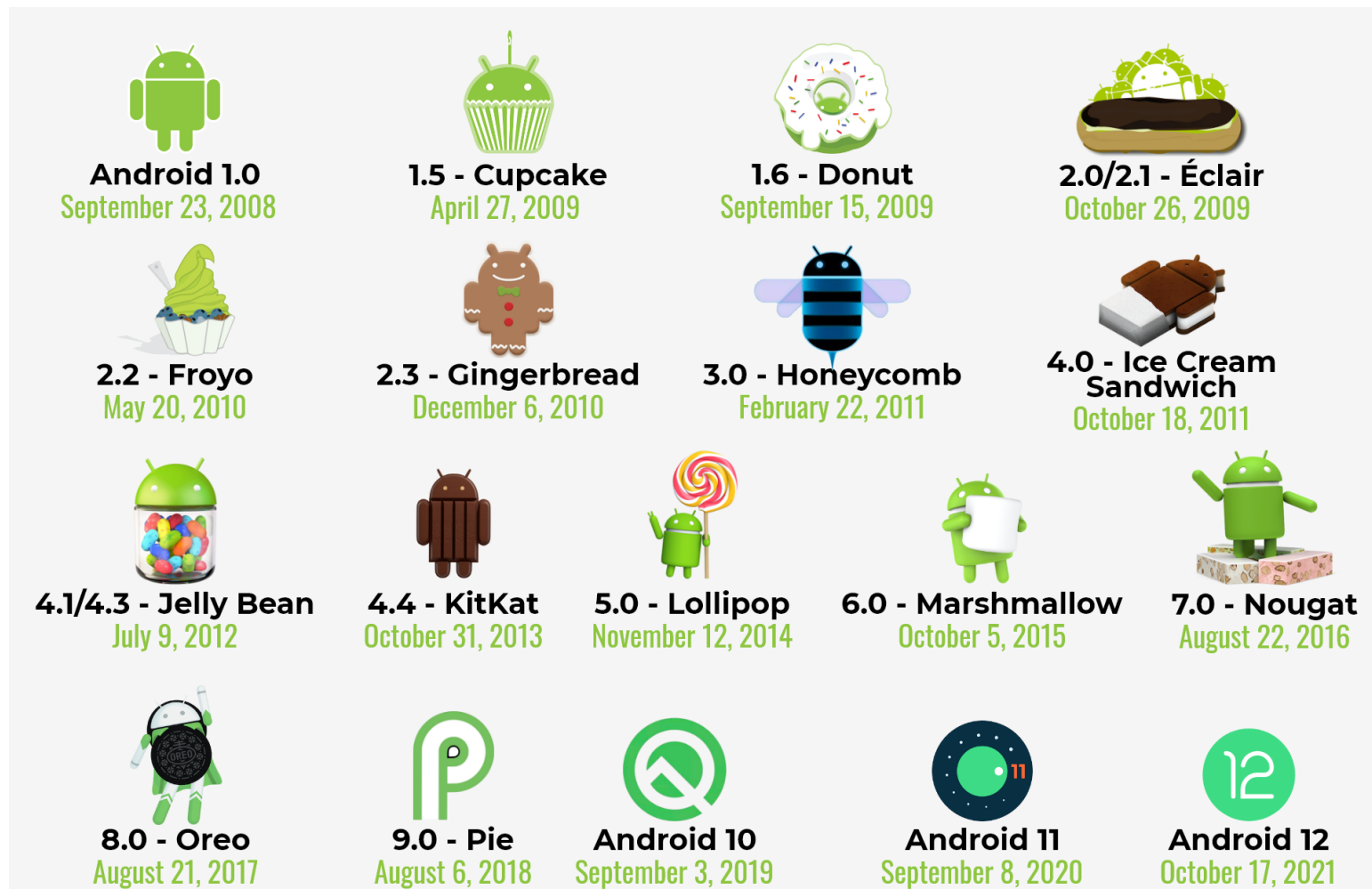


<https://www.youtube.com/watch?v=l7eucqQMXDw>



<https://www.youtube.com/watch?v=Wiu5yjZ7QnQ>

Evolução histórica dos Sistemas Operativos



10 February 2022

<https://www.youtube.com/watch?v=EGUPob3XJus>

Evolução histórica dos Sistemas Operativos

Timeframe	Representative OS(s)	Computer system	Main characteristics
1956	GM-NAA I/O	IBM 704	The first practical OS Simple batch processing I/O management
1960s	IBM OS/360 series	IBM 360 series—mainframes	Time-sharing Multibatch processing Memory management Virtual machines (VM/370)
1970s	Unix	Minicomputers/workstations	First modern OS Developed with machine-independent languages (C) Provides standard interfaces Integrated development environment
1980s	Mac OS, Windows, Linux	Personal computers (PCs)	Provides modern GUI Improves usability for personal users
2000s	Apple iOS, Google Android, Windows Phone	Smartphones	Customization of traditional OSs Improves usability for mobile devices New app delivery model (App Store, Google Play)

Utilização de Sistemas Operativos



Fonte: <https://gs.statcounter.com/os-market-share/desktop>

2ª Geração [1955-1965]

- FMS (*Fortran Monitor System*) desenvolvido pela North American Aviation
 - O sistema operativo *batch* para o IBM 709
- IBSYS desenvolvido pela IBM
 - O sistema operativo *batch* para a série 709x da IBM
 - Predecessor do sistema operativo OS/360

3ª Geração [1965-1980]

- OS/360 desenvolvido pela IBM
 - Um dos primeiros sistemas operativos a fazer uso da multi-programação para a série System/360
- CTSS (Compatible Time Sharing System) desenvolvido pelo MIT
 - Um dos primeiros sistemas operativos a fazer uso do conceito de *time sharing*
- MULTICS desenvolvido pelo MIT
 - Conceitos novos: gestão de memória, protecção de recursos
 - Exerceu uma grande influência nos sistemas operativos subsequentes

Disseminação dos mini-computadores

- (1961) DEC PDP-1 vendeu inúmeras unidades (custo por unidade ~ \$120,000)
- Ken Thompson escreve o sistema operativo UNIX num PDP-7
 - O UNIX começou por ser uma versão do MULTICS para um único utilizador (UNICS)
 - A divulgação do código fonte do UNIX levou ao aparecimento de vários sistemas UNIX *based* com destaque para:
 - System V (AT&T)
 - BSD (Berkeley Software Foundation) da Universidade de Berkeley
- IEEE desenvolveu a norma POSIX
 - Interoperabilidade entre sistemas UNIX

4ª Geração [1980 – Presente] (computadores pessoais)

- (1974) CP/M (Controlo Program for Microcomputer)
 - Sistema operativo escrito por Gary Killdall para testar o Intel 8080
 - Mais tarde Killdall adquiriu os direitos do CP/M e fundou a empresa Digital Research
 - Comercializou o CP/M para vários tipos de CPUs (8080, Z80, etc)
- Em 1980 a IBM desenha o IBM PC
 - Contacta Bill Gates (Microsoft) para obter uma licença para o interpretador BASIC e inquirir sobre um sistema operativo para o IBM PC
 - Bill Gates sugere a Digital Research (CP/M)

- IBM PC (cont.)
 - Killdall recusa o encontro com a IBM e envia um subordinado
 - A Digital Research recusa-se a assinar um NDA para o IBM-PC
 - IBM volta a contactar Bill Gates e este compra um sistema operativo chamado DOS (*Disk Operating System*) escrito por Tim Paterson pertencente à empresa chamada *Seattle Computer Products*
 - Vende o sistema à IBM com o nome MS-DOS e mantém os direitos sobre o sistema operativo
 - A disseminação dos IBM PCs (*clones*) contribuiu para o império a que hoje chamamos Microsoft
 - Sistemas Operativos produzidos pela Microsoft:
 - MS-DOS, Windows 3.x, Windows NT, Windows 9x, Windows Me, Windows 200x, ...

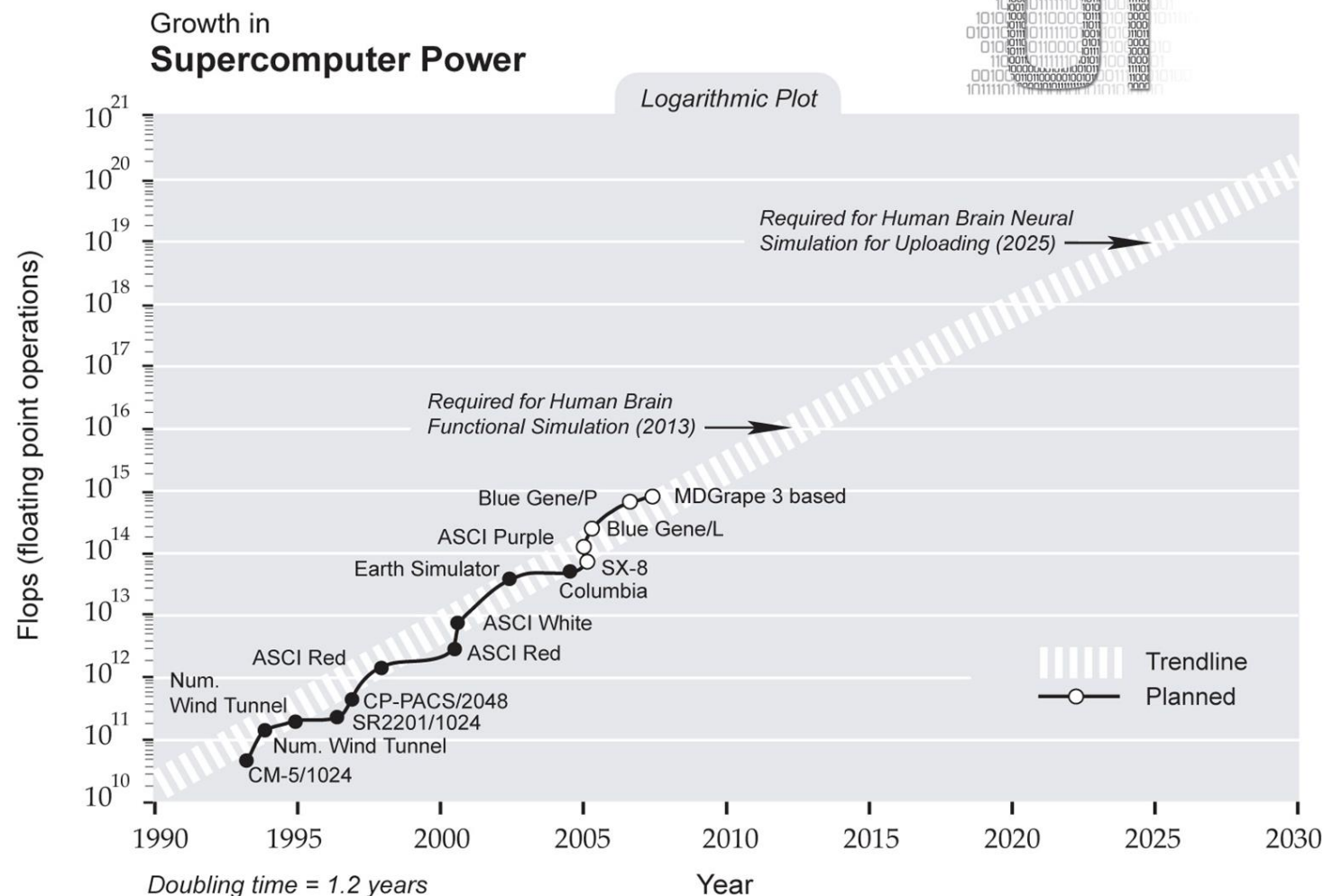
A história do Linux

- Em 1987, Andrew Tanenbaum escreve o sistema operativo MINIX (clone do UNIX para fins educativos)
 - Um dos primeiros sistemas operativos UNIX a utilizar uma arquitectura de micro-kernel
 - Sistema difundido e discutido na Internet
 - Muitos dos pedidos dos utilizadores eram recusados
- Em 1991, Linus Torvalds (estudante finlandês) desenvolve o Linux (clone do UNIX) para divulgação livre
 - O Linux possuía 2 grandes vantagens:
 - 1) Era livre e gratuito
 - 2) Tirava proveito da arquitectura dos processadores 386

At last but not the least

- Apple Macintosh
 - Primeiro computador a incorporar o conceito de GUI (Graphical User Interface)
 - Conjunto de conceitos inventado na Xerox (janelas, ícones, menus, rato, etc) para facilitar a interacção com os sistemas
- Este conceito foi “re-inventado” pela Microsoft com o sistema operativo Windows

E como será o futuro?



- O que é um Sistema Operativo?
- Qual a sua função?
- Que tipos de sistemas operativos existem?
- Como evoluíram os sistemas operativos