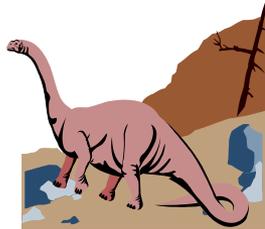




# Capítulo 1: Introdução

## SUMÁRIO:

- O que é um sistema operativo?
  - Definições e Objectivos
  - Componentes dum sistema de computação
  - Organização dum Computador
  - Definições
- Ambientes de Computação e SO
  - Primeiros computadores Primeiros SO's
  - Sistemas Mono e Multi Programado
  - Time Sharing (Partilha do Tempo)
  - Sistemas paralelos e distribuídos
  - Real Time (tempo Real)
  - Ambientes de Computação Atuais

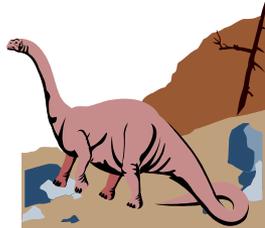




# 1.1 O que é um sistema operativo?

Antes: O que é um computador moderno ?

- **CPU** : Um ou mais processadores com um ou mais cores
- **Memoria** Principal : RAM : Volátil  
Secundária : Discos : Persistente
- **I/O** : Dispositivos de Entrada e Saída
  - ecrã, impressora, rato etc.
- Ainda é uma maquina do “Modelo” Von-Neumann
- A gestão deste componentes necessita de algo...  
uma camada de software chamado “sistema operativo”





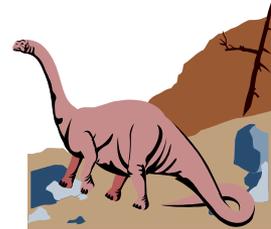
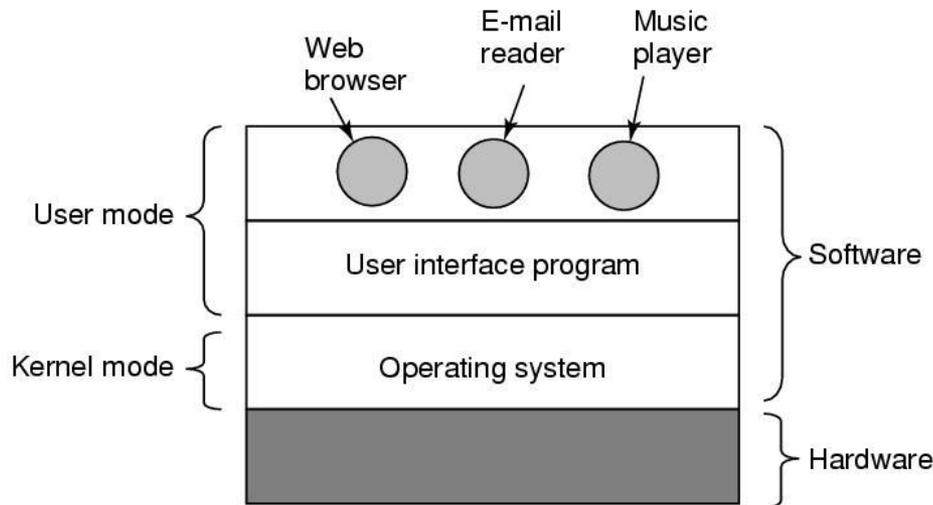
# 1.1 O que é um sistema operativo?

- **Definição preliminar:**

É um programa intermediário entre o utilizador e o hardware ou máquina física

- **Objectivos dum SO:**

- Executar e Gerir os programas do utilizador
- Tornar mais fácil a resolução dos problemas dos utilizadores.
- Tornar fácil o uso da máquina.
- Utilizar o hardware do computador duma forma eficiente.



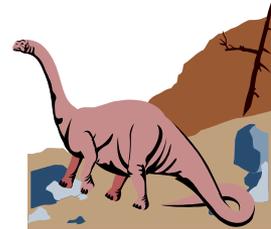
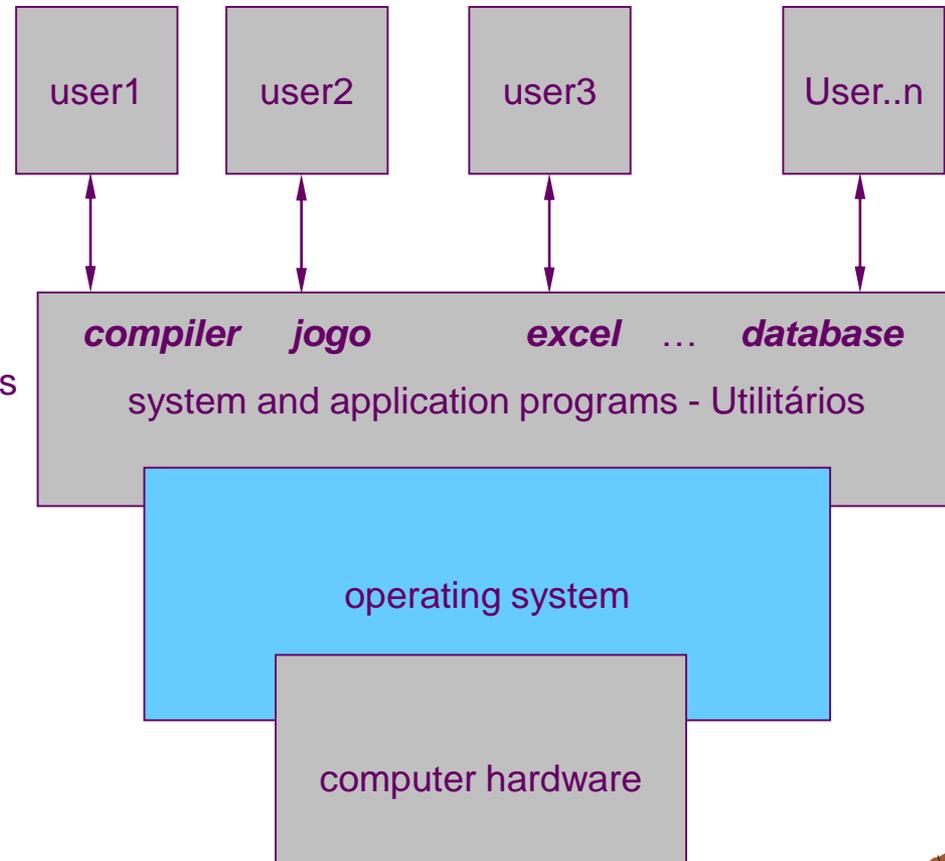
# Componentes dum Sistema de Computação



- 1. Hardware** – os recursos físicos CPU, memória, dispositivos I/O.
- 2. Sistema operativo** – controla e coordena a utilização do hardware durante a execução de vários programas de aplicação, eventualmente pertencentes a diferentes utilizadores.
- 3. Programas**

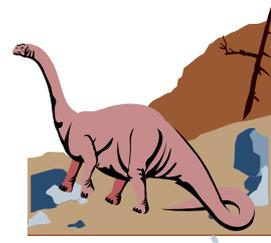
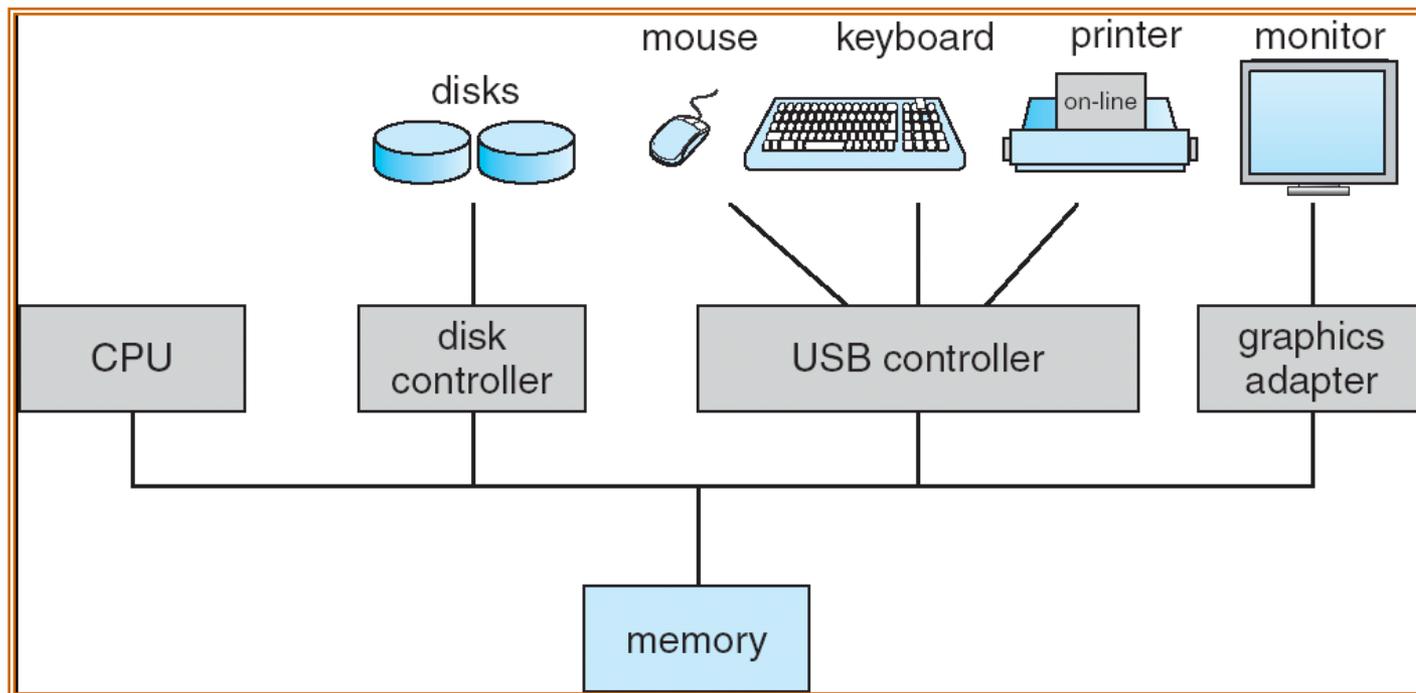
**Aplicações** – definem as formas de utilização dos recursos do sistema com o objectivo de resolver problemas dos utilizadores (compiladores, sistemas de bases de dados, jogos de vídeo, programas de contabilidade, etc.).

**Utilitários** – Conjunto de programas de sistema que têm funcionalidades muito úteis Gestor de Ficheiros, Instalador de Software
- 4. Utilizadores** -pessoas, máquinas, outros computadores , serviço de web  
O utilizador final do sistema que não está preocupado com a sua arquitectura



# Organização dum Sistema de Computação

- Funcionamento dum sistema computacional.
  - Um ou mais cpu's, controladores de dispositivos, ligados através dum ligação comum (bus) a uma memoria comum (memoria principal-RAM)
  - Execução concorrente dos CPU's e dispositivos, concorrência para os recursos de memoria e CPU.

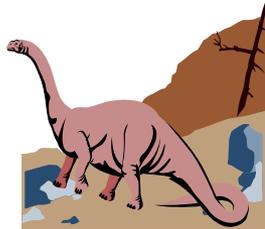




# Sistema Operativo

## O que fazem ?

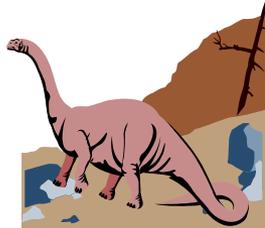
- **Depende do ponto da vista !**
- **Desktop/Laptop User**
  - **Facilidade do uso e bom desempenho**
    - **Não querem saber da utilização eficaz de recursos**
  - **Server**
    - **Load balancing, server response time, Otimização do desempenho**
  - **Dispositivos moveis**
    - **Otimizar tudo para bateria, recursos (e chamadas..)**
  - **Embedded / Critical**
    - **Sem interface Grafico !**
    - **Cumprir prazos temporais**



# Sistema Operativo

## Definição

- Uma Camada de Software (um programa) intermediário entre o utilizador e o hardware ou máquina física. **Um Gestor de recursos** – gere e reserva recursos da máquina
  - Gerir e Proteger os recursos : Memória, I/O devices etc.
  - Permite múltiplos programas (de vários utilizadores) utilizar a máquina ao mesmo tempo .. **multiplexing** (partilha) de recursos em duas maneiras
    - Em tempo
    - Em espaço
- **É um Programa de controlo** – controla a execução de programas dos utilizadores e de operações dos dispositivos I/O para evitar erros e utilização errada.
  - **NOTA: Núcleo (kernel)** – único programa que está sempre pronto a correr (todos os outros são “aplicações” ou de “programas de sistema”).



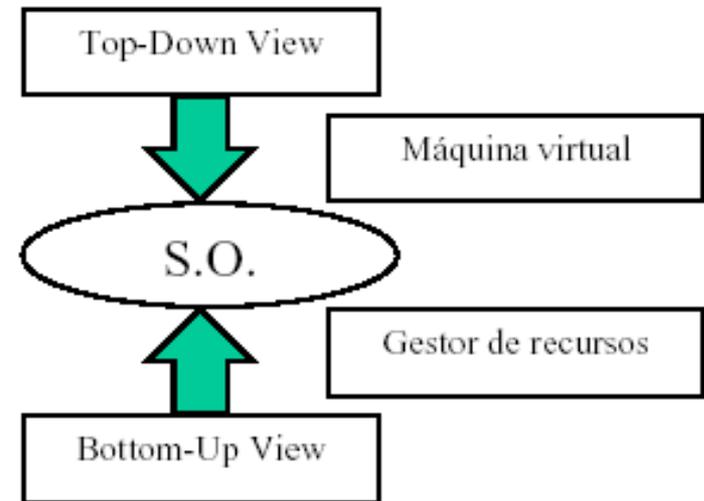


# Motivos históricos para o desenvolvimento dos Sistemas Operativos

libertar o user da complexidade do hardware

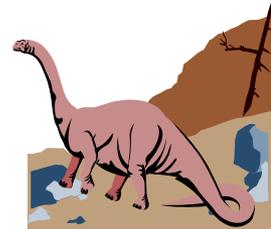
## SO como Máquina Virtual

- **Top-Down View**
- Transformar um conjunto diversificado de hardware numa máquina simples de utilizar.
- SO virtualiza o H/W. A abstração apresentada ao utilizador pretende-se simples e de fácil compreensão
- Apresenta ao utilizador uma interface que trata dum modo uniforme operações sobre entidades semelhantes
- Garantir fiabilidade e segurança



## SO como Gestor de Recursos

- **Bottom-up view**
- Obter o máximo rendimento do hardware
- Garantir uma Gestão dos Recursos
- Otimização do desempenho



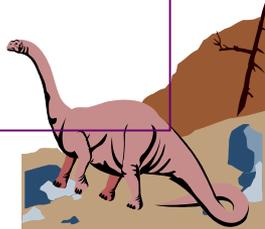
# Evolução dos SO's

Computadores mais caros do que humanos,  
Objectivo dum SO: maximizar a utilização do Hardware por exemplo com operações de I/O em simultaneo.

Custos de computação barato  
Humanos Caros

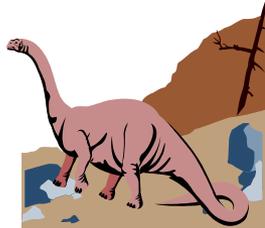
Objectivo principal dum SO é fazer os humanos (users/developers) mais eficaz e produtivos.

- Os primeiros sistemas “bare machines”, antes dos anos 50 (1945–55) *Vacuum Tubes and Plugboards*
- Sistemas Monoprogramados de tratamento por lotes (Batch Systems). 1º Sistema Operativo, anos 50, desenvolvido pela General Motors para um IBM701.
- Sistemas Multiprogramados de tratamento por lotes (Multiprogrammed Batch Systems) - anos 50
- Sistemas Multiprogramados Interactivos de partilha de tempo (Time-Sharing Systems) - CTSS, 1961, MIT
- Sistemas de Tempo Real - anos 60
- Sistemas Paralelos - anos 60
- Sistemas Embutidos - anos 60
- Sistemas Distribuídos - anos 80
- Sistemas para Dispositivos móveis
- HyperVisors (suporte para virtualização)



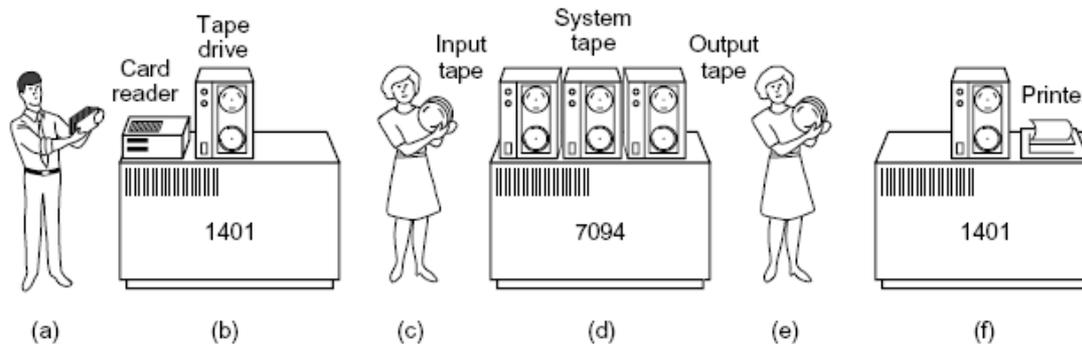
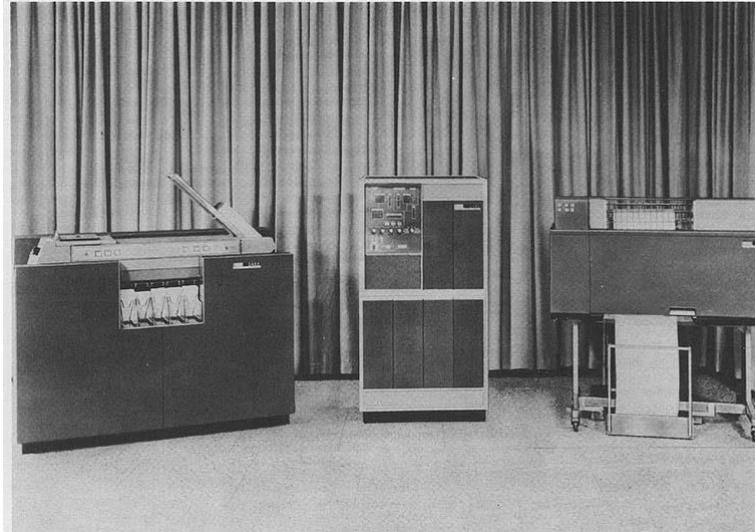
# Sistemas Mainframe

- Historicamente os Primeiros “Computadores”
  - Tempo de Execução do CPU - Muito Caro
  - Eficácia na gestão do CPU era muito importante
- Necessidades
  - Sequenciamento automático de jobs – transfere automaticamente o controlo de um job para outro.
  - Redução do “setup time” através do sequenciamento (batching) de jobs similares com os mesmos recursos (discos/fitas etc).
- “Monitor” residente: primeiro sistema operativo rudimentar
  - Controlo inicial pelo “monitor”
  - Transferência de controlo para um job
  - Transferência de controlo para o monitor após terminação dum job

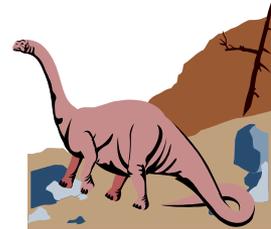




# IBM Series 1401 (1959-1971)



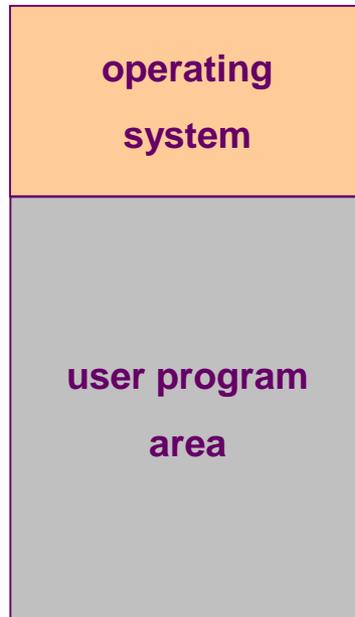
Ref. Tannenbaum: IBMS 1401 e 7094 trabalhando em conjunto !





# Partição de Memória em Sistemas Batch

## Sistema Mono-programado

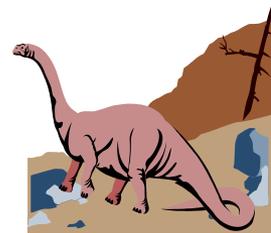


Multiplexagem: Definição:  
Combinação de dois ou mais canais (de informação) num meio de comunicação comum.

## Sistema Multi-Programado



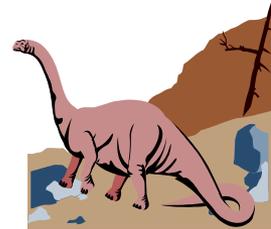
Vários **jobs** são mantidos em memória principal, simultaneamente, e a CPU é multiplexada entre eles.  
Partilha em Espaço





# Características dum SO necessárias à Multiprogramação

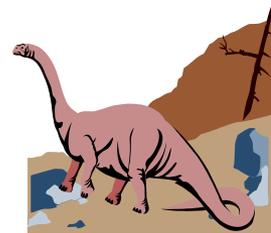
- Gestão de memória
  - o sistema tem de alocar e gerir memória para vários jobs.
- Escalonamento da CPU
  - o sistema tem de escolher entre os vários jobs prontos para correr.
- Rotinas de I/O fornecidas pelo sistema.
  - Bibliotecas de código
  - Interface Comum de acesso - Chamadas ao Sistema
- Reserva e Gestão de dispositivos (Device Management)
  - Device drivers



# Sistemas por Partilha de Tempo – Computação Interactiva –



- O sistema “on-line” tem de estar disponível aos utilizadores para acesso a dados e código.
  - É fornecida comunicação **on-line** entre o utilizador e o sistema.
    - quando o sistema operativo termina a execução dum comando, procura a próxima “instrução de controlo” introduzida por um utilizador via teclado.
- O sistema “batch” tem de executar programas de utilizadores “off-line”
- Problema : Memória Insuficiente para todos os jobs !!
- A CPU é atribuída a um job só se ele estiver em memória.
  - Um job é “swapped in” e “swapped out” da memória para o disco
- CPU é multiplexada entre os vários jobs em memória e no disco
  - aumento da complexidade
- Gestão de privacidade dos dados dos utilizadores
  - sistema de protecção





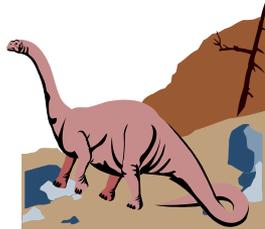
# Amdahl 5860

# Amdahl V7



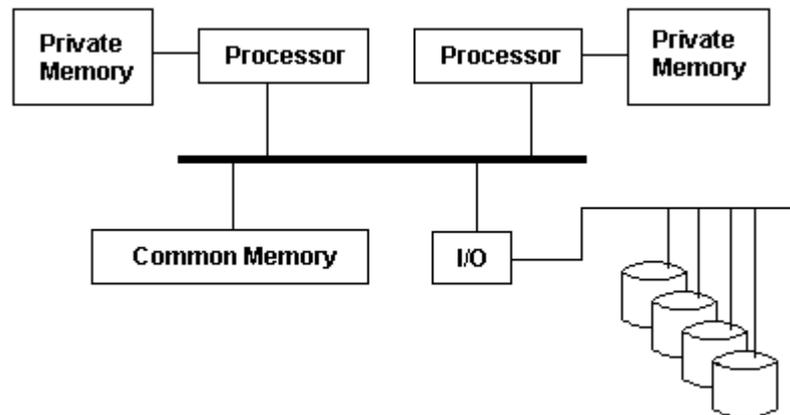
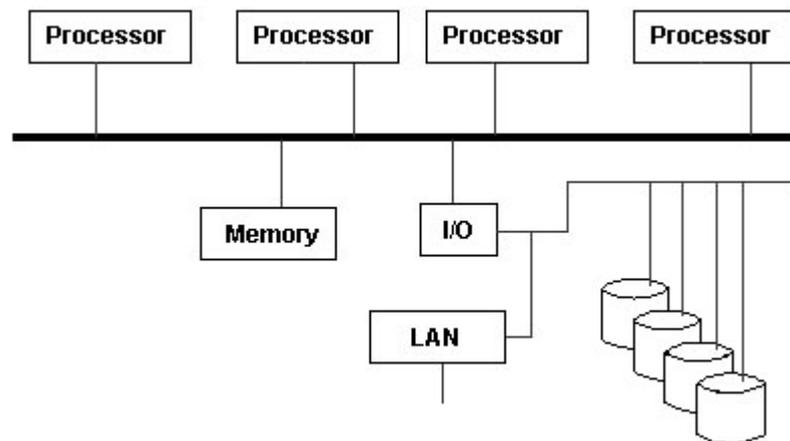
<http://www.staff.ncl.ac.uk/roger.broughton/>

OS VM/CMS



# Sistemas Multi-Processador/Core

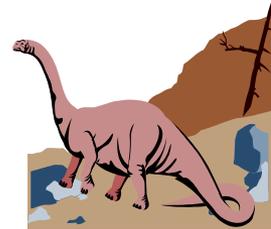
- *Symmetric multiprocessing (SMP)*
  - Cada processador poderá correr a única instancia do sistema operativo.
  - O SO escalone trabalho a qualquer dos processadores
  - Os seus portáteis Multi-Core
- *Asymmetric multiprocessing (AMP)*
  - Processador-mestre corre o sistema operativo; o processador-mestre escalona trabalho aos processadores-escravos.
  - Processadores-escravos correm aplicações. É atribuída uma tarefa específica a cada processador-escravo; mais comum em sistemas extremamente grandes.





# Outros Sistemas

- Distribuídos
- Cliente-Server
- Peer-to –Peer
- Fault Tolerant Clustering
- Real Time
- Safety Critical
- Embedded
- Mobile
- Cloud
- Automóvel
- etc





# Sistemas de Tempo Real

## Real-time system

- Sistema com necessidades temporais onde resultados têm que ser produzidos dentro dum tempo específico/prazo (chamado **deadline**)

## Embedded system

- Um dispositivo de computação embutido num sistema “maior”, tais como sistemas de aviação, automóveis, satélites etc. ( muitos vezes múltiplos inputs e outputs físicos )

## Safety-critical system

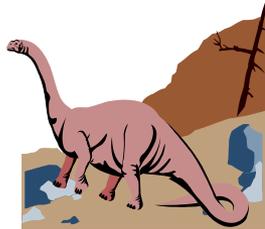
- Um sistema de tempo real com efeitos catastróficos em caso de falha. P.ex. Controlo de comportas, central nuclear, sistema medico

## Hard real-time

- Sistema que “garante” que tarefas são executados dentro das suas deadlines

## Soft real-time system

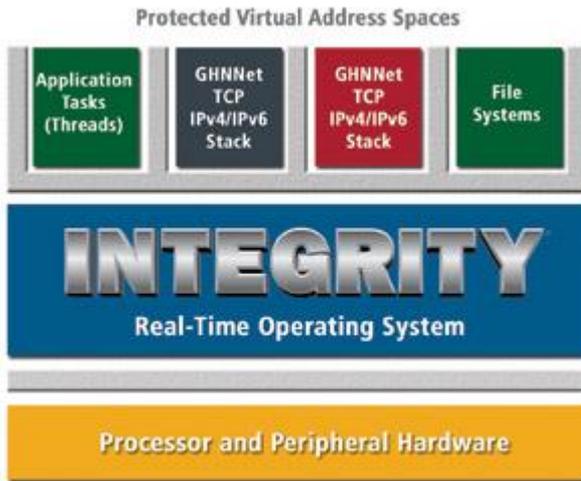
- Sistema onde tarefas de tempo real têm prioridade sobre tarefas normais





# INTEGRITY

## Real-Time Operating System

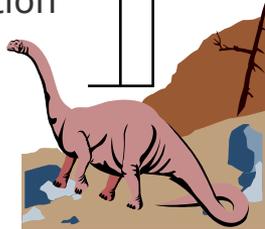


### Guaranteed security

The INTEGRITY RTOS provides all the capabilities embedded designers need to enforce the policies of separation, damage limitation, and information flow control as well as provide secure networking for today's more complex and connected applications.

INTEGRITY's Multiple Independent Levels of Security (MILS) separation kernel architecture provides a highly robust mechanism to separate security functions. A true MILS kernel, INTEGRITY has been certified to EAL 6+ High Robustness, the most rigorous Common Criteria security evaluation ever achieved for a commercial operating system.

INTEGRITY's **separation kernel** protects against damage from errant or malicious code by preventing processes from writing beyond assigned memory regions. In addition, INTEGRITY's partitions prevent unintended access to data from outside the partition where the data resides.





# RTEMS

## Real-Time Executive for Multiprocessor Systems

full featured RTOS that supports a variety of open API and interface standards

POSIX 1003.1b API (including threads) - VMEbus Industry Trade Association RTEID/ORKID Classic API – uITRON 3.0 API

TCP/IP including BSD Sockets - GNU Toolset Supports - File Systems: Fat.x etc

**RTEMS CENTRE**

EDI SOFT

RTEMS Centre

User Login

**RTEMS Centre**

- RTEMS Tools
- RTEMS Centre NEWS
- About RTEMS
- RTEMS Architecture
- Weblinks
- Mailing List
- Tools Licenses
- User Management

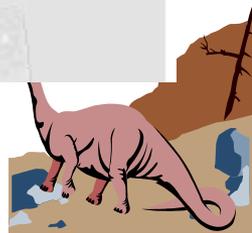
**About RTEMS Centre**

The **RTEMS Centre** is a project that is currently being sponsored by **ESA** (ESTEC Contract 20049/05/NL/JD/jk) in the scope of the **ESA-Portugal Task Force protocol**. Its purpose is to establish an European centre of expertise around RTEMS and in that way represent an added value for future space missions.

The goal of the RTEMS Centre website is to promote the use of RTEMS and show the world what is currently being done by Edisoft in respect to RTEMS.

In particular, the **RTEMS/Support Tools** shall be made available here.

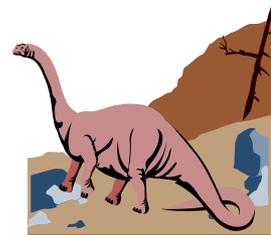
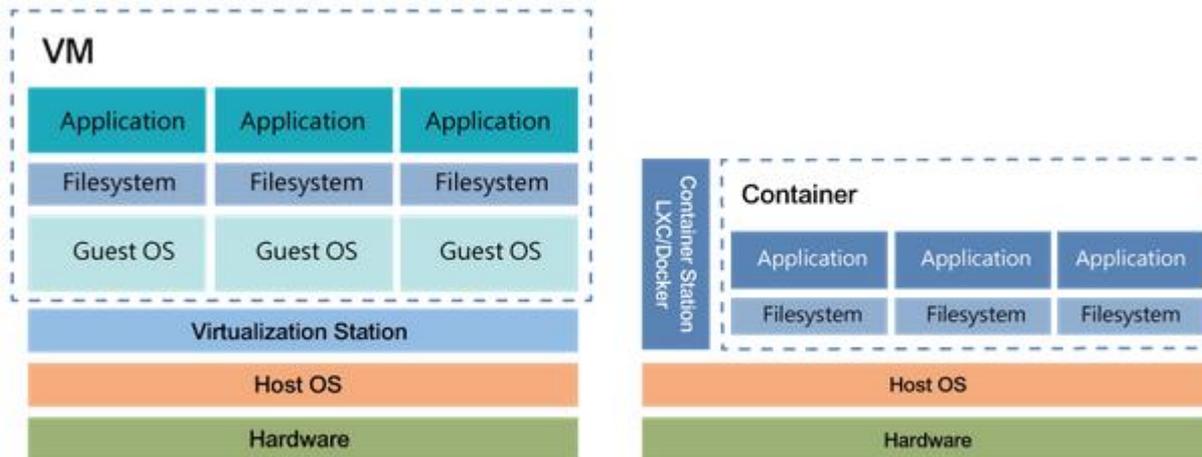
**News Update !!!**





# MIRAGE OS

- ❑ MirageOS is a library operating system that constructs unikernels **for secure, high-performance network applications** across a variety of cloud computing and mobile platforms.
- ❑ Code can be developed on a normal OS such as Linux or MacOS X, and then compiled into a fully-standalone, specialised unikernel that runs under a Xen or KVM hypervisor.
- ❑ MirageOS uses the OCaml language, with libraries that provide networking, storage and concurrency support that work under Unix during development, but become operating system drivers when being compiled for production deployment



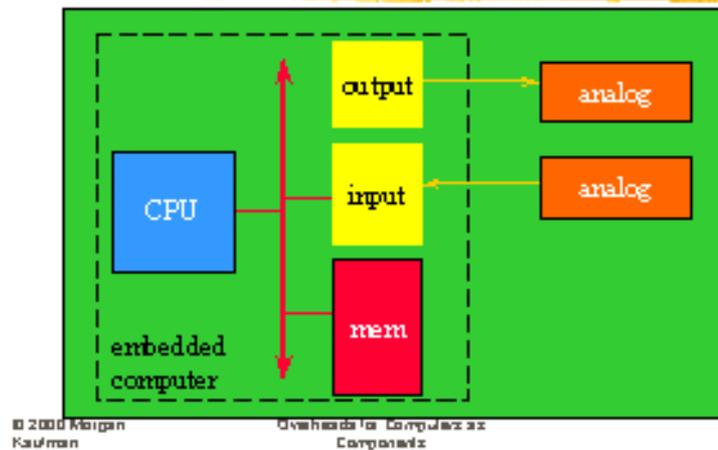


# Ambientes de Computação Actuais

- ❑ Computação Tradicional
- ❑ Computação Baseada na Web
- ❑ Computação Ubíqua
- ❑ Computação Embutida

- ❑ Printers
- ❑ Automóveis
- ❑ Televisões
- ❑ Redes de Sensores
- ❑ Domótica
- ❑ Personal Digital Assistants (PDAs)
- ❑ Telefones e Pads
  - ❑ Symbian –Dead
  - ❑ Blackberry – Almost Dead
  - ❑ Windows Mobile (10 Unified .. But Dead)
  - ❑ Android - Google
  - ❑ Apple -IOS

## Embedding a computer



- ❑ Questões em aberto:
  - ❑ Memória limitada
  - ❑ Processadores lentos
  - ❑ Ecrãs pequenos.
  - ❑ Ecrã Tácteis

FIM DE CAPÍTULO



# Migração dos Conceitos e Características dos Sistemas Operativos

