



# 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 Actuais



## 1.1 O que é um sistema operativo?

Antes: O que é um computador moderno ?

- Um ou mais processadores com um ou mais cpu cores
- Memória Principal (ram, caches etc.)
- Memória Secundária Discos
- Dispositivos de Entrada e Saída
  
- Ainda é uma maquina do Modelo Von-Neumann
  
- A gestão deste componentes necessita de algo... uma camada de software chamado "sistema operativo"





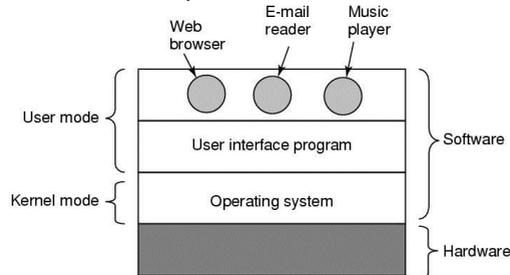
## 1.1 O que é um sistema operativo?

### ■ Definição preliminar:

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

### ■ Objectivos :

- ☞ Executar programas do utilizador e tornar mais fácil a resolução de problemas.
- ☞ Tornar fácil o uso da máquina.
- ☞ Utilizar o hardware do computador duma forma eficiente.



Operating System Concepts

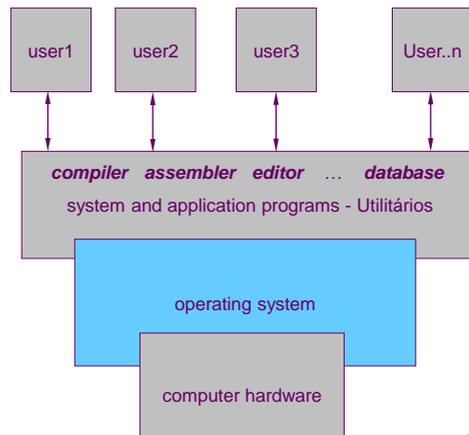
1.3

Silberschatz, Galvin and Gagne ©2002



## 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.
2. **Programas de aplicação** – 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.).
3. **Utilitários** – Conjunto de programas de sistema que têm funcionalidades muito úteis
4. **Utilizadores** -pessoas, máquinas, outros computadores – utilizador final do sistema que não está preocupado com a sua arquitectura



Operating System Concepts

1.4

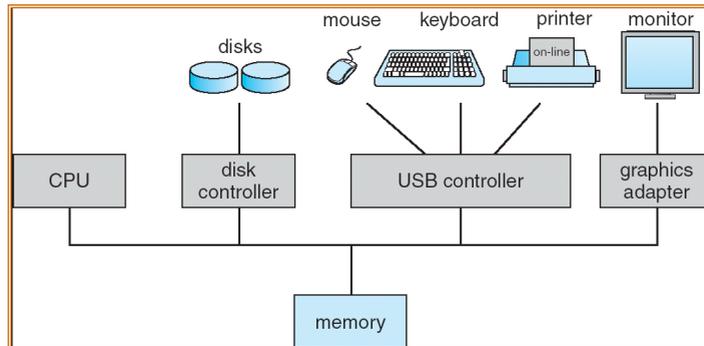
Silberschatz, Galvin and Gagne ©2002





## Organização dum Sistema Computacional

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



Operating System Concepts

1.5

Silberschatz, Galvin and Gagne ©2002



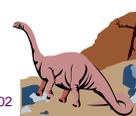
## Sistema Operativo: Definições

- **Gestor de recursos** – gere e reserva recursos.
  - ☞ Permite múltiplos programas de vários utilizadores utilizar a máquina ao mesmo tempo
  - ☞ Gerir e Proteger a Memória, I/O devices etc.
  - ☞ Permite multiplexing (partilha) de recursos em duas maneiras
    - 📁 Em tempo
    - 📁 Em espaço
- **Programa de controlo** – controla a execução de programas dos utilizadores e de operações dos dispositivos I/O.
- **Núcleo (kernel)** – o único programa que está sempre pronto a correr (todos os outros são programas de aplicação).

Operating System Concepts

1.6

Silberschatz, Galvin and Gagne ©2002





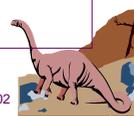
## Evolução dos SO's

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.

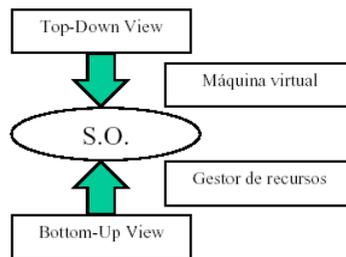
- Os primeiros sistemas "bare machines", antes dos anos 50 (1945–55) *Vacuum Tubes and Plugboards*
- Sistemas Monoprogramados de tratamento por lotes (Batch Systems). 1º Sistema Operativo, anos 50, desenvolvido pela General Motors para um IBM701.
- Sistemas Multiprogramados de tratamento por lotes (Multiprogrammed 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 Distribuídos - anos 80
- Sistemas Embutidos e Dispositivos móveis



## 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 abstracção 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
  - ☞ Optimização do desempenho





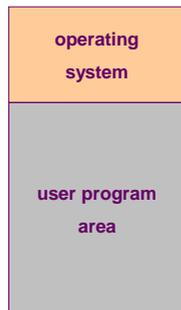
# Sistemas Mainframe

- Historicamente os Primeiros “Computadores”
  - ☞ Tempo de Execução do CPU - Muito Caro
  - ☞ Eficácia na gestão do CPU era muito importancia
  
- 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

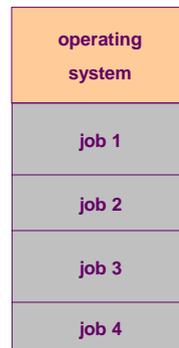


# Partição de Memória em Sistemas Batch

## Sistema Mono-programado



## Sistema Multi-Programado



Multiplexagem:

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

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





## 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 Mangement)
  - ☞ Device drivers



## Sistemas por Partilha de Tempo – Computação Interactiva –

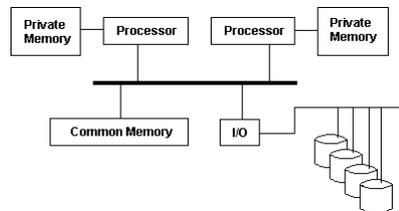
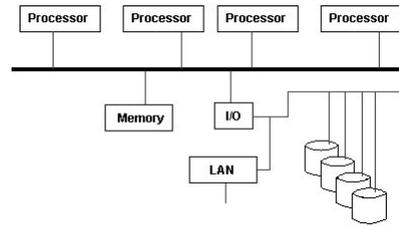
- O sistema “*on-line*” tem de estar disponível aos utilizadores para acesso a dados e código.
  - ☞ É fornecida comunicação *on-line* entre o utilizador e o sistema.
    - quando o sistema operativo termina a execução dum comando, procura a próxima “instrução de controlo” introduzida por um utilizador via teclado.
- O sistema “*batch*” tem de executar programas de utilizadores “*off-line*”
- 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





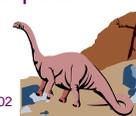
## Sistemas Paralelos - Multiprocessador

- **Symmetric multiprocessing (SMP)**
  - ☞ Cada processador corre uma cópia idêntica do sistema operativo.
  - ☞ Muitos processos podem correr simultaneamente sem degradação no desempenho.
  - ☞ A maior parte dos sistemas operativos modernos suportam SMP.
- **Asymmetric multiprocessing (AMP)**
  - ☞ Processador-mestre corre o sistema operativo; o processador-mestre escala o trabalho aos processadores-escravos.
  - ☞ Processadores-escravos correm aplicações. É atribuída uma tarefa específica a cada processador-escravo; mais comum em sistemas extremamente grandes.



## Sistemas Distribuídos

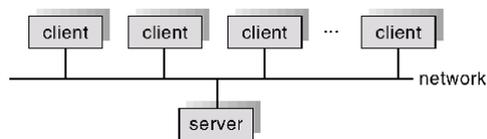
- Distribui a computação por vários processadores físicos.
- *Loosely coupled system* – cada processador tem a sua própria memória local; os processadores comunicam entre si através de várias linhas de comunicação tais como barramentos (buses) de alta-velocidade ou redes dedicados ou ad-hocs
- Vantagens:
  - ☞ Partilha de recursos
  - ☞ Maior rapidez de computação – partilha de carga
  - ☞ Fiabilidade
  - ☞ Comunicações
- Requer infra-estrutura de rede.
- Local area networks (LAN) e Wide area networks (WAN)
- Podem ser : “sistemas client-server” ou “sistemas peer-to-peer”.





## Arquitectura de Rede Cliente-Servidor

- Nesta arquitectura, há computadores ou processos que são servidores e outros que são clientes daqueles.
- Os servidores são máquinas potentes ou processos que fazem a gestão de discos rígidos (file servers), de impressoras (printer servers), ou de tráfico de rede (network servers).
- Os clientes são PCs ou workstations nos quais os utilizadores correm aplicações. Os clientes servem-se dos recursos dos servidores, tais como ficheiros, dispositivos, ou mesmo poder de processamento.



## Arquitectura de Rede entre Pares (Peer-to-Peer)

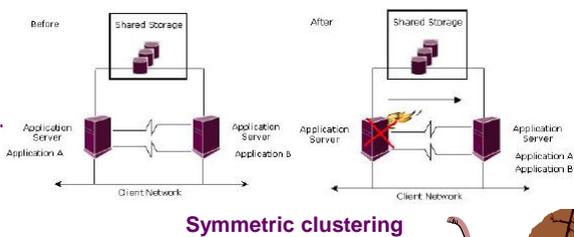
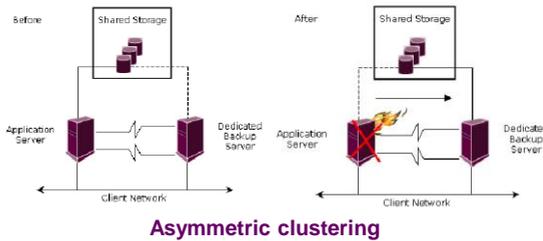
- Um tipo de rede na qual cada workstation tem capacidades e responsabilidades equivalentes.
- Ao invés, na arquitectura cliente-servidor há computadores que estão dedicados a servir outros (os clientes).
- As redes peer-to-peer são geralmente mais simples, mas não oferecem o mesmo desempenho face a sobrecargas.





## Sistemas Aglomerados Re-configuráveis sob Falhas (Fail Over Clustering)

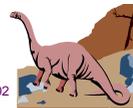
- Clustering permite que dois ou mais sistemas partilhem memória (storage).
- Fornece elevada fiabilidade. Fornece acesso contínuo a dados de aplicações mesmo se um dos servidores entra em falha.
- *Asymmetric clustering*: Um servidor corre aplicações enquanto os outros servidores ficam em standby.
- *Symmetric clustering*: Todos os servidores correm aplicações.



Operating System Concepts

1.17

Silberschatz, Galvin and Gagne ©2002



## Sistemas de Tempo-Real

- Restrições temporais bem-definidas.
- É usado muitas vezes como dispositivo de controlo numa aplicação dedicada (ex., controlo de experiências científicas, sistemas de imagem médica, sistemas de controlo industrial, etc.).
- Tipos de sistemas de tempo real: *hard* real-time e *soft* real-time.
- **Hard real-time**
  - ☞ Armazenamento secundário limitado ou ausente, dados armazenados em memória de curto-prazo, ou read-only memory (ROM)
  - ☞ Conflictos com sistemas de partilha de tempo; não suportados por sistemas de propósito geral
- **Soft real-time**
  - ☞ Utilidade limitada em controlo industrial via robótica
  - ☞ Útil em aplicações (multimedia, realidade virtual) que requerem características avançadas de sistemas operativos.

Operating System Concepts

1.18

Silberschatz, Galvin and Gagne ©2002





# 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

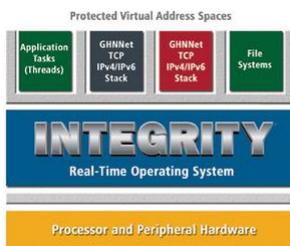
- Sistema que “garante” que tarefas são executados dentro das suas deadlines

## Soft real-time system

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



# INTEGRITY Real-Time Operating System

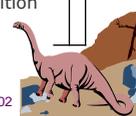


## Guaranteed security

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**

**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.

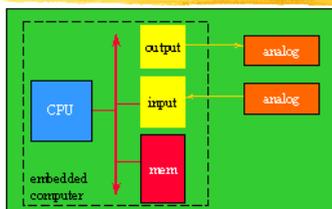
**News Update !!!**



# Ambientes de Computação Actuais

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

### Embedding a computer



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

- Printers
- Automóveis
- Televisões
- Redes de Sensores
- Domotica
- Personal Digital Assistants (PDAs)
- Telefones celulares
  - ☞ Symbian SO –2008 20 milhões telemóveis
  - ☞ Windows Mobile
  - ☞ Windows CE (embedded)
  - ☞ LiMo - Linux Mobile
  - ☞ Android - Google
  - ☞ Palm OS
  - ☞ BlackBerry

FIM DE CAPÍTULO



