

Introdução

Sistemas Operativos

2008 / 2009

Para que serve um Sistema Operativo?

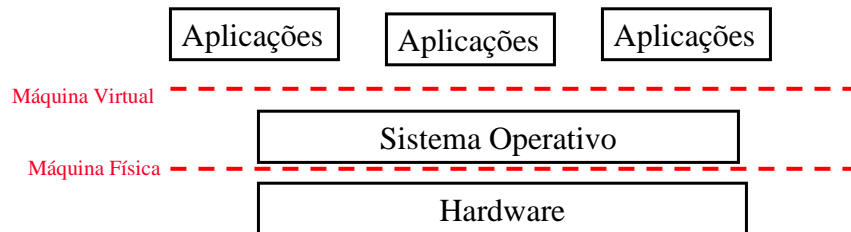
Para que serve um Sistema Operativo?

- Gerir Recursos
 - Seria difícil às aplicações controlar todos os aspectos da máquina física (interrupções, organização da memória, dispositivos, ...)
 - SO permite abstrair os **recursos físicos**, oferecendo às aplicações um conjunto de **recursos lógicos**.
Exemplos?

Para que serve um Sistema Operativo?

- Fornecer uma interface de acesso / gestão dos recursos lógicos
- Existem duas interfaces:
 1. Interface Operacional
 - Comandos para usar / criar / eliminar recursos
 - Exemplos?
 2. Biblioteca de Funções do Sistema Operativos
 - API para interagir com recursos do SO
 - Exemplos?

Missão do Sistema Operativo



- Criar uma máquina virtual sobre a máquina física que ofereça os recursos lógicos básicos necessários ao desenvolvimento das aplicações
- Independente do hardware onde se executa

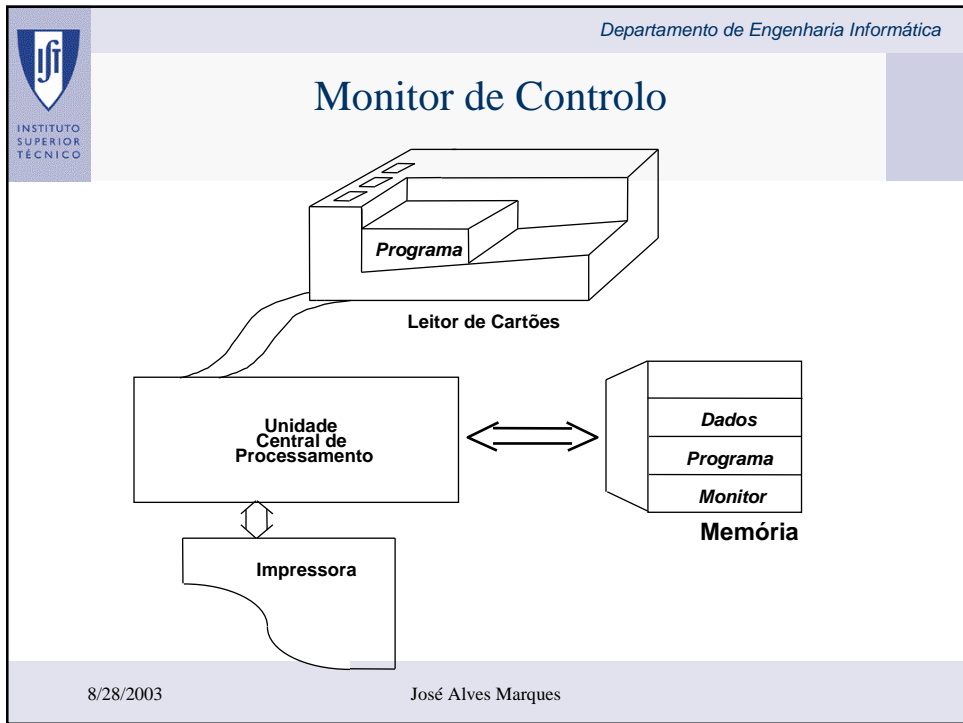
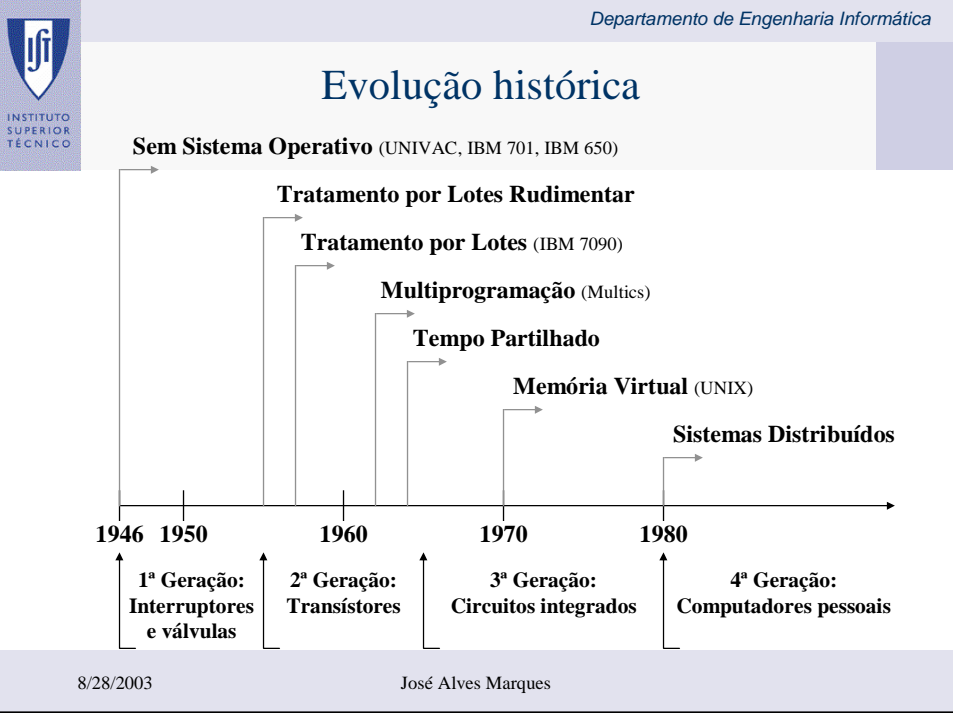
Alternativas ao Sistema Operativo

- As linguagens de programação podiam produzir todo o código necessário para que um programa se executasse directamente sobre o hardware.
- Desvantagens?
 - O esforço de programação seria muito grande
 - Um conjunto significativo de funções seria repetido
 - Cada aplicação poderia otimizar o seu desempenho mas globalmente a máquina ficaria subaproveitada.
 - Não seria possível ter políticas globais de segurança, tolerância a faltas, optimização

Critérios de Qualidade do SO

Critérios de Qualidade do SO

- **Desempenho**
 - Gestão eficiente dos recursos físicos que suportam os recursos lógicos
- **Segurança.**
 - Isolamento dos Utilizadores
 - Permitir partilha segura de recursos lógicos
- **Fiabilidade e Disponibilidade**
 - Detectar um conjunto de faltas
 - Tolerar um conjunto de erros
- **Interface de programação completa e simples.**
 - Facilitar a concepção das aplicações, a sua manutenção e portabilidade
- **Interface de operação e gestão dos recursos lógicos fácil de utilizar**



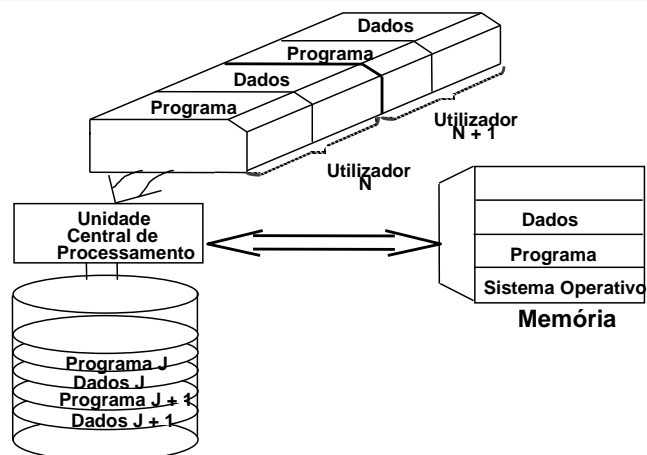
Monitor de Controlo

- Permite ao utilizador:
 - Carregar programas em memória, editá-los, etc.
 - Resultados dos programas: listagens, fitas perfuradas
- Cada utilizador tem um determinado tempo atribuído durante o qual tem o computador apenas para si
- O monitor é formado por um conjunto de utilitários:
 - Interpretador de linguagem de comando
 - Compilador, Assemblador (Assembler)
 - Editor de ligações (linker)
 - Carregador de programas em memória (loader)
 - Biblioteca de rotinas para controlo de periféricos (consola, leitor de cartões, etc.)
- Limitação Principal?

8/28/2003

José Alves Marques

Tratamento em Lotes (*Batch*)



8/28/2003

José Alves Marques

Tratamento em Lotes (cont.)

- Os periféricos mecânicos (ex.: impressoras, leitores/perfuradores de fita) eram muito lentos quando comparados com a velocidade de processamento do computador
- Solução inicial:
 - Separar as Entradas/Saídas do processamento
 - Entrada:
 - Computador auxiliar lê os trabalhos e executar escreve-os para ficheiros em banda magnética
 - Quando o trabalho em curso termina o SO vai à lista de trabalhos e selecciona o próximo a executar-se
 - Saída
 - Em vez de imprimir directamente os programas escrevem a saída em ficheiros que são enviados para a impressora quando a aplicação termina - spooling
- Evolução:
 - Os periféricos executam tarefas autónomas e avisam o processador do fim da sua execução através de interrupções.
 - Execução em paralelo dos programas e das E/S

8/28/2003

José Alves Marques

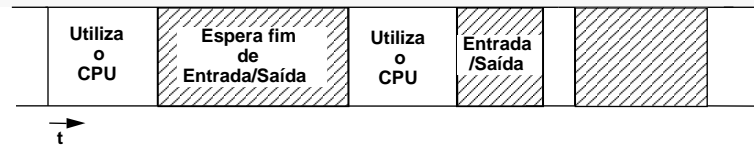
Multiprogramação

- O mecanismo de interrupções permite multiplexar o processador entre várias actividades concorrentes.
 - No exemplo anterior entre um programa e as entradas/saídas,
 - Mas esta capacidade de alternar a execução pode ser estendida à multiplexagem de vários programas residentes na memória.
- Execução concorrente de vários programas:
 - permite otimizar a utilização do processador
 - ex.: Programa P1 acede ao disco e fica bloqueado enquanto o controlador de disco funciona; durante esse tempo, o Programa 2 pode ser executado pelo processador

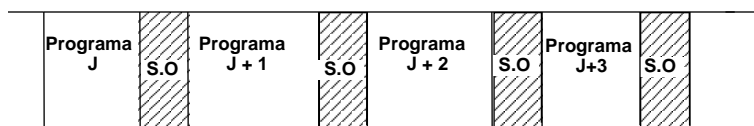
8/28/2003

José Alves Marques

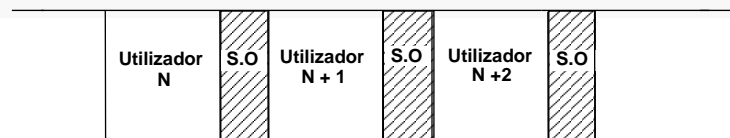
Multiprogramação



Utilização do processador num sistema monoprogramado



Tempo Partilhado



- Cria a ilusão que o computador está permanentemente disponível para o utilizador
- Impulsionado por:
 - Aplicações em que vários utilizadores interactuavam directamente com o sistema (sistemas de reservas)
 - Descida dos preços dos terminais
- Possível porque a maioria das aplicações interactivas usa pouco o processador
- Ex.: CTSS (início dos anos 60)

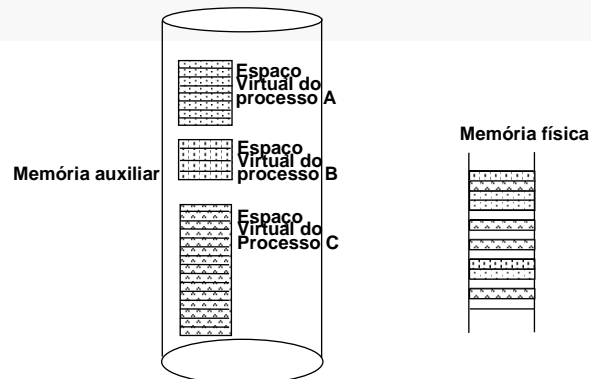
Consequências do Tempo Partilhado

- Revisão dos algoritmos de escalonamento
- Definição de mecanismos de segurança
- Aparecimento dos sistemas de ficheiros
- Outros aspectos?
- Tempo partilhado → vários programas em memória simultaneamente
- Hierarquia de memória

8/28/2003

José Alves Marques

Memória Virtual



- Elimina a restrição física imposta pelo tamanho da memória física permitindo um grau de multiprogramação muito superior
- Impulsionada pela propriedade da **localidade**

8/28/2003

José Alves Marques

Departamento de Engenharia Informática

Interface Gráfica

The screenshot displays a Windows desktop environment. In the foreground, a Microsoft PowerPoint window is open, showing a slide titled "Memória Virtual" (Virtual Memory). The slide content includes a diagram illustrating the relationship between physical memory and virtual memory, and two bullet points: "Elimina a restrição física imposta pelo tamanho da memória física permitindo um grau de multiprogramação muito superior" and "Impulsionada pela propriedade da localidade". The desktop background is a light blue gradient. In the top left corner, there is a logo for Instituto Superior Técnico. The taskbar at the bottom shows several open applications, including Firefox, and the system tray displays the date and time as 8/28/2003, 15:42.

Departamento de Engenharia Informática

Sistemas Distribuídos

- Consequência da evolução de:
 - Redes de dados
 - Computadores pessoais – PC
 - Sistemas abertos (normalização oficial e *de facto*)
 - Custo e desempenho da electrónica digital
- Aplicações iniciais: Mail, FTP
- SO tem vindo a englobar funcionalidade dos sistemas distribuídos
 - Exemplos: protocolos de rede, sistema de ficheiros distribuído

8/28/2003

José Alves Marques

Tipos de Sistemas Operativos

Tempo Virtual vs. Tempo Real

- **Tempo virtual:**
 - O tempo de execução dos programas não tem relação com o tempo cronológico exterior ao computador.
 - São os sistemas habituais utilizados na maioria dos computadores quer os que se executam nas máquinas clientes, quer nos servidores (ex: Windows, Linux, MAC/OS)
- **Tempo real:**
 - Tentam garantir que o computador produz uma resposta a um acontecimento externo num intervalo de tempo determinado. (Caso contrário o sistema não cumpre a sua especificação – falha.)

Sistemas de Tempo Real

- Inicialmente usados para controlo de processos industriais
- Hoje também usados para jogos, sistemas de controlo em automóveis, aviões, etc.
- Oferta extensa de SOs de tempo real, na sua maioria usada em **sistemas embebidos**:
 - Software integrado com o hardware
 - O sistema não oferece interface para desenvolver aplicações

8/28/2003

José Alves Marques

Sistemas de Tempo Real (cont.)

- Os requisitos de tempo real não são idênticos para diferentes aplicações, o que levou a distinguir dois tipos de sistemas tempo real
 - Tempo real relaxado ou soft real time – sistema onde se admite que certas respostas a acontecimentos externos podem não ser dadas exactamente nos intervalos de tempo especificados
 - Tempo real estrito ou hard real time – sistema onde o não cumprimento de um requisito temporal corresponde a uma falha.

A gestão de tempo real estrito obriga a um escalonamento de processos que torna estes sistemas totalmente incompatíveis com o funcionamento em tempo virtual interactivo, por essa razão não os iremos considerar neste curso.

8/28/2003

José Alves Marques

Sistemas Proprietários vs. Abertos

- 1970s: Vasta oferta de SOs de tempo partilhado (Unix, VMS, OS390, ...)
- Todos eles **sistemas proprietários**:
 - Tinham restrições (legais e técnicas) na sua utilização e cópia
 - Desenvolvidos pelo fabricante de hardware
 - Código assembly, não divulgado
 - Gera dependências: aplicação / SO / Plataforma

Sistemas Abertos

- Sistema portátil e interoperável
- Oferece uma interface normalizada
- Génese no sistema Unix
 - Divulgação do código fonte
 - Interfaces de programação standard
 - Encorajava o desenvolvimento de SW e HW por terceiros
- Linux – open source, segue a licença GPL

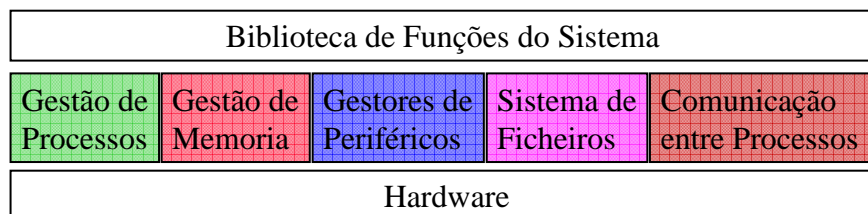
Organização do Sistema Operativo

8/28/2003

José Alves Marques

Organização Típica

Processo Sistema
Processo Sistema
Processo Sistema



- SO divide-se em:
- Núcleo (*kernel*) – dividido em módulos
- Biblioteca das funções sistema (*system calls*) – usadas pelas aplicações
- Processos sistema

8/28/2003

José Alves Marques

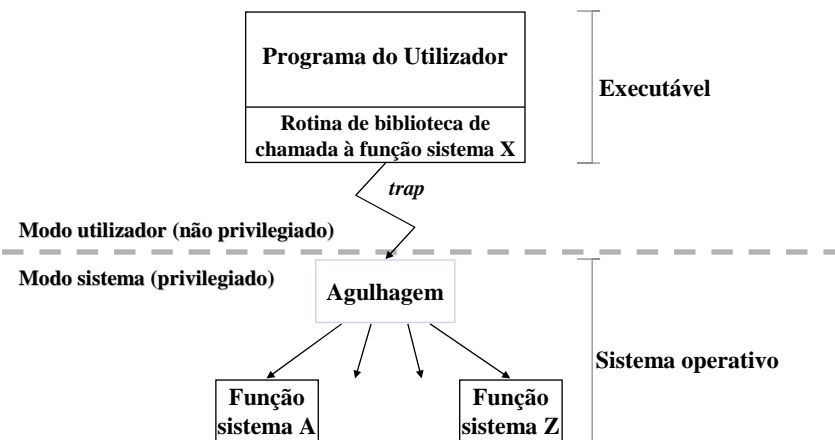
Modo Núcleo vs. Modo Utilizador

- Processos (aplicações) dos utilizadores não podem interferir com dados/execução do SO. Porquê?
- Devem também estar vedadas às aplicações dos utilizadores a execução de determinadas instruções. Exemplos?
- Ajuda do processador: espaços de endereçamento disjuntos, e diferentes **modos** de execução
- Barreira de protecção
 - Para “saltar” essa barreira, a aplicação tem de invocar uma interrupção de software

8/28/2003

José Alves Marques

Modo Núcleo vs. Modo Utilizador



8/28/2003

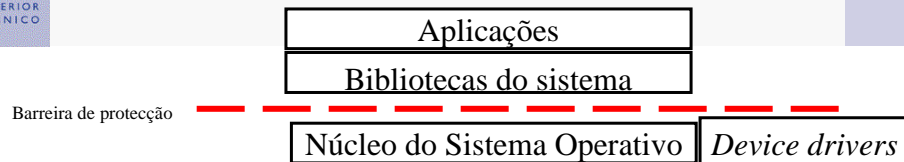
José Alves Marques

Organização do Sistema Operativo (Alternativas / Suporte à Evolução)

8/28/2003

José Alves Marques

Estrutura Monolítica



- Um único sistema
- Internamente organizado em módulos
- Estruturas de dados globais
- Problema: como dar suporte à evolução
 - Em particular, novos periféricos
- Solução para este caso particular: gestores de dispositivos (*device drivers*)
- Problemas?

8/28/2003

José Alves Marques

Sistemas em Camadas



- Cada camada usa os serviços da camada precedente
- Fácil modificar código de uma camada
- Mecanismos de protecção → maior segurança e robustez
- Influenciou arquitecturas como Intel
- Desvantagem principal?

8/28/2003

José Alves Marques

Micro-núcleo

- Propostas de investigação → separação entre:
- Um micro-núcleo de reduzidas dimensões e que só continha o essencial do sistema operativo:
 - Gestão de fluxos de execução - threads
 - Gestão dos espaços de endereçamento
 - Comunicação entre processos
 - Gestão das interrupções
- Servidores sistema que executavam em processos independentes a restante funcionalidade:
 - Gestão de processos
 - Memória virtual
 - Device drivers
 - Sistema de ficheiro

8/28/2003

José Alves Marques

Os Sistemas de Referência

8/28/2003

José Alves Marques

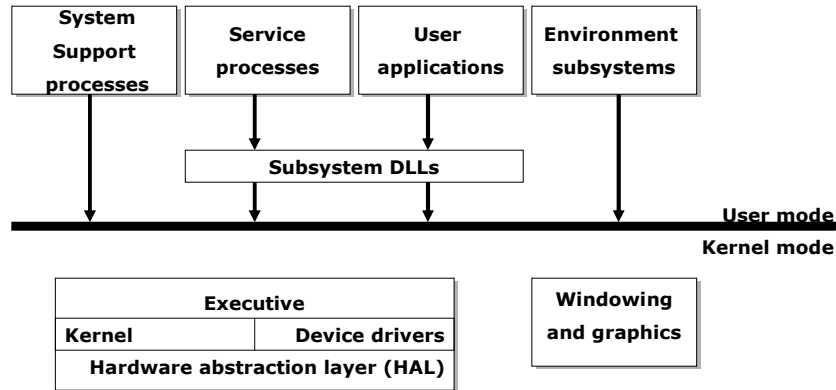
Unix

System calls				Interrupts and traps			
Terminal handing	Sockets		File naming	Map-ping	Page faults	Signal handling	Process Creation and Termination
Raw tty	Cooked tty	Network protocols	File systems	Virtual memory			
	Line disciplines	Routing	Buffer cache	Page cache		Process scheduling	
Character devices	Netwok device drivers	Disk device drivers		Process dispatching			
Hardware							
Gestão de Processos	Gestão de Memória	Gestão de Periféricos	Sistemas de Ficheiros	Comunicação			

8/28/2003

José Alves Marques

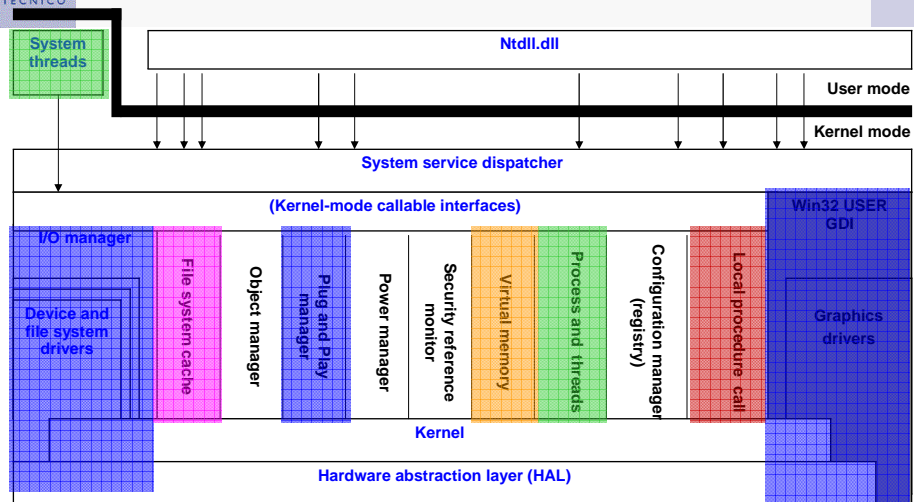
Windows 2000



8/28/2003

José Alves Marques

Windows 2000



8/28/2003

José Alves Marques

Novos conceitos, até agora:

- pseudo-parallelismo
- processos, threads
- comutação, escalonamento de processos
- comunicação entre processos
- execução assíncrona
- chamadas a funções sistema
- memória virtual

Próxima Aula: Pseudo-concorrência

