

## ESTRUTURA DO SISTEMA OPERATIVO

- Componentes do sistema operativo
- Serviços do sistema operativo
- Chamadas ao sistema
- Programas de sistema
- Estrutura do sistema operativo



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

## Componentes do S.O.

Pontos de vista de um sistema operativo:

- ♦ serviços que fornece
- ♦ interface que disponibiliza p/ utilizadores e programadores
- ♦ seus componentes e interligações

Componentes do sistema operativo:

- ♦ Gestão de processos
- ♦ Gestão da memória principal
- ♦ Gestão da memória secundária
- ♦ Gestão de entradas/saídas
- ♦ Gestão de ficheiros
- ♦ Gestão de rede
- ♦ Sistema de protecção



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

## Serviços do S.O.

- **Serviços que facilitam a tarefa do programador**
  - ♦ criação de programas (editores e *debuggers*)
  - ♦ execução de programas
  - ♦ acesso a dispositivos de I/O
  - ♦ acesso a ficheiros
  - ♦ comunicações
  - ♦ detecção de erros/falhas
  - ♦ contabilidade da utilização
  - ♦ ...
- **Serviços que garantem um funcionamento eficiente do sistema**
  - ♦ alocação de recursos
  - ♦ contabilidade do sistema
  - ♦ **protecção e segurança**
    - » protecção – impedir que um processo interfira com outros
    - » segurança – impedir acessos não autorizados



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

## Programas de sistema

- ♦ Fornecem um ambiente conveniente para o desenvolvimento de aplicações e execução de programas.
- ♦ Alguns dos programas de sistema são apenas interfaces simples para chamadas ao sistema.
- ♦ O interpretador de comandos é o programa de sistema mais importante.

### Programas:

- ♦ edição de texto
- ♦ manipulação de ficheiros e directórios
- ♦ informação de estado
- ♦ suporte a linguagens de programação
- ♦ carregamento e execução de programas
- ♦ comunicações
- ♦ ...

Do ponto de vista do S.O. não há distinção entre os programas do utilizador e os programas de sistema.



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

## Estrutura de um S.O.

Analizamos o S.O. do ponto de vista exterior.  
Vamos ver como ele é interiormente.

### Estrutura:

- ◆ monolítica
- ◆ em camadas
- ◆ *microkernel*



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

## Estrutura monolítica

- Primeiros S.O.'s.
- Não há estruturação ...
  - ◆ o S.O. é escrito como um conjunto de procedimentos cada um dos quais pode chamar qualquer outro.

programas do utilizador



hardware

sistema de ficheiros, memória virtual,  
*drivers* de I/O, controlo de processos,  
serviços de sistema, *swapping*,  
serviços de rede, protecção,  
tratamento de interrupções, ...

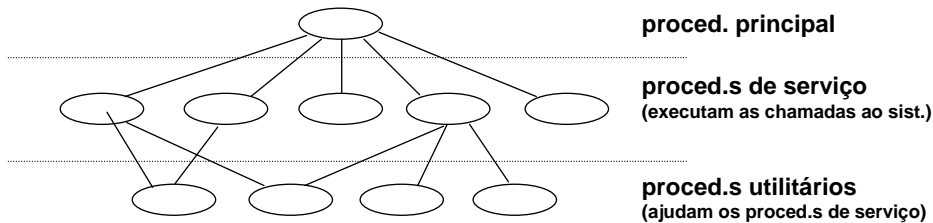


FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

## Estrutura monolítica

... ou há uma pequena estruturação:



### Dificuldades da estrutura monolítica:

- ♦ difícil de compreender
- ♦ difícil de modificar
- ♦ pouco fiável (um erro "em qualquer lado" pode provocar um *crash*)
- ♦ difícil de manter



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

## Estrutura em camadas

- O S.O. é dividido num certo número de camadas (níveis) cada qual construída por cima da anterior.
  - ♦ camada de mais alto nível - interface com o utilizador
  - ♦ camada 0 – *hardware*
- Sistema operativo modular
  - ♦ Para cada camada especificar a funcionalidade e as características.
  - ♦ É possível alterar a estrutura interna de cada camada desde que a interface com as outras camadas se mantenha inalterada.
  - ♦ Cada camada só usa funções e serviços das camadas inferiores.
  - ♦ Uma camada não necessita de "saber" como as operações da camada inferior são implementadas, mas apenas o que elas fazem.



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

**Dificuldades da estruturação em camadas:**♦ **Definição adequada das camadas**

- » porque cada camada só deveria poder usar as funções do nível inferior
- » o sistema de gestão de ficheiros deveria ser um processo numa camada superior à de gestão de memória virtual; por sua vez, esta deverá poder usar ficheiros (!)

♦ **Tende a ser menos eficiente do que outros tipos**

- » ex: para um programa do utilizador executar uma operação de I/O
  - executa uma chamada ao sistema
  - que faz um *trap* à camada de I/O
  - que chama a camada de ...
  - ... até chegar ao *hardware*

Os sistemas são frequentemente modelados como estruturas em camadas mas nem sempre são construídos dessa forma.

A estrutura em camadas foi usada pela 1ª vez no sistema THE.



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

**Exemplo teórico de hierarquia de camadas (Stallings):**


<u>Level</u>	<u>Name</u>	<u>Objects</u>	<u>Example Operations</u>
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printer, displays and keyboards	Create, destroy, open, close, read, write
9	File system	Files	Create, destroy, open, close, read, write
8	Communications	Pipes	Create, destroy, open, close, read, write



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

Sistemas Operativos		Estrutura do Sistema Operativo	
Level	Name	Objects	Example Operations
7	Virtual Memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive process, semaphores, ready list	Suspend, resume, wait, signal
<b>HARDWARE</b>			
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack	Mark stack, call, return
2	Instruction Set	Evaluation stack, micro-program interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement



MIEIC

Faculdade de Engenharia da Universidade do Porto


Sistemas Operativos
Estrutura do Sistema Operativo

## Estrutura baseada em *microkernel*

- **Tendência nos S.O.'s modernos:**
  - ◆ Deslocar código para as camadas superiores deixando um *kernel* mínimo.
  - ◆ O *kernel* implementa a funcionalidade mínima referente a
    - gestão de memória
    - gestão básica da *CPU*
    - comunicação entre processos
    - suporte de *I/O*
  - ◆ A restante funcionalidade do S.O. é implementada em proc.<sup>os</sup> de sistema que correm em modo de utilizador. Estes processos comunicam entre si através de mensagens (modelo cliente-servidor)

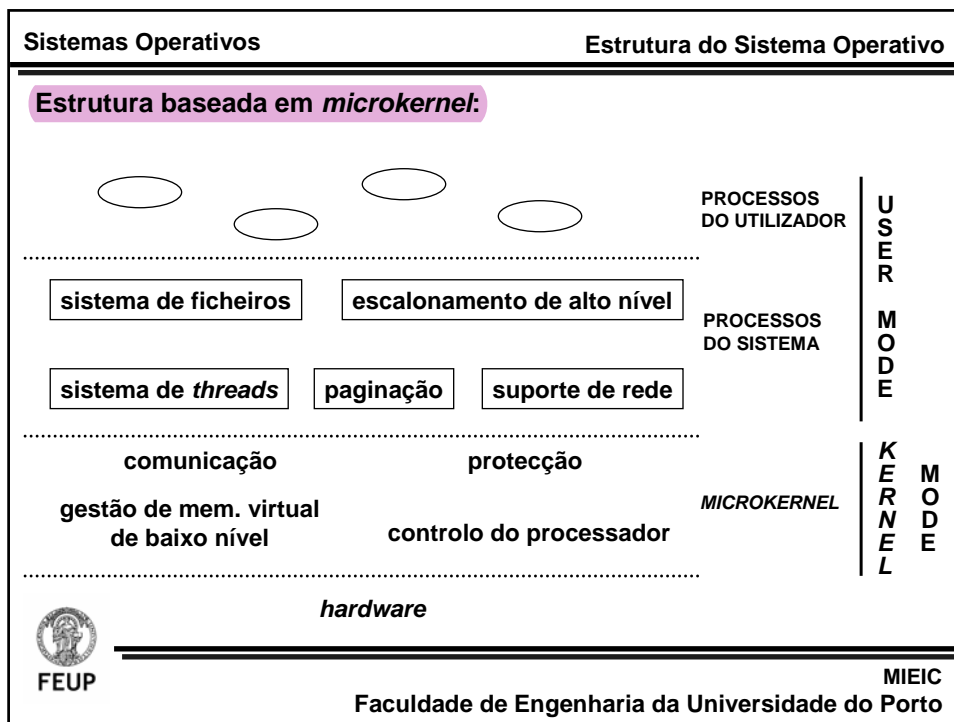
Primeiro sistema baseado em *microkernel*: Hydra (CMU, 1970)  
 Outros exemplos: Mach (CMU), Chorus (Unix-like, francês)

Windows NT/2000 - estrutura *microkernel* modificada;  
 ao contrário de uma arquitectura *microkernel* "pura"  
 muitas das funções de sistema fora do *microkernel*  
 executam em modo *kernel*, por razões de performance



MIEIC

Faculdade de Engenharia da Universidade do Porto



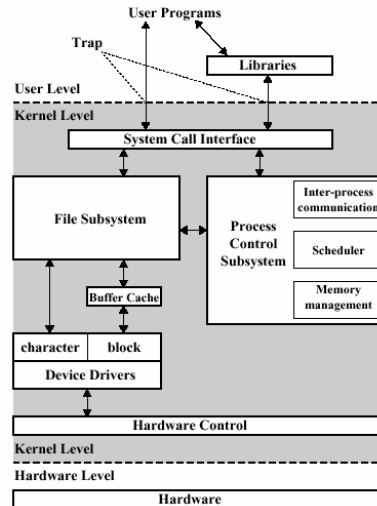
Sistemas Operativos	Estrutura do Sistema Operativo
<h2 style="margin: 0;">Estrutura de alguns S.O.'s</h2> <div style="background-color: #f0f0f0; padding: 10px; display: inline-block; margin: 10px auto; width: 150px;">Linux</div>	
<ul style="list-style-type: none"> <li>♦ <b>Estrutura essencialmente monolítica</b> (semelhante, em termos de grandes blocos à da pág. seguinte-&gt;UNIX, Bach 1986)</li> <li>♦ No entanto, é <b>relativamente modular</b>, internamente, de tal forma que a realização de modificações ou acrescentos não é considerada muito difícil.</li> <li>♦ O <b>kernel</b> está estruturado em 2 camadas essenciais:               <ul style="list-style-type: none"> <li>» uma camada independente da arquitectura do <i>hardware</i> (portável)</li> <li>» uma camada dependente da arquitectura do <i>hardware</i> (não portável)</li> </ul> </li> </ul> <p style="margin-top: 20px;">Inicialmente desenvolvido para processadores x86 Intel tem sido portado para sistemas baseados em Alpha, Motorola 680x0, PowerPC, SPARC, ...</p> <p>(estima-se que, em média, seja necessário reescrever 50000 linhas de código para portar o <i>kernel</i> do Linux + <i>drivers</i> para novas arquitecturas)</p>	
 <b>FEUP</b>	<b>MIEIC</b> Faculdade de Engenharia da Universidade do Porto

# Arquitectura do UNIX

(Bach,1986)

Algumas características:

- **Multitasking** e multiutilizador
- **Kernel** monolítico
- **Várias** arquitecturas consoante os fabricantes



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto

# Arquitectura do Windows 2000/XP

(Stallings,2002)

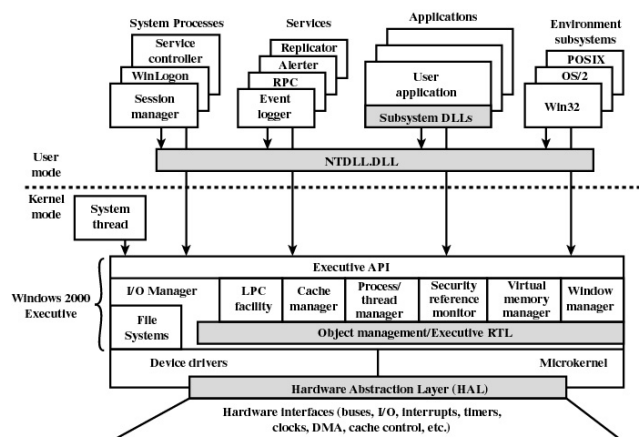


Figure 2.13 Windows 2000 Architecture



FEUP

MIEIC  
Faculdade de Engenharia da Universidade do Porto



## Windows 2000/XP

### Algumas características:

- ♦ Sistema *multitasking* com um único utilizador
- ♦ Suporta multiprocessamento simétrico
  - » vários processadores, partilhando memória e módulos de I/O
  - » todos os processadores podem executar qualquer tarefa
- ♦ Suporta *multithreading*
  - » dividir um processo em várias sequências de execução (v. adiante)
- ♦ Suporta interfaces de outros sistemas operativos (subsistemas)
- ♦ Estrutura *microkernel* modificada
  - » os serviços correm em modo privilegiado
- ♦ Camadas:
  - » *HAL - Hardware Abstraction Layer*
    - Cria uma interface comum para as várias plataformas de *hardware* suportadas
    - Facilita a portabilidade do S.O.
  - » *Microkernel*
  - » Serviços executivos
    - Fornecem uma interface para a camada que corre em modo utilizador (subsistemas)
  - » Subsistemas
    - Interface de programação para os programas de aplicação

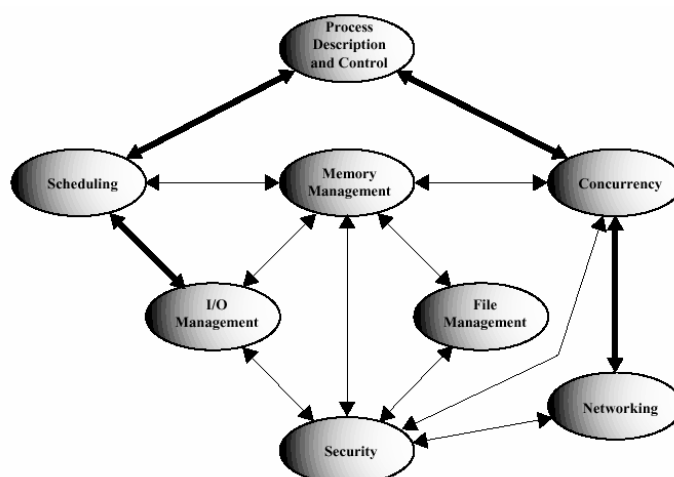


FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto

## A seguir: tópicos sobre S.O.'s



FEUP

MIEIC

Faculdade de Engenharia da Universidade do Porto