

# Sistemas Operativos: Introdução

February 23, 2012

# Sumário

## Introdução aos Sistemas Operativos

Organização de Computadores

Sistema Operativo

Abstracções Oferecidas por um SO

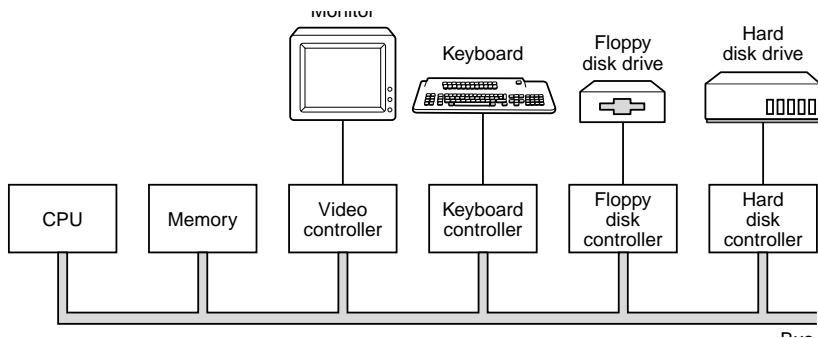
Serviços dum SO

Interfaces dum SO

Critérios de Qualidade

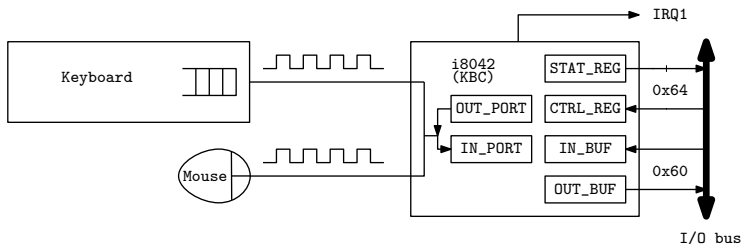
Classes de SO

# Componentes dum Computador Pessoal



- ▶ Desenvolver código directamente sobre o HW dum computador é uma tarefa hercúlea

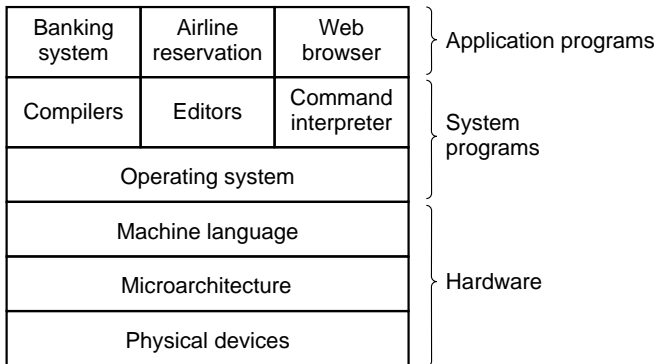
# Teclado do PC



- ▶ Quando uma tecla é premida o microcontrolador do teclado gera um *scancode* que é colocado num *buffer*
  - ▶ Quando uma tecla deixa de ser premida o microcontrolador gera outro *scancode*
- ▶ Este *scancode* é posteriormente transferido para o registo OUT\_BUF controlador do teclado (KBC)
- ▶ O KBC pode gerar uma interrupção
  - ▶ Alternativamente pode usar-se *polling*
- ▶ A identificação do carácter premido depende dos *scancodes* previamente recebidos, e do *mapa de teclado*

# Modelo em camadas dum sistema computacional

- ▶ O **sistema operativo** é uma “camada” de SW que assenta sobre o HW



- ▶ Cada nível define uma **máquina virtual**, excepto o nível mais baixo.

# Parentesis: Modelo em camadas

- ▶ Estratégia típica na resolução de problemas complexos (outro exemplo são as redes de computadores)
  - ▶ Suporta **abstracção**
- ▶ Vantagens deste modelo

**Decomposição** Um problema “intratável” é decomposto em problemas mais pequenos e solúveis

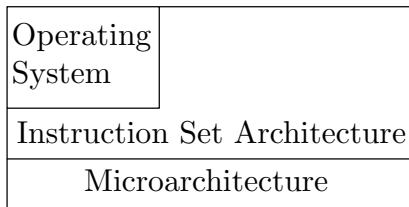
**Modularidade** É relativamente fácil acrescentar funcionalidades ou alterar a implementação, desde que se **preservem as interfaces**

# Instruction Set Architecture (ISA) Level

- ▶ Define o HW e o conjunto de instruções visíveis para um programador de *assembly*
- ▶ Compreende um conjunto de 50 a algumas centenas de instruções para:
  - ▶ Transferir dados entre os diferentes componentes;
  - ▶ Realizar operações aritméticas e lógicas;
  - ▶ Controlar o fluxo de instruções
- ▶ A este nível operações de entrada e saída são realizadas escrevendo e lendo registos dos controladores dos dispositivos de E/S

# Nível do Sistema Operativo

Define a interface tipicamente acessível a um programador.



- Oferece um conjunto de operações – **chamadas ao sistema (*system calls*)** – que fornecem um nível de abstracção muito mais conveniente.
  - A maioria das operações do nível ISA continua acessível.
  - Algumas contudo são escondidas, essencialmente por razões de segurança.



# O SO como um Gestor de Recursos

## Uma descrição alternativa do papel do SO

- ▶ Durante a sua execução, os programas fazem uso de recursos (CPU, memória, disco, ...).
- ▶ A maioria dos “computadores” executa várias aplicações (possivelmente) de diferentes utilizadores em simultâneo:
  - ▶ O SO gere os recursos dum computador, facilitando a sua partilha entre diferentes aplicações.
  - ▶ As aplicações não podem aceder directamente aos recursos (todo o acesso é mediado pelo SO):
    - ▶ Evita-se que as aplicações interfiram na gestão dos recursos
    - ▶ Protege-se os recursos afectados a cada utilizador de acessos não autorizados por outros utilizadores

# Abstracções Oferecidas por um SO

- ▶ Utilizador (em especial em sistemas interactivos);
- ▶ Processo;
- ▶ Ficheiro.

# Utilizador

- ▶ Essencial em sistemas interactivos.
- ▶ Inclui entre outros os seguintes atributos:
  - ▶ nome (*username*);
  - ▶ identidade (*userid*);
  - ▶ grupos (*groupname* e *groupid*).
- ▶ Em sistemas multiutilizador, o conceito de utilizador é central para protecção de recursos (p.ex. ficheiros).
- ▶ O utilizador pode estar associado não só a uma pessoa como a uma função, p. ex. correio electrónico.

# Processo

- ▶ Representa um **programa em execução**.
  - ▶ Um programa é um **objeto passivo**, tipicamente guardado em disco ou outros dispositivos de armazenamento de dados
  - ▶ Um processo é um **objeto activo**, cujo estado varia à medida que é executado
- ▶ Actualmente, quase todos os sistemas operativos são **multi-processo** (Linux, Windows XP, Windows Vista):  
Um computador pode executar vários programas ao “mesmo tempo”.
- ▶ Um processo está associado a um utilizador, o seu **dono**:  
aquele que invoca o programa correspondente.  
A identidade do utilizador **dono** do processo determina os **recursos** a que um processo pode aceder, bem como o tipo de operações que pode realizar sobre esses recursos.

# Ficheiros

- ▶ Representam uma **fonte/poço de informação**
  - ▶ Para o utilizador não técnico são uma abstração do disco (e outros dispositivos de armazenamento de dados)
  - ▶ Podem contudo abstrair outros **dispositivos de E/S**
- ▶ Suportam três operações: **leitura**, **escrita** e **execução**
- ▶ Tipicamente organizados numa forma hierárquica, usando **directórios**:

Ficheiros que contêm outros ficheiros.

- ▶ Cada ficheiro/directório tem um utilizador que é o seu **dono**
  - ▶ Usado para controlo de acesso

# Serviços dum SO

- ▶ Suportam as abstracções oferecidas pelo SO.
- ▶ Gerem os recursos do sistema.
- ▶ Por exemplo:
  - ▶ gestão de processos;
  - ▶ gestão de memória;
  - ▶ gestão de ficheiros e directórios;
  - ▶ gestão de dispositivos de entrada/saída;
  - ▶ comunicação através de rede;
  - ▶ detecção e notificação de eventos;
  - ▶ autenticação e controlo de acesso.

# Serviços de Gestão de Processos

- ▶ Criação e destruição de processos.
- ▶ Suspensão e retoma (continuação) dum processo.
- ▶ Mecanismos para:
  - ▶ sincronização de processos;
  - ▶ comunicação entre processos.
- ▶ Alteração do *dono* dum processo.

Um processo é essencialmente um programa em execução, assim estes serviços são de suporte à execução de programas.

# Serviços de Gestão de Memória Principal

- ▶ Programas em execução exigem memória:
  - ▶ para o código;
  - ▶ para os dados.
- ▶ O SO tem que gerir a memória:
  - ▶ Alocação e libertação de memória.
  - ▶ Mapeamento de ficheiros na memória principal.
  - ▶ Outras funções usadas internamente pelo SO:  
partilha da memória física entre diferentes processos e o próprio SO.



# Serviços de Gestão de Ficheiros/Directórios

- ▶ Criação e destruição de ficheiros/directórios.
- ▶ Leitura do conteúdo dum ficheiro (listagem dos ficheiros/directórios num directório).
- ▶ Alteração do conteúdo dum ficheiro/directório.
- ▶ Alteração do nome dum ficheiro/directório.
- ▶ Alteração de atributos (e.g. dono, permissões) dum ficheiro/directório.
- ▶ Vizualização de atributos dum ficheiro/directório.

# Serviços de Comunicação em Rede

- ▶ Suportam a comunicação entre processos em diferentes computadores:
  - ▶ estabelecimento e terminação de canais de comunicação;
  - ▶ configuração e inspecção do estado do canal de comunicação;
  - ▶ transmissão e recepção de mensagens.
- ▶ De facto, estes mesmos serviços podem ser usados na comunicação entre processos no mesmo computador (sem carta de rede).

# Serviços de Detecção/Notificação de Eventos

- ▶ Usados para comunicação entre:
  - ▶ processos;
  - ▶ o SO e um processo.
- ▶ Tipo de eventos detectados pelo SO:
  - ▶ expiração de temporização;
  - ▶ erros internos na execução de processos.

# Serviços “internos”

- ▶ Tipicamente, não são directamente acessíveis às aplicações.
  - ▶ O SO é frequentemente organizado em camadas

**Entrada/Saída:** permitem aceder a dispositivos de E/S:

- ▶ portas série;
- ▶ cartas gráficas;
- ▶ discos;
- ▶ cartas de rede.

duma forma uniforme

**Gestão de discos** incluindo:

- ▶ alocação e libertação de blocos;
- ▶ escalonamento dos pedidos.

# Serviços de Autenticação e Controlo de Acesso

**autenticação:** validar a identidade dum utilizador. É essencial para proteger o acesso a recursos.

**controlo de acesso:** limitar o acesso a recursos apenas às entidades autorizadas a fazê-lo:

- ▶ pressupõe autenticação;
- ▶ é um serviço transversal: abrange todos os outros serviços.

# Quais os Serviços dum SO?

- ▶ Será que um *web browser* faz parte do SO?
- ▶ E um *media player*?
- ▶ E um *anti-virus*?

# Interfaces dum SO

**Problema:** Como se acede aos serviços do sistema operativo?

**Solução:** Através das suas interfaces:

**Programática** Consiste num conjunto de operações designadas por **chamadas ao sistema**

- ▶ Para cada tipo de serviço, o SO oferece um conjunto de chamadas ao sistema.

**Operacional** Pode ser:

- ▶ Textual
- ▶ Gráfica

Faz uso da interface programática.

# Critérios de Qualidade

## Desempenho

## Segurança

- ▶ Garantir que os acessos autorizados são realizados
- ▶ Impedir os acessos não autorizados sejam realizados

## Robustez

**Fiabilidade** probabilidade de não falhar

**Disponibilidade** probabilidade de prestar serviço

## Qualidade das Interfaces

- ▶ Completude
- ▶ Simplicidade



# Critérios de Classificação de SO

**Tempo-Real vs. Não Tempo-Real** Sistemas de tempo-real podem adicionalmente ser classificados em:

*Hard Real-Time*

*Soft Real-Time*

**De acordo com o computador** Nomeadamente sistemas embebidos

- ▶ Frequentemente têm requisitos de tempo real
- ▶ Uma área bastante activa, especialmente SO para *smart-phones*

**Código aberto**

# Leitura Adicional

- ▶ Secções 1.1, 1.2 e 1.4 de José Alves Marques e outros, *Sistemas Operativos*, FCA - Editora Informática, 2009
- ▶ Secções 1.1, 1.5 de Andrew Tanenbaum, *Modern Operating Systems*, 2nd Ed.
- ▶ Secções 1.1, 1.2, 1.6, 1.7, 1.8, 1.9, 2.1 e 2.2 de Silberschatz e outros, *Operating System Concepts*, 7th Ed.
- ▶ Outra documentação (transparências e enunciados dos TPs) na [página da disciplina](#)