

HARDWARE DE UM SISTEMA DE COMPUTAÇÃO

- Revisão de conceitos básicos sobre *hardware*
 - » Processador
 - » Interrupções
 - » Processamento de E/S
 - » Memória
- Protecção do *hardware*
- Arranque do sistema (*booting*)



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Elementos básicos do *hardware*

A maior parte dos computadores são baseados na arquitectura de Von Neumann.

4 elementos principais do *hardware*:

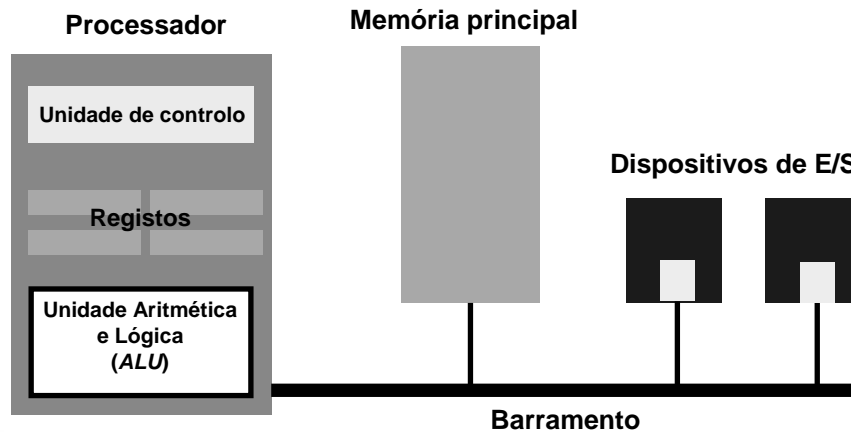
- Processador ou Unidade de Processamento Central (*Central Processing Unit - CPU*)
 - » controla a operação do computador
 - » processa os dados
- Memória principal
 - » armazena programas e dados
- Dispositivos de E/S-Entrada/Saída (*I/O-Input/Output*)
 - » transferem os dados entre o computador e o exterior
- Ligações entre os outros elementos (barramentos)



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Arquitectura básica do *hardware*



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Registos do processador

Program Counter - PC

Contém o endereço da próxima instrução.

Instruction Register - IR

Contém a instrução actual.

Processor Status Word - PSW

Contém informação acerca do estado do processador, interrupções, *flags* (*carry, zero, overflow, ...*)

Stack Pointer - SP

Aponta para o topo da *stack*.

Registos do utilizador

Usados para vários fins, na programação.

Outros

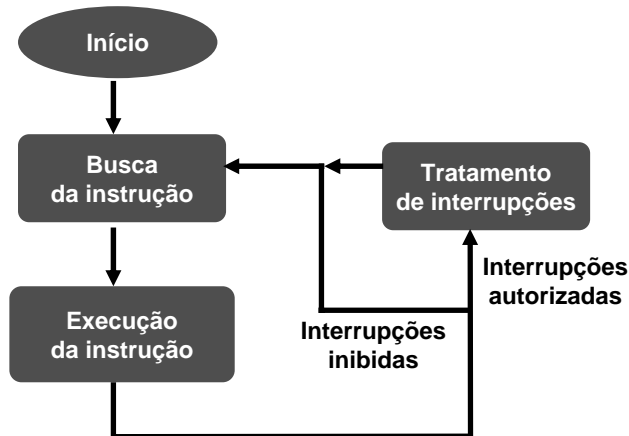
Gestão de memória, etc.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Execução de instruções



FEUP

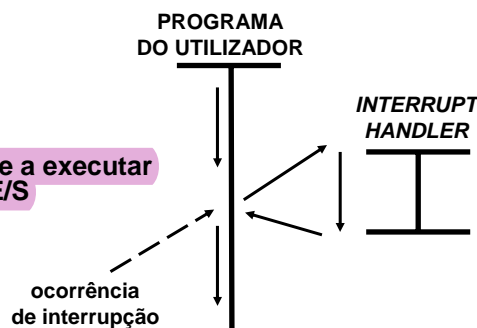
MIEIC
Faculdade de Engenharia da Universidade do Porto

Interrupções

Uma interrupção é um mecanismo que permite que o processamento normal de um processador seja interrompido.

As interrupções são usadas para aumentar a eficiência, especialmente quando se usam componentes que operam a velocidades diferentes.

Permitem que o processador continue a executar enquanto decorre uma operação de E/S
→ base da multiprogramação



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Classes de interrupções

Programa

geradas por uma condição que resulta da execução de uma instrução
(ex: *overflow*, divisão por zero, ref.^a a memória protegida, ...)

Temporizador (*timer*)

geradas por um temporizador;
permite que o SO execute certas tarefas regularmente.

E/S

geradas por um controlador de E/S
para assinalar o fim de uma operação ou certos erros.

Falha de *hardware*

geradas por uma falha
(ex: alimentação, erro de paridade da memória, ...)



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Processamento de interrupções

A rotina a executar
em resposta a uma determinada interrupção
é determinada com base num vector de interrupções.

Cada plataforma de *hardware* tem
um procedimento particular
para tratar uma determinada interrupção.

Deve existir qualquer forma de *interrupt handler*.

Combinação de responsabilidade
do SO e do *hardware* no *interrupt handler*.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Processamento de uma interrupção

Hardware

- 1) Ocorrência da interrupção
- 2) O processador termina instrução actual
- 3) O processador assinala aceitação da interrupção
- 4) O processador guarda a *PSW* e o *PC* na *stack*
- 5) O processador carrega *PC*
c/o endereço da rotina de tratamento da interrupção

Software

- 6) Guardar restante informação do estado do processo
(registos do processador, ...)
- 7) Processar a interrupção (executar rotina)
- 8) Restaurar informação do estado do processo
- 9) Restaurar *PSW* e *PC*



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Interrupções múltiplas

O que acontece quando
uma interrupção interrompe outra interrupção ?

- **Processamento sequencial**
 - » inibir as interrupções
durante o processamento de uma interrupção
- **Processamento embutido**
 - » algumas interrupções
são mais importantes do que outras
⇒ prioridades de interrupção



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Interrupções múltiplas

Processamento sequencial

PROGRAMA DE UTILIZADOR

INTERRUPT HANDLER X

INTERRUPT HANDLER Y



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Interrupções múltiplas

Processamento embutido

PROGRAMA DE UTILIZADOR

INTERRUPT HANDLER X
(ex: impressora)INTERRUPT HANDLER Y
(ex: rede)

FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Entrada / Saída

A gestão eficiente das E/S's
é uma das principais responsabilidades de um SO.

3 mecanismos principais para executar E/S's

- E/S programada (ou *polling*)
- E/S guiada por interrupção
- Acesso directo à memória
(*Direct Memory Access - DMA*)



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

E/S programada

- É o módulo de E/S que controla a acção, não o processador
 - O módulo de E/S indica o seu estado num *Status Register*
 - Não há interrupções
 - O processador está sempre ocupado a verificar o estado do módulo de E/S
- Problemas quando o dispositivo de E/S é lento.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

E/S guiada por interrupção

- O processador ainda tem de intervir na operação de transferência dos dados
- Quando a E/S se completar é gerada uma interrupção pelo dispositivo de E/S
- O processador fica livre durante a operação do dispositivo de E/S
- Mais eficiente do que a E/S programada mas o processador ainda é responsável pela transferência de dados entre a memória e o dispositivo de E/S.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Acesso Directo à Memória (DMA)

Necessário um controlador de DMA ligado ao barramento do sistema.

Quando é necessário fazer E/S

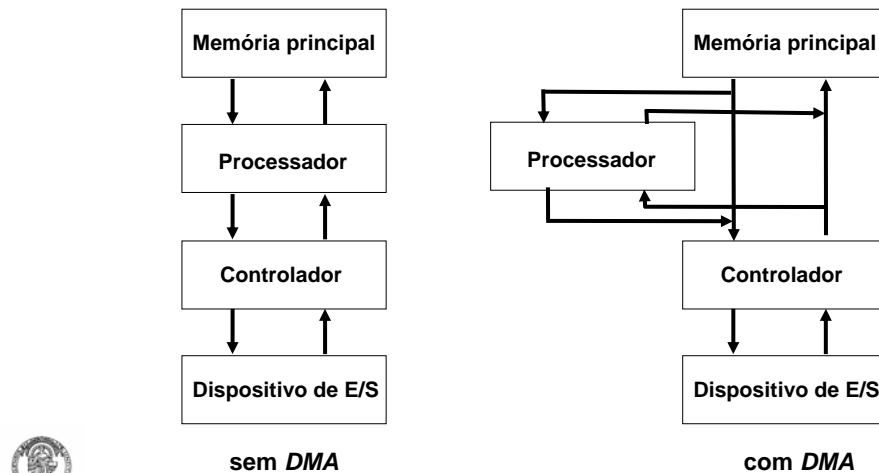
- o processador informa o controlador do dispositivo de E/S do que pretende fazer e onde está ou vai ficar a informação a transferir
- o processador continua a executar outras instruções
- o dispositivo de E/S transfere a informação directamente de/para a memória
- quando o DMA termina é gerada uma interrupção



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

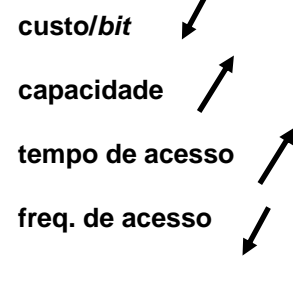
Entrada / saída



Memória

Hierarquia

- Registos
- Cache
- Memória principal
- Cache de disco
- Disco magnético
- Disco óptico / Fita magnética



Localidade de referência

- As referências do processador à memória tendem a ocorrer em posições contíguas durante certos intervalos de tempo → utiliz. cache



Protecção do *hardware*

Aumentar a utilização do sistema

- ⇒ partilha do sistema
- ⇒ vários programas a executar em simultâneo
- ⇒ protecção

O SO deve impedir que um programa incorrecto ou "mal intencionado" impeça os outros programas de executar.

Alguns erros de programação são detectados pelo *hardware*. Normalmente estes erros são tratados pelo SO.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Proteccção do *hardware* (cont.)

Protecção

- duplo modo de operação
 - » modo utilizador
 - » modo supervisor / sistema / monitor / privilegiado (instruções privilegiadas)
- protecção de E/S's
- protecção da memória
- protecção do processador



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Duplo modo de operação

O *hardware* tem de suportar pelo menos 2 modos de operação.

O computador arranca em modo supervisor.

O SO é carregado e

dá início aos processos do utilizador em modo utilizador.

Quando ocorre uma excepção (*trap* ou interrupção)

o *hardware* comuta para modo supervisor.

O sistema comuta sempre para modo utilizador

antes de ceder o controlo a um programa do utilizador.



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Chamadas ao sistema

É o método usado para
um processo pedir uma acção ao sistema operativo.

- implicam a comutação do modo utilizador para o modo supervisor
- usam, em geral, interrupções por *software*
- os parâmetros da chamada são, geralmente, passados nos registos do processador



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Protecção das E/S's

Impedir um utilizador de executar E/S "ilegal"

- definir todas as operações de E/S como privilegiadas
 - » os utilizadores não conseguem fazer E/S directamente, só através do SO



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Protecção da memória

Fundamental proteger

- vector de interrupções
- rotinas de serviço de interrupção

Protecção da área de memória de cada utilizador

- feita por *hardware*
 - » 2 registos determinam a gama de endereços válidos a que um programa pode aceder
 - » estes registos só podem ser manipulados pelo SO



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Protecção do processador

Impedir que um programa do utilizador tome conta do procesador indefinidamente (ex: ciclo infinito) e não retorne o controlo ao SO.

Usar um temporizador que após um período especificado interrompe o programa em execução.

As instruções de manipulação do temporizador são privilegiadas.

O temporizador também é usado para

- implementar *time-sharing*
- manter actualizada a hora do sistema



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Requisitos de *hardware* para multiprogramação

Um SO com multiprogramação necessita de suporte de *hardware*:

- temporizador
- *hardware* de *DMA*
- mecanismo de interrupções com prioridades
- duplo modo de operação do processador
- mecanismo de protecção da memória
- mecanismo de atribuição dinâmica de endereços
- ...



FEUP

MIEIC
Faculdade de Engenharia da Universidade do Porto

Arranque de um PC (*booting*)

- Após o *POST (Power On Self Test)* que verifica o estado do *hardware* é feito o *reset* do processador
- O processador procura uma instrução no endereço 0xFFFFFFF0
- A instrução neste endereço é um salto para o início do *BIOS*, em *ROM*
- O *BIOS* determina, na sua configuração, qual o *boot device* (disquete/disco/...)
- O *BIOS* lê o *MBR-Master Boot Record* (primeiro sector) do *boot device*; o *MBR* contém informação acerca das partições existentes no disco e o endereço do *boot sector*
- O *BIOS* carrega, em RAM, um pequeno programa, contido neste sector que poderá, por sua vez executar outros programas, cuja execução culminará com o carregamento do sistema operativo
- O sistema operativo executa vários procedimentos de inicialização
- Alguns processos começam a ser executados

