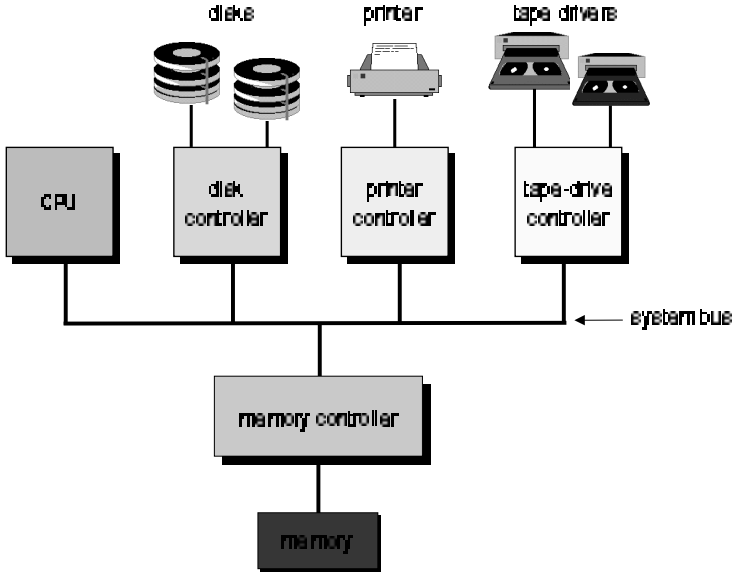
Sistemas Operacionais

Estruturas Básicas dos Sistemas de Computação

Computer-System Architecture



Computer-System Operation

- I/O devices and the CPU can execute concurrently.
- Each device controller is in charge of a particular device type.
- Each device controller has a local buffer.
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller.
- Device controller informs CPU that it has finished its operation by causing an *interrupt*.

Interrupções

- SINAL (HW e/ou SW) ASSÍNCRONO QUE SUSPENDE A EXECUÇÃO DO PROCESSO EM EXECUÇÃO NO PROCESSADOR, FAZENDO-O DAR ATENÇÃO A OUTRO PROCESSO.
- QUALQUER PROCESSO PODE SER INTERROMPIDO NO DECORRER DE SEU PROCESSAMENTO.
- A INTERRUPÇÃO NÃO PODE ALTERAR O RESULTADO FINAL DO PROCESSO (a menos que seja mensageira de alguma ordem).
- A INTERRUPÇÃO NÃO PODE OBRIGAR O PROCESSO A EXECUTAR NENHUM RETRABALHO.

Common Functions of Interrupts

- Interrupts transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*.
- A *trap* is a software-generated interrupt caused either by an error or a user request.
- An operating system is *interrupt* driven.

Tipos de Interrupção

- Hardware
  - ENTRADA/SAÍDA
    - acesso a disco, pacote chegou pela rede
    - teclado, mouse
  - TEMPO
    - estouro de temporizadores, intervalos
  - FALHAS DE MÁQUINA
    - paridade, falta de energia
- Software
  - chamadas ao núcleo do sistema operacional
    - também chamadas de
      - interrupção de software
      - trap (armadilha)
    - serviços do SO
  - exceções
    - overflow, div 0,
    - instrução ilegal

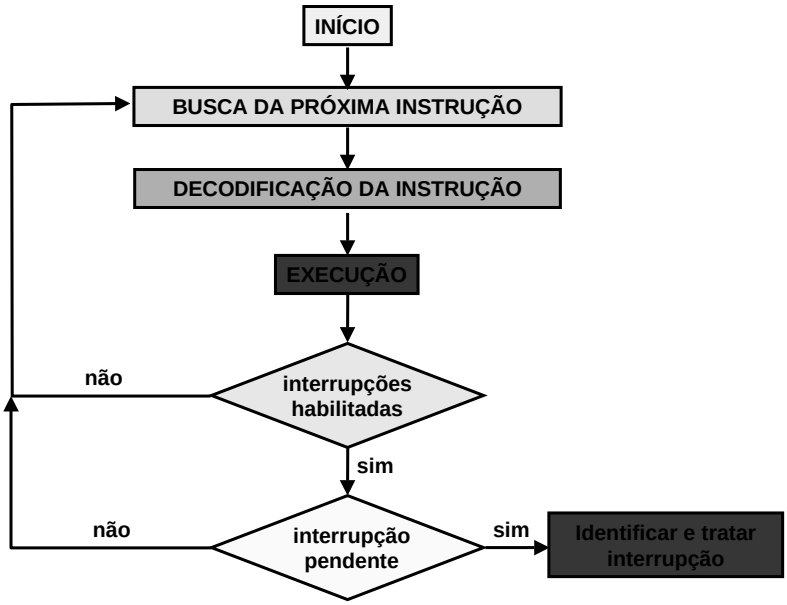
Prioridade e Controle

- **Arquitetura de interrupções pode ter um esquema de prioridades**
  - **interrupções mais prioritárias podem "interromper" as menos prioritárias**
  - **pode causar problema de inversão de prioridades ou deadlock**
- **Desabilitáveis (maskable)**
  - **interrupções que podem ser desabilitadas**
  - **garante que outras interrupções não serão atendidas**
- **Não-desabilitáveis (non-maskable)**
  - **interrupções de alta prioridade que não podem ser interrompidas**
    - **pane na máquina**
    - **paridade, falta de energia**

Tratamento de Interrupções (*Interrupt Handling*)

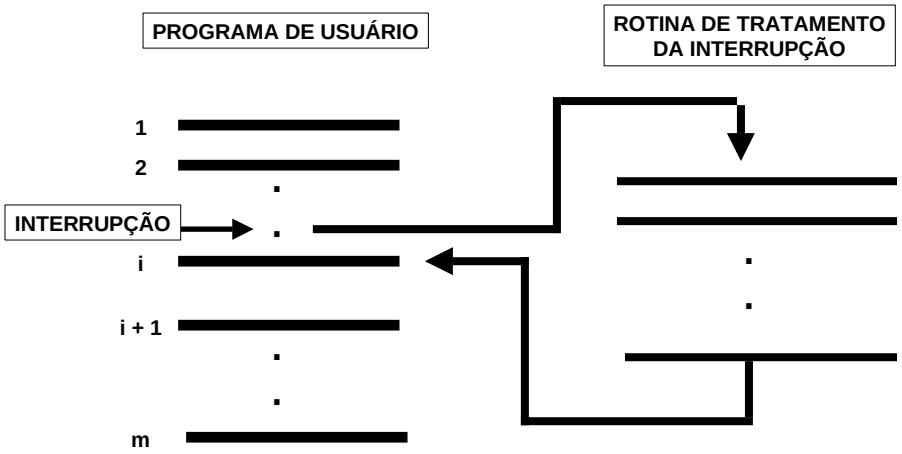
- Mecanismos para identificar o tipo de interrupção
  - quem gera a interrupção já envia a ident. (*vectored interrupt system*)
  - o SO "pergunta"quem interrompeu (*poll*)
- identificação da rotina de tratamento
  - consulta a tabela ou endereço fixo
  - leva ao endereço do início da rotina
- rotina de tratamento de interrupção (*handler*)
  - Termina de salvar contexto do processo interrompido
  - Separate segments of code determine what action should be taken for each type of interrupt
  - Faz o serviço
    - Ex.: transfere dados entre dispositivos, memória ou CPU
  - Restaura o contexto do processo interrompido (pilha)
    - ou "chama" o escalonador para decidir "quem" volta

O CICLO DE INSTRUÇÃO DA CPU COM INTERRUPÇÃO

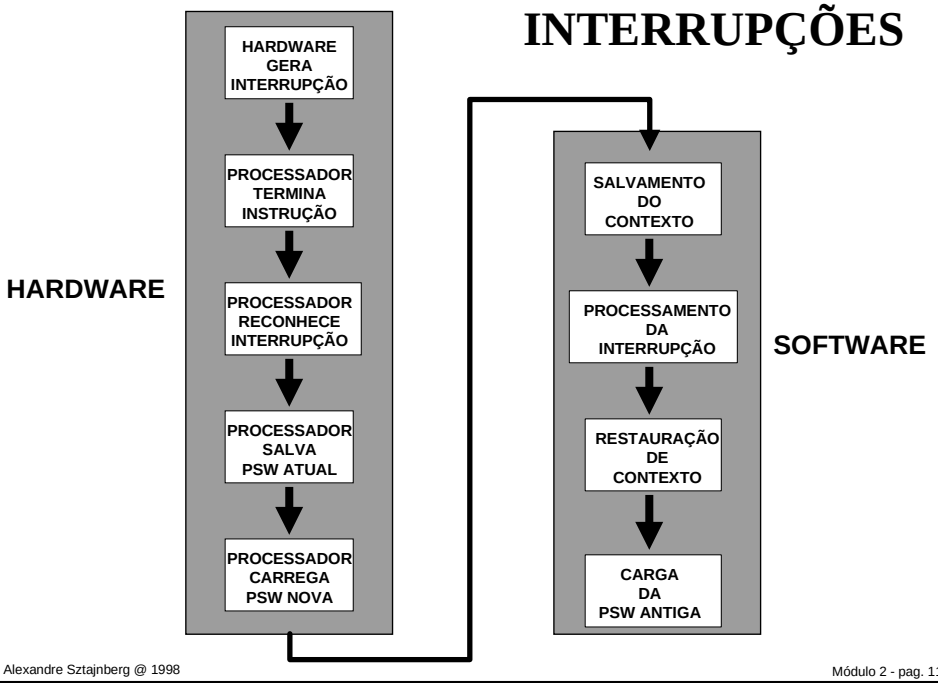


INTERRUPÇÕES

➔ A INTERRUPÇÃO E O CICLO DE INSTRUÇÃO

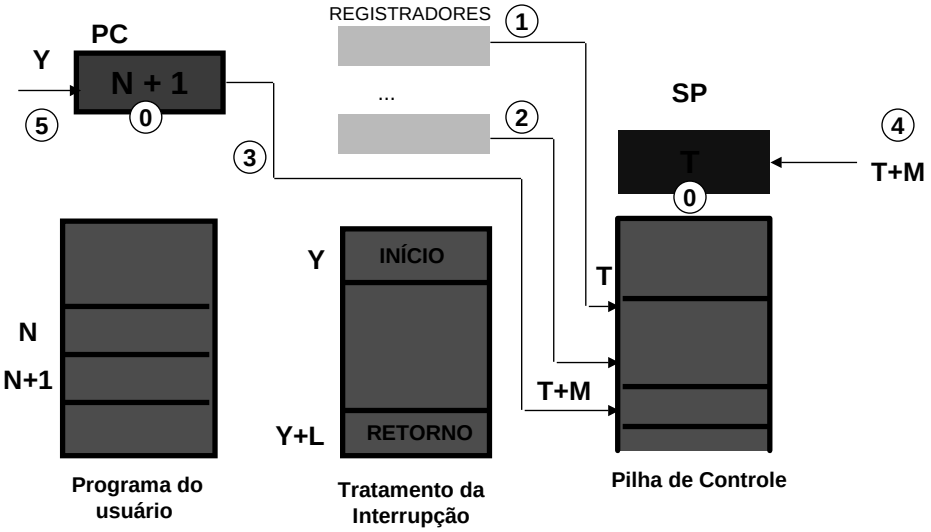


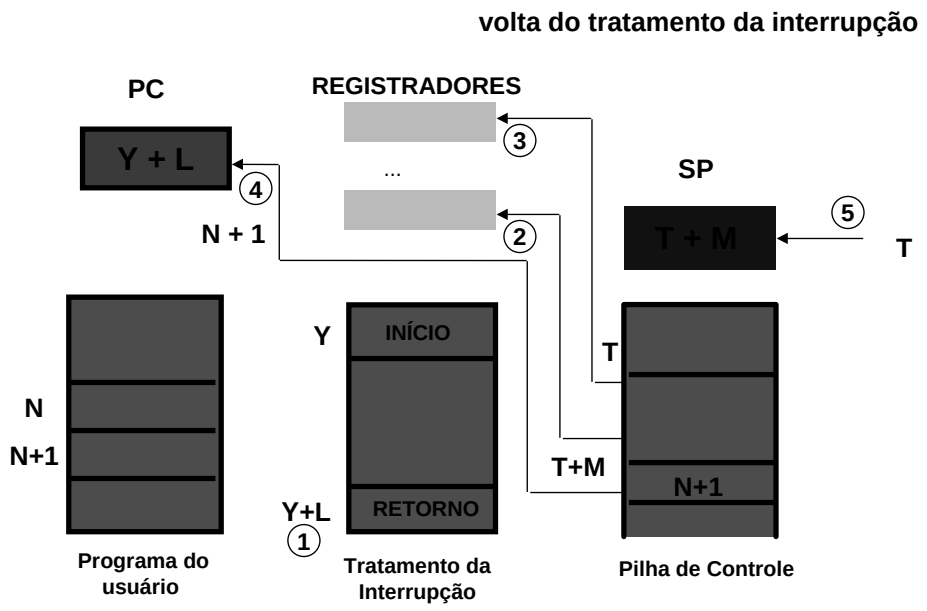
INTERRUPÇÕES



ALTERAÇÕES NA MEMÓRIA E NOS REGISTRADORES DEVIDAS AO PROCESSAMENTO DA INTERRUPÇÃO

- a interrupção ocorre depois da instrução na posição N





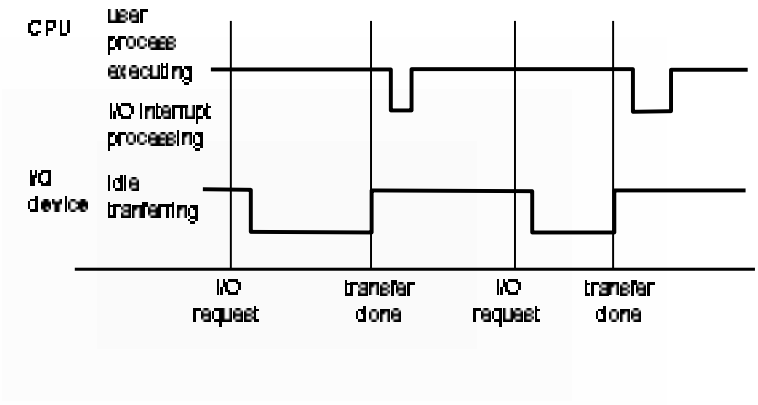
I/O Structure

- After I/O starts, control returns to user program only upon I/O completion.
  - wait instruction idles the CPU until the next interrupt
  - wait loop (contention for memory access).
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing.
- After I/O starts, control returns to user program without waiting for I/O completion.
  - *System call* – request to the operating system to allow user to wait for I/O completion.
  - *Device-status table* contains entry for each I/O device indicating its type, address, and state.
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt.

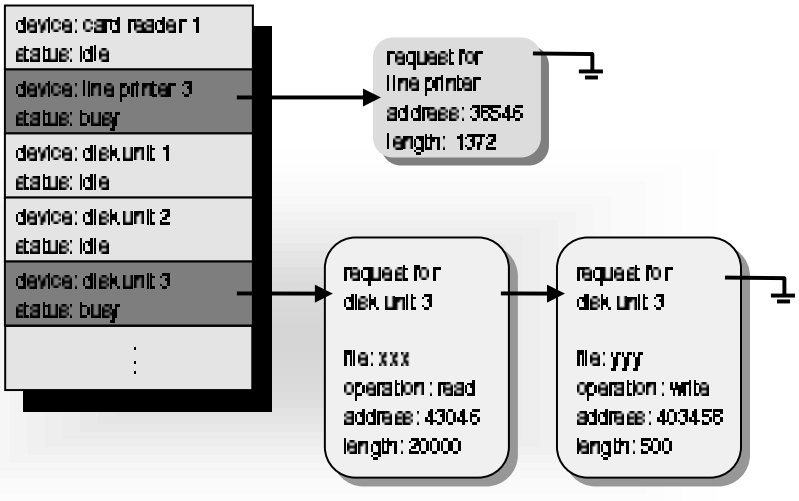
ENTRADA/SAÍDA

- **NORMALMENTE É O PROCESSADOR QUE CONTROLA A OPERAÇÃO DA MÁQUINA.**
- **INTERAÇÃO COM OS DISPOSITIVOS DE E/S**
  - **TROCAM DADOS DIRETO COM O PROCESSADOR.**
  - **TROCAM DADOS DIRETO COM A MEMÓRIA - DMA**
- **NO DMA O PROCESSADOR TRANSFERE O CONTROLE TEMPORARIAMENTE A UM PROCESSADOR AUXILIAR.**
- **TRANFERÊNCIA DE CONTROLE É REALIZADA ATRAVÉS DE INTERRUPÇÕES.**

Interrupt Time Line For a Single Process Doing Output

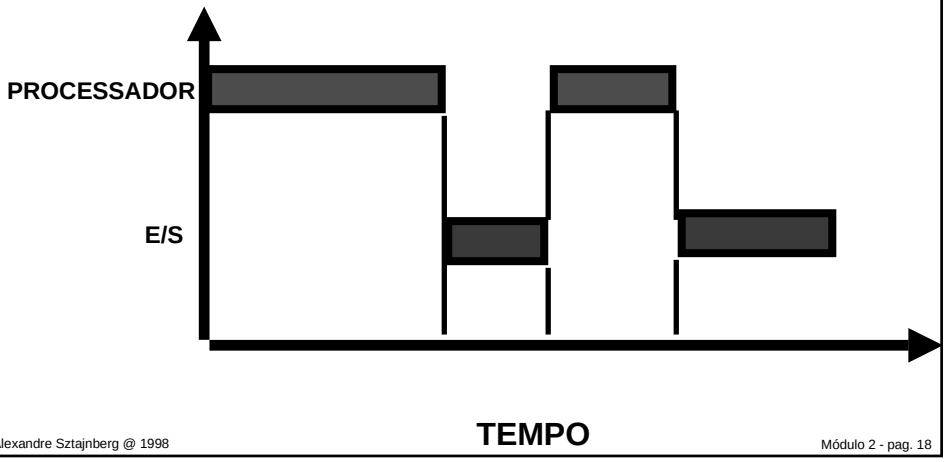


Device-Status Table



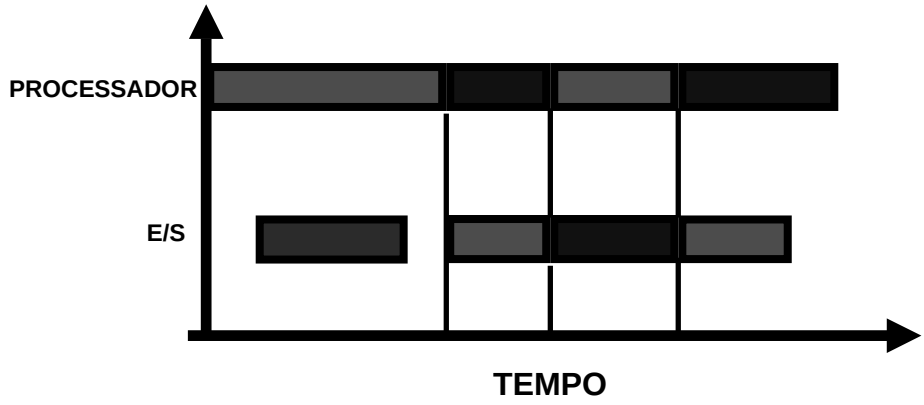
MULTIPROGRAMAÇÃO

- **EXECUÇÃO DE UM PROCESSO SEM MULTIPROGRAMAÇÃO**



MULTIPROGRAMAÇÃO

- **EXECUÇÃO DE PROCESSOS COM MULTIPROGRAMAÇÃO**
  - **princípio do Intercalamento**
  - **chaveamento de processos (interrupção)**

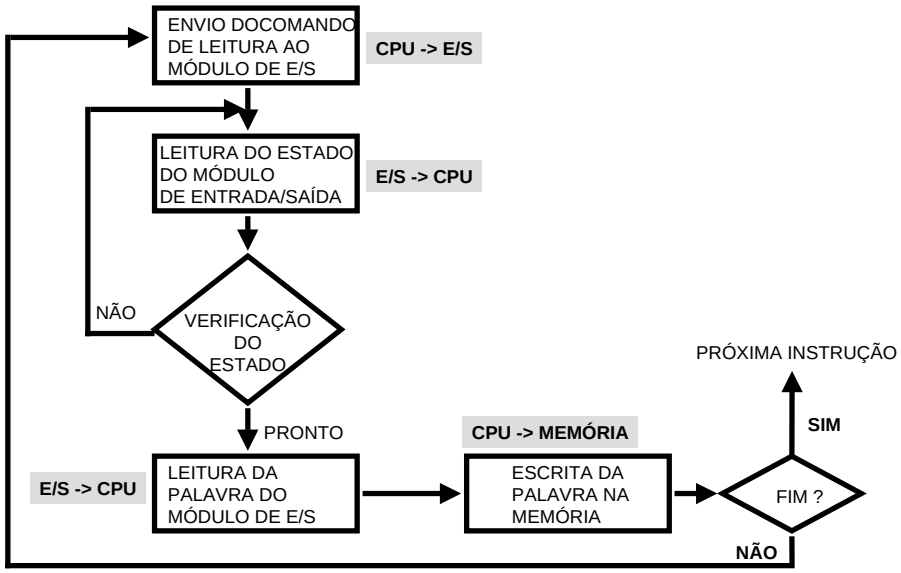


TÉCNICAS DE ENTRADA/SAÍDA

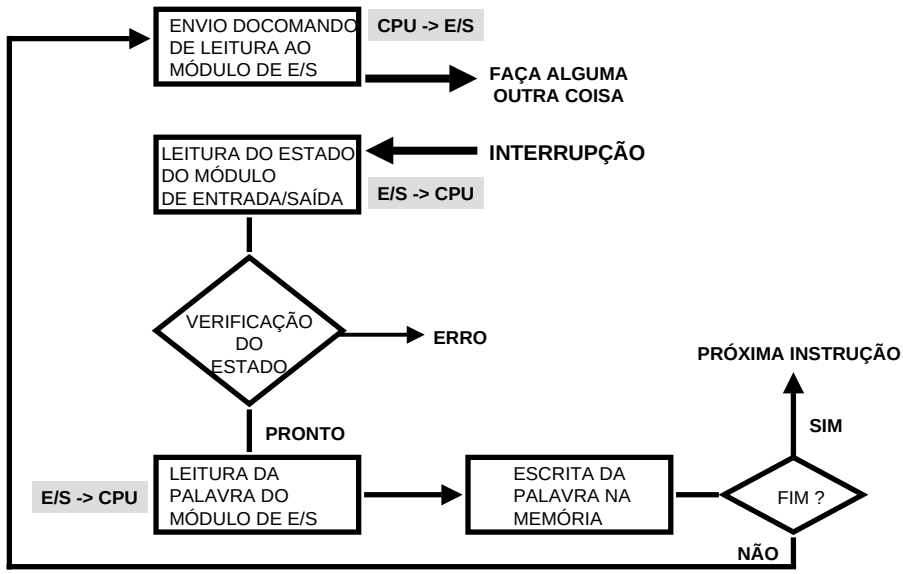
- **E/S PROGRAMADA**
  - **espera ocupada (busy wait)**
  - **polling (um pouco mais eficiente)**
- **E/S DIRIGIDA POR INTERRUPÇÕES**
- **E/S POR ACESSO DIRETO À MEMÓRIA**
  - **CPU só trabalha no início e no fim**
  - **área de buffer reservada para os dados**
- **E/S POR CANAL DEDICADO**

(ex: IBM *mainframe*)

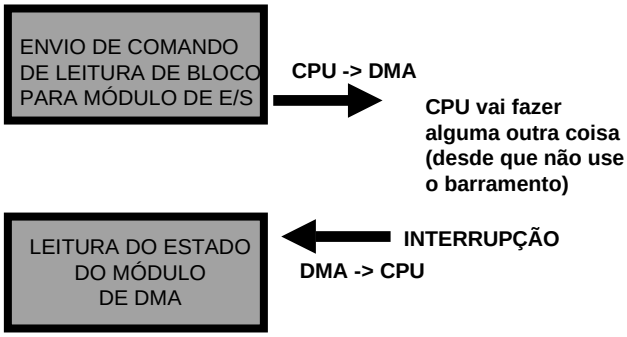
E/S PROGRAMADA



E/S DIRIGIDA POR INTERRUPÇÃO

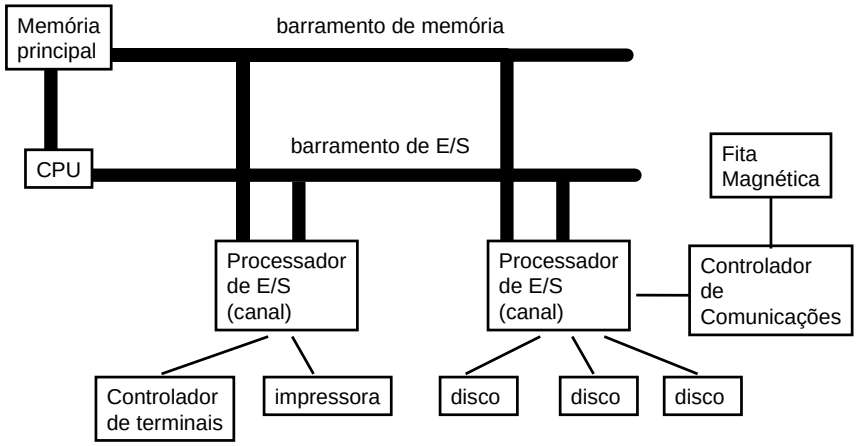


ACESSO DIRETO À MEMÓRIA - DMA



E\S - Canal

➔ **a diferença de um *mainframe*...**



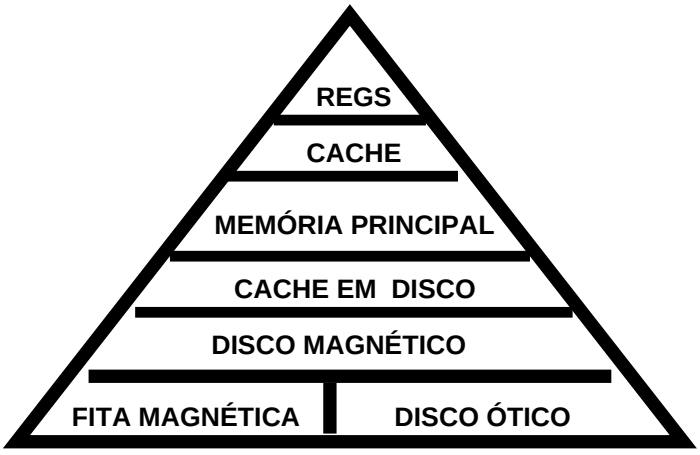
HIERARQUIA DE MEMÓRIA

➔ **CONCEITO DE LOCALIDADE**

- **LOCALIDADE TEMPORAL**
- **LOCALIDADE ESPACIAL**

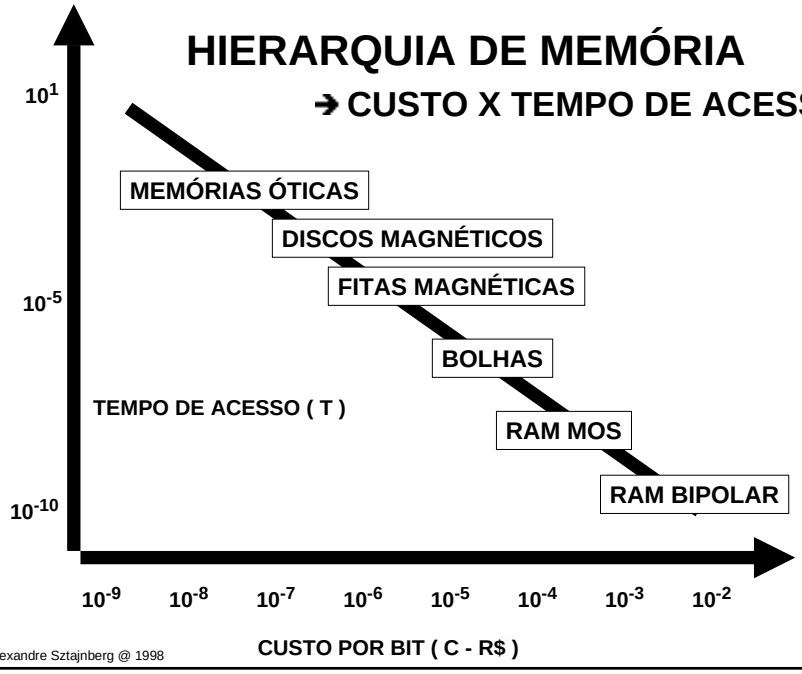
HIERARQUIA DE MEMÓRIA

BASEADA NO TEMPO DE ACESSO



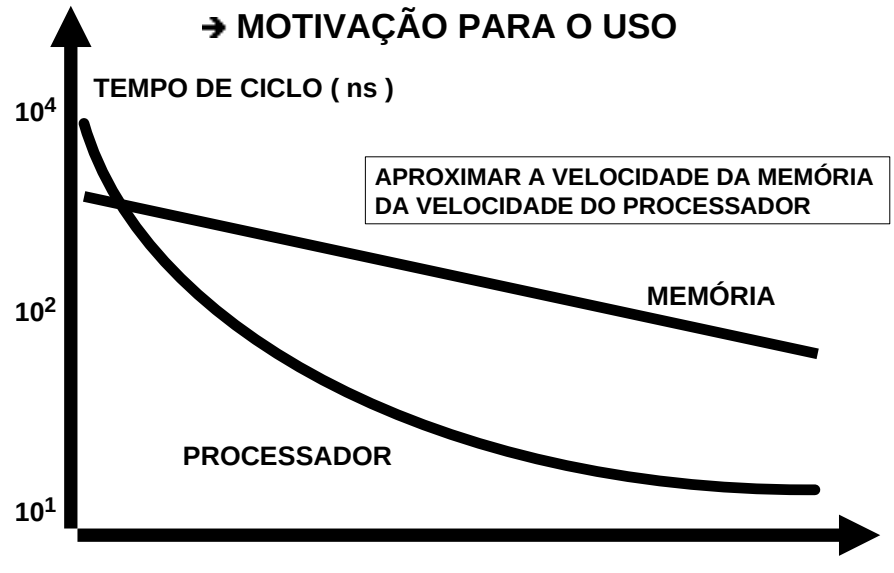
HIERARQUIA DE MEMÓRIA

➔ CUSTO X TEMPO DE ACESSO



Hierarquia de Memória

➔ MOTIVAÇÃO PARA O USO

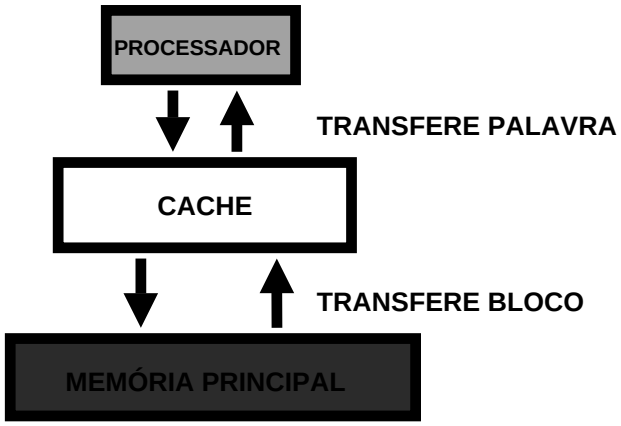


Memória Cache

- cacher do francês = esconder
- princípio da localidade
  - palavra é usada várias vezes - pode ficar “mais perto”
  - uma referência à memória lenta e várias à cache
  - taxa de acerto (HIT) e taxa de erro (MISS)
- memória de $\mathbf{2^m}$ bytes é dividida em blocos de **b** bytes
- tipos de organização
  - cache associativo (hardware especial)
  - cache com mapeamento direto (hash code + tag)
- escritas
  - write through (escreve imediatamente na memória)
  - copy back (usa bit de *alterado*)

MEMÓRIA CACHE

➔ **CACHE / PROCESSADOR /MEMÓRIA PRINCIPAL**



MEMÓRIA CACHE

➔ **PROJETO DO SISTEMA CACHE/MEMÓRIA PRINCIPAL**

MEMÓRIA PRINCIPAL

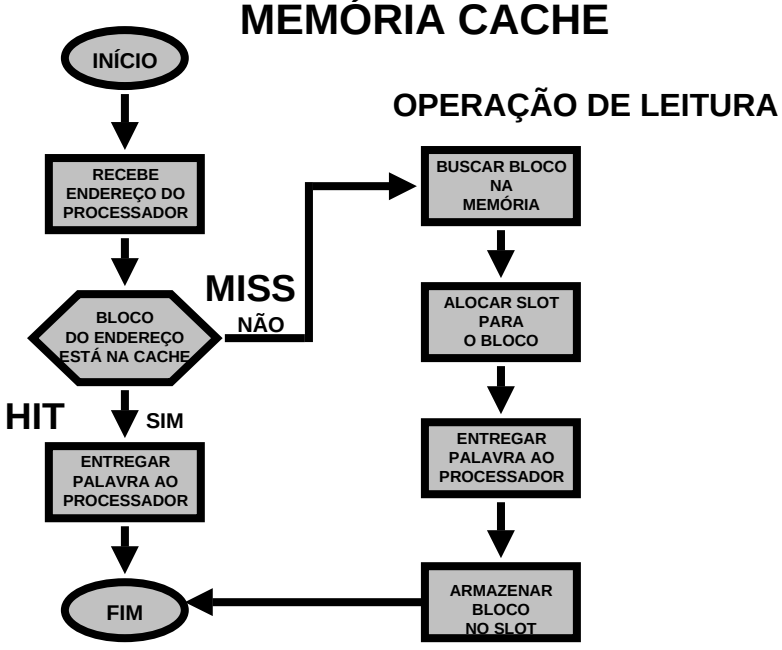
2^n ENDEREÇOS
DIVIDIDA EM BLOCOS DE K ENDEREÇOS
M = 2^n / K BLOCOS

CACHE

C SLOTS DE K ENDEREÇOS

C <<<< M

MEMÓRIA CACHE



Hardware Protection

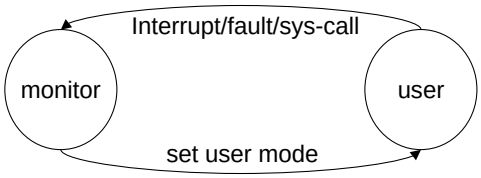
- Dual-Mode Operation
- I/O Protection
- Memory Protection
- CPU Protection

Dual-Mode Operation

- Sharing system resources requires operating system to ensure that an incorrect program cannot cause other programs to execute incorrectly.
- Provide hardware support to differentiate between at least two modes of operations.
  1. *User mode* – execution done on behalf of a user.
  2. *Monitor mode* (also *supervisor mode* or *system mode*) – execution done on behalf of operating system.

Dual-Mode Operation (Cont.)

- ***Mode bit*** **added to computer hardware to indicate the current mode: monitor (0) or user (1).**
- **When an interrupt or fault occurs hardware switches to monitor mode.**



- ***Privileged instructions*** **can be issued only in monitor mode.**

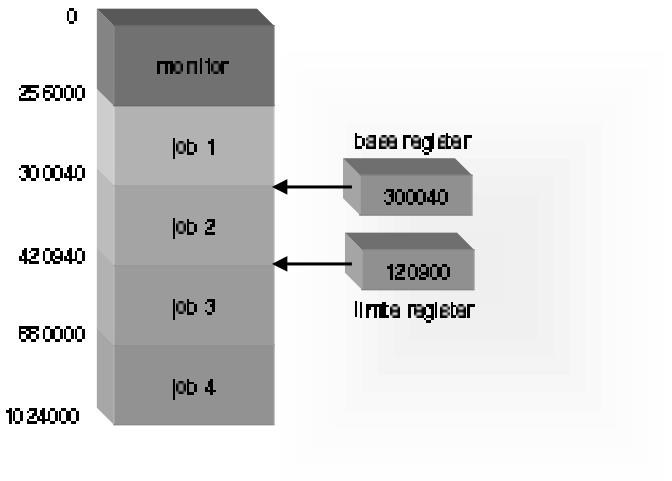
I/O Protection

- **All I/O instructions are privileged instructions.**
- **Must ensure that a user program could never gain control of the computer in monitor mode (I.e., a user program that, as part of its execution, stores a new address in the interrupt vector).**

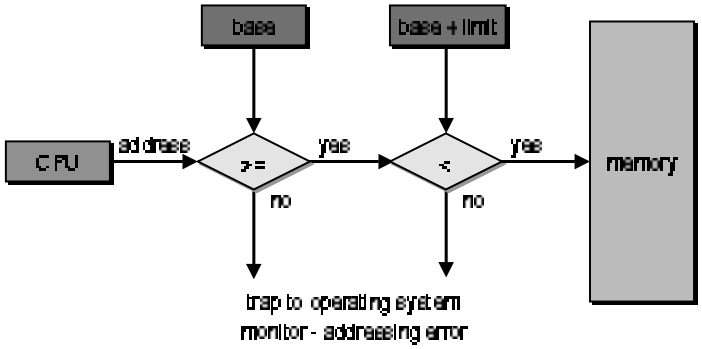
Memory Protection

- **Must provide memory protection at least for the interrupt vector and the interrupt service routines.**
- **In order to have memory protection, add two registers that determine the range of legal addresses a program may access:**
  - base register **– holds the smallest legal physical memory address.**
  - Limit register **– contains the size of the range**
- **Memory outside the defined range is protected.**

A Base And A limit Register Define A Logical Address Space



Protection Hardware



- When executing in monitor mode, the operating system has unrestricted access to both monitor and user's memory.
- The load instructions for the *base* and *limit* registers are privileged instructions.

CPU Protection

- *Timer* – interrupts computer after specified period to ensure operating system maintains control.
  - Timer is decremented every clock tick.
  - When timer reaches the value 0, an interrupt occurs.
- Timer commonly used to implement time sharing.
- Time also used to compute the current time.
- Load-timer is a privileged instruction.

General-System Architecture

- Given the I/O instructions are privileged, how does the user program perform I/O?
- System call – the method used by a process to request action by the operating system.
  - Usually takes the form of a trap to a specific location in the interrupt vector.
  - Control passes through the interrupt vector to a service routine in the OS, and the mode bit is set to monitor mode.
  - The monitor verifies that the parameters are correct and legal, executes the request, and returns control to the instruction following the system call.

Use of A System Call to Perform I/O