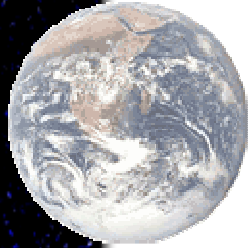


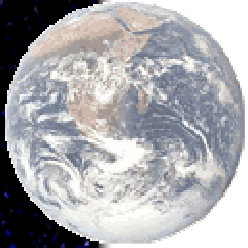
Componentes básicos de um sistema computacional

Cap. 1 (Stallings)



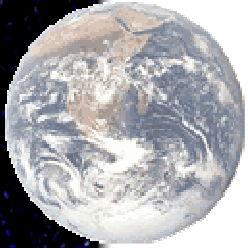
Sistema de Operação

- Explora recursos de hardware de um ou mais processadores
- Provê um conjunto de serviços aos utilizadores
- Gerencia memória secundária e dispositivos de entrada e saída



Elementos Básicos

- Processador
- Memória principal
 - volátil
 - Conhecida também como memória real ou primária
- Módulos de entrada e saída
 - Dispositivos de memória secundária
 - Dispositivos de comunicação
 - terminais
- Barramento do sistema
 - Comunicação entre processadores, memória e módulos de entrada e saída



Processador

- 2 registradores internos
 - Memory address register (MAR)
 - Especifica o próximo endereço para leitura ou escrita
 - Memory buffer register (MBR)
 - Contém dado escrito em memória ou recebe dado lido da memória
 - I/O address register
 - I/O buffer register

Componentes de mais alto nível

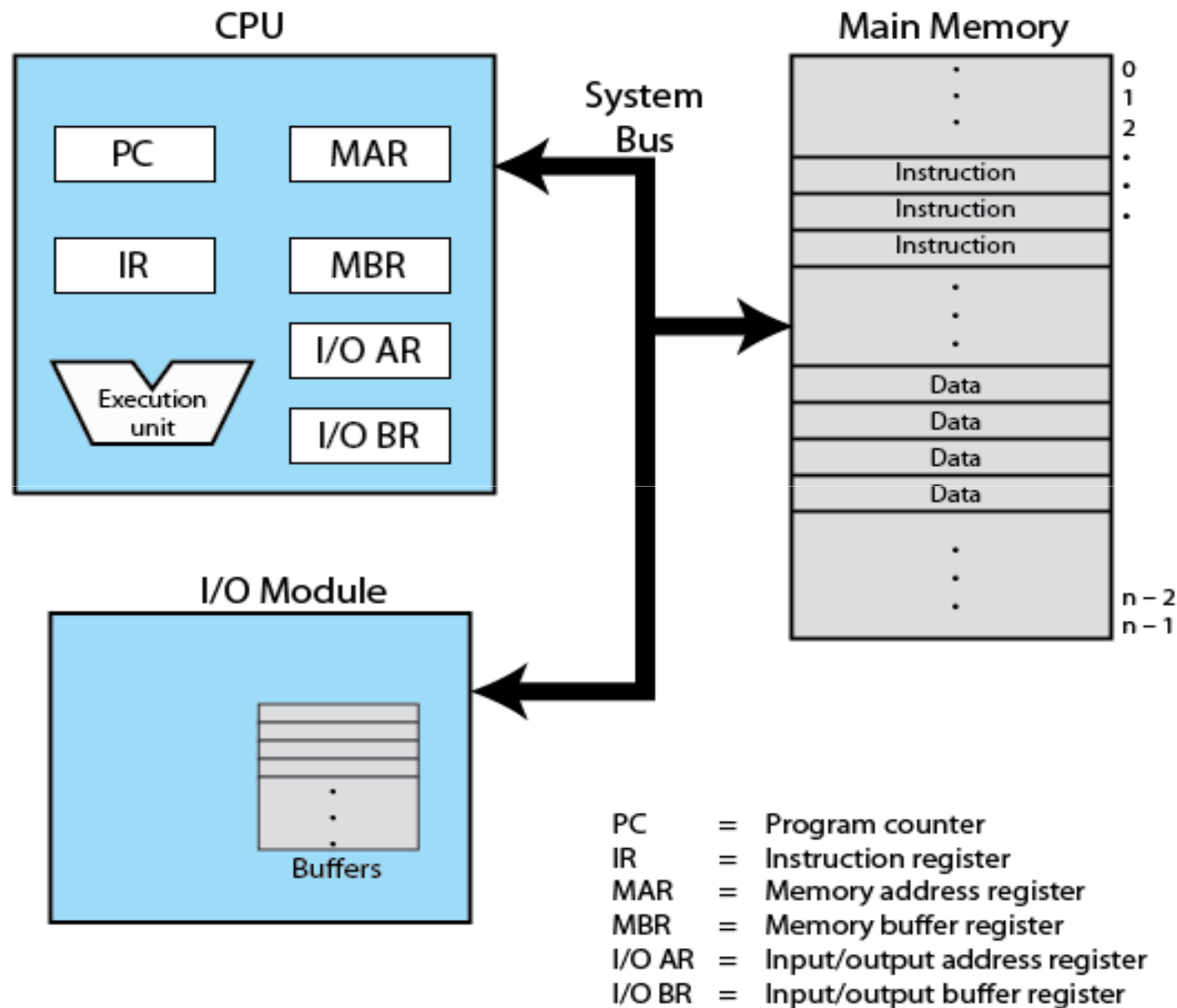
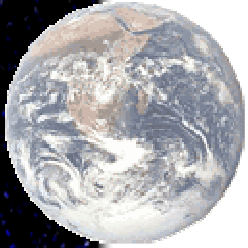
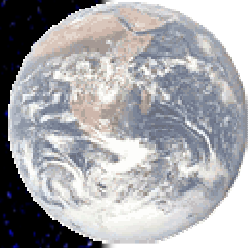


Figure 1.1 Computer Components: Top-Level View



Registradores do Processador

- Visíveis ao utilizador
 - Ajudam o programador a minimizar referências à memória principal otimizando a sua utilização
- Registradores de controle e estado
 - Usados pelo proc para controlar a sua operação
 - Usados por rotinas privilegiadas do sistema de operação para controlar a execução dos programas



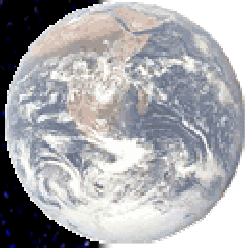
Registradores visíveis ao utilizador

- Podem ser referenciados pela linguagem de máquina
- Disponível a todos os programas (aplicações do utilizador e do sistema)
- Tipos de registradores
 - Dados
 - Endereços
 - Índice
 - Apontador para segmentos
 - Apontador para a pilha de execução



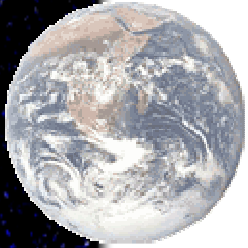
Registradores visíveis ao utilizador

- Registradores de endereço
 - Índice
 - Adiciona um índice a um endereço base para aceder um determinado endereço
 - Apontador para segmento
 - Quando a memória é dividida em segmentos, este registrador indica o segmento e um offset
 - Apontador para a pilha de execução
 - Aponta para o topo da pilha



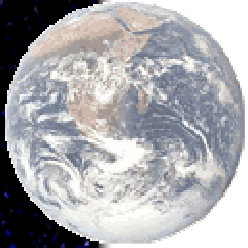
Regs de estado e controle

- Contador de programa (PC)
 - Contém o endereço da próxima instrução do programa a ser buscada da memória
- Registrador de Instruções (IR)
 - Contém a instrução mais recente que foi buscada da memória
- Estado do programa (PSW)
 - Códigos de condição
 - Habilita ou desabilita interrupções
 - Modo utilizador ou supervisor (privilegiado)



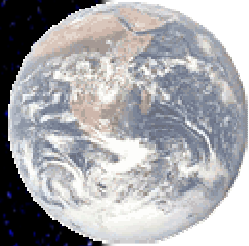
Regs de estado e controle

- Flags e códigos de condição
 - Bits que precisam ser “setados” pelo hw como resultados de operações
 - Exemplos
 - Resultado positivo
 - Resultado negativo
 - Zero
 - “Overflow”



Execução de instruções

- 2 passos
 - Processador lê próxima instrução da memória e coloca no **Registrador de Instruções (IR)**
 - “Fetches”
 - **PC** passa a apontar para a próxima instrução
 - Processador executa instrução



Ciclo de instrução

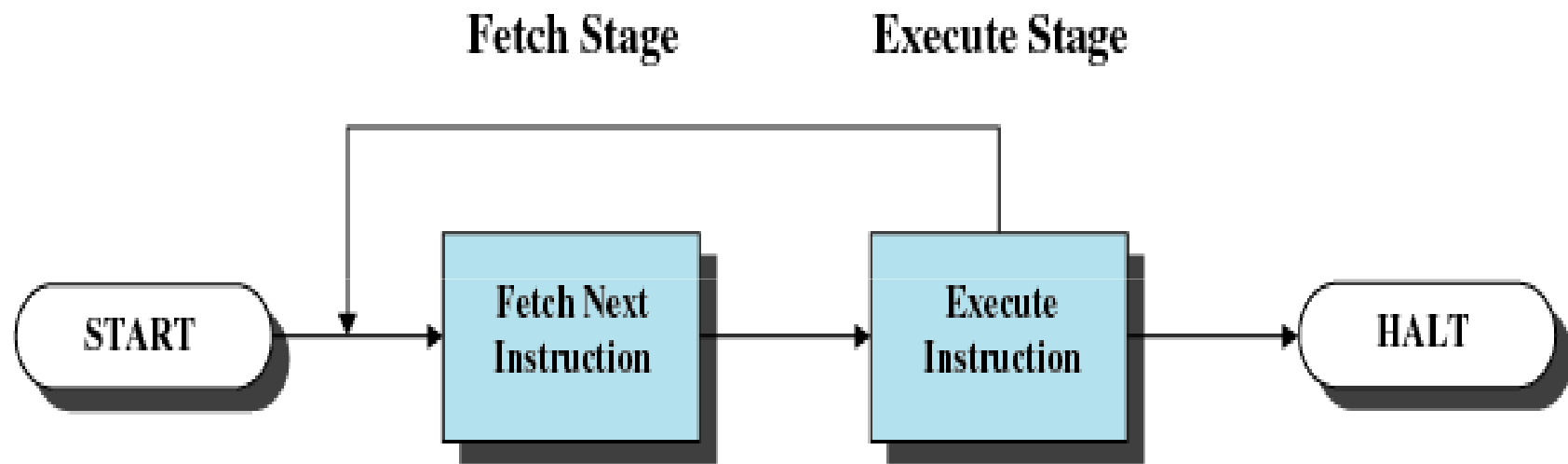
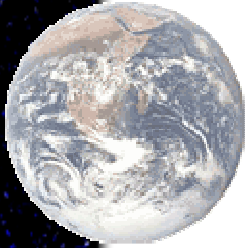
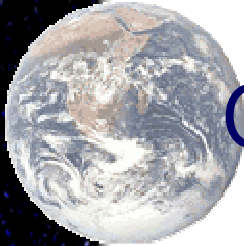


Figure 1.2 Basic Instruction Cycle

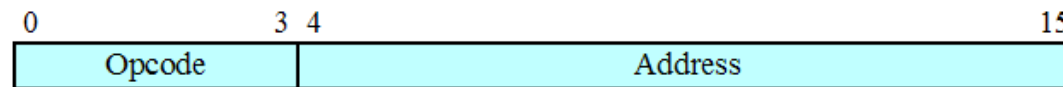


IR – Registrador de Instrução

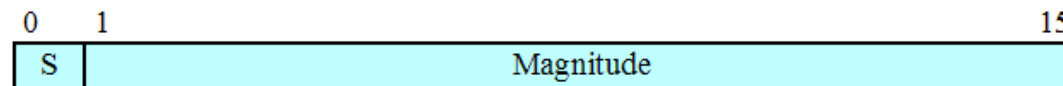
- Instrução “fetched” (buscada na memória) é colocada no IR
- Categorias de instruções
 - Processador-memória
 - Transfere dados entre processador e memória
 - Processador-I/O
 - Dados transferidos de ou para um dispositivo periférico
 - Processamento de dados
 - Operação aritmética ou lógica sobre o dado
 - Controle
 - Altera sequência de execução



Características de uma máquina hipotética



(a) Instruction format



(b) Integer format

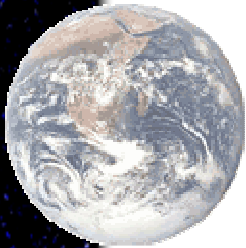
Program Counter (PC) = Address of instruction
Instruction Register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory
0010 = Store AC to Memory
0101 = Add to AC from Memory

(d) Partial list of opcodes

Figure 1.3 Characteristics of a Hypothetical Machine



Exemplo de Execução de Programa

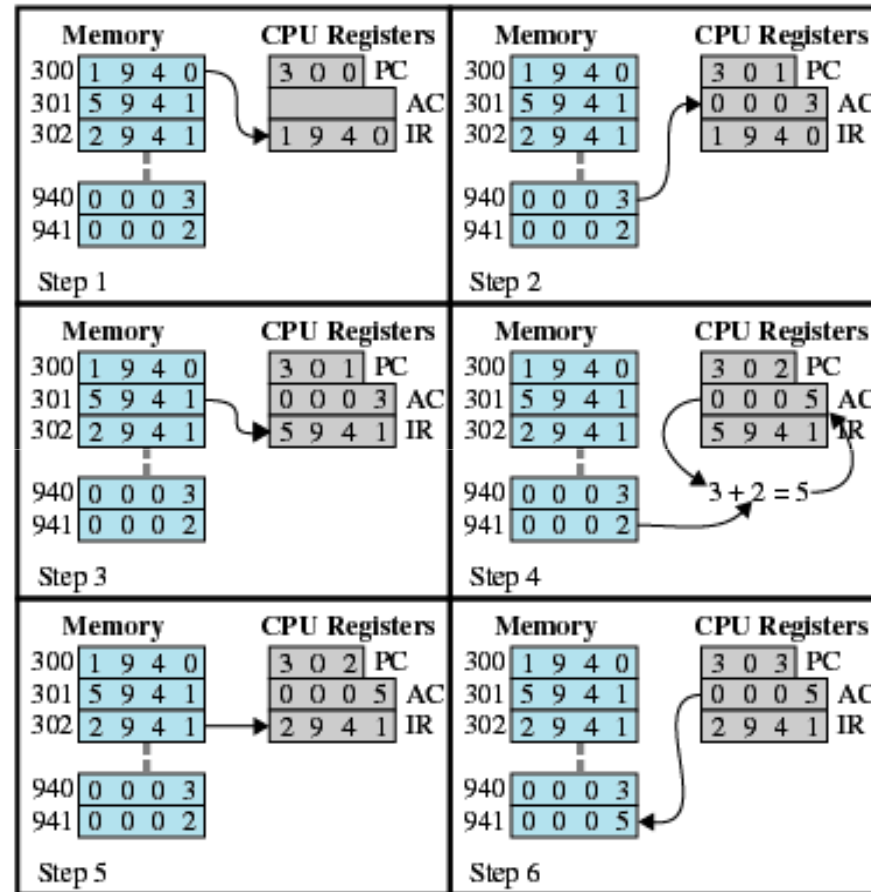
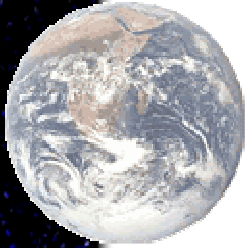


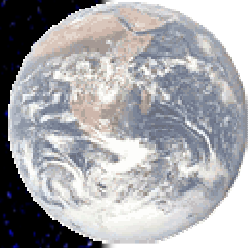
Figure 1.4 Example of Program Execution
(contents of memory and registers in hexadecimal)



Direct Memory Access (DMA)

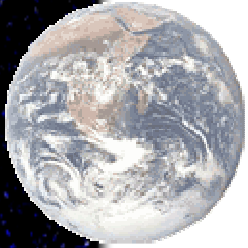
Acesso Direto à Memória

- Dados de I/O são trocados diretamente entre memória e dispositivo.
- Processador dá autoridade ao módulo de I/O para ler e escrever de ou na memória
- Retira do processador a responsabilidade por esta comunicação



Interrupções

- Interrompe a sequência normal de execução do processador
- Dispositivos de I/O normalmente são mais lentos do que o processador
 - Processador deve parar para esperar pelo dispositivo

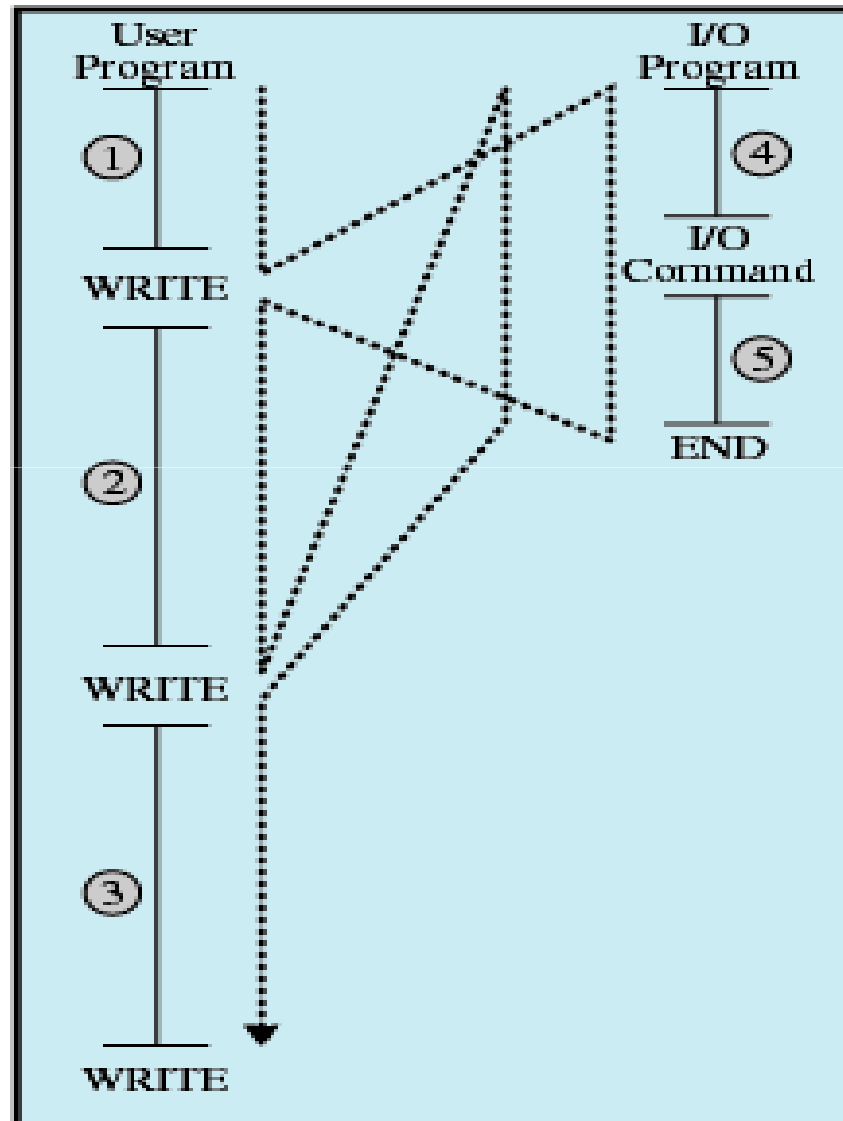


Classes de Interrupções

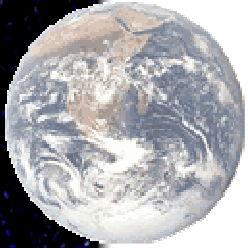
- **Programa:** gerada por alguma condição que ocorre como resultado da execução de uma instrução
 - Overflow aritmético
 - Divisão por zero
 - Tentativa de executar instrução ilegal
 - Referência a endereço fora do espaço de endereçamento do utilizador
- **Timer (relógio):** gerada por relógio interno ao processador. Permite ao SO executar certas funções com regularidade.
- **I/O:** gerada por algum controlador de I/O para sinalizar término normal de uma operação ou sinalizar condições de erro.
- **Falha de hw:** gerada por uma falha, pe, falta de energia ou erro de paridade de memória.



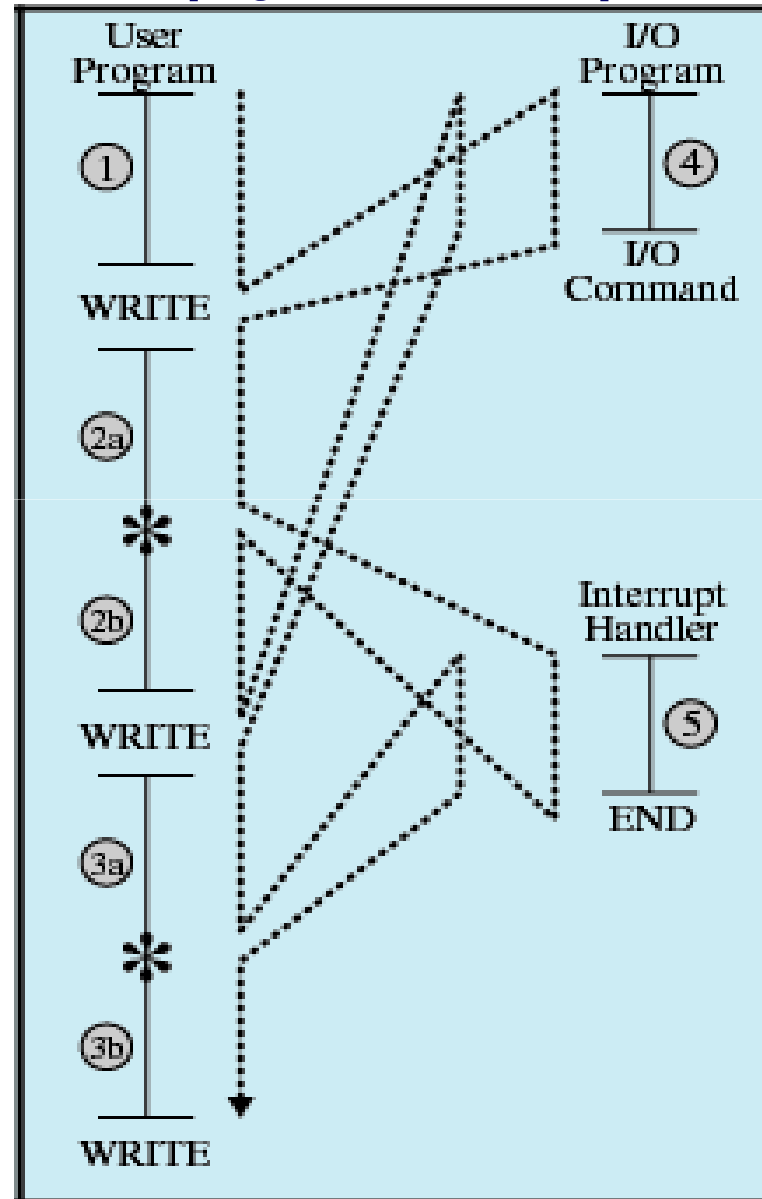
Fluxo de controle de um programa **SEM** interrupções

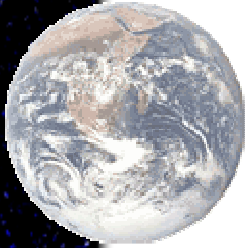


(a) No interrupts

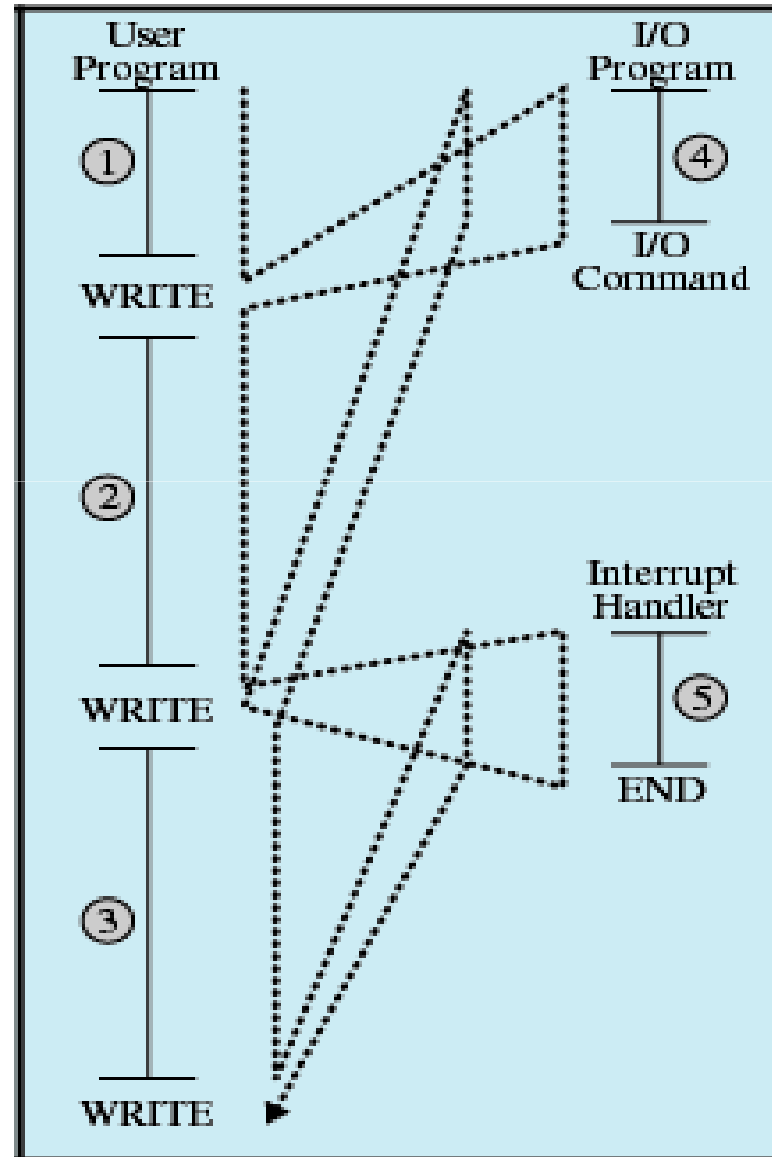


Fluxo de controle de um programa COM interrupções, espera curta

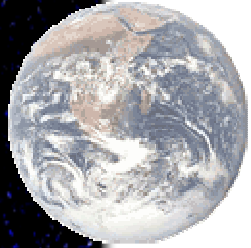




Fluxo de controle de um programa COM interrupções, espera longa

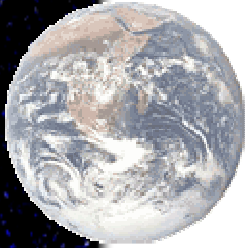


(c) Interrupts: long I/O wait



Interrupt Handler

- Programa que serve um dispositivo de I/O
- Geralmente parte do SO



Interrupções

- Suspende a sequência normal de execução

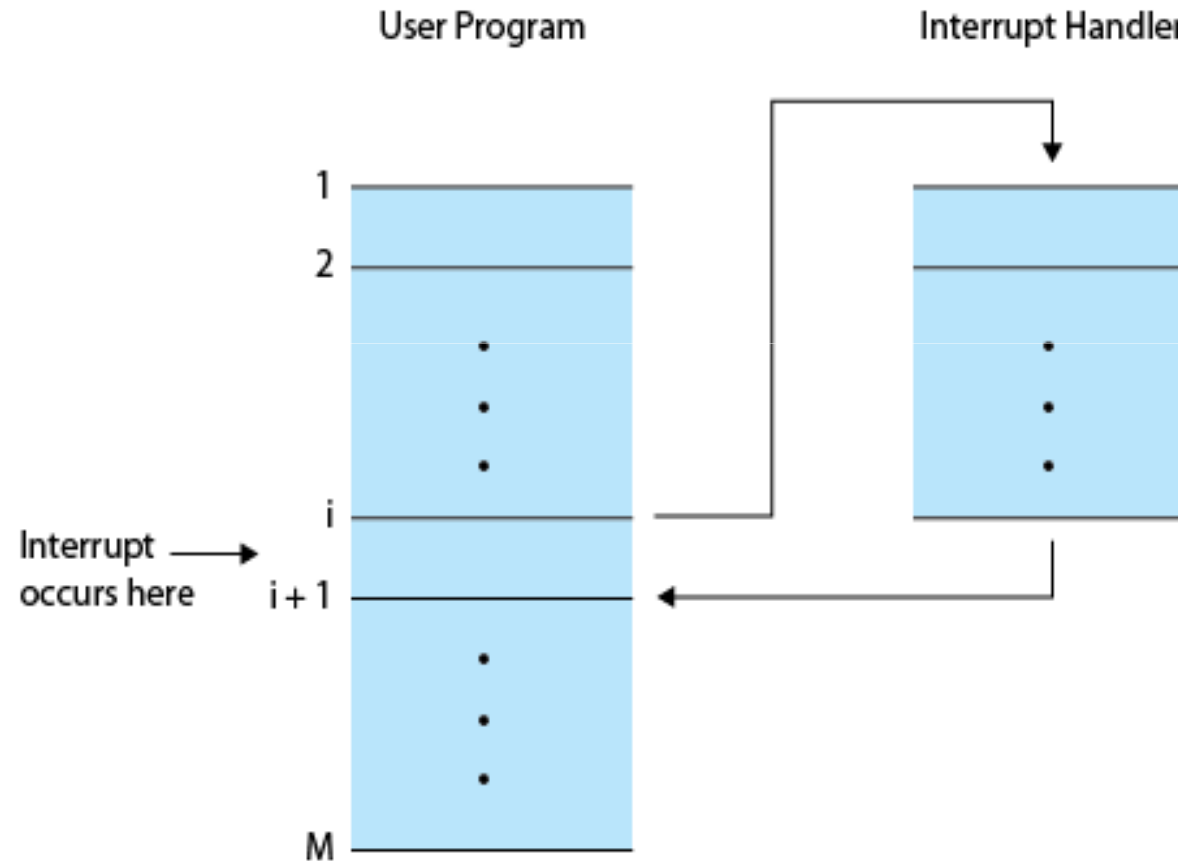
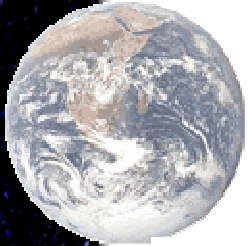


Figure 1.6 Transfer of Control via Interrupts



Ciclo de instruções com interrupção

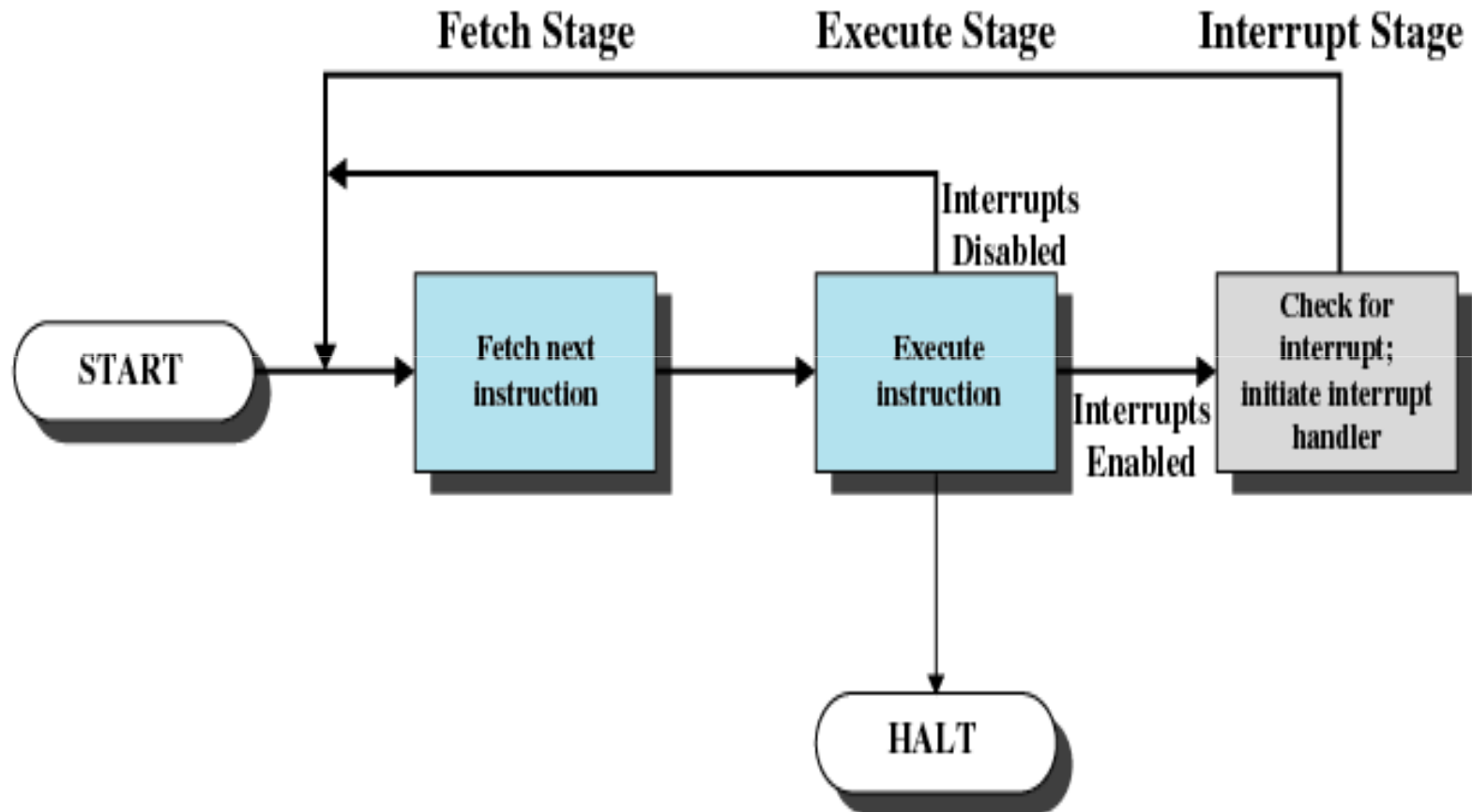
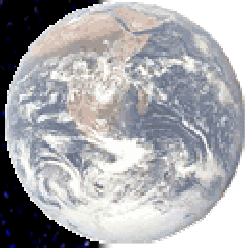


Figure 1.7 Instruction Cycle with Interrupts



Processamento de interrupções

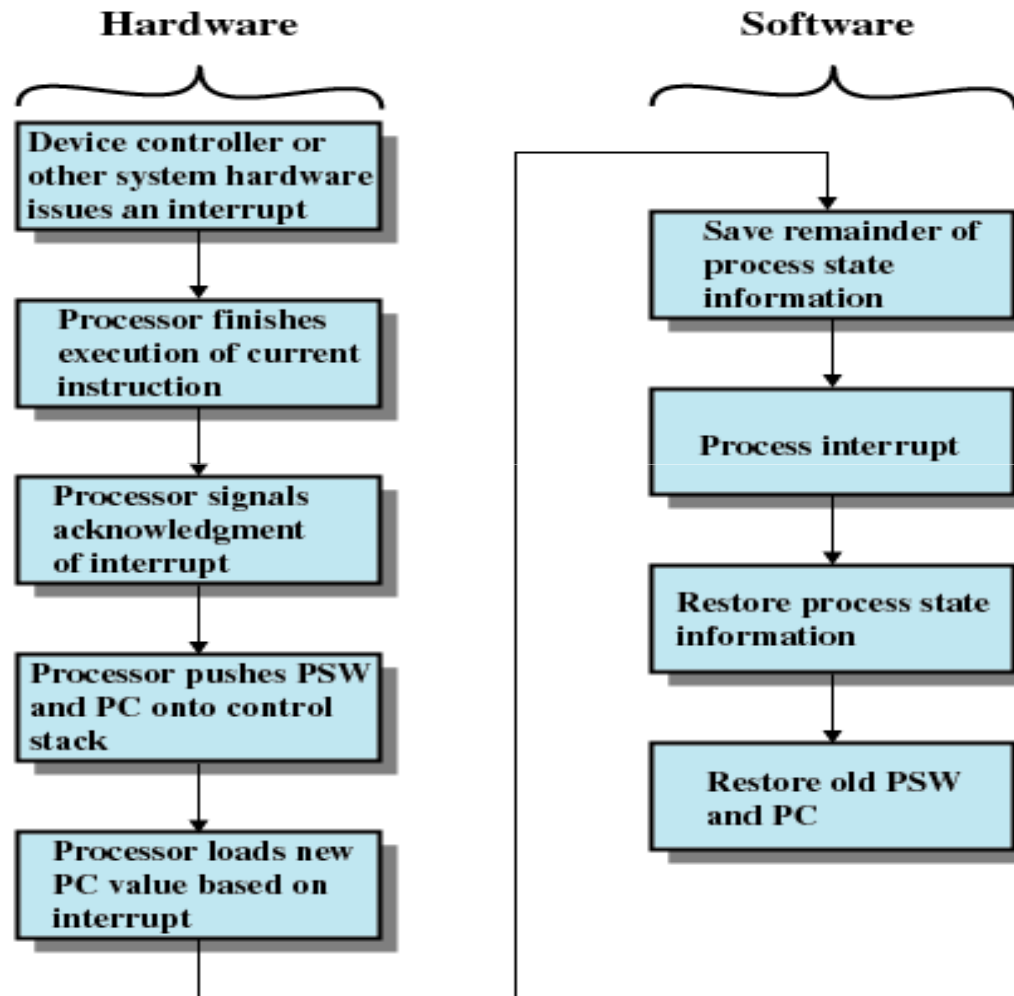


Figure 1.10 Simple Interrupt Processing

Hierarquia de Memória

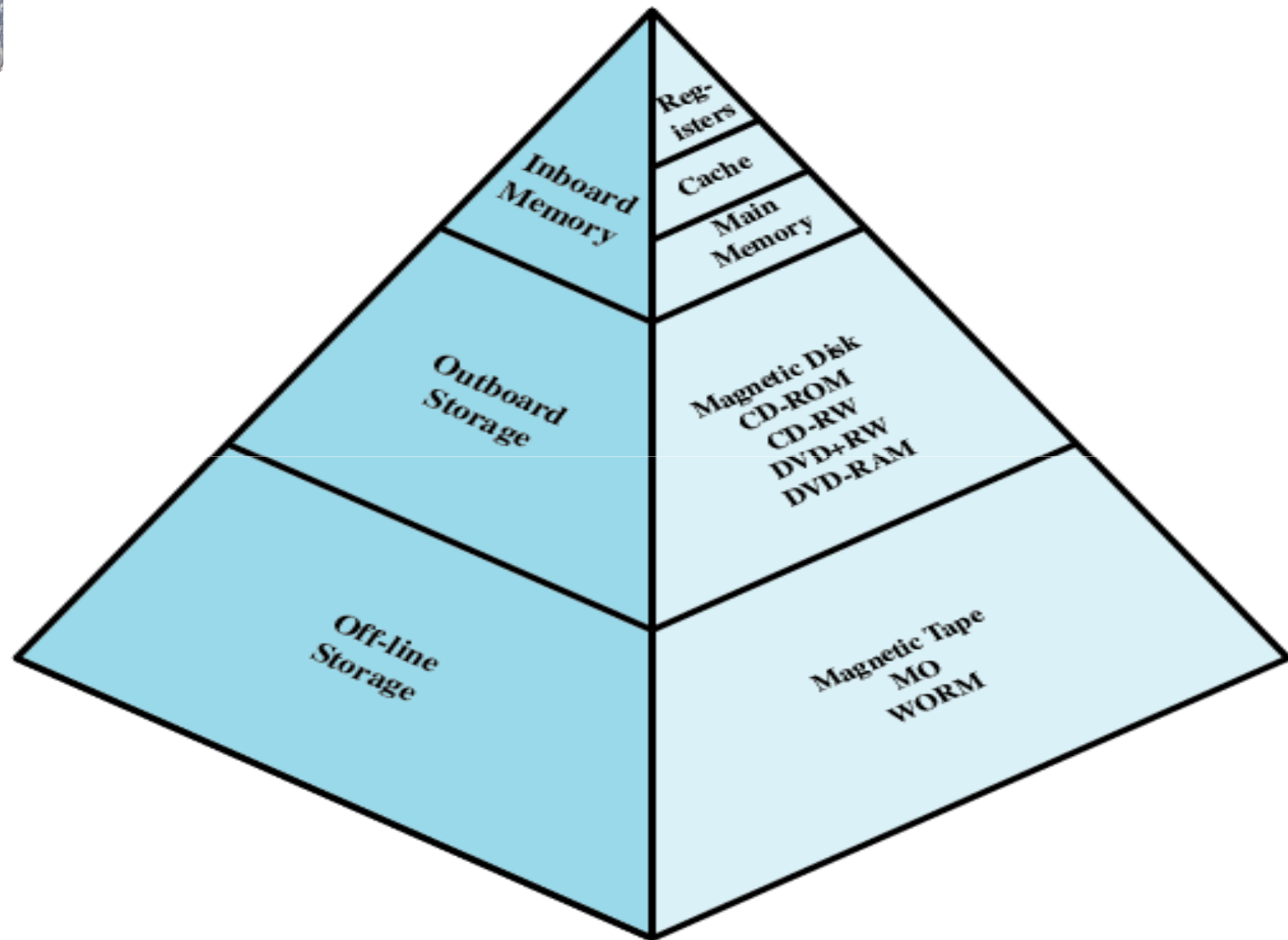
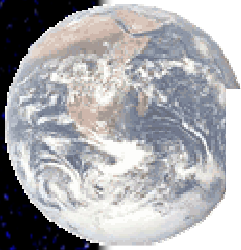
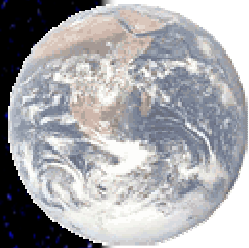
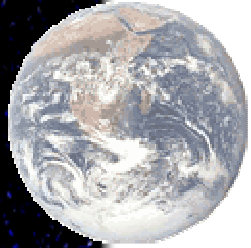


Figure 1.14 The Memory Hierarchy



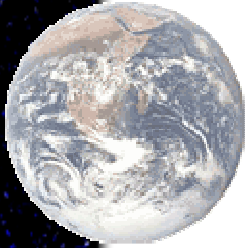
Hierarquia de Memória

Item	Tempo	Escala do Tempo Humano (100 milhões vezes + devagar)
ciclo CPU	10 ns (100MHz)	1 segundo
acesso a cache	30 ns	3 segundos
Acesso a memória	60 ns	6 segundos
Troca contexto	10,000 ns (100 μ s)	166 minutos
Acesso a disco	10,000,000 ns (10 ms)	11 dias
Quantum	100,000,000 ns (100 ms)	116 dias



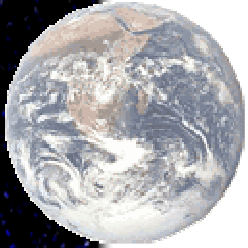
Memória Secundária

- Não volátil
- auxiliar
- Usada para guardar programas e ficheiros de dados



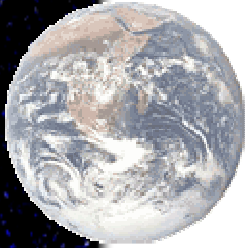
Cache de disco

- Porção de memória utilizada como um “buffer” para armazenar temporariamente dados de disco
- Escritas em disco são agrupadas
- Algum dado que foi escrito pode ser referenciado novamente. Este dado pode ser recuperado rapidamente se estiver na cache de disco



Memória Cache

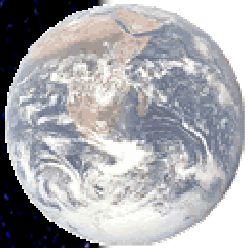
- Invisível ao SO
- Melhora taxa de acesso a dados
- Pois velocidade do processador é mais alta do que a de acesso à memória principal
- Explora o princípio de **localidade de dados**



Operações de I/O

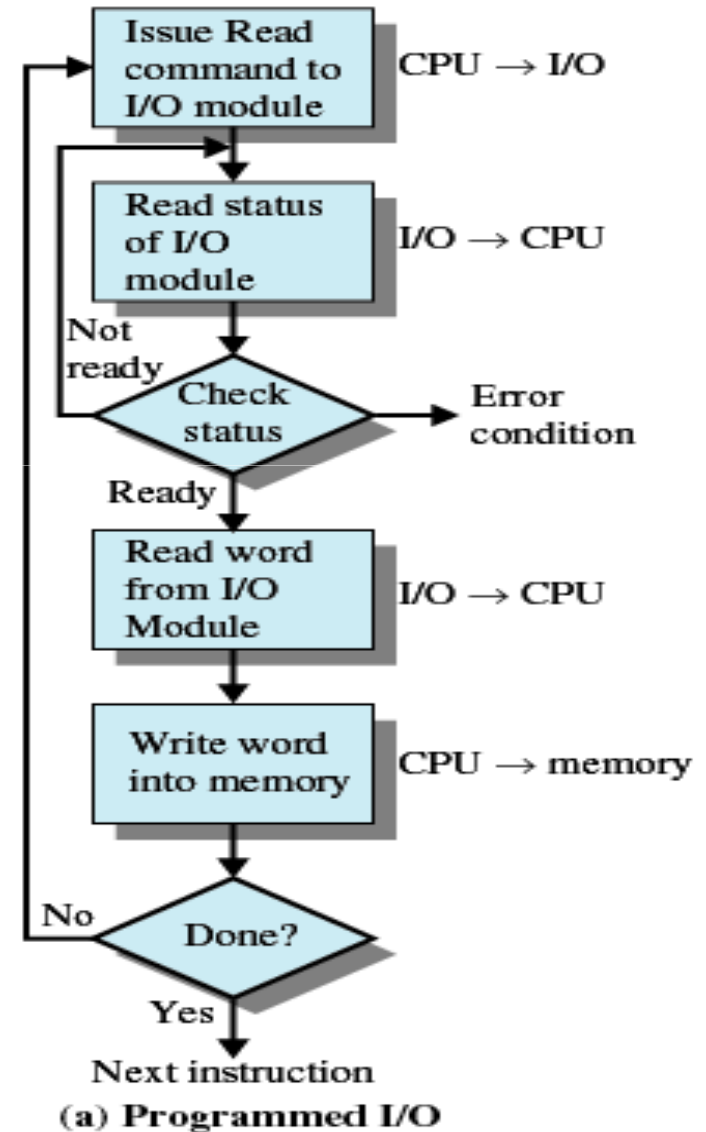
3 tipos:

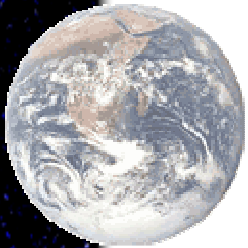
- Programmed I/O (I/O programado)
- Dirigidas por Interrupções
- DMA (Direct memory access)



Programmed I/O

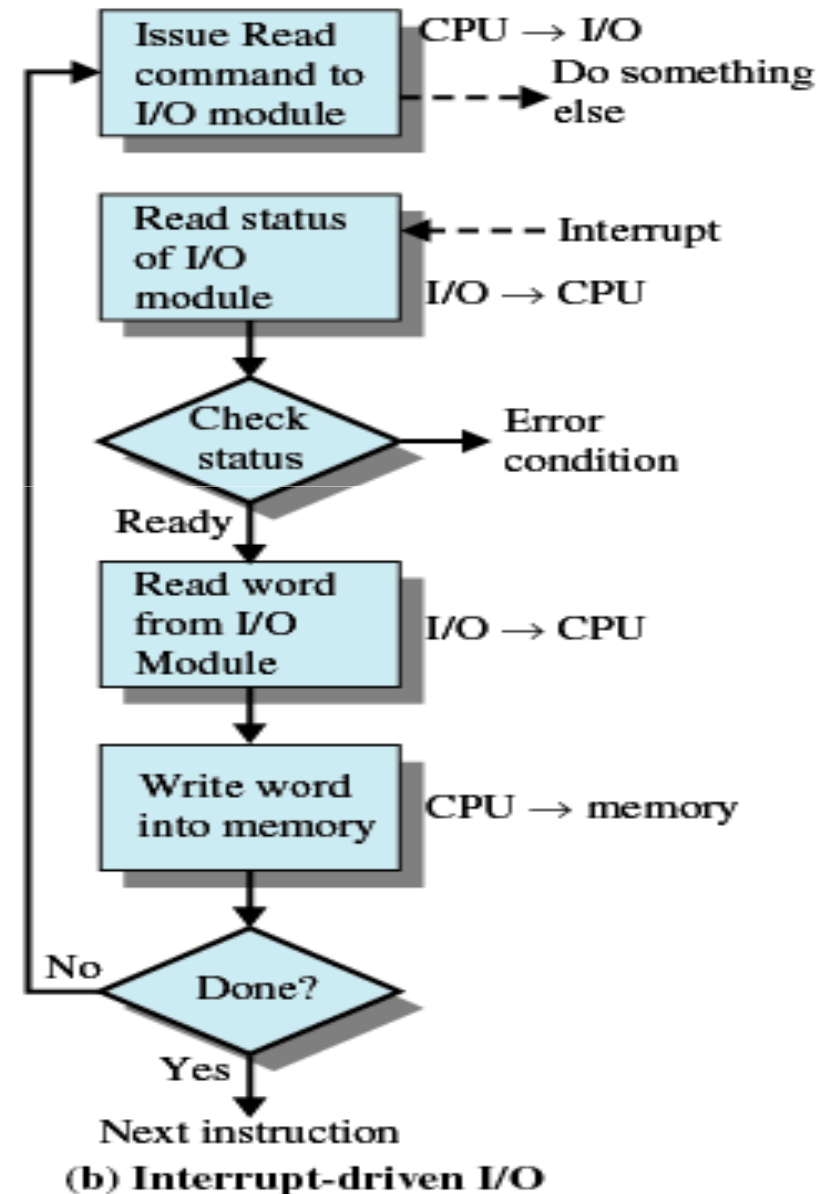
- Módulo de I/O executa a ação. Processador não é envolvido
- “Seta” bits apropriados no registrador de estado de I/O
- Nenhuma interrupção é causada pelo módulo de I/O
- Processador responsável por verificar se a operação já está completa (faz “polling” do módulo de I/O)

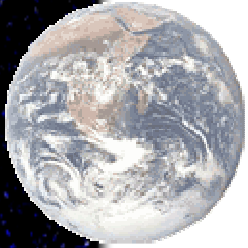




I/O dirigida por interrupção

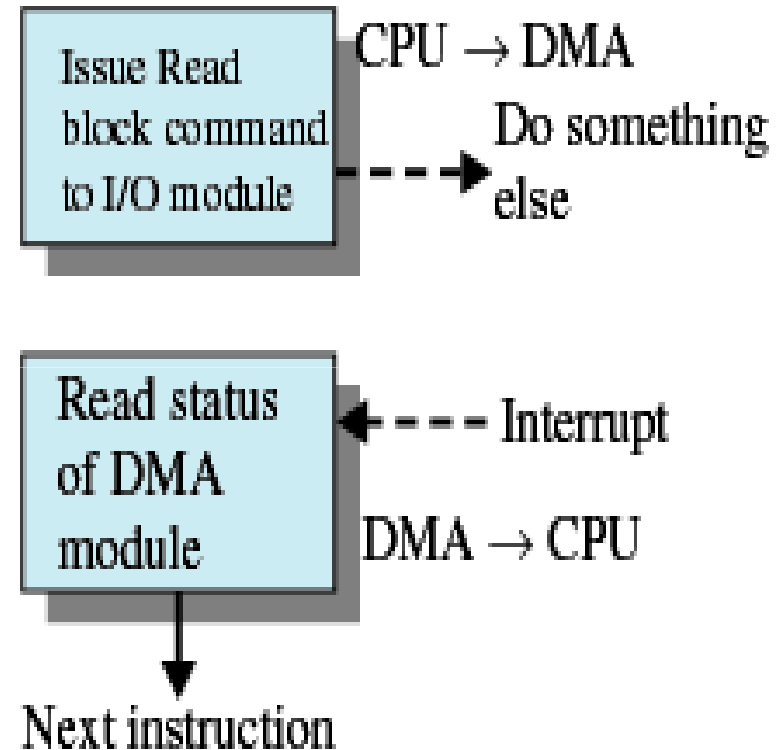
- Processador é interrompido quando o módulo de I/O está pronto com os dados
- Processador salva o contexto do programa e inicia o interrupt handler
- Não há espera desnecessária
- Pode consumir muito do tempo do processador porque toda escrita ou leitura tem que passar pelo processador



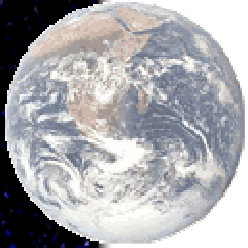


Direct Memory Access

- Transfere blocos de dados diretamente de e para a memória
- Uma interrupção é sinalizada quando a transferência está completa
- Processador continua com outra tarefa



(c) Direct memory access



Proteção a nível de hardware

- Dois modos de operação (dual mode: kernel/supervisor e utilizador)
- Proteção de I/O
 - Acesso a dispositivos feito apenas em modo kernel
- Proteção de acesso à memória
 - Registradores base e limite
- Proteção da CPU
 - timer