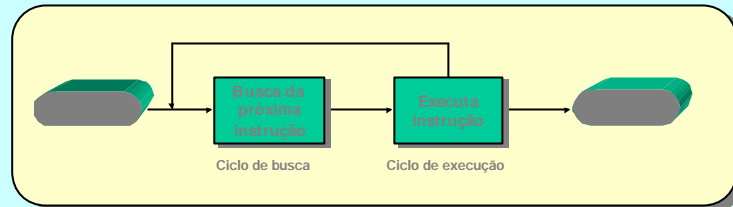


Ciclo Básico de Instruções



Programa → Sequência de instruções em memória

Tradutor : Compilador X Interpretador / Linkeditor

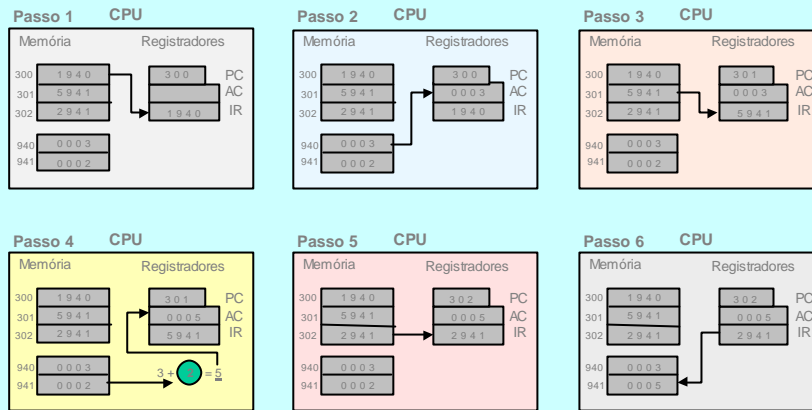
- Converte programa-fonte em programa objeto (linguagem de máquina)
- Programa compilado = mais rápido
- Programa Interpretado = interativo, conversacional

Tipos de Instruções

- Acesso à memória
 - Transferência de dados entre o processador e a memória
- Entrada / saída
 - Transferência de dados entre o processador e o dispositivo
- Tratamento de dados
 - Operações aritméticas ou lógicas
- Controle (desvios)
 - Alteração da sequência de execução de instruções

Formatos de Instrução : 0 / 1 / 2 / 3 endereços

Exemplo de Execução de Instruções

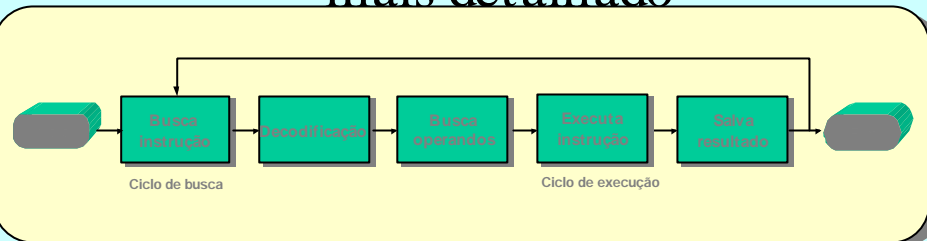


Lista parcial dos códigos de operação
 0001 = Carrega AC da memória
 0010 = Armazena AC na memória
 0101 = Adiciona em AC o valor da memória

Seqüência de Execução das Instruções

- ⌘ Passo 1) PC contém o endereço da primeira instrução. Esta instrução é carregada no IR
- ⌘ Passo 2) Os primeiros 04 bits do IR (OpCode) indicam que o AC deve ser carregado a partir de uma posição de memória (940).
- ⌘ Passo 3) PC é incrementado e é carregada a próxima instrução
- ⌘ Passo 4) O antigo conteúdo de AC e o conteúdo da posição 941 são somados e o resultado é armazenado em AC
- ⌘ Passo 5) PC é incrementado e é carregada a próxima instrução
- ⌘ Passo 6) O conteúdo de AC é armazenado na posição 941

Ciclo de Instrução - mais detalhado -



Procedimentos:

- Procedimento de busca inclui atualização do PC
- Procedimento de decodificação analisa o código de operação e verifica as ações a serem tomadas
- Ciclo de busca de operandos pode existir ou não. Depende do formato da instrução

Interrupções

- Permite a execução de uma outra instrução enquanto uma operação de E/S está sendo executada
- Suspende o processo em execução pela ocorrência de um evento externo (interrupção)
- Melhora a eficiência do processador

Observação:

Após o término da interrupção, o processo suspenso pode reformar à execução ou um outro ser escalonado

Tipos de Interrupções

Síncrona:

Estados de exceção

- ✓ estouro aritmético
- ✓ divisão por zero
- ✓ instrução ilegal
- ✓ acesso não permitido
- ✓ chamada de sistema (system call)



Temporizador

- ✓ Tentativa de um programa de executar uma outra tarefa

Assíncrona:

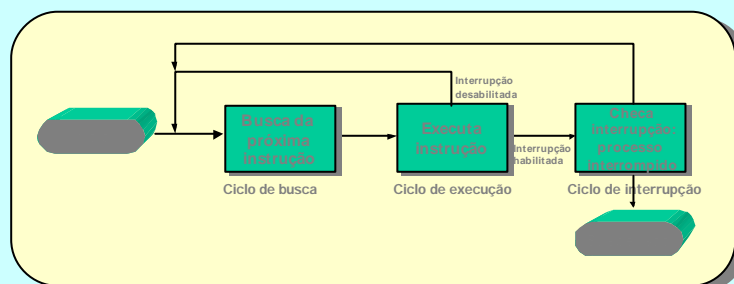
Falha de hardware

- ✓ Erro de paridade de memória

E/S

- ✓ Sinais normais de conclusão da operação ou erro

Ciclo de Instrução com Interrupções



Procedimentos:

- CPU verifica se há interrupções após cada instrução
- Se não houver interrupção, então busca a próxima instrução
- Se uma interrupção está pendente, então suspende a execução do programa corrente, e executa a **Rotina de serviço da interrupção**.

Rotina de Serviço da Interrupção

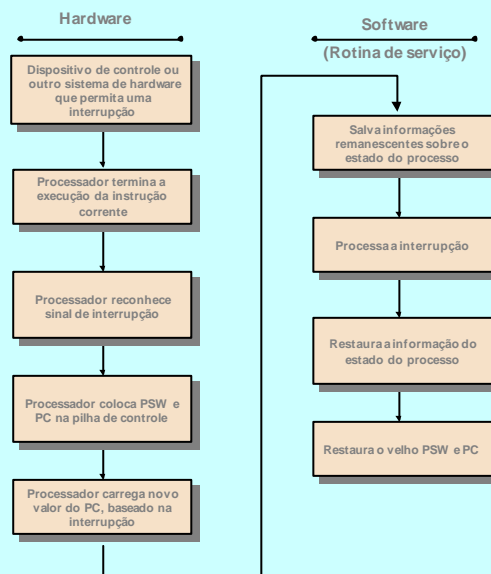
Definição:

- É um programa que determina a natureza da interrupção e executa as ações que sejam necessárias

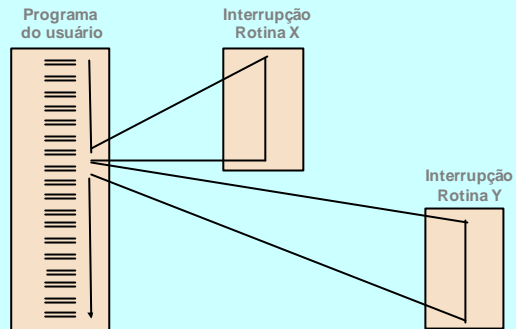
Características:

- O controle é transferido para este programa
- Os endereços iniciais das rotinas de serviço normalmente são mantidos na tabela de interrupção
- É necessário salvar o status do programa corrente (conteúdo do PC, PSW, registradores, ...)
- A rotina é ativada diretamente pelo hardware
- O ponto de interrupção pode ocorrer em qualquer ponto do programa
- As rotinas de serviço normalmente fazem parte do Sistema Operacional
- Overhead adicional para ativar e executar a rotina de serviço.

Processamento de Interrupção Simples

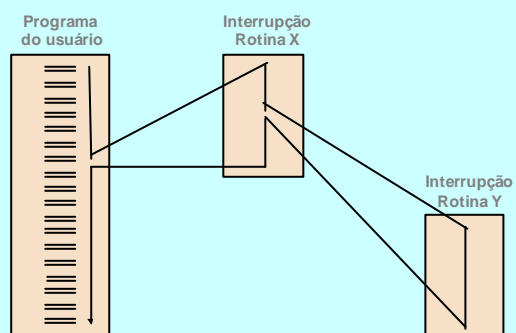


Múltiplas interrupções: Processamento Sequencial



- A Rotina de Serviço desabilita as interrupções
- Uma nova interrupção só é tratada após o retorno
- A interrupção pode demorar a ser tratada, o que pode eventualmente ocasionar uma perda de dados
- Finalizada a Rotina de Serviço de interrupção, o processador checa por interrupções adicionais

Múltiplas Interrupções: Processamento Aninhado



- Interrupções têm prioridade
- Interrupções com alta prioridade interrompem rotinas de serviço de interrupções de menor prioridade
- Exemplos de prioridade:
 - impressora -
 - disco
 - comunicação +

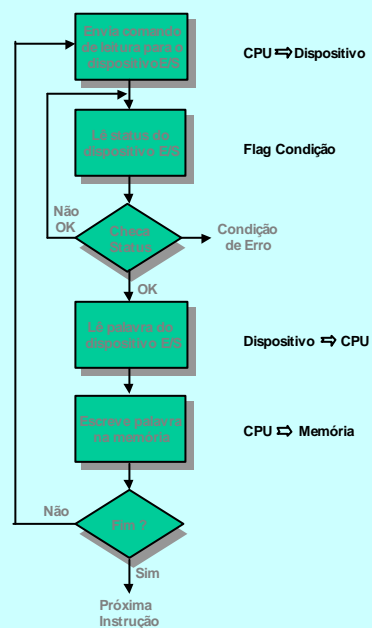
Técnicas de E/S

- **E/S Programada**
 - ⌘ Não usa interrupções.
 - ⌘ CPU tem que esperar pela finalização de cada operação de E/S
- **E/S com Interrupção**
 - ⌘ CPU pode executar outros programas durante a operação de E/S.
 - ⌘ É interrompida quando a operação de E/S é concluída.
- **Acesso direto à memória (DMA)**
 - ⌘ Um bloco de dados é transferido diretamente entre o dispositivo de E/S e a memória sem interrupção da CPU

E/S Programada

Descrição:

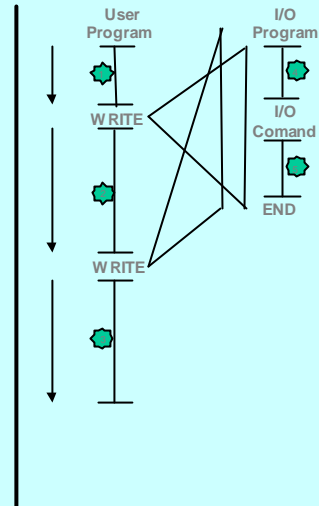
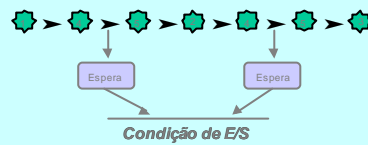
- ⌘ SO requisita operação de E/S ao dispositivo
- ⌘ SO aguarda em loop pelo término da operação de E/S
- ⌘ SO retorna o controle da CPU ao programa do usuário



E/S Programada (Exemplo)

Descrição:

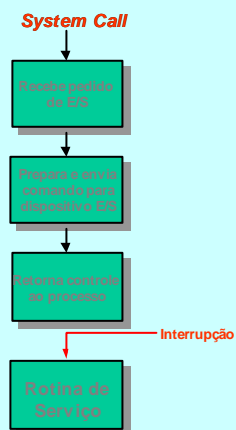
- ❧ O programa do usuário executa uma série de chamadas de WRITE (System Call).
- ❧ Os segmentos 1, 2 e 3 se referem à uma seqüência de instruções que não envolve E/S.
- ❧ Seqüência de instruções:



E/S com Interrupção

Descrição:

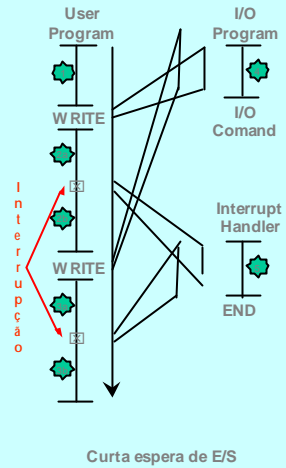
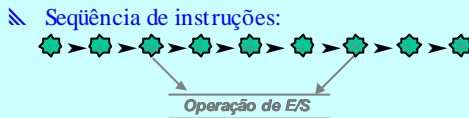
- ❧ Processador é interrompido quando o módulo de E/S está pronto para trocar dados
- ❧ Processador está livre para executar uma outra tarefa
- ❧ Não é necessário aguardar
- ❧ Consome muito tempo do processador porque cada palavra lida ou escrita passa através do processador



Exemplo 1

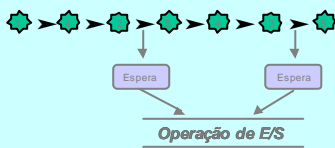
Descrição:

- Programa prepara o módulo e executa o comando de E/S (ex.: impressão)
- Programa de E/S retorna para o programa do usuário
- Código do usuário é executado durante a operação de E/S, sem espera
- Programa do usuário é interrompido (x). A operação de E/S é executada e desviada para a operação de interrupção para examinar o status do módulo de E/S
- Prosseguimento do código do usuário

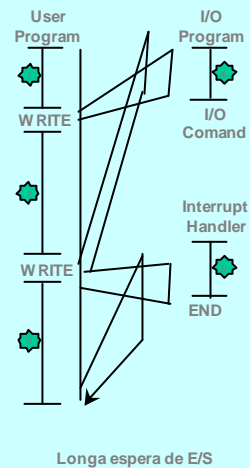


Exemplo 2

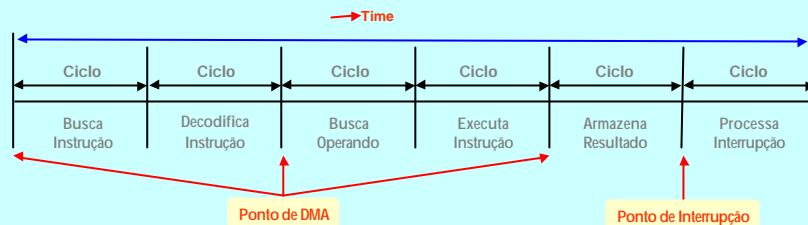
Seqüência de instruções:



(*) A operação de E/S levou mais tempo do que o programa para chegar ao outro WRITE



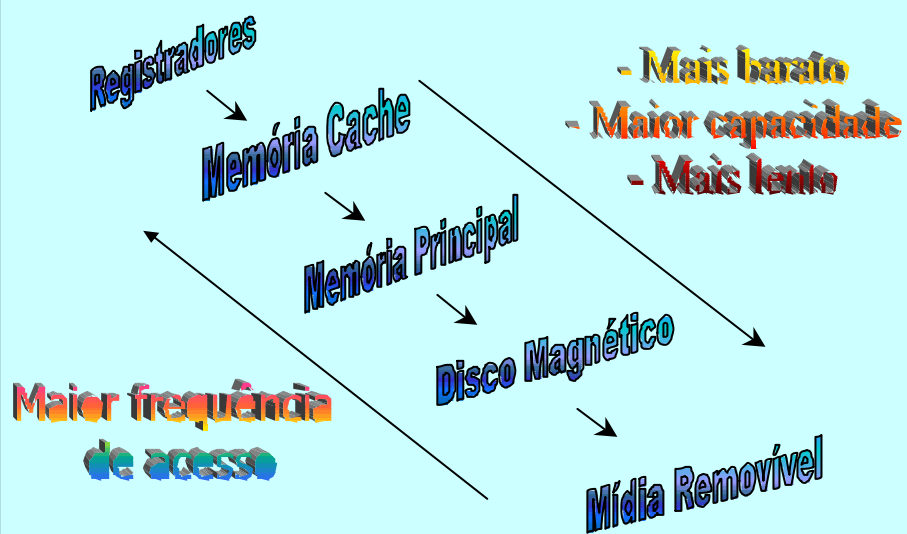
Acesso Direto à Memória



Procedimentos:

- ≡ CPU envia pedido para o módulo DMA
- ≡ Módulo DMA transfere um bloco de dados diretamente para/da memória (sem interrupção da CPU)
- ≡ Uma interrupção é enviada quando a tarefa é finalizada
- ≡ A CPU é envolvida somente no princípio e no final da transferência
- ≡ A CPU está livre para executar outras tarefas durante a transferência de dados

Hierarquia de Memória



Decrescendo a Hierarquia

- **Diminuição do custo por bit**
- **Aumento da capacidade**
- **Aumento do tempo de acesso**
- **Diminuição da frequência de acesso da memória pelo processador**