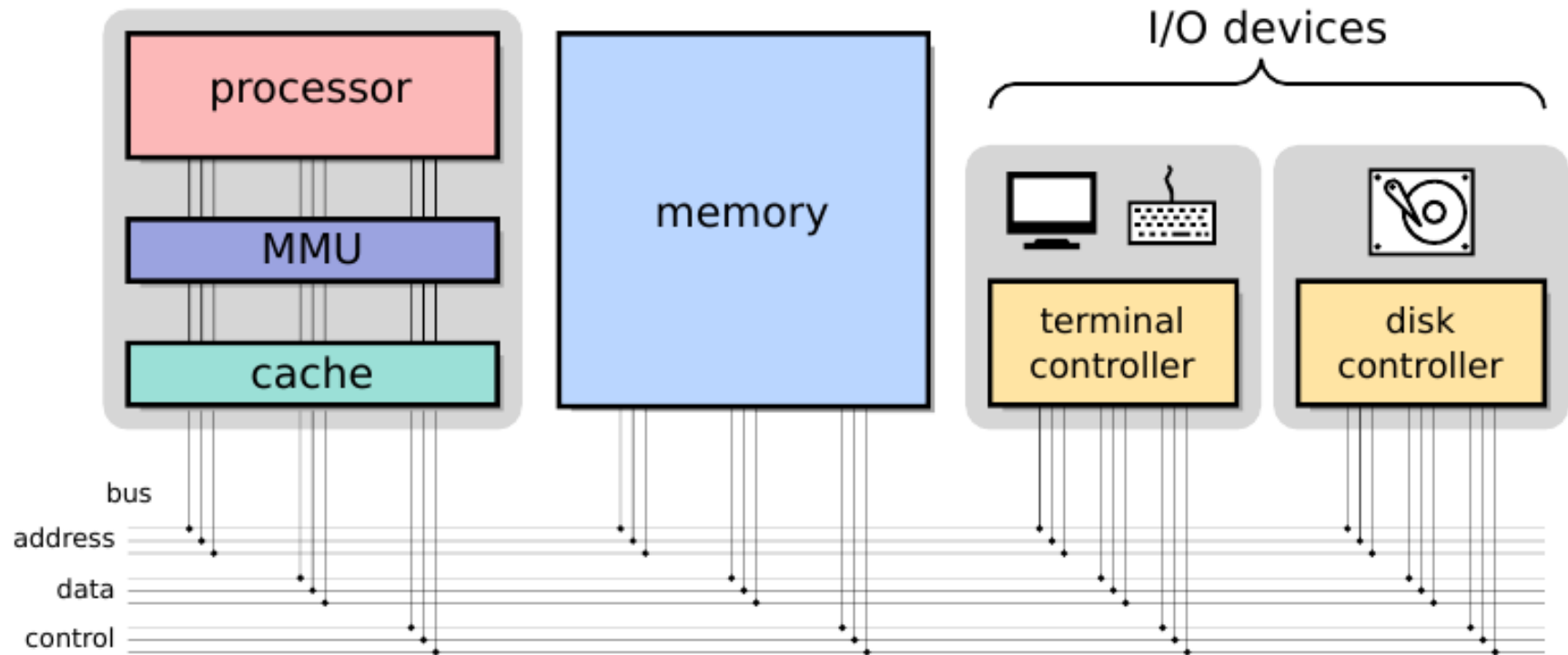


Conceitos básicos

Fonte:
Tanenbaum cap 1

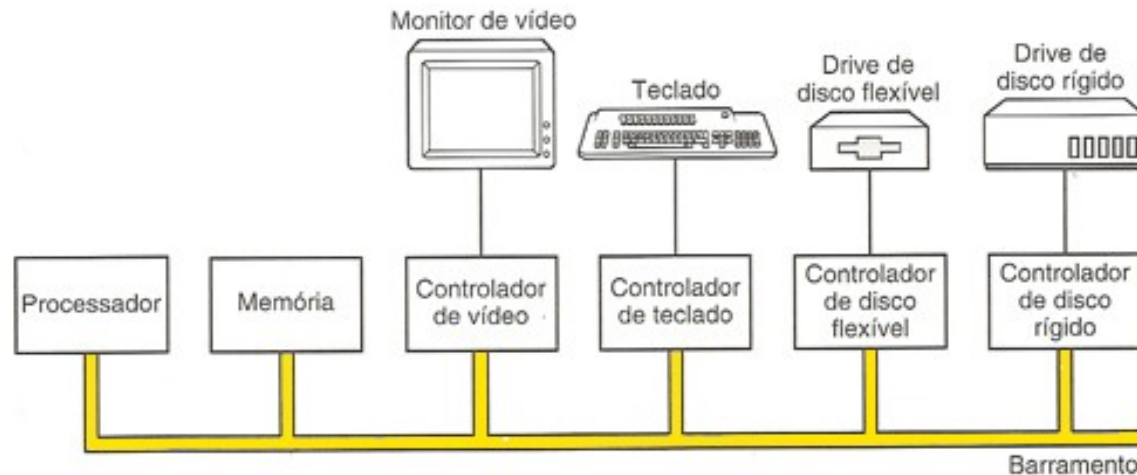
- Sistema Computacional
- Interrupção
- Proteção

Arquitetura de um computador típico (Von Neumann)



Sistema computacional (1)

- 1/+ CPUs + memória + controladores de dispositivos conectados por barramento com acesso à memória compartilhada
- CPU e controladores executam **concorrentemente**
 - Cada controlador é responsável por um tipo de dispositivo



**Concorrência
X
Paralelismo**

Sistema computacional (2)

- Funções básicas de um computador:
 - Processamento de dados
 - Armazenamento de dados
 - Movimentação de dados
 - Controle
- Funções da CPU:
 - Busca de instruções
 - Interpretação de instruções
 - Busca de dados
 - Processamento de dados

Níveis de um sistema computacional



Interrupção

- Evento externo sinalizado ao processador
 - **Suspende** a execução de um processo
- **Desvia** da rotina normal p/ rotina específica
 - tratador de interrupções
 - p/ cada interrupção -- rotina de tratamento
- Vetor de interrupções
 - contém **endereços** de rotinas de serviços
- HW deve salvar o endereço da instrução interrompida (PC)
- **SO → orientado a interrupções**

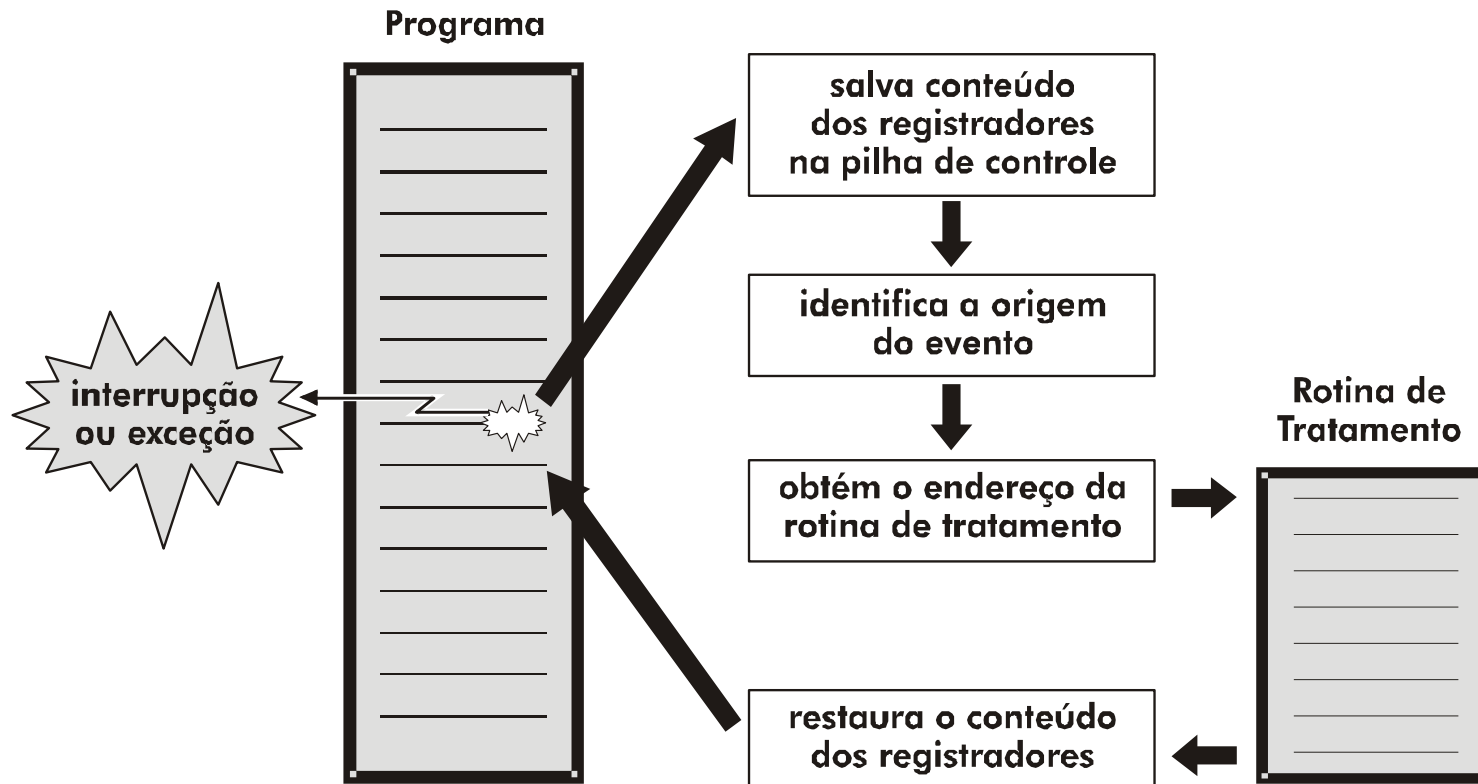
Características das Interrupções

- Desabilitável = **mascarável** (*maskable*)
 - pode ser inibida (desabilitada por SW)
 - garante que outras interrupções possam ser atendidas
- Não desabilitável = **não mascarável** (*non-maskable*)
 - não pode ser inibida, sempre é atendida
 - possui alta prioridade

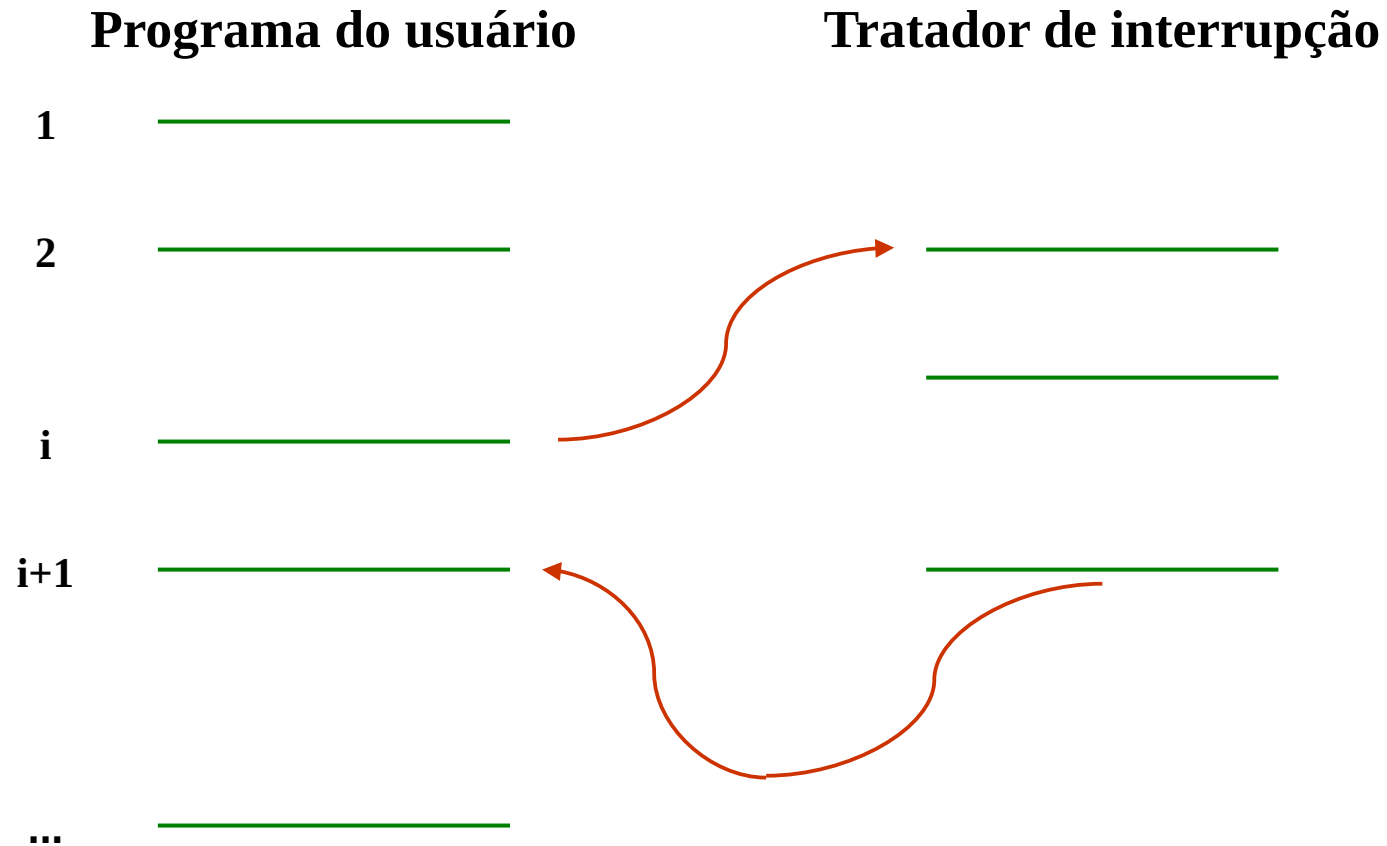
Tipos de Interrupção

- Hardware:
 - sinal enviado p/ CPU via bus
 - admite prioridades e pode ser desabilitada
- Software = “trap”
 - uso mais comum: **chamada de sistema**
 - não pode ser desabilitada

Mecanismo de Interrupção



Transferência de controle

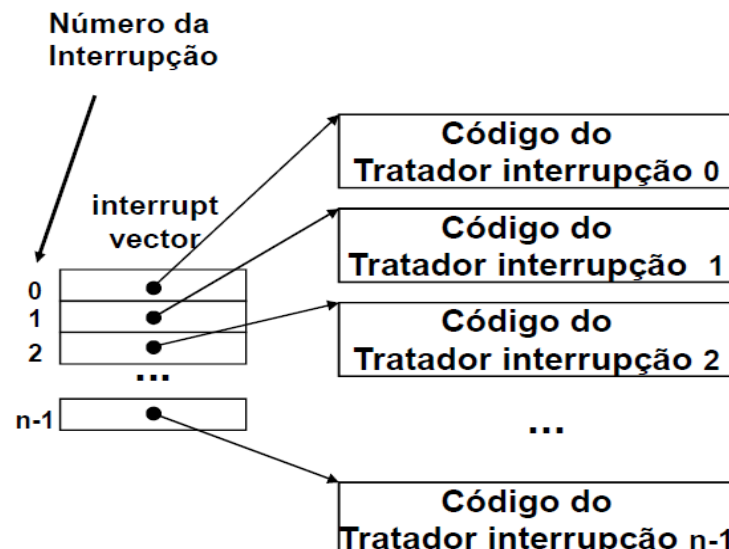


Tratador de interrupção (interrupt handler)

- Programa que determina a natureza da interrupção e realiza o tratamento adequado
- Controle é transferido para este programa após o salvamento de algumas informações de controle
- É parte do SO

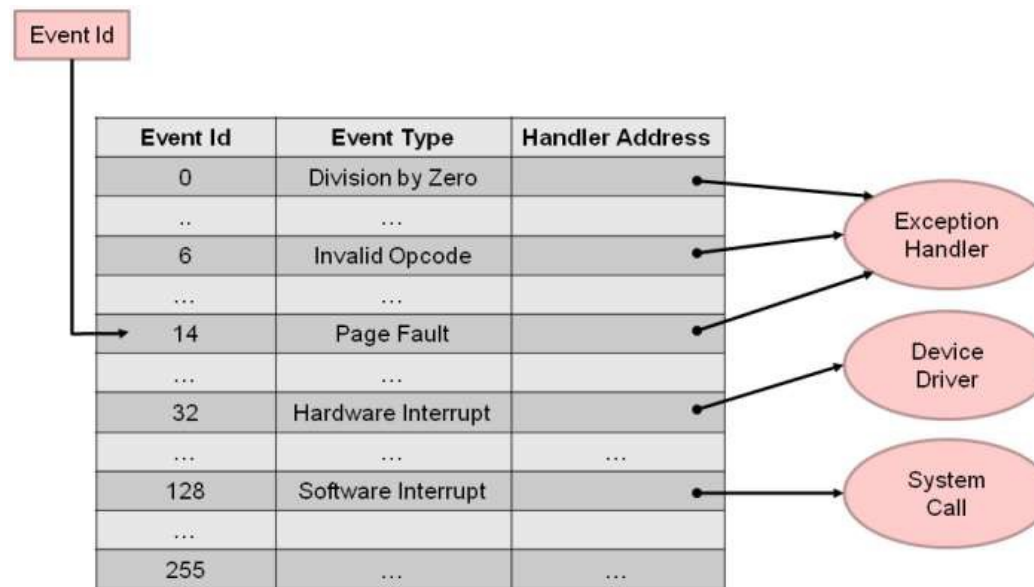
Vetor de Interrupção (1)

- Reside em uma posição fixa da memória RAM ou em uma posição indicada por um registrador da CPU
- Cada interrupção possui um número k (índice) para uma entrada do vetor de interrupção

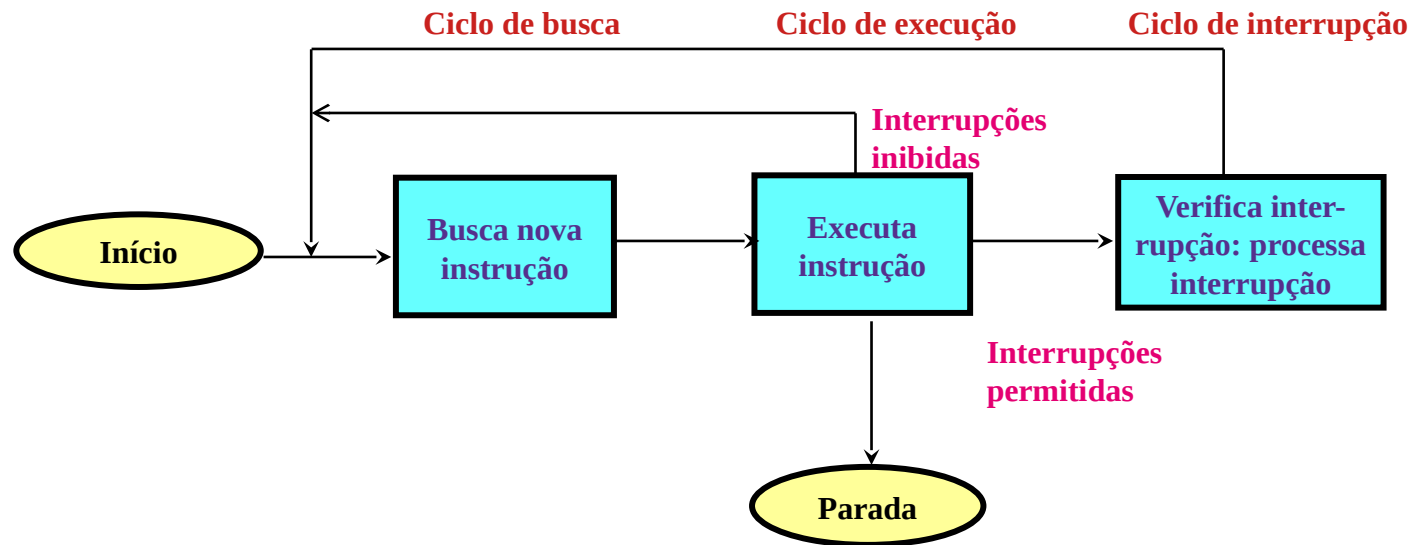


Vetor de Interrupção (2)

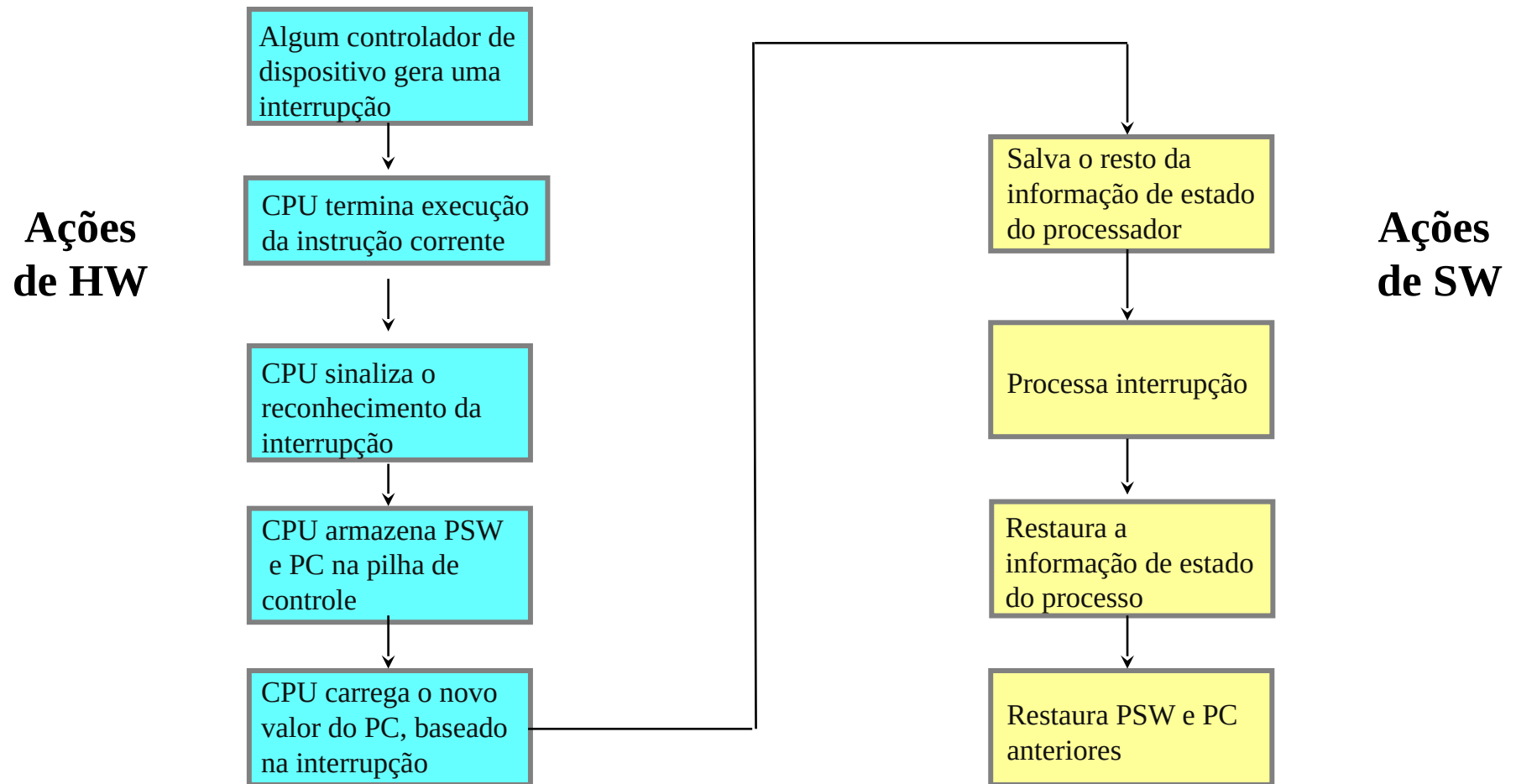
- Cada entrada aponta p/ o endereço da rotina de tratamento da interrupção
 - ex.: se entrada 5 possui o valor 3C20h, a rotina de tratamento da IRQ 5 inicia na posição 3C20h



Ciclo de interrupção

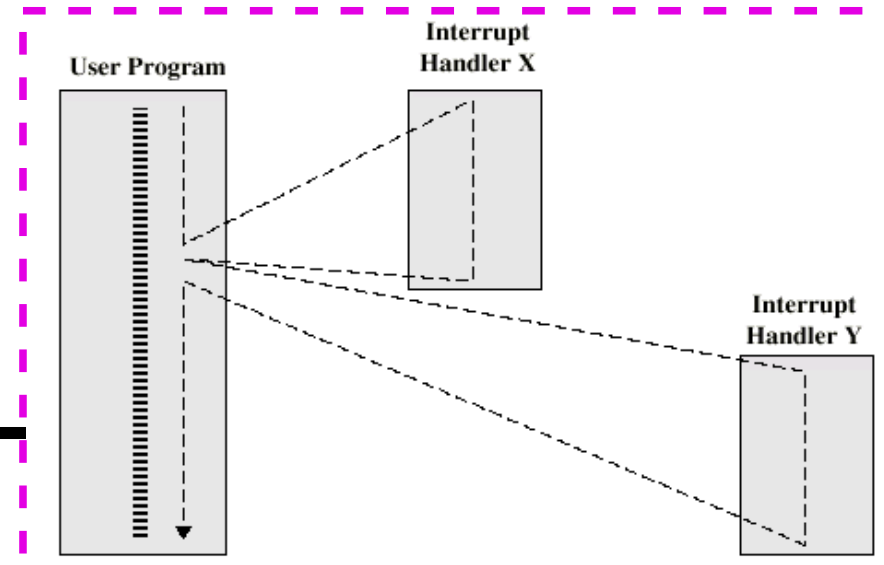


Tratamento da Interrupção: ações



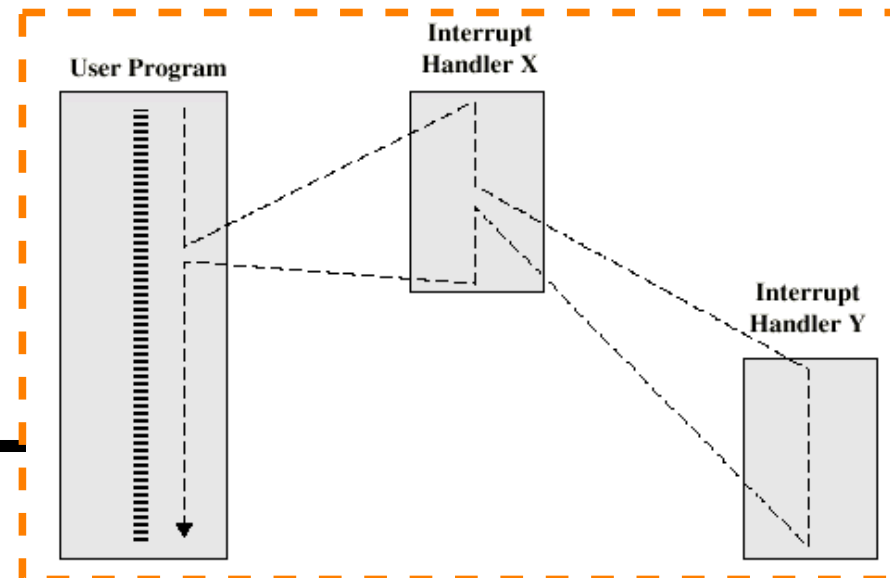
Interrupções múltiplas (1)

- Desabilita outras interrupções: **sequencial**
 - CPU ignora outras interrupções enquanto processa uma interrupção
 - interrupções pendentes somente são verificadas ao fim do tratamento da interrupção corrente
 - interrupções tratadas na ordem sequencial de ocorrência
 - quando terminar o tratamento, as interrupções são habilitadas
- Vantagem: simplicidade
- Desvantagem: falta de critério



Interrupções múltiplas (2)

- Definição de **prioridades**
 - interrupções de menor prioridade podem ser interrompidas pelas de maior prioridade
 - quando terminar o tratamento das mais prioritárias, a CPU trata as de menor prioridade
- Resolve o problema de falta de critério



Ex.: tratamento por prioridade

t=0 instruções sendo executadas

t=10 interrupção devido à impressora
executa ações de HW e SW

t=15 rotina de tratamento ainda sendo executada
linha de comunicação interrompe
rotina é interrompida e o estado é salvo
trata interrupção devido à linha de comunicação

t=25 termina o tratamento devido à linha de comunicação
restaura o tratamento devido à impressora

...

Interrupções mais comuns

- Programa
 - Ex.: divisão por zero – *trap*
- Temporização
 - Ex.: escalonamento de processo
- E/S
 - Ex.: fim de escrita em disco
- Falha de HW
 - Ex.: falta de energia

Interrupção x Exceção

- Exceção
 - resultado direto da execução de uma instrução do próprio programa → evento interno
 - erros durante execução (ex.: divisão por zero, acesso inválido à memória, overflow, ...)
 - considerada “interrupção síncrona”
 - objetivo: controlar o sistema (evitar travamento)
- Exceção x Interrupção
 - Exceção → gerada por um evento **síncrono**
 - quando é resultado direto da execução do programa corrente
 - Interrupção → gerada por um evento **assíncrono**
 - quando ocorre independentemente da execução do programa corrente

Interrupção e Proteção

- Interrupções auxiliam na proteção
 - objetivo: proibir o uso de determinadas instruções pelos programas de usuário
 - Instruções proibidas = privilegiadas (protegidas)
- **Instruções privilegiadas:**
 - Instruções perigosas – uso incorreto (intencional/não) pode gerar problemas para o SO
 - Uso exclusivo do SO
 - Ex.: instruções para desabilitar interrupções, instruções para acionar periféricos, instrução para setar o timer, ...

Proteção de HW

- Objetivo: melhorar a utilização do sistema
 - SO compartilha recursos entre diversos programas simultaneamente
 - ♦ erro em um programa
 - ♦ execução de instrução ilegal
 - ♦ acesso a endereço de memória não permitido



podem afetar outros programas ou o próprio SO

HW causa um *trap* p/ SO

Proteção de HW

- Modo de acesso
 - Operação em modo dual
- Proteção de memória
- Proteção de E/S
- Proteção de CPU

Modo de operação (1)

- Mecanismo para impedir problemas de segurança e violação do sistema
- Processadores possuem 2 modos de operação:
 - **Modo usuário/normal**
 - Aplicação só pode executar instruções não privilegiadas
 - Acesso a um número reduzido de instruções
 - **Modo monitor/kernel/sistema/privilegiado**
 - Aplicação pode ter acesso ao conjunto total de instruções do processador

Modo de operação (2)

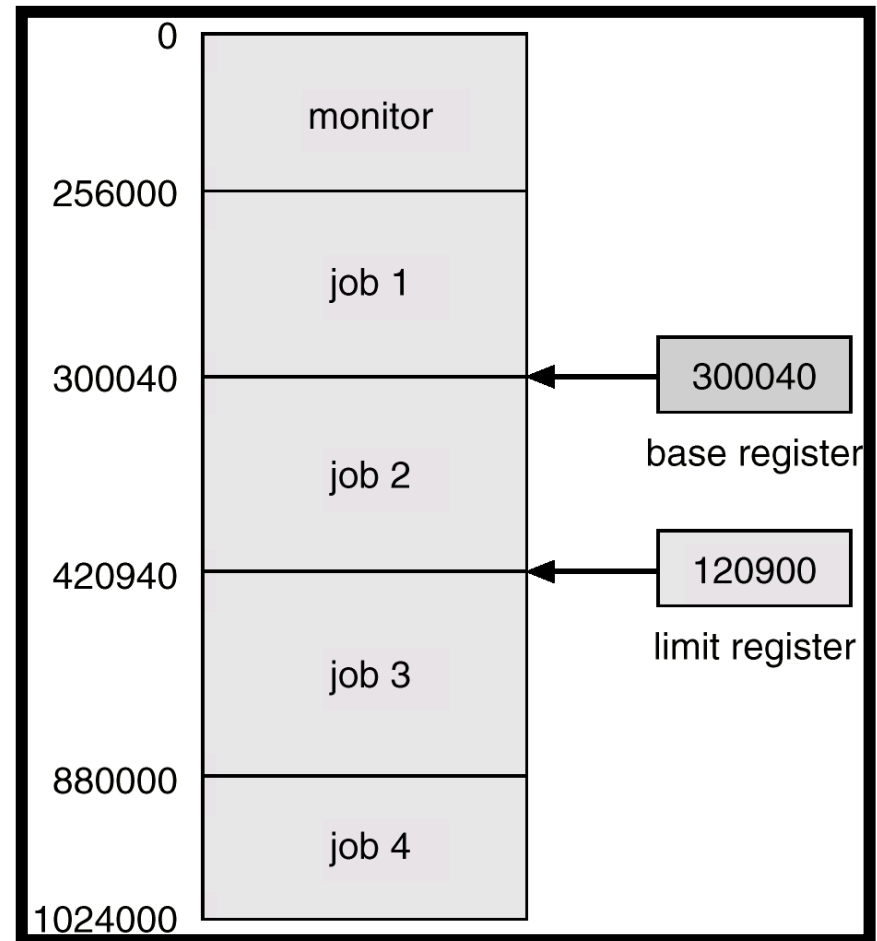
- Determinado por um conjunto de bits, localizado no registrador de status (PSW – program status word) do processador, que indica o modo de acesso corrente
- Por meio do registrador PSW, o HW verifica se a instrução pode ou não ser executada

Proteção de E/S

- Programa do usuário tentar executar instruções de E/S ilegais
 - ex.: acesso à posição de memória do SO, recusa em abandonar CPU
- Instruções de E/S → privilegiadas
 - SO → *system call*
- Programa do usuário não deve receber controle em modo monitor

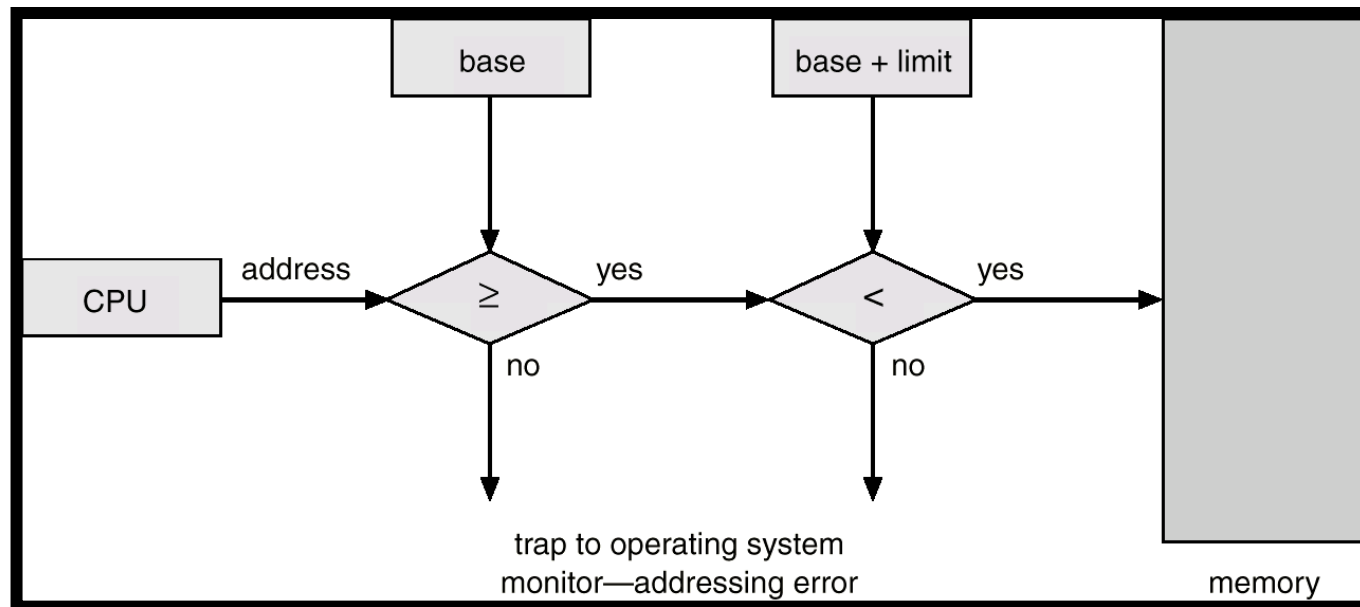
Proteção de memória (1)

- Mínima: p/ vetor de interrupções e rotinas de serviço do SO
- Registrador **base**: 1º end. de memória válido
- Registrador **limite**: tamanho da região
- Instr. carga p/ base e limite → instr. privilegiadas



Proteção de memória (2)

- Implementação da proteção de memória:
 - endereço gerado em modo usuário é comparado com base e limite
 - violação trap p/ o monitor (*fatal error*)



Proteção de CPU

- Timer
 - interrompe processo após determinado período de tempo
 - garante que SO mantém o controle
 - evita loop infinito
 - usado para implementar time-sharing
 - carga do timer → instrução privilegiada