

Conceitos básicos

- Processo de Boot
- Interrupção
- Proteção

Sistema de computador

- 1 CPU + N controladores de dispositivos conectados por barramento com acesso a memória compartilhada
- CPU e controladores executam concorrentemente
- cada controlador é responsável por um tipo de dispositivo
 - controlador de memória sincroniza acesso à memória compartilhada (garante acesso “organizado”)

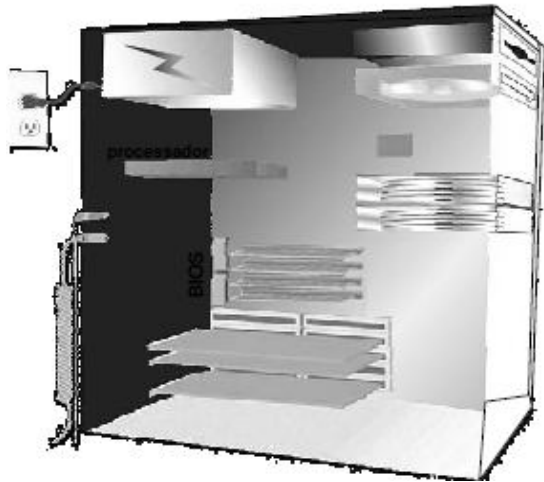
SO: composição básica

- Um SO consiste, basicamente, de um **núcleo** (kernel), alguns **programas do sistema** e **aplicações utilitárias** que executam diversas tarefas
- Kernel
 - Parte central do SO (núcleo)
 - Responsável pela carga do sistema (boot)

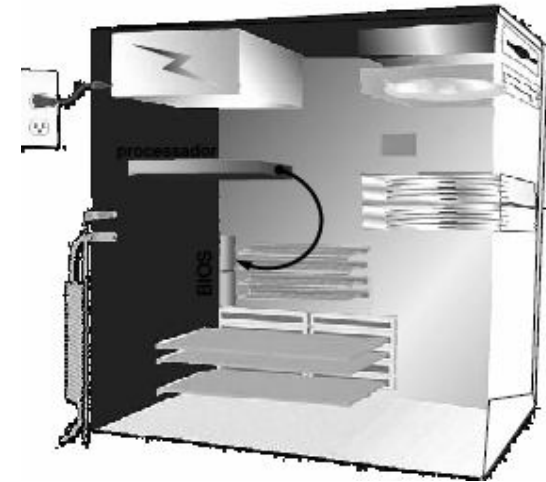
Booting (1)

Basic Input/Output System

1) fonte de alimentação
fornece energia elétrica p/
as diferentes partes do
sistema



2) Processador procura a BIOS
(firmware c/ instruções de
inicialização)

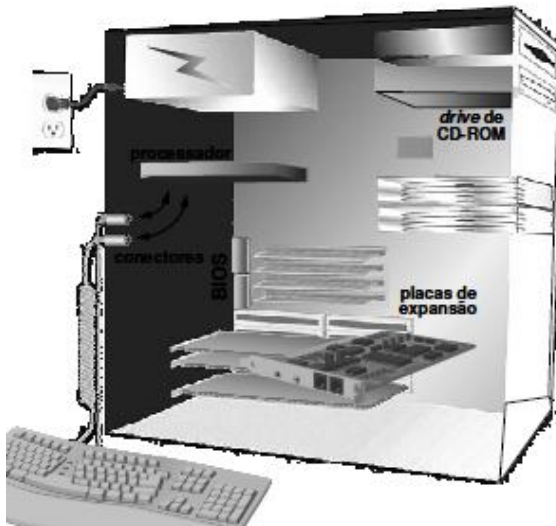


Booting (2)

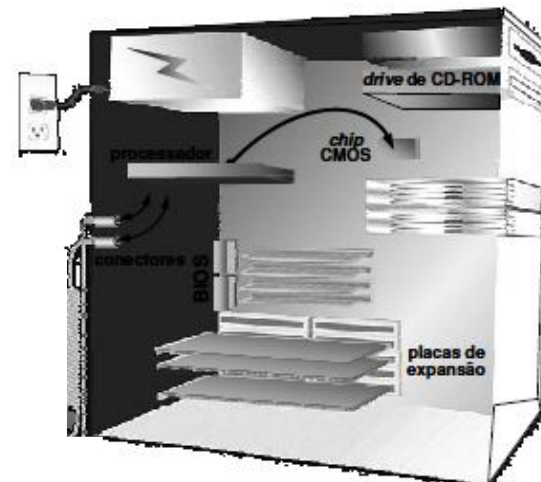
Complementary Metal
Oxide Semiconductor

- 3) BIOS realiza o
POST – teste de verificação
de compon. (mouse, teclado,
conectores e placas de
expansão)

Power-On
Self Test

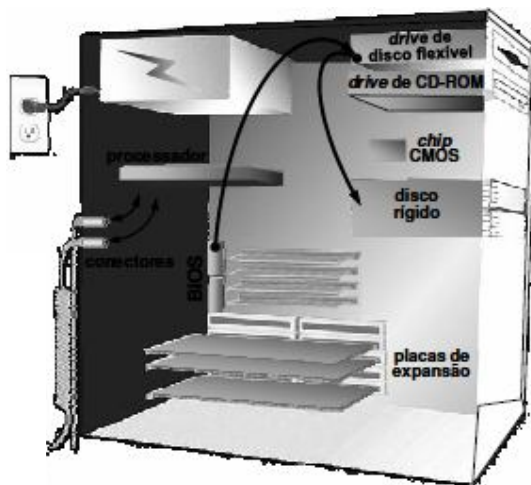


- 4) Resultados do POST são
comparados c/ dados
armazenados no chip CMOS
(armazena inf. de configuração
do comp. e detecta novos
dispositivos)

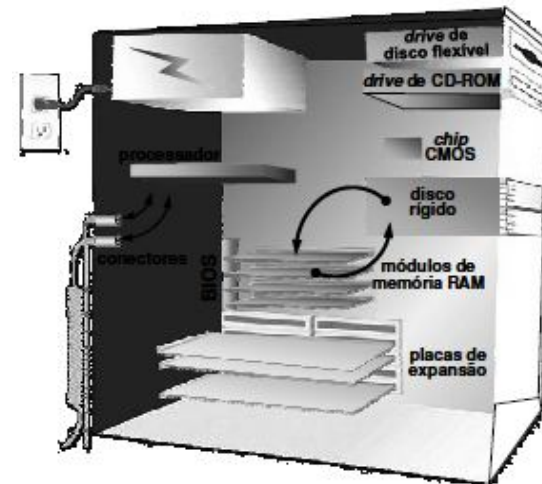


Booting (3)

5) BIOS procura os arquivos de sistema no disco



6) Programa de boot carrega na RAM o kernel do SO, o qual inicia executando o processo inicial e espera a ocorrência de um evento (interrupção)



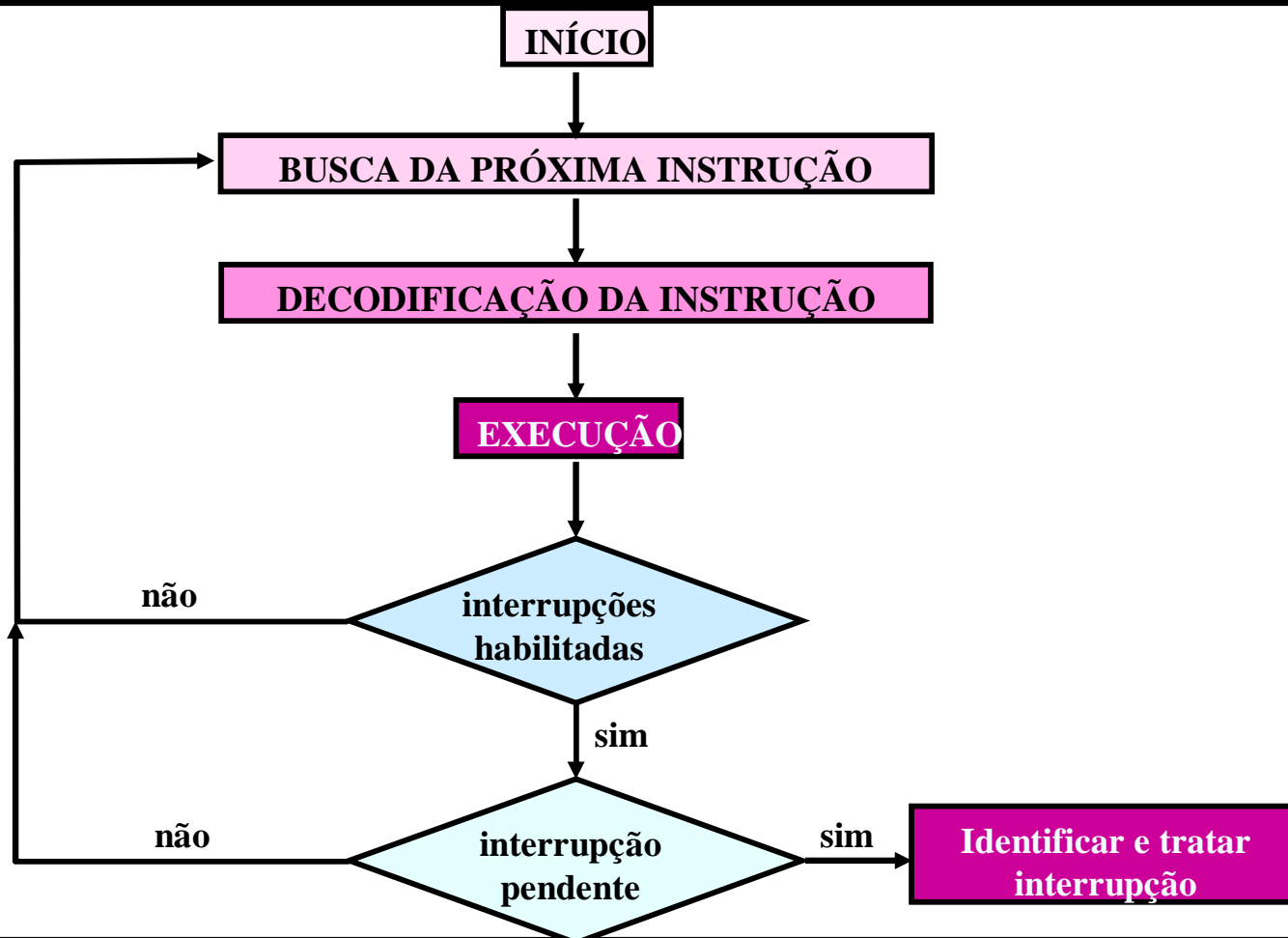
Operação de um computador: exemplo Linux

1. Processador executa código da BIOS
2. BIOS conhece dispositivos bootáveis e carrega o programa de 1º estágio de boot
3. Programa de 1º estágio de boot (512k) carrega programa do 2º estágio
4. Programa do 2º estágio (GRUB, LILO), que possui sistema de arquivos básico, localiza, carrega, descomprime e executa o kernel
5. Kernel carrega 1º processo de usuário (inicializador de processos): processo *init*

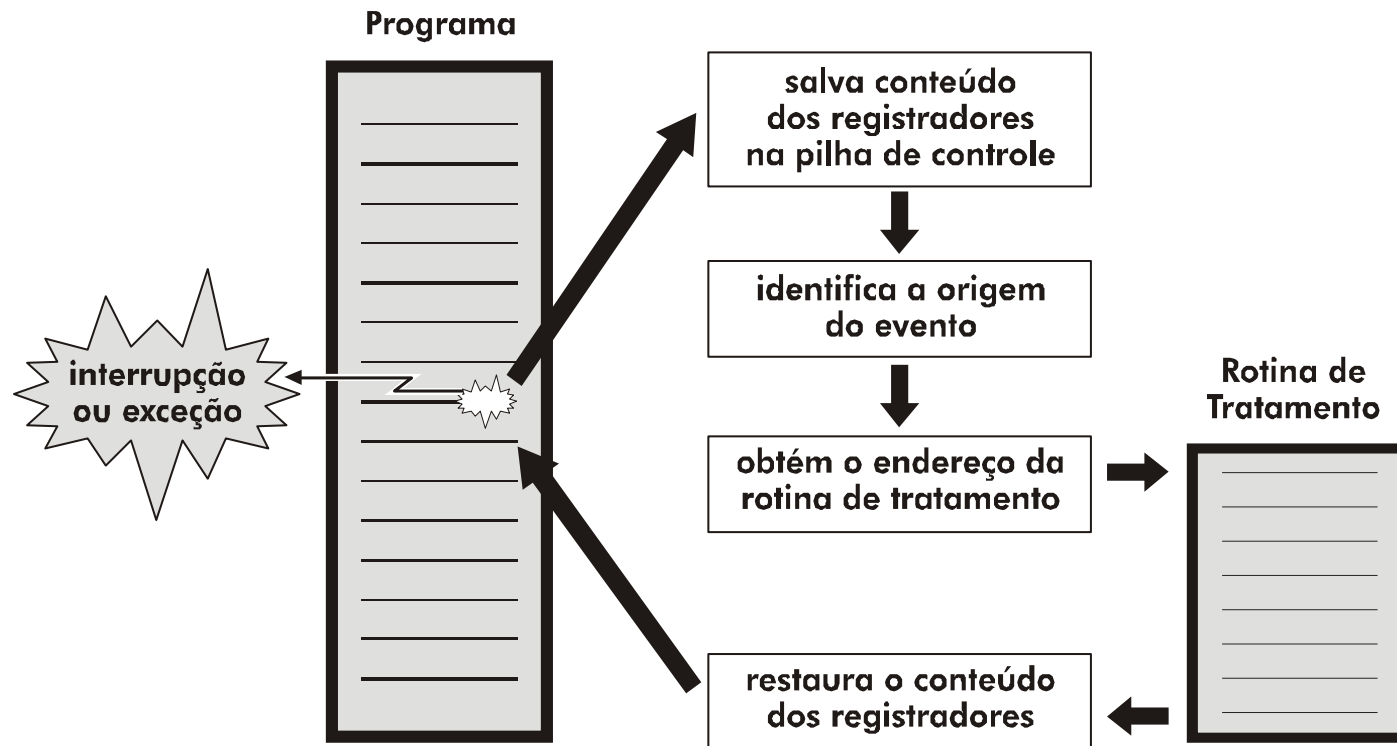
Interrupção

- Evento externo sinalizado ao processador
 - Suspende a execução de um processo
 - desvia da rotina normal p/ rotina específica
 - tratador de interrupções
 - p/ cada interrupção \Rightarrow rotina de tratamento
 - vetor de interrupções
 - contém endereços de rotinas de serviços
 - HW deve salvar endereço da instrução interrompida (PC)
 - SO \rightarrow orientado a interrupções
-

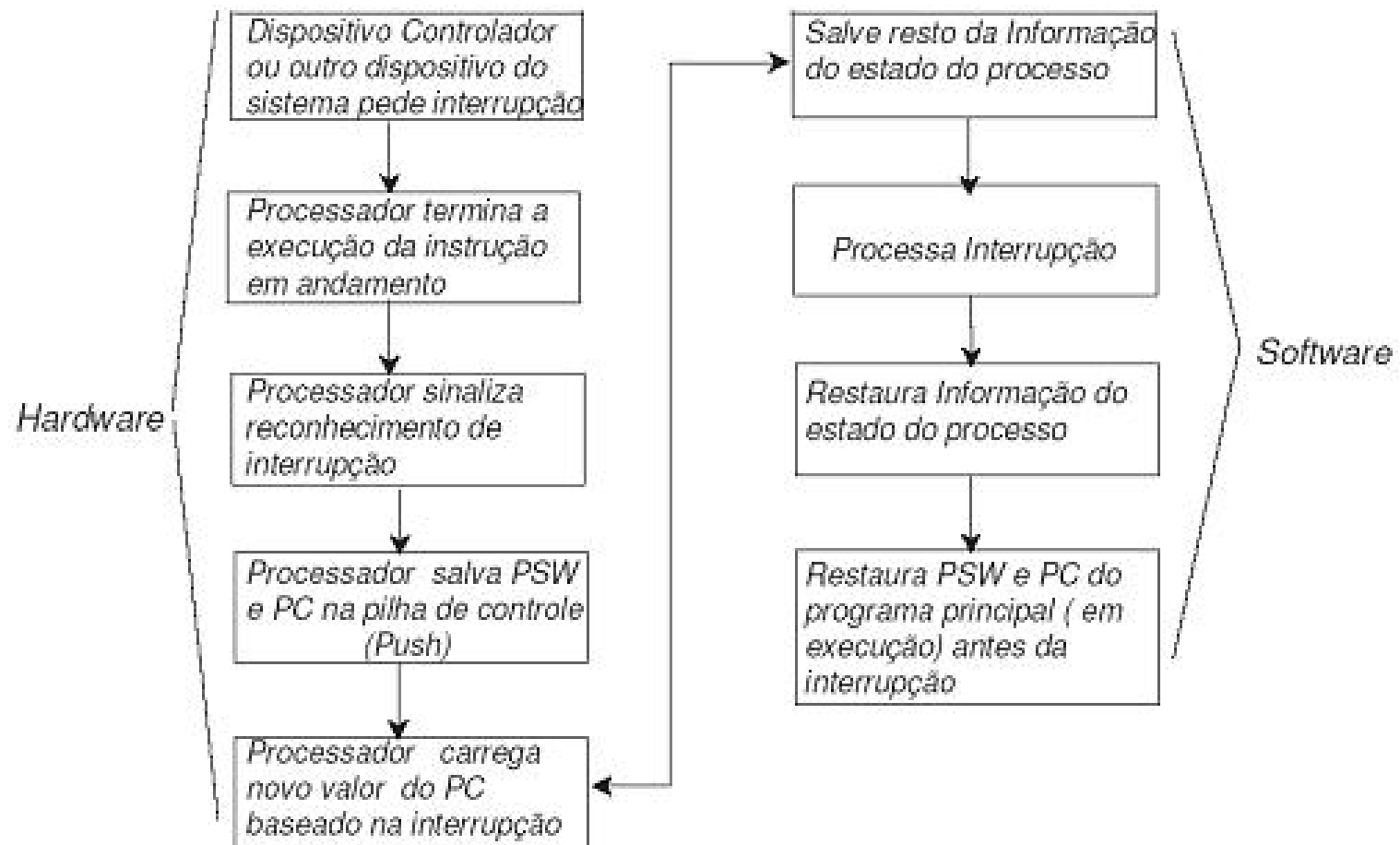
Ciclo de instrução com interrupção



Mecanismo de Interrupção

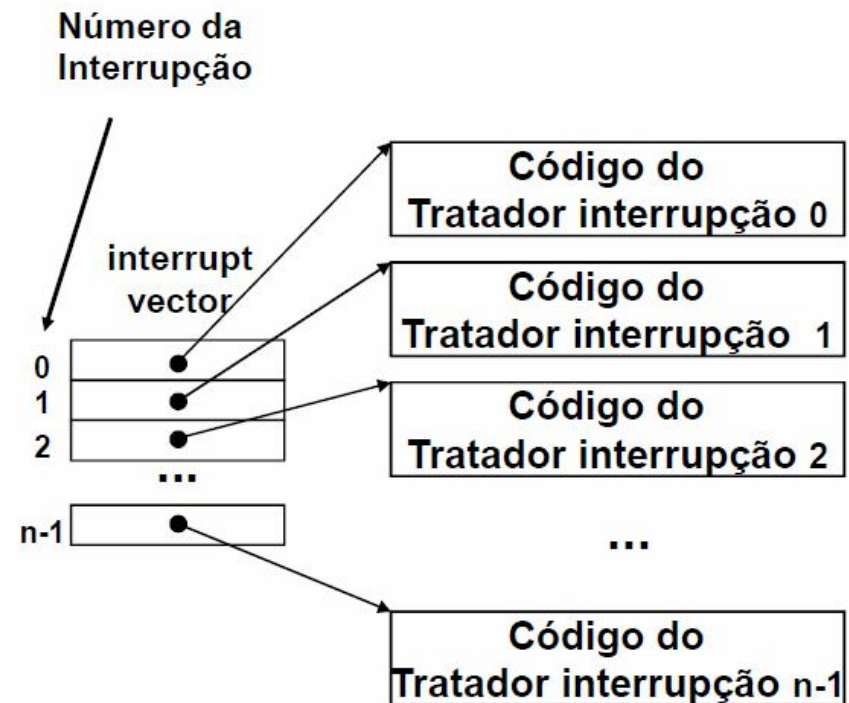


Mecanismo de Interrupção: ações de HW e SW



Vetor de Interrupção

- Cada interrupção possui um número $k \rightarrow$ índice para uma entrada do vetor de interrupção
- Cada entrada aponta p/ o endereço da rotina de tratamento da interrupção



Características das Interrupções

- Desabilitáveis = mascaráveis (*maskable*)
 - podem ser inibidas (desabilitadas por SW)
 - garante que outras interrupções possam ser atendidas
- Não desabilitáveis = não mascaráveis (*non-maskable*)
 - não podem ser inibidas, sempre são atendidas
 - possuem 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

Interrupção x Exceção

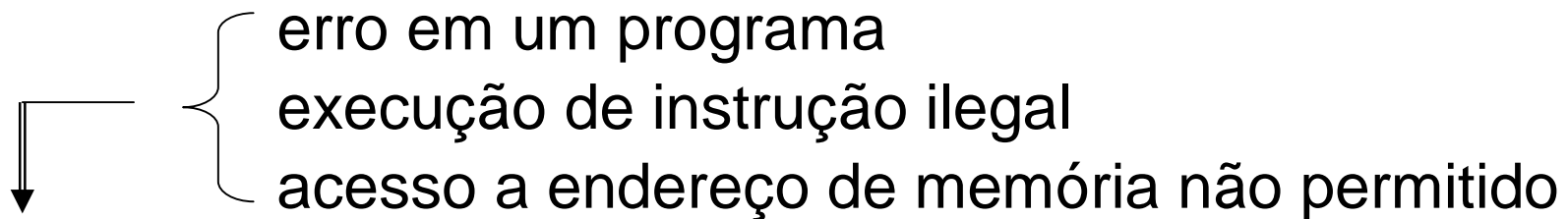
- Exceção:
 - resultado direto da execução de uma instrução do próprio programa
 - erros durante execução (ex.: divisão por zero, acesso inválido à memória, overflow, ...)
 - considerada “interrupção síncrona”
 - objetivo: controlar o sistema (evitar travamento)
- **exceção x interrupção**
 - exceção é gerada por um evento **síncrono** (quando é resultado direto da execução do programa corrente)
 - interrupção é gerada por eventos **assíncronos** (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 = protegidas/privilegiadas
 - **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, ...
-

Proteção de HW

- Objetivo: melhorar a utilização do sistema
 - SO compartilha recursos entre diversos programas simultaneamente



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)

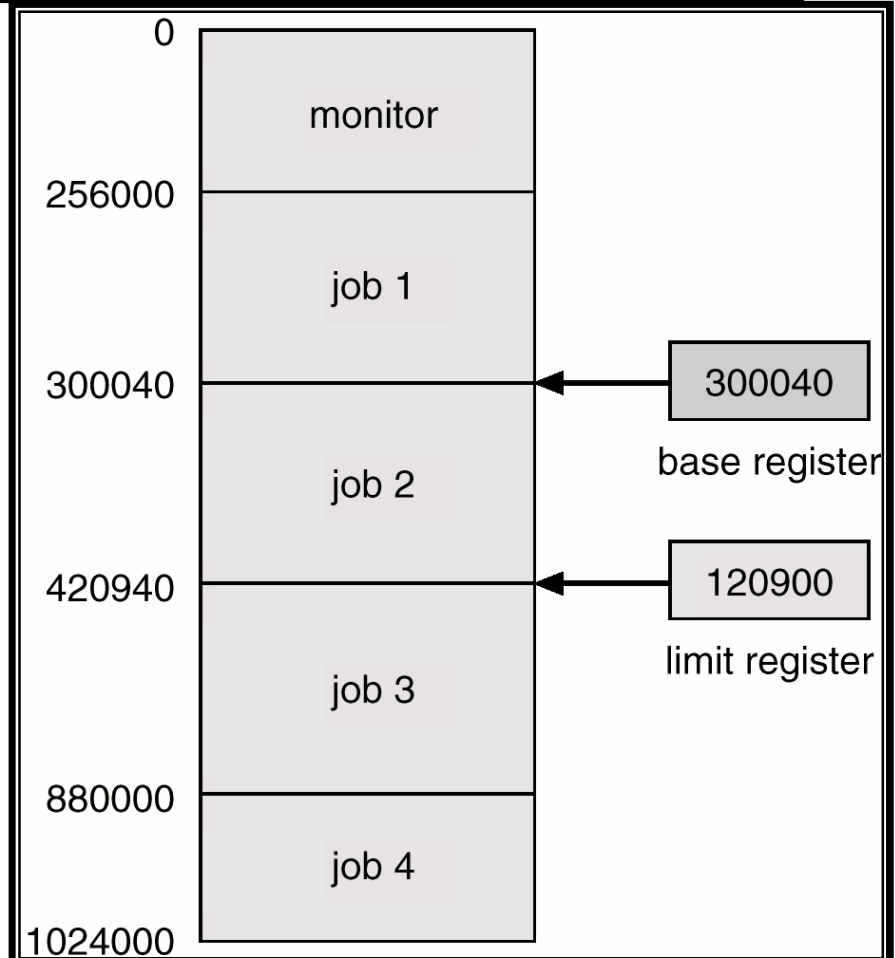
- Determinado por um conjunto de bits, localizado no registrador de status (PSW) do processador, que indica o modo de acesso corrente
- Por meio desse registrador, o HW verifica se a instrução pode ou não ser executada

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

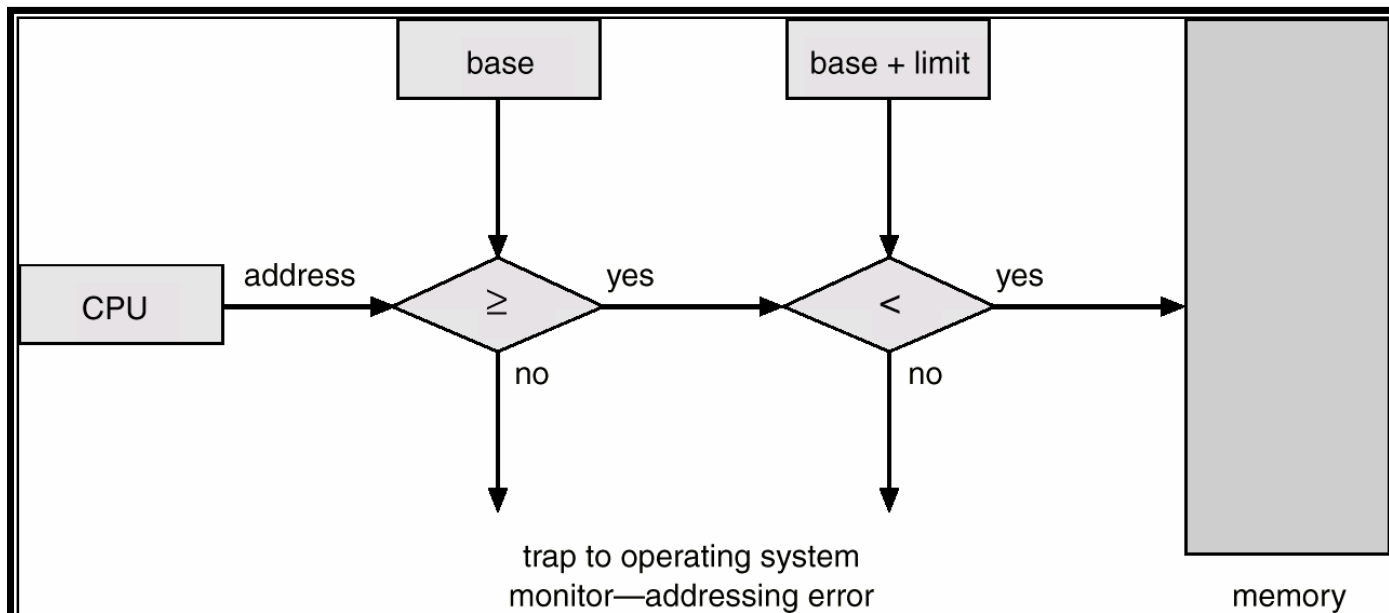
Proteção de memória (1)

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