

Sistemas Operacionais



Conceitos Básicos

Prof. José Roberto Bezerra

Agenda

- ▣ Conceito de sistema operacional
- ▣ Conceito de sistema computacional
- ▣ Elementos de um SC
 - ◆ CPU
 - Ciclo de instruções
 - Pipeline
 - Superescalar
 - Modo núcleo/usuário
 - ◆ Memória
 - ◆ E/S
 - *Busy waiting*
 - Interrupções
 - DMA
 - Barramentos

Sistema Operacional

- ▣ Funções de um sistema operacional?
 - ◆ Estender a máquina
 - ◆ Gerenciar recursos

SO como máquina estendida

- ▣ Apresentar ao usuário/programador uma máquina estendida e acessível e ainda que seja mais fácil de programar do que o *hardware* diretamente.
- ▣ Fornece uma variedade de serviços que os programas podem obter usando instruções especiais (chamadas de sistema)

SO como máquina estendida

- ▣ Camada intermediária que facilite e uniformize o acesso aos recursos computacionais
 - ◆ Memórias
 - ◆ Dispositivos de E/S

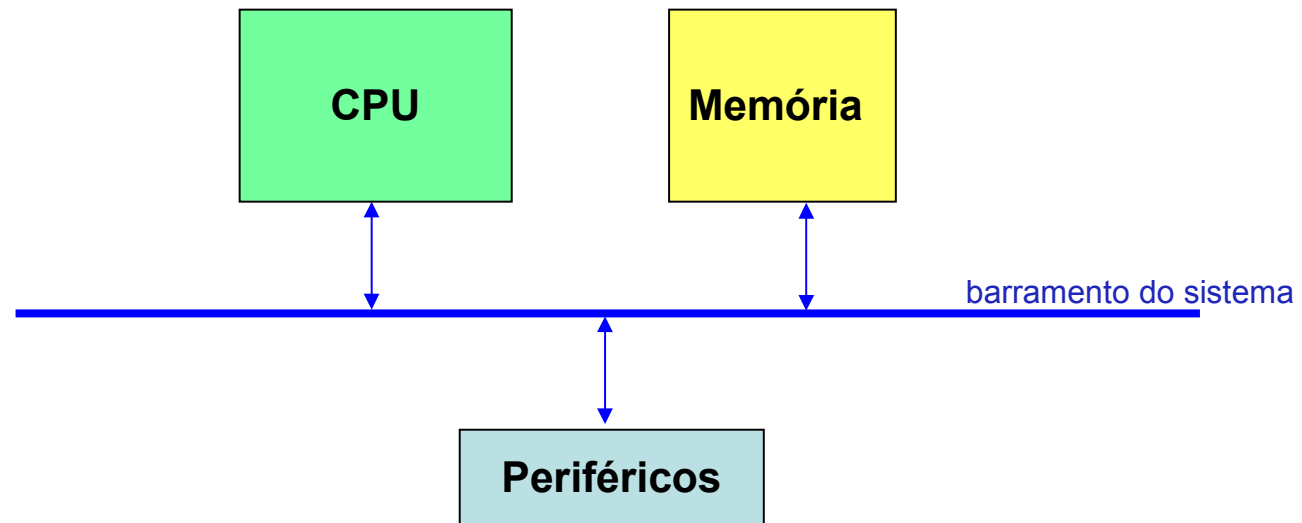
SO como gerenciador de recursos

- Fornecer uma alocação ordenada e controlada de processadores, memórias e dispositivos de E/S entre vários programas que competem pelo uso destes recursos

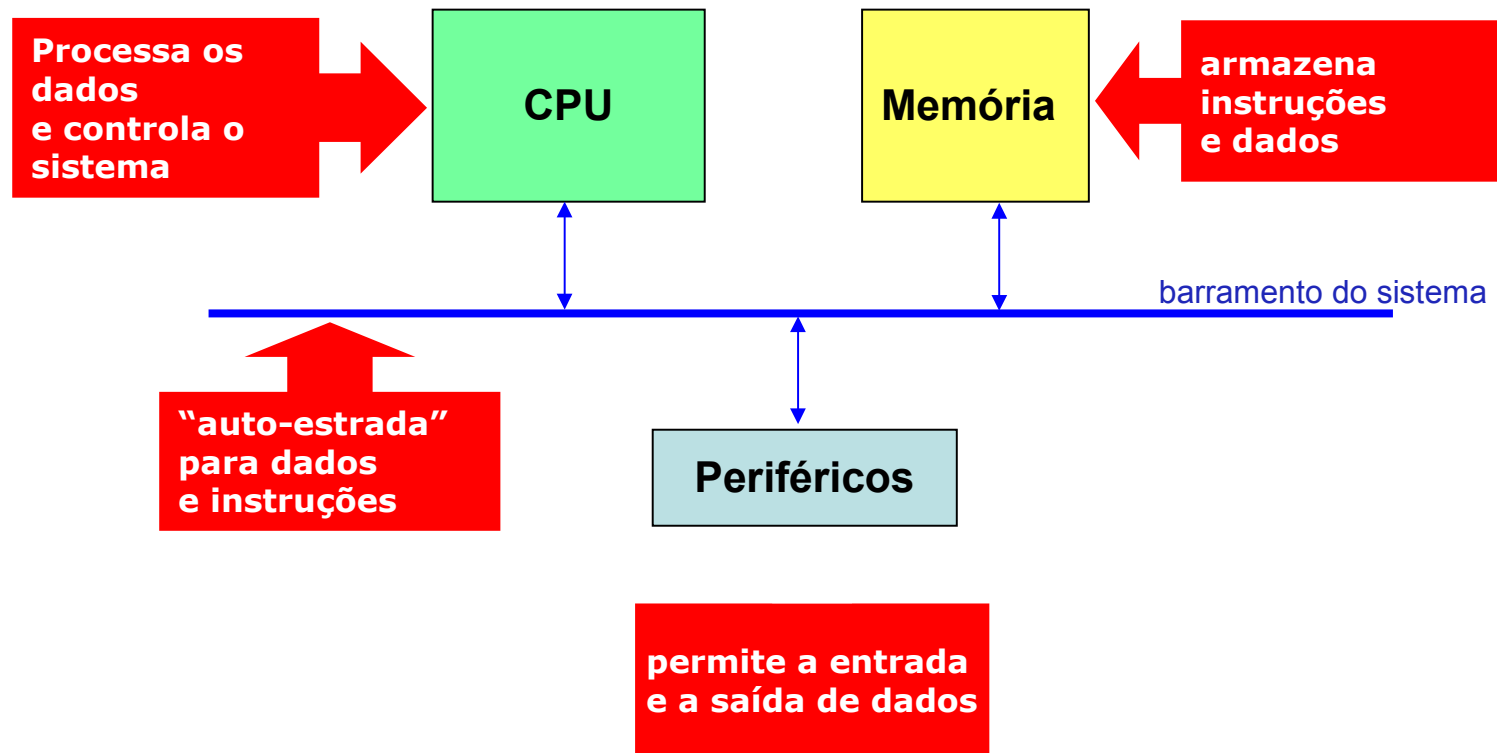
Sistema Computacional

- ▣ O que é um sistema computacional?
- ▣ Onde encontramos um sistema computacional?
- ▣ Qual a finalidade de um sistema computacional?

Sistema Computacional



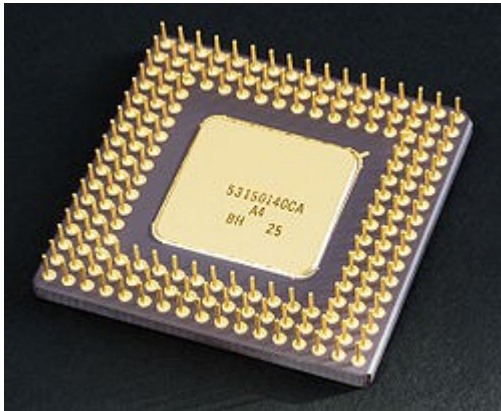
Sistema Computacional



Exemplos de SC

- *Mainframes*
- *Desktops*
- *Laptops*
- *Palmtops*
- *Celulares/Smartphones*
- *Tablets*
- *Alguns eletrodomésticos*

Processadores (CPU)



- Funções essenciais
- Ciclo básico
 - ◆ Busca instruções na memória
 - ◆ Decodifica para determinar os operandos
 - ◆ Executa
- Transferência de dados entre memória e dispositivos de entrada/saída (E/S)
- Responder interrupções externas
- Prover sinais de temporização e controle

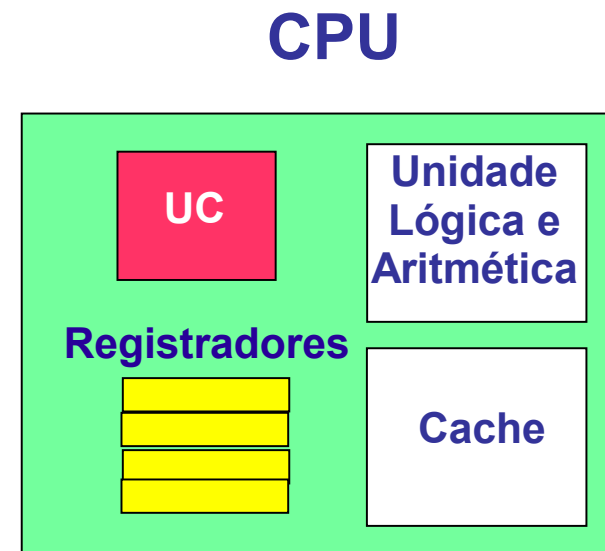
Instruções

- Cada CPU possui um conjunto específico de instruções

- ◆ Aritméticas
- ◆ Lógicas
- ◆ Transferência de dados
- ◆ Desvio

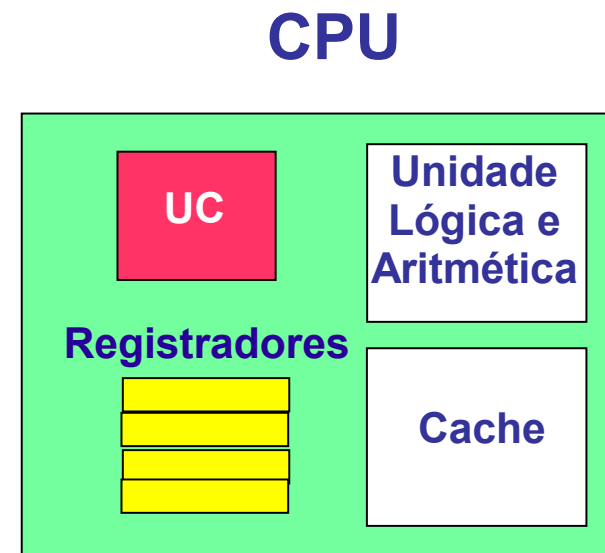
- Registradores internos

- ◆ *Program Counter*
- ◆ *PSW (Program Status Word)*



Instruções

- ▣ Registradores internos
 - ◆ *Program Counter (PC)*
 - ◆ *PSW (Program Status Word)*



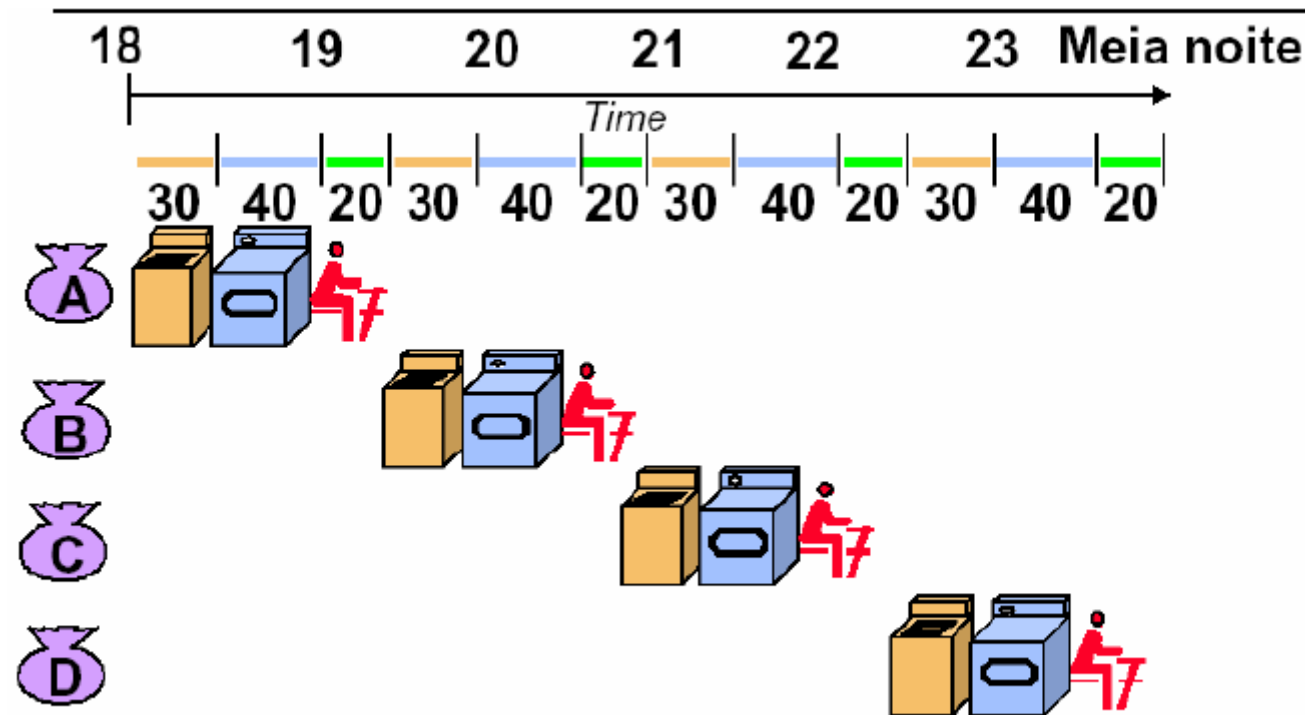
Ciclo convencional



Pipeline

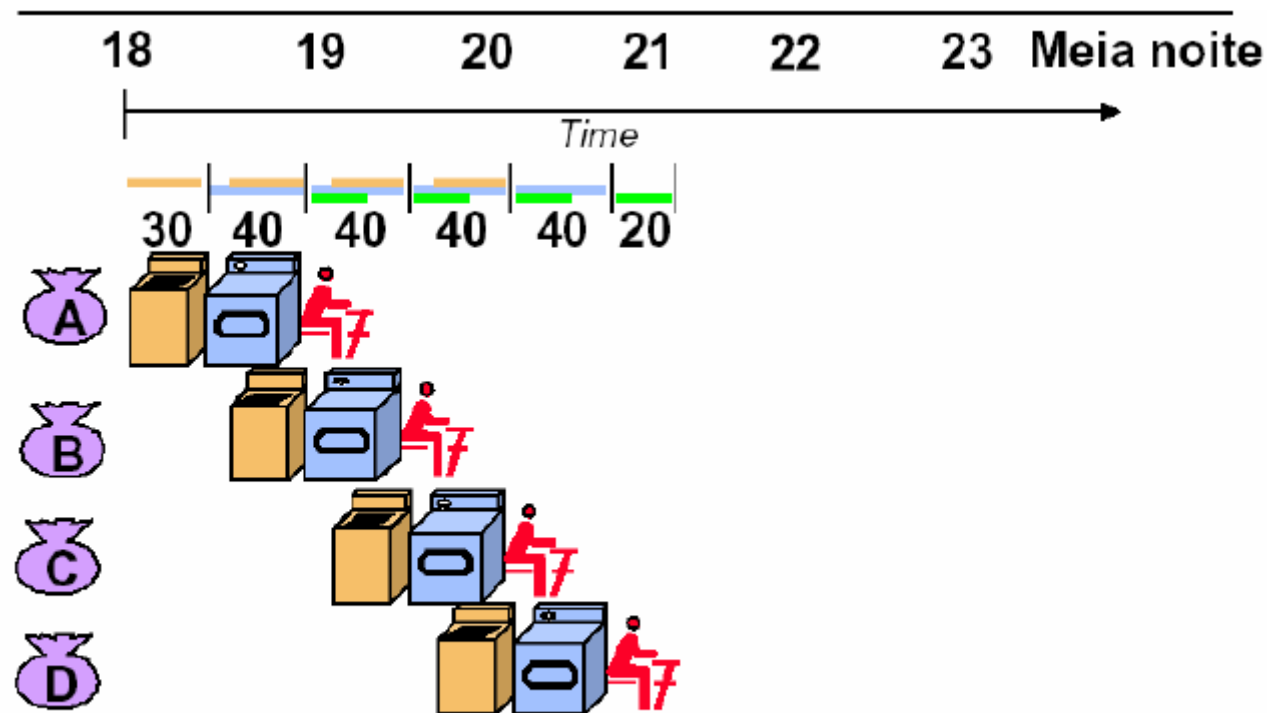
- Consiste em aceitar novas instruções antes que as aceitas previamente tenham terminado
- Permite a CPU 'executar' mais de uma instrução ao mesmo tempo
- Unidades independentes de busca, decodificação e execução
- Enquanto executa a instrução n , decodifica a instrução $n+1$ e busca a instrução $n+2$

Lavanderia sequencial



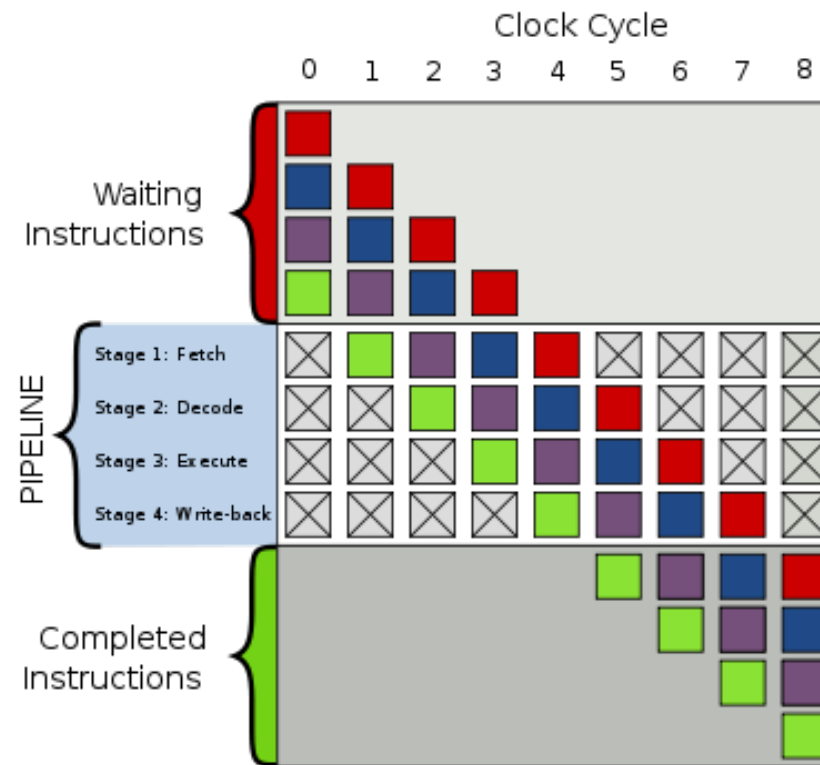
➤ Lavanderia sequencial: 6 horas para 4 cargas

Lavanderia com Pipeline



► Lavanderia pipeline levaria 3.5 horas para 4 cargas

Pipeline



Superescalar

- Arquiteturas Superescalares são aquelas capazes de buscar, decodificar e executar mais de uma instrução por ciclo de máquina
- Possui múltiplas unidades de execução
 - ◆ Números inteiros
 - ◆ Ponto Flutuante
 - ◆ Operações lógicas
- A cada ciclo são buscadas e decodificadas duas ou mais instruções e colocadas em um *buffer*
- A medida que uma unidade de execução está livre as instruções são executadas

Pipeline simples

1

B
u
s
c
a

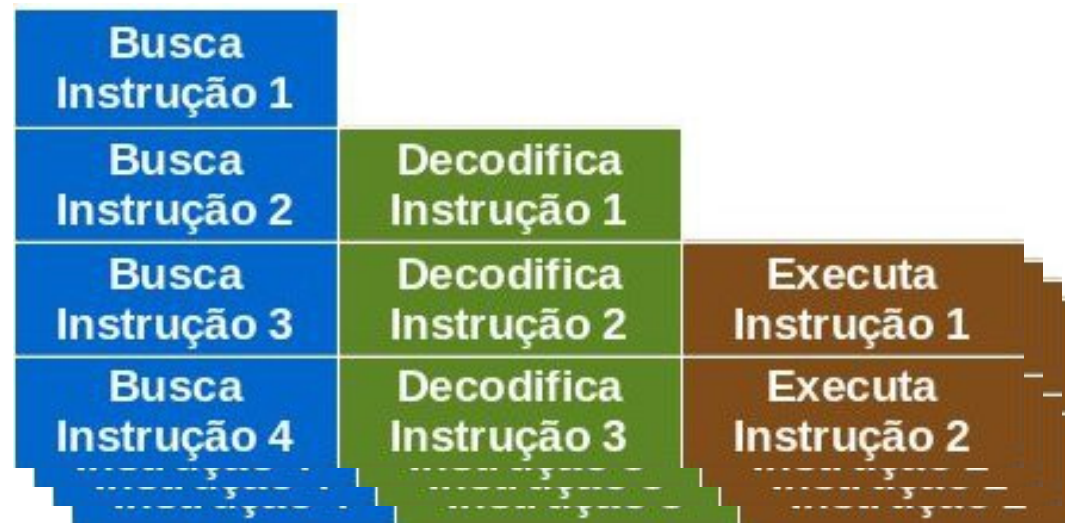
I
n
s
t
r
u
ç
ã
o

1

B
u
s
c

Decodifica
Instrução 1

Superescalar *Pipeline*



Modo Núcleo e Usuário

- Controlado pelo PSW
- Modo Núcleo
 - ◆ Qualquer instrução pode ser executada
 - ◆ Sistema operacional roda neste modo
- Modo Usuário
 - ◆ Não permite o uso de instruções de que envolvem E/S e proteção de memória são inacessíveis
 - ◆ Programas de usuário utilizam este modo
- Chamadas ao sistema que incluem a instrução TRAP alterna entre os modos

Memória

- **Memória Ideal**
 - ◆ Rápida (mais veloz que a execução de uma instrução pela CPU)
 - ◆ Grande
 - ◆ Baixo custo
- **Existem vários níveis/tipos de memória**
 - ◆ Registradores
 - ◆ Cache
 - ◆ RAM
 - ◆ Discos

Registradores

- Internos a CPU e constituídos do mesmo material
- Não impõe atrasos a CPU
- Capacidades típicas
 - ◆ 32x32 bits (máquinas 32 bits)
 - ◆ 64x64 bits (máquinas 64 bits)

Memória *cache*

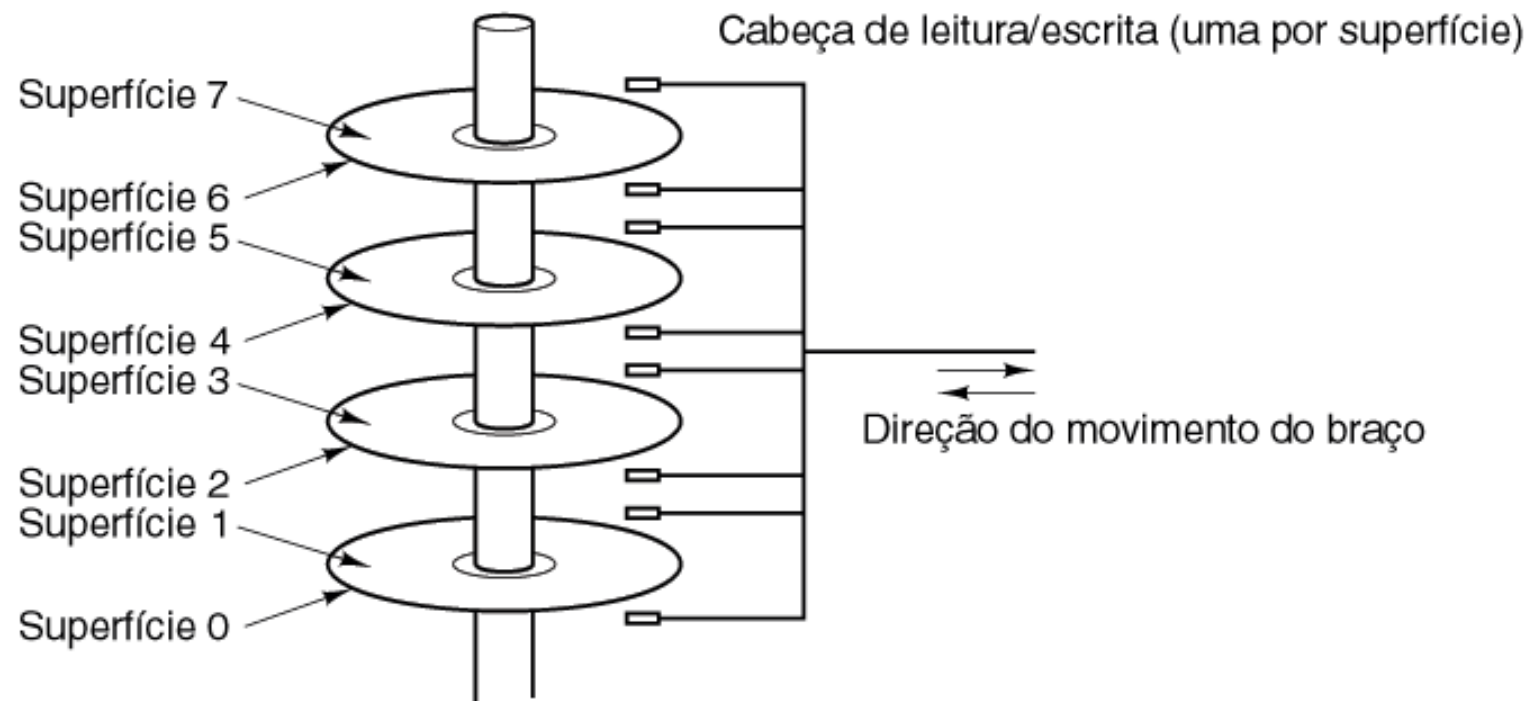
- ▣ Divide-se a memória principal em linhas
- ▣ As posições de memória mais utilizadas são copiadas para a cache
- ▣ Quando um programa faz uma busca na memória o hardware que gerencia a MC, procura pelo endereço na cache
- ▣ Caso esteja responde a requisição
- ▣ Caso contrário faz um busca na memória principal
- ▣ O tempo de resposta da cache equivale a 2 ciclos de CPU

Memória Principal

- ▣ RAM (*Random Access Memory*)
- ▣ Ordem de Gigabytes
- ▣ Possui unidade de gerenciamento própria (MMU)

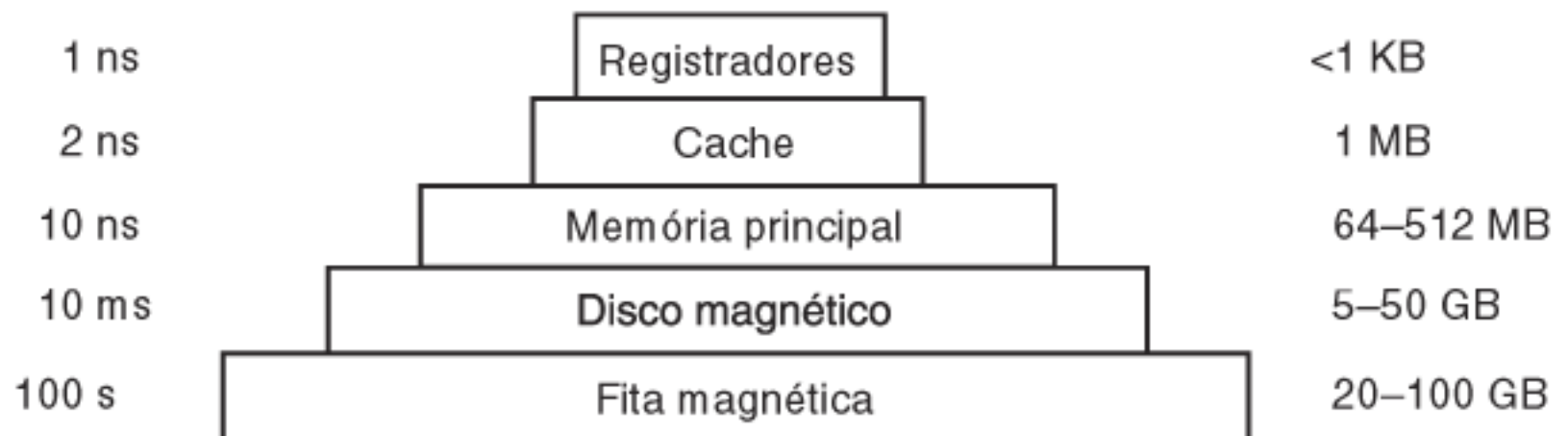
Discos Magnéticos

- ❑ Memória permanente
- ❑ Capacidade de armazenamento superior
- ❑ Tempo de acesso aleatório (dezenas de ms)
- ❑ Velocidades 5400, 7200 e 10800 rpm



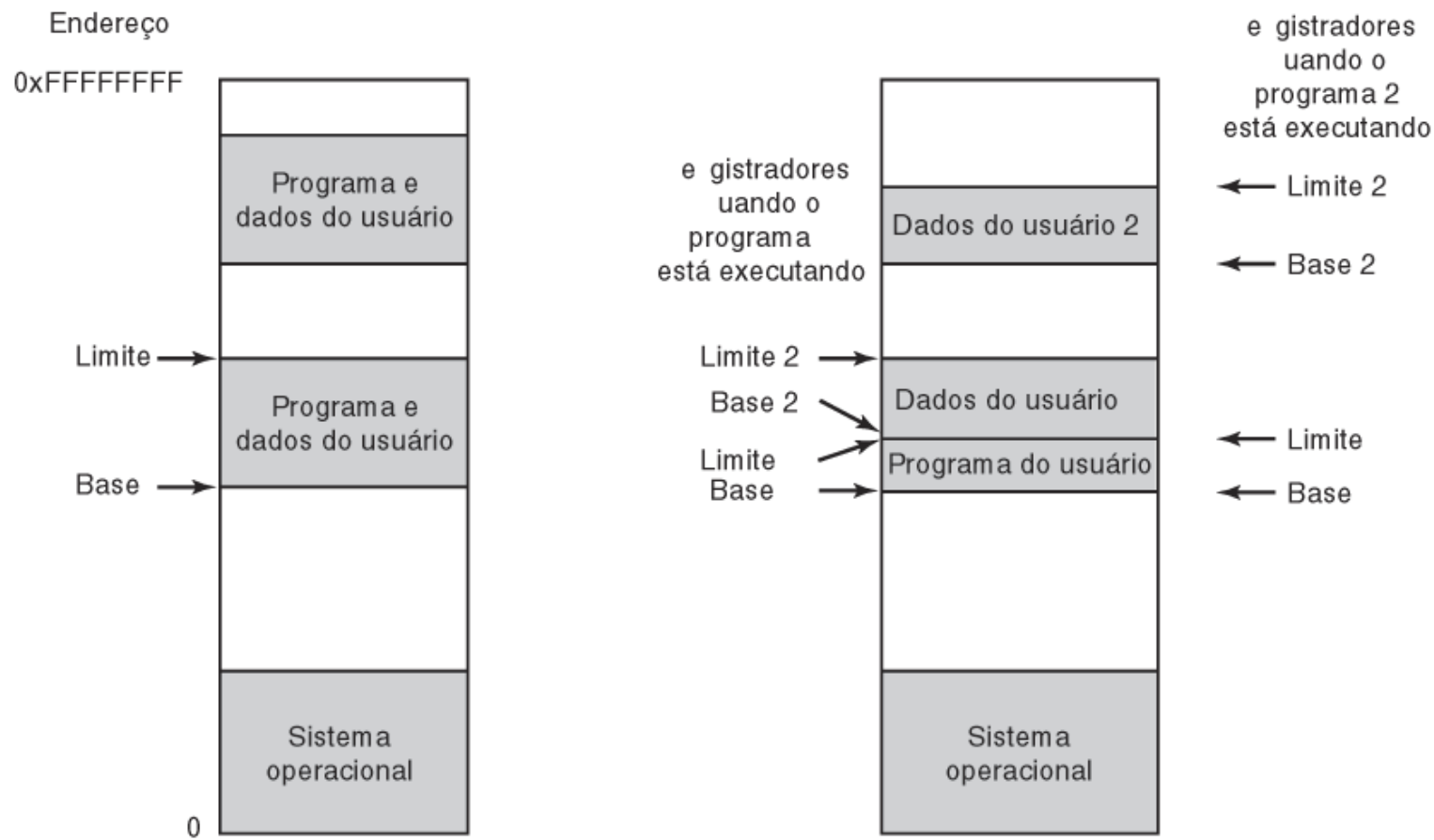
Tempo de acesso típico

Capacidade típica



Memory Management Unit (MMU)

- Proteger os programas de outros programas e o SO de todos os programas
- Realocar a memória
- Registrador-base e Registrador-limite



Dispositivos de E/S

- Duas partes:
 - ◆ Controlador
 - ◆ Próprio dispositivo de E/S
- *Driver* de Dispositivo é o software necessário para que o SO possa controlar o dispositivo
 - ◆ São específicos para cada dispositivo e SO

Três formas de E/S

- ▣ Espera ociosa (*busy waiting*)
- ▣ Interrupções
- ▣ DMA (*Direct Memory Access*)

Espera Ociosa (*Busy Waiting*)

- ▣ Programa emite chamada ao sistema para acessar determinado dispositivo
- ▣ O SO, por sua vez, faz chamada ao respectivo *driver* de dispositivo
- ▣ O *driver* inicia um *loop* perguntando continuamente ao dispositivo se a operação de E/S foi concluída
- ▣ Quando isto acontece o *driver* escreve os dados em memória e retorna ao SO
- ▣ O controle é devolvido ao programa

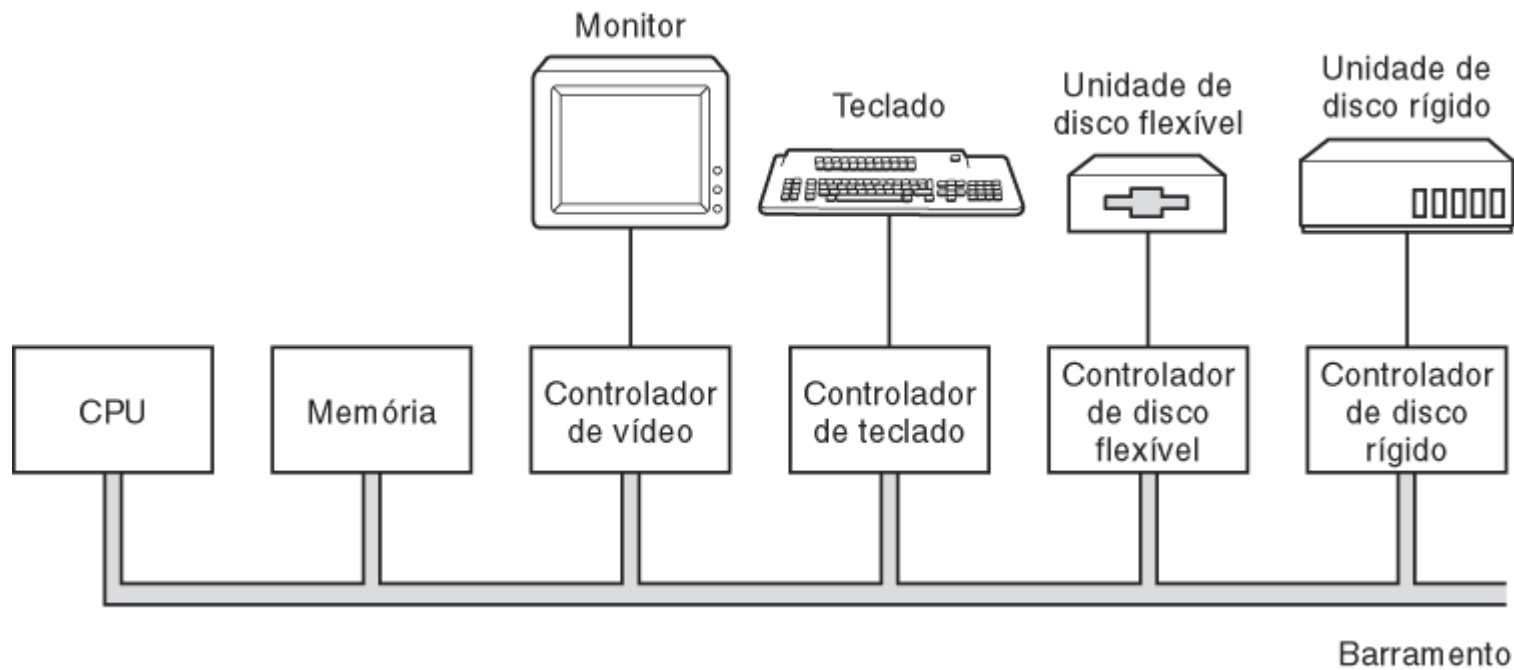
Interrupção

- ▣ Programa necessita de operação de E/S
- ▣ Utilizando interrupção o driver inicia o dispositivo e devolve o controle da CPU ao SO
- ▣ O SO bloqueia o programa enquanto a operação de E/S é concluída
- ▣ SO busca outras tarefas para executar
- ▣ Quando a E/S é concluída o dispositivo gera uma interrupção e o SO volta a dar atenção ao dispositivo
- ▣ SO devolve o controle ao programa

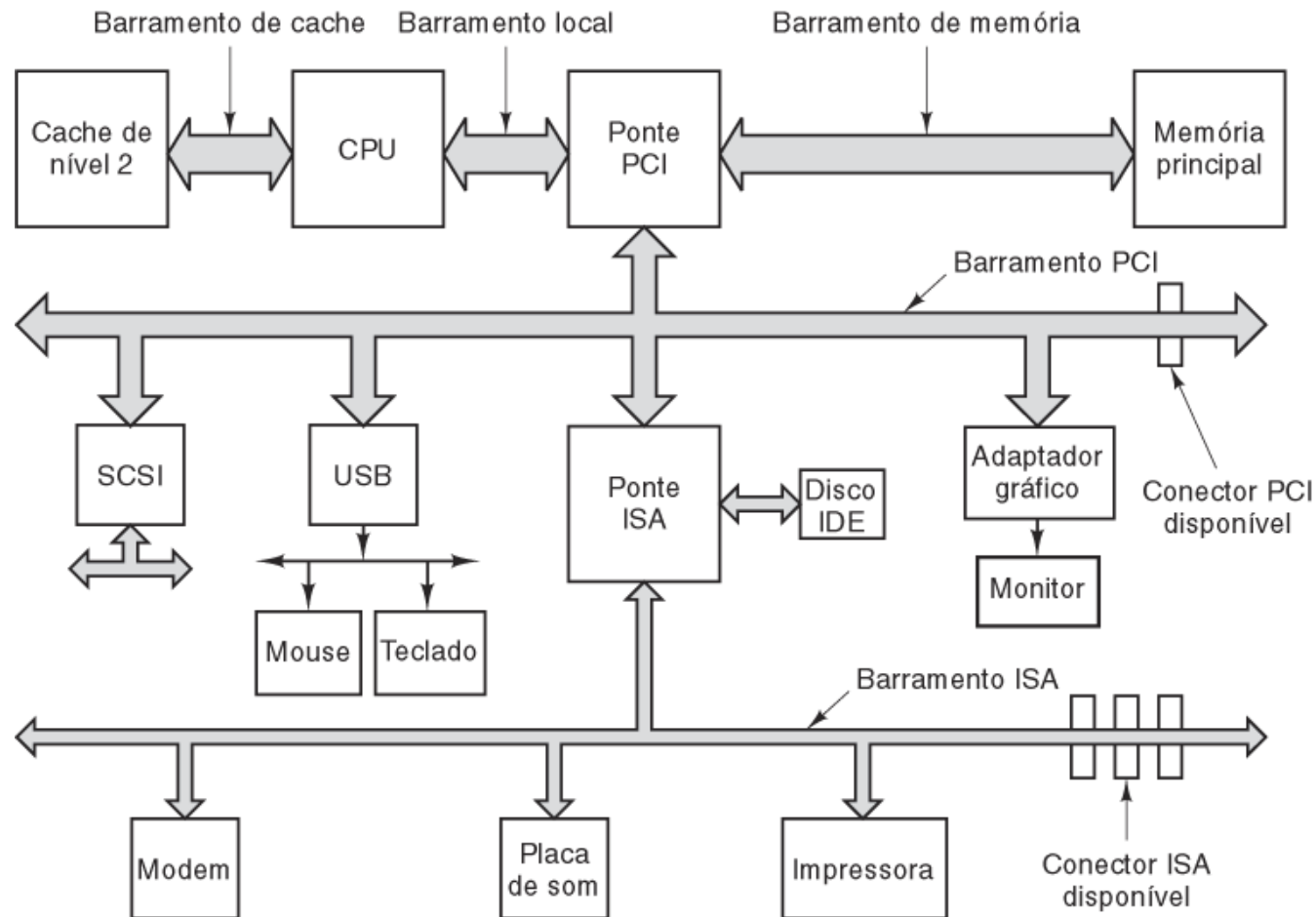
DMA

- ▣ *Hardware* especial para tratar as interrupções
- ▣ Os dispositivos escrevem diretamente na memória sem a intervenção da CPU
- ▣ Evita que o SO tenha de copiar os dados do *buffer* do dispositivo para a memória do sistema

Barramentos idealizados



Barramentos atuais



Dúvidas e Perguntas

Referências

- Tanenbaum, A. S. **Sistemas Operacionais Modernos**. 4a. Edição. Editora Pearson. (Capítulo 1)

OBSERVAÇÃO

A disponibilização das notas de aula através de slides serve apenas como apoio aos estudos. Para um bom aproveitamento e aprendizado é necessário a leitura das referências (livro texto) e estar atento às aulas

Leitura sugerida

- ▣ **Pipeline x Superscalar** (em pdf).
 - ◆ Prof. Clodoaldo Lima (Unicamp)
 - ◆ Prof. Fernando Von Zuben (Unicamp)

FIM