



Universidade Federal do Ceará
Campus Quixadá

Sistemas Operacionais

AULA 1: INTRODUÇÃO AOS SISTEMAS OPERACIONAIS
SLIDES ADAPTADOS DE LEONARA BRAZ

PROF SIDARTHA CARVALHO

Seção 1

INTRODUÇÃO AOS SISTEMAS OPERACIONAIS

Estrutura de um sistema computacional

Usuários

Aplicativos (programas)

Sistema Operacional

Hardware

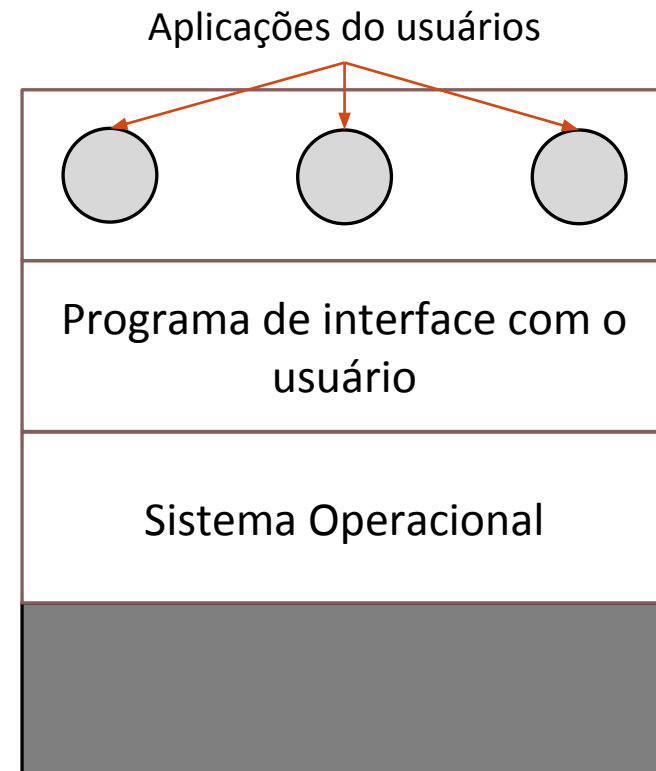


Figura: Onde o SO se encaixa

Estrutura de um sistema computacional

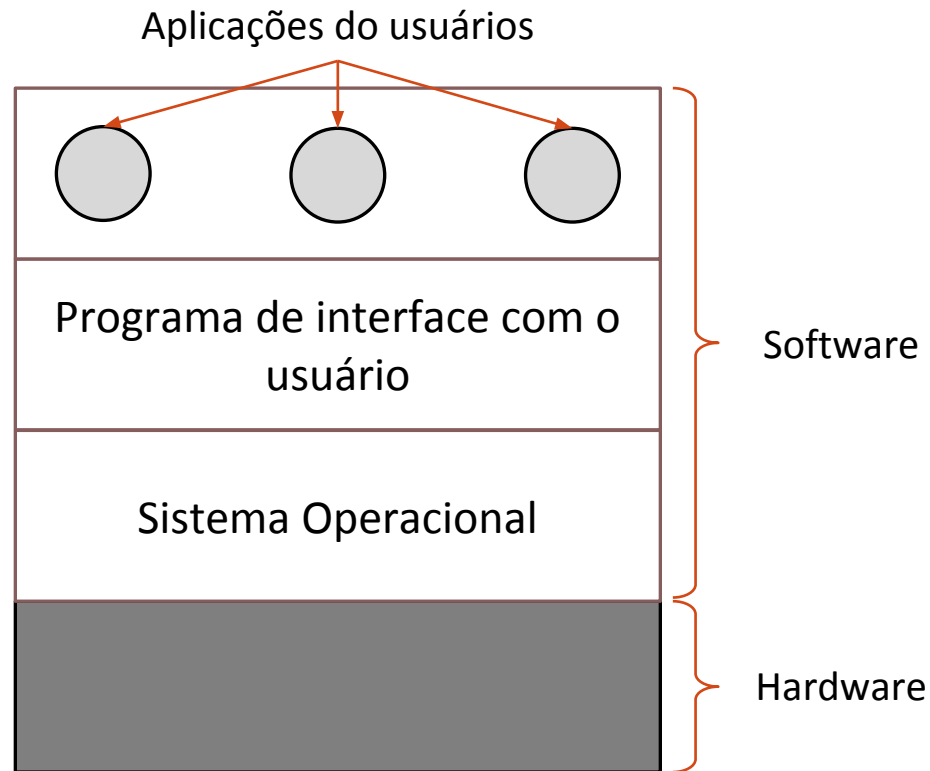


Figura: Onde o SO se encaixa

Hardware

- Chips, placas, discos, teclado – Objetos físicos

Software

- O programa de interface com o usuário é o nível mais inferior
- Permite ao usuário iniciar programas como navegador web, leitor de e-mails, etc

Sistema Operacional

- Opera diretamente no hardware e fornece a base para os outros programas

Programas e Aplicativos

Os programas exigem diferentes recursos do Hardware

Os programadores nem sempre são cuidadosos:

- Especialmente em relação ao fechamento e liberação de recursos, tais como (regiões de memória, sockets, arquivos, etc).
- **Isso pode prejudicar o funcionamento geral do sistema**

Hardware

Elevada pluralidade no hardware de um computador:

- Um ou mais processadores, memória principal, discos, impressoras, vários dispositivos de entrada e saída, etc.
- Cada componente tem diferentes fabricantes

Como criar um programa que fosse capaz de conversar com todos esses tipos de componentes e fabricantes?

Sistema Operacional

“Computadores têm um componente de software conhecido como **Sistema Operacional**, cujo trabalho é **fornecer** aos programas do usuário **um modelo de computador** melhor, mais simples e mais limpo de **lidar com o gerenciamento de todos os recursos de hardware** que formam um sistema computacional.”

TANEMBAUM (2010)

Funções básicas do Sistema Operacional

1. Fornecer aos programadores de aplicativos (e aos programas) um conjunto de recursos abstratos claros

2. Gerenciar os recursos do Hardware

Objetivos do Sistema Operacional

1. Executar programas do usuário de forma a ajudá-lo a resolver seus problemas de forma mais simples
2. Facilitar o uso de um sistema computacional
3. Utilizar o hardware disponível de forma eficiente

SO como máquina estendida

A **arquitetura** da maioria dos computadores em nível de linguagem de máquina é primitiva e de difícil programação

- Especialmente a entrada/saída

Exemplo: Chip controlador NEC-PD765

NEC-PD765

Chip controlador de disquetes em computadores pessoais baseados em processadores Intel

Possui 16 comandos para leitura e escrita de dados, para movimentação do braço e para formatação das trilhas

Os comandos mais básicos são de leitura e escrita:

- Cada um requer 13 parâmetros agrupados em 9 bytes
- Os parâmetros especificam endereço de bloco de dados, número de setores, modo de gravação

Ao completar a operação, o chip controlador retorna 23 campos de status e erros agrupados em 7 bytes

Abstração

O programador busca lidar com essas unidades de um modo mais abstrato e simples

Como podemos abstrair o procedimento das unidades de disco?

Abstração

Uma abstração típica seria aquela compreendida por um disco que contém uma **coleção de arquivos** com nomes. Cada arquivo pode ser aberto para leitura ou escrita e, então, **ser lido ou escrito** e, por fim, fechado.

Detalhes sobre o processo **não apareceriam** na abstração apresentada ao programador da aplicação.

Abstração

Abstração é o elemento-chave para gerenciar complexidade

- Boas abstrações transformam uma tarefa quase impossível em duas manejáveis
 - Definição e implementação da abstração
 - Uso da abstração para resolução de problema

A tarefa principal do SO é criar boas abstrações!

Exemplo de abstração: Arquivo

- Fragmento de informação útil

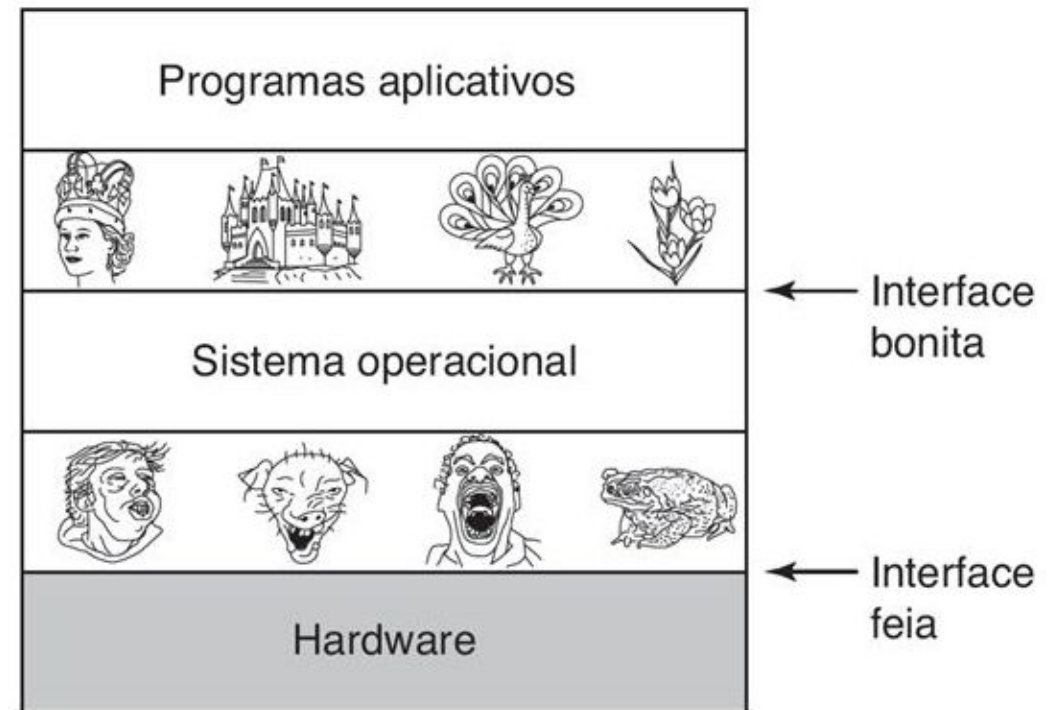
Abstração

O hardware é feio

Uma das principais tarefas do sistema operacional é ocultar o hardware

- Oferecendo abstrações precisas, claras e coerentes

O SO transforma o feio em bonito



IMPORTANT E

Os clientes reais do Sistema Operacional são os programas

Os usuários finais lidam com abstrações fornecidas pela interface do usuário



```
Administrador: C:\Windows\System32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>systeminfo

Nome do host: PC
Nome do sistema operacional: Microsoft Windows 7 Ultimate
Versão do sistema operacional: 6.1.7601 Service Pack 1 compilação 7601
Fabricante do sistema operacional: Microsoft Corporation
Configuração do SO: Estação de trabalho autônoma
Tipo de compilação do sistema operacional: Multiprocessor Free
Proprietário registrado: edivaldo
Organização registrada:
Identificação do produto: 00406 0FM 0000600 000001
Data da instalação original: 03/02/2013, 10:23:50
Tempo de Inicialização do Sistema: 25/11/2013, 08:31:25
Fabricante do sistema: Positivo Informatica SA
Modelo do sistema: POS-ECIG41BS
Tipo de sistema: x64-based PC
Processador(es): 1 processador(es) instalado(s).
[01]: Intel64 Family 6 Model 23 Stepp
ing 10 GenuineIntel ~2793 Mhz
Versão do BIOS:
a SA 080015 , 14/05/2010
Desenvolvida para Positivo Informatic
```

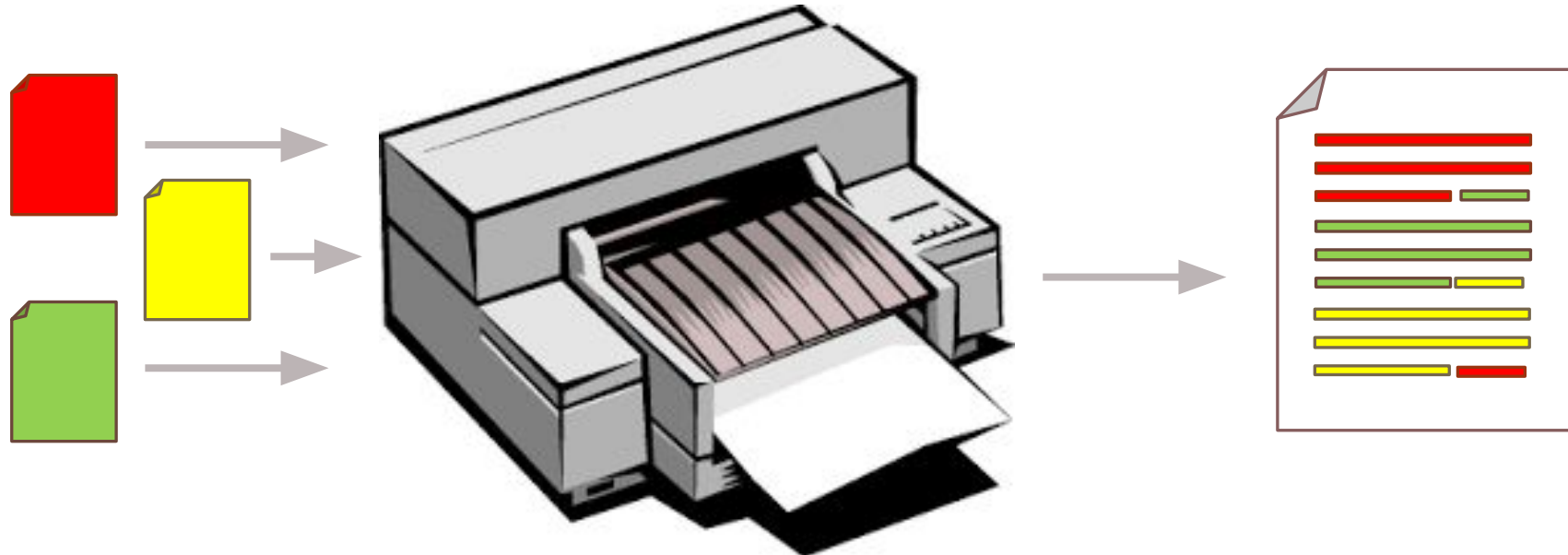

SO como gerenciador de recurso

Um sistema operacional **gerencia** todas **as partes** de um sistema complexo.

O trabalho de um sistema operacional é **fornecer** uma **alocação ordenada e controlada** de processadores, memórias e dispositivos de E/S **entre** vários **programas** que competem por eles

SO como gerenciador de recurso

Imagine o que aconteceria se três programas em execução tentassem imprimir suas saídas simultaneamente na mesma impressora



SO como gerenciador de recurso

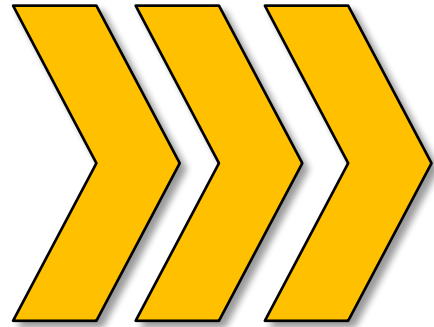
Algumas vezes, se faz necessário que usuários e programas compartilhem hardware e informações

- O SO também deve estar preparado para tal ocasião

O compartilhamento de recursos pode ser realizado de duas maneiras: **tempo** e **espaço**

- **No tempo:** Diferentes programas esperam sua vez para utilizar o recurso
- **No espaço:** Em lugar de esperar sua vez, cada programa ocupa uma parte do recurso compartilhado

Tipos de compartilhamento

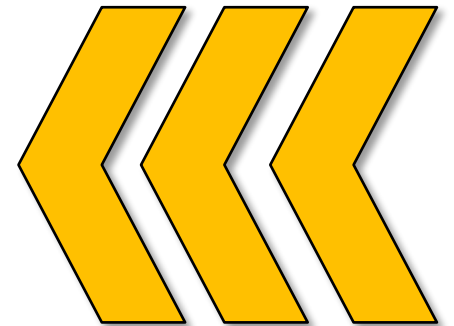


COMPARTILHAMENTO NO TEMPO

O processador somente recebe um programa por vez para ser executado, os demais esperam

COMPARTILHAMENTO NO ESPAÇO

A memória principal é normalmente segmentada entre vários programas em execução



Funções do SO no compartilhamento

Determinar como o recurso é compartilhado no tempo:

- Qual processo vai utilizar o recurso e por quanto tempo será

Determinar como o recurso é compartilhado no espaço:

- Tamanho do espaço para cada processo; justiça; proteção; etc

Tudo é feito através de componentes do SO:

- **Escalonador de processos** para as questões temporais
- **Gerenciador de memória** para as questões espaciais

Escalonador de Processos

Quando um computador é multiprogramado, ele muitas vezes possui diversos processos que competem pela CPU, surgindo assim a necessidade de alguma entidade para escalonar a CPU entre os processos

- Essa entidade é conhecida como escalonador
- O algoritmo que é usado é chamado de algoritmo de escalonamento.

O **objetivo do escalonador** é atribuir processos para serem executados pelos processadores de modo a atingir parâmetros de performance, tais como *tempo de resposta, vazão e eficiência do processador*.

Escalonador de Processos

Um escalonador deve possuir um algoritmo de escalonamento que se preocupe com 5 regras:

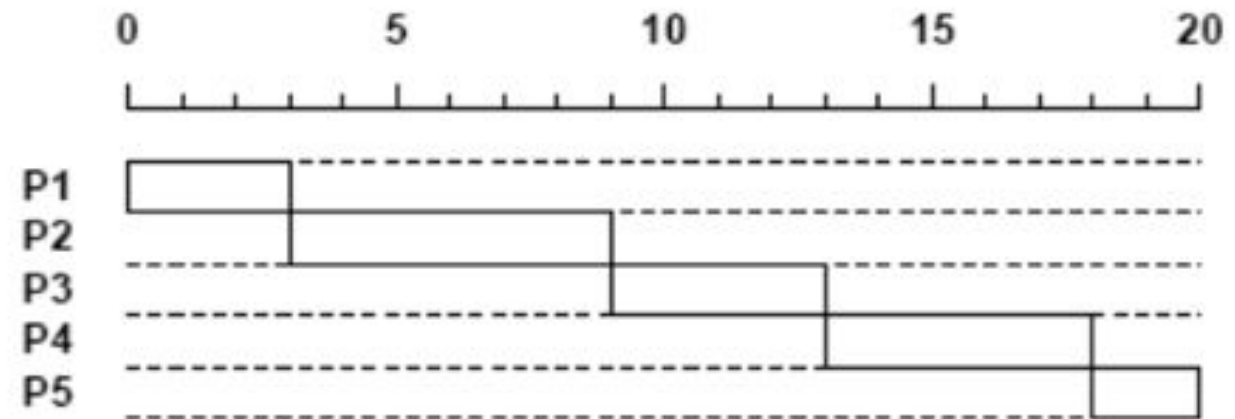
- 1. Justiça** – Todos processos devem ter acesso a CPU
- 2. Eficiência** – A meta do escalonador é chegar mais perto dos 100% de utilização da CPU
- 3. Minimizar o Tempo de Resposta**
- 4. Turnaround** – Minimiza os usuários batch
- 5. Throughput** – Maximizar o número de jobs processados

Algoritmos de Escalonamento

Primeiro a chegar, primeiro a sair:

- A CPU é atribuída aos processos na ordem em que eles a requisitam. Basicamente há uma fila única de processos prontos. Exemplo

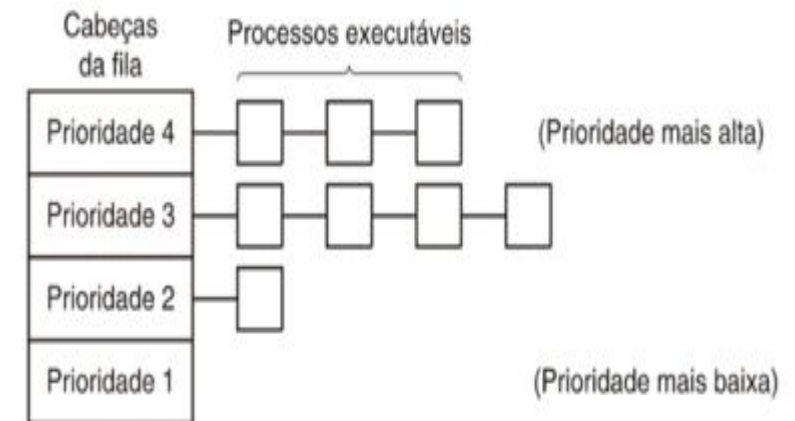
Processo	Tempo de chegada	Tempo de serviço
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2



Algoritmos de Escalonamento

Escalonamento por prioridade:

- A cada processo é atribuído uma prioridade, o processo com maior prioridade é executado primeiro.
- Para evitar que processos de alta prioridade executem infinitamente, o escalonador pode reduzir a prioridade do processo a cada execução do mesmo.
- As prioridades são atribuídas pelo S.O.



Multiprogramação

Um único programa não pode, em geral, manter a CPU ou os dispositivos de E/S ocupados por muito tempo – Justiça

- Isso impossibilitaria a execução simultânea de programas

O SO mantém várias tarefas na memória simultaneamente

O SO escolhe uma das tarefas na memória para executar na CPU e pode trocar a tarefa atual por outra da memória

Exercício 1

Com base no conhecimento adquirido na aula de hoje, elabore a sua própria definição de Sistema Operacional.

Não esqueça de, na sua definição, incluir as abordagens top-down (sentido usuários/hardware) e bottom-up (sentido hardware/usuários).

Exercício 2

Em casa, leia as seções **1.3 Revisão sobre hardware de Computadores** e **1.5 Conceitos sobre Sistemas Operacionais** de (TANEMBAUM, 2010/2016).

Seção 2

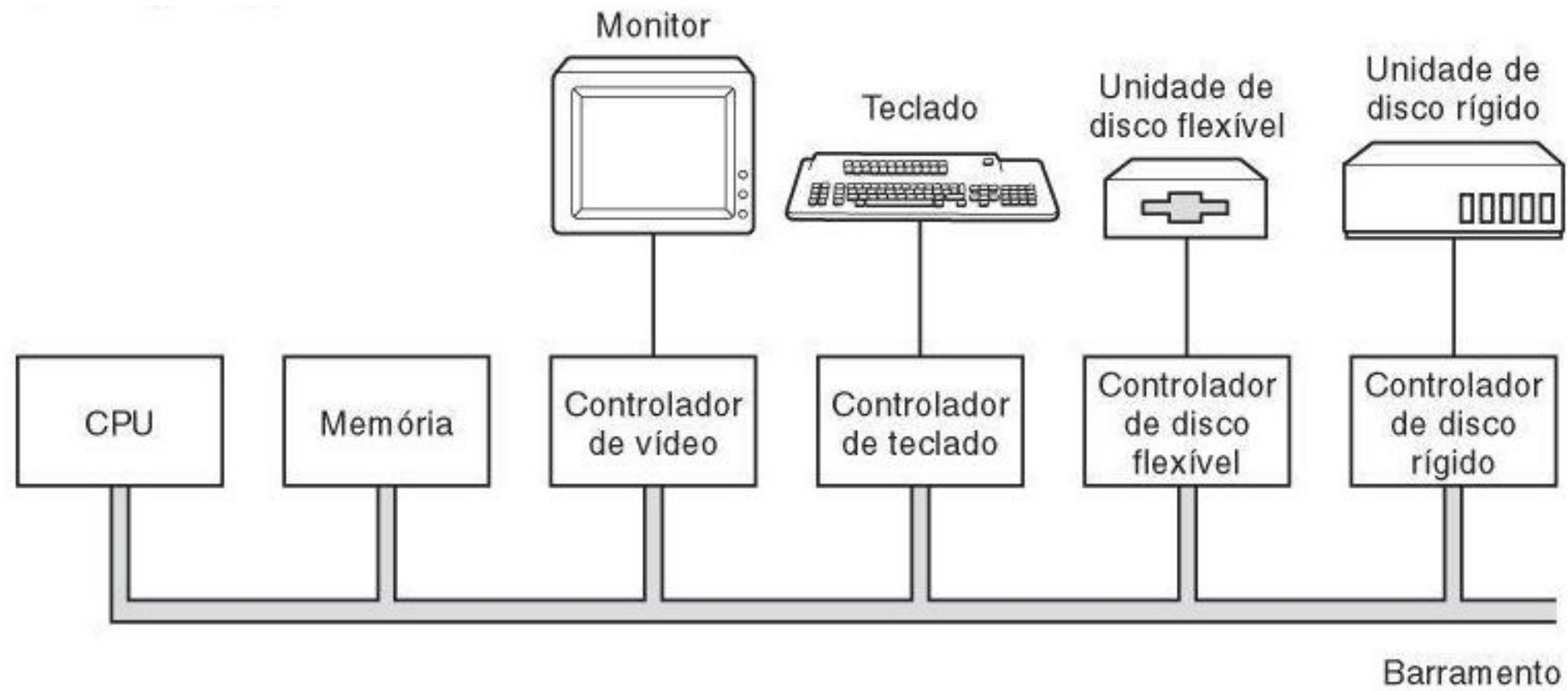
REVISÃO DA ARQUITETURA DE COMPUTADORES

Organização geral

Existem quatro componentes estruturais principais:

- **Unidade Central de Processamento (CPU)**
 - Controla as operações e o processamento dos dados
- **Memória Principal**
- **Entrada e Saída (E/S)**
 - Fornece uma interface entre o computador e o mundo externo
- **Interconexão do Sistema (Barramento)**
 - Mecanismo que oferece comunicação entre os componentes estruturais

Organização geral



CPU – Unidade Central de Processamento

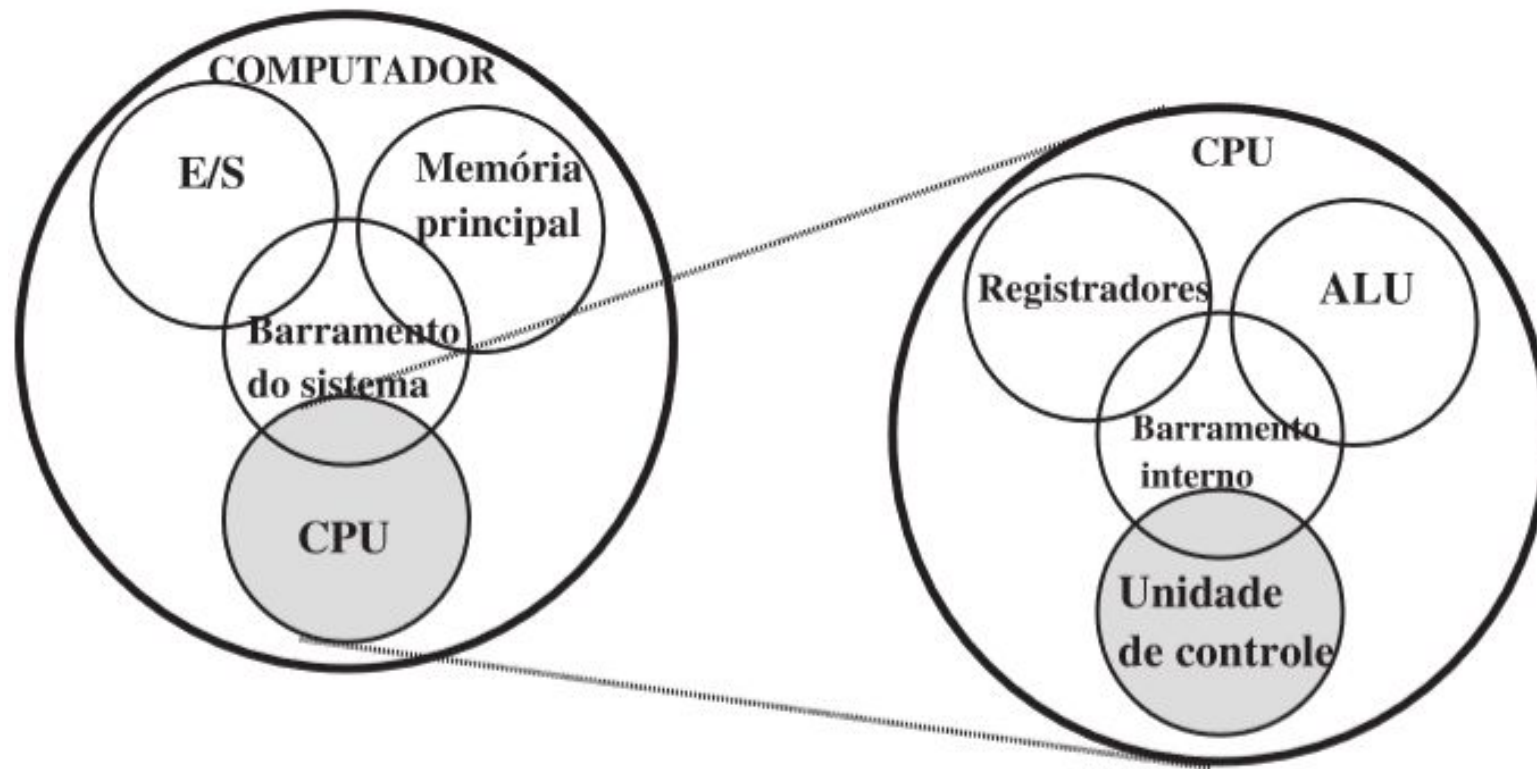
Computar as operações lógicas e aritméticas

Gerenciar o funcionamento de todo o computador

Composto por quatro componentes principais:

- Unidade Lógica e Aritmética (ULA)
- Unidade de Controle
- Registradores

CPU – Unidade Central de Processamento



CPU – Unidade Central de Processamento

Os diversos chips em uma placa mãe precisam de uma sincronização única para manter o **fluxo de dados coerente**

Para isso um circuito de relógio (**clock**) distribui um sinal único para todos os chips que passarão a se comunicar de forma coerente. Sua velocidade é medida em unidades de ciclos por segundo (*hertz, megahertz e gigahertz*)

Quanto mais rápidos os impulso de relógio, mais rapidamente o computador funciona. Porém, o pulso de disparo não pode funcionar mais rapidamente do que a capacidade dos chips

CPU – Unidade Central de Processamento

Um outro fator importante para a velocidade do microprocessador é a quantidade de bits de entrada/saída que ele pode processar simultaneamente em um determinado instante (8 bits) – atualmente é 64 bits

Uso de uma memória **cache** para permitir ao processador guardar instruções/dados recentemente usados

Habilidade de alguns processadores centrais de trabalhar ao mesmo tempo com instruções múltiplas em paralelo

ULA – UNIDADE LÓGICA E ARITMÉTICA

A ULA é uma peça fundamental da unidade central de processamento (CPU)

O matemático **John von Neumann** propôs o conceito de ULA em 1945

A ULA executa as principais operações lógicas e aritméticas do computador.

- Além de executar funções aritméticas, uma ULA deve ser capaz de determinar se uma quantidade é menor ou maior que outra e quando quantidades são iguais

ULA – UNIDADE LÓGICA E ARITMÉTICA

Operações da ULA:

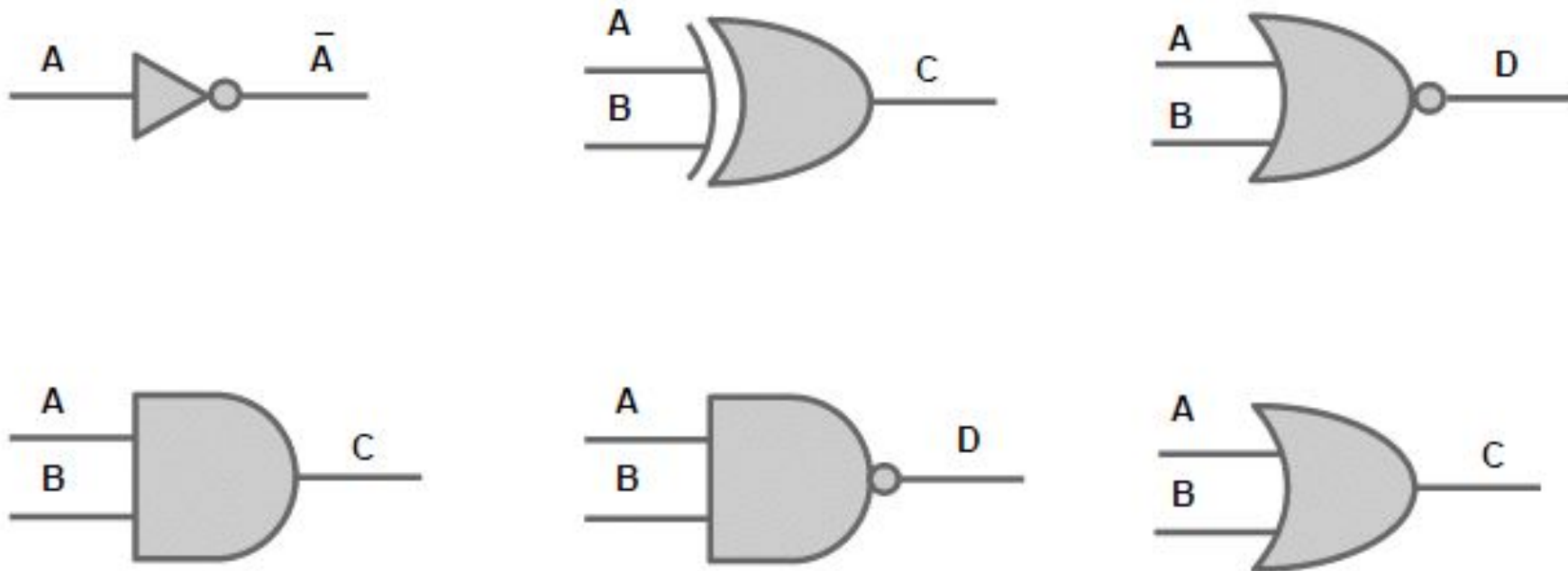
- Operações aritméticas (adição e subtração)
- Operações lógicas em bit: AND, OR, XOR e NOT
- Operações de deslocamento de bit

Comumente, as operações ocorrem sobre os registradores

Podem ser implementadas funções mais complexas:

- Contudo aumenta a complexidade dos circuitos lógicos

ULA – UNIDADE LÓGICA E ARITMÉTICA



ULA – UNIDADE LÓGICA E ARITMÉTICA

Para realizar corretamente as operações aritméticas, uma ULA deve operar números com a mesma **base** e o mesmo **formato**

- Os primeiros computadores usavam diversos sistemas numéricos, bases numéricas, representações de sinais, complemento de um, complemento de dois, sinal-magnitude
- Para cada um desses conjuntos de parâmetros é necessário realizar um projeto de ULA diferente, além da necessidade de realizar a conversão de todos os dados quando se trocava de computador
- Nos computadores modernos (últimos 30 anos), quase sempre se utiliza número binário e caracteres ASCII

ULA – UNIDADE LÓGICA E ARITMÉTICA

Outro fator importante relacionado à representação de números em um computador é o **tamanho** e a **precisão** que um processador pode tratar

Além da ULA, existe a **Unidade de Ponto Flutuante** que realiza operações aritméticas com números em representação de ponto flutuante

- Realiza operações aritméticas entre dois números em representação de ponto flutuante

Registradores

Os registradores estão no topo da hierarquia de memória

- Sendo assim, são o meio mais rápido e caro de se armazenar um dado

São circuitos digitais capazes de armazenar e deslocar informações binárias, e são tipicamente usados como um dispositivo de **armazenamento temporário**

Apesar dos registradores apresentarem um bom desempenho no quesito rapidez, possuem uma **capacidade de armazenamento baixíssima** comparada a outros circuitos de armazenamento

Registradores

São utilizados na execução de **programas de computadores** para disponibilizar um local para armazenar dados temporários

Pelo fato de estarem **próximos da ULA** e de ser construída com circuitos de memória muito rápidos, possibilita o **maior desempenho no processamento**

Registradores de Dados (utilizados para armazenar valores numéricos ou caracteres) e **Registradores de Endereço** (recebem o endereço de um objeto)

Unidade de Controle

Busca, Decodifica e Executa a instrução

Pode ser classificada como:

- **Fixa:** Instruções definidas pelo fabricantes
- **Programável:** Instruções definidas pelo desenvolvedor

Gera sinais de controle da UCP

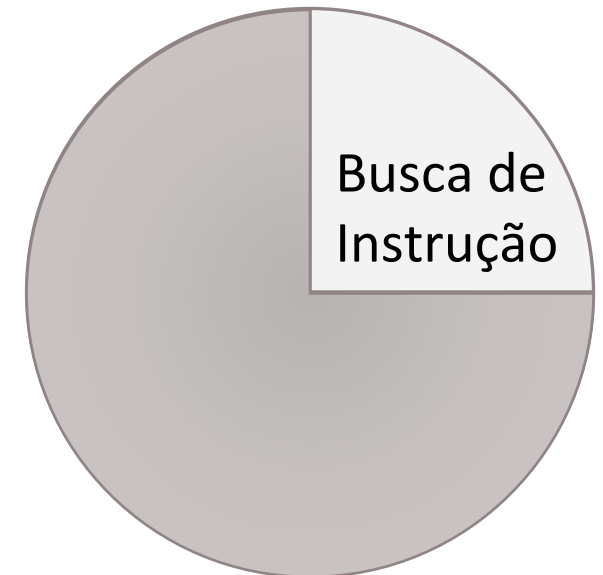
- Controle interno e a interação com o mundo externo

Unidade de Controle

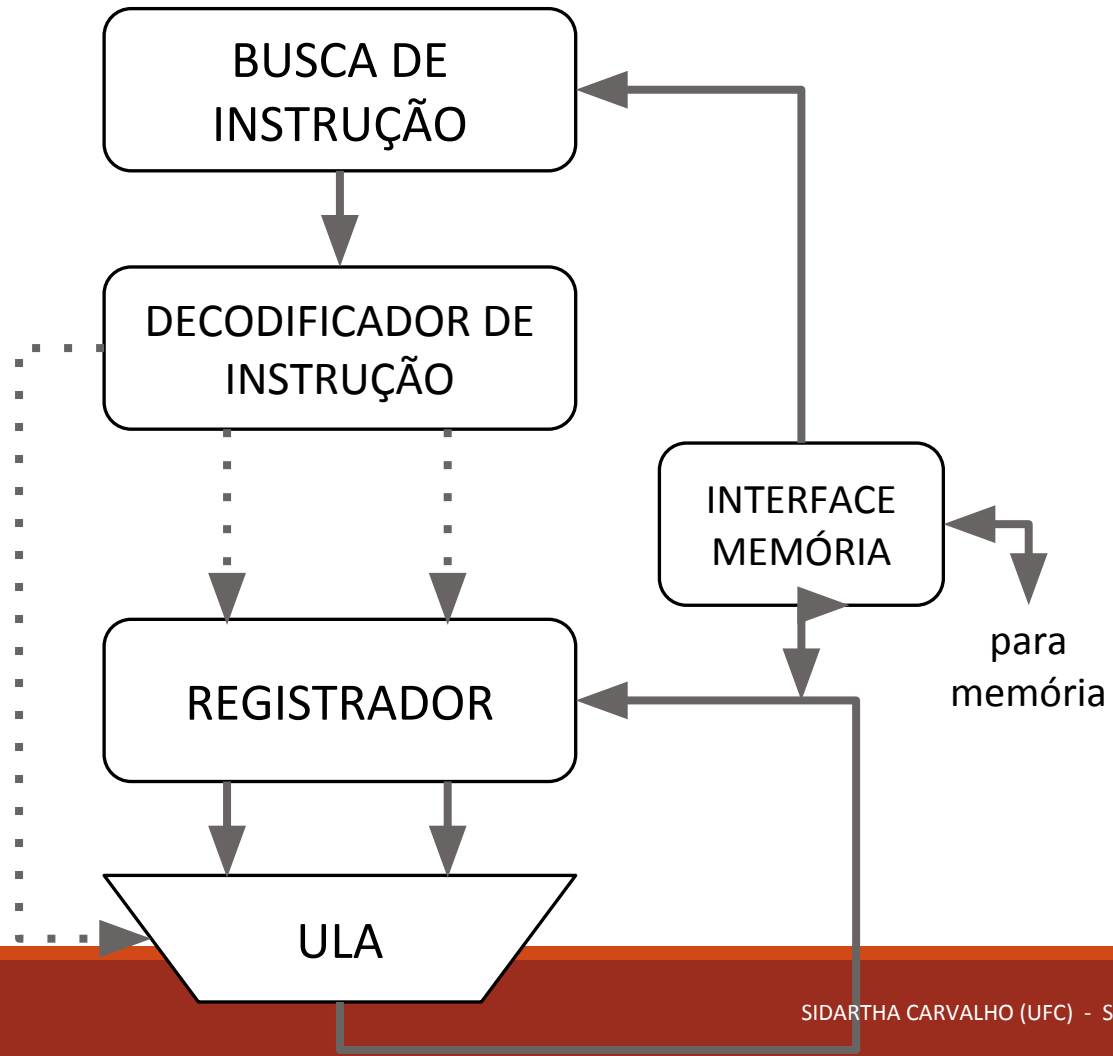
A função fundamental da CPU é executar uma sequência de instruções armazenadas chamada Programa

O Programa é representado por uma série de números que são mantidos em algum tipo de memória no computador

A Unidade de Controle, através do Ciclo de Instrução, determina como uma instrução vai ser executada



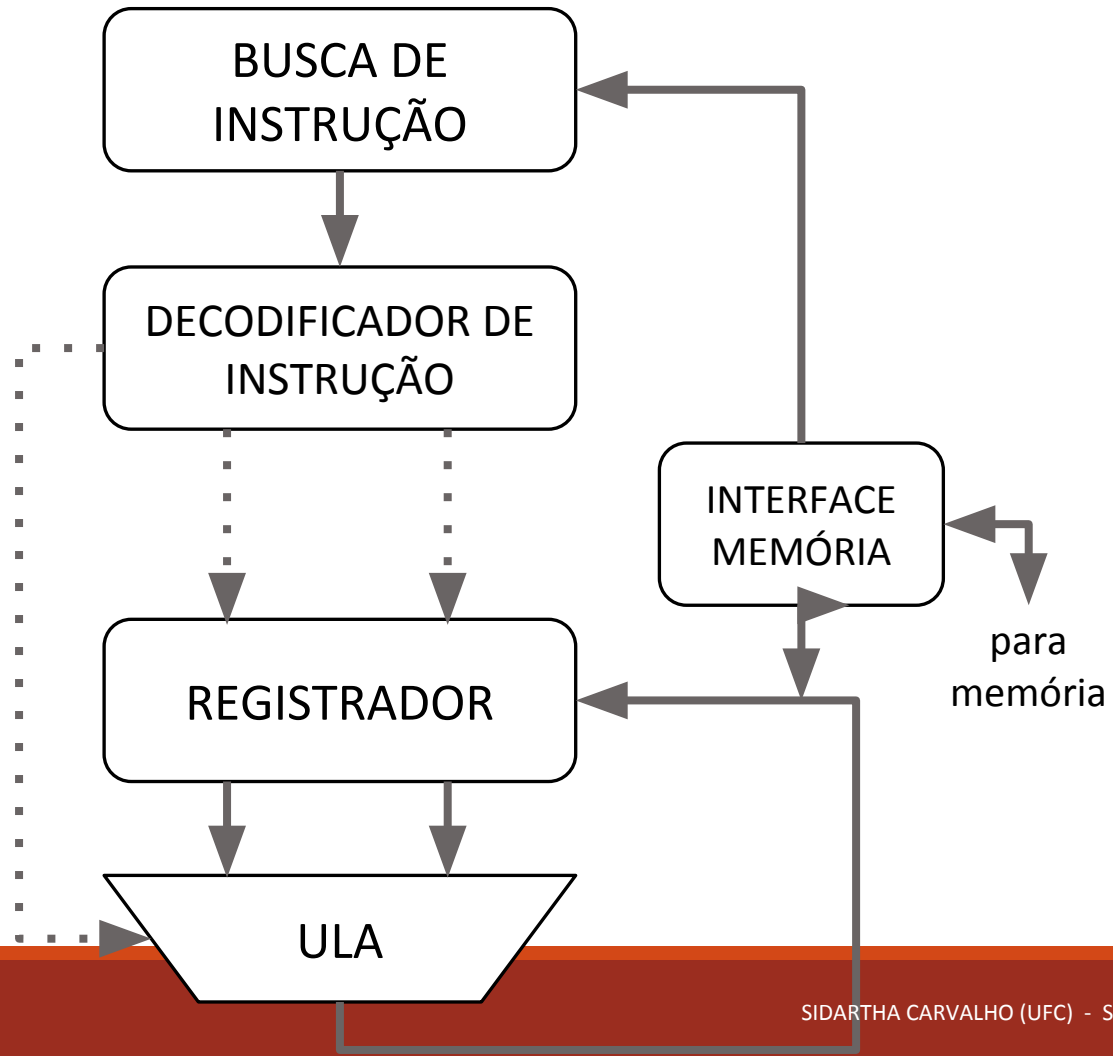
Unidade de Controle



Envolve **recuperar** uma instrução (representada por um número ou por uma sequência de números) na memória de programa

A posição da instrução em execução na memória de programa é determinada por um **Contador de Programa**, o qual também guarda o endereço da instrução em questão

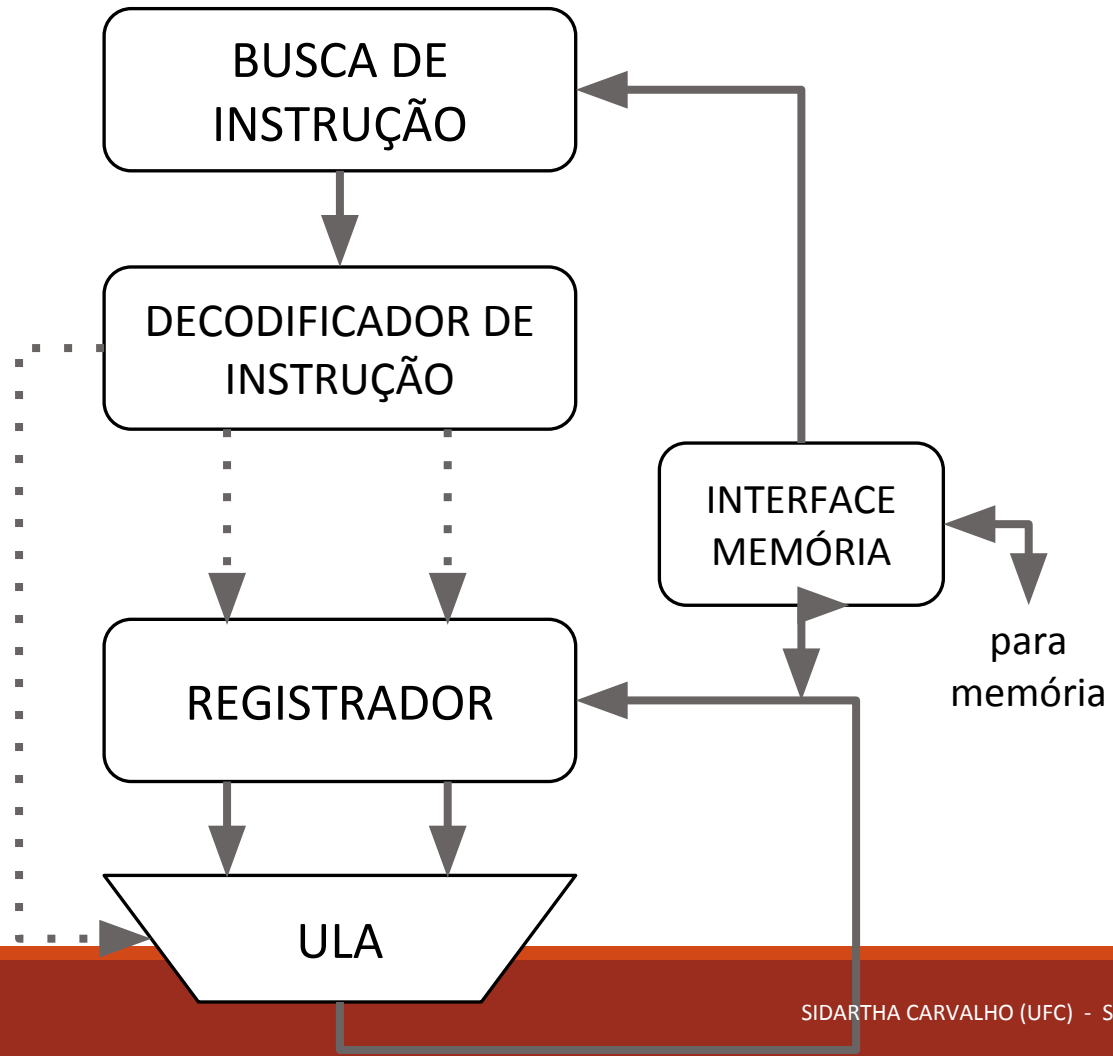
Unidade de Controle



A instrução é **quebrada** em partes, e cada uma tem um significado para as outras partes da CPU

A maneira como um valor numérico de instrução é interpretado é definida pelo **Conjunto de Instruções**

Unidade de Controle



Durante a etapa de **Execução da Instrução**, as várias partes do processador central são conectadas e podem executar a operação desejada

Se uma operação de adição foi solicitada, uma Unidade de Lógica e Aritmética (ULA) será conectada a um registrador de entradas e de saídas

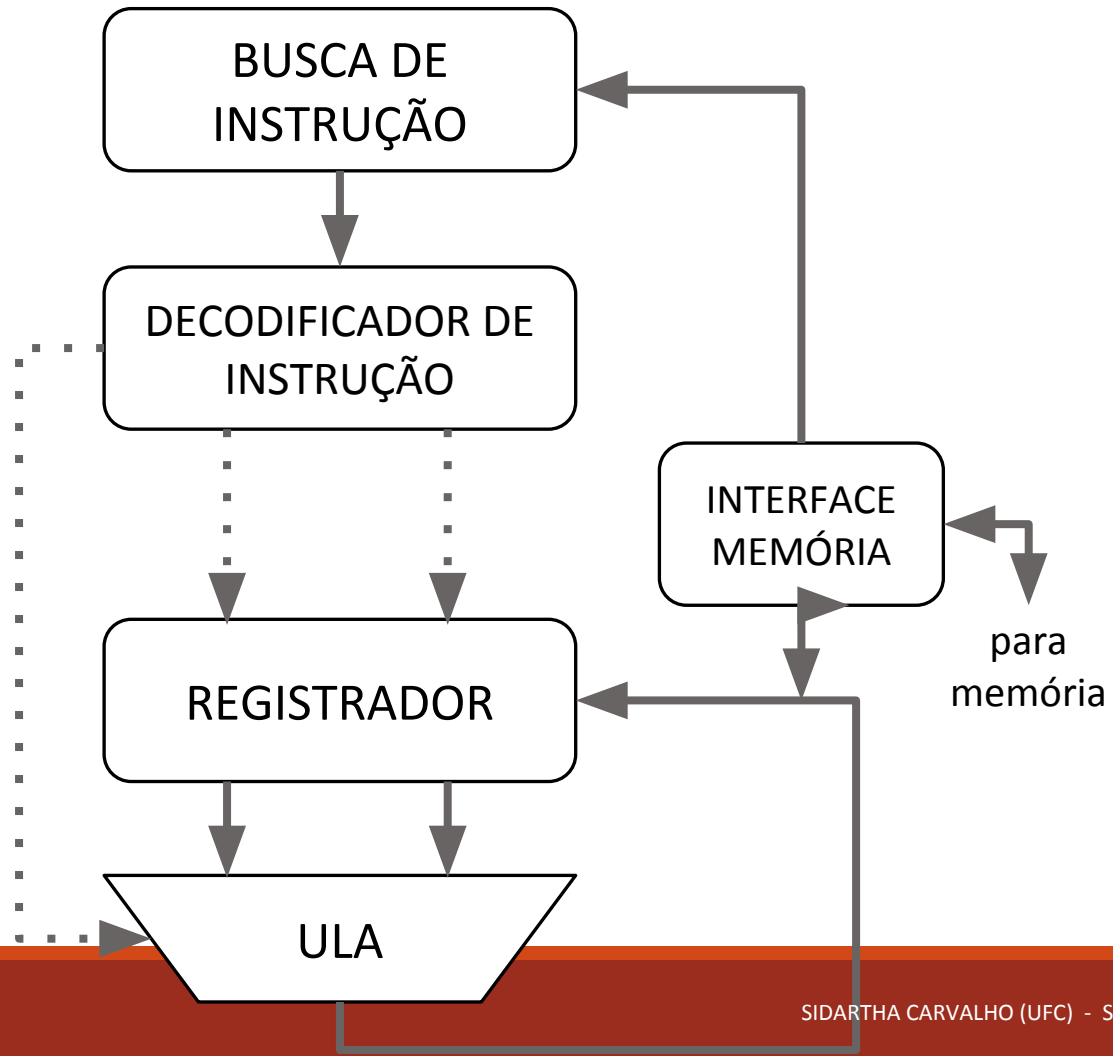
Unidade de Controle



Finalmente na **Escrita de Dados**, escreve-se os resultados da etapa da execução em algum local da memória

Os resultados podem ser escritos em algum **registrador** interno do processador ou em uma **memória externa**

Unidade de Controle



Após a escrita dos dados resultantes, haverá a repetição de todo o processo, com o ciclo de busca da instrução seguinte devido ao incremento do valor do **Contador de Programa**

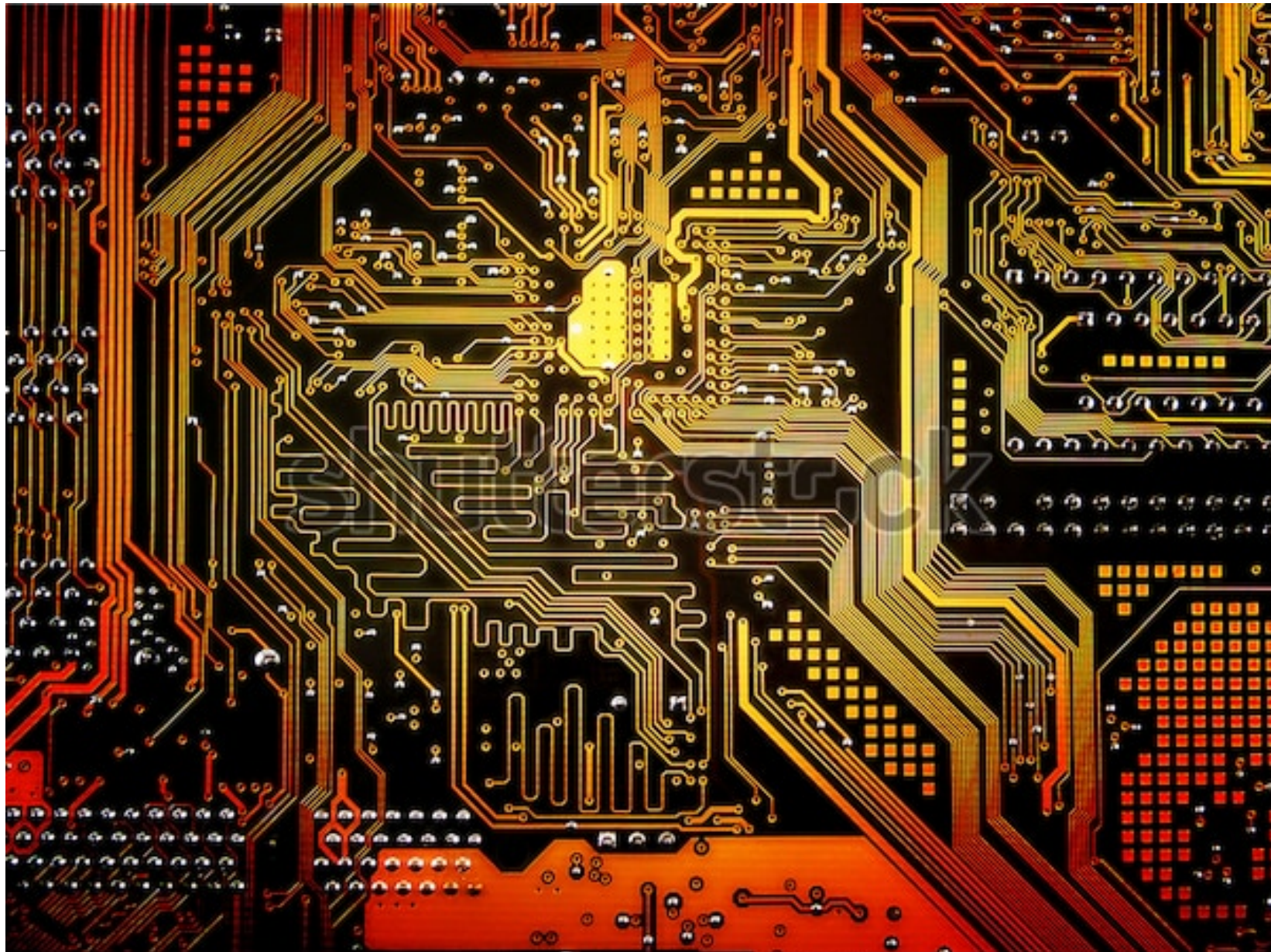
Barramento

Em um computador real existem vários **níveis de barramento**, com velocidades diferentes

- Quanto mais próximo da CPU mais rápido ele é

O barramento de dados é a estrada elétrica com várias vias que **ligam** o processador central aos outros componentes da placa mãe

- Tais como memória RAM, disco rígido e controladores de I/O



www.shutterstock.com · 2545091

Barramento

Os barramentos são basicamente linhas de comunicação elétricas em forma **paralela** ou **serial** com **conexões múltiplas**, isto é, permite ligar vários dispositivos

Barramento Interno:

- Objetiva conectar a CPU, memória e interfaces. Pretendem conectar os dispositivos locais, e não outras máquinas externas ao computador

Barramento Externo:

- Conecta computadores aos periféricos.

Barramento

Características de um barramento:

- **BARRAMENTO DE DADOS:** transferir dados e referências de endereço de memória
- **BARRAMENTO DE ENDEREÇO:** identificar a origem e o destino do dado
- **BARRAMENTO DE CONTROLE:** enviar dados de controle entre os subsistemas

Memória

A memória é o segundo principal componente do computador

Idealmente a memória deveria ser:

- Muito rápida
- Muito grade
- Barata

Memória

Memória é um componente de computador, dispositivos ou mídia de gravação que **retêm** os dados digitais usados pelo computador durante um intervalo do tempo

É um dos componentes fundamentais de qualquer computador moderno, que, junto com a CPU, constitui o **esqueleto básico** de um computador

Memória

Características de uma memória:

- **LOCALIZAÇÃO:** dentro da CPU, interna ou externa
- **CAPACIDADE:** definida pelo tamanho e quantidade de bytes
- **VOLATILIDADE:** a memória pode ser permanente

Memória

Desempenho da memória:

- **TEMPO DE ACESSO:** tempo que se leva para alcançar uma determinada posição na memória
- **TAXA DE TRANSFERÊNCIA:** taxa em que a informação pode ser lida ou escrita na memória
- **CICLO DE MEMÓRIA:** estabelece o tempo em que uma memória precisa esperar para realizar uma nova leitura e escrita
- Obs.: Executar o benchmark de disco no Ubuntu para mostrar valores reais.
- Pesquisas em memórias para economizar energia (smartphones)

Memória RAM

Random Access Memory (RAM): todo o bit da memória pode ser endereçado a qualquer instante

A RAM consiste em bancos de transistores do circuito integrado onde podem estar ligados ou desligados (representando um ou zero)

Os chips de RAM precisam de energia constante para manter o que é armazenado nele

- Um desligamento da energia pode fazer com que a memória RAM perca seus dados. Por esse motivo, a memória RAM é chamada de **memória volátil**

Memória RAM

A CPU trabalha diretamente com a memória RAM. Assim, todos os dados processados pelo chip são enviados pelos módulos de memória RAM

Porém, a memória RAM é muito mais lenta do que o processador.

- Isso resulta em longos períodos de ociosidade e, conseqüentemente, desperdício de capacidade do processador.

Para resolver esse problema e fazer com que a CPU trabalhe com a força máxima possível, foi criada a memória cache

Memória Cache

Esse tipo de memória possui alta velocidade e tem por função armazenar dados e instruções que a CPU poderá precisar em breve

- O objetivo da cache é potencializar o desempenho do chip de processamento, evitando que fique ocioso por longos períodos

Quando uma informação não é encontrada nos registradores, a cache é o primeiro lugar onde o processador irá buscar essa informação

- Se os dados ou a instrução requisitada estiverem na cache (**cache hit**), a informação é transferida para o processador
- Caso contrário (**cache miss**), o processador aguarda enquanto a informação desejada é transferida da memória principal para a cache e posteriormente para o processador

Memória ROM

Read Only Memory (ROM): não precisa de energia para guardar suas informações mas também não podem ser reescritas

Quando o computador é ligado, o programa armazenado na ROM é executado, dando os primeiros comandos para o computador funcionar

Esse programa é chamado de **BIOS** (*Basic Input/Output System*) o qual executa as funções iniciais do computador

Memória Flash

Tipo especial de memória RAM que apresenta a característica de não ser volátil. Apesar de ser um pouco mais lenta que as memórias RAM voláteis ela é muito mais rápida que os discos rígidos

Se popularizou com a disponibilidade de interface USB, que permitiu a construção de memórias não voláteis portáteis

O mesmo tipo da memória Flash pode ser usado em pequenos cartões de memória como cartão SD, Compact Flash, Memory Stick

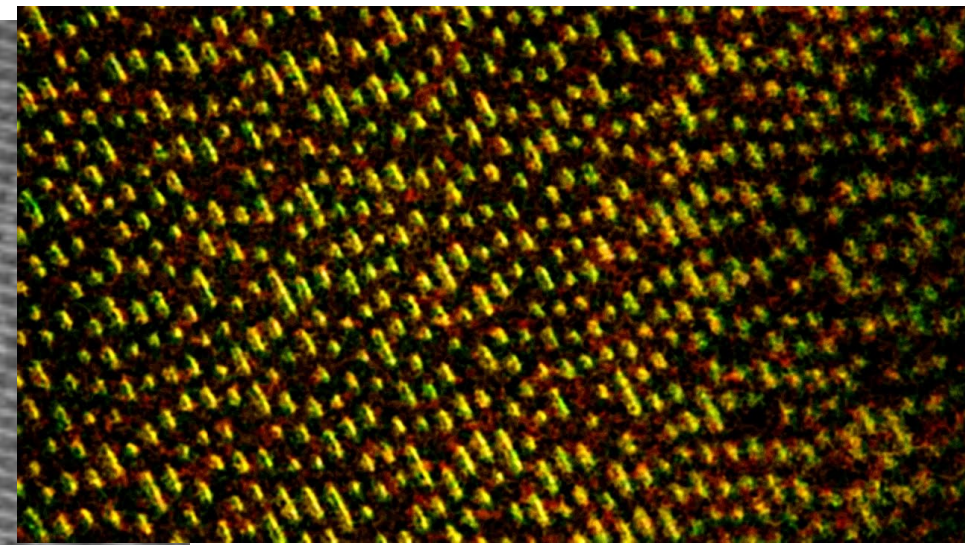
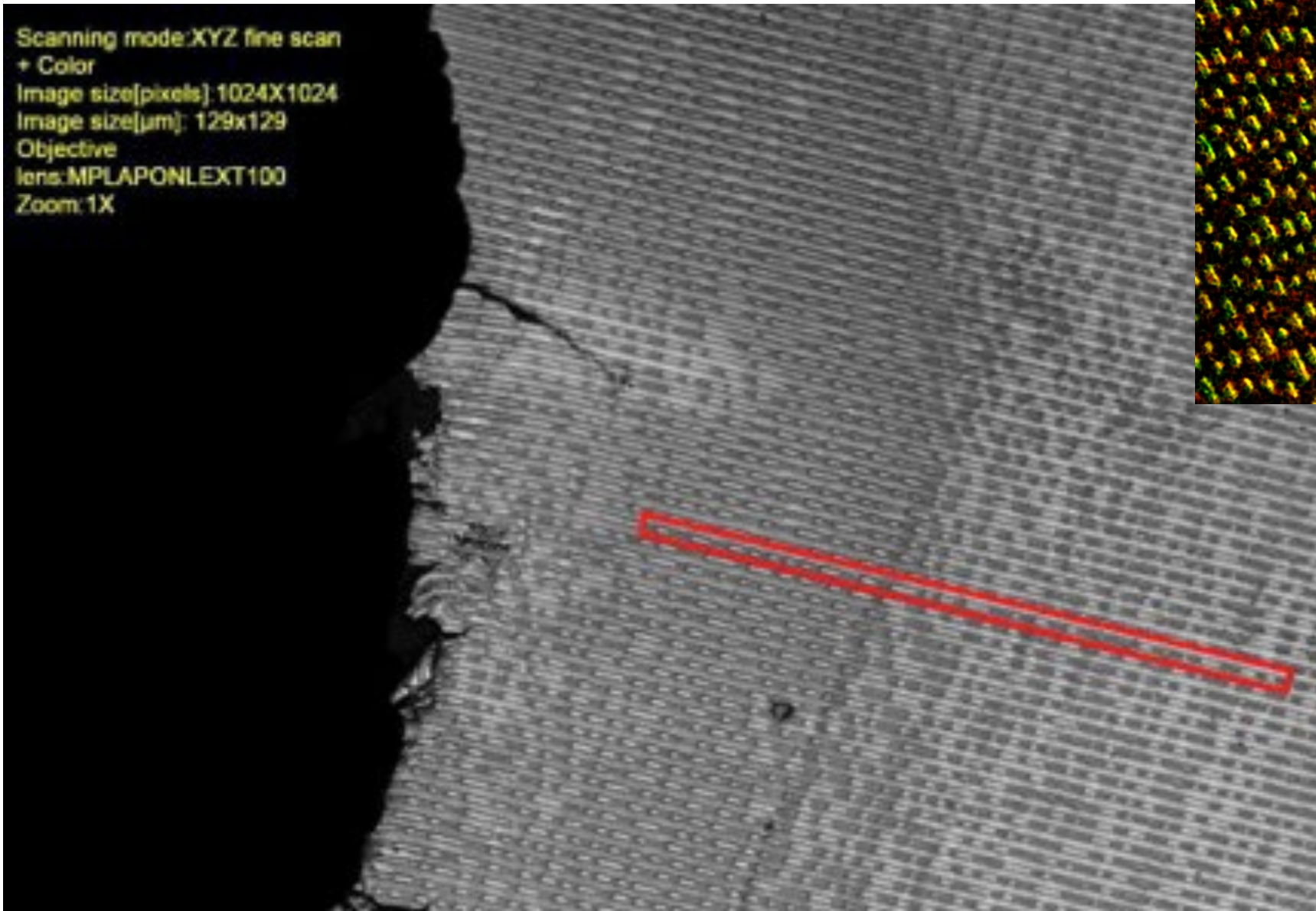
Disco Rígido

A **Memória Secundária** é o último nível da hierarquia de memória. É composta pelos dispositivos de armazenamento de massa, normalmente discos rígidos, de grande capacidade e menor custo por byte armazenado.

Os programas e arquivos são armazenados integralmente na memória secundária, que são dispositivos de memória **não voláteis**. As principais tecnologias utilizadas em memórias secundárias são:

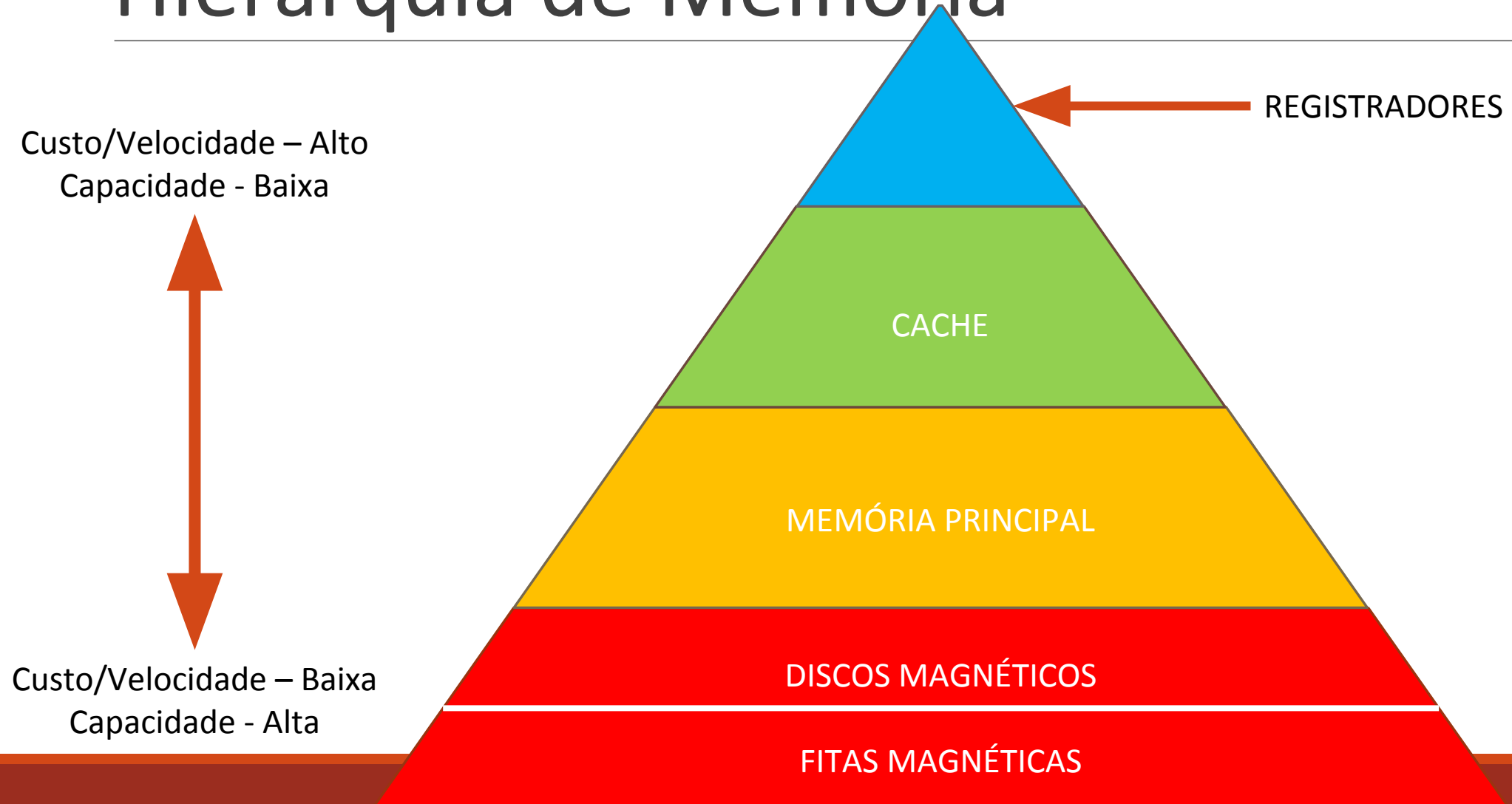
- Memórias Magnéticas
- Memórias Ópticas
- Memórias de Estado Sólido

Scanning mode:XYZ fine scan
+ Color
Image size[pixels]: 1024X1024
Image size[μm]: 129x129
Objective
lens:MPLAPONLEXT100
Zoom:1X





Hierarquia de Memória



Hierarquia de Memória

Devido a essas diferenças no custo e no tempo de acesso, é vantajoso construir memória como uma hierarquia de níveis

A memória mais rápida está próxima do processador e a memória mais lenta e barata está mais abaixo na pirâmide

O objetivo é oferecer ao usuário o máximo de memória disponível na tecnologia mais barata, enquanto se fornece acesso na velocidade oferecida pela memória mais rápida

Hierarquia de Memória

O uso da hierarquia de memória é viável por conta do **princípio de localidade** que está presente na grande maioria das aplicações computacionais

Princípio de Localidade:

- Afirma que existe uma grande probabilidade de o processador executar os mesmos trechos de código e utilizar repetidamente dados próximos

Hierarquia de Memória

Princípio de Localidade

- Há dois tipos diferentes de localidade:
 - **Localidade temporal:** se um local de dados é referenciado, então, ele tenderá a ser referenciado novamente em breve
 - **Localidade espacial:** se um local de dados é referenciado, então, os dados com endereços próximos tenderão a ser referenciados em breve

Hierarquia de Memória

Registradores:

- Armazena tipicamente uma palavra de dados
- Memória cara, pequena e de acesso rápido

Memória Cache:

- Armazena os blocos de dados mais usados
- Aumenta o desempenho do sistema

Hierarquia de Memória

Memória principal:

- Memória volátil, de porte e velocidade médios
- Administrada pela Unidade de Gerência de Memória

Memória secundária:

- Não é diretamente acessível pelo processador
- Uso de interfaces para transferência de informações
- Não volátil, lenta e de grande capacidade

Dispositivos de E/S

Um terceiro elemento fundamental de um sistema de computação é o conjunto de módulos de E/S

- Cada módulo se conecta com o barramento do sistema e controla um ou mais dispositivos periféricos.

Tipos de dispositivos externos:

- **Comunicação humana:** Teclado, monitor, mouse...
- **Comunicação com máquina:** Disco, Fita Magnética...
- **Comunicação remota:** Interface de rede, modem...

Dispositivos de E/S

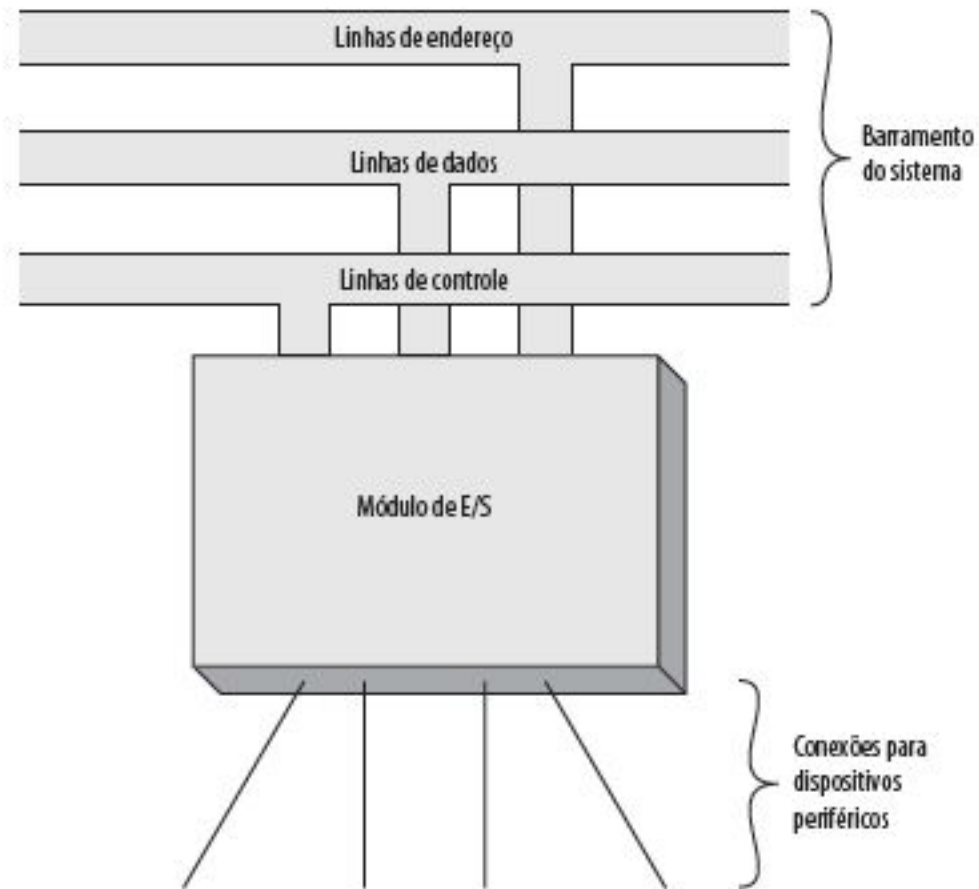
Os periféricos não são diretamente conectados ao barramento do sistema por alguns motivos:

- Grande variedade de periféricos
- Taxa de transferência de dados menor
- Uso de formato de dados diferentes

Módulo de E/S

- Fornece uma interface com o processador e a memória, através do barramento do sistema e permite a interface com um ou mais periféricos.

Módulo de E/S genérico



Módulo de E/S

As principais funções do módulo de E/S são:

- Controle e temporização.
- Comunicação com o processador.
- Comunicação com o dispositivo.
- Armazenamento temporário (buffering) de dados.
- Detecção de erro.

Módulo de E/S – Comunicação com o Processador

1. Processador interroga o módulo de E/S sobre o estado do dispositivo a ele conectado (**sinais de estado**)
2. O módulo de E/S retorna o estado do dispositivo
3. Se o dispositivo estiver pronto, o processador solicita a transferência de dados através de um comando
4. O módulo obtém uma unidade de dados do periférico (**sinais de controle e sinais de dados**)
5. Dados são transferidos do módulo para o processador

Módulo de E/S – Armazenamento Temporário

Discrepância na velocidade de transferência de dados

- Tarefa essencial do módulo de E/S é o **armazenamento temporário de dados**.

1. A transferência de dados da memória principal para o módulo de E/S é feita rapidamente
2. Esses dados são temporariamente armazenados
3. Por fim, enviados para o dispositivo periférico em uma taxa adequada

Tipos de Entrada e Saída

1. Entrada e Saída Programada:
CPU controla diretamente o módulo de E/S

2. Entrada e Saída por Interrupção:
Módulo de E/S notifica a CPU quando a operação tem fim

3. Entrada e Saída por DMA:
CPU terceiriza a ação de gerenciar a operação de E/S

Entrada e Saída Programada

1. Quando o processador está executando um programa e encontra uma **instrução relacionada a E/S**, ele executa essa instrução emitindo um **comando** ao módulo de E/S apropriado.
2. Na E/S programada, o **módulo de E/S executa a operação** requisitada e **sinaliza o término** da operação carregando um valor apropriado no registrador de estado de E/S. *Nenhuma outra ação é executada para alertar o processador sobre o término da operação.*
3. Assim, é responsabilidade de processador **verificar periodicamente** o estado do módulo de E/S até descobrir se a operação terminou.

Entrada e Saída Programada – Comandos

Para executar uma instrução relacionada a E/S, o processador emite um endereço, especificando o módulo de E/S e dispositivo externo em particular, e um comando de entrada e saída. Os comandos de E/S são:

1. **Controle:** usado para ativar um periférico e dizer-lhe o que fazer.
2. **Teste:** usado para testar diversas condições de estado associadas a um módulo de E/S e seus periféricos.
3. **Leitura:** faz com que o módulo de E/S obtenha um item de dados do periférico e o coloque em um buffer interno.
4. **Escrita:** faz com que o módulo de E/S apanhe um item de dado do barramento de dados e depois transmita esse item de dado ao periférico

Entrada e Saída Programada - Desvantagem

A principal desvantagem desta técnica é o **consumo de tempo do processador**, mantendo-o desnecessariamente ocupado

Entrada e Saída Dirigida por Interrupções

O problema da E/S programada é que o processador tem de esperar um longo tempo até que o módulo de E/S esteja pronto para receber ou enviar dados.

Uma alternativa é o processador **enviar um comando** de E/S para o módulo e **continuar executando outras ações**. O processador será **interrompido pelo módulo** quando este estiver pronto para a troca de dados. Os dados são trocados e o processador retoma seu processamento original.

Entrada e Saída Dirigida por Interrupções Ponto de Vista do Processador

1. O processador emite um comando READ
2. Ele prossegue com outras tarefas
3. Ao final de cada ciclo de instrução o processador verifica se há interrupções
4. Quando ocorre uma interrupção do módulo de E/S o processador salva o contexto do programa atual e processa a interrupção
5. O processador restaura o contexto

Entrada e Saída Dirigida por Interrupções Ponto de Vista do Módulo de E/S

1. O módulo recebe um comando READ do processador
2. O módulo lê o dado do periférico
3. Quando termina, o módulo envia um sinal de interrupção ao processador através de uma linha de comando
4. O módulo espera até que seus dados sejam solicitados pelo processador

Acesso direto à Memória

O acesso direto à memória (DMA) envolve um módulo adicional ao barramento do sistema.

- Esse módulo é capaz de imitar o processador e, na realidade, assumir o controle do sistema do processador.

Ele precisa fazer isso para transferir dados de e para a memória pelo barramento do sistema.

Para essa finalidade, o módulo DMA precisa usar o barramento apenas quando o processador não precisa dele

- Ou então ele precisa forçar o processador a suspender a operação temporariamente (roubo de ciclo).

Acesso direto à Memória

As possíveis configurações são:

- Barramento único, DMA separado
- Barramento único, DMA integrado
- Barramento específico de E/S

Acesso direto à Memória

Barramento único, DMA separado

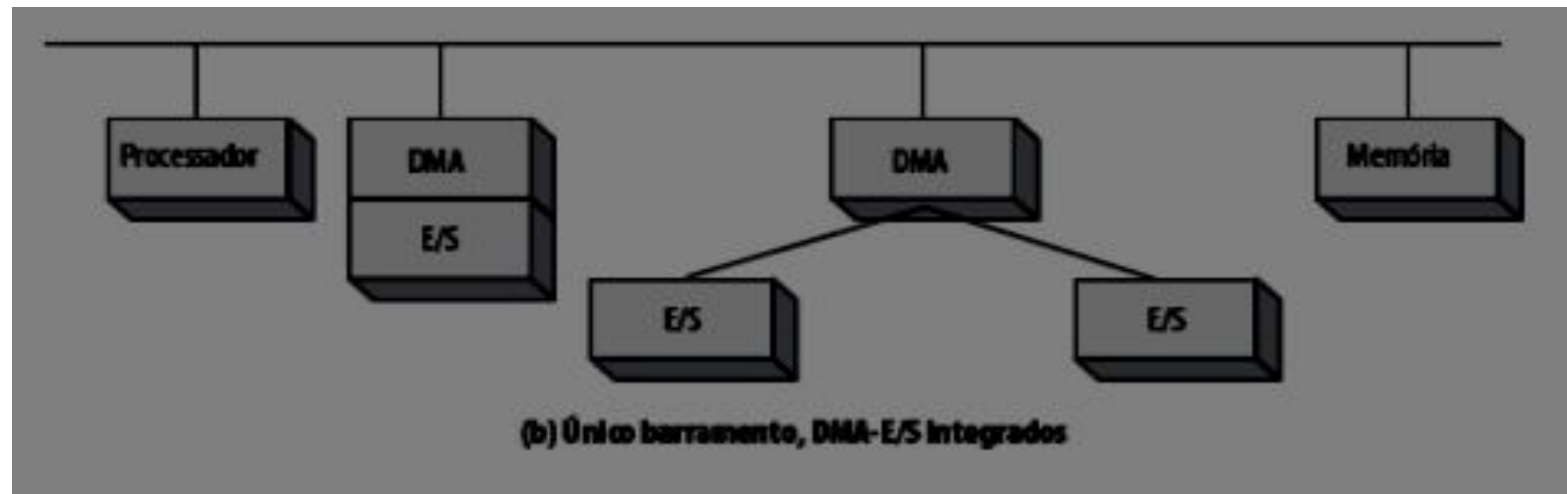
- Uma solução barata, mas ineficiente; pois a transferência de cada palavra consome vários ciclos de barramento.



Acesso direto à Memória

Barramento único, DMA integrado

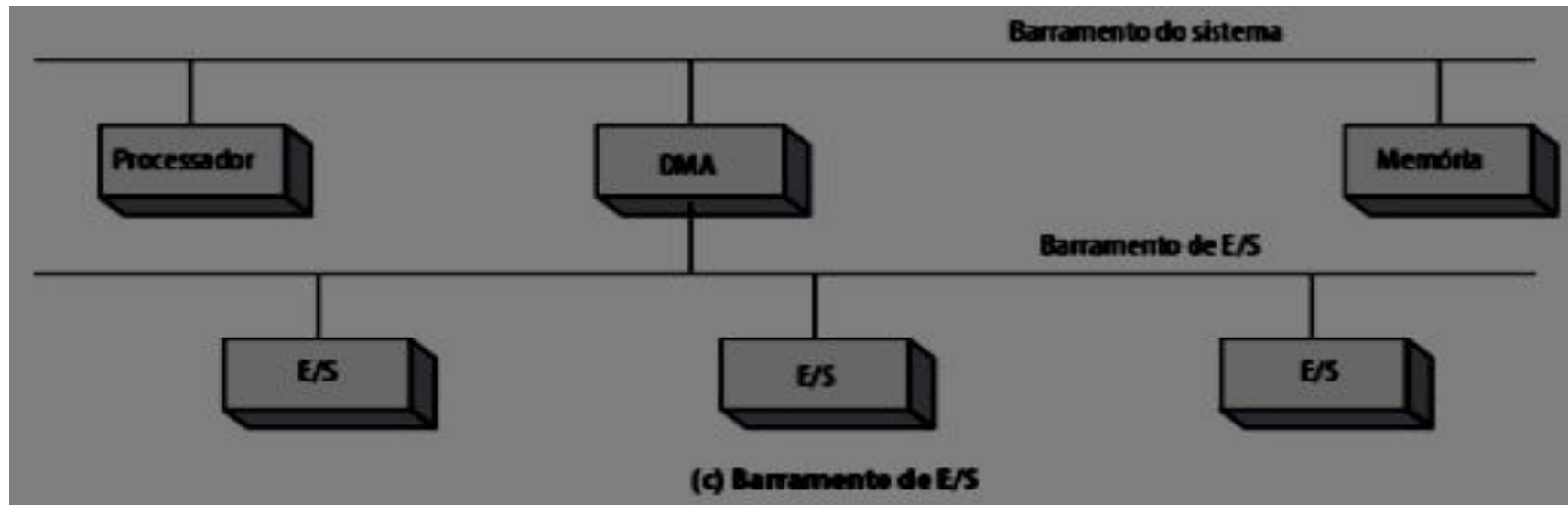
- O DMA é integrado ao barramento. Uma solução mais eficiente, visto que cada transferência usa o barramento apenas uma vez (DMA para memória).



Acesso direto à Memória

Barramento específico de E/S

- É adicionado um barramento específico para os dispositivos de entrada e saída.
- Eficiente e custoso



Principais Arquiteturas

Arquitetura de Harvard:

- Separação das memórias de instrução e de dados

Arquitetura de Von Neumann:

- Uma memória para armazenar instruções e dados

Arquitetura de Von Neumann

Se caracteriza pela possibilidade de uma máquina digital armazenar seus programas no mesmo espaço de memória que os dados

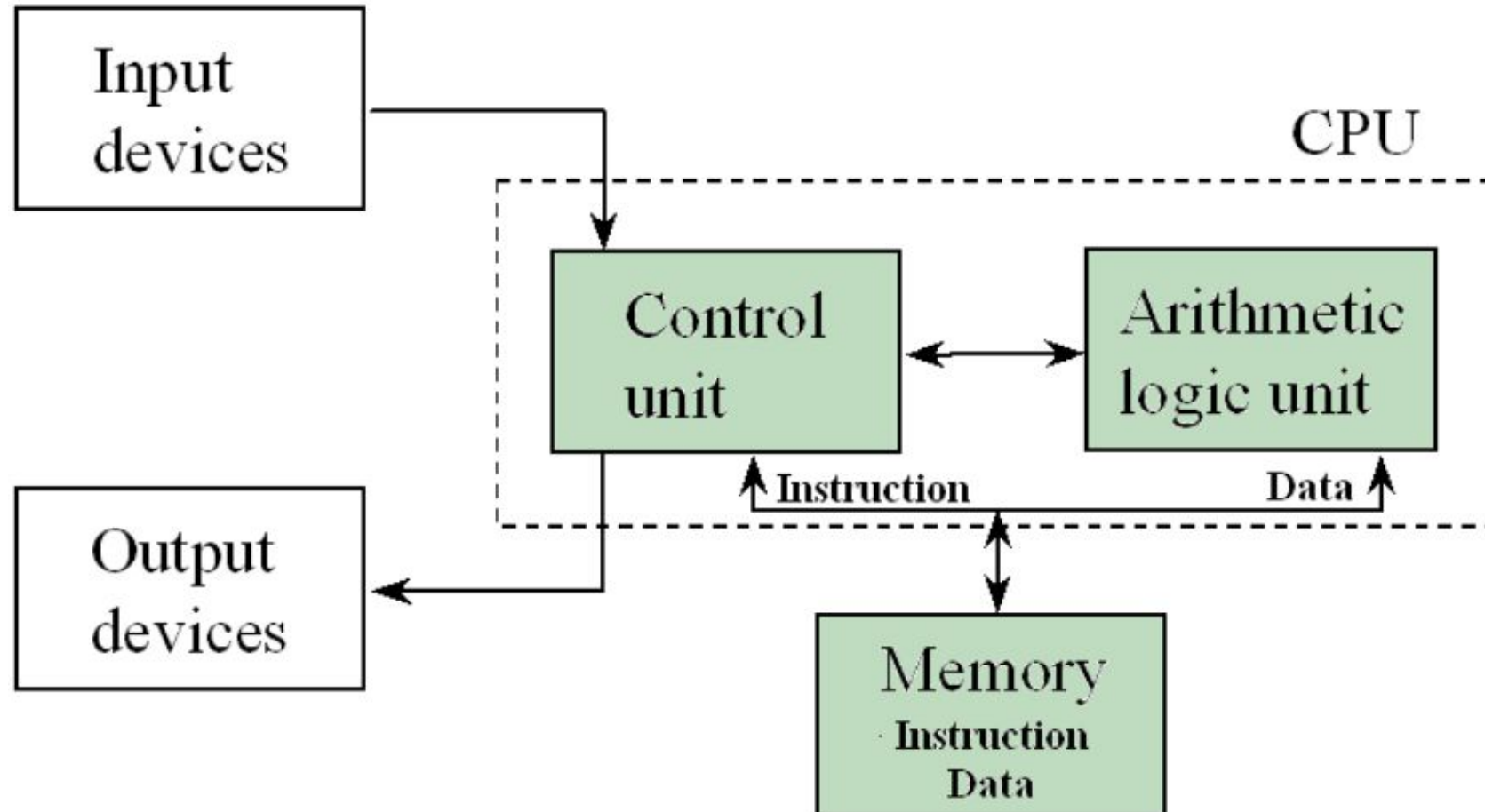
- Utiliza uma unidade de processamento (CPU) e uma de armazenamento ("memória") para comportar, respectivamente, instruções e dados

Arquitetura de Von Neumann

A máquina proposta por Von Neumann reúne os seguintes componentes:

- **Uma memória**
- **Uma unidade aritmética e lógica (ALU)**
- **Uma unidade central de processamento (CPU)**, composta por diversos registradores
- **Uma Unidade de Controle**, responsável por buscar um programa na memória, instrução por instrução, e executá-lo sobre os dados de entrada.

Arquitetura de Von Neumann



Arquitetura de Harvard

É uma arquitetura de computador que se distingue das outras por possuir duas memórias diferentes e independentes em termos de barramento e ligação ao processador.

- Tem, como principal característica, o acesso à memória de dados de modo separado em relação à memória de programa

Propõe a separação de barramentos de dados das memórias – onde estão as instruções de programa – e das memórias de dados

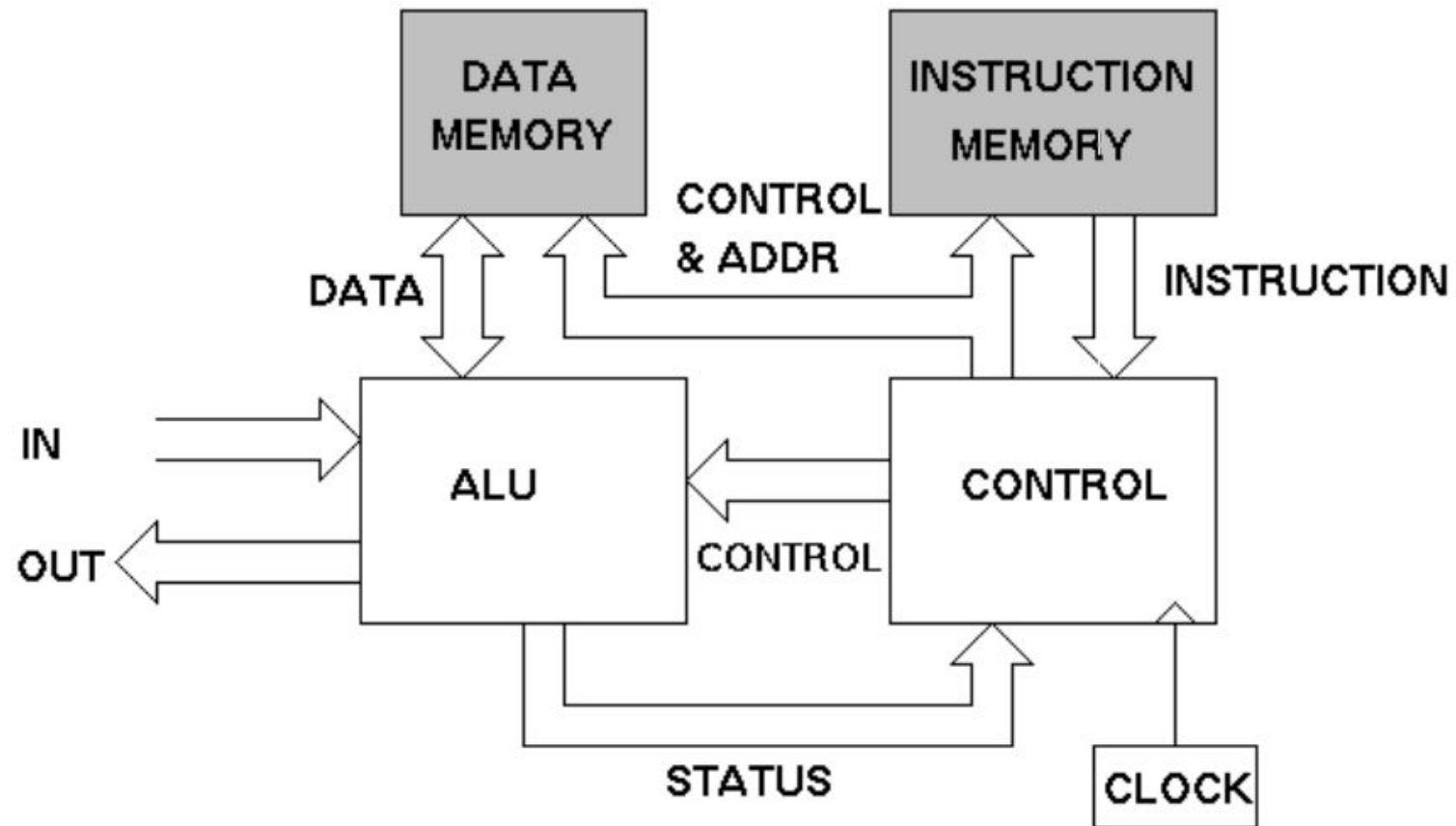
- Permitindo que um processador possa acessar as duas simultaneamente

Arquitetura de Harvard

Principal Vantagem:

- Leitura de instruções e de alguns tipos de operandos pode ser feita ao mesmo tempo em que a execução das instruções
 - O sistema fica todo o tempo executando instruções
- Significativo ganho de velocidade
- Possui um repertório com menos instruções e que são executadas apenas num único ciclo de clock

Arquitetura de Harvard



Havard x Von Neumann (Qual a mais utilizada?)

Arquitetura de Harvard

- Armazena dados e instruções em memórias distintas
- V: pode acessar instruções e dados simultaneamente
- D: circuito eletrônico maior e mais complexo

Arquitetura de Von Neumann

- Armazena dados e instruções no mesmo local
- V: circuito eletrônico mais simples
- D: não pode ler instruções e dados simultaneamente

Para Pensar

O barramento do sistema é usado para conectar os componentes principais do computador, como processador, memória e dispositivos de entrada/saída (E/S). Em geral, o sistema dispõe de barramentos de dados, de endereços e de controle.

- a) Certo
- b) Errado

Para Pensar

Em termos de organização e arquitetura de computadores, é correto afirmar que:

- a) quando o controlador de E/S usa o DMA para ler ou escrever dados de ou para a memória, ele não necessita de intervenção da CPU.
- b) para ler ou escrever dados de ou para a memória, usando o DMA, o controlador necessita executar uma rotina de interrupção para avisar a CPU que iniciará essa tarefa.
- c) em geral a CPU tem precedência sobre o controlador de E/S na utilização do barramento.
- d) o processo denominado roubo de ciclo consiste em a CPU apropriar-se de ciclos do barramento que estão em execução para atender dispositivos de E/S
- e) o barramento EISA, além de contar com uma conexão dedicada com o controlador de memória, sem prejudicar o tráfego CPU-memória, também conecta na outra extremidade periféricos de alta largura de banda.

Para Pensar

Analise as afirmativas a seguir e identifique a opção correta

- I. Uma das funções da unidade central de processamento é a de executar instruções armazenadas na memória principal do computador
 - II. As memórias primárias possuem como característica a capacidade de armazenamento maior do que as memórias secundárias
 - III. Em um computador existem, pelo menos, dois barramentos: o de memória e o de entrada/saída
- a) Somente I e II estão corretas
 - b) Somente II e III estão corretas
 - c) Somente I e III estão corretas
 - d) Somente I está correta
 - e) Somente II está correta

Para Pensar

É um recurso responsável pela coordenação do funcionamento da memória de um microcomputador, que transforma as instruções lógicas (virtuais) em endereços físicos nos bancos de memória e anota onde cada informação do sistema está hospedada na memória. É ele que informa onde o processador deve procurar os dados e instruções na memória. Trata-se de:

- a) Unidade de Controle
- b) Unidade Lógica e Aritmética
- c) Unidade Central de Processamento – CPU
- d) Unidade de Gerenciamento de Memória
- e) Registrador

Seção 3

CONCEITOS SOBRE SISTEMAS OPERACIONAIS

Processo

Um **processo** é basicamente um programa em execução. Associado a cada processo está o seu **espaço de endereçamento** e um **conjunto de recursos e informações** necessárias para executar um programa.

Um processo é fundamentalmente um **container** que armazena todas as **informações necessárias para executar um programa**

Processo

Um processo precisa de vários recursos para executar uma tarefa

- Exemplo: tempo de CPU, memória, dispositivos de E/S, etc.

É responsabilidade do sistema operacional “organizar” esses processos no computador, permitindo que os mesmos compartilhem recursos de forma organizada

Podemos dizer que o Sistema Operacional atua como um **gerente de processos**.

- É cargo do Sistema Operacional gerenciar os processos, determinando a ordem de uso do processador, garantindo que cada um tenha acesso e tempo suficiente para executar normalmente.

Processo

Em diferentes momentos, um processo pode estar utilizando-se do processador, ou simplesmente aguardando.

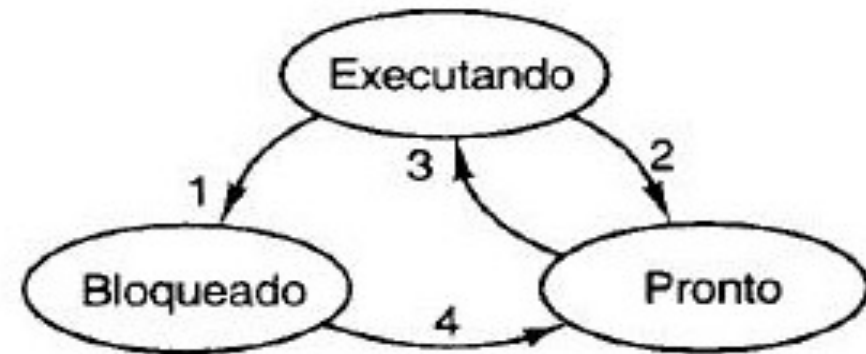
Existem situações onde o processo, mesmo tendo o processador disponível, fica impossibilitado de prosseguir a execução, pois geralmente está aguardando por alguma entrada.

- Para que não exista desperdício no uso do processador, processos que estejam aguardando algum dado externo ficam bloqueados, deixando aos demais processos disputarem o uso do processador

Processo

Dessa forma, podemos dizer que os processos possuem três estados:

- 1. Em execução:** realmente utilizando o processador
- 2. Pronto:** aguardando pelo uso do processador
- 3. Bloqueado:** impossibilitado de usar o processador até que algum evento externo aconteça.



Gerenciamento de Processo

Ações relacionadas ao gerenciamento de processos:

- Criação e remoção de processos do usuário e do sistema
- Suspende e reiniciar a execução de processos
- Prover mecanismos para permitir a sincronização entre processos
- Prover mecanismos para permitir a comunicação entre processos
- Prover mecanismos para lidar com deadlocks

Memória Principal

A memória principal é um vetor de bytes ou palavras:

- Cada palavra ou byte do vetor possui endereço único

Funciona como um imenso repositório de dados compartilhados pela CPU e dispositivos de E/S:

- Os dados devem, primeiramente, passar pela memória principal, antes de serem entregues a CPU ou a E/S
- Velocidade próxima a da CPU (oferece rápido acesso)

Mas aonde entra o Sistema Operacional nessa história?

Memória Principal e o SO

Para um programa ser executado, o seu código-fonte deve estar presente na memória principal:

- O SO é responsável por alocar o espaço inicial e atribuir todos endereços necessários as palavras ou bytes

Durante a execução do programa, novas informações (dados e códigos) podem surgir e devem estar na memória principal:

- O SO também é responsável por gerenciar tudo isso

Quando o programa termina, o SO também é responsável por liberar todos os espaços e deixar livre para outros programas

Gerenciamento de Memória

O problema básico para o gerenciamento de memória é que os programas atuais são muito grandes para rodarem, completamente, na memória cache

O gerenciador de memória deve ser capaz de controlar parte da memória que está em uso (e quais não estão)

- Alocar memória para processos quando eles necessitam e desalocar quando eles terminam
- Gerenciar a troca entre a memória principal e o disco

Armazenamento

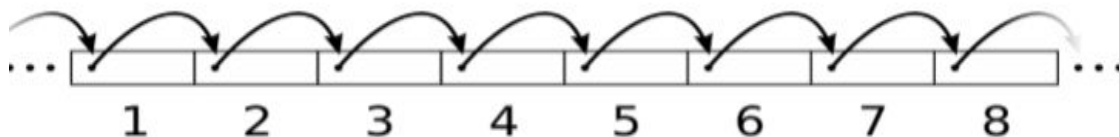
Computadores podem ter tipos distintos de meios físicos:

- Por exemplo: Discos, fitas magnéticas, discos ópticos...

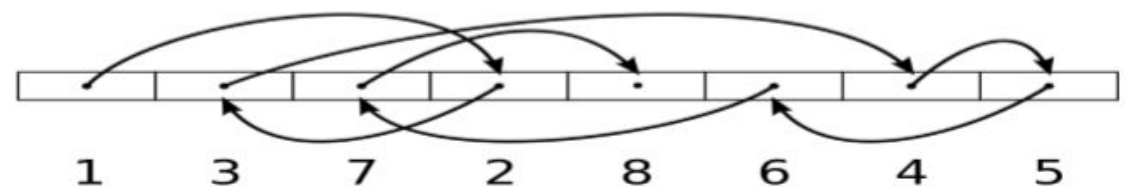
Cada meio físico possui uma organização interna própria e características exclusivas

- Por exemplo: método de acesso

Sequential access



Random access



Armazenamento

Uma das principais funções do sistema operacional é ocultar as peculiaridades dos discos e de outros dispositivos de E/S

- Fornecendo ao programador um modelo de arquivos agradável e claro, independente de dispositivos

Outro conceito fundamental que compõe todos os Sistemas Operacionais é o **sistema de arquivos**

Sistema de Arquivo

Os **sistemas de arquivos** estruturam a informação guardada em uma unidade de armazenamento, podendo ser representada de forma **textual** ou **gráfica**, utilizando um gerenciador de arquivos

Sistema de Arquivo

Mecanismo de abstração utilizado para armazenar dados no disco e de acessá-las de forma simples, rápida e segura

Sistemas Operacionais disponibilizam diretórios:

- Arquivos especiais que armazenam outros arquivos

Um diretório pode armazenar arquivos e/ou outros diretórios criando um sistema hierárquico de arquivos

Armazenamento e SO

De uma maneira geral, em relação aos arquivos, são tarefas comuns ao Sistema Operacional:

- Criar e remover arquivos e diretórios
- Primitivas para manipular arquivos e diretórios
- Mapear arquivos no armazenamento secundário
- Fazer backup de arquivos em mídias de armazenamento não volátil
- Gerenciamento do espaço livre e do espaço ocupado

Segurança

Se um sistema computacional possui **múltiplos usuários** e permite a **execução simultânea de múltiplos processos**, então o acesso aos dados deve ser gerenciado.

Por isso, há mecanismos que garantem que arquivos, segmentos de memória, CPU e outros recursos sejam operados somente por processos que foram **autorizados** para isso

Segurança e Proteção

Proteção: É qualquer mecanismo para controlar o acesso de processos ou usuários aos recursos definidos pelo SO

Na memória principal podem estar vários processos:

- É tarefa do SO garantir que cada processo só tenha acesso às regiões que foram atribuídas a ele

Isso também é válido para outros recursos de alto nível:

- Proteção pode melhorar a confiabilidade do sistema ao detectar erros nas interfaces entre os subsistemas
- Isso evita que um subsistema falho influencie no outro

Segurança e Proteção

Mas somente a proteção não é o suficiente para o sistema

Considere a seguinte situação:

- Um hacker invade o seu sistema e apaga os seus arquivos com sua conta de usuário (a proteção não foi violada)

Segurança: Defesa do sistema contra ataques internos ou externos

Proteções e segurança requerem que o sistema esteja apto para distinguir um usuário dos demais

Seção 4

TIPOS DE SISTEMAS OPERACIONAIS

Principais tipos de Sistemas Operacionais

1. Sistemas Operacionais para computadores de grande porte
2. Sistemas Operacionais para computadores portáteis
3. Sistemas Operacionais embarcados
4. Sistemas Operacionais de tempo real

Sistemas Operacionais para Computadores de Grande Porte

Computadores de grande porte

- Aqueles que ocupam uma sala inteira, ainda encontrados em grandes corporações

Principal diferença: grande capacidade em termos de E/S

- Um grande número de discos e usuários no sistema

Processo simultâneo de inúmeras tarefas

- A maioria delas necessita de quantidades prodigiosas de E/S

Há três tipos de serviços:

- Processamento em batch (lote)
- Processamento de transações e
- Processamento em tempo compartilhado

Tipos de Serviços

1. Processamento em batch (lote)

Processamento de tarefas de rotina, sem a presença interativa dos usuários do sistema

2. Processamento de transações

Gerenciam grandes quantidades de pequenas requisições

3. Processamento de tempo compartilhado

Possibilitam que múltiplos usuários remotos executem suas tarefas simultaneamente no computador

Sistemas Operacionais para Computadores Portáteis

É um tipo de sistema operacional desenvolvido especificamente para smartphones, tablets, PDAs ou outros dispositivos móveis.

- Embora alguns computadores, como um típico laptop, sejam portáteis, os sistemas operacionais geralmente usados neles não são considerados móveis

Combinam características de um sistema operacional do computador pessoal com outros recursos úteis para uso móvel

- Tela sensível ao toque, Bluetooth, GPS de navegação móvel, câmera fotográfica, reconhecimento de voz

Sistemas Operacionais Embarcados

Sistemas embarcados:

- São computadores que gerenciam dispositivos que não são considerados computadores de propósito geral:
- Ex: Micro-ondas, televisões, aparelhos de som, carros...

Geralmente, nenhum software não confiável será executado por esse tipo de Sistema Operacional:

- Normalmente, todas as aplicações já estão no sistema
- Caso contrário, as aplicações são distribuídas pelas próprias empresas fabricantes

Não há a necessidade de proteção entre as aplicações

Sistemas Operacionais de Tempo Real

O sistema de tempo real é um sistema computacional que requer não somente os **resultados** computacionais estarem **corretos**, mas também que os resultados produzidos estejam dentro de um **período pré-estabelecido**. Resultados produzidos após o deadline, mesmo que corretos, podem não ser úteis.

Sistemas Operacionais de Tempo Real

Possuem o **tempo como parâmetro fundamental**:

- Prazos rígidos para a execução de determinadas tarefas
- **Exemplo:** considere uma linha de montagem:
 - Cada ação deve ser executada por período específico
 - Se a ação for tomada muito cedo, muito depois ou durar um tempo a mais ou a menos, pode prejudicar o produto

Muitos exemplos são encontrados no controle de processos aeronáuticos, militares, etc

Tipos de Sistemas Operacionais de Tempo Real

Sistema de tempo real crítico

Ações precisam necessariamente acontecer em dados instantes ou períodos de tempo

Sistema de tempo real não crítico

Descumprimento ocasional de um prazo, embora não desejável, é perfeitamente aceitável

Sistemas de Tempo Real Crítico

Tempo de execução da tarefa (deadline), não pode sofrer qualquer tipo de atraso

- **Exemplos:** Controle de vôo, controle de esteiras de fábrica, sinais de trânsito, usinas nucleares, equipamentos para monitoramento de saúde, freios automotivos ABS



Sistemas de Tempo Real Não Crítico

Tempo é parâmetro fundamental, porém uma possível falha ou retardo não ocasionará danos irreversíveis

- **Exemplos:** LynxOS (UNIX), MP3 player, gravadoras de CD/DVD, Playstation, telefones digitais



Exercício

- Pesquise por sistemas operacionais de tempo real (crítico e não crítico / soft and hard) e suas características.
 - Pelo menos 2 soft RTS e 2 hard RTS
 - Descreva as similaridades e diferenças
- RTS = Real time system