

# Conceitos Fundamentais

DCA0104 – Arquitetura de Computadores

Diogo Pedrosa

[diogo@dca.ufrn.br](mailto:diogo@dca.ufrn.br)

DCA – CT – UFRN

# Introdução

- Alguns desafios para o estudo de computadores:
  - **Variedades de dispositivos** que podem ser chamados de computadores
  - **Evolução rápida da tecnologia** relacionada aos computadores

# Introdução

- Um ponto importante...
  - Há conceitos fundamentais que se aplicam nos mais diferentes dispositivos, ao longo da história da computação
- Necessário saber o que significa **arquitetura de computadores** e **organização de computadores**

# Introdução

- **Arquitetura de computadores** → refere-se aos atributos de um sistema que são visíveis a um programador
  - Impacto direto sobre a execução lógica do programa
  - Conjunto de instruções, número de bits para representar dados, técnicas de endereçamento para memória, mecanismos para E/S, ...

# Introdução

- **Organização de computadores** → consiste nas unidades operacionais (processador, memória e E/S) e suas interconexões, o que permitem realizar as especificações de arquitetura
  - Sinais de controle (necessários para a execução de instruções), interfaces dos periféricos, tecnologia de memória, ...

# Introdução

- Fabricantes possuem **famílias** de dispositivos que têm a mesma arquitetura, mas organizações diferentes

# Classes de Aplicações

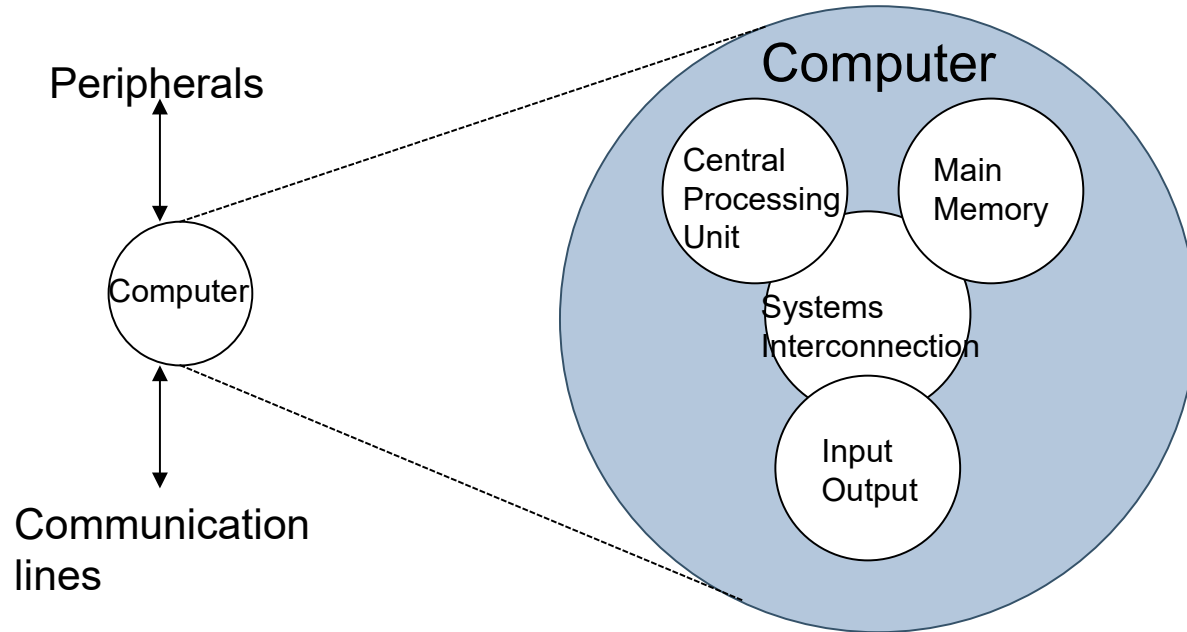
- Computadores *desktop*
- Supercomputadores
- Computadores embutidos

# Computadores – Estrutura e Função

- Computadores são sistemas complexos
- Necessitam de uma visão hierárquica
  - Subsistemas inter-relacionados
  - **Estrutura** → maneira como os componentes de um subsistema se relacionam
  - **Função** → maneira como cada componente funciona, individualmente



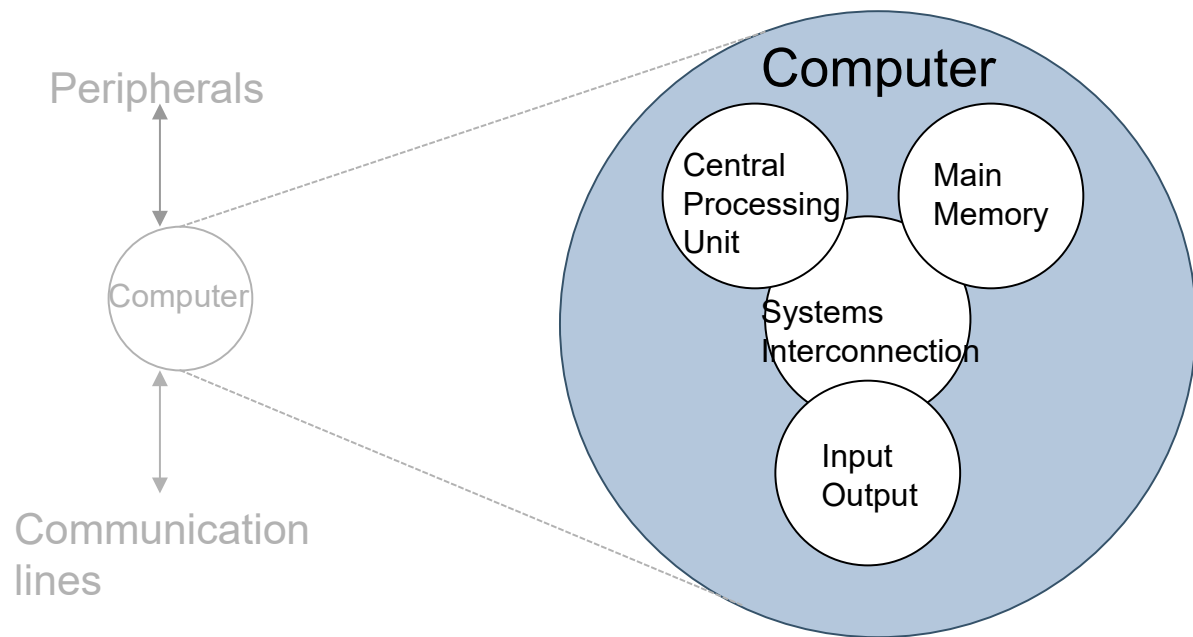
# Computadores – Estrutura e Função



As funcionalidades exercidas por um computador são:

- Processamento de dados
- Armazenamento de dados
- Movimentação de dados
- Controle

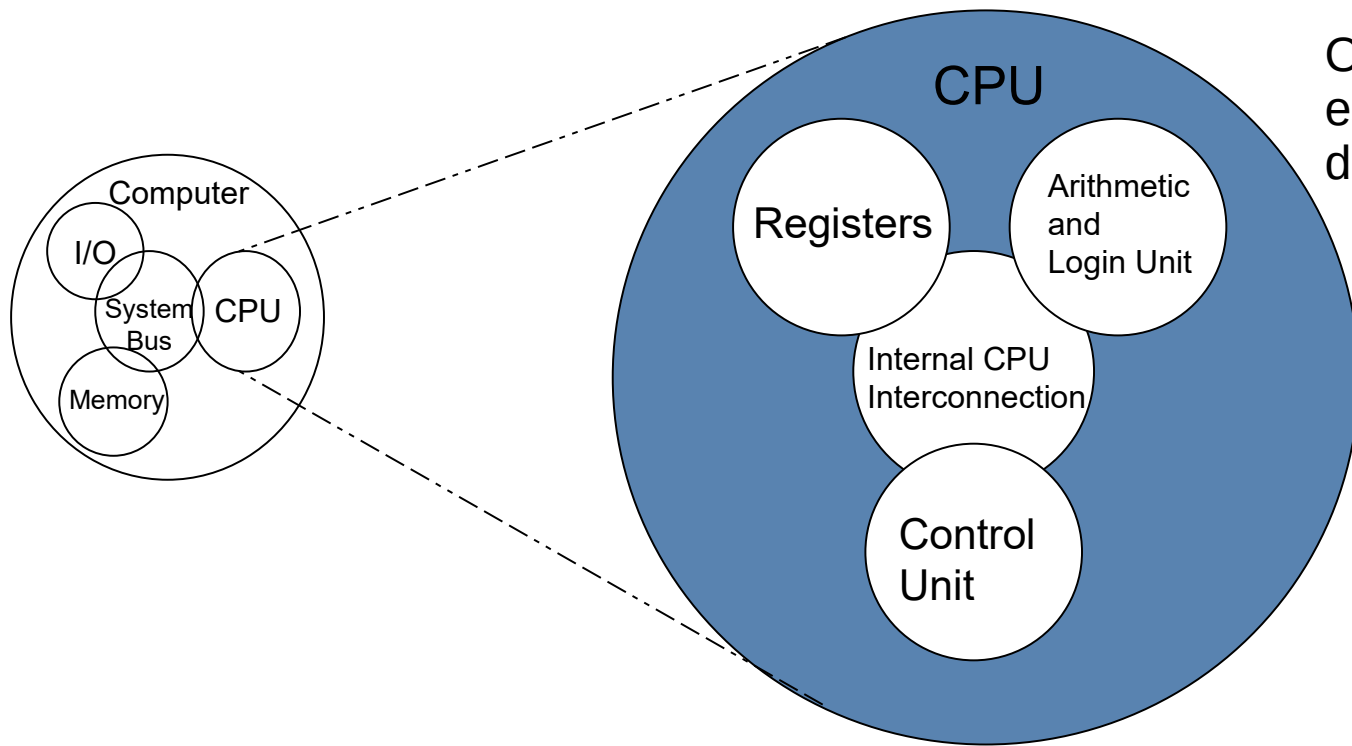
# Estrutura Interna



Estrutura Interna:

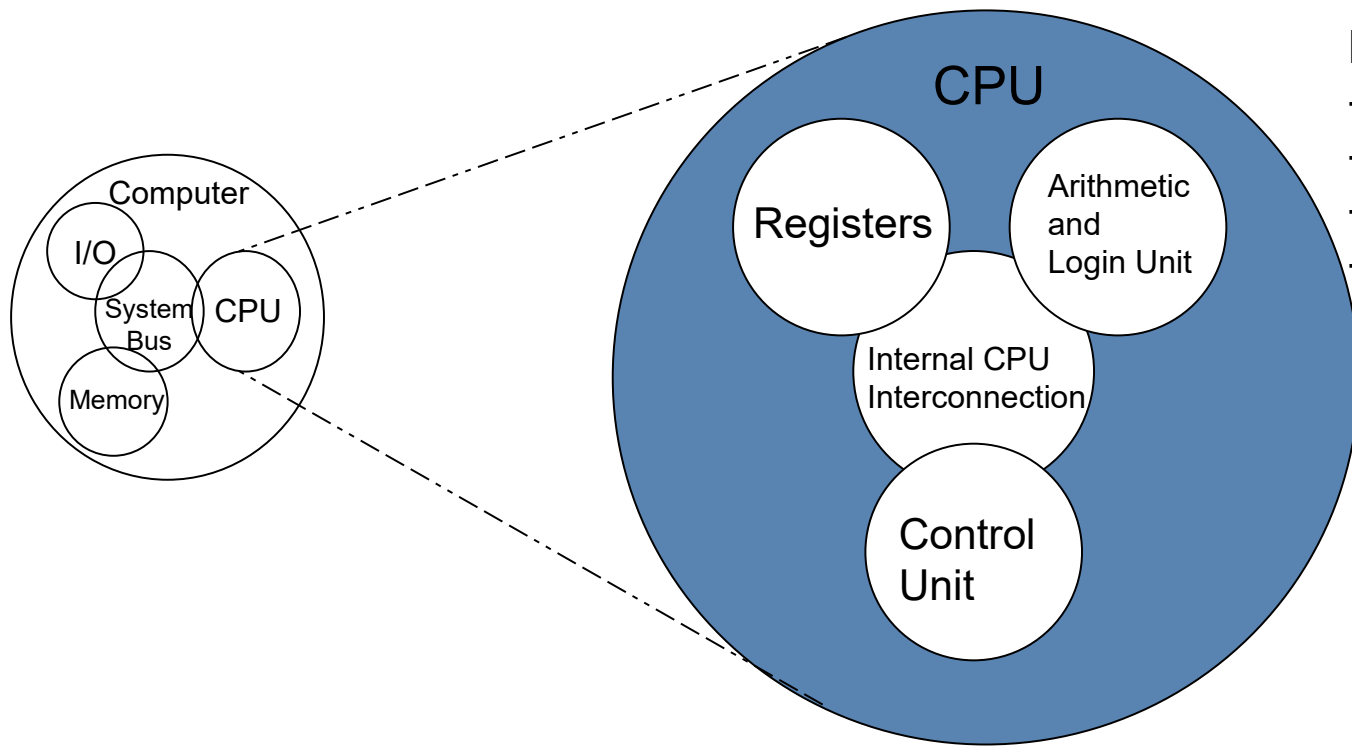
- Unidade Central de Processamento (CPU)
- Memória principal
- Entrada e saída
- Interconexão do sistema

# Estrutura Interna



O aprofundamento da hierarquia estrutural pode ser feito até o nível de interesse

# Estrutura Interna

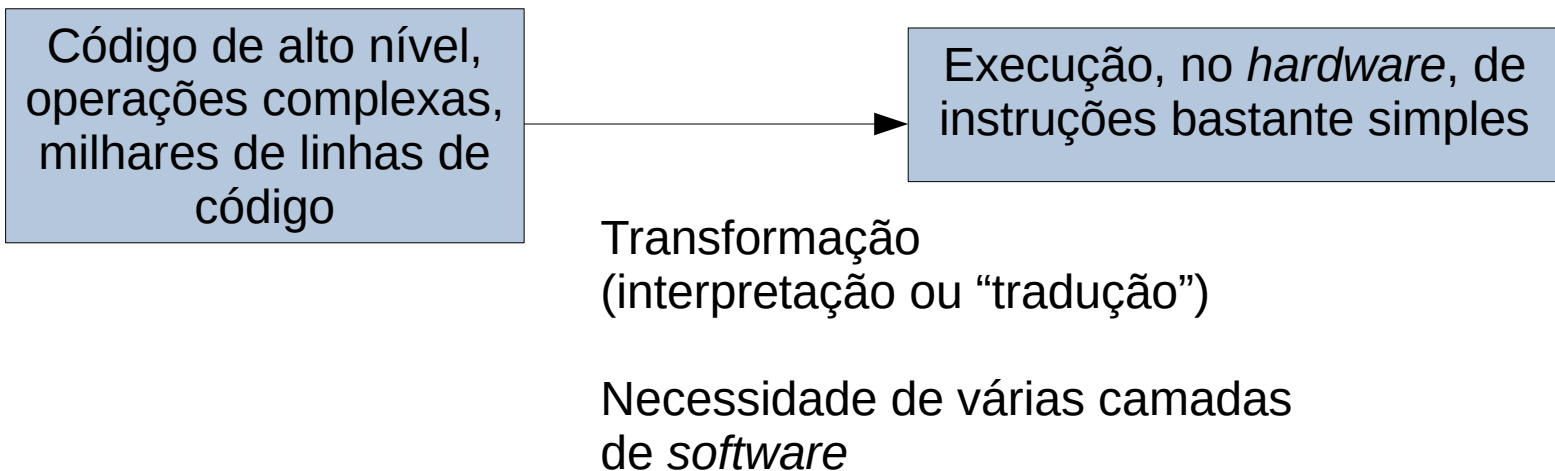


Blocos estruturais da CPU:

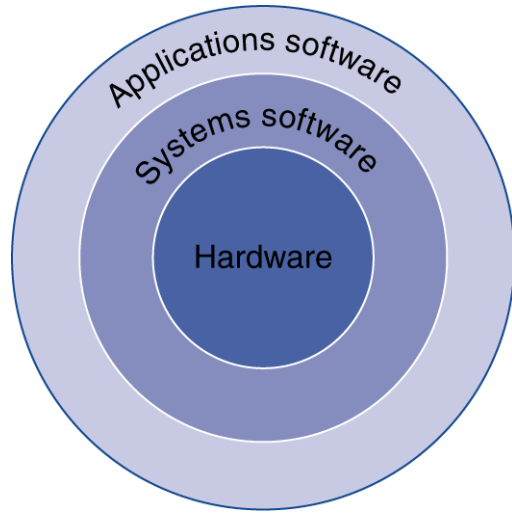
- Unidade de controle (UC)
- Unidade aritmética e lógica (ULA)
- Registradores
- Interconexão da CPU

# Visão Abstrata do Computador

Abstração da complexidade dos sistemas



# Visão Abstrata



Camada de aplicações (*softwares* aplicativos)  
- Definição das aplicações de usuários

Camada do *software* de sistema  
- *Software* de sistema → fornece serviços úteis para o sistema (computador)

- Sistema operacional → programa de supervisão que gerencia os **recursos** de um computador para o benefício dos programas executados nessa máquina
- Compilador → programa que traduz as instruções de linguagem *assembly*

# Visão Abstrata

- Padrões de funcionamento dos circuitos eletrônicos do computador: *bit* (estados de presença, ou não, de corrente elétrica nos terminais)
- Combinação de *bits* → comando específico ou alguma informação
- Programação arcaica → especificação de uma sequência de palavras binárias
  - Erros
  - Complexidade
  - Falta de produtividade

# Visão Abstrata

- Desenvolvimento de programas **montadores**
  - Traduzir uma linguagem simbólica, de **montagem** (*assembly*) para o padrão de bits (ou **linguagem de máquina**)
- Avanço da tecnologia: criação de linguagem de programação de alto nível
  - **Compiladores** também tiveram que ser desenvolvidos



# Visão Abstrata

High-level  
language  
program  
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compiladores  
transformam código de  
alto nível em *assembly*

Compiler

Assembly  
language  
program  
(for MIPS)

```
swap:
  muli $2, $5, 4
  add  $2, $4, $2
  lw   $15, 0($2)
  lw   $16, 4($2)
  sw   $16, 0($2)
  sw   $15, 4($2)
  jr   $31
```

Montadores (*assemblers*)  
transformam código de  
baixo nível em linguagem  
de máquina

Assembler

Binary machine  
language  
program  
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
1000110011110010000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
000000111110000000000000000001000
```

# Recomendações

- Leitura do capítulo 1 do livro de Stallings
- Leitura do capítulo 1 do livro de Patterson
- Acessar referências no SIGAA

# Conceitos Fundamentais

DCA0104 – Arquitetura de Computadores

Diogo Pedrosa

[diogo@dca.ufrn.br](mailto:diogo@dca.ufrn.br)

DCA – CT – UFRN