

Aula 2 - Introdução à Arquitetura de Computadores

(04/nov)

Conteúdo

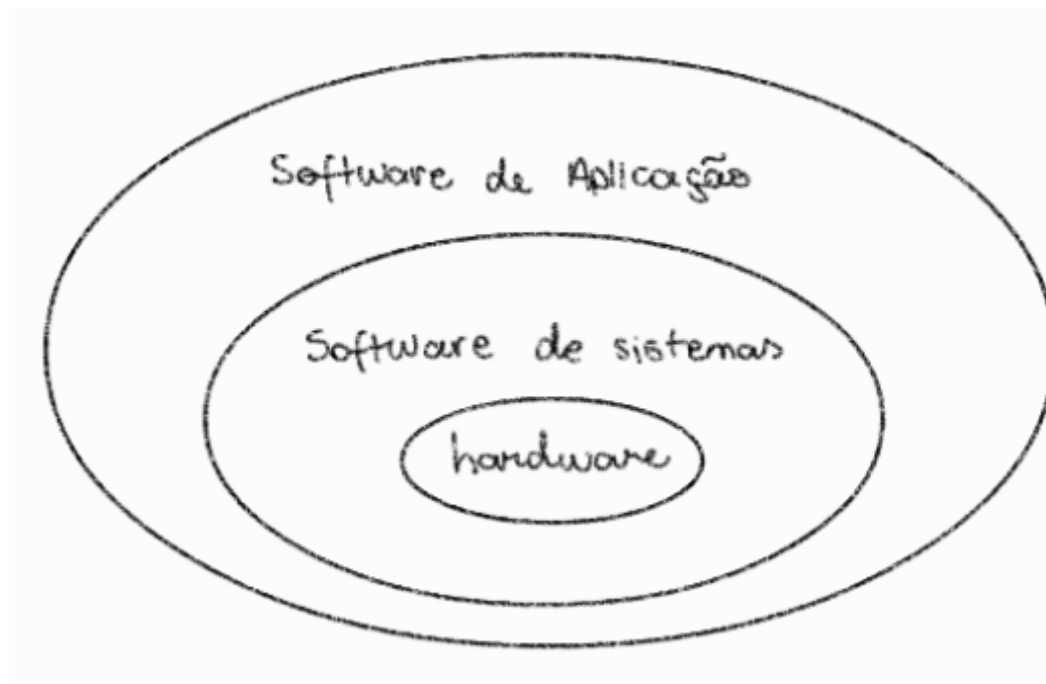
Arquitetura de computadores é o projeto estrutural de um computador, envolvendo os componentes físicos e lógicos essenciais ao seu funcionamento.

- Arquitetura de um sistema de hardware: "Operações que o sistema é capaz de executar". "Processo eletrônico que o computador físico utiliza para fazer uma operação".

Os componentes essenciais de um computador são divididos em 3 camadas:

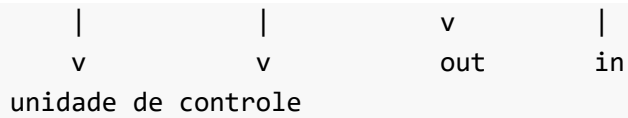
aplicação | sistema | hardware

- **Software de aplicação:** são aplicações típicas que os usuários utilizam - Ex: Banco de dados, sistemas de gestão, IDEs para desenvolvimento.
- **Software de sistema:** são programas que fornecem intermediários entre aplicação e hardware - Ex: Sistemas operacionais, compiladores e montadores.
- **Hardware:** é a parte física, executa apenas operações binárias. O hardware entende apenas sinais elétricos, que se restringe a ligado e desligado. deste modo, o alfabeto de um computador é composto de apenas duas letras: os dígitos binários ou bits (0 e 1)



Arquitetura de Von Neumann:

memória <-> unidade lógica de aritmética
^ ^ | ^



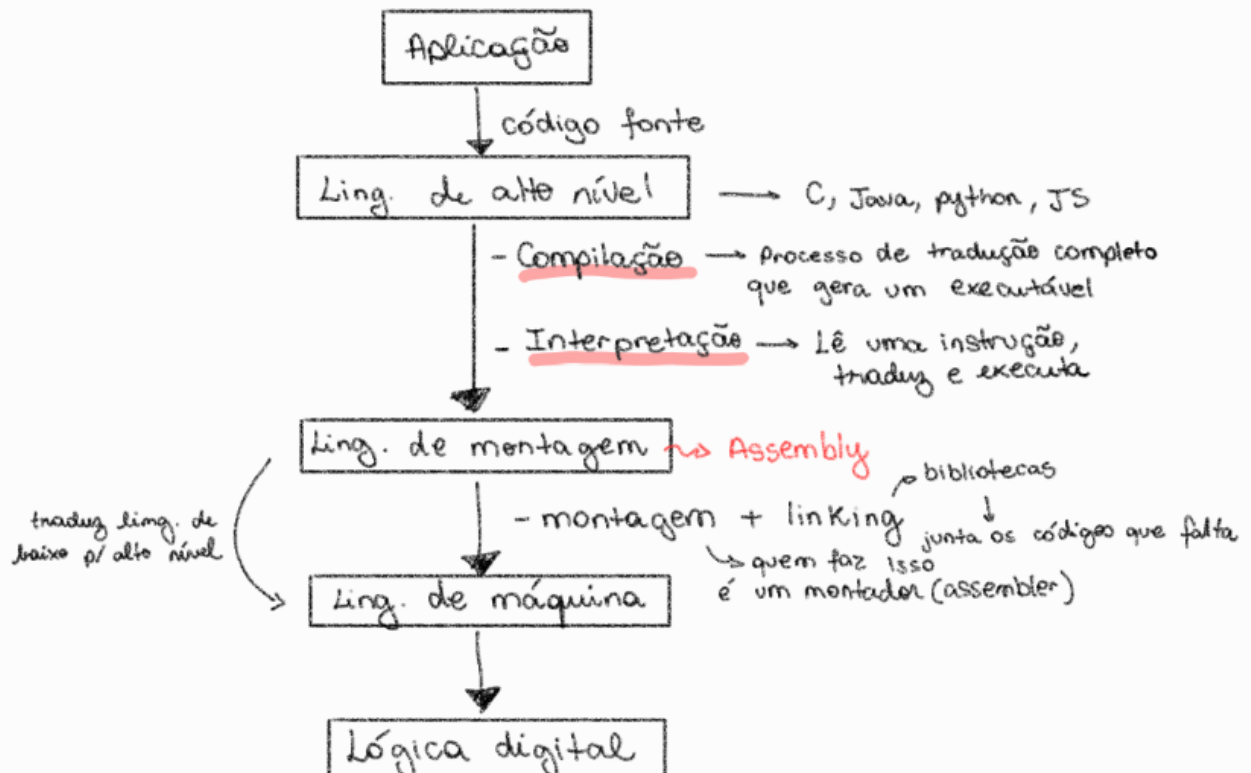
- Como a interação com o hardware usando código binário é ineficiente e improdutivo, foi-se criando linguagens que se assemelham mais à linguagem humana.
- Todavia, as primeiras linguagens eram muito arcaicas, e se pareciam muito com as operações que o hardware era capaz de executar. As linguagens de alto nível foram aparecendo com anos de desenvolvimento e aparecem até hoje.
- Elas tem por objetivo:

- 1 - Fazer com que o programador escreva algo mais próximo de sua linguagem.
- 2 - Aumentar a produtividade.
- 3 - Fazer com que programas sejam independentes da plataforma.

Tipos de computadores

- Celulares
- Iot
- Notebook
- Desktop
- Servidores
- Geladeiras e outros eletrodomésticos -> Sistemas embarcados
- Computadores de bordo

Níveis de código



```

Aplicação
  ↓ código fonte
Linguagem de alto nível
  ↓ compilador
Linguagem de montagem (assembly)
  ↓ montador (assembler)
Sistema operacional
  ↓ linker
Linguagem de máquina
  ↓
Microprograma
  ↓
Processador (Lógica Digital)
  
```

- Linguagem de alto nível -> Mais abstrata, perto da linguagem do ser humano

Observação - PROVA

1. O assembler (montador) faz a tradução do assembly (linguagem de montagem) para a linguagem de máquina (binário). Essa tradução é 1 para 1.
2. O computador/interpretador faz a tradução da linguagem de alto nível para linguagem de montagem (assembly). Essa tradução é 1 para n → Traduzir assembly para alto nível pode não voltar ao código original.

GCC - Tem 4 níveis de compilação (O1, O2, O3, O4)

```
codigo.c  
gcc -O3 codigo.c
```

Linguagem de montagem

- Representação textual de instruções (assembly)
- Esta relacionada com uma arquitetura e é chamada de Instruction Set Architecture (ISA) ou Arquitetura do Conjunto de INstruções ---> A interface entre hardware/software
- Cada arquitetura possui uma linguagem de montagem
- **Exemplos** : x86, x64, ARM, risc-v, mips, entre outras
 - Vamos estudar a arquitetura mips

Compilação x interpretação

Compilação

- É quando todo o código é traduzido de uma vez.
- ganha desempenho
- Ex: linguagem c, usa o gcc

Interpretação

- Código é traduzido por instrução, em tempo de execução;
- Pega uma linha, traduz, roda -> Pega linha traduz, executa -> ...
- Ex: Python

Misto:

- Compilado por máquina virtual
- Jvm interpreta para linguagem de máquina
- Ex: Java