

Caderno de InfraHard

Marconi Gomes

August 11, 2019

1 Introdução - Instruções e funcionamento básico

1.1 Abstração

As linguagens de programação podem ser divididas em **4 níveis**:

- Linguagem de Máquina (binário)
- Linguagem de montagem (Assembly)
- Linguagem de alto nível (Java, C++, etc)
- Linguagem de 4ª geração (PL/SQL, NATURAL, etc)

O menor nível de abstração que o programador pode ver antes do código de realmente chegar ao binário, chama-se Instruction Set Architecture (ISA), que é um **repositório de instruções**, ela é realmente a interface entre Software e Hardware. Ela vai me dizer quais as instruções e registradores que posso usar, como acessar a memória, etc.

1.2 Assembly

É uma linguagem que é dependente de arquitetura, ou seja, para cada tipo (x86, ARM) é um tipo de assembly diferente.

1.3 Compilador

Definição: é um programa que traduz de uma linguagem de mais alto nível (ex. Java) para uma de menor nível (assembly) que o computador entende.

A diferença entre um **compilador** e um **interpretador** é que o compilador traduz tudo primeiramente apenas e depois executa. O interpretador traduz e executa cada linha por vez.

Exemplos de linguagens compiladas (completamente): C, C++, etc.

Exemplos de linguagens interpretadas (completamente): JavaScript, Python.

Exemplos de linguagens semi-interpretadas e semi-compiladas: Java!

1.4 Visão funcional de um computador

Um computador pode (e deve) realizar 4 ações:

- Mover dados (Barramento)
- Controlar ações (CPU)
- Armazenar dados (Memória)

→ Processar dados (CPU)

A CPU faz sempre as seguintes coisas:

Busca → Decodificação → Execução

Os seguintes registradores são os mais comuns num computador:

PC (Program counter): Buscar o endereço da instrução

MAR (Memory Address Register): Guarda dinamicamente endereços que possam ser usados posteriormente.

IR (Instruction Register): Recebe a instrução do PC e a armazena.

AC (Accumulator): É um registrador comum genérico.

References