Linguagens de Programação

Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador. Esses programas instruem o computador a realizar determinadas tarefas específicas. Cada linguagem possui um conjunto único de palavras-chaves (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções dos programas.

Os programas de computador podem ser escritos em várias linguagens de programação, algumas diretamente compreensíveis pelo computador e outras que exigem passos de tradução intermediária. As linguagens de programação podem ser divididas em três tipos, com relação à sua similaridade com a linguagem humana:

- **Linguagem de máquina;**
- **Linguagem simbólica;**
- **Linguagem de alto nível.**

Linguagens de Programação

Linguagem de máquina (machine languale): é a linguagem de mais baixo nível de entendimento pelo ser humano e a única, na verdade, entendida pelo processador (UCP).

É constituída inteiramente de números, o que torna praticamente impossível entendê-la diretamente. Cada UCP tem seu conjunto único de instruções que definem sua linguagem de máquina, estabelecido pelo fabricante do chip.

Uma instrução típica em linguagem de máquina seria algo como:

0100 1111 1010

Essa linguagem é também classificada como uma linguagem de primeira geração.

Linguagens de Programação

Linguagem simbólica (assembly): é a linguagem de nível imediatamente acima da linguagem de máquina. Ela possui a mesma estrutura e conjunto de instruções que a linguagem de máquina, porém permite que o programador utilize nomes (chamados mnemônicos) e símbolos em lugar de números. A linguagem simbólica é também única para cada tipo de UCP, de forma que um programa escrito em linguagem simbólica para uma UCP poderá não ser executado em outra UCP de uma família diferente.

Nos primórdios da programação todos os programas eram escritos nessa linguagem.

Linguagens de Programação

Hoje a linguagem simbólica, é utilizada quando a velocidade de execução ou o tamanho do programa executável gerado são essenciais. A conversão da linguagem simbólica para a linguagem de máquina se chama montagem, e é feito por um programa chamado montador (ou assembler). Uma típica instrução em linguagem simbólica seria:

ADD A,B.

Essa linguagem é também classificada como linguagem de segunda geração, e, assim como a linguagem de máquina, é considerada uma linguagem de baixo nível.

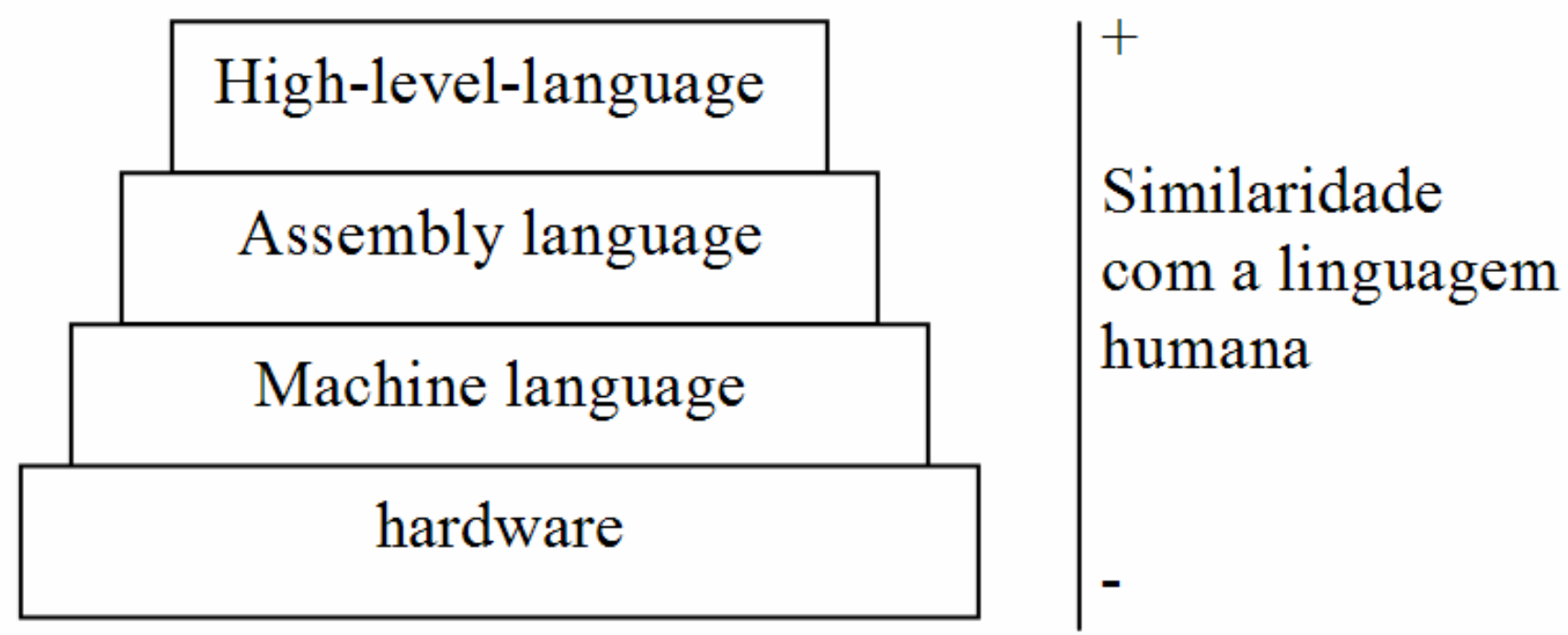
Linguagens de Programação

Linguagem de alto nível: São as linguagens de programação que possuem uma estrutura e palavras-chave que são mais próximas da linguagem humana. Tornando os programas mais fáceis de serem lidos e escritos. Esta é a sua principal vantagem sobre as linguagens de nível mais baixo. Os programas escritos nessas linguagens são convertidos para a linguagem de baixo nível através de um programa denominado compilador ou de um interpretador.

Uma instrução típica de uma linguagem de alto nível é:

```
if (A>10) then A:=A-7;
```

Linguagens de Programação



Breve histórico de “C”

- Criada por Dennis Ritchie;
- Em 1972;
- Centro de Pesquisas da Bell Laboratories;
- Para utilização no S.O. UNIX;
- O C é uma linguagem de propósito geral.

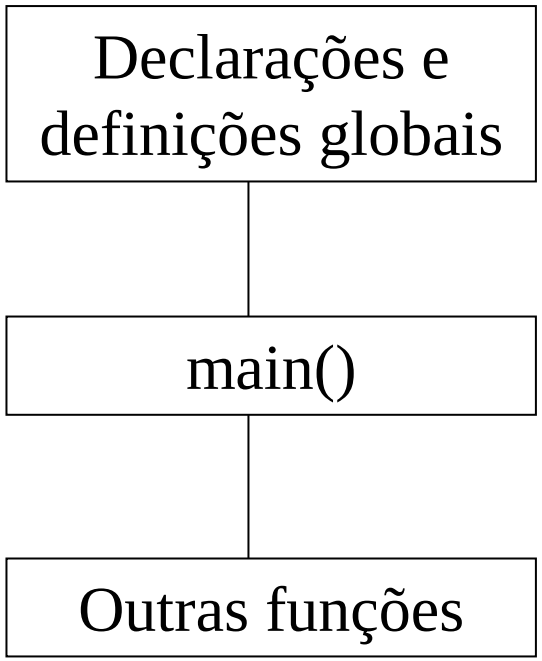
Características básicas da linguagem

- Case sensitive;
- Tipos de dados primitivos: caractere, inteiro e real;
- Possui estruturas de controle de fluxo para viabilizar a programação estrurada;
- Operadores aritiméticos, lógicos, relacionais, condicionais, bit a bit, de entrada e saída;
- Todo programa tem uma função chamada main();
- Todo linha do programa termina com ";".

ANSI

Em 1983, o **Instituto Norte-Americano de Padrões** (**ANSI**) formou um comitê, X3j11, para estabelecer uma especificação do padrão da linguagem C. O padrão foi completo em 1989 e ratificado como ANSI X3.159-1989 “Programming Language C” (**C ANSI**).

Estrutura de um programa em C



Conceitos Básicos – Linguagem C

Constantes

Exemplos:

- Decimal (10, -23768)
- Hexadecimal (0x12,0x1fea28)
- Octal (0123)
- Real (2.34, 2.34E+05, 2.14E-9)
- Caractere (‘a’,’%’)

Palavras-reservadas

- Palavras Reservadas

asm	const	else	for	Near	sizeof	union
auto	continue	enum	goto	Register	static	unsigned
break	default	extern	if	Return	struct	void
case	do	far	int	Short	switch	volatile
char	double	float	long	Signed	typedef	while

- Comentários
  - // ccccccccccccccccccccccccc
  - /* ccccccccccccccccccccccccccccccccccccccccccccccccc
    ccccccccccccccccccccccccccccccccccccccccccccccccc*/

Tipos Primitivos

- **Caractere**
  - Definido por char;
  - Ocupa 8 bits (1 byte)
  - Faixa de valores: -128 à 127
  - Exemplo: char letra;

    letra = ‘A’;

Tipos Primitivos

- **Inteiro**
  - Definido por int;
  - Ocupa 16 bits (2 bytes)
  - Faixa de valores: -32768 à 32767
  - Exemplo: int num;

    num = -73;

Tipos Primitivos

- **Ponto flutuante e ponto flutuante de precisão dupla**
  - float → 4 bytes
  - double → 8 bytes
  - faixa mínima de um valor em ponto flutuante
    - 1E-37 a 1E+37
  - Exemplo: float a,b,c=2.34;

    double x=2.38,y=3.1415,z;

Modificadores deTipos

- **signed**
- **unsigned**
- **long**
- **short**

Exemplo: unsigned char letra;
long int numero1, numero2;

Tipos de dados definidos no padrão ANSI

Tipo	Tamanho aproximado em bits	Faixa mínima
char	8	-127 a 127
unsigned char	8	0 a 255
signed char	8	-127 a 127
int	16	-32.767 a 32.767
unsigned int	16	0 a 65.535
signed int	16	O mesmo que int
short int	16	O mesmo que int
unsigned short int	16	0 a 65.535
signed short int	16	O mesmo que short int
long int	32	-2.147.483.647 a 2.147.483.647
signed long int	32	O mesmo que long int
unsigned long int	32	0 a 4.294.967.295
float	32	Seis dígitos de precisão
double	64	Dez dígitos de precisão
long double	80	Dez dígitos de precisão

Tipos

```
#include <stdio.h>
main()
{
    char c;
    int i;
    short int si;
    unsigned int ui;
    long int li;
    float f;
    double d;
    printf("char %d \n",sizeof(c));
    printf("int %d \n",sizeof(i));
    printf("short int %d \n",sizeof(si));
    printf("unsigned int %d \n",sizeof(ui));
    printf("long int %d \n",sizeof(li));
    printf("float %d \n",sizeof(f));
    printf("double %d \n",sizeof(d));
    printf("double %d",sizeof(double));
}
```