

Linguagens de Alto Nível

Fundamentos

Introdução

- 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-chave (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções (instruções mesmo) dos programas.

Linguagens Programação

Classificação

- As linguagens de programação podem ser classificadas, em uma escala relativa à sua semelhança com a linguagem humana, em:
 - Linguagem de Máquina
 - Linguagem Assembly
 - Linguagem de Alto Nível
 - Linguagem de Quarta Geração

Linguagem de Máquina

- É a linguagem de mais baixo nível de entendimento pelo ser humano e a única, na verdade, entendida pelo processador (CPU).
- É constituída inteiramente de números, o que torna praticamente impossível entendê-la diretamente.
- Cada CPU tem seu conjunto único de linguagem de máquina, definido pelo fabricante do chip.
- Uma instrução típica em linguagem de máquina seria algo como:

0100 1111 1010 0110

Linguagem de Máquina

- As instruções presentes na linguagem de máquina são as mesmas da linguagem do nível mais acima (linguagem assembly).
- Os programas escritos nas linguagens de mais alto nível são convertidos (compilados ou montados) para a linguagem de máquina específica, para que possam ser executados pelo computador.

Linguagem de Máquina

- Um programa em linguagem de alto nível que foi compilado para executar em determinada CPU precisa ser recompilado (e muitas vezes reescrito), para que possa ser executado em outra CPU que não tenha o mesmo conjunto de instruções.
- Essa linguagem é também classificada como linguagem de primeira geração

Linguagem 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 dos números.
- A linguagem assembly é também única para cada tipo de CPU, de forma que um programa escrito em linguagem assembly para uma CPU poderá não ser executado em outra CPU de uma família diferente.

Linguagem Assembly

- Nos primórdios da programação todos os programas eram escritos nessa linguagem. Hoje, a linguagem assembly, é utilizada quando a velocidade de execução ou o tamanho do programa executável gerado são essenciais.
- Atualmente a maioria dos programas é escrita em linguagens de alto nível, tal como C, Pascal , Delphi, Kylix, Visual Basic, etc. devido à facilidade de criação e manutenção dos programas. Todos os programas escritos nessas linguagens são convertidos para a linguagem de máquina para serem executados pelo processador.

Linguagem Assembly

- A conversão da linguagem assembly para a linguagem de máquina se chama **montagem**, e é feita por um programa chamado montador (ou **assembler**).
- Uma típica instrução em assembly seria :

```
ORG 100  
LDAA # $10  
END
```
- Essa linguagem é também classificada como linguagem de segunda geração, e, assim como a linguagem de máquina, é considerada uma **linguagem de baixo nível**.

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 máquina através de um **programa compilador** ou de um **interpretador**.
- Exemplos: FORTRAN, Prolog, Basic, COBOL, C, Pascal, Delphi, C++, Java, etc.

Linguagem de Quarta Geração

- Usualmente abreviada por 4GL.
- São linguagens de programação com estrutura mais próxima da linguagem humana do que as linguagens de programação de alto nível.
- A maioria delas é usada para acessar bancos de dados, a SQL (Structured Query Language) é um bom exemplo dessa geração.
- Por exemplo, um comando típico seria:

```
SELECT NOME,MATRICULA FROM ALUNOS WHERE  
NOME = "Carlos" AND ESTADO IN ("RJ","SP","MG")
```

Paradigmas de Programação

- Paradigmas de programação refere-se às diferentes formas e estilos de se organizar a lógica de um programa de computador.
 - Os principais paradigmas são:
 - Programação procedural (Imperativa)
 - Programação Funcional
 - Programação Orientada a Objetos
 - Programação Lógica
 - Programação Concorrente

Programação Procedural

- As linguagens procedurais são orientadas a ações; isto é, as instruções da programação são visualizadas como uma seqüência de ações.
- Por exemplo, comandos de atribuição, comparação, etc.
- Exemplos: Pascal e C
- Aplicações:
 - Computação Científica
 - Desenvolvimento de Aplicativos

Programação Funcional

- As linguagens funcionais se caracterizam pela presença de estruturas que permitem a recursividade (valor de uma expressão depende apenas dos valores de suas sub-expressões), coleta de lixo automática e gerenciamento automático das alocações de valores.
- Exemplos: Lisp, Haskell, Sloth e Scheme
- Aplicações:
 - Inteligência Artificial
 - Sistemas Especialistas

Programação Orientada a Objetos

- As linguagens orientadas a objetos tratam todas as fases de um programa como uma coleção de objetos que interagem entre si.
- Os objetos, geralmente, são organizados em uma hierarquia de classes.
- Exemplos: C++, Smalltalk, Java e Eiffel
- Aplicações:
 - Desenvolvimento de Sistemas Complexos

Programação Lógica

- As linguagens lógicas lidam com relações que constituídas por fatos e regras.
- A partir desses elementos, é possível deduzir as respostas de um determinado sistema.
- Exemplos: Prolog
- Aplicações:
 - Processamento em linguagem natural
 - Sistemas especialistas

Programação Concorrente

- As linguagens concorrentes são aquelas cujas operações podem ser executadas em paralelo.
- Exemplos: ADA
- Aplicações:
 - Sistemas de tempo real

Linguagens Estruturadas e Não-Estruturadas

- Na programação estruturada não se utiliza os comandos de desvio incondicional (GOTO <label>)
- A execução de um programa estruturado é feita sempre para adiante, sem desvios e sem retornos, pois a seqüência lógica das instruções coincide com a ordem física em que elas são escritas
- Características de um programa estruturado:
 - Utilização massiva de sub-rotinas (procedimentos e funções)
 - Facilidade de manutenção
 - Fácil entendimento

Interpretadores e Compiladores

- Interpretadores e Compiladores são programas que operam sobre programas escritos pelo programador (**programa fonte**).
- **Interpretador**: programa que lê o programa fonte e executa as suas instruções linha após linha.
 - Dessa forma, o interpretador deve sempre estar presente e ativo durante a execução de um programa;
 - Não há geração de código-objeto (executável)
- **Compilador**: programa que lê todo o programa fonte e o converte para um código-objeto, que é uma tradução do código fonte em uma forma que pode ser diretamente executada pelo processador.
 - Há geração de código-objeto
 - Após a geração do código não há mais a necessidade do compilador;
 - Desvantagem: tempo de compilação do programa

Máquinas Multiníveis Contemporâneas

- As linguagens de programação atuais são utilizadas, normalmente, em máquinas consideradas multiníveis.
- A maioria dos computadores modernos possuem 6 níveis.
- **Nível 0:** É o nível de lógica digital. É o hardware verdadeiro da máquina (portas lógicas). Seus circuitos executam os programas em linguagens de máquina.
- **Nível 1:** É o nível de microprogramação. Ao contrário do nível 0, onde não existe o conceito de programa como uma seqüência de instruções a serem executadas, nesse nível há definitivamente um programa denominado microprograma cuja função é interpretar as instruções de nível 2.
 - **Microprograma:** programa que controla os registradores, os barramentos, ALUs, as memórias da máquina, etc.

Máquinas Multiníveis Contemporâneas

- **Nível 2:** É o nível convencional de máquina. Nesse nível a máquina tem uma memória orientada por bytes ou palavras e instruções como MOV, JUMP e ADD.
- **Nível 3:** É o nível de Sistema Operacional. O Sistema operacional pode ser considerado como um interpretador para certas características não encontradas no nível 2. As mais importantes delas são as instruções de E/S, as facilidades para o processamento compartilhado e a memória virtual.
- **Nível 4:** É o nível de linguagem de montagem. É implementado por tradução e corresponde a uma representação simbólica de algum programa em linguagem de máquina.
- **Nível 5:** É o nível de linguagem orientada a problemas.

Portabilidade de Programas

- Um programa que pode ser transferido de um computador para outro, com relativamente pouco esforço é dito ser **portável**.
- Algumas das dificuldades para a portabilidade de programas são:
 - O comprimento das palavras variam de máquina para máquina;
 - O número e tipos de dispositivos de E/S diferem de máquina para máquina;
 - Algumas instruções podem depender do sistema operacional;
 - O programa pode ser grande demais para a memória da nova máquina;
 - Os conjuntos de caracteres podem ser incompatíveis (um certo padrão de bits pode representar um caractere imprimível em um computador e ser ilegal ou uma marca de fim de registro em outro)