

Linguagem de Programação

O meio mais eficaz de comunicação entre as pessoas é a linguagem (língua ou idioma). Na programação de computadores, uma linguagem de programação serve como meio de comunicação entre o indivíduo que deseja resolver um determinado problema e o computador escolhido para ajudá-lo na solução. A linguagem de programação deve fazer a ligação entre o pensamento humano (muitas vezes de natureza não estruturada) e a precisão requerida para o processamento pela máquina.

O desenvolvimento de um programa torna-se mais fácil se a linguagem de programação em uso estiver próxima ao problema a ser resolvido. Isto é, se a linguagem inclui construções que refletem a terminologia e/ou os elementos usados na descrição do problema. Tal tipo de linguagem de programação é considerada de alto nível. Os computadores digitais, por sua vez, aceitam e entendem somente sua própria linguagem de máquina (dita de baixo nível), a qual consiste tipicamente de seqüências de zeros e uns. Esse tipo de linguagem é bem diferente da linguagem de alto nível usada para descrever um problema em uma dada área de aplicação.

Atualmente as linguagens de programação mais utilizadas são aquelas classificadas como de alto nível, consideradas mais próximas as linguagens naturais ou ao domínio da aplicação em questão.

A evolução da linguagens de programação (LPs)

1ª Geração

Os primeiros computadores eram programados em linguagem de máquina, em notação binária. Cada instrução de máquina é, em geral, formada por um código de operação e um ou dois endereços de registradores ou de memória. As linguagens de máquina permitem a comunicação direta com o computador em termos de bits, registradores e operações de máquina bastante primitivas. Como um programa escrito em linguagem de máquina nada mais é do que uma sequência de zeros e uns, a programação de um algoritmo usando este tipo de linguagem é complexa, consativa e fortemente sujeita a erros.

2ª Geração

Compreende as linguagens simbólicas ou de montagem (*Assembly*), projetadas para minimizar as dificuldades da programação em notação binária. Códigos de operações e endereços binários foram substituídos por mnemônicos. Neste tipo de linguagem, a maioria das instruções são representações simbólicas de instruções de máquina. Por isso, o processamento de um programa em *Assembly* requer tradução para linguagem de máquina antes de ser executado. Essa tradução é feita por um programa especial chamado montador (que aceita como entrada uma representação textual de algoritmo em *Assembly* e produz uma representação do mesmo algoritmo em linguagem de máquina), e é quase sempre direta, não envolvendo muito complexidade.

Exemplo de instruções em *Assembly*:

MOV A, B

Mnemônico MOV (abreviação de MOVE) e dois registradores como parâmetro: A e B. Quando o processador executa essa intrução (depois de traduzida para linguagem de máquina pelo montador), ele comanda o movimento do conteúdo do registrador B para o registrador A.

ADD A, B

Mnemônico ADD (abreviação de ADDITION) e os registradores A e B como parâmetro. Quando o processador executa essa instrução ele comanda a adição do conteúdo do registrador B com o conteúdo do registrador A e o resultado é armazenado em A.

Com isso podemos observar que a linguagem *Assembly* é apenas uma versão legível da linguagem de máquina. Considerando que cada tipo de processador entende um tipo de linguagem de máquina (aquela decodificada pela UC), e sabendo que as linguagens *Assembly* são apenas uma tradução da linguagem de máquina, então um programa escrito em linguagem *Assembly* para determinada família de processadores somente poderá ser entendido dentro da mesma família.

As linguagens da 1ª e 2ª geração são chamadas de linguagem de baixo nível. Com a evolução das máquinas e a necessidade de se fazer programas cada vez mais complexos e legíveis, as linguagens de programação foram se aproximando cada vez mais das linguagens naturais dando origem às linguagens de alto nível (a partir da 3ª geração).

3ª Geração

As linguagens de programação desta geração já apresentam uma sintaxe mais próxima da linguagem natural, fazendo uso de palavras reservadas extraídas do vocabulário corrente (como READ, WRITE, TYPE, etc) e permitem a manipulação de dados nas mais diversas formas (números inteiros, reais, caracteres, etc), enquanto a linguagem *Assembly* trabalha com bits, bytes, palavras armazenadas em memória.

Estas linguagens surgiram na década de 60. Algumas delas são orientadas a soluções de problemas científicos, tais como FORTRAN, PASCAL e ALGOL; outras, tal como COBOL, são usadas para aplicações comerciais. Linguagens como PL/I e ADA contém facilidades para ambos os tipos de computações (comercial e científica).

As linguagens de 3ª geração podem ser classificadas em:

- Linguagens Procedimentais: onde um programa especifica um procedimento, isto é, uma sequência de passos a serem seguidos para solucionar o problema. Este tipo de linguagem normalmente oferece três classes de instruções: instruções de E/S, instruções de cálculos aritméticos ou lógicos e instruções de controle de fluxo lógico (desvios condicionais, incondicionais e processamento iterativo). Exemplos de linguagens procedimentais: BASIC, ALGOL, PASCAL, ADA, C, etc.

- Linguagens Declarativas: que se dividem em duas classes:

- * Linguagens Funcionais : as quais se baseiam na teoria das funções recursivas. A programação funcional envolve, essencialmente a definição e a chamada de funções. Exemplo de linguagem funcional: LISP;

- * Linguagens Lógicas: cuja a base é a lógica matemática. Um programa declara fatos (dados e relações entre eles) e cláusulas lógicas (regras de dedução), que permitirão deduzir novas verdades a partir de fatos conhecidos. Exemplo de linguagem lógica: PROLOG.

A passagem de um programa escrito em linguagem de alto nível para um programa em linguagem de máquina é bem mais complexa se comparada a linguagem *Assembly*. Esta passagem é feita por programas especiais chamados compiladores. Compiladores aceitam como entrada uma representação textual da solução do problema expresso em linguagem de alto nível (chamada de linguagem fonte), e

produz uma representação do mesmo algoritmo expresso em linguagem de máquina (chamada de linguagem objeto).

Um programa escrito em linguagem de alto nível teoricamente pode ser usado em qualquer máquina, bastando escolher o programa correspondente para aquela máquina e recompilar o programa fonte.

As linguagens de 3ª geração foram projetadas para profissionais de processamento de dados e não para usuários finais. Encontrar e arrumar erros em programas (debugar) escritos nessas linguagens ainda consome tempo, e a modificação de sistemas complexos é relativamente difícil. As linguagens de 4ª geração foram projetadas em resposta a esses problemas.

4ª Geração

Os programas escritos nessas linguagens necessitam de menor número de linhas de código quando comparados a programas correspondentes em linguagens de programação convencionais. Além dos comandos disponíveis nas linguagens de 3ª geração, estas linguagens empregam outros mecanismos, como preenchimento de formulários, interação via vídeo (menus), e auxílio para construção de gráficos.

As linguagens dessa geração variam bastante no número de facilidades oferecidas aos usuários. Algumas delas são, meramente, geradores de relatórios ou pacotes gráficos; outras são capazes de gerar aplicações completas. Em geral, essas linguagens são projetadas para atender a classe específicas de aplicações. Exemplo de linguagens de 4ª geração: LOTUS 1-2-3, EXCEL, SQL, entre outras.