

Estrutura da Linguagem C

Um programa escrito em C é composto por unidades que são as *expressões*, *sentenças* e *blocos*. Neste capítulo descreveremos cada um destes conceitos para preparar para o estudo da sintaxe de programas escrito em C.

De forma simplificada, pode-se dizer que expressões realizam operações sobre dados. Já uma sentença é formada por uma ou mais expressões. Um bloco agrupa sentenças.

Expressões e Operadores

A **expressão** é a unidade indivisível de construção de um programa em C.

Muitas expressões em C possuem aparência muito semelhante a fórmulas aritméticas. Muitas vezes, este fato leva os novatos a entender erroneamente que um programa em C é um conjunto de fórmulas matemáticas.

Expressões definem como realizar operações sobre dados.

É importante entender como C é diferente da matemática. Veja o exemplo a seguir.

Na matemática, quando escrevemos $b = a$, estamos afirmando que as variáveis a e b sempre têm o mesmo valor. Ao atribuir um novo valor para a , para manter a igualdade válida, a variável b assume imediatamente o mesmo valor.

A principal diferença para C é que a execução de uma expressão não pode alterar o resultado de uma outra atribuição, executada em um momento anterior. Em C, quando escrevemos $b = a$, estamos atribuindo a b uma cópia do valor corrente de a . Ao atribuir um novo valor para a , o programa não lembra mais de $b = a$, e a variável b permanece inalterada, contendo o valor anterior de a .

Sentenças

Sentença é uma unidade completa em C. A sentença pode ser:

- Declaração ou definição (ex: variáveis)
- Comando primitivo de C (ex: atribuição ou `=` ou `return`)
- Chamada à rotinas, tais como rotinas de leitura ou escritas (ex: `printf`, `scanf`)
- Controle de fluxo de execução em C (ex.: `if`, `while`)

Todas as sentenças são terminadas com o ponto-e-vírgula!

O ponto e vírgula é justamente o símbolo que delimita uma sentença da outra. Normalmente, por motivos de clareza de código, costuma-se também escrever uma sentença por linha.

No exemplo abaixo, cada linha é uma sentença. Por este motivo, elas terminam com um ponto-e-vírgula:

```
float nota1, nota2;
```

```

float media;

printf("Digite as duas notas: ");
scanf("%f", &nota1, &nota2);

media = (nota1 + nota2) / 2;
printf("Média: %f", media);
return 0;

```

As duas primeiras linhas são declarações; as demais são chamadas de rotinas (linhas 3, 4, e 6). A linha 5 contém uma expressão matemática e uma atribuição, e a linha 7 é um comando primitivo de C.

Sentenças são executadas uma de cada vez, de forma independente uma da outra.

Uma sentença corresponde a um passo do algoritmo ou a uma declaração.

Se o código fonte descreve um algoritmo, então sentenças correspondem a passos do algoritmo, se não forem declarações. Aqui, obviamente, queremos dizer passos elementares do algoritmo *na linguagem C*. Conforme já sabemos, o compilador traduzirá cada um desses passos elementares em várias instruções de máquina.

Blocos

Um bloco é um conjunto de sentenças que estão agrupadas entre os símbolos de chaves ({ e }). Um bloco também pode conter outro bloco. Dessa forma, podemos ter blocos aninhados.

O caso típico de um bloco pode ser encontrado no programa principal:

```

int main(int argc, char* argv[]) {
    float nota1, nota2;
    float media;

    printf("Digite as duas notas: ");
    scanf("%f", &nota1, &nota2);

    media = (nota1 + nota2) / 2;
    printf("Média: %f", media);
    return 0;
}

```

As sentenças que pertencem a um mesmo bloco estão atreladas ao mesmo fluxo de execução. Ou seja, todas elas são executadas em seqüência, uma após a outra, obedecendo-se eventuais desvios.