

Lógica de Programação e Algoritmos em C

Introdução

Nos laboratórios da empresa americana: “**Bell Labs**”, nos anos 70, **Dennis M. Ritchie** e **Ken Thompson** criaram a linguagem C. Com origem na linguagem B de Thompson, que foi uma evolução da linguagem BCPL. B era a primeira letra de BCPL e C a segunda, portanto os autores acharam que seria lógico chamar a linguagem de C.

A estrutura modular da linguagem C resulta em programas mais legíveis e documentados. Quando utilizamos as técnicas de OOP (Object-Oriented Programming - Programação Orientada a Objetos) estamos fazendo uso da linguagem C++.

A linguagem C é utilizada no desenvolvimento de sistemas operacionais e em diversos tipos de aplicações e tem influência direta nas linguagens C++, Java, C#, PHP e JavaScript.

Esta linguagem é reconhecida entre os programadores como uma linguagem de “**alto/baixo nível**”, ou seja, podemos trabalhar como uma linguagem próxima da linguagem humana fazendo operações do tipo: “ **$a=b+c;$** ” e também com operações de baixo nível, como por exemplo: ligando ou desligando portas no hardware do computador.

Como a utilização da linguagem C tornou-se muito popular foi criado um padrão para a linguagem, chamado de: “**Padrão ANSI**”. É chamado desta maneira porque “**ANSI**” é uma sigla que representa: “**American National Standards Institute**”, algo parecido com a sigla: “**ABNT**”, que no Brasil significa: “**Associação Brasileira de Normas Técnicas**”.

Antes do “**Padrão ANSI**” a única referência para a linguagem C era o livro: “**Linguagem de Programação C**”, de **Brian Kernighan** e **Dennis Ritchie**. Este livro causou algumas divergências entre compiladores da linguagem C, então o comitê ANSI teve que intervir para criar um padrão a ser utilizado por todos eles.

A linguagem C++ é uma sequência natural da linguagem C.

A linguagem C utiliza a técnica da programação estruturada, mas C++ agrega a técnica da programação orientada a objetos.

No início dos anos 80, **Bjarne Stroustrup**, da “**Bell Labs**”, foi quem desenvolveu a linguagem C++. Esta linguagem teve como objetivo fazer algumas simulações orientadas a eventos, mas a principal meta foi manter a compatibilidade com a linguagem C, para manterem protegidas as milhares de linhas de códigos fontes escritos em linguagem C. Esta é a razão principal de podermos utilizar os compiladores atuais para C++ e utilizar a estrutura da linguagem C e tudo funcionar corretamente.

As linguagens C e C++ são uma das melhores alternativas para entrar no mundo da programação de computadores.

O primeiro programa em C

O programa a seguir escreve na tela do computador a frase: **“Meu primeiro programa”**. Veja, abaixo, como é o código fonte do programa:

```
main( )  
{  
    printf(“Meu primeiro programa \n”);  
}
```

Um programa em C consiste de uma ou mais funções que especificam o que deve ser feito.

No exemplo acima **main()** é uma função.

Qualquer programa em C começa com a função **main()** que marca o ponto inicial da execução do programa.

Todas as instruções devem estar dentro de um bloco de comandos envoltos em uma chave que abre “{” e uma chave que fecha “}” .

A linha: **printf(“Meu primeiro programa”);** é uma chamada a função **printf**, com o argumento **“Meu primeiro programa”**. Esta sequência de caracteres entre aspas é chamada de: **cadeia de caracteres**, ou também **string**.

A palavra **printf** é uma função de biblioteca que escreve o argumento na tela do seu computador.

Os parênteses devem estar presentes mesmo quando não há argumentos.

As instruções C são sempre encerradas por um **ponto-e-vírgula (;)**.

Na execução do programa vemos que ele escreve a mensagem na tela: **“Meu primeiro programa”** e faz uma nova linha quando encontrou o código: **“\n”**.

Veja abaixo outros códigos que poderiam ser utilizados:

Códigos	Significado
\n	Nova linha
\t	Tabulação
\b	Retrocesso
\a	Alarme (sinal sonoro)
\\	Para a própria barra
\0	Zero (null)
\'	Apóstrofo
\"	Para aspas duplas

Detalhes importantes sobre os programas em C

Um programa de computador é uma tarefa que exige um bom preparo do programador.

Exige técnicas devidamente analisadas e testadas.

Exige cuidado e capricho na escrita do código fonte.

Uma das principais preocupações do programador, além de raciocínio lógico, é de fazer o programa funcionar, mas, existe ainda os detalhes que irão facilitar a manutenção do programa.

Vejamos alguns destes detalhes.

Identação do código fonte

Na tipografia, indentação seria o recuo do texto em relação à margem.

Entre os programadores é utilizado para definir a estrutura de um algoritmo.

É usado no código fonte da maioria das linguagens de programação, para ressaltar a estrutura do algoritmo e aumentar a sua legibilidade.

Se você escrever o programa desta maneira:

```
main()
{
    printf("Meu primeiro programa \n");
}
```

Ou escrever desta outra forma, não vai ocasionar erros de compilação, apenas ficará mais difícil de ler o código fonte e depois dar manutenção no programa.

```
main(){printf("Meu primeiro programa \n"); }
```

Quais programas acima você achou mais fácil de ler o código fonte?

Se a sua resposta foi o primeiro, é porque ele segue os padrões de indentação.

Case Sensitive

Dentro das aspas não haverá nenhum problema: **"meu primeiro programa"**, **"MEU PRIMEIRO PROGRAMA"** ou **"MeU PrImEiRo PrOgRaMa"**. Estas frases não irão produzir erros quando compilarmos o programa.

Mas, se você escrever o comando: **Printf**, **PRINTF** ou **pRiNtF**, não é a mesma coisa, pois a linguagem é do tipo **"case sensitive"**, isto é, diferencia as letras minúsculas das letras maiúsculas, portanto são comandos diferentes que não serão reconhecidos como fazendo parte da sintaxe da linguagem.

Comentários

É importante que possamos escrever alguns comentários sobre as linhas de comando ou sobre partes do programa.

Estes comentários não devem ser interpretados pelo compilador como uma linha de comando, ou seja, o compilador nem vai ler estas linhas.

Para fazer um comentário de várias linhas marcamos o início do comentário com um sinal de `/*` e quando terminamos o texto do comentário colocamos a marca de final de comentário com o sinal `*/`.