Introdução à programação em C

Este material foi produzido a partir de diversas fontes, entre elas as transparências do prof. Lucas Ferrari (UFPR) e o tutorial online Learn C Programming, entre outros.

Um breve histórico

A linguagem C foi criada por Dennis Ritchie em 1972 para reescrever de forma portável o sistema operacional UNIX, que antes era escrito em *assembly*. Sua estrutura e seu nome provêm de uma linguagem anterior B, que era uma simplificação da linguagem de programação BCPL, escrita em 1966.

Em 1978 foi publicado o famoso livro C Programming Language por Brian Kernighan e Dennis Ritchie, consagrando a linguagem no padrão chamado K&R. Durante muitos anos esse livro foi considerado como a especificação da linguagem.

Em 1989 o padrão C foi oficializado pelo ANSI (*American National Standards Institute*) através do padrão ANSI X3.159-1989, conhecido hoje como ANSI C, C89 ou C90.

Em 1999 a ISO (*International Standards Organization*) publicou um novo padrão da linguagem chamado ISO/IEC 9899:1999 e conhecido como C99. Existem padrões mais recentes (C11 e C17/C18), mas o C99 é o padrão mais usado atualmente.

Apesar de sua idade, C continua sendo uma das linguagens mais utilizadas hoje em dia (TIOBE Index, Ranking IEEE). A linguagem C inspirou a criação de muitas outras linguagens de programação mais recentes, como C++, Java, C#, JavaScript, Perl, PHP, Lua, etc.

Características da linguagem

Algumas das principais características da linguagem C são:

- linguagem procedural, modular e estruturada, com tipagem estática de dados.
- geralmente é compilada para o código de máquina da plataforma alvo, gerando código compacto, eficiente e sem necessidade de um *runtime* de execução sofisticado.
- Possui facilidades para acesso de baixo nível à memória, registradores e portas de E/S.
- Extremamente portável, pode executar em plataformas de microcontroladores a supercomputadores.
- É muito utilizada para escrever software de sistema, como sistemas operacionais, compiladores, serviços de rede, interfaces gráficas, bancos de dados, editores de texto, jogos, etc.
- A maior parte das funcionalidades da linguagem provém de vastas bibliotecas, como a biblioteca padrão C (LibC ou Standard C Library).

Exemplo de um programa em C

O código a seguir traz um `Hello World` em C:

hello.c

```
/* Este é um exemplo simples de código em C,
   que escreve a string 'Hello, world!' na
   saída padrão do terminal.              */
```

```c
#include <stdio.h>              // inclusão de definições

int main ()                     // função principal
{
  printf ("Hello, world!\n") ;  // função de entrada/saída
  return (0) ;                  // retorno (fim) da função
}
```

Algumas observações importantes sobre esse código:

- A linha `#include` é um comando de preprocessador; ele indica que na fase inicial da compilação deve ser lido (incluído) o arquivo `stdio.h`, que contém as definições das funções de entrada/saída da biblioteca padrão C (como a função `printf`).
- todo programa em C inicia sua execução pela função `main`, que sempre deve ser definida.
- Os blocos de código são delimitados por chaves (`{` e `}`).
- Por ser uma função, `main` deve retornar um valor (linha `return`).
- Comentários são iniciados pelos caracteres `//` ou delimitados pelos caracteres `/*` e `*/`.
- A endentação não é obrigatória, mas é fundamental para a legibilidade do código.

Para poder executar o código acima, ele deve ser salvo em um arquivo (`hello.c`) e compilado:

```
$ cc hello.c -o hello
$ ./hello
Hello, world!
```

Estrutura de um programa em C

Um programa em C é composto basicamente das seguintes partes:

- Comandos do preprocessador (linhas iniciando com #)
- Comentários
- Definição de funções
- definição de variáveis e tipos
- Comandos e expressões

Mais um exemplo simples, para concluir:

soma.c

```c
/*
  Calcula a soma e a média de dois inteiros,
  escrevendo o resultado na saída padrão.
*/

#include <stdio.h>

int main ()
{
  int a, b, soma ;
  float media ;

  a = 10 ;
  b = 43 ;
```

```
  soma  = a + b ;
  media = soma / 2.0 ;

  printf ("A soma  de %d e %d vale %d\n", a, b, soma) ;
  printf ("A media de %d e %d vale %f\n", a, b, media) ;

  return (0) ;
}
```



Procure comentar o código adequadamente, para facilitar sua compreensão. Um bom comentário deve explicar **o que** um bloco de código faz, não **como** ele faz (pois isso deve estar claro no próprio código).

Comentar código corretamente exige alguma técnica. Este texto traz algumas boas dicas sobre comentários.

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=c:introducao_a_linguagem_c

Last update: **2023/09/07 17:08**