Linguagem C

Introdução

MsC. Douglas Santiago Kridi

Programação I - 2018.2

Bacharelado em Ciência da Computação

Universidade Estadual do Piauí

douglaskridi@gmail.com

Introdução

- Podemos definir uma linguagem de programação como:
  - Um *método padronizado* para comunicar instruções para um computador.
  - Ou, um *conjunto de regras sintáticas e semânticas* usadas para definir um programa de computador.

Tradução

- Uma linguagem de programação pode ser convertida, ou traduzida, em código de máquina por *compilação* ou por *interpretação*.
  - Em ambas ocorre a tradução do código fonte para código de máquina.

Tradução

- Se o método utilizado *traduz todo código* do programa, para só *depois* executa-lo, então diz-se que o programa foi compilado.
  - E que o mecanismo utilizado para a tradução é um *compilador*.
- Na versão compilada, o programa pode ser executado um número indefinido de vezes sem que seja necessária nova compilação, o que economiza tempo.
  - Isso acontece com linguagens como *Pascal*, *C* e *C++*.

Tradução

- Se o código é executado *à medida que vai sendo traduzido*, num processo de tradução de trechos seguidos de sua execução *imediata*, então diz-se que o programa foi interpretado.
  - E que o mecanismo utilizado para a tradução é um *interpretador*.
- Programas interpretados são geralmente mais lentos do que os compilados, no entanto, são mais flexíveis, já que podem interagir com o ambiente mais facilmente.
  - São exemplos de linguagens interpretadas *JavaScript*, *BASIC* e *Python*.

Tradução

- E o caso do Java?
  - Existem linguagens que são compiladas e interpretadas, como é o caso do Java.
- Um compilador traduz o código java para o código intermediário (*bytecode*) e portável da JVM.
- As JVMs originais interpretavam esse código, de acordo com o código de máquina do computador hospedeiro.

Origem

- ***C*** é uma linguagem e de programação *compilada* de propósito geral, *estruturada*, *imperativa*, *procedural*, padronizada pela *ISO*.

- Criada em 1972, por ***Dennis Ritchie***, no *AT&T Bell Labs*, para desenvolver o sistema operacional *Unix* (que foi originalmente escrito em *Assembly*).

Origem

- C é uma das linguagens de programação mais populares e existem poucas arquiteturas para as quais não existem compiladores para C.
  - Também tem influenciado muitas outras linguagens, por exemplo *C++,* que originalmente começou como uma extensão para C.

Dennis Ritchie e Ken Thompson

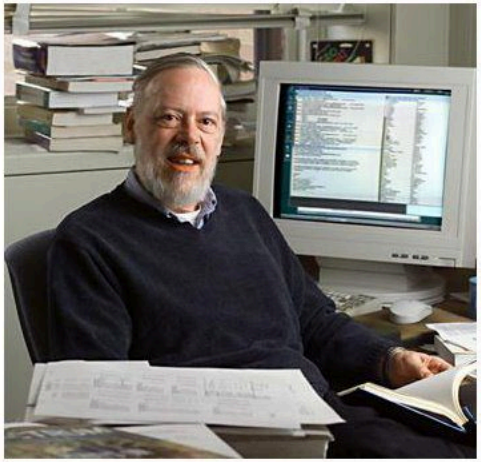
Origem

- Reflita:

Retirado da internet. Fonte desconhecida.

ATTENTION

both Died in the same year
and same month
But steve was considered as a hero
and dennis was ignored by the
world.
Only a handful of Programmers
who really know the value of
dennis even know his death



WIthout Steve there is no Iphone,IPad,Mac,and Apple Computers.But is that a big deal?I mean I am in Computer science Field and I have'nt used an apple product in my life and Im Good enough.

But think !!!!

Without Dennis there is no C.If there is No C,then there is no C++ that means No Unix,Windows,Linux, No Crysis warhead and other cool games,No Photoshop,No Firefox No VLC No FL Studio,No playstation,No XBOX and the list continues.90% of the applications in the world are written in C and C++

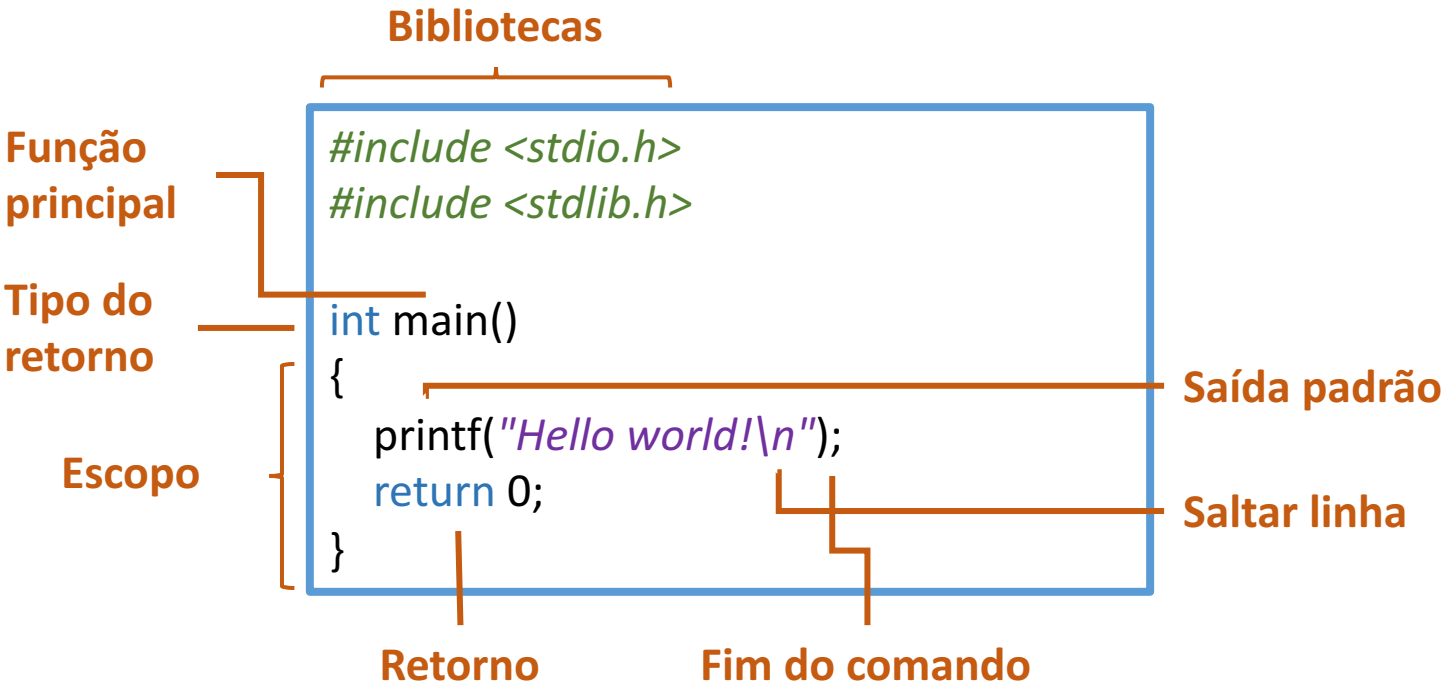
IF YOU THINK DENNIS DESERVE A RESPECT.PLEASE SHARE IT

Características

- Um programa em C é composto por um *conjunto de* ***Funções***. A função pela qual o programa começa a ser executado chama-se ***Main***.
- Após cada comando em C deve-se colocar um **;** (*ponto-e-vírgula*).
- Um programa em C deve ser *identado* para que possa ser lido com mais facilidade.

Características

- Exemplo. Olá Mundo!



Características: Identificadores

- São os nomes que podem ser dados para variáveis e funções.
- Para a escolha destes nomes é necessário seguir algumas regras:
  - Um identificador deve iniciar por uma letra ou por um "_" (*underline*);
  - A partir do segundo *caracter* pode conter letras, números e *underline*;
  - Deve-se usar nomes significativos dentro do contexto do programa;
  - C é uma linguagem *case-sensitive*, ou seja, faz diferença entre nomes com letras maiúsculas e nomes com letras minúsculas.
    - ex.: Peso e peso são diferentes.
  - Costuma-se usar maiúsculas e minúsculas para separar palavras;
    - Ex.: pesoTotal.
  - Devem ser diferente dos comandos da linguagem (palavras reservadas);
  - Devem ter no máximo 31 caracteres;
  - Pode conter números a partir do segundo *caracter*;
    - Ex.: Idade, PesoDoCarro,Usuario_1, RaioDoCirculo, etc.

Características: Variáveis

- Uma variável é uma *posição de memória* que pode ser *identificada* por meio de um nome (*identificador*).
- Podem ter seu conteúdo alterado por um comando de atribuição.
  - Após a atribuição mudam de valor.
- Exemplo:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
   int a, b, SomaGeral;
   a = 3;   // a recebe o valor 3
   b = a * 2;   // b recebe o dobro do valor de a
   SomaGeral = a + b + 2;   // SomaGeral recebe 11
   return 0;
}
```

Características: Tipos de variáveis

- Todas as variáveis em C tem um tipo;
- Cada tipo define os valores que a variável pode armazenar;
- Cada tipo ocupa uma certa quantidade de memória.

int	Valores numéricos inteiros
char	Letras e símbolos alfanuméricos: 'a', 'H', '^', '*','1'
float	Valores numéricos de ponto flutuante. 6 dígitos de precisão
double	Ponto flutuante com até 10 dígitos de precisão

short int	deve possuir tamanho mínimo de 16 bits
long int	deve possuir tamanho mínimo de 32 bits
long long int	deve possuir tamanho mínimo de 64 bits

Características: Declaração de variáveis

- Todas as variáveis tem que ser declaradas antes de serem usadas;
  - Não há uma inicialização implícita na declaração

```
#include <stdio.h>
void main() {
  int contador;          // declarações simples
  double TaxaDeCambio;
  char LetraDigitada;
  int IdadeManoel, IdadeJoao, IdadeMaria;
  // Pode colocar mais de uma variável na mesma linha
  double  TaxaDoDolar,
          TaxaDoMarco,
          TaxaDoPeso,
          TaxaDoFranco; // Também pode trocar de linha no meio
}
```

Características: Declaração de variáveis

- Também é possível inicializar variáveis ainda na declaração:

```
#include <stdio.h>
void main() {
  int NroDeHoras = 0;          // declara e inicializa com Zero
  float PrecoDoQuilo = 10.53;  // declara e inicializa com 10.53
  double  TaxaDoDolar = 1.8,
          TaxaDoMarco = 1.956,
          TaxaDoPeso = 1.75,
          TaxaDoFranco = 0.2;
}
```

Características: Constantes

- Constantes são identificadores que não podem ter seus valores alterados durante a execução do programa.
- Para criar uma constante existe o comando #define que, em geral é colocado no início do código-fonte.

```
#define LARGURA_MAXIMA 50
#define NRO_DE_DIAS_DA_SEMANA 7
#define NRO_DE_HORAS_DO_DIA 24
#define VALOR_DE_PI  3.1415
// Não se coloca ponto-e-vírgula após o valor
void main ()
{
   int TotalDeHoras;

    TotalDeHoras = 10 * NRO_DE_DIAS_DA_SEMANA * NRO_DE_HORAS_DO_DIA;
}
```

Características: Strings

- Uma String é uma sequência de caracteres entre aspas duplas: *"exemplo de uma string em C"*.
- Constantes também podem ser definidas como String:

```
#define UNIVERSIDADE “Universidade Estadual do Piauí"
// deve-se colocar entre aspas
#include <sdtio.h>

void main() {
    printf("%s", UNIVERSIDADE); // Imprime a constante
}
```

Atividade

1. Implemente um programa que escreve na tela a frase "O primeiro programa a gente nunca esquece!“
2. Elabore um programa com variáveis que guardem seu nome, sobrenome, endereço, CEP e telefone. Ele deve imprirmir seu nome completo na primeira linha, seu endereço na segunda, e o CEP e telefone na terceira.
3. Escrever um programa que mostre a seguinte figura :

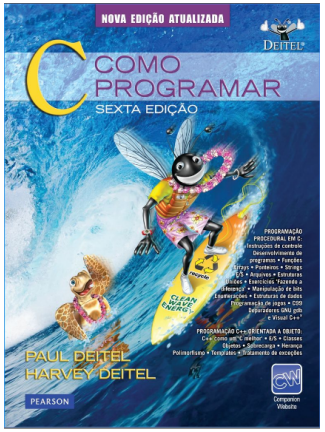
```
XXXXX
X   X
X   X
X   X
XXXXX
```

4. Escreva um programa que produza a seguinte saída na tela:

```
ALUNO(A)        NOTA
=========       =====
ALINE            9.0
MÁRIO            DEZ
SÉRGIO           4.5
SHIRLEY          7.0
```

5. Faça um programa que converta Celsius para Fahrenheit.

Referências Bibliográficas



C: COMO PROGRAMAR

DEITEL, Harvey M.; DEITEL, Paul J. Editora Pearson - 6ª ed. 2011

Fundamentos da Programação de Computadores

Ascencio, Ana F. G., Campos, Edilene A. V. de, - Editora Pearson 2012

Lógica de Programação e Estrutura de Dados

Puga, Sandra. Risseti, Gerson. – Ed. Pearson - 2016

Lógica de Programação Algorítmica (Apostila)

Guedes, Sergio. - Editora Pearson/Ser - 2014