



Conteúdo 4

Programação Estruturada e Algoritmos 1

Introdução à Linguagem de Programação C

Prof. Fernando Vieira Duarte

Sumário

2

- ❑ Objetivos
- ❑ Tradução de algoritmos em código em Linguagem C
- ❑ Estrutura básica de um programa em C
- ❑ Processo de compilação e geração de código executável
- ❑ Variáveis e tipos de dados em C
- ❑ Comandos de entrada e de saída de dados
- ❑ Operadores aritméticos em C
- ❑ Referências bibliográficas

Objetivos

3

- ❑ Abordar a tradução de algoritmos em linguagem de programação C.
- ❑ Fazer uma introdução ao IDE Visual Studio Code, apresentando o processo de compilação e execução de programas no VS Code.
- ❑ Apresentar os operadores de atribuição e aritméticos (+, -, *, / e %) e os comandos entrada e de saída de dados.

Tradução de Algoritmo em Linguagem C

4

- Como é costume, o primeiro programa que se aprende a fazer em qualquer linguagem de programação é um programa que mostra uma frase na tela: “Olá, Mundo!”.

Tradução de Algoritmo em Linguagem C

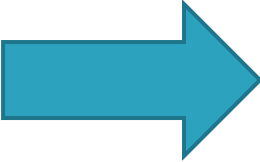
5

algoritmo Primeiro

início

escreva ("Olá, Mundo!");

fim.



```
1. #include <stdio.h>
2. int main()
3. {
4.     printf("Olá, Mundo!\n");
5.     getchar();
6.     return 0;
7. }
```

Estrutura Básica de um Programa em C

6

1. `#include <stdio.h>`

#include permite a inclusão de um arquivo de cabeçalho.

Standard In/Out, contém rotinas necessárias para a entrada e saída padrão de dados.

Bibliotecas incluem funções pré-escritas por outros programadores que auxiliam no desenvolvimento do código, sem a necessidade de reescrever algumas rotinas.

Exemplos de funções da stdio: fprintf, printf, fscanf, scanf, getchar, putc, getc, fopen, fclose.

Estrutura Básica de um Programa em C

7

1. **int main()**

2. **{**

3. *//aqui vem todo*

4. *//o corpo do programa*

5. **}**

Todo programa C deve, obrigatoriamente, ter esta função.

main determina o local onde os comandos do programa são escritos e delimita seu bloco de instruções por meio de um par de chaves

Estrutura Básica de um Programa em C

8

```
1. #include <stdio.h>
2. int main()
3. {
4.     printf("Hello world!\n");
5.     getchar();
6.     return 0;
7. }
```

Exibe informações (que estiverem dentro dos parênteses e aspas) na tela

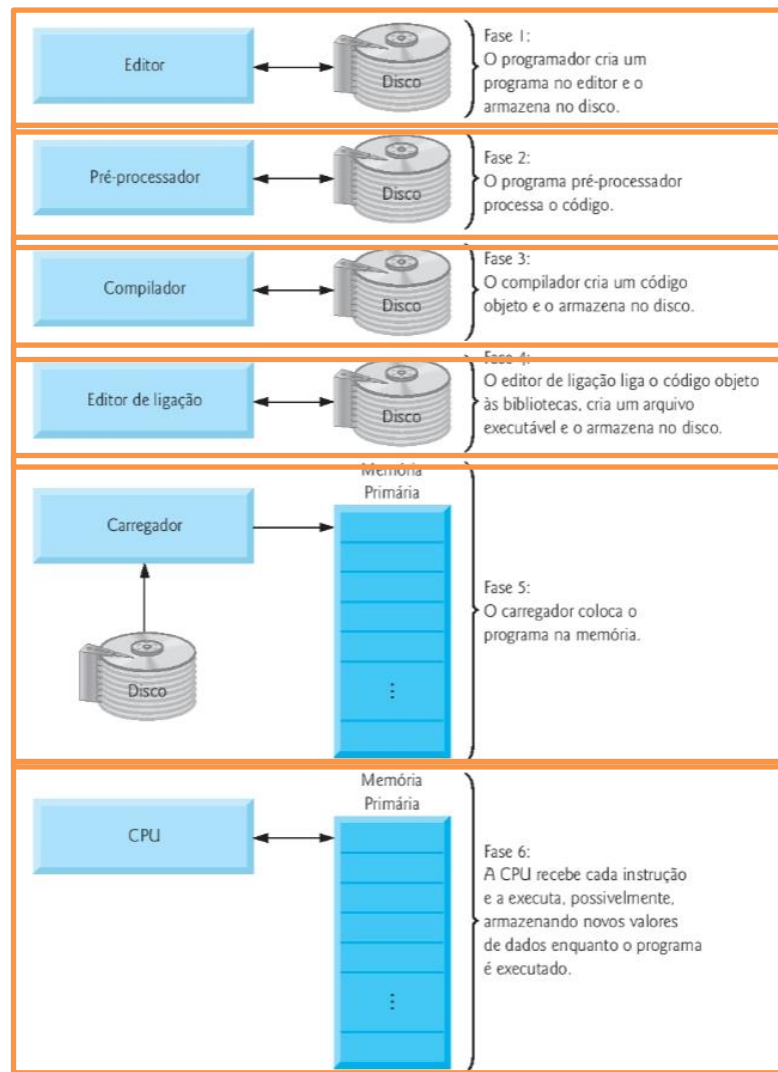
Sequência de escape	Descrição
\n	Nova linha. Posiciona o cursor da tela no início da próxima linha.
\t	Tabulação horizontal. Move o cursor da tela para a próxima posição de tabulação.
\a	Alerta. Faz soar o alarme do sistema.
\\	Barra invertida. Insere um caractere de barra invertida em uma string.
\"	Aspas. Insere um caractere de aspas em uma string.

É útil para causar uma pausa após a execução do código e manter a janela aberta

Serve para informar ao compilador que ocorreu tudo certo com a função main().

Se a main retornasse um outro valor diferente de 0 haveria, um problema em sua execução, que seria informada ao compilador.

Processo de Compilação e Geração de Código Executável



Variáveis em Linguagem C

10

- ❑ Variável é uma posição, frequentemente localizada na memória do computador, que armazena um único valor de cada vez.
- ❑ As variáveis existem somente durante a execução de um programa (em tempo de execução), perdendo seu conteúdo quando o programa se encerra.
- ❑ Declaração:

tipo_de_dado nome_variavel;



(char, int, float ou double)

Tipos de Dados em Linguagem C

- Existem 5 tipos de dados primitivos (pré-definidos) em C:

Palavra-chave	Tipo	Tamanho
char	Caractere	1 byte (8 bits)
int	Inteiro	2 bytes (16 bits)*
float	Real de precisão simples (6 dígitos de precisão)	4 bytes (32 bits)
double	Real de precisão dupla (até 10 dígitos de precisão)	8 bytes (64 bits)
void	Vazio (sem valor)	

* depende da máquina em que o programa está rodando.

Variáveis e Tipos de Dados em Linguagem C

12

- O programa a seguir armazena o valor da idade de uma pessoa em uma variável e mostra o valor da idade armazenado na variável na tela.

Variáveis e Tipos de Dados em Linguagem C

13

□ Pseudocódigo:

```
algoritmo Mostraldade  
declare  
    VAR idade: inteiro;  
  
Início  
    idade ← 20;  
    escreva (idade);  
fim.
```

Variáveis e Tipos de Dados em Linguagem C

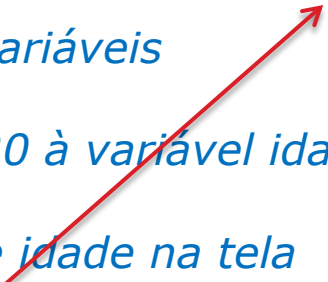
14

□ Código em C:

Especificadores de formato

- %d** ou **%i**: número inteiro, sem casas decimais
- %f**: número em ponto flutuante, ou seja, com casas decimais.
- %c**: um único caractere (letra, número ou qualquer outro caractere).
- %s**: uma string (conjunto) de caracteres.

```
1. #include <stdio.h>
2. int main()
3. {
4.     //declaração de variáveis
5.     int idade;
6.     //atribui o valor 20 à variável idade. O sinal = indica atribuição.
7.     idade = 20;
8.     //exibe o valor de idade na tela
9.     printf("Idade = %d\n", idade);
10.    getchar();
11.    return 0;
12.}
```



Entrada e Saída de Dados em Linguagem C

15

- O programa a seguir armazena o ano atual em uma variável e pergunta a idade e o ano em que o usuário nasceu, armazenando em outras duas variáveis. Calcula a idade que usuário terá ao final do ano atual e mostra na tela o valor da idade fornecido pelo usuário e a idade calculada pelo programa.

Entrada e Saída de Dados em Linguagem C

16

□ Pseudocódigo:

algoritmo CalculaIdade

declare

VAR idade, anoNasc, qtdeAnos, anoAtual: inteiro;

Início

anoAtual ← 2023;

escreva (Qual sua idade?);

leia(idade);

escreva (Qual o ano que você nasceu?);

leia(anoNasc);

qtdeAnos = anoAtual – anoNasc;

escreva("Idade informada: ", idade);

escreva("Idade ao final do ano:", qtdeAnos);

fim.

Entrada e Saída de Dados em Linguagem C

17

□ Código
em C:

```
1.  #include <stdio.h>
2.  int main()
3.  {
4.      //declaração de variáveis
5.      int idade, anoNasc, qtdeAnos;
6.      int anoAtual = 2021; //atribuição de valor durante a declaração da variável
7.      printf("Digite sua idade: ");
8.      scanf("%d", &idade); //atribuição de valor a partir do teclado
9.      printf("Qual ano voce nasceu? ");
10.     scanf("%d", &anoNasc);
11.     qtdeAnos = anoAtual - anoNasc; //atribuição de valor a partir de cálculo
12.     //imprime valores de variáveis na tela
13.     printf("\nIdade informada pelo usuario = %d\n", idade);
14.     printf("Idade que tera ao final deste ano: %d\n\n", qtdeAnos);
15.     getchar();
16.     return 0;
17. }
```

Operadores Aritméticos em Linguagem C

18

Operação em C	Operador aritmético	Expressão algébrica	Expressão em C
Adição	+	$f + 7$	$f + 7$
Subtração	-	$p - c$	$p - c$
Multiplicação	*	bm	$b * m$
Divisão	/	x / y ou $\frac{x}{y}$ ou $x \div y$	x / y
Módulo ou resto da divisão entre 2 inteiros	%	$r \bmod s$	$r \% s$

Precedência de Operadores Aritméticos em Linguagem C

19

Operador(es)	Operação(ões)	Ordem de avaliação (precedência)
()	Parênteses	Avaliados em primeiro lugar. Se os parênteses forem aninhados, a expressão no par mais interno é a primeira a ser avaliada. Se houver vários pares de parênteses 'no mesmo nível' (ou seja, não aninhados), eles serão avaliados da esquerda para a direita.
* / %	Multiplicação Divisão Módulo	Avaliados em segundo lugar. Se houver vários, serão avaliados da esquerda para a direita.
+ -	Adição Subtração	Avaliados por último. Se houver vários, serão avaliados da esquerda para a direita.

Referências Bibliográficas

20

- ❑ PUGA, Sandra; RISSETTI, Gerson. **Lógica de programação e estruturas de dados:** com aplicações em JAVA. São Paulo: Pearson Prentice Hall, 2006. 254 p.
- ❑ DEITEL, P. J.; DEITEL, H. M. **C: como programar.** 6. ed. São Paulo: Pearson Prentice Hall, 2011. 818 p.