

1 A maneira que C trabalha

C é uma linguagem para criar programas rápidos. É com ela que todos sistemas básicos são construídos. Os passos básicos de construção de um programa estão listados na figura 1.

Figura 1: Processo de criação de um programa.

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

int main(int argc, char const *argv[])
{
    puts("Hello World!");
    return EXIT_SUCCESS;
}
```

hello.c

```
$ gcc hello.c -o hello
```



hello

Estrutura de um programa C.

- 1. Comentário serve para descrever o propósito do código.

```
/**/ para comentar diversas linhas.
```

```
// para comentar uma única linha.
```
- 2. Inclusão de bibliotecas para reutilizar código.

```
#include<biblioteca>
```

 torna disponível as

1 - cria código fonte.

2 - compila para traduzir o código em comandos do computador.

3 - um executável é gerado como produto da compilação.

funções implementadas na biblioteca da linguagem.

```
#include“biblioteca”
```

 torna disponível as funções implementadas na biblioteca do usuário.

- 3. Função main(). Função que é chamada no início da execução do programa.
Pode receber valores durante a execução. Veremos exemplos depois.
Deve retornar um inteiro.
Deve ter seu código envolto entre chaves.

2 Dados

Programa = comandos + dados.

- Comandos = Operações disponíveis no computador.
- Dados = Valores a serem manipulados pelas operações.
 - Variáveis = valores que podem sofrer alterações.
 - Constantes = valores imutáveis.

2.1 Tipos de dados básicos do C

Tipo de dados	domínio dos dados	quantidade de memória (em bytes)	exemplos de operações
char	símbolos da tabela ASCII ou número inteiro equivalente entre -128 e +127	1	detectar se caractere é dígito (0, ..., 9) ou letra (A, ..., Z)
int	-2.147.483.648 a 2.147.483.647	4	Soma de números Inteiros
float	números reais com 6 dígitos de precisão	4	subtração de números reais
double	números reais com 10 dígitos de precisão	8	divisão de números reais
void	sem valor	Não se aplica	Não se aplica

- char é para armazenar caracteres. Mas o caractere 'A' não é armazenado assim. Armazena-se um inteiro pequeno que corresponde a ele, no caso

- 65. Isso ocorre devido a uma limitação do computador, que só sabe trabalhar com números. Assim, qualquer coisa que não seja número tem que ser convertida para um. A primeira conversão de caracteres em números pode ser vista na tabela ASCII.
- int é para armazenar números inteiros maiores que a capacidade de armazenamento de um char.
- float é para armazenar números fracionários, ou reais.
- double também é para números reais, mas com uma precisão muito maior que o float. Ou seja, suporta bem mais casa decimais.
- void serve para dizer que há ausência de valor. Entende-se melhor quando for utilizá-lo.

2.2 Definição de variáveis

Pode-se definir uma variável da forma seguinte:

```
tipo_de_dado nome_variável;
```

Exemplos:

```
char tecla;
char opcao;
int x, y, z;
float comissao, desconto, salario;
```

Definir variável é útil para informar ao compilador que um dado tem um tipo e nome determinado, mesmo que não tenha um valor. Porém é possível atribuir valor durante a definição, com o operador =.

Exemplos:

```
int d = 3, f = 5;
float f = 9.53;
byte z = 22;
char x = 'x';
```

2.3 Modificadores de tipos

Modificam as propriedades básicas dos tipos int e char.

- unsigned: só números positivos, 0 e maiores que ele.
- short: diminui a capacidade de armazenamento, limita a dois bytes.
- long e long long: aumenta a capacidade de armazenamento.

long no linux é 64 bits e no windows é 32 bits.

long long no linux é 64 bits e no windows é 64 bits.

Mas quero que meu código fique independente de sistema operacional. Inclua inttypes.h, ela disponibilizará os tipos abaixo.

Fixed width integer	signed	unsigned
8 bit	int8_t	uint8_t
16 bit	int16_t	uint16_t
32 bit	int32_t	uint32_t
64 bit	int64_t	uint64_t

Exemplos de variáveis modificadas:

```
unsigned char contador;
unsigned int a, b, c;
long int tam_arquivo;
```

3 Entrada e saída formatada de dados

A função printf exibe informações formatadas na tela. Ela tem a seguinte organização: printf(“formatação”, arg1 , arg2 , ..., argn);. Formatação é uma mensagem que será exibida, podendo ter partes substituídas pelos valores contidos nos argumentos(arg1 , arg2 , ..., argn).

Exemplo:
printf(“Hello world!”);
Imprimirá: Hello world!
printf(“%i é par.”, 2);
Imprimirá: 2 é par.

É possível imprimir valores de uma variável.

Exemplo:
int a = 65;
char b = 65;
float c = 65.0;
printf(“int: %i char: %c float: %f.\n”, a, b, c);

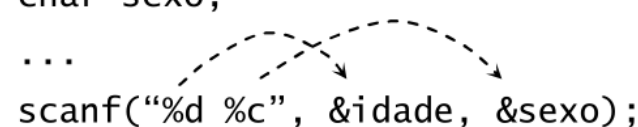
Cada tipo tem um especificador para formatação.

Type	Constant Examples	printf chars
char	'a', '\n'	%c
_Bool	0, 1	%i, %u
short int	—	%hi, %hx, %ho
unsigned short int	—	%hu, %hx, %ho
int	12, -97, 0xFFE0, 0177	%i, %x, %o
unsigned int	12u, 100U, 0XFFu	%u, %x, %o
long int	12L, -2001, 0xffffL	%li, %lx, %lo
unsigned long int	12UL, 100ul, 0xffeeUL	%lu, %lx, %lo
long long int	0xe5e5e5e5LL, 5001l	%lli, %llx, %llo
unsigned long long int	12ull, 0xffeeULL	%llu, %llx, %llo
float	12.34f, 3.1e-5f, 0x1.5p10, 0x1P-1	%f, %e, %g, %a
double	12.34, 3.1e-5, 0x.1p3	%f, %e, %g, %a
long double	12.341, 3.1e-51	%Lf, %Le, %Lg

Já a função scanf serve para entrada formatada de dados. Ela tem a organização seguinte: scanf(“formatação”, arg1, arg2, ...,argn);. Dada uma string que foi informada no sistema, se ela segue o formato da formatação, ela terá valores extraídos e colocados nos argumentos (arg1, arg2, ...,argn). Exemplo:

```
int idade;
char sexo;

...
scanf(“%d %c”, &idade, &sexo);
```



Não se esqueça de usar & antes do argumento, caso seja uma variável dos tipos que vimos até agora. No futuro, com ponteiros, isso é desnecessário.

Obs: %d é para inteiros, assim como %i.

4 Operações aritméticas básicas

C oferece operadores para realização de operações aritméticas. As básicas são mostradas abaixo. Assuma que a variável A tem valor 10 e a B valor 20.

Operator	Description	Example
+	Adds two operands.	A + B = 30
-	Subtracts second operand from the first.	A - B = -10
*	Multiplies both operands.	A * B = 200
/	Divides numerator by de-numerator.	B / A = 2
%	Modulus Operator and remainder of after an integer division.	B % A = 0
++	Increment operator increases the integer value by one.	A++ = 11
--	Decrement operator decreases the integer value by one.	A-- = 9

5 Exercícios

Exercício 1. Dadas as duas notas de um aluno, informe a sua média final.

Exercício 2. Dados uma distância e o total de litros de combustível gasto por um automóvel para percorrê-la, informe o consumo médio.

Exercício 3. Dado um caracter, informe o seu código ASCII em octal, decimal e hexadecimal.

Exercício 4. Dada um temperatura em graus Fahrenheit, informe o valor correspondente em graus Celsius. [Dica: $C = (F - 32) \times (5 \div 9)$].

Exercício 5. Dadas as medidas dos catetos de um triângulo retângulo, informe a medida da hipotenusa. [Dica: para calcular a raiz quadrada use a função sqrt(), definida no arquivo math.h].