



Aula 11: Bibliotecas

Introdução a Programação

Túlio Toffolo & Puca Huachi
<http://www.toffolo.com.br>

BCC201 – 2019/2
Departamento de Computação – UFOP

Aulas anteriores

- Funções
- Ponteiros
- Passagem por referência
- Macros

Aula de hoje

- 1 Revisão: arquivos de cabeçalho
- 2 Biblioteca padrão do C (**libc**)
- 3 Operador ternário
- 4 Exercício

Aula de hoje

- 1 Revisão: arquivos de cabeçalho
- 2 Biblioteca padrão do C (**libc**)
- 3 Operador ternário
- 4 Exercício

Arquivos de cabeçalho (*header files*)

Códigos escritos em C geralmente são divididos em dois arquivos:

- Cabeçalho (arquivo **.h**): contém os protótipos das funções.
- Código fonte (arquivo **.c**): contém a implementação das funções.

Dividir o código em arquivos traz inúmeras vantagens, entre elas:

- Melhor organização;
- Flexibilidade para exportar as funções desenvolvidas por você.

Exemplo:

Vamos desenvolver uma simples biblioteca contendo funções para converter unidades de temperatura.

- Criaremos dois arquivos: **temperatura.h** e **temperatura.c**

```
_____ temperatura.h _____  
1  double celsiusToFahrenheit(double celsius);  
2  double fahrenheitToCelsius(double fahrenheit);  
3  double celsiusToKelvin(double celsius);  
4  double kelvinToCelsius(double kelvin);  
5  double fahrenheitToKelvin(double fahrenheit);  
6  double kelvinToFahrenheit(double kelvin);
```

Exemplo:

_____ temperatura.c _____

```
1 #include "temperatura.h" // inclui o arquivo de cabeçalho
2
3 double celsiusToFahrenheit(double celsius) {
4     return (celsius * 9.0/5.0) + 32;
5 }
6
7 double fahrenheitToCelsius(double fahrenheit) {
8     return (fahrenheit - 32) * 5.0/9.0;
9 }
10
11 double celsiusToKelvin(double celsius) {
12     return celsius + 273.15;
13 }
14
15 double kelvinToCelsius(double kelvin) {
16     return kelvin - 273.15;
17 }
18
19 double fahrenheitToKelvin(double fahrenheit) {
20     return celsiusToKelvin(fahrenheitToCelsius(fahrenheit));
21 }
22
23 double kelvinToFahrenheit(double kelvin) {
24     return celsiusToFahrenheit(kelvinToCelsius(kelvin));
25 }
```

Exemplo:

Note o compilador recebe como argumento apenas o arquivo com o código fonte.

```
1 $ gcc temperatura.c -Wall -o conversor
```

No entanto, o compilador retornará um erro informando a ausência do método `main`. Se quisermos criar apenas uma biblioteca, fazemos:

```
1 $ gcc -c temperatura.c -Wall -o temperatura.o
```

O comando acima (usando a flag `-c`) compilará o arquivo sem executar a etapa de *link*, gerando o arquivo **temperatura.o** (`.o` de *object file*), que é o binário da biblioteca.

Um executável em C precisa do método `main`. Temos duas opções:

- Adicionar o método `main` em `temperatura.c`
- Criar um outro arquivo com o método `main`.

main.c

```
1  #include <stdio.h>
2  #include "temperatura.h"
3
4  int main()
5  {
6      double celsius;
7      printf("Digite a temperatura em Celsius: ");
8      scanf("%lf", &celsius);
9
10     double fahrenheit = celsiusToFahrenheit(celsius);
11     printf("Temperatura em Fahrenheit: %lf\n", fahrenheit);
12
13     double kelvin = celsiusToKelvin(celsius);
14     printf("Temperatura em Kelvin: %lf\n", kelvin);
15
16     return 0;
17 }
```

Exemplo:

Para compilar o programa podemos fazer:

```
1 $ gcc main.c temperatura.c -Wall -o conversor
```

Ou, alternativamente, podemos usar o arquivo `.o` gerado anteriormente:

```
1 $ gcc main.c temperatura.o -Wall -o conversor
```

Aula de hoje

- 1 Revisão: arquivos de cabeçalho
- 2 Biblioteca padrão do C (**libc**)
- 3 Operador ternário
- 4 Exercício

Biblioteca padrão do C

- A biblioteca padrão do C (*Standard C Library*, ou **libc**) contém uma série de **funções** e constantes úteis, importadas por meio de diferentes arquivos de cabeçalhos (*header files*).
- Nós já utilizamos algumas (`<stdio.h>` e `<math.h>`).
- Hoje veremos outras funções úteis, presentes em:
 - `<locale.h>`
 - `<limits.h>`
 - `<stdlib.h>`
 - `<time.h>`

Biblioteca <locale.h>

A linguagem C utiliza o arquivo de cabeçalho locale.h para implementar a “localização” de programas (características de idioma ou de uma região).

- Exemplo:

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main()
5 {
6     setlocale(LC_ALL, "Portuguese");
7     printf("Utilizando acentuação da língua portuguesa!\n\n");
8     return 0;
9 }
```

Biblioteca <limits.h>

Esta biblioteca contém constantes para determinar valores máximos e mínimos para vários tipos de dados:

- Exemplo:

```
1 #include <stdio.h>
2 #include <limits.h>
3
4 int main()
5 {
6     printf("int: [%d, %d]\n", INT_MIN, INT_MAX);
7     printf("long: [%ld, %ld]\n\n", LONG_MIN, LONG_MAX);
8     return 0;
9 }
```

Biblioteca <stdlib.h>

Provê funções para alocação de memória (usaremos muito no futuro), controle de processos, conversão, etc.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int i = atoi("-10"); // atoi converte string -> int
7     float f = atof("10.5"); // atof converte string -> float
8     double d = strtod("10.5", NULL); // strtod converte string -> double
9
10    system("clear"); // executa o comando clear no terminal
11
12    srand(0); // seleciona a semente para geração de nros aleatórios
13    int r = rand(); // r recebe um nro aleatório
14    printf("Número aleatório: %d\n\n", r);
15
16    printf("i = %d, f = %f, d = %lf\n", i, f, d);
17    printf("Valor absoluto de i: %d\n\n", abs(i));
18
19    exit(0); // função que finaliza o programa imediatamente
20    return 0;
21 }
```

Biblioteca <time.h>

Esta biblioteca provê funções para manipulação de data e horários.

- Exemplo:

```
1 #include <stdio.h>
2 #include <time.h>
3
4 int main()
5 {
6     time_t timer = time(NULL);
7     printf("CTIME = %s\n", ctime(&timer));
8     return 0;
9 }
```


Aula de hoje

- 1 Revisão: arquivos de cabeçalho
- 2 Biblioteca padrão do C (**libc**)
- 3 Operador ternário
- 4 Exercício

Operador ternário em C

- O operador ternário é uma alternativa para substituir o `if...else` em algumas situações por ser um comando bem enxuto.
- Permite fazer uma comparação em uma única linha.
- Particularmente úteis em **macros** e atribuições simples.

Sintaxe:

```
1 <condição> ? <op. se verdadeiro> : <op. se falso>;
```

Operador ternário em C

Exemplo de uso:

- Determinar o maior número:

```
1 x = a > b ? a : b;
```

- Equivalente a “ $x = a$ se $a > b$ ou b caso contrário”

Exemplo

Cria uma macro para indicar o maior valor entre dois números e apresente um exemplo de uso (ou seja, implemente o método `main`).

```
1  #include <stdio.h>
2
3  #define MAX(a,b) (a > b ? a : b)
4
5  int main()
6  {
7      int n1, n2;
8      printf("Digite dois números inteiros: ");
9      scanf("%d %d", &n1, &n2);
10
11     printf("O maior número é %d\n", MAX(n1, n2));
12     return 0;
13 }
```

Exemplo 2

Cria uma macro para obter um número inteiro aleatório no intervalo $[a, b]$ e apresente um exemplo de uso (ou seja, implemente o método `main`).

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 #define RAND(a,b) (rand() % (b-a+1) + a)
6
7 int main()
8 {
9     srand(time(NULL));
10    int n1, n2;
11    printf("Digite o intervalo (dois números inteiros): ");
12    scanf("%d %d", &n1, &n2);
13
14    if (n1 <= n2)
15        printf("Número sorteado: %d\n", RAND(n1, n2));
16
17    return 0;
18 }
```

Aula de hoje

- 1 Revisão: arquivos de cabeçalho
- 2 Biblioteca padrão do C (**libc**)
- 3 Operador ternário
- 4 Exercício

Exercício

Questão 01

Crie um programa em C que lê os valores de x e y e em seguida calcula o valor de r (ver expressão a seguir) e imprime o resultado para o usuário.

$$r = \begin{cases} x^2 + y & \text{se } x \text{ for par} \\ x + y^2 & \text{se } x \text{ for ímpar} \end{cases}$$

Implemente seu programa de duas formas:

- 1 Utilizando `if...else`
- 2 Utilizando o operador ternário



Perguntas?