

Parâmetros de Funções

Módulo 6 Aula 1

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

Os parâmetros são a forma de entrada de dados nas funções e procedimentos em C e podemos declarar quantos parâmetros forem necessários. Os parâmetros consistem basicamente em variáveis locais, que existirão apenas no escopo da função específica e deverão ser inicializados na chamada da função. Todos os tipos de dados aprendidos até aqui podem ser utilizados na declaração dos parâmetros, que estarão entre os parênteses do cabeçalho da função. No Box 1 temos o exemplo do protótipo de uma função sem parâmetros.

```
void func1(void);
```

Box 1 - Protótipo de função sem parâmetros.

No Box 2 apresentamos uma função do tipo *void*, que não retorna nada, porém recebe um parâmetro de entrada do tipo inteiro.

```
void func2(int a);
```

Box 2 - Protótipo de função com parâmetro inteiro.

Logo, a função *func2* espera um parâmetro do tipo inteiro e quando chamada pela função principal, ou no corpo de outra função, este parâmetro deverá ser preenchido. Veja o exemplo do Box 3.

```
void func2(int a);  
  
main()  
{  
    func2(73);  
}  
/* end main */  
  
void func2(int a)  
{  
    printf("a= %d\n",a);  
}  
/* end func2 */
```

Box 3 - Chamada da função com parâmetro inteiro.

No exemplo do Box 3, colocamos o número inteiro 73 como parâmetro da função. O que *func2* fará é imprimir o 73 na tela, pois no corpo da função, utilizamos *printf* para imprimir o valor que foi carregado no parâmetro a.

Quando utilizar mais parâmetros, você os declara separados por vírgula, onde todos eles deverão receber a designação do tipo de dado, ainda que tenhamos dados do mesmo tipo. Observe no Box 4.

```
void func3(int a, char b, float c);    /* declaração correta */
void func4(int x, int y, int z);       /* declaração correta */
void func5(int h, i, j);               /* declaração INCORRETA */
```

Box 4 - Declaração de mais de um parâmetro.

A função a seguir imprime a divisão e o resto da divisão de dois números (Box 5).

```
void div_rest(int a, int b)
{
    printf("Divisao= %d\n", a/b);
    printf("Resto=   %d\n", a%b);
} /* end div_rest */
```

Box 5 - Função para imprimir divisão e o resto de dois números.

A função a seguir converte o caractere digitado para maiúscula, caso o usuário entre com a, b, c ou d. Se digitar qualquer outro, imprime um erro (Box 6).

```
void carac(char ch)
{
    switch(ch)
    {
        case 'a': putchar('A'); break;
        case 'b': putchar('B'); break;
        case 'c': putchar('C'); break;
        case 'd': putchar('D'); break;
        default:  putchar('E');
                  putchar('r');
                  putchar('r');
                  putchar('o');
    } /* end switch */
    putchar('\n');
} /* end carac */
```

Exercício resolvido: desenvolver uma função em C para calcular a frequência de um oscilador com o circuito integrado 555. A função deve receber os parâmetros de entrada: R_A , R_B e C e imprimir o valor da frequência na tela. A equação é obtida do *datasheet* do 555 e pode ser vista abaixo:

$$f = \frac{1,44}{(R_A + 2R_B) \times C}$$

Exercício proposto: desenvolva uma função em C que imprima o valor da frequência de corte de um filtro passa-baixas, onde o usuário entrará com os parâmetros: R, C e 'A'. A equação é a que segue:

$$f = \frac{1}{2\pi RC}$$

A mesma função deverá calcular e imprimir na tela o valor da constante de tempo, resultado do produto de R e C, caso o usuário entre com R, C e 'B'. A equação é a que segue:

$$\tau = RC$$

Exemplo: o usuário entra com 1000, 1E-6 e 'A'. O cálculo matemático é

$$f = \frac{1}{2\pi \times 1000 \times 1 \times 10^{-6}} = 159,15Hz$$

No console, deverá ser impresso "f= 159.15Hz".

Exemplo: o usuário entra com 2700, 100E-9 e 'B'. O cálculo matemático é

$$\tau = 2700 \times 100 \times 10^{-9} = 0,00027s$$

No console, deverá ser impresso "t= 0.00027s".

Bibliografia: DAMAS, Luís; Linguagem C, décima edição.

Disponível em: <https://amzn.to/3nGdlbN>