

Macros

Módulo 6 Aula 4

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

As macros basicamente são trechos de código substituídos pelo pré-processador, antes que o compilador passe pelo código. Para criar uma macro, basta utilizar a diretiva *#define* em seu código. No Box 1 temos o exemplo de uma macro que calcula a multiplicação entre dois números.

```
#define  multip(a,b)  a*b
```

Box 1 - Macro de multiplicação.

Observe que a linha do Box 1 não é Linguagem C e sim um comando que damos ao pré-processador. O fato de não existir o ponto-e-vírgula no final da linha é um indicativo de que realmente não se trata de Linguagem C. Ou seja, antes de compilar, a diretiva *#define* irá substituir “multip(a,b)” em seu código, por “a*b”.

Uma das vantagens das macros é que as mesmas não diferenciam tipos de dados, dessa forma, podemos ter uma macro que realiza a divisão de dois números e usá-la conforme o código do Box 2.

```
#define  div(a,b)  a/b

main()
{
    int    result_i;
    float  result_f;

    result_i = div(10,2);          /* calcula a divisão de inteiros */
    result_f = div(33.5, 7.8);     /* calcula a divisão de reais */

    printf("result_i= %d\n",result_i);
    printf("result_f= %f\n",result_f);
} /* end main */
```

Box 2 - Macro para divisão de dois números.

A macro do Box 3 retorna o maior valor entre dois números.

```
#define  maior(a,b)  a>b ? a : b
```

Box 3 - Macro que retorna o maior entre dois números.

Sempre é conveniente utilizar parênteses nas macros para evitar erros nos seus resultados, com os seguintes critérios:

- Inserir parênteses em cada um dos parâmetros existentes na expansão da macro;
- Inserir parênteses em volta de cada expressão;
- Inserir parênteses em volta de toda a macro.

Com estas regras as macros dos Boxes 1, 2 e 3 devem ser escritas conforme o Box 4.

```
#define  multip(a,b)  ((a)*(b))  
#define  div(a,b)    ((a)/(b))  
#define  maior(a,b)  (((a)>(b)) ? (a) : (b))
```

Box 4 - Macros com utilização de parênteses.

Quando for necessário escrever a macro em mais linhas, deve-se utilizar barra invertida (\) como operador de continuidade. Apresentamos um exemplo no Box 5.

```
#define  hello  printf("Hello ");    \  
          printf("beautiful ");    \  
          printf("world!\n")
```

Box 5 - Macro que ocupa mais linhas, com operador de continuidade \.

Repare no uso do ponto-e-vírgula ao final das suas primeiras chamadas de *printf* e em sua omissão na última chamada. Isso ocorre pois no código sua macro será chamada como no Box 6.

```
main()  
{  
    hello;    /* chama a macro que imprime "Hello beautiful world! */  
} /* end main */
```

Box 6 - Chamada da macro de mais linhas.

O operador # pode ser utilizado em sua macro para indicar expressões entre aspas, o que é interessante dependendo da situação. Segue um exemplo no Box 7.

```
#define disp(text) printf(#text)

main()
{
    disp(Texto sem aspas!\n);
}
```

Box 7 - Usando o operador # para substituir aspas.

Quando desejamos separar variáveis por índices, podemos utilizar o operador de concatenação ##, conforme exemplo do Box 8.

```
#define number(index) id##index

main()
{
    int id1 = 20, id2 = 30;

    printf("%d\n",number(1));
    printf("%d\n",number(2));
}
```

Box 8 - Utilizando o operador de concatenação ##.

Exercício resolvido: implemente uma macro que faça o sorteio de 10 números aleatórios de 8 bits e imprima-os na tela.

Exercício proposto: desenvolva uma macro que calcule a média ponderada entre 3 notas de 1 a 10. Desenvolva o projeto da interface, onde o usuário entra com a nota e o respectivo peso. Ao final do terceiro peso, o sistema calcula o resultado.

Exemplo: o usuário entra com

Nota 1= 7,0 Peso 1 = 2

Nota 2 = 8,5 Peso 2 = 3

Nota 3 = 9,1 Peso 3 = 5

O sistema resolverá a equação

$$Mp = \frac{nota1 \times peso1 + nota2 \times peso2 + nota3 \times peso3}{peso1 + peso2 + peso3}$$

$$Mp = \frac{7 \times 2 + 8,5 \times 3 + 9,1 \times 5}{2 + 3 + 5} = 8,5$$

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

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