

O laço *for*

Módulo 5 Aula 4

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

O laço *for* consiste em uma poderosa estrutura de repetição, sendo que com ele podemos criar loops infinitos, finitos e definir ranges específicos para as nossas iterações. Além de muitas aplicações que vão do básico ao mais avançado, este último em aplicações pouco convencionais do *for*. A sintaxe convencional da instrução *for* está explícita no Box 1.

```
for(cargas iniciais ; condição ; pós-instrução )  
    instruções;
```

Box 1 - Sintaxe do for convencional.

Quem já programou em outras linguagens, pode ter utilizado o laço *for*, pois ele é comum a linguagens baseadas em procedimentos. No entanto em C, o mesmo apresenta uma incrível flexibilidade. O laço se comporta da seguinte forma:

- O código contido em cargas iniciais é executado;
- A condição é avaliada. Se o resultado der falso, o laço encerra sua execução e o programa segue após o encerramento da instrução associada ao *for*. Se o resultado for verdadeiro a instrução ou bloco de instruções é executada.
- Quando a instrução ou bloco de instruções forem executadas, processa-se a pós-instrução, onde normalmente trabalha-se com incremento e decremento de uma variável para conseguirmos avançar nas iterações do laço. A condição é avaliada novamente após a execução da pós-instrução.

No Box 2 temos o primeiro exemplo de uso do *for*. Um programa que imprime os números de 1 a 10 na horizontal na tela.

```
main()  
{  
    char i;  
  
    for(i=1; i<=10; i++)    /* for para 10 iterações */  
        printf("%d ",i);    /* repete 10 vezes */  
  
    putchar('\n');          /* insere nova linha após executar o for */  
  
} /* end main */
```

Box 2 - Escreve 1 a 10 na tela.

Como pode-se observar, conseguimos utilizar *for* em um contexto onde deseja-se um número específico de repetições em seu código.

O programa do Box 3 efetua mais de uma instrução com o *for*, e utilizamos o operador de decremento na pós-instrução. Ele imprime o cubo dos números de 7 até 1 na tela.

```
main()
{
    unsigned k, j;

    for(k=7; k>=1; k--)
    {
        j = k*k*k;
        printf("O cubo de %d \x82 %3d\n",k,j);

    } /* end for */
} /* end main */
```

Box 3 - Imprime os cubos de 7 até 1 na tela.

***for* não convencional**

Existem algumas formas não convencionais de utilizar o *for* que aumentam ainda mais as possibilidades em nossos códigos. Podemos, por exemplo, utilizar mais de uma variável de controle no laço, como no Box 4, que imprimirá os números pares de 0 a 50 na tela.

```
main()
{
    int a, b;

    for(a=0, b=0; a+b<=50; a++, b++)
        printf("%3d\n",a+b);

} /* end main */
```

Box 4 - Imprime os pares de 0 a 50.

No Box 5, temos outro exemplo onde tentamos acertar um número.

```
main()
{
    int chute, number;

    printf("Insira um numero para o seu amigo acertar: ");
    scanf("%d",&number);
    system("CLS");

    for(chute=0; chute!= number; )
    {
        printf("Tente acertar o numero: ");
        scanf("%d",&chute);
    } /* end for */

    printf("Acertou!!!\n");

} /* end main */
```

Box 5 - Programa para tentar acertar o número.

Perceba que neste caso, não utilizou-se a pós-instrução, apenas a carga em condição.

Outra aplicação do *for* é na criação de loops infinitos no código. Frequentemente programadores utilizam *for* ao invés de *while* para um loop infinito. O programa do Box 6 fica imprimindo “BOM DIA!” na tela, até que uma tecla seja pressionada.

```
main()
{
    for( ; ; )
    {
        printf("BOM DIA!\n");
        if(kbhit()) break;

    } /* end for */
} /* end main */
```

Box 6 - Imprime "BOM DIA!" até clicar em uma tecla.

Perceba que para que um *for* gere loop infinito, você só precisa deixar de preencher a carga inicial, condição e pós-instrução.

Outra aplicação é utilizar o *for* aninhado, isso é especialmente útil para construção de tabelas em C. O programa do Box 7 imprime 5 linhas e 10 colunas de 1's na tela.

```
main()
{
    int i,j;

    for(i=0;i<5;i++)
    {
        for(j=0;j<10;j++)
            printf(" 1 ");

        putchar('\n');

    }
} /* end main */
```

Box 7 - Imprime 5 linhas e 10 colunas de 1s.

Exercício resolvido: você precisa imprimir o mapa de uma memória que tem 64 bytes de endereços. Os 6 primeiros endereços deverão ter os dados 00h, 01h, 02h, 03h, 04h, 05h e todos os restantes FFh. Desenvolva um projeto em C que gere este mapa na tela, considerando que cada linha terá 16 bytes.

Exercício proposto: Crie um projeto em C para solicitar ao usuário o número de degraus de uma escada a ser desenhada na tela. Exemplo, se o número for 3, a saída de tela será conforme o Box 8.

```
#  
##  
###
```

Box 8 - Escada com 3 degraus.

Se o número for 7, a saída do programa será conforme o Box 9.

```
#  
##  
###  
####  
#####  
#####  
#####  
#####
```

Box 9 - Escada de 7 degraus.

E assim por diante, conforme entradas do usuário.

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

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