
UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES
PROGRAMA DE EDUCAÇÃO TUTORIAL

Dicas PET-Tele

Apostila Básica de Funções em C

Autor: Rodrigo Duque Ramos Brasil

Tutor: Alexandre Santos de la Vega

Niterói-RJ
Junho / 2012

Sumário

1	Definição	2
2	Compiladores	2
3	Bibliotecas	3
4	Variáveis	3
5	Principais Funções, Comandos e Operações	4
5.1	printf	4
5.2	scanf	4
5.3	Operações Matemáticas	5
5.4	for	6
5.5	while	6
5.6	if e else	7

1 Definição

O C é uma linguagem de programação criada por Dennis Ritchie, em 1972 e que continua a ser muito utilizada até os dias atuais. Seu uso é bastante amplo, podendo ser utilizada para automatizar ferramentas e construção de softwares. Este tutorial demonstrará as funções básicas em C, apenas com o intuito de orientar um usuário a entender sobre a linguagem e ver como o código funciona.

2 Compiladores

Os compiladores são ferramentas utilizadas para traduzir uma determinada linguagem (código fonte) em um código objeto. Esse código objeto permite a um programa, através de uma linguagem binária, conversar com um sistema, por exemplo. Para C, é bastante utilizado o GCC (em Linux) e o Dev C++ (para Windows). No caso do GCC, temos um tutorial explicando a utilização dele no link:

<http://www.telecom.uff.br/pet/petws/downloads/dicas/DicasPetTeleGcc.pdf>

Com um compilador, fica mais fácil entender o funcionamento de um programa. No caso do Dev C++, que é mais voltada para usuários do Windows, não disponibilizamos de uma apostila completa, no entanto, é uma opção para quem não tiver acesso ao ambiente unix.

3 Bibliotecas

Fazendo uma analogia, podemos comparar um programa com uma casa. Como sabemos, uma casa é dividida em vários cômodos com determinadas funções. Por exemplo, o quarto é onde podemos dormir (usar a cama) ou estudar (usar a mesa). Na sala, é um lugar de descanso, em que podemos assistir televisão (utilizar a TV), e na cozinha podemos utilizar o liquidificador e o fogão. Os cômodos seriam as bibliotecas e os objetos citados (como cama, mesa, etc) seriam funções. Num programa, podemos escrever um código e colocá-lo como uma função, armazenando em uma biblioteca. Essa biblioteca pode armazenar inúmeras funções de variados tipos. Para deixar uma biblioteca organizada e não muito pesada, os programadores costumam dividir suas funções em cômodos diferentes (bibliotecas). Dessa forma, um usuário pode andar livremente em sua casa (programar), podendo incluir novos cômodos em sua casa à medida que ele precise. A biblioteca mais utilizada em C é a `stdio`, que contém as suas funções básicas. A inclusão de bibliotecas, ocorre, na grande maioria das vezes, em primeiro lugar, ocorrendo da seguinte forma (incluindo a `stdio`):

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

4 Variáveis

As variáveis são como caixas, em que você pode armazenar alguma informação. Computacionalmente falando, elas são posições na memória do computador capazes de armazenar um determinado valor que um usuário pode atribuir. Sempre que formos utilizar uma variável, mostra-se necessário declará-la na tela. Os principais tipos são:

- `int` (números inteiros);
- `float` (números reais);
- `string` (letras).

Uma variável inteira pode ser declarada através do código:

```
int n;
```

5 Principais Funções, Comandos e Operações

5.1 printf

A função printf é utilizada para escrever algo na tela. Quando utilizado entre aspas, apenas mostra uma frase que o usuário digitou.

Exemplo:

```
#include <stdio.h>

int main()
{

printf("Hello World! \n");
return 0;

}
```

O programa apenas escreverá Hello World! na tela. O \n foi utilizado para pular linha. As chaves são para iniciar e encerrar a função main. O int antes da função main demonstra que a função retornará um valor inteiro, justificando o return 0 na penúltima linha.

5.2 scanf

A função scanf lê uma letra que o usuário digitará. É utilizada para armazenar um valor em uma dada variável.

Exemplo:

```
#include <stdio.h>
int main ()
{
    int a;

    printf ("Escreva um nmero: ");
    scanf ("%d", &a);
    printf ("%d", a);
    return 0;
}
```

O programa apenas lerá um valor que o usuário digitar, armazenará na variável "a" e escreverá o número na tela. "%d" indica que a variável a ser lida é inteira e o &a está armazenando o valor dado em "a". Se fosse uma string, usaríamos "%s" e &nomedavariavel.

5.3 Operações Matemáticas

As operações matemáticas básicas no C são bastante simples. Neste tutorial, demonstrarei apenas os tipos mais simples de operações:

Soma:
 $a = b + c;$

(A variável a está recebendo o valor da soma de $b + c$. Se representássemos como $a = a + b$, a variável a estaria recebendo a soma de seu valor, até então, com b, armazenando um novo valor.)

Subtração:
 $a = b - c;$

Multiplicação:
 $a = c * b;$

Divisão: $a = b/c;$

(Tome cuidado com uma divisão, uma vez que pode resultar em um número real e você deve armazenar esse valor em uma variável do tipo float)

5.4 for

O comando for é utilizada para determinar repetições presentes em seu intervalo. Enquanto a condição dada for verdadeira, a função se repetirá por várias vezes.

```
#include <stdio.h>
int main()
{
    int i;
    for(i = 1; i <= 10; i++)
    {
        printf("%d", i);
    }
    return 0;
}
```

Se você rodar esse programa, aparecerá na tela:

```
12345678910
```

Isso se deve porque na função, i começa como 1. Terminando a parte do código entre chaves, a função irá verificar se a condição pedida ($i \leq 10$, que significa, matematicamente, $i = 10$) ainda é verdadeira. Caso seja, repetirá o código, mas, dessa vez, com um novo valor para i ($i++$), que estará somando um à variável i ($i = i + 1$), e , por isso, terá valor 2. Isso se repete até i valer 10, que finalizará a condição pedida.

5.5 while

O comando while é semelhante à função for, podendo realizar as mesmas tarefas. No entanto, o while é mais utilizado para realizar funções infinitas.

```
#include <stdio.h>
int main ()
{
    int valor = 1;

    while(valor <= 10)
    {
```

```

    printf("%d", valor);
    valor = valor + 1;
}
return 0;
}

```

Observe que caso não houvesse a linha `valor = valor + 1;`, o loop (repetição) seria infinito.

5.6 if e else

O comando `if` verifica uma condição. Caso a condição seja verdadeira, o programa realizará o que está na função (entre as chaves). Caso contrário, o programa ignorará o que está presente no `if`. Caso você queira realizar uma outra operação se a condição presente no `if` não for estabelecida, o `else` (caso contrário) cumprirá esse papel. Se a condição do `if` for satisfeita, o programa ignorará o que está presente no `else`.

```

#include <stdio.h>
int main ()
{
    int valor;

    printf (Escreva um numero: ");
    scanf ("%d", &valor);

    if(valor > 5){
        printf("O valor e maior que 5.");
    }
    else{
        printf("O valor nao e maior que 5.");
    }
    return 0;
}

```

O programa irá ler um número digitado pelo usuário e armazenará em uma variável em `valor`. Feito isso, a função `if` verificará se `valor` é maior que 5. Se for, aparecerá na tela:

O valor é maior que 5.

Caso contrário, aparecerá:

O valor não é maior que 5.