

Linguagem de Programação C

Passagem de Parâmetros

Cristiano Lehrer

<http://www.ybadoo.com.br/>

Revisão – Funções e Procedimentos

```
int maior(int a, int b) {  
    return (a > b) ? a : b;  
}
```

- Nome da função: `maior`
- Tipo de retorno: `int`
- Lista de parâmetros: `int a, int b`
- Corpo da função: `return (a > b) ? a : b;`
- Uma função que não retorne qualquer tipo, isto é, que retorne `void`, chama-se popularmente procedimento.
- O número e o tipo dos argumentos passados a uma função devem corresponder ao número e ao tipo dos parâmetros com que esta foi definida.

Tipo de Retorno

- Uma função possui apenas um tipo de retorno:

```
int maior(int a, int b)
```

- Um procedimento não possui um tipo de retorno:

```
void escrever(int number)
```

- Caso a função precise retornar mais do que um valor, como isso é possível?

Tipos de Passagem de Parâmetros (1/2)

- Passagem de parâmetros por valor:
 - Numa passagem de parâmetros por valor, são sempre enviadas para a função cópias dos valores de que esta necessita.

```
void troca(int a, int b)    int main()
{
    int tmp;
    tmp = a;
    a = b;
    b = tmp;
}
{
    int n = 10;
    int k = 15;
    printf("%d %d\n", n, k); // 10 15
    troca(n, k);
    printf("%d %d\n", n, k); // 10 15
    return 0;
}
```

Tipos de Passagem de Parâmetros (2/2)

- Passagem de parâmetros por referência:
 - Numa passagem de parâmetros por referência, são sempre enviadas para a função as referências as variáveis de que esta necessita.

```
void troca(int *a, int *b)    int main()
{
    int tmp;
    tmp = *a;
    *a = *b;
    *b = tmp;
}

{
    int n = 10;
    int k = 15;
    printf("%d %d\n", n, k); // 10 15
    troca(&n, &k);
    printf("%d %d\n", n, k); // 15 10
    return 0;
}
```

Passagem de Vetores para Funções

- O nome de um vetor corresponde ao endereço do seu primeiro elemento:
 - Se s for um vetor, então $s == \&s[0]$
- Os vetores são sempre passados às funções sem o $\&$, pois o nome de um vetor é por si só um endereço:

```
int v[] = {1, 2, 3, 4, 5};
```

```
somar(v, 5);
```

```
int somar(int v[], int max)
{
    int soma, i;
    for(i = 0, soma = 0; i < max; i++)
    {
        soma = soma + v[i];
    }
    return soma;
}
```

```
int somar(int *v, int max)
{
    int soma;
    for(soma = 0; max > 0; max--)
    {
        soma = soma + *v++;
    }
    return soma;
}
```

Passagem de Parâmetro na Linha de Comando (1/2)

```
C:/> copy alfa beta
```

- A função `main` pode receber valores passados via linha de comando:

```
int main(int argc, char *argv[])
```

- `argc`
 - É um inteiro que indica quantos argumentos foram passados na linha de comando (incluindo o próprio nome do programa).
- `argv`
 - Vetor contendo todas as strings passadas na linha de comando.

Passagem de Parâmetro na Linha de Comando (2/2)

- Escreva um programa que apresente todos os parâmetros que recebeu a partir da linha de comando.

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int i;

    for(i = 0; i < argc; i++)
    {
        printf("%d - parâmetro: %s\n", i + 1, argv[i]);
    }

    return 0;
}
```