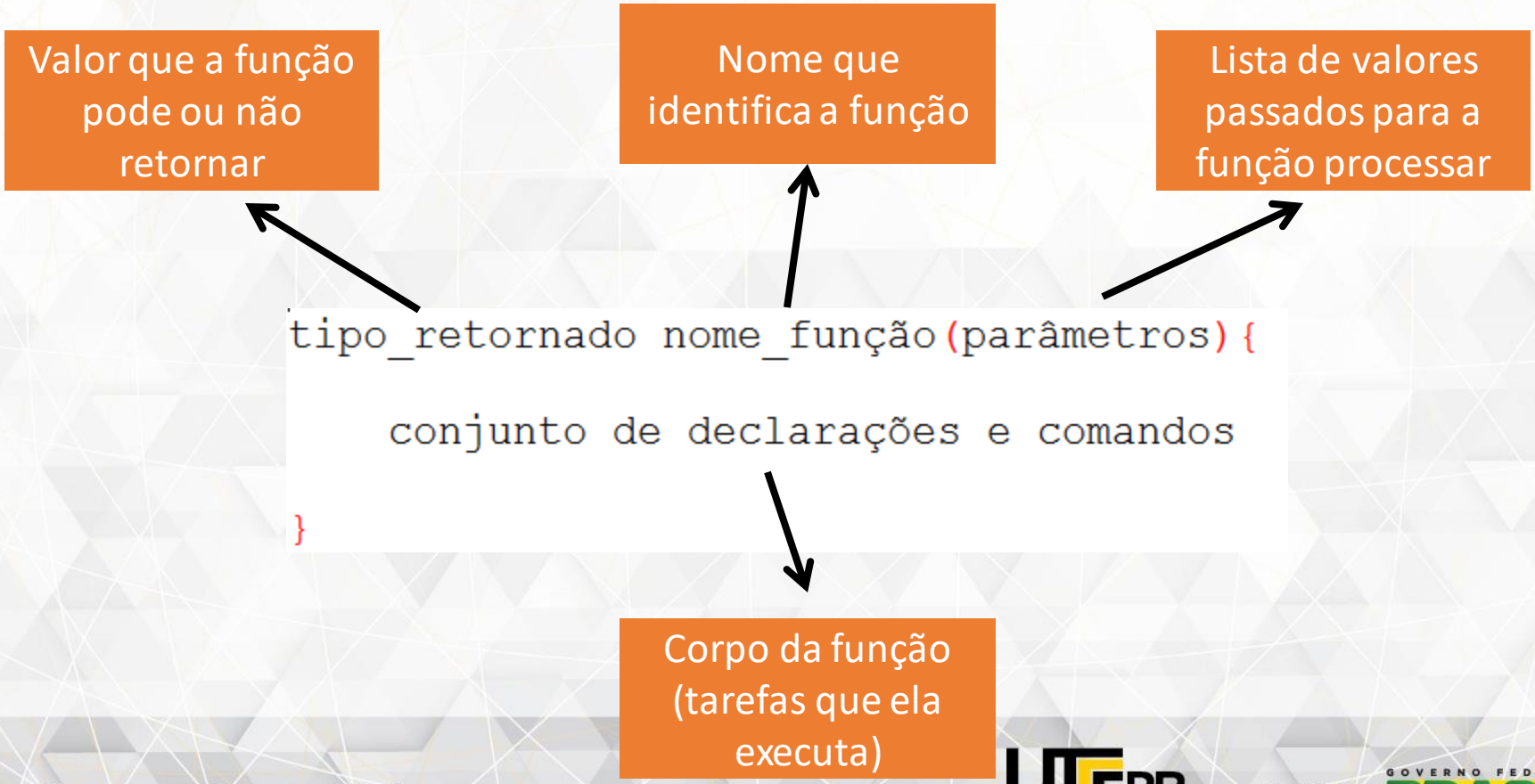
Funções
Parâmetros em Linguagem C

Profa. Dra. Giani Carla Ito

Disciplina: Linguagem de Programação Estruturada

Função - Estrutura

- Forma geral de uma função:



Parâmetros

São canais por onde os dados são transferidos pelo algoritmo chamador a um subalgoritmo.

É por meio dos parâmetros que uma função recebe informação do programa principal

Passagem de parâmetros

Consiste na transferência de informações para as funções

A passagem de parâmetros pode ser de duas formas: por valor e por referência.

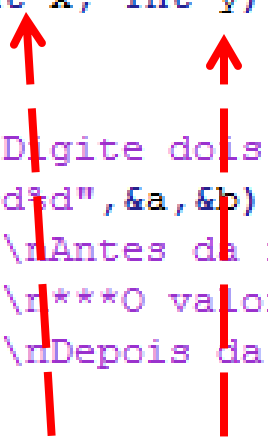
Passagem de parâmetros – Por Valor

Quanto é feita uma "cópia" do valor das variáveis utilizadas para chamar a função nos parâmetros.

Alterações feitas localmente às funções não alteram o valor das variáveis no programa principal.

Exemplo – Passagem de parâmetro por valor

```
#include<stdio.h>
int soma (int x, int y);
void main(){
     int a,b;
     printf("Digite dois valores :");
     scanf("%d%d",&a,&b);
     printf("\nAntes da funcao %d %d",a, b);
     printf("\n***O valor da soma e: %d.***",soma(a,b));
     printf("\nDepois da funcao %d %d",a, b);
}
int soma (int x, int y){
     int r;
     r=x+y;
     x=20;
     y=25;
     printf("\nDentro da funcao %d %d",x, y);
     return(r);
}
```



Passagem por referência

Quando se quer que o valor da variável mude dentro da função, usa-se passagem de parâmetros por ***referência***.

Neste tipo de chamada, não se passa para a função o valor da variável, mas a sua ***referência*** (seu endereço na memória);

Passagem de Parâmetros

Por referência: Se dá a partir do uso de ponteiros.

Um **ponteiro** é uma variável utilizada para receber um endereço de memória, ou seja, aponta para um local da memória.

Assim, as variáveis de parâmetro recebem o endereço de memória das variáveis do programa principal.

Desta forma, as alterações realizadas nas funções são executadas diretamente nas variáveis originais, ou seja, no endereço de memória.

Passagem por referência

Para passar um parâmetro por referência, coloca-se um asterisco "*" na frente do nome do parâmetro na declaração da função:

```
//passagem de parâmetro por valor
void incrementa(int n);

//passagem de parâmetro por referência
void incrementa(int *n);
```

Passagem por referência

Ao se chamar a função, é necessário utilizar o operador “&”, igual como é feito com a função **scanf()**:

```
//passagem de parâmetro por valor
int x = 10;
incrementa(x);

//passagem de parâmetro por referência
int x = 10;
incrementa(&x);
```

Passagem por referência

No corpo da função, é necessário usar colocar um asterisco “*” sempre que se desejar acessar o conteúdo do parâmetro passado por referência.

```
//passagem de parâmetro por valor
void incrementa(int n){
     n = n + 1;
}
//passagem de parâmetro por referência
void incrementa(int *n){
     *n = *n + 1;
}
```

Passagem por referência

```
int *n = &x;
```

```c
void incrementa(int *n){
    *n = *n + 1;

    printf("Dentro da funcao: x = %d\n",n);
}

int main(){
    int x = 5;
    printf("Antes da funcao: x = %d\n",x);

    incrementa(&x);

    printf("Depois da funcao: x = %d\n",x);
    return 0;
}
```

Saída:

```
Antes da funcao: x = 5
Dentro da funcao: x = 6
Depois da funcao: x = 6
```

Material complementar

Vídeos números 15, 16, 17, 18 e 19 disponíveis no moodle.

Referências

Backes, A. Linguagem C: Funções