

INTRODUÇÃO À PROGRAMAÇÃO

Prof.^a Priscilla Abreu

priscilla.braz@rj.senac.br



Introdução à Programação



Roteiro de Aula

- Objetivo da aula
- Funções e procedimentos
 - Passagem de parâmetro por referência

Introdução à Programação



Objetivo da aula

Implementar técnicas de programação modular utilizando a linguagem C.

REVISANDO...

Introdução à Programação



FUNÇÕES – EXEMPLO

```
#include <stdio.h>
int soma (int a, int b) {
    return ( a + b ) ;
}
int main ( ) {
    int n1, n2;
    printf ( "Entre com dois números: " ) ;
    scanf ( "%d" , &n1 ) ;
    scanf ( "%d" , &n2 ) ;
    printf ( "A soma de %d e  %d é : %d\n", n1, n2,
soma(n1, n2)) ;
}
```

Introdução à Programação



PARÂMETROS – EXEMPLO

```
#include <stdio.h>
float AreaTriangulo(int b, int a) {
    return ( a*b/2 );
}
int main ( ) {
    int base, alt ; float res;
    printf ( "Informe a base do triângulo: " ) ;
    scanf ( "%d" , &base) ;
    printf ( "Informe a altura do triângulo: " ) ;
    scanf ( "%d" , &alt) ;
    res = AreaTriangulo(base, alt) ;
    printf ( "A área do triângulo é: %f\n", res) ;
}
```

Introdução à Programação



PARÂMETROS

A passagem de parâmetros entre as sub-rotinas ocorre de duas formas:

- Passagem por valor;
- Passagem por referência.

PARÂMETROS

Parâmetros formais – passagem por valor

Na passagem por valor uma cópia do valor da variável é passado para a função.

Neste caso, a função que recebe este valor, ao fazer modificações no parâmetro, não estará alterando o valor original, que somente existe na função que a chamou.

Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

```
#include <stdio.h>
int soma_dobro (int a, int b){
    int soma;
    a = 2*a;           b = 2 *b;
    soma = a + b;
    return soma;
}
```

Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

```
int main (){  
    int x, y, res;  
    printf ("Entre com o primeiro numero: ");  
    scanf ("%d",&x);  
    printf ("Entre com o segundo numero: ");  
    scanf ("%d",&y);  
    res=soma_dobro(x, y);  
    printf ("Soma: %d",res);  
}
```

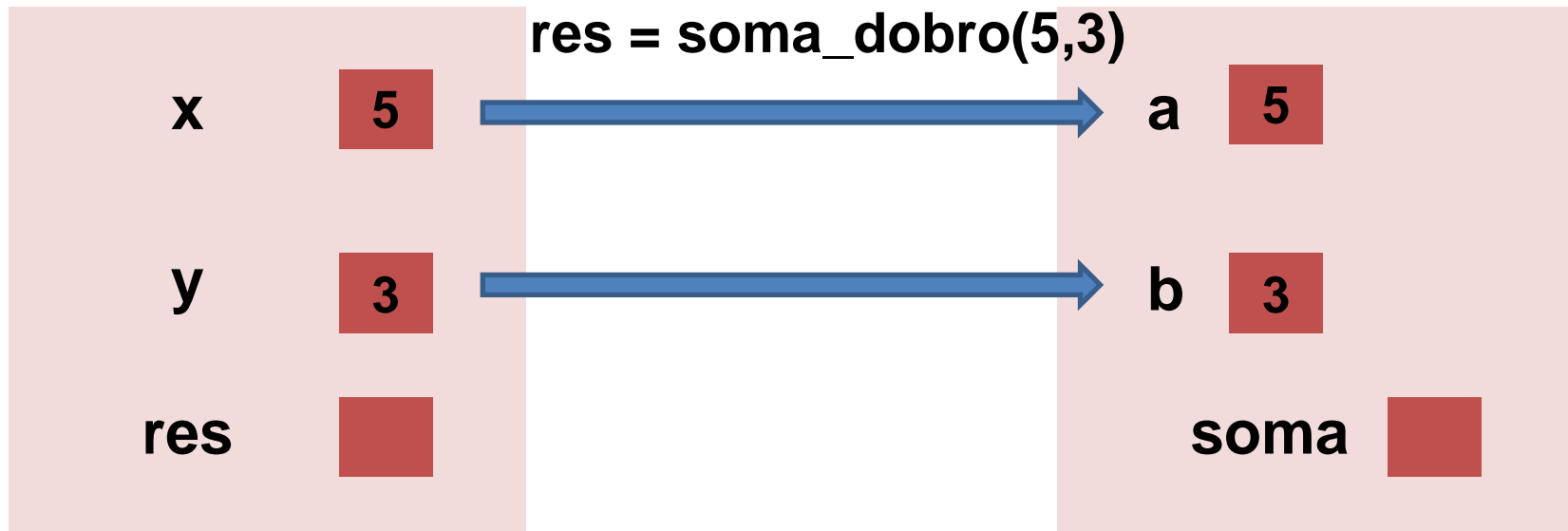
Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

Função main

Função soma_dobro



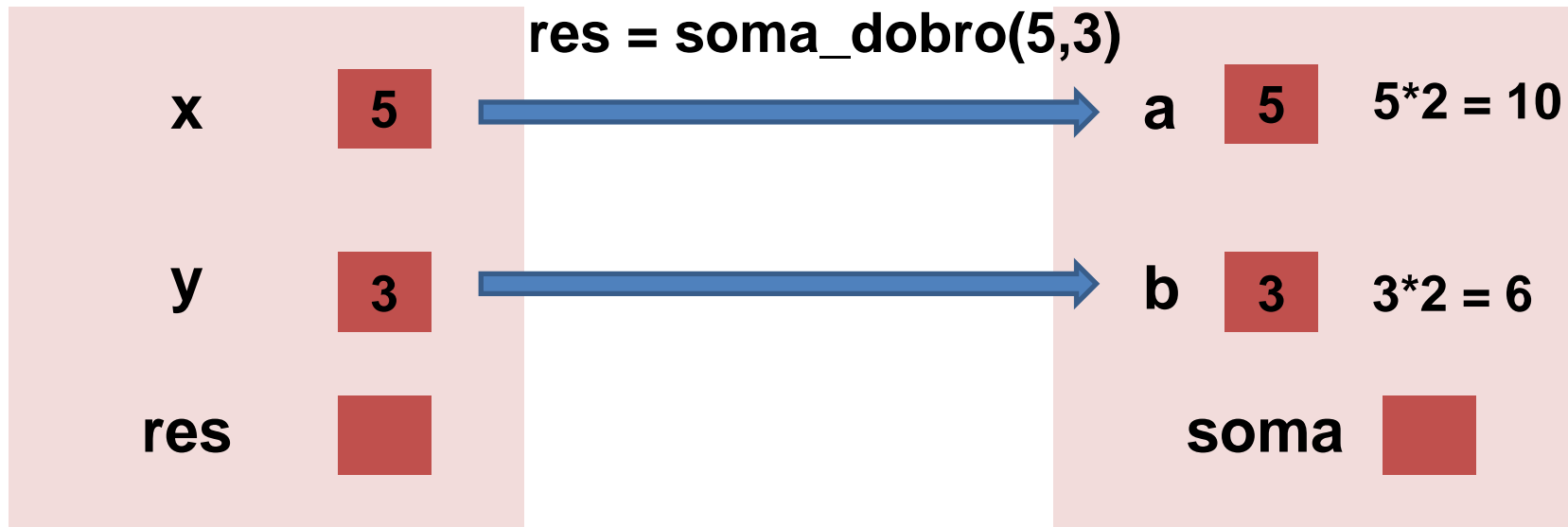
Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

Função main

Função soma_dobro



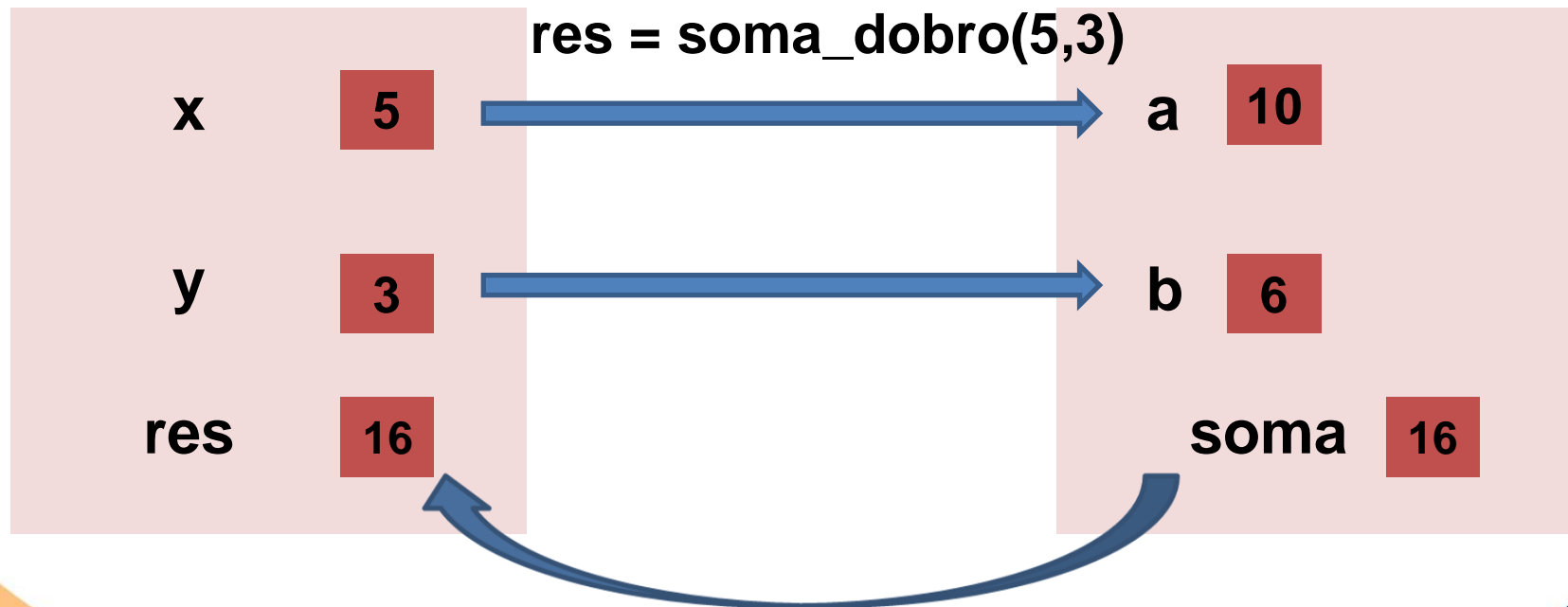
Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

Função main

Função soma_dobro



Introdução à Programação



PARÂMETROS – PASSAGEM POR VALOR

Função main

| | |
|------------|-----------|
| x | 5 |
| y | 3 |
| res | 16 |

Função soma_dobro

Introdução à Programação



PARÂMETROS

Agora, imagine a seguinte situação:

Um programa que recebe dois valores do usuário, armazena-os em duas variáveis e troca os valores entre si.

Introdução à Programação



PARÂMETROS

```
#include <stdio.h>
int main(){
    int a, b, aux;
    printf("Informe dois números: ");
    scanf("%d %d", &a, &b);
    printf("Antes da troca: %d - %d\n",a,b);
    //troca
    aux = a;
    a = b;
    b = aux;
    printf("Após a troca: %d - %d\n",a,b);
}
```


É possível construir uma função que executa essa operação?

Introdução à Programação



PARÂMETROS

```
#include <stdio.h>
void troca(int x,int y){
    int aux;
    aux = x;
    x=y;
    y= aux;
}
```

```
int main(){
    int a, b;
    printf("Informe dois números:
");
    scanf("%d", &a);
    scanf("%d", &b);
    printf("Antes da troca:  %d  -
%d\n", a,b);
    troca(a,b);
    printf("Após a troca:   %d  -
%d\n", a,b);
}
```

**Como podemos verificar, esta
função não funciona, pois
passamos os parâmetros por
valor!**

Introdução à Programação



PARÂMETROS

Para resolver esse problema, uma solução seria passar para a função troca uma referência às variáveis a e b que estão na memória, isto é, o endereço delas. Assim, a função poderia alterar diretamente os valores de a e b!

Chamamos esta característica de **passagem de parâmetros por referência**;

Mas como faremos isso?

Introdução à Programação



PARÂMETROS

Relembrando nossas primeiras aulas...

Cada variável tem a ela associado:

- Um tipo de dado;
- Um nome (identificador);
- Um endereço de memória;
- Um valor.

Como fazer uma variável receber a referência (endereço) de outra?

Uso do operador &, o mesmo utilizado no comando
`scanf("%d",&num);`

Introdução à Programação



PARÂMETROS

```
#include <stdio.h>  
void troca(int x,int y){  
    int aux;  
    aux=x;  
    x=y;  
    y=aux;  
}
```



```
int main(){  
    int a, b;  
    printf("Informe dois números: ");  
    scanf("%d %d", &a, &b);  
    printf("Antes: %d - %d\n",a,b);  
    //troca  
    troca(&a, &b);  
    printf("Após: %d - %d\n",a,b);  
}
```

Os parâmetros definidos na função precisam ser de um tipo de dado que aceite armazenar endereços de memória de outras variáveis

Uso de ponteiros

PARÂMETROS

Parâmetros – passagem por referência

O que é passado para a função é o endereço da variável. Portanto, a função pode modificar o valor da variável através do endereço.

Uso de ponteiros: operadores * e &.

& - retorna o endereço de uma variável na memória.

* - retorna o conteúdo armazenado no endereço referenciado pelo ponteiro.

Introdução à Programação



PARÂMETROS

Ponteiros

```
#include <stdio.h>
void troca(int *x,int *y){
    int aux;
    aux = *x;
    *x = *y;
    *y = aux;
}
```



```
int main(){
    int a, b;
    printf("Informe dois números: ");
    scanf("%d %d", &a, &b);
    printf("Antes: %d - %d\n",a,b);
    //troca
    troca(&a, &b);
    printf("Após: %d - %d\n",a,b);
}
```

Introdução à Programação



PARÂMETROS

Função main

a

5

800

b

3

300

troca(&a, &b)

Função troca

aux



x



y



Introdução à Programação



PARÂMETROS

Função main

a

5

800

b

3

300

troca(800, 300)

Função troca

aux



x



y



Introdução à Programação



PARÂMETROS

Função main

a

5

800

b

3

300

troca(800, 300)

Função troca

aux

x

800

y

300

Introdução à Programação



PARÂMETROS

Função main

a

5
800

b

3
300

troca(800, 300)

Função troca

aux

aux = *x

x

800

y

300

Introdução à Programação



PARÂMETROS

Função main

a **5**
800

b **3**
300

troca(800, 300)

Função troca

aux **5** **aux = *x**

→ **x** **800**

→ **y** **300**

Introdução à Programação



PARÂMETROS

Função main

a ~~5~~
800

b 3
300

troca(800, 300)

Função troca

aux 5 **aux = *x**

x 800 ***x = *y**

y 300

Introdução à Programação



PARÂMETROS

Função main

a **3**
800

b **3**
300

troca(800, 300)

Função troca

aux **5** **aux = *x**

x **800** ***x = *y**

y **300**

Introdução à Programação



PARÂMETROS

Função main

a 3
800

b ~~3~~
300

troca(800, 300)

Função troca

aux 5 **aux = *x**

x 800 ***x = *y**

y 300 ***y = aux**

Introdução à Programação



PARÂMETROS

Função main

a **3**
800

b **5**
300

troca(800, 300)

Função troca

aux **5** **aux = *x**

x **800** ***x = *y**

y **300** ***y = aux**

Introdução à Programação



PARÂMETROS

Função main

a

3

800

b

5

300

troca(800, 300)

Função troca

Introdução à Programação



USO DA PASSAGEM POR REFERÊNCIA

A passagem por referência também pode ser utilizada em casos em que é necessário o retorno de mais de um resultado por uma função.

Exemplo:

Faça um programa que leia o lado de um quadrado e utilize uma mesma função para calcular e retornar a área e o perímetro do quadrado sem o uso de variáveis globais.

Introdução à Programação



USO DA PASSAGEM POR REFERÊNCIA

```
#include <stdio.h>
int CalcQuad(int lado, int *area){
    int p;
    *area=lado*lado;
    p=lado*4;
    return p;
}
int main(){
    int l, a,p;
    printf("Informe o lado do quadrado: ");
    scanf("%d",&l);
    p=CalcQuad(l,&a);
    printf("Lado: %d\n Perímetro: %d\n Área:%d",l,p,a);
}
```

Introdução à Programação



VERSÃO COM PROCEDIMENTO

```
#include <stdio.h>
void calcula(int l, int *a, int *p){
    *a = l*l;
    *p = 4 * l;
}
int main(){
    int lado, area, per;
    printf("Lado: "); scanf("%d",&lado);
    calcula(lado, &area, &per);
    printf("A área do quadrado e: %d\n", area);
    printf("O perímetro do quadrado e: %d\n", per);
}
```

Introdução à Programação



CONSIDERAÇÕES...

Os parâmetros são um importante recurso para possibilitar a comunicação entre diferentes funções sobre suas variáveis;

A passagem de parâmetro por valor passa uma cópia do valor de uma determinada variável para a referida função.

Já a passagem de parâmetro por referência passa o endereço de uma variável, permitindo que a função possa manipular esta variável que está declarada em outra função.

Introdução à Programação



PROGRAMAÇÃO MODULAR

A programação modular é uma importante técnica de Programação Estruturada que nos permite organizar nosso código em módulos ou trechos, que podem ser reutilizados;

Pode ser implementada através de procedimentos e funções.