

MINIAULA DE ALGORITMOS TIPOS DE PASSAGEM DE PARÂMETROS

Prof. Ivanilton Polato

Departamento Acadêmico de Computação (DACOM-CM)

ipolato@utfpr.edu.br

Passagem de parâmetros: 2 maneiras!

- Passagem de parâmetros por valor: a função trabalhará com cópias dos valores passados no momento de sua chamada.
 - *Foi este tipo de passagem que utilizamos até o momento. Neste esquema quando a função termina suas variáveis locais são perdidas.*
- Passagem de parâmetros por referência: nesta modalidade os parâmetros passados para uma função correspondem a endereços de memória ocupados pelas variáveis.
 - *Toda vez que for necessário acessar o valor de uma variável, isso será feito por meio de referência, ou seja, apontando ao seu endereço.*
 - *Ainda, toda vez que a função modifica uma variável, estamos modificando o valor no endereço da variável original, ou seja, o valor original é modificado!*

Passagem de parâmetros: tipos!

- Por valor:

```
res = soma (x, y) ;
```

- Por referência:

```
res = soma (&x, &y) ;
```

- A diferença é o símbolo **&** que indica o endereço da variável, ao invés de seu conteúdo!
- Mas existem diferenças na declaração dos parâmetros na função também!

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Exemplo: soma dobro!

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

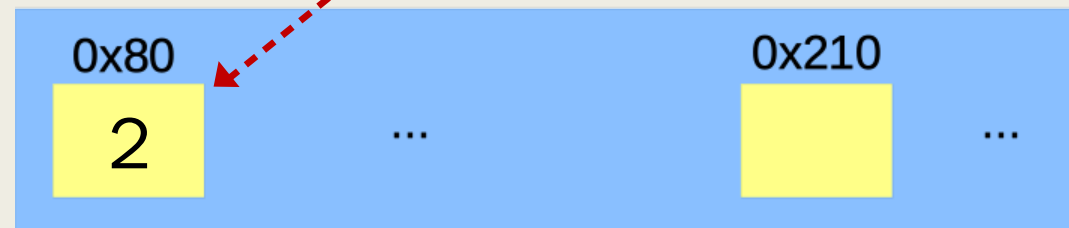
```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	2	

Memória Principal (com endereços físicos)



O usuário digita
o valor 2!

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	2	3

Memória Principal (com endereços físicos)

0x80		0x210
2	...	3

O usuário digita
o valor 3!

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Variáveis da soma_dobro()

soma	*n1	*n2
	0x80	0x210

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	2	3

Memória Principal (com endereços físicos)

0x80		0x210
2	...	3

A função soma_dobro é chamada!
Ao invés de valores, são enviados os
endereços das variáveis!

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Na função soma_dobro quando **n1** é manipulado, estamos manipulando o conteúdo de x, através do endereço dele! Logo, estamos dobrando o valor de x!

Variáveis da soma_dobro()

soma	*n1	*n2
	0x80	0x210

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	4	3

Memória Principal (com endereços físicos)

0x80	...	0x210	...
4		3	

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

O mesmo vale para n2,
que altera o valor do
endereço da variável y!

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Variáveis da soma_dobro()

soma	*n1	*n2
	0x80	0x210

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	4	6

Memória Principal (com endereços físicos)

0x80		0x210
4	...	6

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

soma recebe a soma dos conteúdos de n1 e n2, que apontam, respectivamente, para x e y!

Variáveis da soma_dobro()

soma	*n1	*n2
10	0x80	0x210

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
	4	6

Memória Principal (com endereços físicos)

0x80		0x210
4	...	6

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Na instrução return, o valor de soma é copiado para res, antes que seja destruído!

Variáveis da soma_dobro()

soma	*n1	*n2
10	0x80	0x210

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
10	4	6

Memória Principal (com endereços físicos)

0x80	...	0x210	...
4		6	

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
10	4	6

Memória Principal (com endereços físicos)

0x80		0x210
4	...	6

A função termina e o controle volta para o main(), destruindo o conteúdo de soma_dobro!

Exemplo: soma dobro!

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

```
int soma_dobro(int *n1, int *n2) {  
    int soma;  
    *n1 = 2 * (*n1);  
    *n2 = 2 * (*n2);  
    soma = *n1 + *n2;  
    return soma;  
}
```

```
void main () {  
    int x, y, res;  
    printf("Digite o primeiro valor: \n");  
    scanf("%d", &x);  
    printf("Digite o segundo valor: \n");  
    scanf("%d", &y);  
    res = soma_dobro(&x, &y);  
    printf("A soma entre %d e %d é %d\n", x, y, res);  
}
```

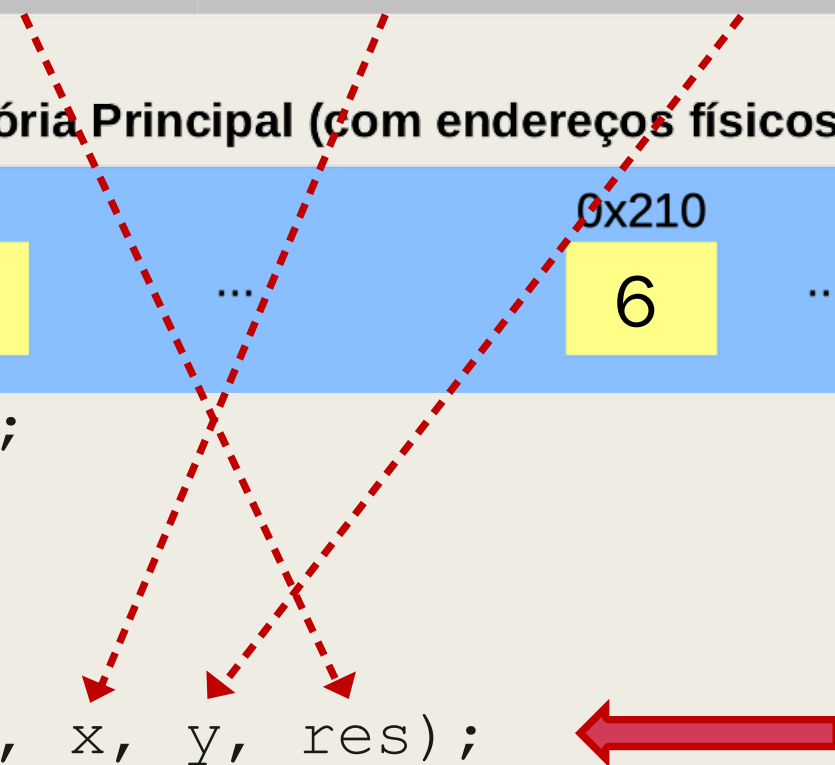
A função main() termina com a impressão em tela dos valores, incluindo aqueles que foram modificados pela função soma_dobro!

Variáveis do main() (nome/endereço físico)

res (0x300)	x (0x80)	y (0x210)
10	4	6

Memória Principal (com endereços físicos)

0x80	...	0x210	...
4		6	



Lembrete!

- Na passagem por referência:
 - *Na declaração, utiliza-se o * nos parâmetros*

```
int soma_dobro (int *n1, int *n2) { ... }
```

- *Na chamada, utiliza-se o & nos parâmetros*

```
res = soma_dobro (&x, &y) ;
```