

Algoritmo e Programação

Funções e Passagem de Parâmetros

Prof. Dr. Gilberto Fernandes Jr.





Passagem de Parâmetros por Referência

- Passagem por referência com ponteiros;
- Passagem de vetores e matrizes;
- Passagem de structs por referência

Passagem de parâmetros **por referência**

- O uso de funções que passam parâmetros por referência está relacionado aos conceitos de **ponteiros** e **endereços** de memória
 - Definição da função

```
int testar(int* parametro1, int*  
parametro2)
```

- Chamada da função:

```
resultado = testar(&n1, &n2)
```



Exemplo!

```
void main(){
```

```
    int var = 10;
```

```
    funcao( &var );
```

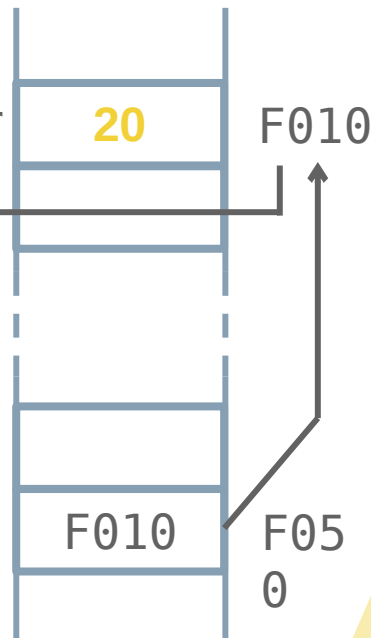
```
}
```

```
void funcao (int *n){
```

```
    *n = 20;
```

```
}
```

O valor de **var** é alterado



Passagem de vetores

- A passagem de um vetor é sempre realizada implicitamente por referência

- Definição da função (duas possibilidades)

```
int testar1( int v1[], int v2[] ) {...}  
int testar2( int* v1, int* v2 ) {...}
```

- Chamada da função (Não usa &):

```
resultado1 = testar1( n1, n2 );  
resultado2 = testar2( n1, n2 );
```



Passagem de matrizes

- A passagem de uma matriz também é sempre realizada implicitamente por referência

- Definição da função (duas possibilidades)

```
testeMat1( int mat[2][3] ) {...}  
testeMat2( int mat[][3] ) {...}
```

- Chamada da função (Não usa &):

```
testeMat1(m1); // m1 é uma matriz 2x3  
testeMat2(m1);  
testeMat2(m2); // m2 é uma matriz 5x3
```



Passagem de structs

- Dentro da função, utilizamos o operador "->" para referenciar os campos da estrutura passada por referência com ponteiros

- Definição da função:

```
void registra(struct Teste *var) {  
    var->a = 10;  
}
```

- Chamada da função:

```
registra(&t1); // t1 é uma struct Teste
```





Vamos Praticar?

Vamos utilizar o Visual Studio Code para construir alguns algoritmos com passagem por referência na linguagem C