

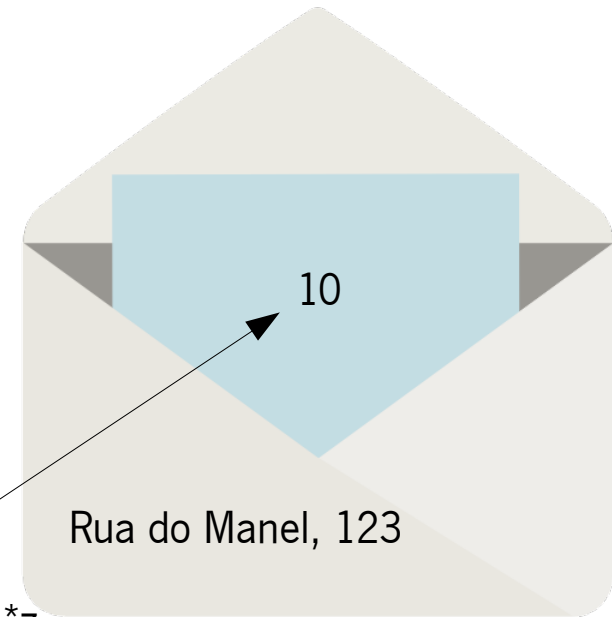
PASSEI – MP2 – 1ª Sessão

Apontadores



Analogia

Consideremos uma carta como sendo uma variável. Denominemos a carta de **m** e o seu endereço é guardado num apontador **z**.



Conteúdo da Carta = $m = *z$

Endereço Destino da Carta = $\&m = z$

O que é na realidade um apontador?

- É uma variável que guarda um endereço de memória de outra variável, ou simplesmente um endereço de memória.

Exemplo:

```
int m = 10; int *z;
```

`z = &m;` (a variável **z** vai guardar o endereço de memória da variável **m**)

O que é na realidade um apontador?

- O que podemos ver nos vários blocos de memória do computador:

Para a variável **m**:

- 10 - 0X71234 (endereço de memória de **m**)

Para a variável **z**:

- 0x71234 - 0x74567 (endereço de memória de **z**)

Como obter o conteúdo de um apontador?

- O operador * (asterisco) permite-nos obter o conteúdo de um apontador.

Pegando no exemplo anterior:

```
int m=10;

int *z = &m;

printf("Conteúdo de z: %d \n",*z);
printf("Valor de z: %p \n",z);
printf("Endereço de m: %p \n",&m);
printf("Valor de m: %d \n",m);
```

Output:

```
Conteúdo de z: 10
Valor de z: 0x765423
Endereço de m: 0x765423
Valor de m: 10
```

Exercício 1

- Escreva um programa que demonstre a utilização dos operadores **&** (endereço de) e ***** (valor no endereço).

Output:

m = 300

fx = 300.600006

cht = z

Utilizando somente o operador &:

endereço de m = 0x7ffda2eeeeec8

endereço de fx = 0x7ffda2eeeecc

endereço de cht = 0x7ffda2eeeeec7

Utilizando os operadores & e *:

valor no endereço de m = 300

valor no endereço de fx = 300.600006

valor no endereço de cht = z

Utilizando somente os apontadores declarados:

endereço de m = 0x7ffda2eeeeec8

endereço de fx = 0x7ffda2eeeecc

endereço de cht = 0x7ffda2eeeeec7

Utilizando somente o operador*:

valor no endereço de m = 300

valor no endereço de fx = 300.600006

valor no endereço de cht = z

Exercício 2

- Escreva um programa que some dois n^os inteiros utilizando apontadores.

Input:

Primeiro número: 5

Segundo número: 6

Output:

Resultado: 11

Exercício 3

- Escreva um programa que calcule a soma de todos os elementos de um array de inteiros utilizando apontadores.

Input:

Nº de Elementos do Array (Máx 10): 5

Elemento 1 : 2

Elemento 2 : 3

Elemento 3 : 4

Elemento 4 : 5

Elemento 5 : 6

Output:

Soma: 20

Exercício 4

- Escreva um programa que imprima uma string ao contrário utilizando apontadores.

Input:

Introduza a string: netium

Output:

Resultado: muiten

Passagem por valor

- Quando utilizamos a chamada de uma função em que lhe passamos variáveis que não são apontadores denominamos **passagem por valor**.

Exemplo:

```
int a = 10;
```

```
int b = 11;
```

```
int sum = somar(a,b);
```

```
int somar(int a,int b){
```

```
    int x = a+b
```

```
    return x;
```

```
}
```

Passagem por referência

- Quando pretendemos alterar o valor de uma variável e para tal passamos o seu endereço à função.

Exemplo:

```
int a = 5;          void somar(int a,int b,int *c){  
int b = 6;          *c = a+b;  
int c = 0;          }  
somar(a,b,&c);
```

Exercício 5

- Escreva um programa que troque o valor entre 3 números utilizando a passagem por referência. Desenvolva uma função para esse efeito denominada **swapNumbers**.

Input:

Introduza o 1º número: 5

Introduza o 2º número: 6

Introduza o 3º número: 7

Output:

Antes da inversão

1º número: 5

2º número: 6

3º número: 7

Depois da inversão

1º número: 7

2º número: 5

3º número: 6