Linguagem de Programação 2

Ponteiros

Prof. Flávio José Mendes Coelho

Ponteiros

&

operador de endereço

Fornece o endereço de memória de uma variável.

operador de derreferência

Declara uma variável do tipo de dado "**ponteiro para** algum tipo de dado".

operador de derreferência

Uma variável do tipo ponteiro guarda o **endereço de memória** de uma outra variável.

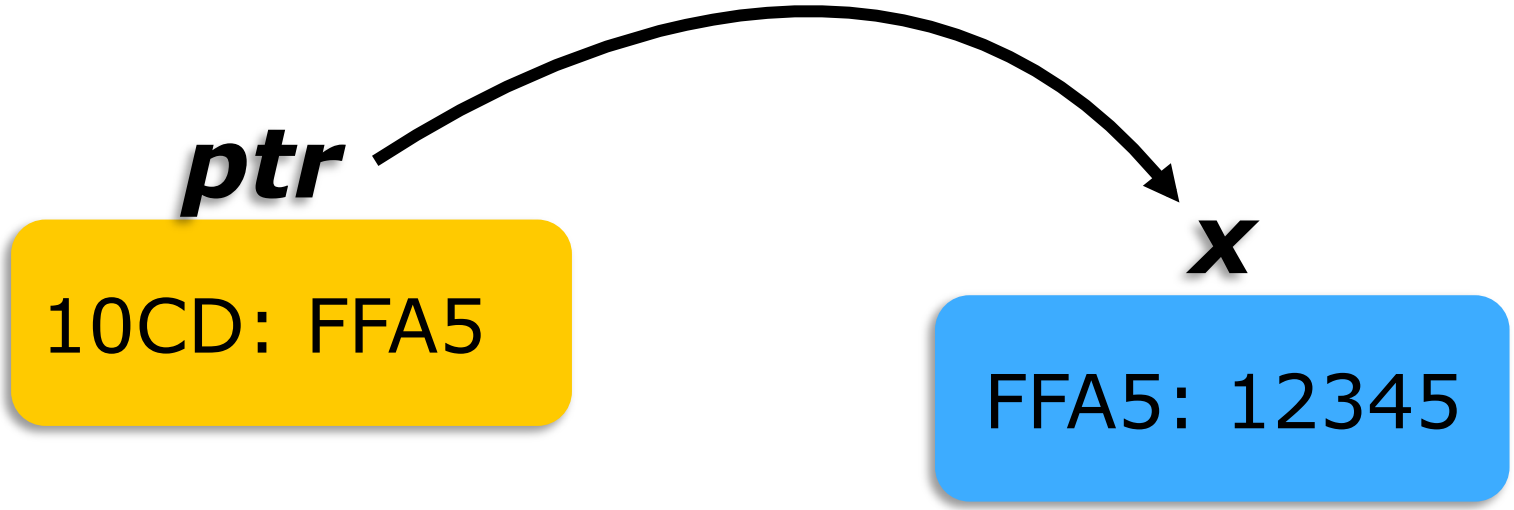
operador de derreferência

Quando um variável ***ptr*** é do tipo ponteiro e armazena o endereço de uma outra variável ***x***...

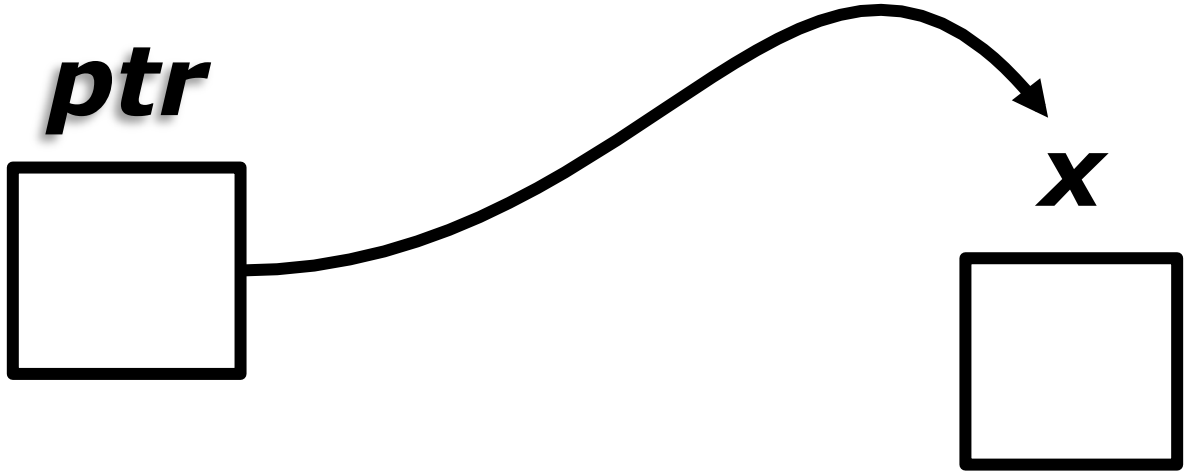
operador de derreferência

...dizemos que "***ptr*** aponta para ***x***".

operador de derreferência

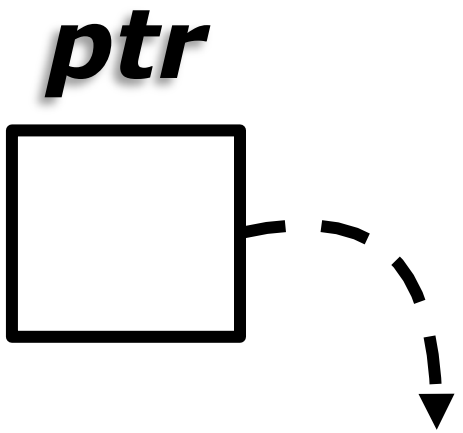


operador de derreferência

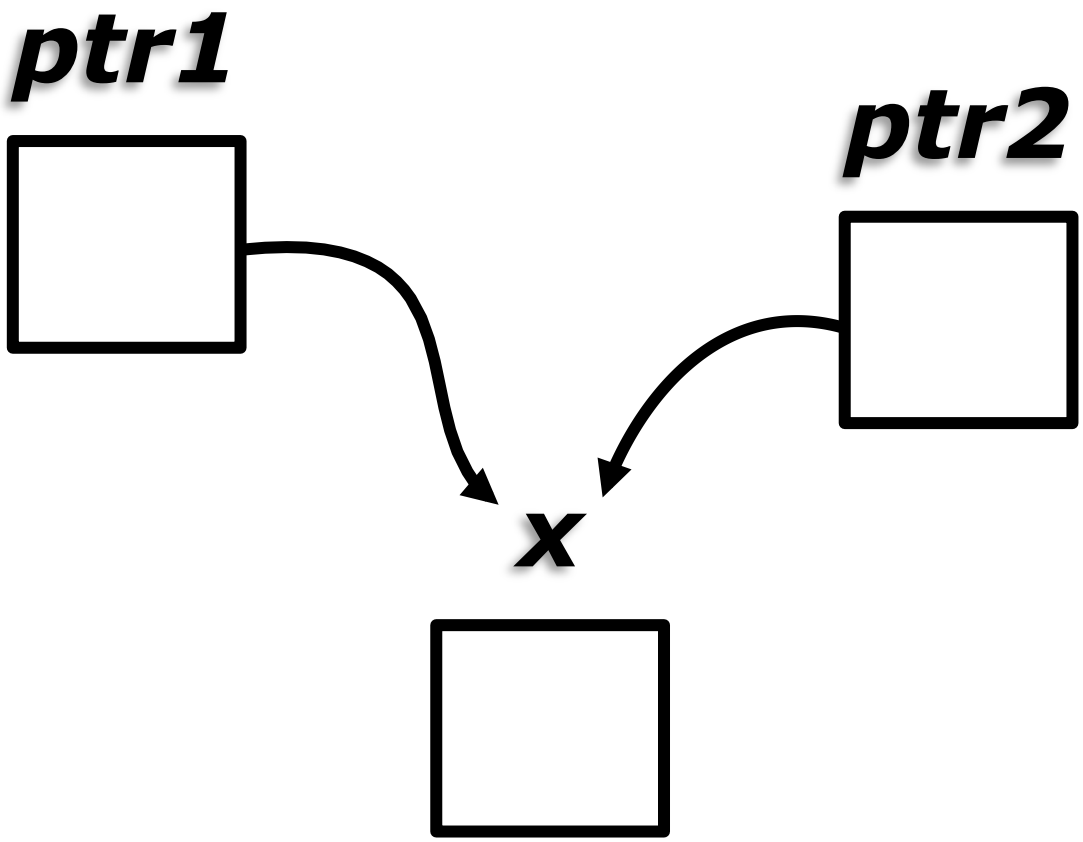


operador de derreferência

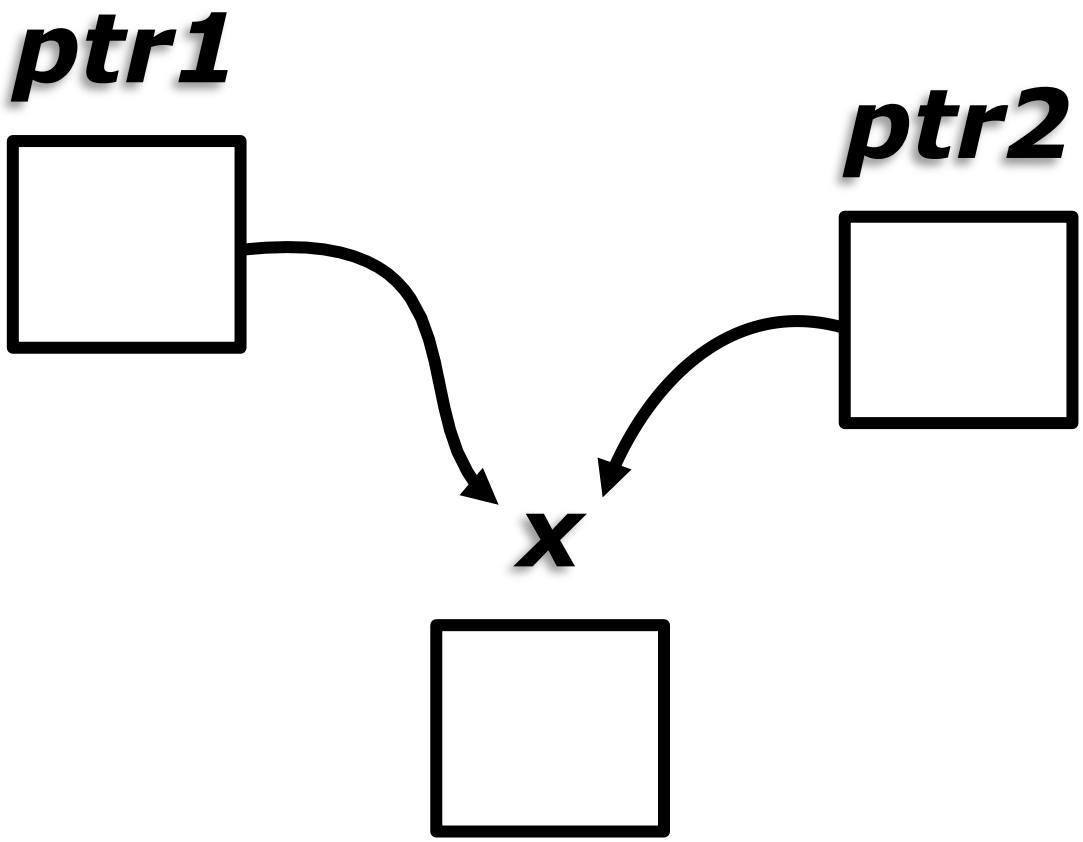
Quando um ponteiro ***ptr*** aponta para uma variável ***x***, é possível acessar/ modificar o valor de ***x*** por meio de ***ptr***.



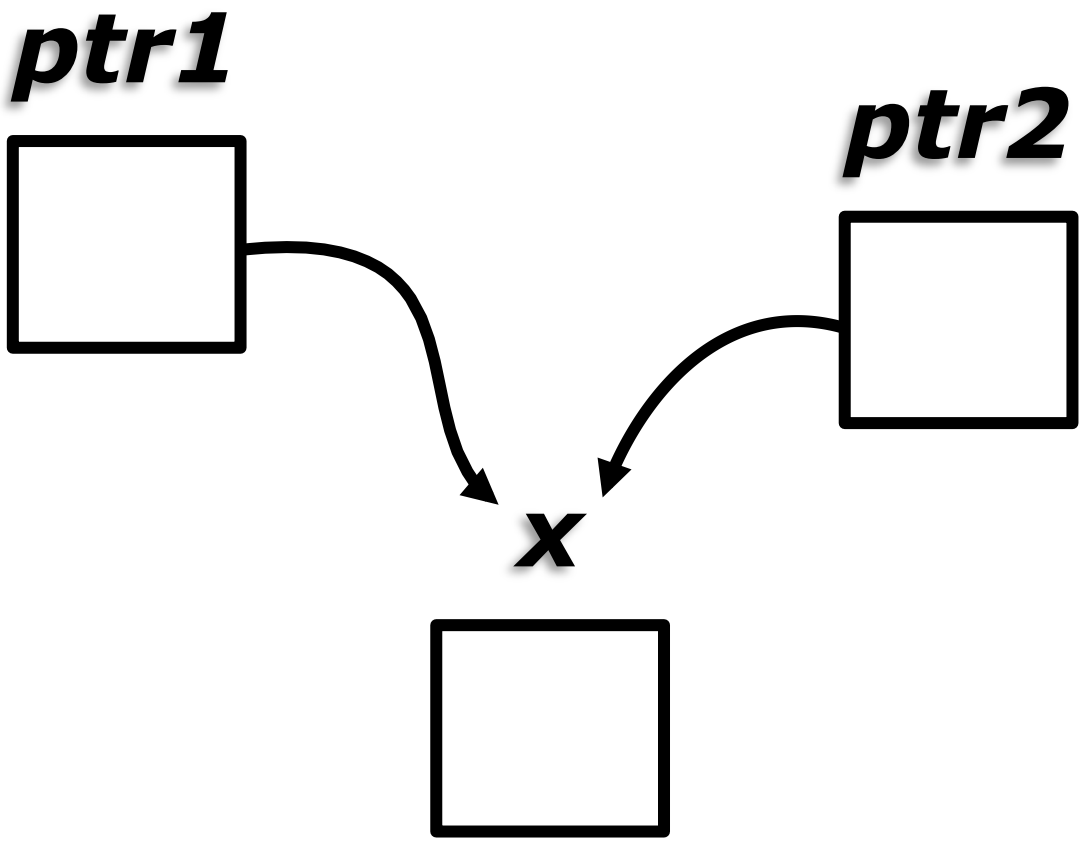
É seguro inicializar um ponteiro com **0** ou **nullptr**. Assim o ponteiro não aponta para lugar algum.



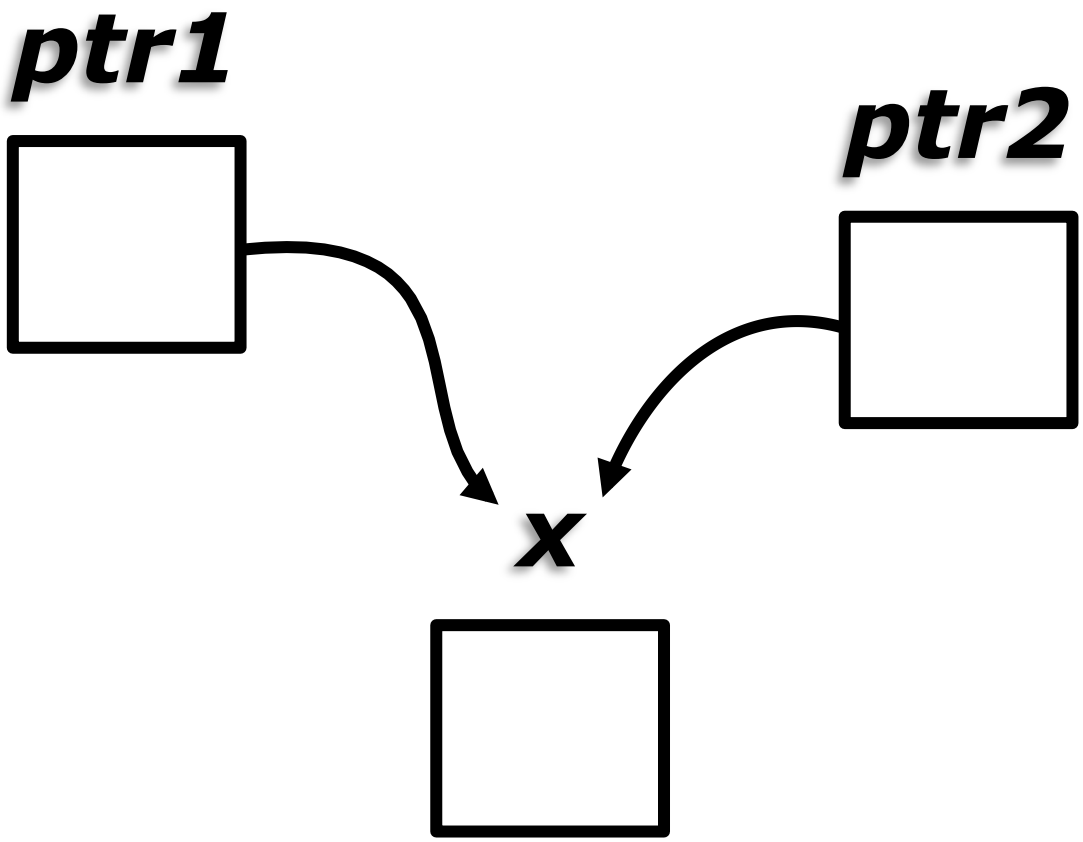
Dois ou mais ponteiros podem apontar para uma mesma variável.



Portanto, também podem acessá-la e modificar seu valor.

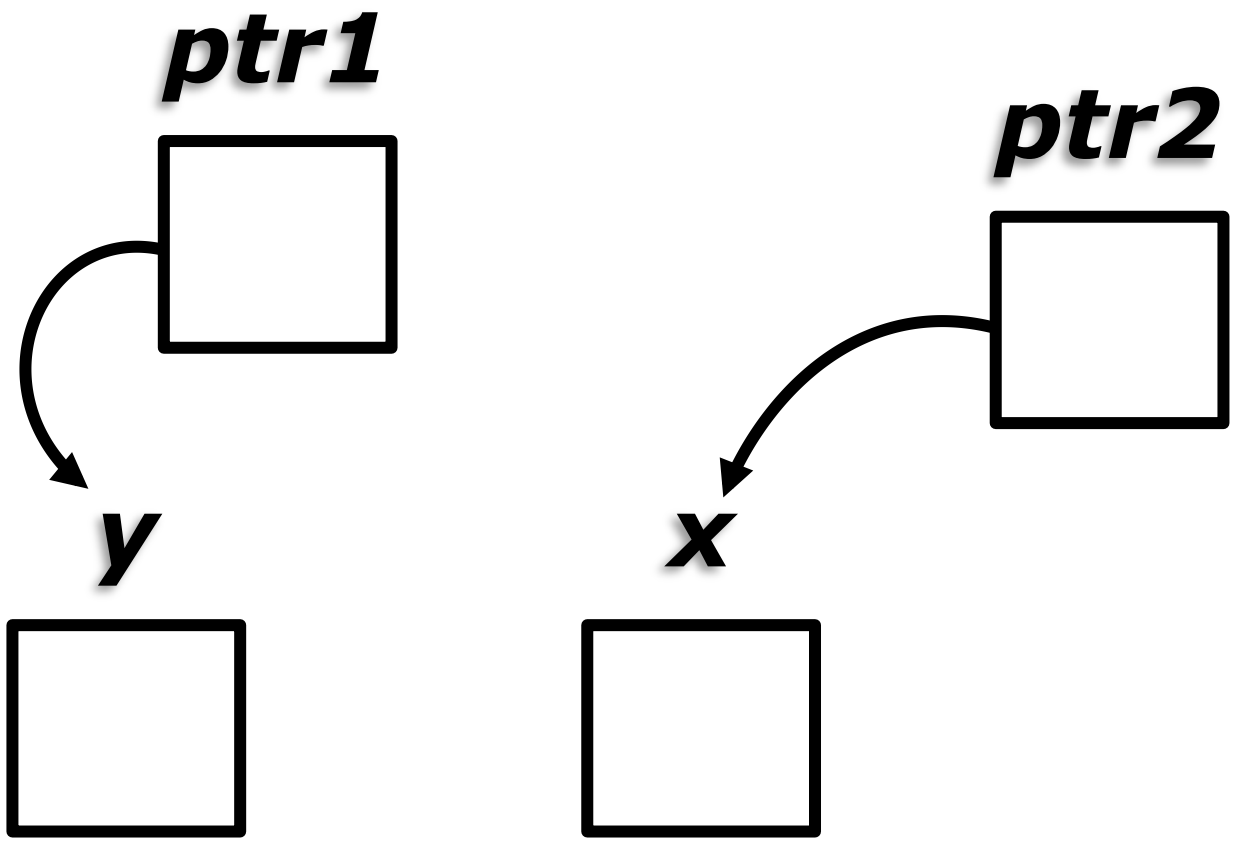



```
ptr2 = &x;
ptr1 = &x;
```

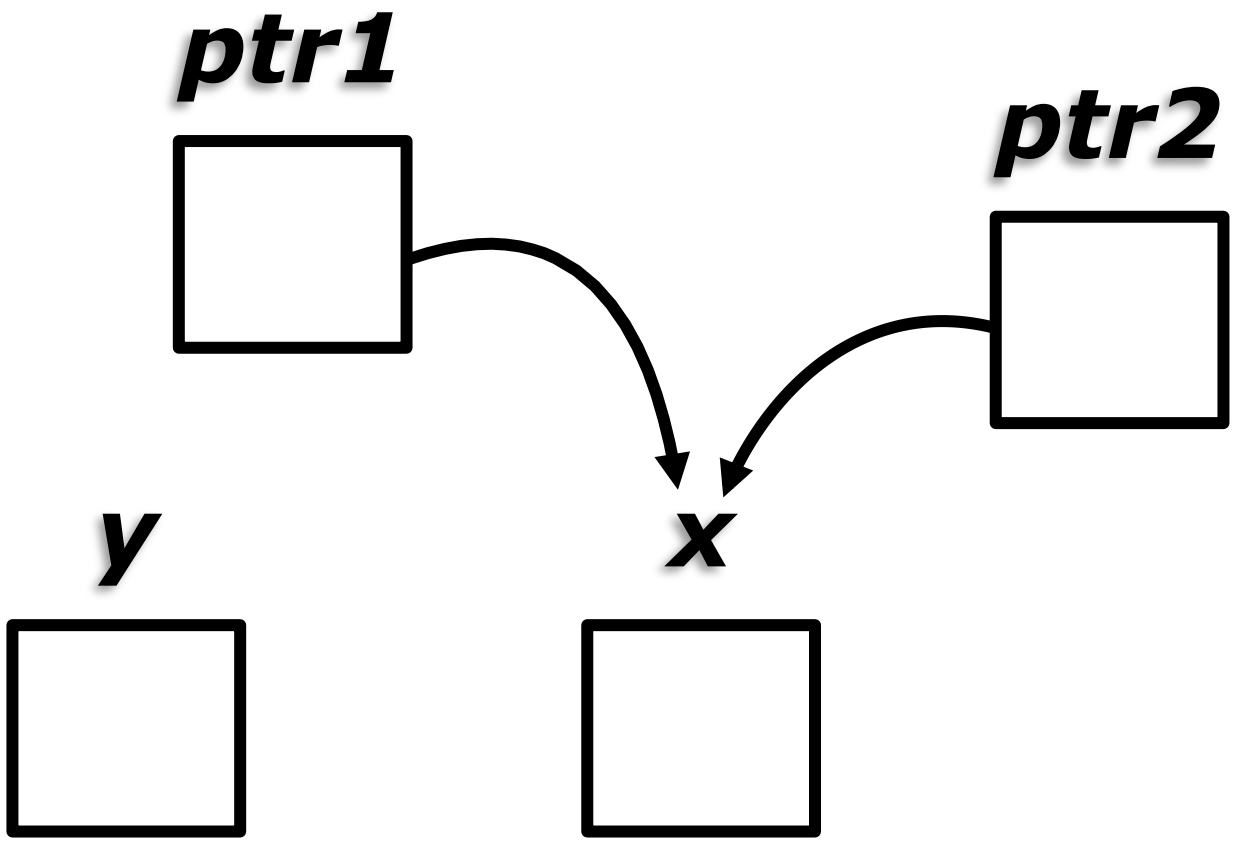


ou... ptr2 = &x;

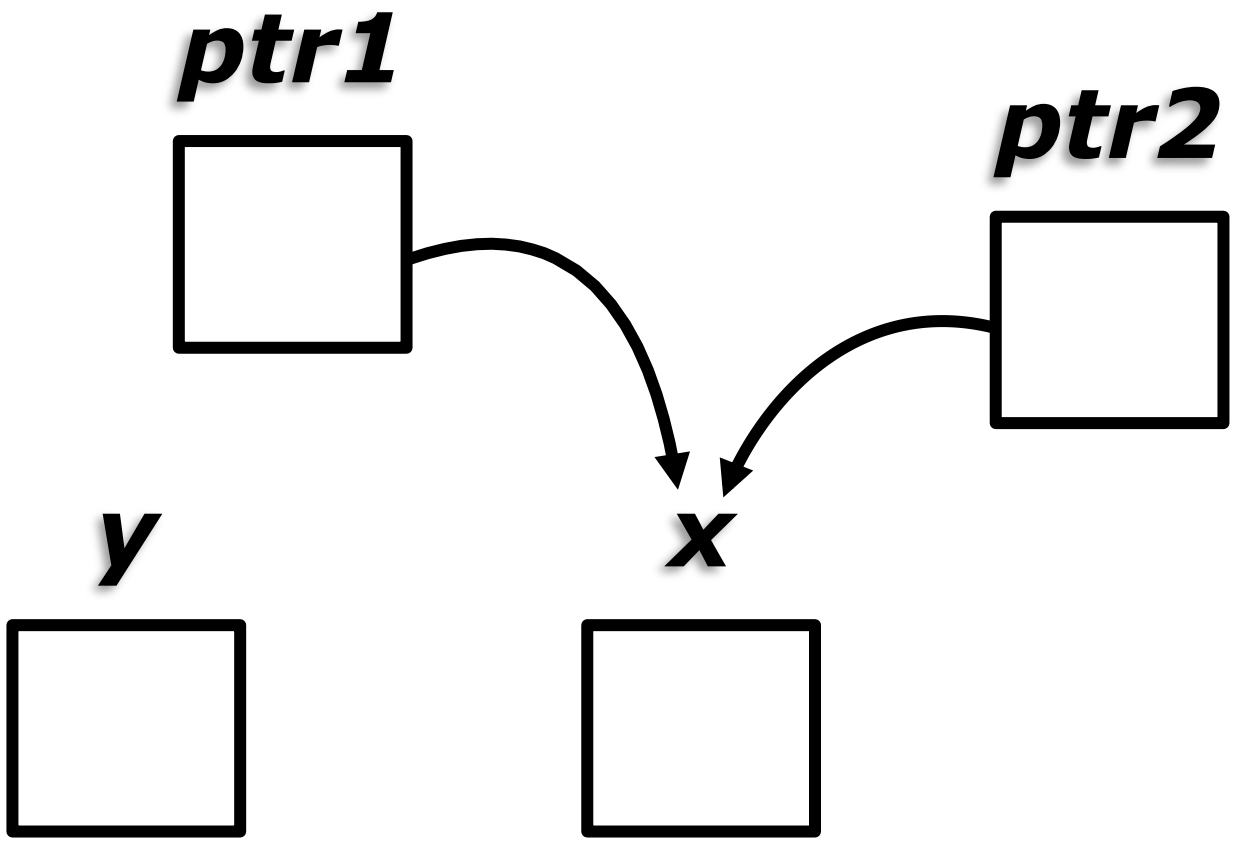
ptr1 = ptr2;



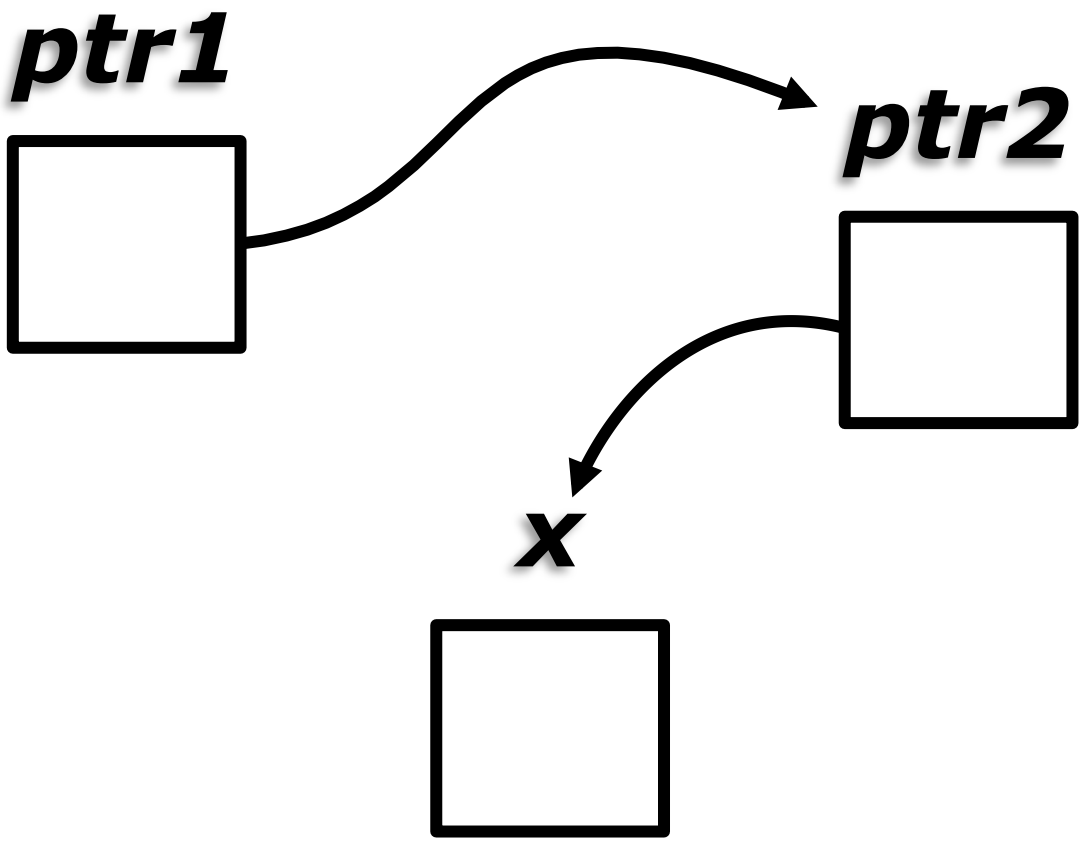
Um ponteiro para um variável, pode passar a apontar para...



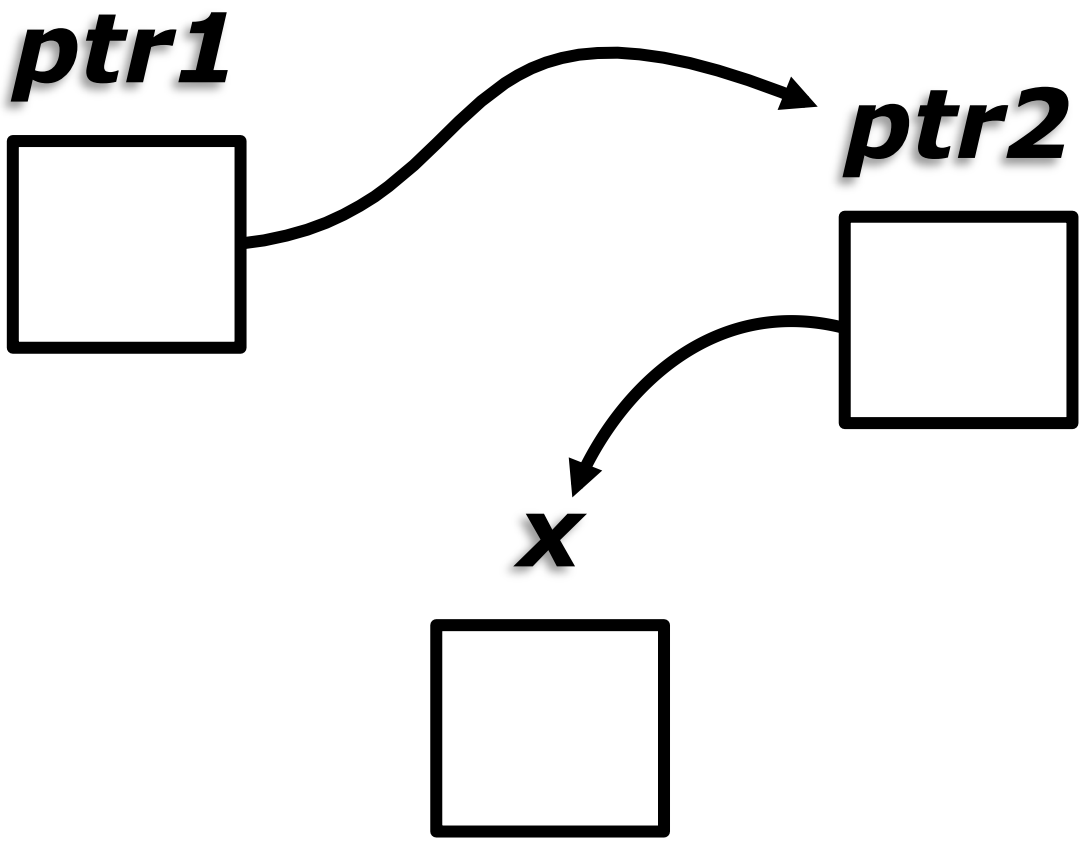
... outra variável (apontando para onde aponta um ponteiro para esta variável).



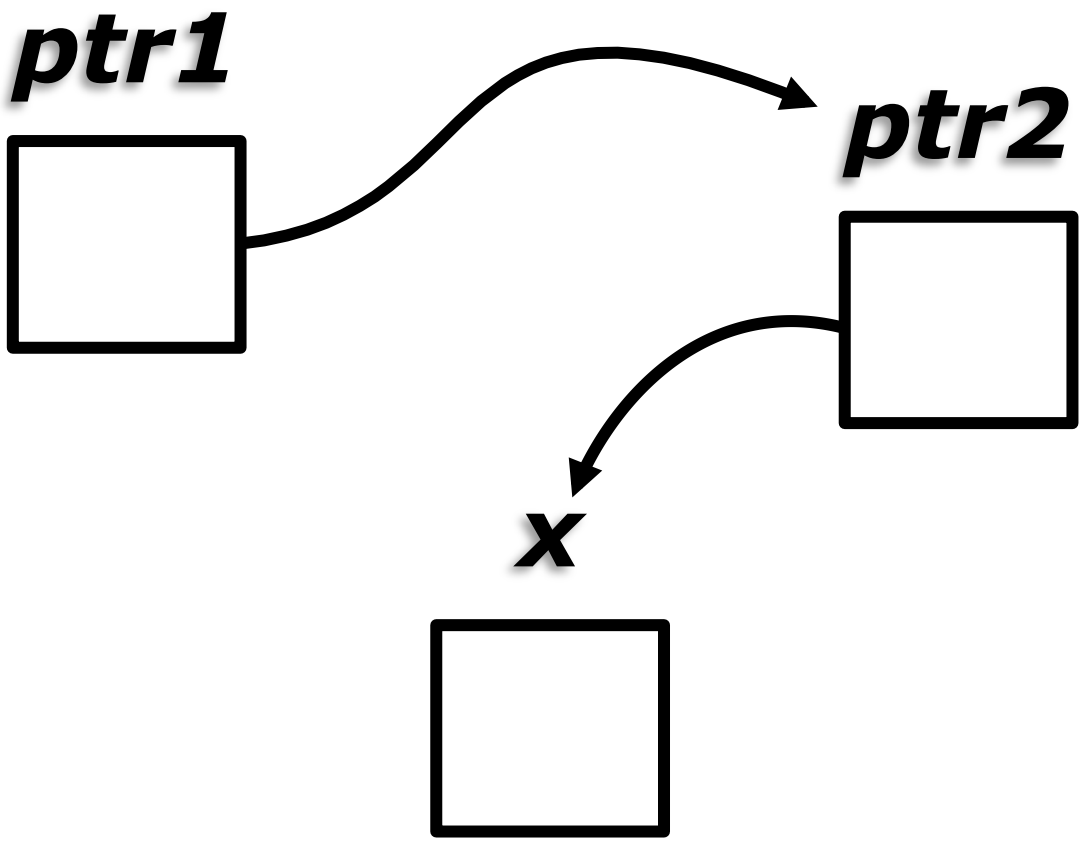

```
ptr1 = &y; ptr2 = &x; ptr1 = ptr2;
printf("%d", *prt1); // imprime x
```



Um ponteiro ***ptr1*** pode apontar para um outro ponteiro ***ptr2*** que aponta para uma variável ***x***.



Neste caso temos um **ponteiro para ponteiro** e usamos dois operadores de derreferência: int **ptr1 = 0;




```
int *ptr2 = &x;
int **ptr1 = &ptr2; printf("%d",
**ptr1);
```

```
int *ptrInt = 0
float *ptrFloat = 0
char *ponteiro = 0
Aluno *a = 0
```

Podemos declarar ponteiros para todos os tipos primitivos do C/C++, para structs, ponteiros, inclusive para tipos definidos pelo programador.

Bibliografia

- Deitel, H. M., Deitel, P. J. C - Como Programar. 6a. ed. Pearson, 2011.