

Linguagens de Programação 1

Francisco Sant'Anna

Sala 6020-B

`francisco@ime.uerj.br`

`http://github.com/fsantanna-uerj/LP1`

Ponteiros

Apontadores

Pointers

R-value vs L-value

```
int x;
```

```
x = 10;
```

```
...
```

```
int y;
```

```
y = x;
```

```
...
```

```
x = 100;
```

endereço	id	valor
	x	
	y	
	...	

r-value
vs
l-value

Endereço '&'

```
int x = 10;
```

```
int y = 99;
```

```
...
```

endereço	id	valor
	x	
	y	
	...	

```
printf ("%d %d\n", x, y);
```

```
printf ("%p %p\n", &x, &y);
```

Ponteiro '*'

```
int    x    = 10;  
int*   px   = &x;  
... 
```

endereço	id	valor
	x	
	px	
	...	

```
printf ("%p %p\n", &x, px);
```

```
printf ("%d %d\n", x, *px);
```

```
*px = 100;
```

C - Primeiros Passos

```
// 02-num.c

#include <stdio.h>

int main (void) {
    int num;
    printf("Escolha um numero: ");
    scanf("%d", &num);
    printf("Voce escolheu %d\n", num);
    return 0;
}
```

```
$ gcc 02-num.c -o num.exe
$ ./num.exe
```

```
# 02-num.py

print("Escolha um numero:")
num = input()
print("Voce escolheu", num)
```

Resumo

- Um ponteiro guarda um endereço de memória
- Declaração de um ponteiro

- Tipo seguido de '*':

- `int* p;`

- Endereço de uma variável (referência)

- Identificador precedido por '&':

- `int x = 10;`

- `int* p = &x;`

- Conteúdo de um ponteiro (dereferência)

- Identificador precedido por '*':

- `int x = 10;`

- `int* p = &x;`

- `printf("%d\n", *p); // lê 10`

- `*p = 100; // escreve 100`

Tipos

int x;

int* p;

expressão	tipo
x	
p	
&x	
&p	
*x	
*p	

atribuição	OK?
x = p	
x = &p	
p = &x	
&x = p	
*p = x	
x = *p	
*x = &p	

Exercício 5.1

- Leia dois inteiros a e b .
- Exiba os valores de a e b .
- Crie um ponteiro p para a variável com o maior valor.
- Através de p subtraia 50 da variável com o maior valor.
- Exiba os valores de a e b novamente.

Exercício 5.2

- Uma conta é representada por um inteiro que guarda o saldo total:

- `int minha_conta;`

- Uma compra na internet é efetuada por uma chamada `compra(conta, valor)`. A função recebe um ponteiro para uma conta e um valor a ser debitado:

- `void compra (int* conta, int valor) { ... }`

- Um casal tem duas contas e quer usar a conta com maior saldo para fazer uma compra de 500 reais.
- Faça um programa que leia o saldo das duas contas e efetue a compra corretamente. Ao final, o programa deve exibir os saldos das duas contas.

Exercício 5.3

- Agora, o casal tem uma lista de compras a efetuar:
 - `int compras[] = { 100, 50, 80, 30, 20 };`
- Faça um programa que leia o saldo das duas contas e efetue as compras corretamente, sempre usando a conta com o maior saldo.
- A cada compra, o programa deve exibir os saldos das duas contas.

Exercício 5.4

- Crie uma função `troca` que receba dois ponteiros para inteiros `p1` e `p2` e troque os conteúdos por eles apontados:

```
int x=10, y=20;  
troca(&x, &y); // definir essa funcao  
printf("%d %d\n", x, y); // 20 10
```