



Programação 2

Aula 4

Thiago Cavalcante – thiago.cavalcante@penedo.ufal.br

13 de novembro de 2019

Universidade Federal de Alagoas – UFAL

Campus Arapiraca

Unidade de Ensino de Penedo

Ponteiros

Tipo **especial** de variável que armazena um **endereço na memória** ao invés de um valor

Toda informação está armazenada na memória
Ponteiros "apontam" para endereços de memória
programador → nome, programa → endereço

Ponteiros podem ser **perigosos**



Podem apontar para um espaço na memória que está sendo usado para um outro propósito

Declaração de ponteiro

```
tipo_do_ponteiro *nome_do_ponteiro;
```

Não confundir com o operador de multiplicação!

Inicialização e atribuição

- `int *p;` (aponta para um **lugar indefinido** ⚠)
- `p = NULL;` (aponta para **lugar nenhum** – diferente)
- `p = &x;` (aponta para variável `x`)
- `printf("%d", *p)` (imprime o valor guardado na variável apontada por `p`)
- `int *p2 = p;` (ponteiro `p2` aponta para o mesmo lugar que `p` ⚠)

`&`: operador de endereçamento

Operadores

*****: Usado na **declaração** de um ponteiro e na **obtenção do conteúdo** para onde ele aponta
(dois usos distintos)

&: Usado para **obter o endereço** de uma variável

Aritmética com ponteiros

- Adição e subtração: incrementam ou decrementam o tamanho **em bytes** do tipo do ponteiro ⚠
- Outras operações: não são realizadas no ponteiro, mas no seu conteúdo (***p**)

Também é possível comparar ponteiros com operadores relacionais: ==, !=, >, <, >=, <=

Ponteiros genéricos

- `void *nome_do_ponteiro;`
- Aponta para qualquer tipo de dado ⚠
- Acesso precisa ser feito com um *typecast*: `*(int*)p`
- Incremento é feito de byte em byte

Ponteiros e arrays

O nome de um array é apenas um **ponteiro para o primeiro elemento** do array

```
ponteiro = nome_array  
ponteiro = &nome_array[0]
```

```
ponteiro[i] = 2;  
*(ponteiro + i) = 2;
```

Array de ponteiros

```
tipo_de_dado *nome_do_array[tamanho];
```

```
nome_do_array[indice] = &variavel
```

Ponteiro para ponteiro

```
tipo_de_dado **nome_do_ponteiro;
```