# Laboratório de Algoritmos e Técnicas de Programação II

## Aula 02 - Ponteiro: conceitos básicos e formas de manipulação.

Álvaro Magri Nogueira da Cruz

# Agenda

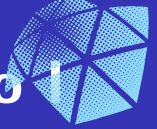

1. Introdução
2. Declarações e inicialização de variáveis-ponteiro
3. Operadores de ponteiros
4. Exemplos com ponteiros
5. Exercícios
6. Referências

# Introdução I

## Definição

- Os ponteiros estão entre as capacidades mais difíceis de se dominar na linguagem em C;
- Eles permitem que os programas **simulem uma chamada por referência**;
- **Criem** e **manipulem** estruturas **dinâmicas** de dados:
  - Estruturas de dados que podem **crescer e encolher** no tempo de execução;
  - E.g., listas interligadas, filas, pilhas e árvores.

# Declarações e inicialização de variáveis-ponteiro I



## Observação

- Os ponteiros são variáveis cujos valores são **endereços de memória**;
- Normalmente, uma **variável** claramente contém um **valor específico**;
- Um ponteiro, por outro lado, contém um endereço de uma variável que contém um valor específico;
- De certa forma, um nome de variável referencia um valor **diretamente**, enquanto um **ponteiro referencia um valor indiretamente**;
- A referência de um valor por meio de um ponteiro é chamada de **indireção**.

Declaração

```
int *countPtr, count;//*countPtr é um ponteiro para um inteiro
```

# Declarações e inicialização de variáveis-ponteiro II

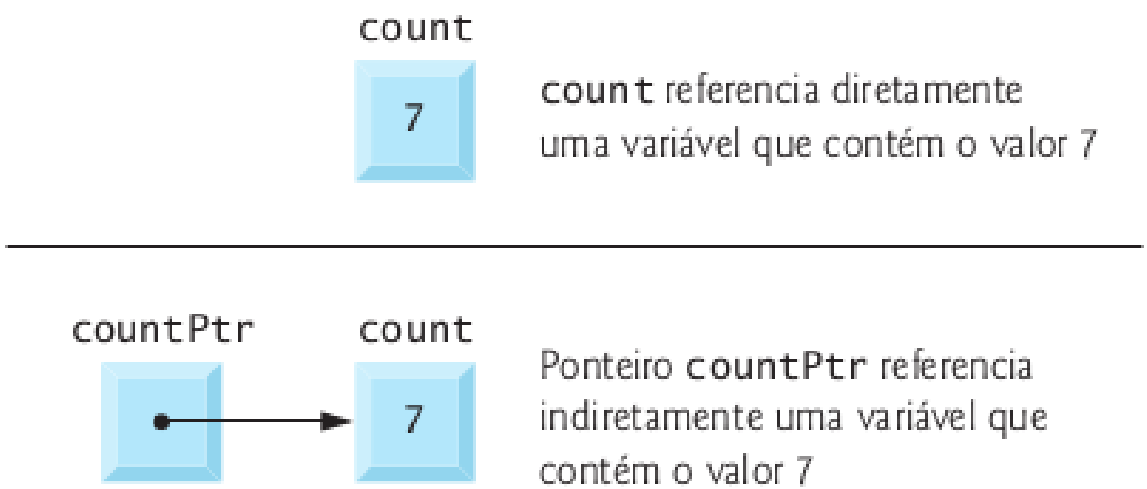



**Figura 1:** Referências direta e indireta de uma variável.

# Declarações e inicialização de variáveis-ponteiro !!!

**Atenção!!!**

- Ponteiros e entidades baseadas em ponteiros, por exemplo, *arrays* e *strings*, quando usados de modo indevido, intencional ou acidentalmente, podem provocar erros e brechas na segurança.

# Operadores de ponteiros I

## Operador & (endereço)

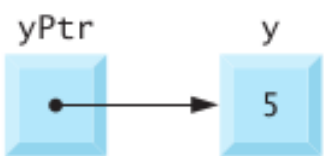
- O & , ou operador de endereço, é um operador unário que retorna o endereço de seu operando;
- Por exemplo, considerando as definições:

  **int** y = 5;

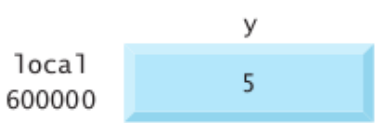
  **int** *yPtr;

  yPtr = &y;

- Atribui o **endereço** da variável *y* à variável de ponteiro *yPtr*.







# Operadores de ponteiros II

## Operador * (indireção)

- O operador unário * , normalmente chamado operador de **indireção** ou de **desreferenciação**, retorna o **valor** do objeto apontado por seu operando (ou seja, um ponteiro);
- Por exemplo:

  printf( "%d", *yPtr);
- Imprime o valor da variável *y* , a saber, 5. Esse uso de * é chamado desreferenciação de um ponteiro.

# Operadores de ponteiros III

## Exemplo de uso dos dois operadores (& e *)

```c
#include <stdio.h>

int main(){
    int valor;
    int *valorPonteiro;
    valor = 25;
    valorPonteiro = &valor;
    printf("O ENDERECO de 'valor' eh %p\n", &valor);
    printf("O VALOR de valorPonteiro eh %p\n", valorPonteiro
    );
    printf("\nO VALOR de 'valor' eh %d\n", valor);
    printf("O VALOR de *valorPonteiro (indirecao) eh %d\n",
    *valorPonteiro);
    printf("\n\nMostrando que * e & sao complementos um do
    outro\n&*valorPonteiro = %p\n*&valorPonteiro = %p\n", &*
    valorPonteiro, *&valorPonteiro);
    return 0;
}
```

# Operadores de ponteiros IV

Codes : bash — Konsole

Arquivo Editar Exibir Favoritos Configurações Ajuda

```
bruna@bruna:~/Documentos/Documentos/UNESP/Lab. Prog. II/Aulas/Aula 02/Codes
$ ./ponteiro
O ENDERECO de 'valor' eh 0x7ffc6e39ef4c
O VALOR de valorPonteiro eh 0x7ffc6e39ef4c

O VALOR de 'valor' eh 25
O VALOR de *valorPonteiro (indirecao) eh 25


Mostrando que * e & são complementos um do outro
&*valorPonteiro = 0x7ffc6e39ef4c
*&valorPonteiro = 0x7ffc6e39ef4c
bruna@bruna:~/Documentos/Documentos/UNESP/Lab. Prog. II/Aulas/Aula 02/Codes
$
```

# Exemplos com ponteiros I

## Soma de dois números por meio de ponteiros

```c
#include <stdio.h>

int main(){
    float a, b, c;
    float *aPonteiro, *bPonteiro, *cPonteiro;
    printf("Digite o primeiro numero da soma: ");
    scanf("%f", &a);
    printf("\nDigite o segundo numero da soma: ");
    scanf("%f", &b);
    aPonteiro = &a;
    bPonteiro = &b;
    cPonteiro = &c;
    *cPonteiro = *aPonteiro + *bPonteiro;
    printf("\nO resultado de %f + %f = %f\n", *aPonteiro, *
    bPonteiro, c);
    return 0;
}
```

# Exercícios

1. Crie um vetor de tamanho N (o usuário entrará com o tamanho), preencha este vetor com número aleatórios. Crie duas variáveis, uma para calcular a média e a outra para a soma de todos os elementos do vetor. Por fim, crie um ponteiro para cada uma das variáveis citadas, e imprima o endereço que estão localizadas e o conteúdo de cada uma;

# Referências I


1. **Deitel H. M., e Deitel P.J; "C: Como programar". 6.ed. Pearson Prentice Hall, 2011. 818p.**
2. Jean Paul Tremblay & Richard P. Bunt. "Ciência dos Computadores - Uma abordagem algorítmica". McGraw-Hill.
3. Jaime Evaristo. "Aprendendo a Programar / Programando em Turbo Pascal". Edufal - Editora da Univ. Federal de Alagoas. Maceió, 1996.
4. Harry Farrer et al. "Pascal Estruturado (da série "Programação Estruturada de Computadores")". Editora Guanabara Dois. Rio de Janeiro, 1985.
5. Stephen O'Brien. "Turbo Pascal 6 Completo e Total". Makron Books.

# Referências II


6. Celes, W., Cerqueira, R., Rangel, J.L. "Introdução a Estrutura de Dados". Elsevier, 2004.
7. Feofiloff, P. "Algoritmos em Linguagem C". Elsevier, 2009. 208p.
8. Schildt, H. "C Completo e Total". 3ª ed. Pearson. 1996. 852p.