

Relação entre Arrays e Ponteiros

Andressa Andrade, Renata Antunes e Rodrigo Fernandes

Conceitos

❖ Ponteiros

- Um ponteiro é uma variável que contém o endereço de um dado
- Declaração: `tipo_dado *nome_ponteiro`
- O “*” indica que a variável é um ponteiro

❖ Arrays

- Arrays são agrupamentos de dados adjacentes na memória
- Declaração: `tipo_dado nome_array[<tamanho>];`
- Ele define um arranjo de <tamanho> elementos adjacentes na memória do tipo `tipo_dado`

Criando um ponteiro

- ❖ Para criarmos um array:

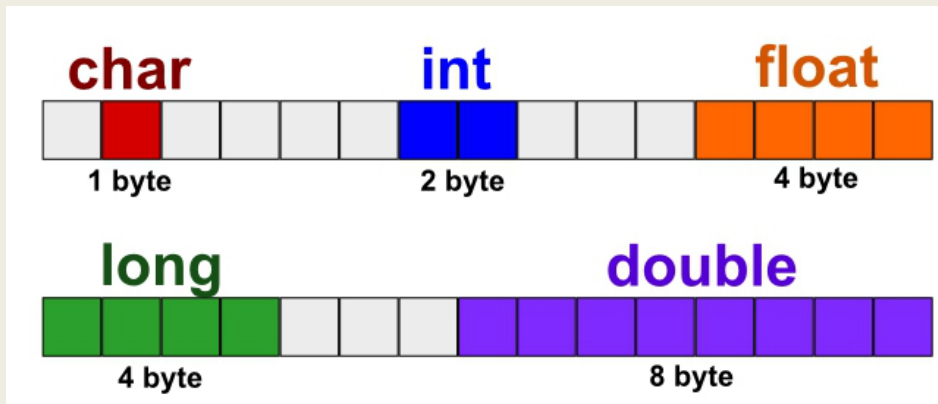
```
#include <stdio.h>
int main(void)
{
    int array[n] = {1,2,...,n};
```

- ❖ Para criarmos um ponteiro relacionado ao array, podemos usar duas formas:

- `int *ponteiro = &array[0]`
- `int *ponteiro = array`

Os tipos de variáveis e suas alocações

- ❖ A alocação dos tipos de variáveis na memória será:



- ❖ No exemplo do int: uma variável tipo int ocupará 2 bytes no computador de 16 bits. Já em um computador 32 bits, ela ocupará 4 bytes.

Endereço ou Conteúdo

- ❖ Se quisermos adicionar 1 em algum elemento do vetor, usaremos:
 - `*(ponteiro + i) += 1`
- ❖ Já, para adicionar 1 ao endereço de algum vetor, usaremos:
 - `ponteiro = &vetor[i] + 1;`
- ❖ Exemplo:
 - Adicionar 1 ao endereço do 3º elemento => `ponteiro = &vetor[2] + 1;`
 - Adicionar 1 ao 2º elemento => `*(ponteiro + 1) += 1;`

Ponteiro e Array

- ❖ Um ponteiro é uma variável, mas o nome de um array não é uma variável, ele é uma constante.

- `array = array + 2;` // ERRADO: array não é variável

- `array++;` // ERRADO: array não é variável

- `ponteiro = array;` // CERTO: ponteiro é variável

- `ponteiro = array+2;` // CERTO: ponteiro é variável

- `ponteiro++;` // CERTO: ponteiro é variável

Exemplo 1

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int array[10] = {1,2,3,4,5,6,7,8,9,0};
```

```
    int *ponteiro = array;
```

```
    int i;
```

```
    for(i = 0; i < 10; i++)
```

```
    {
```

```
        ponteiro = &array[i];
```

```
        printf("%p ", ponteiro);
```

```
    }
```

```
    return 0;
```

```
}
```

Exemplo 2

```
#include <stdio.h>

int main(void)
{
    int vetor[5];
    int i;
    int *ponteiro = vetor;
    printf("Digite 5 valores inteiros, separados por espaco:\n");
    scanf("%i %i %i %i %i",&vetor[0],&vetor[1],&vetor[2],&vetor[3],&vetor[4]);

    for(i = 0; i < 5; i++)
    {
        *(ponteiro + i) += 5;
        printf("%i ", vetor[i]);
    }

    return 0;
}
```