INTRODUÇÃO À PROGRAMAÇÃO II

VARIÁVEIS COMPOSTAS HOMOGÊNEAS UNIDIMENSIONAIS

PARTE 2

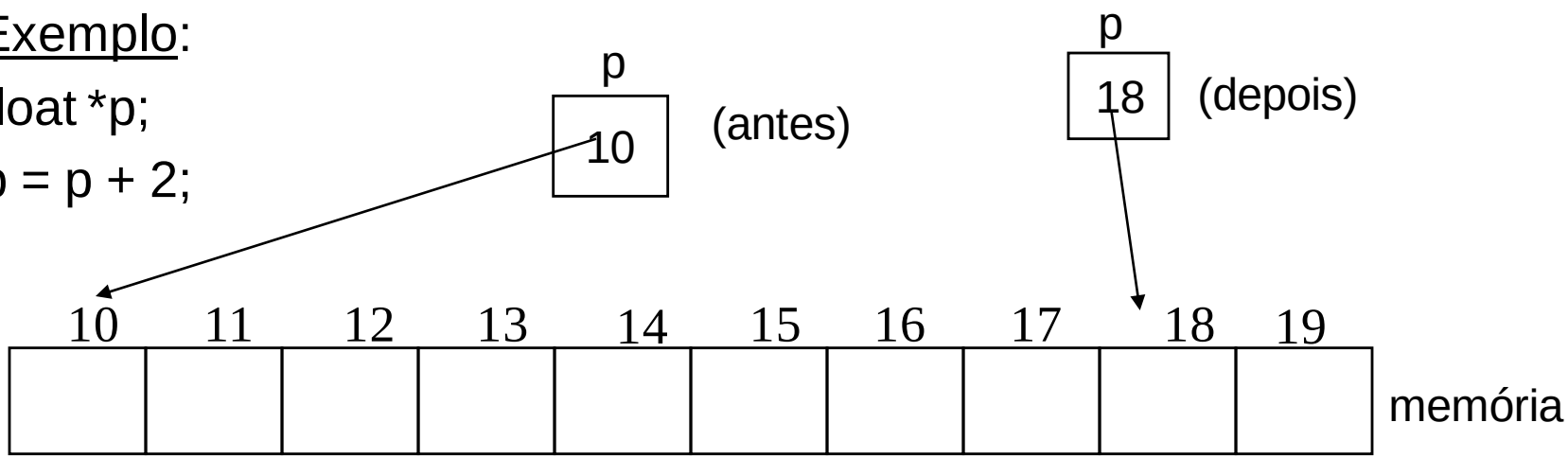
Material da Prof. Ana Eliza

Aritmética de Ponteiros

- **Adição**: somar um inteiro a um ponteiro

Exemplo:

```
float *p;
p = p + 2;
```



Fator de escala: é o tamanho (número de bytes) do objeto apontado.

Exemplo: fator de escala de uma variável do tipo float: 4

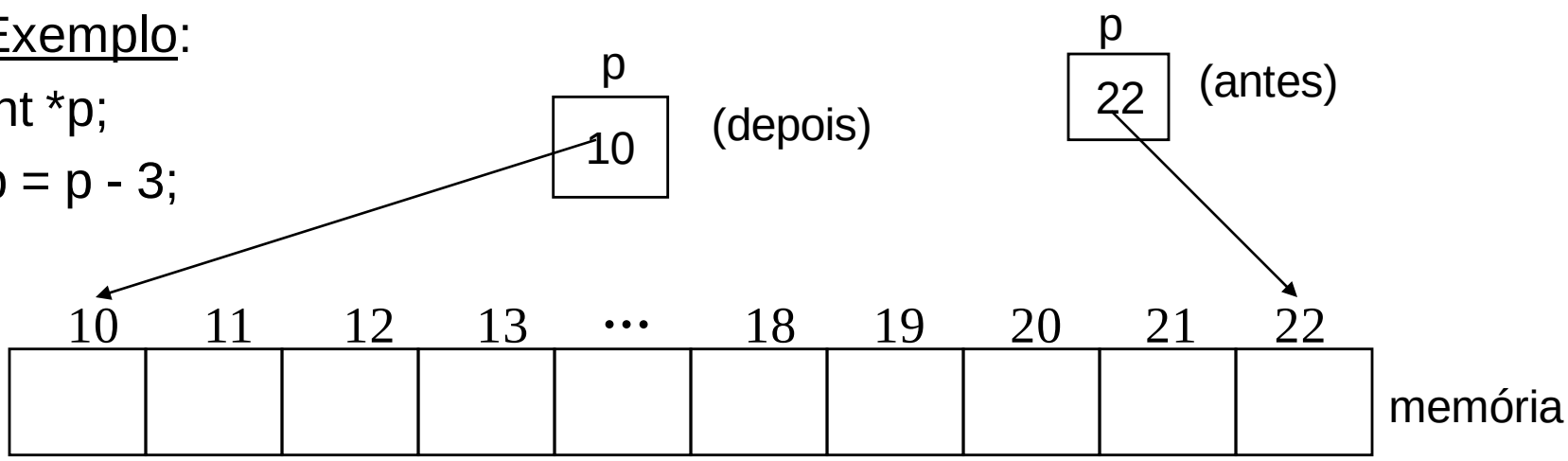
p = p + 2	significa	**p = p + 2 * fator de escala**

Aritmética de Ponteiros

- **Subtração**: subtrair um inteiro de um ponteiro

Exemplo:

```
int *p;
p = p - 3;
```



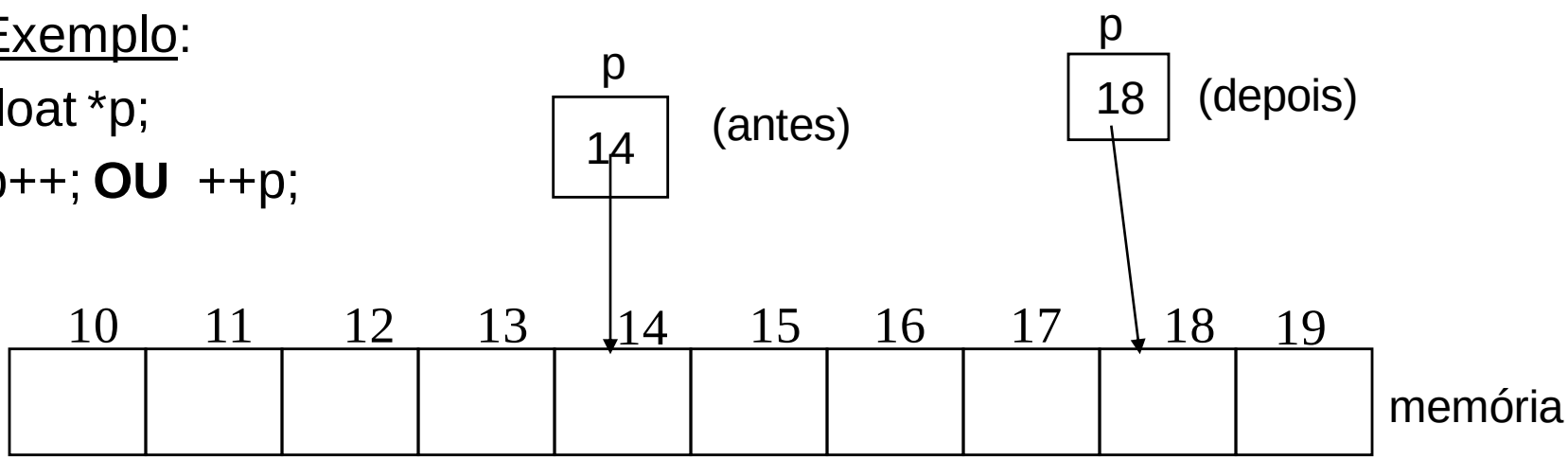
p = p - 3 significa **p = p - 3 * fator de escala**

Aritmética de Ponteiros

- **Incremento**: somar um a um ponteiro

Exemplo:

```
float *p;
p++; OU  ++p;
```



p++ significa **p = p + fator de escala**

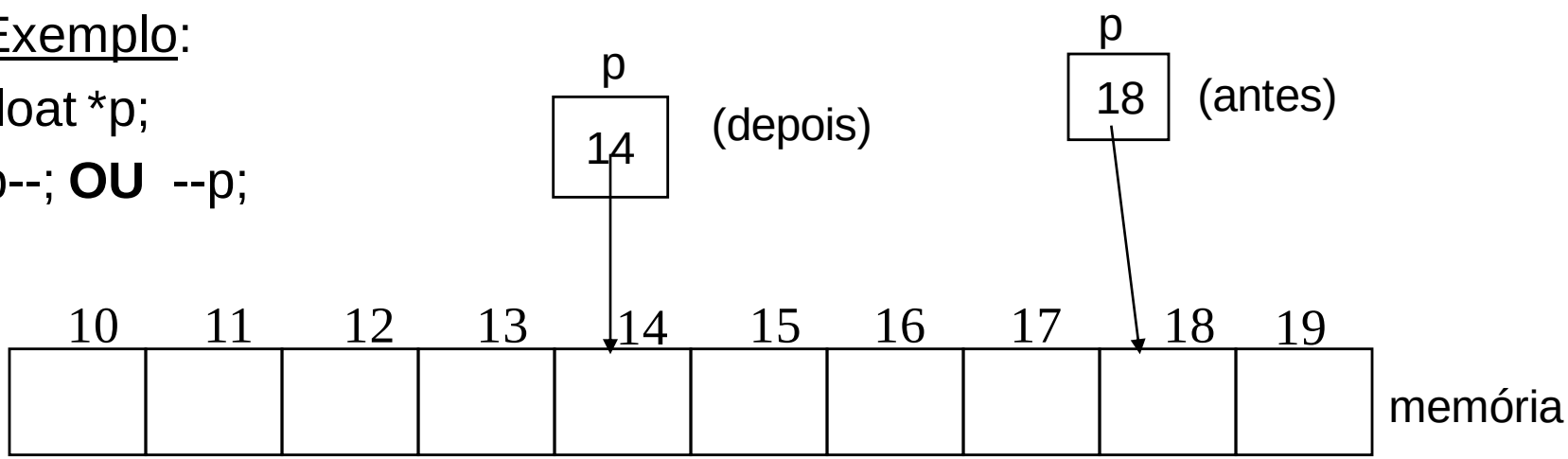
Aritmética de Ponteiros

- **Decremento**: subtrair um de um ponteiro

Exemplo:

float *p;

p--; **OU** --p;



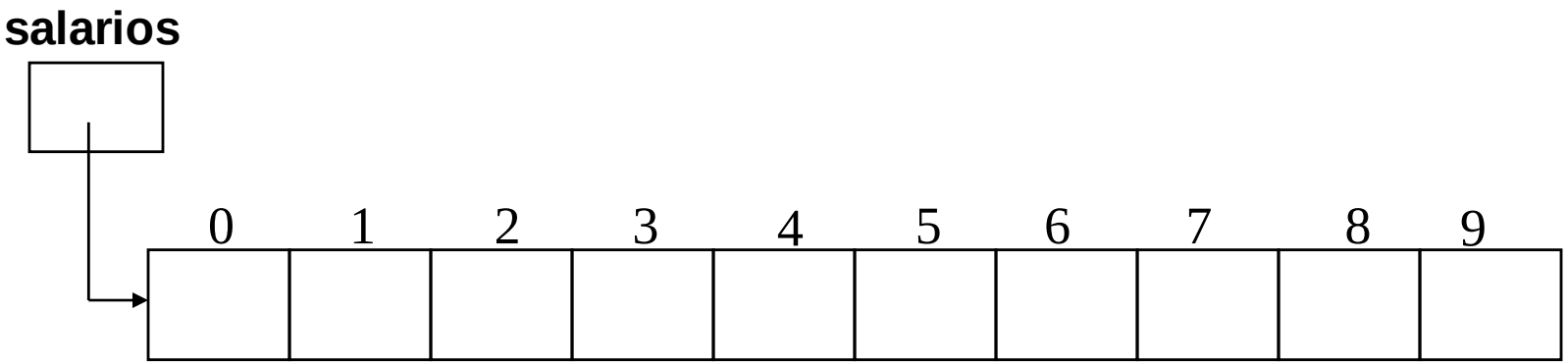
p--	significa	**p = p - fator de escala**

Arrays e Ponteiros

- O nome de um array, considerado isoladamente, é interpretado como um ponteiro para o início do array.

  Exemplo: float salarios [10];

  "A variável **salarios** é um ponteiro para o início do array de floats."

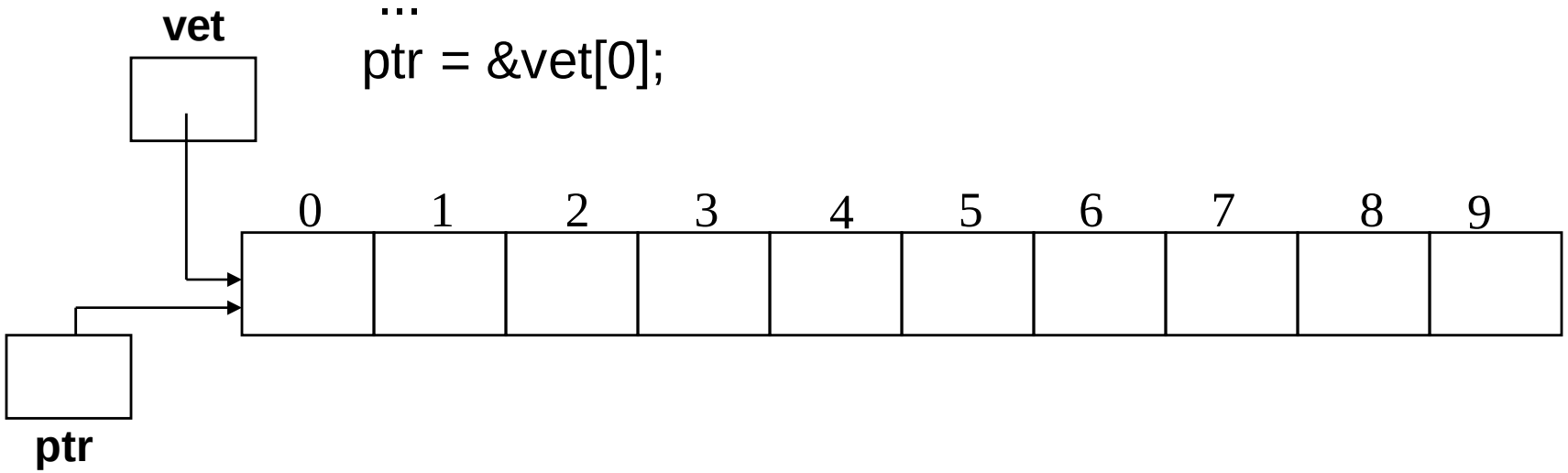


salario é o mesmo que **&salario [0]**

Arrays e Ponteiros

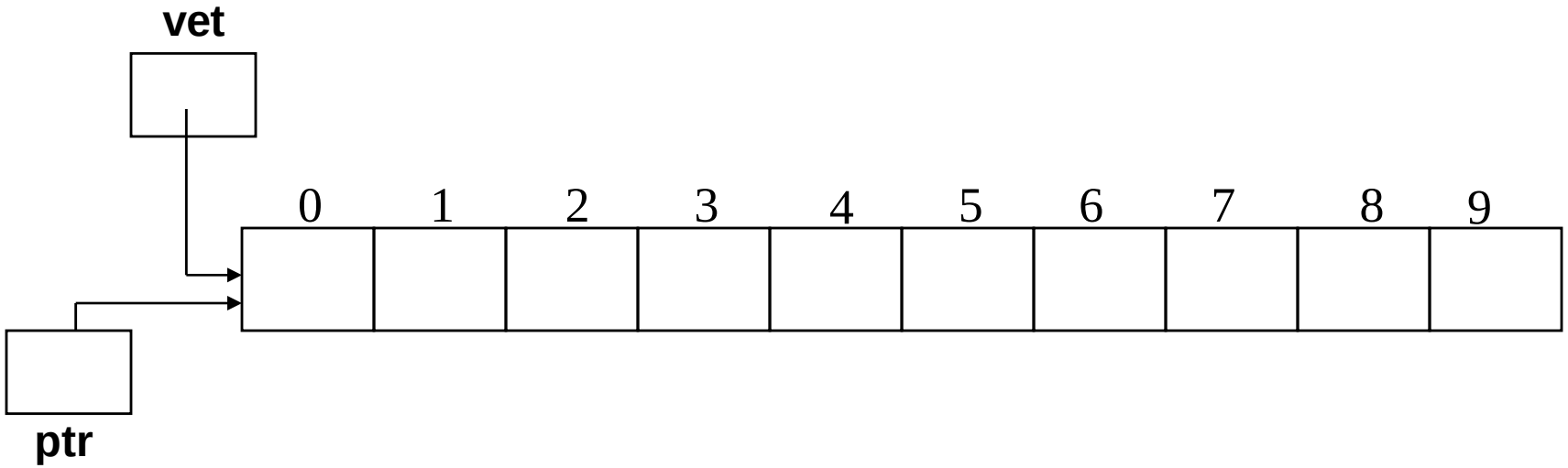
- Acesso aos elementos de um array utilizando ponteiros.

Exemplo:
```
int vet [10];
int * ptr;
 ...
ptr = &vet[0];
```



Arrays e Ponteiros

- Acesso aos elementos de um array utilizando ponteiros (cont).

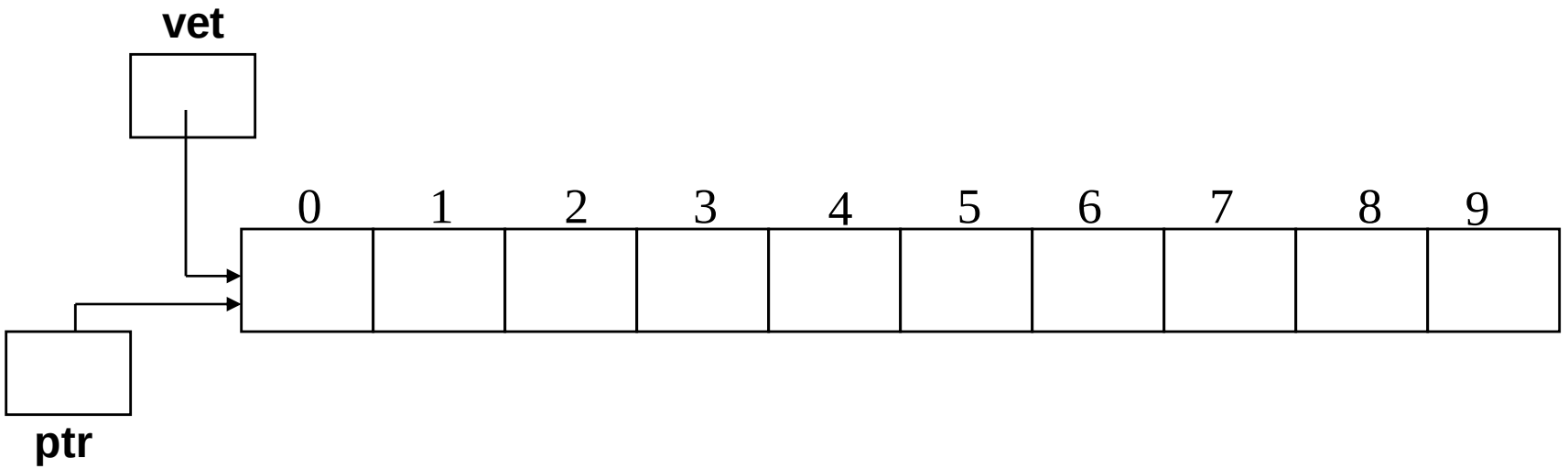


*(ptr + i)	é o mesmo que	vet[i]	*(ptr + 2) = 10; **OU** vet[2] = 10;
*(vet + i)	é o mesmo que	vet[i]	*(vet + 3) = 25; **OU** vet[3] = 25;

Arrays e Ponteiros

- Acesso aos elementos de um array utilizando ponteiros (cont).

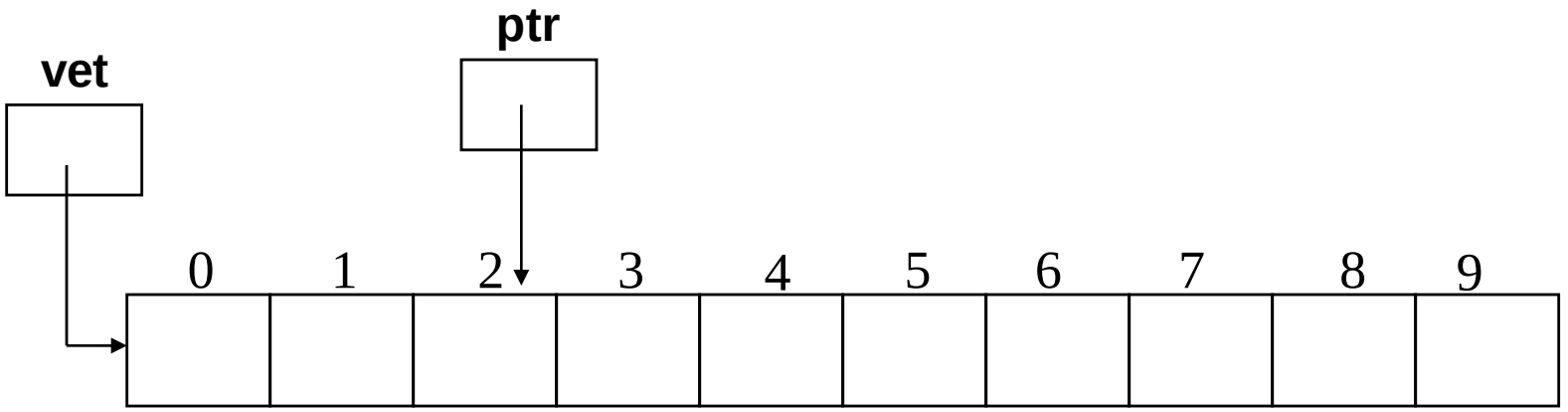
```
for (i = 0; i <= 9; i++)
    *(ptr + i) = i;  OU  vet[i] = i;
```



Arrays e Ponteiros

- Acesso aos elementos de um array utilizando ponteiros (cont).

```
ptr = ptr + 2;
```



OBS: **vet** é um ponteiro constante ("*fixo*") e, portanto, não pode sofrer atribuições.

Arrays e Ponteiros

- Arrays como parâmetro de sub-programas

  O **parâmetro formal** deve ser um ponteiro para o elemento inicial do array.
  Formas de declaração de parâmetros do tipo array:

  (1) tipo_do_elemento * nome_parâmetro
  Exemplo: void MinhaFuncao (float * vet)

  (2) tipo_do_elemento nome_parâmetro [ ]
  Exemplo: void MinhaFuncao (float vet [ ])

  (3) tipo_do_elemento nome_parâmetro [qtd_elementos]
  Exemplo: void MinhaFuncao (float vet [25])

Arrays e Ponteiros

- Arrays como parâmetro de sub-programas

O **parâmetro real** deve ser o nome do array, que será interpretado como o endereço do primeiro elemento do array.

Exemplo:

```
float numeros [25];
 ...
MinhaFuncao (numeros);
```