UNIVERSIDADE FEDERAL DO CEARÁ

CK0109 - 2019.2 - T02

ESTRUTURAS DE DADOS

AULA 03 - 2019-08-09

PONTEIROS

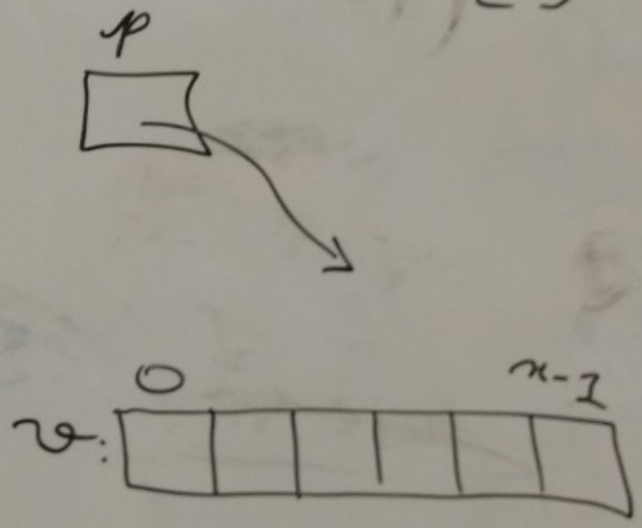
1. COMPLEMENTO SOBRE A FORMA DA DECLARAÇÃO DE PONTEIROS: A INTUIÇÃO NA DECLARAÇÃO DE UM PONTEIRO p P/ TIPO T É QUE, DEPOIS DA DECLARAÇÃO, A EXPRESSÃO $*p$ TERÁ TIPO T.

(RASCUNHOS)

t DECLARA v DE TIPO T

v → (*p) t' DECLARA PONT. p P/ T

INT (*p)[3]



++i
i++

(*p)[7]

v[3]

(/) +

ALÉM DISSO, SERÁ POSSÍVEL OMITIR OS PARÊNTESES DA DECLARAÇÃO DE p QUANDO NÃO HOUVER OUTROS "OPERADORES" DE MAIOR PRECEDÊNCIA PRESENTES E QUE ALTERARIAM O SIGNIFICADO DA DECLARAÇÃO SEM OS PARÊNTESES (COMO "[]").

EXEMPLOS:

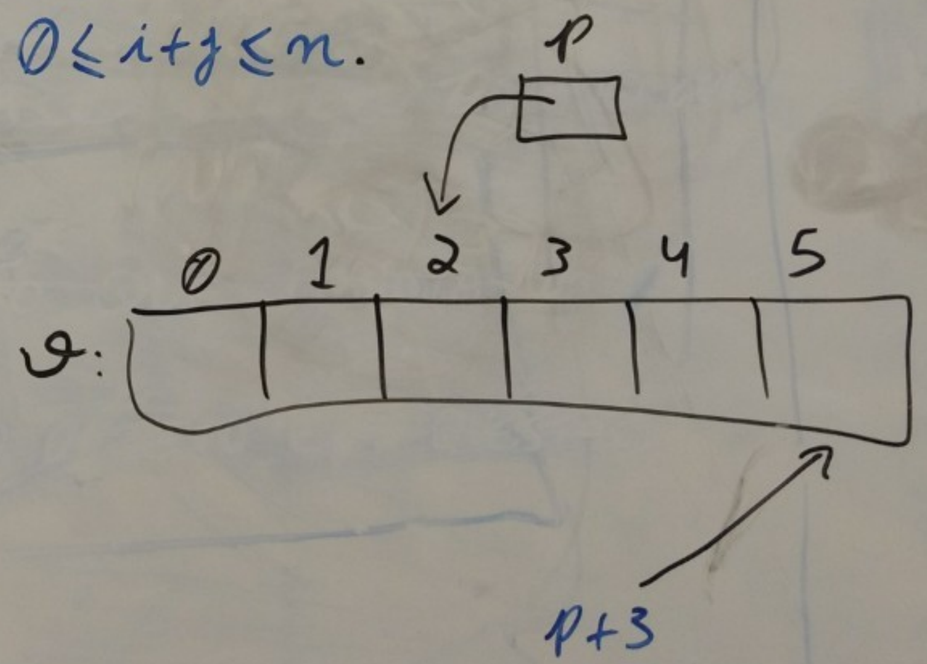
```
INT *p;  //PONTEIRO PARA INT.

INT *v[3]; /* VETOR DE 3
                PONTEIROS P/ INT.*/

INT (*pv)[3]; /* PONTEIRO PARA
                 VETOR DE 3 INT'S */
```

ARITMÉTICA DE PONTEIROS

2. PONTEIRO + INTEIRO: SE p APONTA PARA $v[i]$, SENDO v UM VETOR DE n ELEMENTOS, E SE j É UM INTEIRO, ENTÃO

$$p + j$$

É UMA EXPRESSÃO QUE DENOTA UM PONTEIRO PARA $v[i+j]$, DESDE QUE $0 \leq i+j \leq n$.



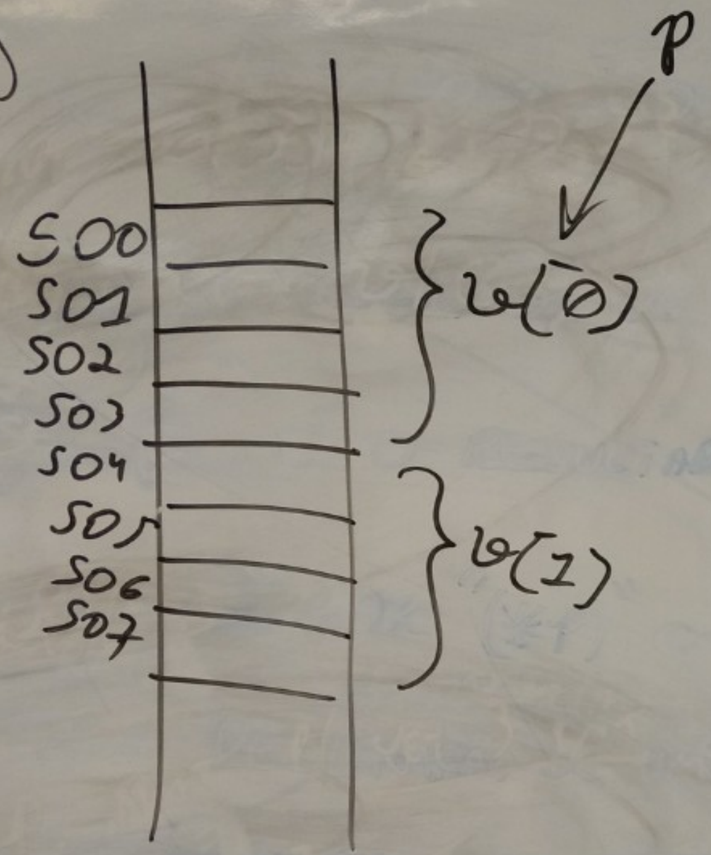
(PARA DEPOIS: CASOS DE BORDA.)

(RASCUNHO)

p - i

p + (-i)

i - p




```
int i = 3;
...i+7...
i = i+7;
```

```
int *p;
...p+7...
p = p+7;
```

3. Comparação de Ponteiros: se p aponta para v[i], sendo v um vetor de n elementos, e se q aponta para v[j], então

p < q se e somente se i < j,

e o mesmo vale para <=, >, >=, == e !=.

Além disso, dois ponteiros p e q podem ser comparados via == e != mesmo que não apontem para elementos de um mesmo vetor.

4. Subtração entre Ponteiros: se p aponta para v[i], sendo v vetor de n elementos, e q aponta para v[j], então o valor de

$$p - q$$

é igual ao de $i - j$ (embora o tipo de p-q seja ptrdiff_t, que é um inteiro com sinal grande o suficiente).

5. Exemplo de Percurso de Vetor via Ponteiros:

```
int *prim;
prim = ... ;
int* p;
```

```
#include <iostream>
using std::cout;
int main()
  {
  int v[3] = {1,2,3};
  int *prim = &v[0], *ult = &v[2], *p;
      // ou ... = v+0, ...    v+2
for(p=prim; p<=ult; ++p)
  {
  cout << "v[" << p-prim << "]: "
       << *p << '\n';
  }
}
```

Re[...]