

Arquitectura de Computadores I

Pedro Miguel Cabral

Aula 05

Índices vs Ponteiros

1

2

• $f = a[i];$ // Com $i \geq 0$

Para aceder ao elemento “ i ” do array “ a ”, o programa começa por calcular o respectivo endereço, a partir do endereço inicial do array:

**endereço do elemento a aceder = endereço inicial do array +
(índice * dimensão em bytes de cada posição do array)**

```
//Exemplo1
int i, array[size];
for (i = 0; i < size; i++)
{
    array[i] = 0;
}

$.DATA
array: $.SPACE size * 4
$.TEXT
(...)
    la    $t2, array      # $t2 = &(array[0]);
    li    $t0, 0          # i = 0;
loop:   bge $t0, $a0, endf # while (i < size) {
        sll  $t1, $t0, 2    # temp = i * 4;
        addu $t1, $t2, $t1  # temp = &(array[i])
        sw   $0, 0($t1)     # array[i] = 0;
        addi $t0, $t0, 1    # i = i + 1;
        j    loop          # }
endf:   ...
```

\$t0 ← i
\$t1 ← temp
\$t2 ← &(array[0])
\$a0 ← size

3

• $f = *pt;$

O endereço do elemento a aceder está armazenado num registo.

**endereço do elemento seguinte = endereço actual +
dimensão em bytes de cada posição do array**

```
//Exemplo2
int *p, array[size];
for (p=&array[0]; p < &array[size]; p++)
{
    *p = 0;
}

$.DATA
array: $.SPACE size * 4
$.TEXT
(...)
    la    $t0, array      # $t0 = &(array[0]);
    sll  $t1, $a0, 2      # $t1 = size * 4;
    addu $t1, $t1, $t0    # $t1 = &(array[size]);
loop:   bgeu $t0, $t1, endf # while (p < &array[size]) {
        sw   $0, 0($t0)    # *p = 0;
        addiu $t0, $t0, 4  # p = p + 1;
        j    loop          # }
endf:   ...
```

\$t0 ← p
\$t1 ← &(array[size])
\$a0 ← size

4