

# Arquitectura de Computadores I

Pedro Miguel Cabral

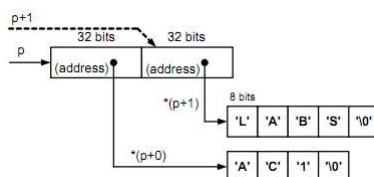
## Aula 6

Utilização de ponteiros em linguagem C – array de ponteiros para caracter.

1

### Array de ponteiros para caracter

Para além do acesso simples a arrays visto na aula anterior, em linguagem C os ponteiros podem ser usados em estruturas de dados mais elaboradas.



Array de ponteiros (dois, no exemplo), cada um deles apontando para um caracter (no exemplo, esse caracter é o primeiro de um array de caracteres).

Se  $p$  for o ponteiro para o início do array de ponteiros então  $p+1$  é um ponteiro para o segundo elemento desse array.

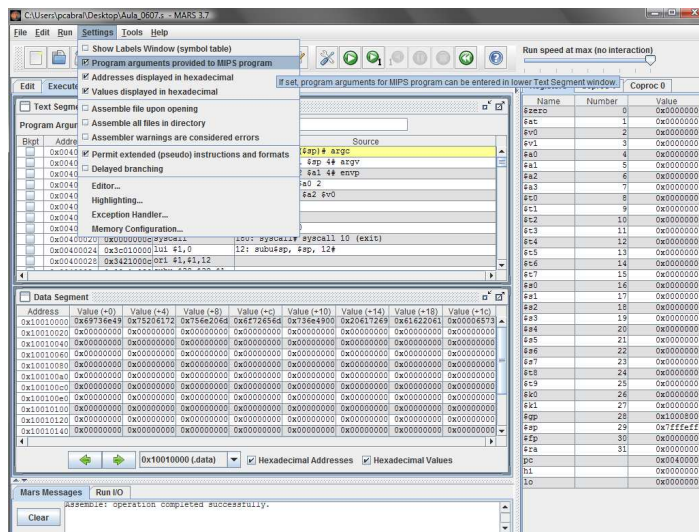
O endereço do caracter apontado pela primeira posição do array  $p$  será então  $*p$  e, de modo idêntico, o endereço do caracter apontado pela segunda posição do array  $p$  será  $*(p+1)$ .

Em linguagem C esta estrutura de dados é definida do seguinte modo:

```
char *p[]={ "AC1", "LABS" };    // Array de ponteiros para caracter
ou,
char *p[2]={ "AC1", "LABS" };   // Array de ponteiros para caracter
```

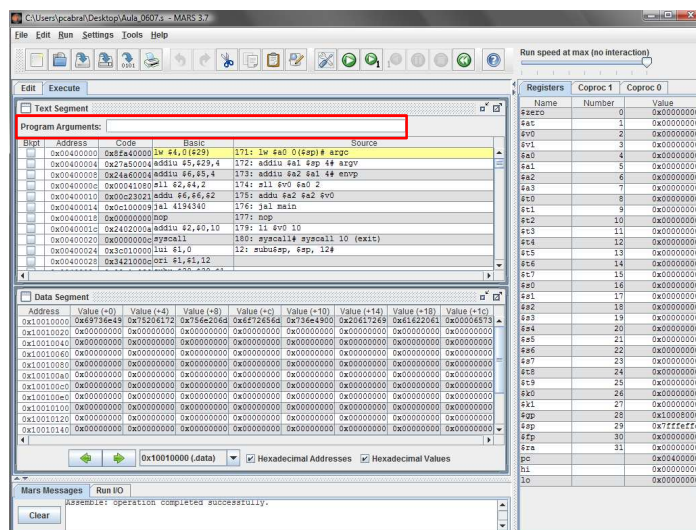
## Passagem de Parâmetros pela Linha de Comandos

AC1  
2013-2014



## Passagem de Parâmetros pela Linha de Comandos

AC1  
2013-2014

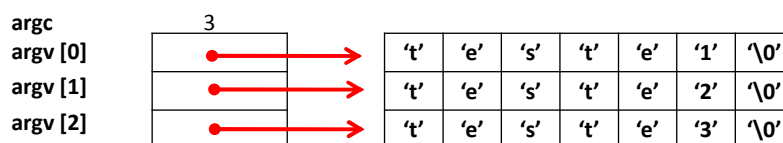


## Passagem de Parâmetros pela Linha de Comandos

AC1  
2013-2014

**argc** nº de argumentos passados  
**argv** array de apontadores para string, apontando para os parâmetros passados ao programa.

Program Arguments: teste1 teste2 teste3



## Passagem de Parâmetros pela Linha de Comandos

AC1  
2013-2014

**argc** nº de argumentos passados  
**argv** array de apontadores para string, apontando para os parâmetros passados ao programa.

Program Arguments: teste1 teste2 teste3

```
.data
label: .asciiz "\n Argumento:\n"

.text
.globl main
main:
    move    $t0, $a0
    move    $t1, $a1

    li      $t2, 0
for:
    bge     $t2, $t0, end_for
    la      $a0, label
    li      $v0, 4
    syscall

    lw      $a0, 0($t1)
    li      $v0, 4
    syscall

    addu    $t1, $t1, 4
    addi    $t2, $t2, 1
    j       for
end_for:
    jr      $ra
```

```
Argumento:
teste1
Argumento:
teste2
Argumento:
teste3
-- program is finished running --
```