

Macros e diretivas de tamanho

Macro

- Prover um modo de um bloco de texto ser representado por um nome.
- Quando o assembler encontra o nome este é substituído pelo bloco de texto associado.
- Este processo é conhecido como expansão da macro.
- Macros podem ser definidas como:

```
%macro      macro_name      para_count
              <macro body>
%endmacro
```

Macro

- O texto macro começa com **%macro** e termina com **%endmacro**.
- O **para_count** especifica a quantidade de parâmetros da macro.
- **macro_name** é o nome da macro.

Macro sem parâmetro

- Exemplo:

```
%macro    multEAX_by_16
           sal      EAX, 4
%endmacro
```

```
    . . .
mov    EAX, 27
multEAX_by_16
    . . .
```

```
    . . .
mov    EAX, 27
sal    EAX, 4
    . . .
```

Macro com parâmetro

- Podemos criar macros mais flexíveis passando parâmetros:

```
%macro    mult_by_16    1
          sal    %1, 4
%endmacro
```

- Exemplo:

```
mult_by_16    DL
```

- O comando acima causa a macro expansão:

```
sal    DL, 4
```

Macro com parâmetro

- Exemplo com variáveis de memória:

```
mult_by_16    count
```

- A expansão causa:

```
sal    count, 4
```

Macro com parâmetro

- Exemplo de troca entre variáveis de memória:

```
%macro    mxchg    2
           xchg     EAX, %1
           xchg     EAX, %2
           xchg     EAX, %1
%endmacro
```

- A macro é invocada:

```
mxchg     value1, value2
```

Macro com parâmetro

- Exemplo:

```
%macro PutInt 1
    push    AX
    mov     AX, %1
    call    proc_PutInt
    pop     AX
%endmacro
```

- Mostra um inteiro no monitor

Directives BYTE PTR, WORD PTR, DWORD PTR

- Existem momentos em que precisamos traduzir referências para dados na memória.
- Exemplo:

`mov [ESI], al ; armazena um valor de 1 byte no local de memória apontado por ESI.`

Directives BYTE PTR, WORD PTR, DWORD PTR

- Quando a instrução não conhece o tamanho do operando:

`mov [ESI], 5` ; Erro: operando deve ter um tamanho especificado.

- Para resolver o erro acima utilizamos as diretivas de ponteiro:

`mov BYTE PTR [ESI], 5` ; armazena valor de 8-bit

`mov WORD PTR [ESI], 5` ; armazena valor de 16-bit

`mov DWORD PTR [ESI], 5` ; armazena valor de 32-bit

Directives BYTE PTR, WORD PTR, DWORD PTR

- Ptr força a expressão para ser tratada como um ponteiro do tipo especificado:

```
.DATA
```

```
    num DWORD 0
```

```
.CODE
```

```
    mov     ax, WORD PTR [num] ; carrega uma word de uma  
DWORD
```

Exercício

- Escreva uma macro que realize a soma de três números e imprima o resultado no monitor
- Escreva uma macro que calcule a média de três números e imprima o resultado no monitor
- Escreva uma macro que chame uma procedure para realizar a subtração entre três números e imprima o resultado no monitor caso o resultado seja maior que zero.