

# **Linguagens de Montagem**

DEMAC – Departamento de Estatística

Matemática Aplicada e Computação

UNESP – Rio Claro

**Prof. Daniel Carlos Guimarães Pedronette**

# Aula 7.

# Macros

# Macros

- Relembrando ...
  - *Statements* em Assembly:
    - Instruções
      - OpCode, geram código de máquina
    - Diretivas
      - Pseudo-Op, não geram código de máquina
    - **Macros**

# Macros

- Definição Geral:
  - Macros permitem ao programador Assembly nomear um conjunto de instruções e referir-se a esse conjunto pelo nome da macro.
  - Similar ao conceito de Função/Procedimento, mas há diferenças!

# Macros

- *Macro Expansion*:
  - Durante o processo de montagem, cada macro é **substituída** pelo grupo de *statements* que representa.
    - Mecanismo sofisticado de substituição de texto
  - Esse processo é chamado de *Macro Expansion* e é executado pelo montador.

# Macros

- Definição de Macros:

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

# Macros

- Exemplo:
  - Macro sem parâmetros: multiplicação por potências de dois pode ser substituída por *shift*.

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

# Macros

- Exemplo de utilização:
  - Ao encontrar o nome da macro, o montador faz a substituição pelas instruções correspondentes

```


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



# Macros

- Exemplo:
  - Macro com parâmetros:

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

mult\_by\_16      DL            sal      DL, 4

# Macros

- Exemplo:
  - Macro para realizar transferência memória-memória:

```
%macro    mxchg    2
           xchg     EAX, %1
           xchg     EAX, %2
           xchg     EAX, %1
           mxchg     value1,value2
%endmacro
```

# Exemplos

Prática:

- Codificar,
- Montar,
- Linkar e
- Testar!

# Exercícios