

O objetivo desta atividade é permitir que o aluno seja capaz de criar um programa em linguagem assembly que utilize saltos condicionais e incondicionais.

### Parte 1 – Rótulos

Os **rótulos** (labels) são pontos no programa definidos pelo programador. O programador pode definir quantos rótulos quiser, onde quiser. A definição de um rótulo é feita colocando-se o nome do rótulo seguido por pelo caractere ":".

Ex:

```
; Define um rótulo chamado INICIO
INICIO:
; Define um rótulo chamado EXIBE
EXIBE:
; Define um rótulo chamado FIM
FIM:
```

### Parte 2 – Instrução de comparação (CMP)

A instrução **CMP** serve para comparar dois valores. O resultado da comparação é armazenado em um registrador especial, chamado registrador de FLAGS.

Ex:

```
; Compara o valor de AX (16 bits) com o valor de BX (16 bits)
CMP AX,BX
; Compara o valor de AL (8 bits) com o valor de BL (8 bits)
CMP AL,BL
; Compara o valor de BL (8 bits) com o caracter "A" (8 bits)
CMP BL,"A"
; Compara o valor de BL (8 bits) com o valor 65 (8 bits)
CMP BL,65
```

### Parte 3 – Instrução de saltos condicionais e incondicionais

A instrução de salto incondicional **JMP** salta para um ponto do programa definido por um rótulo sempre que é executada.

Ex:

```
; Define um rótulo chamado INICIO
INICIO:
; Salta para o ponto do programa onde se encontra o rótulo INICIO
JMP INICIO
```

As instruções de saltos condicionais utilizam os valores armazenados no registrador de FLAGS para decidir se devem ou não saltar para um determinado ponto do programa definido por um rótulo:

- Caso a condição testada pela instrução de salto seja verdadeira, o programa salta para o rótulo especificado;
- Caso a condição testada seja falsa, o programa não salta para o rótulo especificado, ou seja, continua a execução do programa sequencialmente, executando a instrução seguinte à instrução de salto.

Ex:

```
; Define um rótulo chamado INICIO
INICIO:
; Compara o valor de BL (8 bits) com o valor 65 (8 bits)
CMP BL,65

; Se BL for menor que 65, salta para INICIO
JB INICIO
; Se BL for maior ou igual a 65, continua executando sequencialmente
```

A tabela a seguir apresenta algumas instruções de saltos condicionais e incondicionais úteis:

| Salto incondicional     |                             |     |                             |
|-------------------------|-----------------------------|-----|-----------------------------|
| JMP                     | Salta incondicionalmente    |     |                             |
|                         |                             |     |                             |
| Saltos condicionais     |                             |     |                             |
| JE                      | Salta se for igual          | JZ  | Salta se for zero           |
| JNE                     | Salta se não for igual      | JNZ | Salta se não for zero       |
|                         |                             |     |                             |
| Números não-sinalizados |                             |     | Números sinalizados         |
| JA                      | Salta se for maior          | JG  | Salta se for maior          |
| JAe                     | Salta se for maior ou igual | JGE | Salta se for maior ou igual |
| JB                      | Salta se for menor          | JL  | Salta se for menor          |
| JBE                     | Salta se for menor ou igual | JLE | Salta se for menor ou igual |

#### Parte 4 – ATIVIDADE 3.1 -- Programa: NUMERO.ASM

1) **Digite** o programa a seguir:

```
TITLE Numero
.MODEL SMALL
.STACK 100h
.DATA
    MSG1 DB "Digite um caractere: $"
    SIM DB 10,13,"O caractere digitado e um numero.$"
    NAO DB 10,13,"O caractere digitado nao e um numero.$"
.CODE
; Permite o acesso às variáveis definidas em .DATA
MOV AX,@DATA
MOV DS,AX

; Exibe na tela a string MSG1 ("Digite um caractere: ")
MOV AH,9
MOV DX,OFFSET MSG1
INT 21h

; Lê um caractere do teclado e salva o caractere lido em AL
MOV AH,1
INT 21h

; Copia o caractere lido para BL
MOV BL,AL
```

```
; Compara o caractere em BL com o valor 48 (código ASCII do caractere "0")
CMP BL,48

; Se o caractere em BL for menor que 48 ("0"), salta para o rótulo NAOENUMERO
JB NAOENUMERO

; Compara o caractere em BL com o valor 57 (código ASCII do caractere "9")
CMP BL,57

; Se o caractere em BL for maior que 57 ("9"), salta para o rótulo NAOENUMERO
JA NAOENUMERO

; Se chegou até aqui, exibe na tela dizendo que o caracter é um número
MOV AH,9
MOV DX,OFFSET
INT 21h

; Salta para o rótulo FIM
JMP FIM

; Define o rótulo NAOENUMERO
NAOENUMERO:

; Exibe na tela dizendo que o caractere não é um número
MOV AH,9
MOV DX,OFFSET NAO
INT 21h

; Define o rótulo FIM
FIM:

; Finaliza o programa
MOV AH,4Ch
INT 21h

END
```

- 2) Salve o arquivo com o nome **ATIV3\_1.asm** e entregue pelo CANVAS
- 3) Monte-o e execute-o
- 4) O que o programa faz?

### **ATIVIDADE 3.2**

Crie um programa em linguagem assembly chamado **03\_2.asm** que exiba uma mensagem na tela solicitando ao usuário que digite um caractere, leia o caractere digitado do teclado, exiba uma mensagem na linha seguinte dizendo se o caractere digitado é uma letra, um número ou um caractere desconhecido. Depois o programa deverá solicitar novamente o caractere e fazer o teste de novo. A condição de parada será o caractere ESC. Quando terminar o programa imprimir mensagem de fim de programa.

#### **Exemplo:**

Digite ESC para finalizar ; se não foi for ESC

Digite um caractere: 2

O caractere digitado e um numero.

Digite ESC para finalizar ; se não foi ESC

Digite um caractere: A



**Laboratório de  
Organização de Computadores  
Atividade 3**

O caractere digitado e uma letra.

Digite ESC para finalizar ; se não foi ESC

Digite um caractere: ?

O caractere digitado e um caractere desconhecido.

Digite ESC para finalizar ; se digitou ESC ....

Fim do progarma

- 1) Salve o arquivo com o nome **ATIV03\_2.asm** e entregue pelo CANVAS
- 2) Monte-o e execute-o