

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 caracter ":".

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 - Programa: NUMERO.ASM

- 1) Crie a pasta **C:\Temp\OC**
- 2) Faça o download dos arquivos **TASM.EXE** e **TLINK.EXE** na pasta **C:\Temp\OC**
- 3) Abra o **Bloco de Notas** (ou o **JEdit** / ou o **Notepad++**) e digite o programa a seguir:

```
TITLE Numero
.MODEL SMALL
.STACK 100h
.DATA
    MSG1 DB "Digite um caracter: $"
    SIM DB 10,13,"O caracter digitado e um numero.$"
    NAO DB 10,13,"O caracter 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 caracter: ")
MOV AH,9
LEA DX,MSG1
INT 21h

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

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

- ```
; Compara o caracter em BL com o valor 48 (código ASCII do caracter "0")
CMP BL,48
; Se o caracter em BL for menor que 48 ("0"), salta para o rótulo NAOENUMERO
JB NAOENUMERO
; Compara o caracter em BL com o valor 57 (código ASCII do caracter "9")
CMP BL,57
; Se o caracter 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
LEA DX,SIM
INT 21h
; Salta para o rótulo FIM
JMP FIM
; Define o rótulo NAOENUMERO
NAOENUMERO:
; Exibe na tela dizendo que o caracter não é um número
MOV AH,9
LEA DX,NAO
INT 21h
; Define o rótulo FIM
FIM:
; Finaliza o programa
MOV AH,4Ch
INT 21h
END
```
- 4) Salve o arquivo com o nome **NUMERO.ASM** na pasta **C:\Temp\OC**
  - 5) Abra o **DOS Box**
  - 6) Execute o comando: **mount C C:\Temp\OC**
  - 7) Execute o comando: **C:**
  - 8) Execute comando: **DIR** (verifique se os arquivos **TASM.EXE**, **TLINK.EXE** e **NUMERO.ASM** estão na pasta atual)
  - 9) Para compilar o programa, execute o comando: **TASM NUMERO.ASM**
  - 10) Execute o comando: **DIR** (verifique se o arquivo **NUMERO.OBJ** foi criado com sucesso)
  - 11) Para gerar um executável, execute o comando: **TLINK NUMERO.OBJ**
  - 12) Execute o comando: **DIR** (verifique se o arquivo **NUMERO.EXE** foi criado com sucesso)
  - 13) Execute o programa com o comando: **NUMERO.EXE**

### **Atividade para entrega**

Crie um programa em linguagem assembly chamado **ATIV3.ASM** que exibe uma mensagem na tela solicitando ao usuário que digite um caracter, lê o caracter digitado do teclado, exibe uma mensagem na linha seguinte dizendo se o caracter digitado é uma letra, um número ou um caracter desconhecido.

#### **Exemplo:**

C:\> ATIV3.EXE

Digite um caracter: **2**

O caracter digitado e um numero.

Ou

C:\> ATIV3.EXE

Digite um caracter: **A**

O caracter digitado e uma letra.

Ou

C:\> ATIV3.EXE

Digite um caracter: **?**

O caracter digitado e um caracter desconhecido.

#### **ENTREGA**

Cada aluno deve:

- 1) Criar uma pasta em seu escaninho no AVA com o nome **Atividade3**
- 2) Postar o arquivo **ATIV3.ASM** dentro da pasta **Atividade3**.