MC404 - Organização de Computadores e Linguagem de Montagem
Instituto de Computação
Universidade Estadual de Campinas

Lab. 1

1 Objetivos

Nesse laboratório, o aluno deve familiarizar-se com o montador NASM, ser capaz de criar programas em Assembly, monta-los no formato .COM e executa-los em um PC.

2 Atividade 1_A: Somador de dois dígitos

O programa a seguir é capaz de somar dois dígitos entrados pelo teclado e escrever o resultado na tela. Não é feita nenhuma validação da entrada, nem formatação da saída. Ele servirá apenas de exemplo do uso do montador.

```
[BITS 16]               ; Gera codigo de 16 bits
[ORG 0x0100]            ; Indica que o endereco inicial e': 100h
                        ; obrigatorio para arquivos .COM

[SEGMENT .text]         ; Segmento de codigo
    call LeDigito       ; Le um numero
    mov  [NUM1], ax     ; Guarda o numero na variavel NUM1
    call LeDigito       ; Le outro numero
    mov  [NUM2], ax     ; Guarda na variavel NUM2
    mov  bx, [NUM1]
    add  ax, bx         ; Soma os dois numeros
    mov  [RESULTADO], ax; Guarda o resultado em RESULTADO
    call Imprime        ; Imprime AX
    int  20h            ; Encerra o programa

LeDigito:               ; Le um digito e retorna o valor em AX
    mov  ah, 0          ; Servico 0: Le caracter do teclado
    int  16h            ; Rotina 16h da BIOS: Leitura do teclado
    sub  al, '0'        ; Transforma de ASCII para decimal o caracter lido
    xor  ah, ah         ; Zera a parte superior de AX
    ret

Imprime:                ; Imprime o numero contido em AX (0 <= AX <= 19)
    cmp  ax, 9          ; Testa se o numero tem 1 digito apenas
    jle  MenorQ10       ; > 9, o digito 1 deve ser impresso
    push ax             ; Salva AX pois a interrupcao ira altera-lo
```

```
    mov  ah, 14          ; Servico de impressao de caracter
    mov  al, '1'         ; Caracter a ser impresso
    mov  bl, 7           ; Cor do caracter
    int  10h             ; Rotina 10h da BIOS: Escrita na tela
    pop  ax              ; Restaura o AX guardado
    sub  al, 10          ; Retira o digito ja impresso
MenorQ10:
    mov  ah, 14          ; Servico de impressao de caracter
    add  al, '0'         ; Transforma o digito em numero ASCII
    mov  bl, 7           ; Cor do caracter
    int  10h             ; Rotina 10h da BIOOS: Escrita na tela
    ret

[SEGMENT .data]          ; Segmento de dados
NUM1      dw 0           ; Primeiro numero
NUM2      dw 0           ; Segundo numero
RESULTADO dw 0           ; Terceiro numero
```

Algumas diferenças entre esse programa e o visto em sala de aula:

[BITS 16] Indica que o montador deve gerar código de 16 bits;

[ORG 0x0100] Indica ao montador que a primeira instrução do programa deve ser considerada como sendo o endereço 100h. Isso é uma característica dos programas .COM, que reservam os primeiros 256 bytes para passagem de informações sobre os programas;

[SEGMENT .text] Indica o início do segmento de código, onde as instruções do programa devem ser colocadas. Como nos programas .COM, o código fonte é um mapa direto da memória, a primeira instrução ficará exatamente no endereço 100h. Sendo assim, não é possível colocar as variáveis antes dessa instrução e por isso elas foram movidas para o final do programa;

[SEGMENT .data] Indica o início do segmento de dados, onde as variáveis devem ser colocadas;

Após digitar o programa, grave-o com o nome `lab1A.asm` (o sufixo .asm indica um programa em Assembly). E execute o nasm para criar um arquivo .COM:

```
nasm -f bin lab1.asm -o lab1.com
```

Maiores informações sobre as opções de linha de comando podem ser encontradas no manual do NASM.

3 Atividade 1_B: Validação de Entrada

Desenvolver e incorporar ao programa da atividade 1_A uma rotina que faz a validação da entrada, ié que verifica se o dado de entrada é um valor válido. Caso o valor lido não seja um valor válido o programa deve terminar imprimindo uma mensagem de erro.

4 Atividade 1_C: Calculadora

Desenvolver um programa que efetue as 4 operações básicas sobre 2 números de entrada. Para isso você deve:

- Criar a rotina LeNumero, baseada em chamadas à rotina LeDigito, que leia um número inteiro e retorne em AX. Cada caracter lido deve ser mostrado na tela. Não é necessário permitir a edição do número;
- Criar uma rotina para ler a operação a ser executada (adição, subtração, multiplicação e divisão). Essa rotina deve ser chamada entre a leitura dos números;
- Efetuar a operação e mostrar o resultado na tela.