



LINGUAGEM DE MONTAGEM (8086) – PARTE II

Objetivos

- Estudar e executar programas em linguagem “Assembly / 8086”;
- Aprimorar/desenvolver programas básicos em linguagem “Assembly / 8086”;
- Estudar novas instruções do 8086 e utilizar serviços da BIOS/DOS via interrupção de *software*..

Roteiro

1. O programa da listagem 1 exibe na tela o conteúdo da variável ‘x’ (2 dígitos). Crie uma nova subrotina capaz de imprimir números decimais de até 3 dígitos.
2. Considere o programa da listagem II, o qual está incompleto. Complete este programa criando uma subrotina capaz de exibir números decimais de até 4 dígitos. ■
3. A listagem III recebe 1 número de dois dígitos e exibe o seu valor duplicado (2 dígitos). Altere este programa para receber números de até 4 dígitos e exibir números de até 5 dígitos. ■
4. Implemente, execute, entenda e comente o programa da listagem IV. Usando a opção “view” / “memory” no menu do EMU8086 verifique o seguinte: (a) os endereços de vec1, vec2 e vec3; (b) os valores armazenados nas posições de memória de vec3 após a execução do código. ■
5. Altere a listagem IV de forma que o mesmo exiba na tela do computador os dados e resultados **exatamente** no formato indicado a seguir:

Vetor 1 = [a b c d]
Vetor 2 = [e f g h]
Soma = [i j k l]

6. Utilizando ainda como base o programa da listagem IV, escreva um código que calcule o produto escalar entre **vec1** e **vec2**. Armazene em uma nova variável (**escalar**) esse valor. (Lembre-se de que, dados dois vetores no R^3 , descritos por $u = (u_1, u_2, u_3)$ e $v = (v_1, v_2, v_3)$, o produto escalar de **u** e **v** é definido por: $u \bullet v = u_1.v_1 + u_2.v_2 + u_3.v_3$. ■

Tabela 1 – Listagens de programas.

Listagem I	Listagem II	Listagem III	Listagem IV
org 100h jmp ini x db 23 ini: mov al,[x] call imprime_2d .EXIT imprime_2d: mov ah,0 mov dl,10 div dl mov dl,al mov cl,ah mov ah,2 add dl,48 int 21h mov dl,cl add dl,48 int 21h ret	org 100h jmp ini label1 db "tempo = \$" label2 db "dist. = \$" quebra_lin db 0x0d,0x0a,\$' tempo dw 3798 dist dw 9922 ini: lea dx,label1 mov ah,9 int 21h mov ax,[tempo] call imprime_4d lea dx,quebra_lin mov ah,9 int 21h lea dx,label2 mov ah,9 int 21h mov ax,[dist] call imprime_4d .EXIT imprime_4d: ... ret	org 100h jmp ini label1 db "x = \$" label2 db "2*x = \$" quebra_lin db 0x0d,0x0a,\$' ini: lea dx,label1 mov ah,9 int 21h mov ah,1 int 21h sub al,48 mov dl,10 mul dl mov dl,al mov ah,1 int 21h sub al,48 add dl,48 mov ah,0 push ax lea dx,quebra_lin mov ah,9 int 21h lea dx,label2 mov ah,9 int 21h pop ax mov dl,2 mul dl call imprime_2d .EXIT imprime_2d: mov ah,0 mov dl,10 div dl mov dl,al mov cl,ah mov ah,2 add dl,48 int 21h mov dl,cl add dl,48 int 21h ret	org 100h jmp start vec1 db 1, 2, 5, 6 vec2 db 3, 5, 3, 1 vec3 db ?, ?, ?, ? start: lea si, vec1 lea bx, vec2 lead di, vec3 mov cx, 4 sum: mov al, [si] add al, [bx] mov [di], al inc si inc bx inc di loop sum .EXIT

Questões adicionais

- 1) Realize a leitura de uma palavra de 5 caracteres através do serviço 0Ah da INT 21h, e em seguida realize a conversão desta palavra (o usuário deve apenas digitar os caracteres de 0-9) em um número salvo no registrador AX. Para realizar esta tarefa, crie um laço e faça a varredura na área da variável lida utilizando o modo de endereçamento indexado através de SI.
- 2) Realizando a leitura sem eco do teclado, dígito por dígito, implemente uma subrotina capaz de ler 4 números (demais caracteres devem ser ignorados). Caso o caractere seja válido, faça a impressão na tela antes de realizar a próxima leitura.