Lab4 - fibonacci

sexta-feira, 9 de outubro de 2020 11:20

Nesse laboratório temos que gerar a sequência de Fibonacci e imprimir os números na tela.

Seguência de Fibonacci

```
                0
                1
    0 + 1 = 1
    1 + 1 = 2
    1 + 2 = 3
    2 + 3 = 5
    3 + 5 = 8
    5 + 8 = 13
   8 + 13 = 21
  13 + 21 = 34
      ...
```

Lógica de Fibonacci em assembly

```
12      mov ax,0     ;primeiro elemento da série
13      mov bx,1     ;segundo elemento da série
14  L10:
15      mov dx,ax    ;move o valor de ax para dx para fazer a soma
16      add dx,bx    ;faz a soma e dx,bx
17      mov ax,bx    ;guarda o valor de bx em ax para usar novamente na série seguinte
18      mov bx,dx    ;guarda o resultado da série para a soma seguinte
19      call imprimenumero ;chamando imprime numero
20      cmp dx, 0x8000  ;compara o valor de dx(ultimo valor da série) com o numero 32768 decimal
21      jb L10  ;jump if bellow - se dx é menor que 8000hexa, se for, faz mais um calculo da série, senão, ele s
```

Cada vez que códgo faz a operação add, dx,bx temos o próxi valor da série.

A linha 19, é responsável por imprimir o número na tela.
Para isso, temos que converter o número para a tabela ascii.

Para impressão dos números na tela

Para isso, temos que usar a tabela ascii

A tabela ascii é assim:

ASCII		Macintosh or Windows
Dec	Hex	Result
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

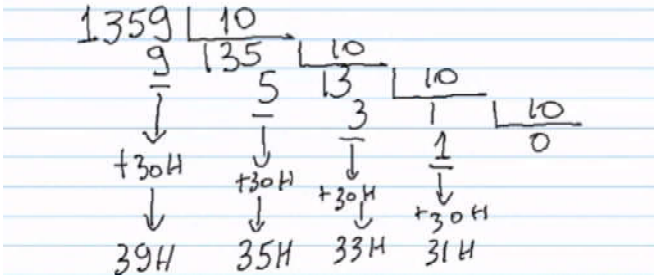
Para imprimir o caracter '9', fazemos:
9+30h = 39
39h é o número correspondente ao '9' na tabela ascii

Para '6':
6+30h = 36
36h é o número correspondente ao '6' na tabela ascii

Lógica do bin2ascii

Dividimos um número por 10 e assim teremos o resto que é o último dígito desse número.

Na imagem abaixo tem o exemplo do número 1359



então, percebe que tivemos que fazer 4 divisões.
A primeira divisão tivemos o resto 9,
A segunda o resto 5 e assim sucessivamente.

Depois de gerado o resultado, somamos com 30h para obter o caracter correspondente na tabela ascii.

Esse trabalho todo começa na linha 19, chamando a função. E continua na linha 32.

Função imprimenumero

A função imprime número, faz exatamente o que o nome diz. Ela vai usar a interrupção usando o mov ah,9 e int 21h. Esse comando imprime caracter na tela.
(Consulte aqui : http://spike.scu.edu.au/~barry/interrupts.html#ah09)

```
32  ;funcao imprime numero
33  imprimenumero:
34      push ax      ;exmpilha ax
35      push bx      ;epilha bx
36      push dx      ;empilha
37      mov di,saida     ;salva o endereço de saída em di
38      call bin2ascii  ;chama a função de conversão
39      mov dx, saida   ;guarda o endereço do vetor saida já com os valores alterados
40      mov ah,9     ;funcao para impressão
41      int 21h      ;interrupção
42      pop dx  ;desimpilhando
43      pop bx  ;desimpilhando
44      pop ax  ;desimpilhando
45      ret ; recupera contexto
46
47  ; fucao bin2ascii
48  bin2ascii:
49      mov ax,dx ;colocando o resultado da serie em ax
50      mov bx,10 ;divisor
51      mov si,4 ;posicao do vetor
52      mov cx,5 ; loop
53  volta:
54      xor dx,dx ;zerando valor de dx
55      div bx
56      add dl,0x30 ;30hexa
57      mov byte[saida + si],dl
58      dec si ;subtrai 1 de si
59      loop volta
```

A função bin2ascii

Essa função se encarrega de fazer a divisão e salvar nas posições dos vetores,
Percebe que na primeira divisão é feito primeiro o número menos significativo, que é o 9, esse número deve ocupar o último espaço do vetor, ficando asssim:

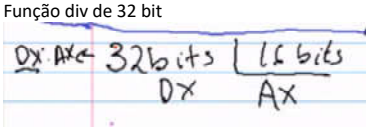
Saida	0	0	0	0	0	
Saida	0	0	0	0	9	Mov byte[saida + 4],dl
Saida	0	0	0	5	9	Move byte[saida +3],dl
Saida	0	0	3	5	9	Move byte[saida +2],dl
Saida	0	1	3	5	9	Move byte[saida +1],dl
Saida	0	1	3	5	9	Move byte[saida +0],l

Sai do loop

Por isso index do vetor (SI) esta subtraindo de 1 na linha 58.

É usado dl no move byte[saida+si],dl , porque o resto ocupa apenas um byte (0 a 9)

Função div de 32 bit



Resultado final: