

Sistemas Embarcados I – Laboratório 04

Objetivo: Implementar programas que fazem chamadas de funções.

O trecho de código abaixo calcula a série de Fibonacci.

1.	segment code	
2.	..start:	
3.	mov	ax, dados
4.	mov	ds, ax
5.	mov	ax, stack
6.	mov	ss,ax
7.	mov	sp,stacktop
8.	; AQUI COMECA A EXECUCAO DO PROGRAMA PRINCIPAL	
9.	mov	dx,mensini ; mensagem de inicio
10.	mov	ah,9
11.	int	21h
12.	mov	ax,0 ; primeiro elemento da série
13.	mov	bx,1 ; segundo elemento da série
14.	L10:	
15.	mov	dx,ax
16.	add	dx,bx ; calcula novo elemento da série
17.	mov	ax,bx
18.	mov	bx,dx
19.	cmp	dx, 0x8000
20.	jb	L10
21.		
22.	; AQUI TERMINA A EXECUCAO DO PROGRAMA PRINCIPAL	
23.	exit:	
24.	mov	dx,mensfim ; mensagem de inicio
25.	mov	ah,9
26.	int	21h
27.	quit:	
28.	mov	ah,4CH ; retorna para o DOS com código 0
29.	int	21h
30.	segment dados	;segmento de dados inicializados
31.	mensini:	db 'Programa que calcula a Série de Fibonacci. ',13,10,'\$'
32.	mensfim:	db 'bye',13,10,'\$'
33.	saida:	db '00000',13,10,'\$'
34.	segment stack	stack
35.	resb	256 ; reserva 256 bytes para formar a pilha
36.	stacktop:	; posição de memória que indica o topo da pilha=SP

A partir deste código, pede-se:

1) Estude o programa. Gere o código executável, usando NASM/FREELINK, carregue-o no *debug* e verifique seu funcionamento. Use a calculadora do Windows para converter números em Hexa para decimal.

2) Na linha 20, troque o comando “jb L10” por “jl L10”, onde “jl= jump if less”. Gere o código executável, usando NASM/FREELINK, carregue-o no *debug* e verifique seu funcionamento. Explique agora o comportamento do código agora usando “jl”.

Crie uma função chamada “imprimenúmero” como se segue:

1	imprimenúmero:
2	;; Aqui, você deve salvar o contexto
3	mov di,saida
4	call bin2ascii
5	mov dx,saida
6	mov ah,9
7	int 21h
8	;; Aqui, você deve recuperar o contexto
9	ret

3) Observe que a(s) chamada(s) da rotina “imprimenúmero” deve(m) ocorrer no programa principal. Dentro de “imprimenúmero”, uma segunda rotina, “bin2ascii”, deve ser chamada para fazer a conversão do número da série calculado para uma string de caracteres ASCII a ser mostrada na tela. Esta string deve estar contida no vetor de caracteres “saída”, declarado na linha 33 do primeiro código apresentado, e precisa ser montada caractere a caractere. Portanto, você deve também implementar “bin2ascii”. Observe que “imprimenúmero” recebe como parâmetro a variável global “saída”. Já “bin2ascii” recebe como parâmetro o registrador “DI”.

4) Uma vez que tenha funcionado corretamente, modifique a passagem de parâmetros de “bin2ascii” de modo a que ela seja feita pela pilha. Ou seja, “imprimenúmero” deve ser implementado como se segue:

1	imprimenúmero:
2	;; Aqui, você deve salvar o contexto
3	mov di,saida
4	push di ; Aqui se empilhou DI. Use SP e BP em “bin2ascii” para pegar DI na pilha.
5	call bin2ascii
6	mov dx,saida
7	mov ah,9
8	int 21h
9	;; Aqui, você deve recuperar o contexto
10	ret

ATENÇÃO:

**Ao trabalhar com pilha, muito cuidado para não gerar estouro de pilha.
Para o item 4, como deve agora ser o comando “ret” de “bin2ascii”?**