

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM - Laboratório 8

- Programa para calculo do fatorial

```

FACTORIAL PROC NEAR
    PUSH BP
    MOV BP,SP
    CMP WORD PTR[BP+4], 1
    JG END_IF
    MOV AX,1
    JMP RETURN
END_IF:
    MOV CX, [BP+4]
    DEC CX
    PUSH CX
    CALL FACTORIAL
    MUL WORD PTR[BP+4]
RETURN:
    POP BP
    RET 2
FACTORIAL ENDP
END MAIN
    
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM - Laboratório 8

O Programa de teste coloca 3 na pilha e chama factorial fazendo o seguinte aspecto

SP	↗		(original) return addr 3 (valor de N)		↖ BP (primeira chamada)
SP	↗		BP (primeira chamada) return addr (linha 28) 2 (valor de N) BP (original) return addr (linha 28) 3		↖ BP (segunda chamada)
SP	↗		BP (segunda chamada) return addr (linha 28) 1 (valor de N) BP (original) return addr (linha 28) 2 (valor de N) BP (original) return addr (linha 8) 3		↖ BP (segunda chamada)

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM - Laboratório 8

•OUTRO EXEMPLO DE UMA PROCEDURE RECURSIVA É O CÁLCULO DO MAIOR NÚMERO DE UMA ARRAY DE N NÚMEROS INTEIROS:

```
PROCEDURE FIND_MAX (INPUT: N, OUTPUT:MAX)
IF N = 1
  THEN
    MAX = A[1]
  ELSE
    CALL FIND_MAX(N-1, MAX)
  IF A[N] > MAX
    THEN
      MAX = A[N]
    ELSE
      MAX = MAX
  END IF
RETURN
```

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM - Laboratório 8

Exercício 1. Escreva um programa assembly recursivo que calcule o fatorial de um número N. O número N deve ser uma entrada do programa.

Exercício 2. Escreva um programa assembly recursivo que ache o maior dos elementos de um array.

Exercício 3: Escreva um programa assembly recursivo para calcular a soma dos elementos de um array.

Exercício 4: Escreva um programa assembly recursivo que calcula a série de Fibonacci (1,1,2,3,5,8,13,21,34,55,...), para um determinado valor N, que deve ser a entrada do programa.

$$F(0) = F(1) = 1$$

$$F(n) = F(n-1) + F(n-2) \quad \text{se } N > 1$$

OBS.

- Usar procedimentos, indentação do código, comentários e usar o TD para verificar os erros de lógica que podem estar acontecendo.
- Entregar junto com os códigos, um esquema de como a recursão está sendo tratada (ver exemplo no slide 6-4)
- Entregar na próxima aula (segunda)

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM - Laboratório 8

Exercícios

- 1. Edite o programa anterior – utilize o EDIT sob o DOS**
 - > EDIT nome do programa (extensão .asm)**
- 2. Gere o código objeto – utilize o montador TASM**
 - > TASM nome do programa (extensão .asm)**
- 3. Gere o executável – utilize o ligador TLINK**
 - > TLINK nome do programa (extensão .obj)**
- 4. Execute o programa**
 - > nome do programa (extensão .exe)**
- 5. Gere o código objeto com informações para o debugger**
 - > TASM nome do programa (extensão .asm) /zi**
- 6. Gere o código executável com informações para o debugger**
 - > TLINK nome do programa (extensão .obj) /v**
- 7. Utilize o debugger e tire as informações sobre os registradores**
 - > TD nome do programa (extensão .exe)**