**Terna Engineering College**
**Computer Engineering Department**

**Program: Sem V**

**Course: Microprocessor Lab**

**Faculty: ARATHI BOYANAPALLI**

LAB Manual

**PART A**

(PART A: TO BE REFERRED BY STUDENTS)

# Experiment No. 7

**A.1 Aim:**
Write an assembly program to find the factorial of a number using the recursive procedure.

**A.2 Prerequisite:**
Basic knowledge of 8086 instruction set, interrupts procedures and macros in 8086.

**A.3 Outcome:**
After successful completion of this experiment, students will be able to -

1. Use appropriate instructions to program microprocessors to perform various tasks.
2. Develop the program in assembly/ mixed language for Intel 8086 processor.
3. Demonstrate the execution and debugging of assembly/ mixed language programs.

**A.4 Theory:**
A recursive procedure is one that calls itself. There are two kinds of recursion: direct and indirect. Indirect recursion, the procedure calls itself and in indirect recursion, the first procedure calls a second procedure, which in turn calls the first procedure.
Recursion could be observed in numerous mathematical algorithms. For example, consider the case of calculating the factorial of a number. Factorial of a number is given by the equation –

Fact (n) = n * fact (n-1) for n > 0

For example factorial of 5 is 1 x 2 x 3 x 4 x 5 = 5 x factorial of 4 and this can be a good example of showing a recursive procedure. Every recursive algorithm must have an ending condition, i.e., the recursive calling of the program should be stopped when a condition is fulfilled. In the case of the factorial algorithm, the end condition is reached when n is 0.

**A.5 Algorithm**

1. Start.
2. Initialize data segment and code segment.
3. Get the number in BX.
4. Load AX registers with 1.
5. Call Factorial Proc.
6. Compare BX with AX.
7. Jump to return if JE=1.
8. Else Push BX and DEC the value.
9. Do Multiplication.
10. Return.

# PART B

(PART B: TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)***

| | |
|---|---|
| Roll No.: 50 | Name: Amey Thakur |
| Class: TE-Comps B | Batch: B3 |
| Date of Experiment: 08/10/2020 | Date of Submission: 08/10/2020 |
| Grade: | |

## B.1 Observations and learning:

***(Software Code written by a student and output of the program)***

- **Input to find factorial of a number -**

```
section.text
global _start
        _start:
          MOV bx, 3
          call  proc_fact
          add   ax, 30h
          MOV  [fact], ax
          MOV edx,len
          MOV ecx,msg
          MOV ebx,1
          MOV eax,4
          int    0x80
          MOV   edx,1
          MOV ecx,fact
          MOV ebx,1
          MOV eax,4
          int    0x80
          MOV eax,1
          int  0x80

        proc_fact:
          cmp   bl, 1
          jg    do_calculation
          MOV   ax, 1
          ret
        do_calculation:
```

```
    dec   bl
    call  proc_fact
    inc   bl
    mul   bl
    ret

section.data
    msg db 'Factorial 3 is:',0xa
    len equ $ - msg

section .bss
    fact resb 1
```

- **Output (Factorial) -**

```
Compilation time: 0.24 sec, absolute running time: 0.16 sec, cpu time: 0.01 sec, memory peak: 5 Mb, absolute service time: 0,55 sec

Factorial 3 is:
6
```

**B.2 Conclusion:**

***(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)***

We successfully learned assembly language program to find factorial of a number using the recursive procedure.

**B.5 Question of Curiosity**

**Q1**. what is a recursive procedure and its functioning with an example program in 8086

**Ans:**

A recursive procedure is a procedure which calls itself. This results in the procedure call to be generated from within the procedures again and again. Every recursive procedure must have an ending condition, i.e., the recursive calling of the program should be stopped when a condition is fulfilled.

E.g. Recursive procedure for finding factorial:

```
  DATA SEGMENT
  ANS DW ?
  val db 5
DATA ENDS
CODE SEGMENT
      FACTO PROC
        mul bl
        dec bl
        mov cl , bl
        cmp cl, 01
```

```
        mov ans , ax
        jnz next
        jmp output
      next:
        call FACTO
      FACTO endp
START:
  mov ax,data
  mov ds, ax
  mov bl , val
  mov al,1
  call FACTO
 output:
  mov ans,ax
  hlt
CODE ENDS
END START
```

Explanation:
Here the procedure fracto has some instructions in it which is followed by a call to itself, each time fracto is executed it calls itself again. The stopping condition for this recursive procedure is CMP CL,01 when this condition is satisfied the control of the program is transferred to the next block through a conditional jump JNZ . The procedure carries out repeated(recursive) multiplication to generate the factorial.

**Q2.** Write a recursive program in 8086 ALP to find the sum of first n integers?
**Ans:**

```
assume cs: code, ds: data

DATA SEGMENT
        ANS DB?
        VAL DB 4
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE,DS:DATA
    FACTORIAL PROC
        ADD AL, BL
        DEC BL
        MOV CL, BL
        CMP CL, 00
        MOV Z , AL
        JNZ NEXT
        JMP FILLY
        NEXT:
        CALL FACTORIAL
```

```
FACTORIAL ENDP
START:
    MOV AX, DATA
    MOV DS, AX
    MOV BL, VAL
    MOV AL, 0
    CALL FACTORIAL
    FILLY:
    MOV Z, AL
    HLT
    CODE ENDS
 END START
```

- **Output -**