MES's Wadia College of Engineering, Pune-1

SUBJECT: MICROPROCESSOR LAB (MPL) **NAME:**	
CLASS:	**ROLL NO.:**
SEMESTER: SEM-II	**YEAR: 2023-24**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED:	

Assignment No-09

Title:- Find factorial of a given integer number on a command line by using recursion

Assignment Name:-Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code.

Objective-

- To understand the assembly language program
- To understand the concept of recursion
- Able to Implement factorial of a integer number using recursive method

Outcome-

- Students will be able to understand different assembly language instruction.
- Students will be familiar with the format of assembly language program and able to Apply the concept of recursion to find factorial of a number

Prerequisite -

System call of Unix for Assembly language Program.

Hardware Requirement-

Desktop PC

Software Requirement-

Ubuntu 14.04,

Assembler: NASM version 2.10.07
Linker: ld

Introduction:-

THEORY:

A recursive procedure is one that calls itself. There are two kind of recursion: direct and indirect. In direct 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 factorial algorithm, the end condition is reached when n is 0.

Instructions needed:

1. AND-AND each bit in a byte or word with corresponding bit in another byte or word

2. INC-Increments specified byte/word by1

3. DEC-Decrements specified byte/word by1

4. JG - The *command JG* simply means: Jump if Greater.

5. CMP-Compares to specified bytes or words

6. MUL - The MUL (Multiply) instruction handles unsigned data

7. CALL-Transfers the control from callingprogramto procedure.

8. ADD- ADD instructions are used for performing simple addition of binary data in byte, word and doubleword size, i.e., for adding 8-bit, 16-bit or 32-bit operands, respectively.

9. RET-Return from where call is made

Algorithm:-

This algorithm use recursive approach to find factorial of N.

1. Start
2. Read: Take input N
3. Retrieve parameter and put it into Register-PUSH
4. Check for base case if n==0

5. move the first argument to %rax
6. If the number is 1, that is our base case, and we simply return.
7. multiply by the result of the last call to factorial.
8. return to the function

Conclusion:-

Questions:-

1. What is Control transfer instructions. Explain in details
2. What different conditions used to find factorial of an integer number.
3. Explain CALL, JG, ADD instructions
4. Explain Pop and Push instruction in detail.