

ASSIGNMENT NO. 9

AIM :- 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.

APPARATUS :

- Core 2 duo/i3/i5/i7 - 64bit processor
- OS – ubuntu 32bit/64bit OS
- Assembler used –nasm (the netwide assembler)
- Editor Used – gedit

THEORY :

Factorial of a number:

In mathematics, the factorial of a non-negative integer n , denoted by $n!$, is the product of all positive integers less than or equal to n . For example,

factorial(3)

=> $3 * \text{factorial}(2)$

=> $3 * 2 * \text{factorial}(1)$

=> $3 * 2 * 1 * \text{factorial}(0)$ { factorial(0) => 1 }

=> $3 * 2 * 1 * 1$

=> 6

Recursion:

In computer programming, a recursion (noun, pronounced ree-KUHR-zhion) is programming that is recursive (adjective), and recursive has two related meanings:

1) A recursive procedure or routine is one that has the ability to call itself. This usually means that it has the capability to save the condition it was in or the particular process it is serving when it calls itself (otherwise, any variable values that have been developed in executing the code are overlaid by the next iteration or go-through). Typically, this is done by saving values in registers or data area stacks before calling itself or at the beginning of the sequence where it has just been reentered.

2) A recursive expression is a function, algorithm, or sequence of instructions (typically, an IF, THEN, ELSE sequence) that loops back to the beginning of itself until it detects that some condition has been satisfied.

Algorithm:

1. Accept Number from User
2. Call Factorial Procedure
3. Define Recursive Factorial Procedure
4. Display Result.

Algorithm : Factorial using recursion

1. Push rcx value onto stack
2. Compare rcx value with 1
3. if Not equal to one then
 - Decrement rcx by one
 - Call factorial procedure again (recursion)
4. If equal to one then
 - Pop rcx value from stack
 - multiply that value with answer variable
 - Store result of multiplication again in answer variable

5. stop

CONCLUSION:
