

## Writing x86 Assembly (formula.c)

---

### *Design*

This program takes in one integer argument. It first runs an error check to make sure that valid number of arguments have been entered. Then, if the user runs formula with the help flag (-h), the program would print “Usage: formula <positive integer>”. After that the program runs another check to make sure the entered argument is an integer greater than or equal than 1. If everything checks out, the programs then converts the given string argument to an integer.

The program now moves into the main part of printing the binomial expression. The program prints out the “long” form of  $(1+x)^n$ , where  $n$  is equal to the argument. However, it needs to first calculate the expression. To do that, it call the nCr function. The nCr function takes in two parameters  $n$  and  $r$  and calculates the binomial coefficient using the formula:

$$nCr = n! / (n - r)!$$

To do the calculation, the nCr function calls factorial function which takes in one parameter and computes the factorial of the given input using recursion. Both these functions were written in nCr.c file and later converted to nCr.s file. The program also takes care of the number where an integer overflow can be caused.

After all the calculations and printing is done, the program prints out its execution time using *gettimeofday()* function in microseconds.

### *Challenges*

- Understanding and writing nCr in assembly
- Using *gettimeofday()* function

### *Analysis*

This program runs in linear time. The most expensive thing that the program does is printing out the binomial expansion. It also occupies linear amount of space.