Formula - David Parsons

This directory is made up of two things: x86 assembly that holds nCr and factorial methods and a C library that implements the assembly.

Formula Library

Takes 1 arguments from the command line.

power - used to determine the long form of (1 + x)n, where n is the argument power.

NOTES on Formula

My implementation of the Formula is contingent upon the power parameter being a non-negative integer as mentioned in the project spec.

My implementation also does not account for overflow despite being outlined in the spec, as Professor Russell stated in class we did not need to.

Because of the limitations of the 32 bit registers used my implementation is also only accurate up to a power of 12.

nCr.s

Contains assembly code written in x86 that represents the nCr and Factorial functions use in the formula library.

Both functions use only registers and do not rely upon the creation of local variables. This was done to avoid the overhead associated with allocating space and keeps the stack frame small.

Factorial Runtime - *O(n)*, as the function has one iterative loop the multiplies the values from (output * n-1), n-1 times.

nCr Runtime - *O(n)*, as the function simply calls Factorial 3 times and performs one division operation. So 3 * runtime(factorial) or 3n.

nCr Function Walkthrough

1. pushes base pointer
2. copies stack pointer into base pointer
3. puts first parameter 'n' into ecx

4. subtracts the value of the second parameter 'r' from ecx resulting in 'n-r'
5. makes ecx the first parameter for the next call
6. calls factorial on ecx
7. unallocates space for parameter
8. copies eax '(n-r)!' into ecx
9. copies second parameter 'r' into ebx
10. pushes value in ebx so its the first parameter in the next call
11. calls factorial on ebx - eax contains 'r!'
12. unallocates space for parameter
13. multiplies eax and ecx, so now ecs contains 'r!x(n-r)!'
14. puts first parameter 'n' into ebx
15. makes ebx the first parameter for the next call
16. calls factorial on ebx - eax contains 'n!'
17. unallocates space for parameter
18. divides eax by ecs or 'n!' / 'r!x(n-r)!'

Factorial Function Walkthrough

1. pushes base pointer
2. copies stack pointer into base pointer
3. copies the first parameter into the return register eax
4. Compares the return register and therefore input parameter to 1
5. jumps to .L1 if the parameter is greater than 1
6. Otherwise copy 1 into the return parameter
7. jump to .L2 to return value of 1
8. subtracts one from the first parameter
9. multiplies the return register by the first parameter
10. compares 2 to the current state of the first parameter
11. jumps to .L1 if the parameter is still greater than 1 or not less than 2

formula.c

C Library that implements nCr.s in order to complete the formula assignment.

Formula Runtime - *O(n^2)*, the main method takes in a input value and computes n values based upon the input. The calculation of each said value is O(n) because the value is computed by running the nCr function. Therefore n * O(n) is O(n^2).

main

1. Converts argument from ascii to integer, storing it in power.

2. Assures the proper number of arguments are used.
3. Gets the current time for calculating runtime.
4. Prints long for of polynomial by calling nCr from nCr.s
5. Gets the current time for calculating runtime.
6. Subtracts and prints the two recorded times that represent the runtime of the main method.