Assembly Language Program to find nth Fibonacci Number

Author : KR Hariharan CS19B079

Date : 03/03/2021

Assignment number : 3

Method of Observation

Output displayed in the form nth Fibonacci number, start timestamp, and timestamp. Output corresponding first 41 FIbonacci number [f(0) to f(40)] obtained using bash script.
Timestamps converted from two 32 bit signed integer representation to 1 64 bit unsigned integer and the difference between start and finish timestamps calculated for n = 0 to n = 40 using python script. This difference is representative of the amount of time (cycles) the program took to execute and display the nth Fibonacci number.

Observation

From the timestamp data, and from general observation, we can see that as n increases, the increase in time for the **iterative program** is not much. This is because the program has one loop with n-1 iterations, and thus it has linear time complexity.

On the other hand, the **recursive program** is similarly fast for small n. However, as n increases, the time taken increases significantly. This is because, for the recursive function,

$$T(n) = T(n-1) + T(n-2) + O(1)$$

Upon solving, we get an exponential time complexity [((1+root(5))/2)^n]. Just as we have observed, as n increases, algorithms with exponential time complexity become very slow.