Mystery Documentation

Matt Klein

April 4, 2013

1 Intro and About

This program, `myster`, will read in an integer and display n^{th} fibonacci number.

2 Data Structures

This program uses built in C ints as the data structure, so there it can only store up to $2^{31} - 1$ for an integer. Anything larger will crash the program, and the program does not allow you to input 47 or above.

3 Run Time Analysis

The worst case of each function is the following:

- `Add -` $O(1)$

  This is always $O(1)$ as we are ust adding both x and y.

- `Compute Fib -` $O(n)$

  This is always $O(n)$ where n^{th} fibonacci number to find. It uses recursion, due to simplicity.

4 Space Analysis

The program can only read up to $2^{31} - 1$, due to the constraints of the program. Afterword, pointers are freed using `free`. Anything larger than 46 gives overflow, and the program will alert the user the integer is too long. Similary, anything less than zero will show the integer is negative. Providing too many arguments will also display to the user, as is no arguments.

5 Figuring it out

I figured out that `mystery.s` is doing a fibonacci sequence because a function called `compute_fib` was left in. I did not do the proper optimizations that were included in the original assembly file, since I did not know what was going on with some registers and with some of the mnemonics. My code is most likely different due to not using an array, and not sign extending as the original is.