Richard Moot
CS162 – 400
December 4, 2015

Lab 10

Expectations

Fibonacci:

Based on the work I have been doing in CS 271 (Computer Architecture & Assembly), I would expect that the recursive approach would be much slower since each call against the function would set a new stack frame. Since it cannot compute the next number until it hits the base case, as N grows larger it would take exponentially longer for it to compute.

Factorial:

I expect that the factorials would have a similar result as Fibonacci, but will less of a discrepancy due to tail recursion optimization. Since the compiler can identify when a function is tail recursive, it can possibly optimize the function to run in a more iterative fashion in order to save resources.

Analysis

My expectations were correct for Fibonacci's different functions. Only in the case of really small values of N did the recursive function operate slightly faster (maybe about 2 to 5 clock cycles less). When values approached up to 40, there was a stark difference between the iterative approach and the recursive approach. Despite having set "long long" types, it would cause the system to hang at around 44+ for the user input.

The most interesting part was that the factorial functions appeared to have no difference between the iterative, the recursive, and the non-tail recursive. No matter what values I had input they had all performed in nearly the same amount of time. They were extremely fast (only taking 0 to 3 clock cycles to run). I had not realized that tail recursion could be optimized so greatly that it would get nearly the same performance between the different functions.

Examples

Large Value	Small Value
Enter a positive number:40	Enter a positive number:5
40th Fibonacci number: 102334155	5th Fibonacci number: 5
Iterative Fibonacci took 28 clock cycles or 2.8e-05 seconds	Iterative Fibonacci took 13 clock cycles or 1.3e-05 seconds
40th Fibonacci number: 102334155	5th Fibonacci number: 5
Recursive Fibonacci took 1021154 clock cycles or 1.02115 seconds	Recursive Fibonacci took 5 clock cycles or 5e-06 seconds
Iterative factorial took 1 cycles or 1e-06 seconds	Iterative factorial took 0 cycles or 0 seconds
Non-tail recursion factorial took 1 cycles or 1e-06 seconds	Non-tail recursion factorial took 1 cycles or 1e-06 seconds
Tail recursion factorial took 1 cycles or 1e-06 seconds	Tail recursion factorial took 1 cycles or 1e-06 seconds