Julio Berina

CS 2640

Project 2

10/26/18

Project 2 Report

My approach for doing the project was to first write Fibonacci code in another programming language since it's a lot easier to write the program in something more higher-level. I did this to figure out the highest number I can express with a 32-bit signed and unsigned integer. The highest Fibonacci number I can express with a signed integer is 1836311903 and 2971215073 for unsigned before rollover. Since the SPIM simulator can only output signed integers, I decided to hard code 2971215073 as a string in the data section and output it when necessary (n = 48).

For inputting n, I decided to be simplistic with my approach. I declared the 1st Fibonacci number to be 0 instead of 1. Although I validated input by only allowing numbers 1-48, I didn't report whether or not the number was too small or too big. If the input was less than 1 or bigger than 48, I simply branched to the **exit** subroutine. In other words, if I can't work with the input, just go ahead and exit.

Because of the issue with the SPIM simulator only outputting signed numbers, I took special precautions in outputting the numbers in the array. I have subroutines and made the right jumps just in case n = 48. Also, I only loaded as many into the array as the input of n. If n = 10, for example, I only loaded 10 items into the array. It was pretty straightforward to calculate the nth Fibonacci number because after loading n items into the array, the address at register $s0 was pointing to the right of the last element. All I had to do was subtract 4 from its current address, which points me to the address of the nth element.

When it came to outputting the elements, I simply started at 0 and iterated through the array. I jumped 4 bytes per address and just outputted the elements in the array. Again, I also took special precautions in case my n was 48 to prevent the problem with SPIM. I ran the program for n = 10 and n = 20 as stated in the project guidelines. I got 34 for n = 10 and 4181 for n = 20.