

Design Document

Program Flow

1. Check for a valid amount of args
2. Verify the args are valid ones
3. Start Fib loop – loops X amount of times (X being the users inputted number)
4. Using xadd to add up the numbers
5. Check the carry flag
6. Add extra values to another variable based on the carry flag
7. Convert the binary result to a hex/oct string
8. Output the result

Notable Data Structures (In the calculation loop)

- RCX is the counter
- RAX is the user's inputted value
- R8 is the lower fib number
- RSI is used to help calculate R8's number
- RDI is the higher fib number
- RDX is used to help calculate RDI's number

Notable Functions

- There are no functions in asm, only jumping....
 - And pain.

Anticipated Challenges

1. Keeping track of registers
2. Comparing
3. Assembly
4. Carrying

Targeted Features

1. -o for octal output
2. Output in decimal
3. Man page

Architecture

The program will be laid out linearly for readability and my own sanity. First the variables will be declared. Then I will check argc and argv for the correct values. Then I will run a loop to calculate the number. Then convert the binary to two hex strings (one for the lower and upper part of the number). Then print the hex strings.