Recreating Results

All code in EvaluateTimer.c.

Software Clock: Uncomment the commented timespec struct and code associated with it.

Adjusting Resolution: Change the number of iterations of the inner while loop.

- *while(i<100)* results in 100,000 ARM instructions and a corresponding resolution of 0.1 milliseconds
- *while(i<1000)* results in 1,000,000 ARM instructions and a corresponding resolution of 1 millisecond
- *while(i<2000)* results in 2,000,000 ARM instructions and a corresponding resolution of 2 milliseconds
- *while(i<5000)* results in 5,000,000 ARM instructions and a corresponding resolution of 5 milliseconds
- *while(i<10000)* results in 10,000,000 ARM instructions and a corresponding resolution of 10 milliseconds
- *while(i<50000)* results in 50,000,000 ARM instructions and a corresponding resolution of 50 milliseconds
- *while(i<100000)* results in 100,000,000 ARM instructions and a corresponding resolution of 100 milliseconds

Evaluation under different CPU loads: Uncomment the commented *fork()* functions and the associated *if...else* statement. The parent runs the timer, and the children run a dummy prime number finder function.

- For 1 extra process, use execute *fork()* once.
- For 4 total processes, use execute *fork()* twice.
- For 8 total processes, use execute *fork()* 3 times.
- For 16 total processes, use execute *fork()* 4 times.
- For 32 total processes, use execute *fork()* 5 times.
- For 64 total processes, use execute *fork()* 6 times.
- For 128 total processes, use execute *fork()* 7 times.