Alexandra Feely

Computer Systems

Lab 10

Results:

**optExample:**

Using time function -

real 0m4.252s

user 0m4.240s

sys 0m0.000s

With Unroll:

real 0m4.751s

user 0m4.724s

sys 0m0.004s

Using callgrind:

isPrime() called 53,371,745

sqrt() called 11,060,816

getNextPrime() called 695,926

**optExample2:**

Using time function -

real 0m2.915s

user 0m2.832s

sys 0m0.000s

With Unroll:

real 0m3.050s

user 0m3.040s

sys 0m0.000s

Using callgrind:

isPrime called 29,288,998

getNextPrime() called 695,926

sqrt() called 400,004

genPrimeSequence() called 259,005

Observations:

optExample2.c ran faster than its counterpart and calling less functions to process the same amount of data. This is largely due to the fact that it is not calling the sqrt() in a loop, so it is only calculating it once per number tested. This resulted in its runtime being cut down by about 25%. Unravelling the loops actually made both iteration run slightly slower: about 1/10th of a second given the same inputs.