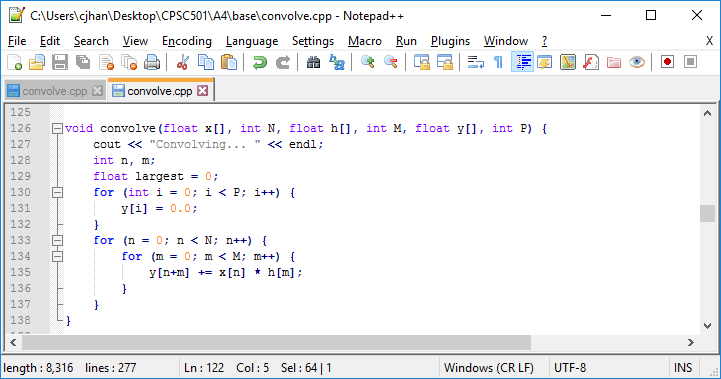
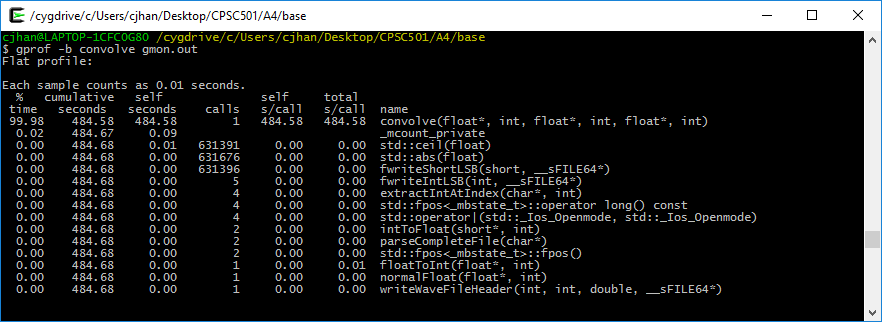
**CPSC 501 Assignment 4 Optimization Report**

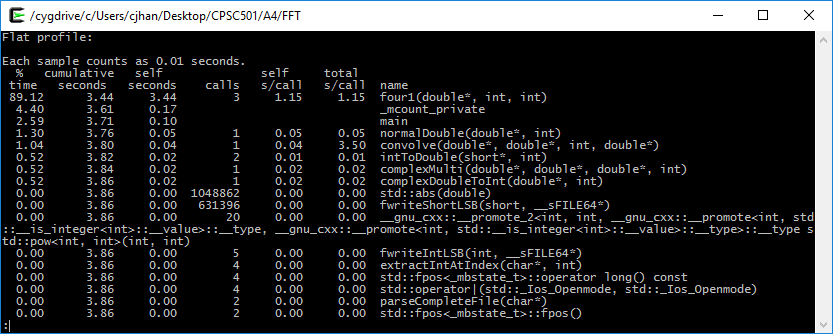
Baseline Program: Our implementation begins with an input-side time-domain convolution algorithm. It operates by parsing both the input and impulse response .wav files for their respective signal data. Once the samples have been extracted, they convolved together using the following convolve function:



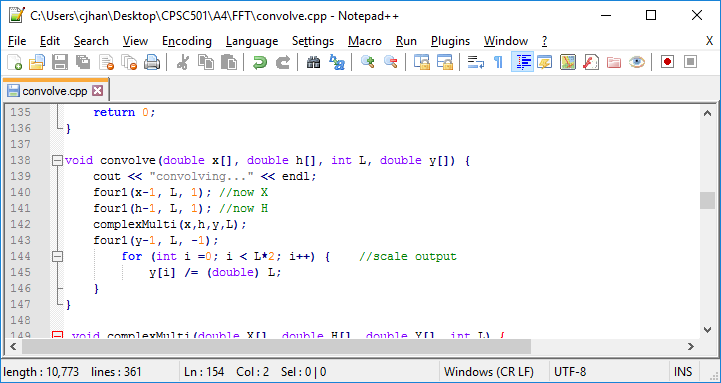
This convolution function, while accurate, is extremely slow. Timing data, collected at initial execution of the convolution of a two-second-long constant 440Hz sine tone with the provided five-second-long impulse response file l960auto\_park.wav resulted in a total profiled convolution time of 484.68 seconds as reported by gprof profiling software:



Algorithm-Based Optimization: The baseline program was optimized by changing the nature of the convolution from a time-domain input-side convolution to a frequency-domain convolution utilizing the provided Fast Fourier Transform function four1:



Hand Tuning 1 (Partial Loop Unrolling): As we can see from the above profile, while four1 has taken over as the bottleneck function, there are several potential candidates for improvements. We will look to them first before attempting any optimizations of four1. Convolve seems a good candidate:



We can immediately detect that the brunt of convolve must come from the for loop which is scaling output from the IFFT call to four1. We can likely reduce the cost of convolve by unrolling the for loop. As we know that L\*2 is a power of 2 by construction, we can be sure that we can handle any smaller power of two cases inside the unrolled for loop. To enhance the effect, we will choose the value 16, leading to the following optimization:

