Ben Haft

Tom Petr

ECSE 4320

1/25/23

**Advanced Computer Systems Project 1 – Multiple Thread Data Compression**

Analysis and Conclusions:

This laboratory exercise was assigned to teach us how to use two real-world tools in software engineering: multithreading and data compression. We successfully completed the lab by utilizing the ZSTD and pthread libraries. While programming our project, we were mindful of efficiency and allocated memory. We found a solution to read data in by 16kB chunks and assign threads to compress the data as the input file was read. This would improve efficiency when compressing large files because some threads could be compressing while the file was still being read and this is an efficient usage of memory since the entire input data was not read at one time. We also freed up dynamic memory when we saw fit.

There were many obstacles to overcome while working on this project. In the beginning, there is little documentation of the “-lzstd” flag that is necessary for compiling any program which uses the ZSTD library. We tried using the streaming compression API and turned back to simple compression because the program was overcomplicated, and we felt it was more suited to a single thread operation. We also attempted to use open\_memstream() as a means to create a buffer of dynamic length. This failed because when we used fwrite(), the data did not appear in our buffer. Another main error we ran into was a faulty decompressing file. The program would only decompress the first frame of the file, so we thought our program was not iterating correctly. We figured out that since we were segmenting our output data, we needed to use the streaming compression API. This allowed us to successfully decompress our “.zst” file.

This lab has many real world applications. Multithreading is an efficient way to load any top of large data. Compression is built into many storage systems and databases. Therefore, pairing them together is a great exercise. We learned how to use multithreading and compression to our advantage in this lab, and are capable of applying this to programming operations in the future.