Trie-d and True

Angela Shen

March 2023

1 Abstract

As the field of computer science grows more complex, the amount of data needed to store and transfer information has increased. In the past, computer memory was quite expensive, with even a single megabyte costing several dollars. But nowadays, it is not unusual for programs, images, and videos to contain several megabytes, or even gigabytes, of data. For example, when one sends memes over Discord, file sizes may be a few megabytes large to send without Discord Nitro, so it may be necessary to compress these images. Many different algorithms have been created to meet this demand for efficient and (usually) lossless data compression. One such method is LZ78.

2 Introduction

LZ78 data compression is implemented with the following files:

Makefile - Compiles the executables decode and encode. Also creates multiple object files (dependencies for decode and encode). For a list of these c files, .c below.

README.md - Details the process of building any necessary files, the command line options for any executables, and any errors or bugs.

DESIGN.pdf - Contains the pseudo code and descriptions of each c file.

WRITEUP.pdf - This document. Describes the assignment in its entirety and discusses the results.

trie.c: - Creates a trie with the TrieNode structure. Used in encode.c to compress the file.

word.c: - Creates a WordTable using the Word structure. Used in decode.c to decompress the original file.

io.c: - Used in decode.c and encode.c to read and write information from infiles and outfiles into a buffer to more efficiently read and write information.

encode.c: - Compresses an infile and outputs the pairs derived from the compression to an outfile. Can accept different infiles (default: stdin), outfiles (stdout), and can print verbose statistics.

decode.c: - Decompresses an infile and outputs the characters derived from the compression to an outfile. Can accept different infiles (default: stdin), outfiles (stdout), and can print verbose statistics.

3 Learning

3.1 Compression

Learning about tables and how they contribute to data compression was interesting. I liked learning about Huffman, even though we did not implement it in this assignment, since it is a really elegant and conceptually simplistic way to effectively do data compression. I think LZ78 is a little more complicated, and it took me a while to fully understand since I kept getting "codes", "symbols", and "pairs" confused.

3.2 syscalls

Every week for the past 9 weeks, CSE13s has punished me for my hubris. For example, I thought fread and fwrite were quite difficult to debug and work with. I was very incorrect. It was interesting to learn how to work around read() and write() not always returning the expected value (namely the number of bytes to be read) and learning to work around that. The man pages for open() and write() were also very intimidating, since there are a lot of macros for specific permissions, not all of which I completely understand, but I know enough to figure out which ones are important for this assignment.

3.3 Entropy

Because of the way a trie is constructed, its adversarial sequence would a completely randomized series of letters. If a sequence is repeated, the trie does not need to create a new node. The more entropy in a file, the more TrieNodes are created, which also means more nodes have to be traversed on average.

3.4 Time Management

I think one of the most important lessons I took from CSE13s was learning to pace myself and start early. It was especially important for this assignment because it is really hard to start without having a holistic understanding of how LZ78 works and the ways individual functions contribute to the implementation. I attended multiple sections for this assignment over the 2 weeks given, primarily for io.c. It did take some debugging, but overall it took a lot less debugging than I expected because I took the time to understand how everything worked. Still could have started earlier though, I did read the assignment the Monday immediately after assignment 5 was due, but I did not really understand it and I probably should have asked for help earlier.

4 Conclusion

It is March 12th. This is the last day this assignment can be turned in on time. CSE13s was an entire experience and I am both sad and ecstatic to let it go. Overall, I think I learned a lot from this assignment and the class in general. Swim High. o7

5 Credit and Notes

Attended Jessie Srinivas' section on 3/6/22 and used some pseudocode for io.c functions (commented in io.c). Implemented decode.c and encode.c based on pseudocode in the assignment spec.