What I Learned About Lempel-Ziv Compression and its Functionality

Alex Yeh

March 10, 2023

1 Introduction

Assignment 6 was centered around Lempel-Ziv Compression. I wrote two programs, encode.c and decode.c, which perform LZ78 compression and decompression, respectively. My encode program can compress any file, text or binary. My decode program can decompress any file, text or binary, that was compressed with encode. Both operate on both little and big endian systems. Both perform read and writes in efficient blocks of 4KB.

2 What I Learned

In this assignment, I learned about how Lempel-Ziv compression works. I also learned the difference between little and big endian systems. The most significant byte is stored first in big endian systems and the least significant byte is stored first in little endian systems. I learned what a trie was and how it can be used to store codes. I also learned about file headers in a compressed file and how a magic number serves as a unique identifier for files compressed by the encode program. The decode program then can only decompress files which have the correct magic number. Finally, I learned about buffers and how we can use buffers to read and write in blocks of 4KB.

3 Functionality of LZ78 Compression

LZ78 compression is a lossless compression algorithm. It utilizes a dictionary to store codes which represent words in the file. The dictionary is initialized with the empty word, or a string of zero length, at the index EMPTY_CODE, which is a macro for 1. The first index in which a new word can be added to is the macro START_CODE, which has the value 2. Characters/symbols are appended to a previously stored code until a new code is found. Together, a code and symbol form a new code. The first character is appended to the empty word, which is stored at index EMPTY_CODE. If we didn't see that word yet, we add it to the dictionary and store it at index START_CODE. We continue on to the next character and append it to the word we just

added to the dictionary. If we didn't see that word yet, we add it to the dictionary and store it at the next available code 3. For example, if we have the word "ab" and we see the character "c", we add the word "abc" to the dictionary and store it at index 3.

4 How Efficiency of My Compression Changes With Entropy

Entropy is defined as the measurement of randomness. If an input file has low entropy, it means that the file has a lot of repeated characters/words. This will result in a smaller compressed and decompressed file size. If an input file has high entropy, it means that the file has a lot of unique characters/words. This will result in a larger compressed and decompressed file size. For example, say we have two text files: like.txt and me.txt.

like.txt contains the following text:
I like to watch football.
I like to watch basketball.
I like to watch Marvel movies.

me.txt contains the following text:
My favorite food is sushi.
I am a Computer Science major.
It's raining right now.

Notice how like.txt contains repeated words and me.txt contains a lot of unique words. After running my compression algorithm on both files, I get the following results:

```
alexyeh@alexyeh:~/cse13s/asgn6$ ./encode -i like.txt -o middle.txt -v
Compressed file size: 88 bytes
Decompressed file size: 84 bytes
Space saving: -4.76%
alexyeh@alexyeh:~/cse13s/asgn6$ ./encode -i me.txt -o middle.txt -v
Compressed file size: 95 bytes
Decompressed file size: 81 bytes
Space saving: -17.28%
```

As you can see, the compressed file size of like.txt is smaller than the compressed file size of me.txt. This is because like.txt has a lot of repeated words (low entropy), which means that the dictionary will be smaller. me.txt has a lot of unique words (high entropy), which means that the dictionary will be larger.

5 Conclusion

Overall, I enjoyed learning about Lempel-Ziv compression and how it works. I found it interesting that this compression algorithm uses a dictionary to store codes to represent words in order to encode and decode files. Personally, I felt that this assignment was the hardest one we had this quarter. With this assignment being our last in the class, I am glad that I was able to finish the quarter off with a challenge and learn something new and fascinating along the way.