LZ-78 Encoding & Decoding

Alexander Ragland #adraglan

March 11, 2023

1 Buffers & Endian-ness

A large portion of the time I spent debugging during this project was on buffer-related issues. Keeping track of a static buffer between multiple separate functions is more difficult then it seems. Not only did I have to make sure I was incrementing and resetting the index in the correct spots, I also had to do so while accounting for the size of bytes vs. bits. Since there are 8 bits in a byte, you need to make sure you are properly dividing or multiplying by 8 at various points.

In order to achieve operability between big endian and little endian systems, I had to swap the bits of codes and symbols and write them accordingly. Fortunately, we were provided a library for swapping bits, though after some investigation I learned that it's simply a matter of bit shifting and twiddling to reverse the bit order.

2 Reading & Writing

Using system calls to read and right is straightforward, in general. However, using them in tandem with static buffers for the purpose of efficiency is a little more involved. For example, when reading from the buffer, you have to make sure that you clear the buffer first with *memset()* so that the updated buffer doesn't have leftover bytes at the end of the newly-read bytes. I also had to write wrapper functions for *read()* and *write()* because they don't always read/write the correct amount of bytes.

3 File Statistics

In order to set and/or check permissions, I had to use *fstat()* on a given file and store the return value in a *stats* structure. File stats are extremely useful for getting information about a file like size, permissions, block size, device ID, etc.

4 Tries

Using tries in the LZ78 algorithm was very efficient, as well as conceptually logical. Previously, my only experience with trees was binary trees and that was limited at best. I learned how to recursively navigate a tree or trie in order to delete nodes in my implementation of the trie functions.

5 References

- Varun Golusupudi was extremely helpful to me when I was implementing *read_pairs()*.
- Ben Grant was extremely helpful in general debugging, especially with the word and trie functions.