Noel Ball
CSE13S

Writeup Document LZ Compression

Writing my makefile for the encode and decode executables was pretty straightforward following the multiple executable setup I learned during the previous assignment. I use an all_objs variable to include the reference of the executable and library object files, just objs for library object files, and sources for the library c files. The normal CFLAGS were fine, and there were no linked libraries to worry about either.

In this assignment while writing encode.c and decode.c, I warmed up to using file descriptors over file pointers like the previous assignment. I certainly preferred using the typical read, write, open, and close rather than their file pointer counterparts. I also improved on using struct data types and referencing their fields, a lot of the library functions used this as well. I learned about (nil) and its use in struct fields instead of NULL.

In trie.c I really grew to understand the use of tries as a data structure and I have grown more comfortable with the use of pointers and structs to create such a data structure. Compared to a list in python I really got to feel like an engineer.

In io.c a lot of the functions needed the use of global buffers as well as a helper function in one case with read_pair. At first I was creating the global buffer in the main c files rather than with the functions that use it and ended up fixing that. Although I struggled probably the most with the bitwise operations that are used with the pairs. That is probably a factor with my project not working.

Noel Ball
CSE13S

I also found the memory allocation and freeing necessary in this project was a little beyond my capability. I spent way too much time trying to fix valgrind memory leaks I still don't understand. Hence, my final project includes a segfault in decode. Honestly, unlike the previous assignment in which I felt I had pushed myself past my limits, this assignment had just burnt me out. I tried my absolute hardest to fix its issues and I'm still at a loss. I believe with more time to understand the problem, I could complete this project.

I believe I got so close to learning from my mistakes, but due to the rigorous testing system this class has to offer, I guess I'll never know. Unless of course I have to retake this class and this assignment happens to stay the same.