Post-project Writeup - Intersect

James Viner

Project Summary

The task was to create a command-line utility that accepts an arbitrary number of files with an arbitrary number of words and print to the console each word found across all files, in alphabetical order.

Challenges

The funny thing about this course is that I keep looking back to my previous projects for guidance and inspiration on how to solve my current problems, but past me was a much worse programmer than I am now. Getting a general idea of some of the structures or functions I can use from my old work is helpful, it just isn't as helpful as I wish it was. For example, this project would be a perfect time to look back at wordsorter for help on using qsort, but the structure of my ws program is difficult to read through and I ended up having to just wing it with my extremely rudimentary understanding of void pointers and (scary) casting. Trying to get the syntax correct for casting and dereferencing void pointers unironically took the longest amount of my time of anything on this project and I'm not happy about that.

Successes

This project went surprisingly well. Maybe I just needed to hit my lowest with the previous project to really get me into the correct mindset to accomplish all of my goals. The biggest success on this particular project was the design plan stage, where I actually sat down and thought about the overall structure that I wanted my program to have. Having essentially a two-dimensional array of linked lists in order to solve the issue of having duplicate words with or without punctuation and hash collisions at the same time worked wonderfully. Normally I spend the tail-end of a project tracking down a pesky memory leak or struggling with some insanely simple concept, but this time around things just overall went pretty well. Good plan, good execution. I just hope it doesn't segfault when you run it, because I wasn't able to get it to.

Additionally, knowing that we can use scripts to do our unit testing is super convenient, as it gave me the opportunity to do some bash scripting, which I haven't used in ages. It might not be quite as robust as having a separate C file to do it on a per-function basis with libcheck, but I think it did its job, especially because most of the functions in intersect aren't easily testable, while the overall program is.

Lessons Learned

Taking additional time on the design plan paid off, simple as that. Normally I'll rush into a project, start prototyping, and have to backtrack a bit to write a design plan that fits the signatures that I sketched up. This time, ensuring that I had an idea of what implementation I was going for, and what the structure of my underlying data was going to be, was key to my success. Also, taking regular, short breaks to avoid getting burned out or overly bored helps. I'm also going to bed at 2AM today instead of like 6AM, so I'm feeling pretty great.