

Write Up

☰ Tags

NOT STARTED

Project Summaries

The purpose of the program is to take a number of command line arguments that are paths to text files. The program will then read through each file and determine which words are in common between all the files. Once the program is done, it will then printout all the words in alphabetical order. When reading from the file, the words are delimited by `isspace(3)` (`\f`, `\n`, `\r`, `\t`, `\v`). The printed word should match the case from when it was first found.

Challenges

The biggest challenge overall is of course juggling my new born and the project itself. I haven't slept since Friday morning. I am having a pretty misserable time. Along with that, is has been the UTF-8 that I tackled that was pretty challenging but I was able to get through it. The project supports `utf-8` and remembers all versions of the sting found with `-a`.

At first I chose the tree map avl as my data structure and I have come to realize that even though I picked it to save some time because of the baby that it was till a great choice considering how manay characters there are in UTF-8. I think the b-tree would have been implemented fine, but I like the way I implemented it. I made my dictionary keys the collation conversions of the strings to do super fast comparisons. Otherwise every time a character is read and compared it would have to be transformed to the collation version of the string to do the comparison. But because I saved it as the key node, it made the tree super fast.

Success

My major success is my tree of course and I made it opaque enough that I could quickly adapt a `intersection` api in front of it to make it work. I also love using the unit tests, it makes things so much easier when you make changes in your code and realize that an old test failed.

Lessons Learned

Is definitely learning how to use the wide character set AND how to use GDB very well. I think I've gained a ton of experience with GDC on this project