

Post-project Writeup - Wordsorter (ws)

James Viner

Project Summary

The task was to create a command-line program that sorts words from either a number of files or standard input. Various options were allowed to be passed on the command line to modify the program behavior in order to change either the algorithm used for sorting or the format of the output by displaying, for example, in reverse order or without duplicate strings.

Challenges

I believe the biggest problem with this particular project came suddenly and unexpectedly at the very end, towards while I was trying to push up my final few commits and go to bed. I realized that a number of improvements or corrections that I had made to the base code fundamentally broke some element of my program, such as changing out `strcmp()` for the more secure `strncmp()` in a number of places, but inputting the wrong `size_t` for the length, resulting in a bug where the program would store only the first letter of any given string that did not begin after a newline character. It was these supposedly small, free-to-implement improvements that would demonstrate to me how important it is to double-check one's work, and especially before pushing changes to a remote repository.

Successes

I noticed that during this project there were a number of moments where I caught myself writing blocks of code that made logical sense, without compiler errors, that built off of learned lessons from previous projects and from class in general. I suppose what I am trying to say is that I can feel myself improving in terms of my competency with C syntax. While I did run into certain challenges, such as with being forced to write an array re-sizer function after realizing how impractical it would be to try and do operations on an array of strings after setting certain indexes to NULL bytes, most of the project went much smoother than with hangman.

Lessons Learned

Probably the biggest lesson learned with this project was related to the usage of function pointers. For a good portion of my repository's lifespan, the work that could be handled by setting a single function pointer to the

desired sorting algorithm was instead handled by setting multiple different Boolean values explicitly on or off depending on which options were passed. This, of course, is silly, and was eventually replaced by the implementation of having a single function pointer value inside of my options struct that would handle which sort was to be used by overwriting itself with the last value passed as an option on the command line. In general, I would say I'm getting better with pointers, and having a trial-by-fire refresher in the form of `q_sort` validation and function pointers to handle command line options was good for me.