WO1 Clayton E. Williams
C I: tickers
2023.05.24

Project Summary

This project will read in a text file of stock symbols and their company names. From there, users are prompted to enter a symbol they would like to search for, and if it exists, will print the symbol, company name, as well as write it to a text file showing the results.

Challenges

The biggest challenge I faced was still understanding how pointers work and exactly how to implement them. Passing character arrays to functions, and being able to access certain elements is still a learning point for me, but I am making progress.

Additionally, understanding exactly how a program takes input from the command line securely, and stores that to a variable is still a challenge. I have a good grasp on reading stdin, but am unsure of exactly how to delimit number of bytes to be able to do so safely.

Successes

I spent several hours researching arrays and pointers, with a scratch file trying to use and manipulate data based on address of or dereferencing, and by the end, was able to pass the various arrays to functions and manipulate the data as needed.

In the past, I have relied heavily on the built-in debugger in various IDEs, however, with this project, although it would run and compile, the VSCode debugger would throw a segfault in my get_file_size function, that required the use of gdb to actually debug. I had to spend about a day just researching and understanding how to use gdb, but it paid off in the end. Not only was I able to debug the program using it, it also helped me to better understand the previously mentioned pointers and character arrays be seeing what was in each index and exactly why the program was behaving like it was.

Lessons Learned

gdb is a powerful tool and should be used more frequently. I always preferred the GUI of the VSCode debugger, but after learning more of gdb, I realize it displays all the same information, to include each individual element of string. Simply using it more often and spending time understanding how it displays that makes it much easier to use.

There is a fine line between simplicity and over simplifying. I initially tried to make my program extremely modular and granular, but nested functions that rely on arrays passed down multiple levels using pointers, and manipulating those arrays at their addresses to use elsewhere can get very confusing, hard to read, and harder to debug. I backed off and went with a more simple design that, in the end, was quicker and easier to write.