buyer_seller writeup

WO1 Clayton E. Williams

December 2023

0.1 Project Summary

buyer_seller is a project consisting of two programs, buyer and seller. The buyer program is a client that reads a data file of 50000 entries, consisting of an account number and order or payment, and sends it to it seller through the use of Unix Domain Sockets. Acting as a server, seller receives these transactions, and through the use of multi-threading, consolidates data from all 10 clients into a single account structure, avoiding collisions.

0.2 Challenges

The main challenge of this project was trying to incorporate threading in conjunction with Unix Domain Sockets. The client side was easy to design and implement, however, despite writing a few code examples of threaded processes, figuring out the logic of how to spin up a thread after a connection and continue to listen for more connections was more difficult than I had anticipated. Additionally, many resources online were difficult to understand, had 0 regard for memory management and signal handling, or continued to reference them in regards to network sockets, which I also don't have a fundamental knowledge of yet either. Unfortunately, because the course material is sub-par, it meant spending alot of time online trying to figure out how to implement Unix Domain Sockets, which can be dangerous.

0.3 Successes

Signal handling. After the last project, I was not happy with my understanding of signal handling and how it was implemented. I did some extra exploratory learning on the subject, and started re-writing that project around signal handling, rather than implementing it after the fact. I used the gained knowledge of signals, as well as the idea that they must be designed around to do my initial design for this project. I think that I did a much better job of implementing signal handling on this project because of that, and it is a more graceful exit on SIGINT.

0.4 Lessons Learned

The biggest lessons learned are all the specific intricacies that I learned while digging through the man pages trying debug my program or implement certain functions. For example, on the client side, I was trying to figure out how to encode an EOF character to send to the server to indicate the client was done transmitting. However, when

reading through the recv man page, I realized that recv returns a 0 on EOF from client, so the work is done for me, and I can leverage the intricacies of the API itself.

Aside from getting a better understanding of Unix Domain Sockets which will translate to network sockets, I have narrowed down and identified reliable, trustworthy resources for these topics. Again, it is impossible to learn all we need for this course from the text, so having a set of resources I know I can turn to that represent good coding practices will be extremely beneficial leading up to the BSLE.