Name: Andrew Cristancho
ID: 2702278

INTRODUCTION

The main goal is to write an rfc specified server to comply with an rfc specified client. In order to properly execute the specs, research into multi-threading, sockets, and software design are needed. In addition, adequate understanding of the rfc specs is key to implementing a proper server.

PROBLEM

The core problem that an rfc specified server needs to solve is the following: setup a valid connection with a valid client in order to execute valid commands. This is a non trivial matter because many fields of computer science need to be well understood in order to build a correct server. Fields that need to be studied include concurrency, networking, software design, parsing, messaging, etc.

METHODOLOGY

My main tool was my notebook. Since I have inadequate information on sockets and concurrency, I needed to design my project ahead of time in order to combat the problems that will inevitably arise from my inexperience. The first step was to design an architecture, a wireframe if you will, of the entire project. This is essential for complex projects because every addition has the potential to affect the whole system, so it is ideal to know the system overview for debugging.

After the architecture was complete, I immediately went back to analyze the specs and verify the overview. After that, I analyzed the time constraints along with my other conflicting classes in order to create a plan of action. After that, I simply executed what I planned. It wasn't a smooth execution since there were many problems along the way that required many hours of debugging. But since I planned my system ahead of time to combat my inexperience, debugging was usually trivial with hard work.

A common theme throughout the execution of the project is that for every hour of coding, I spent two hours of refactoring. This is essential to me because the probability of burnout is correlated to the quality of the code, so I needed to cleanup every addition I made. If there was anything I learned from this project it's this: always leave time for refactoring. Your mind will thank you later.

RESULTS

In the end, the design I came up with in the overview worked pretty

well and was a nice foundation to build upon. I rewrote the client instead of using the default one because I wanted a perfect harmony between client and server. This was an upfront risk because it made me behind schedule early on, but in the end the risk paid off with clearity of mind.

ANALYSIS

This was a very challenging but rewarding project for me. It was easly my favorite project in my years as a computer science student at FIU. Because of this project, I finally understood the basics of concurrency and threading, which will prove to be a solid addition to my toolbox in my professional career. In addition, learning sockets was mind expanding experience because I've always focused on native programming and never on networking.

My favorite field in computer science is software design and architecture. I find it an interesting mental challenge to design complex software constructs that are valid, modular, and tested. This project tested everything I thought knew and I loved it for that. It made me feel uncomfortable all the time, and that's why I respect this project so much. It easily made be a better programmer, regardless of the grade I get in the end.

Quick feedback for next time. Instead of providing a client, it is my unverified opinion that it would be better to provide students an overview/architecture/design of the project instead. Or maybe not, because part of the excitement was figuring out a design for myself. Other than that, solid assignment.

REFERENCES

Python standard library reference