Julia Sales

ID: jesales

Asgn 3

CSE 130-02

Write up

Testing:

For testing my http server, I would test it on my local machine, my school ssh account, and on Ubuntu. I would test small portions of my code. I would go back and forth between my design doc to either update something or to add the logic for a function if something went wrong or completed a portion. I would write to my design doc first and then update my code.

Problems:

I have warnings in my code. However, logging is not working and my hash table. I have not yet been able to implement the hash table or city hash entirely. Overall, a majority of my programming assignment does not work.

Q: Explain the difference between fully resolving a name (to an http name) when the name is created and the approach that you're taking for this assignment. Give an example of when it might be useful.

A: In this assignment, we resolve names by either looking for the target name or the alias name. Compared to the last assignment, we only referenced the 27 letter ascii target name. However, in this assignment, we can reference multiple names that also refer to the original 27 letter ascii target name. Aliases are useful because one alias can refer to more than one object and it is easier to understand. Aliases are also good for security purposes. For example, if someone has the path to a secret file, they can use an alias to hide that path.

path: dir/dir2/dir3/secret/ ← If someone didn't want to expose they path, they can create an alias for secret.

They can make an alias called "dir4" which will reference secret and not expose the path.

Q: What did you learn about system design from this class? In particular, describe how each of the basic techniques (abstraction, layering, hierarchy, and modularity) helped you by simplifying your design, making it more efficient, or making it easier to design.

A: This class definitely helped me improve my coding skills in a sense of logic, design, and implementation. Even though it was hard, I think it was a good challenge for me and to actually test my capabilities as a programmer. I feel like I now know why it is so important to design your code before you start coding. Personally, I feel like it helped me a lot because it allowed me to plan out my logic more clearly and helped me have a better understanding of the assignment.

Abstraction helped my design because it taught me how to simplify things and make sure I wasn't making my program overly complicated. It helped me reduce the complexity of my programs. Layering helped me organize the design of my programs. Such as breaking it up into steps of what the program needs to do. This class helped me understand hierarchy by creating a large system, and in order to do that, I had to combine a bunch of small systems into a large system (our http server). I understood modularity by breaking up my program into functions and not having all my code in one block. Using modularity helped me make my code less complex. I now understood why we kept building off of the previous programs because it was to help us learn these concepts.