The difference between fully resolving a name when the name is created and the approach I took for this assignment are that in my assignment, my server only checks if the file exists by calling open and making sure it doesn't return -1. When you fully resolve a name, you make sure that the file is unambiguous and it destinguishable from any other file and my server just checks to make sure the file exists so it a client called a get, the server checks if the file exists and not that it is the exact correct file. This idea may be useful is there are multiple files with the same name in multiple directories and the server needed to look through and find the correct file. Since my server only runs on one directory(the one is was executed in) it cannot run into this problem because files cannot have the same name in the same directory.

The main thing I learned about system design in this class was how important the use of design documents are. In assignment 0, the design document was not really needed because the assignment was so simple, so I neglected it in assignment 1. In assignment 1, I had a much harder time trying to debug what I had written up and would have saved so much more time if I had spent more time on making sure my design was good. In assignment 2 and 3, I learned from my mistake in assignment 1 and this allowed me to save A LOT more time. I spent nearly half the time on assignment 2 and 3. I think the most important basic technique introduced to me in this class was modularity. If you can create separate functions and test them individually, you can eliminate things from being the error when trying to debug and this makes is substantially easier. The more lines you have to sift through, the harder it is to debug and adding helper functions to do small tasks in my server made it much easier to debug and understand what my server was doing.