Surya Rani

Design Doc asg2

Whats new?

To implement asg2 we must include multithreading into our server which will allow for our throughput to increase as we are able to handle multiple requests at once. With this we also want to log each of the outputs of each thread in a log file. We also want to have a health check of the server so we will implement a way to check how many requests were sent to the server and how many errors were sent back.

How to implement

Main

- First we must parse through the arguments of the server and see what port, if there should be a log file generated and how many threads to make
- We need to create the main thread and the worker threads which the main thread will assign work to
- We must have each thread accessing areas by itself and not allow others to access at the same time by using mutexes

LogFile

- Each thread will go do its process and send a response to the client
- With each response we will get the data that is sent to or from the client and write it in hex to the logfile using pwrite
- Using pwrite we will be able to write simultaneously and with an offset so that no thread is writing out of order

HealthCheck

- Use this if correctly called after server is running with log file and using get to access this
- Use a counter every single time a thread sends a response to the client
- Use a counter every time a thread sends an error to the client
- Use these numbers and form a string with dprintf to send to the client

ErrHealthCheck

- This function will be used if the request is calling for a healthcheck but is
    - Not using log check as one of the arguments to setup the server
    - Using head to access the healthcheck
    - Using put to access the healthcheck

Making struct to hold all request and be able to pass multiple arguments into the worker threads
Main thread will change the circular array of the worker threads so they can deque off of that
While dequeuing and doing job the worker thread must be in lock
Once lock is over and there is no more data it will go back to waiting for the main thread i assume
I kinda dont know exactly how it works

Might need to brush up tmrw

How does main thread continually get requests and send them over