**Networked Spell Checker**

1. **Introduction:**

This program checks the input words by user then compare it with existed words in a database stored on a server using network programming. Server is accessible through network and user just need to input words and server will send back the result either it’s correct or mispelled.

1. **Program design:**

This program works mainly using threads and network programming. The part network programming has already been provided by professor Fiore on how to setup an echo server. For thread, the most important part is mutexes and pthread functions. After setting up the server (set up listenfd, port, file, bind, etc.) We will initialize the server. In the struct server, I have variables to control number of clients and logs, client and log pointers to control read and write operation, 2 mutexes for client and log to lock or unlock so they can read and write, variables to signal if the client or log is full or empty.

Server wil keep listening to the set up port until a word is sent. Server will now get the words from the file, do the comparison and return the result to print out if the word is correct or mispelled.

Functions like: remove\_log, insert\_log are to remove or insert the word into the log queue for comparison. Remove\_client, insert\_client are to create or remove workers to handle the log queue.

Lookup will take the word and pull words from the dictionary out to compare and return the result. Rewind the pointer to the top for the next word when done.

Worker function is to control the worker queue and uses remove or insert client function to handle the queue. It will lock the worker then check if the queue is is empty or not then will start receiving words and compare. Then unlock the worker mutex. Then send the result to client side. After that, it’ll lock the log mutex to manipulate the log file and it’ll write to the log file the result. And signal that log queue is not empty for the log worker to function. Then unlock the log mutex.

Log worker function is to lock the log mutex and check if the log queue is empty, then it’ll remove the logs 1 by 1. Then signal that the log is not full and unlock the log mutex.

Server init is basically init the server and set all the variables to 0 and calloc the client and log arrays.

1. **Test plan:**

* Try running server without any arguments to see if the variables are set correctly
* Try running server with just port as argument
* Try running server with just filename as argument
* Try running server with all arguments
* Try to connect a client to server to see if possible to connect
* Try typing on client side to see if message is sent
* Check if log file is created
* Check if result returned correctly