Networked Spell Checker

This program offers a server that can receive a word from many clients and spellchecks them.

Using the command line arguments called by the server, the program sets a port and a dictionary file to spell check against. Once the port is inputted, a connection is made across that port between the server and client.

Running on an infinite loop, the server sets up a socket descriptor and begins listening for incoming connections on the port. Once, one is accepted, a new descriptor is made to represent that server-client connection. Written on that descriptor is the word which shall be spell checked. The main thread of the program checks the validity of these connections and appends them to a queue (of words to check). If the queue is full, the main thread waits until a slot is opened.

Meanwhile, there are multiple worker threads with access to the same queue. These threads are signaled by the main thread when a slot has been filled. They pull a file descriptor from the queue, then signal back to the main thread that a slot has been emptied. With that descriptor, they read the word which the client asked to check. They search the dictionary file for that word, and if found, the word was spelled correctly. They attach the status of the check to the word and write it back onto that descriptor (replying to the client). Afterward, the word and status are written to a buffer as long as there is room. Otherwise, the worker thread waits until space is made in the buffer.

Finally, there is a single logger thread which is removing words from the buffer which the worker threads were writing to. Its job is to create a text file and copy the words (and their spelling status) onto that log text file. Once it removes a word, it signals to the worker threads that more space has been made in that buffer.