TW-Mailer Pro

Client and Server Architecture

Usage:
./myserver 6543 <mailspool>
./myclient

With two separate files we create a server and a client. The server starts a connection on the localhost on a port that the user can decide on. The server then checks the users credentials and writes their message into a file inside the spool directory.

Concurrency

We achieve concurrency by forking the process with shared memory. Locks reside in this space of shared memory. There are two locks currently in use: the index_lock, which covers the fetching of the current index and the file_lock which, well, locks file access.

Processing

The client is able to connect to a server if it is reachable and send following requests

- LOGIN
- SEND
- LIST
- READ
- DEL
- QUIT

All these requests are implemented based on their requirements.

In doing so, we have distinguished the processes of sending and receiving messages from the actual logic which gathers and processes the sent info into the results.

Authentication

~~LDAP should have been used as an extended login feature, but due to a lot of failed attempts to implement it and the lack of more time and sanity, it was not implemented in the final version of our TW-Mailer Pro.~~
~~"Authentication" is done by inputting any username and any passphrase that's longer than 4 characters. It still locks you after three tries 'tho.~~
We fixed it. It now authenticates over LDAP.

Used Technologies

For development and testing we are using the Linux subsystem on Windows in Combination with VS-Code for efficient and easy development.

We decided to use C++ as a programming language since we felt more comfortable with it than plain C and have overall more experience with it.

Development Strategies / Needed adaptations

First we implemented the server and client based on the tw-mailer basics requirement. And built upon that. LDAP was the last addition to the project.

No further adaptations are planned by our side.