TW Mailer Description

Client and server architecture

	Client	Server
Usage	./client <ip> <port>	./server <port> <directory name>
Function	● Socket client setup ● Connects to server ● Sends requests and receives the responses	● Socket server setup, bind, listen ● Creates blacklist (.txt file) ● Accepts connections in a loop ● Creates a new **thread** for handling the communication with each new client (concurrent server). **Receives, processes** and send a **response** to each request VALID CLIENT REQUESTS: LOGIN → handleLogin() SEND → handleSend() LIST → handleList() READ → handleRead() DELETE → handleDelete() QUIT ● After the corresponding “handle-” X function is called (commandFunctions.h), responds with either OK or ERR
Headers	clientRequests.h	commandFunctions.h → for handing the client requests helperFunctions.h ldapClient.h → LDAPclient class with one public function (Login authentication) mypw.h → getting password for login

LOGIN takes care of user authentication by creating an instance of the “LDAPclient” class and calling its only public function, authentificateUser(). If the LDAP client can bind the user’s credentials successfully, the login is successful.

Used technologies and libraries

Development:
Visual Studio Code WSL (Ubuntu 20.04), C++ 17

Packages, libraries and dependencies:
OpendLDAP and its recommended libraries and tools (OpenSSL 1.1.1+, libevent, …)
std::filesystem
std::iostream, std::fstream
-

Development strategy and needed protocol adaptions

TW Mailer Basic:
For the development of twmailer basic we had 3 main tasks (n. 1, 2, 5) and two unplanned additional ones (n. 3, 4). For the first, each person worked after the other. The rest of the tasks were mostly done parallelly, except for the 2nd and 3rd, as task n. 2 was dependent on 3.

1. **Basic server/client communication**
2. **Processing of SEND and LIST requests**
3. **Fixing buffer issue (mainly server side changes)**
4. **Fixes in formatting and on the client side**
5. **Processing of READ, DELETE requests**

For a more in-depth explanation, please refer to our previous document: link

TW Mailer Pro:

During the second part of the development process, the amount of time each team member had to wait for the other's work to be done was reduced. The tasks and their order were as follows:

6. **LDAP client** (user authentication)
7. **LOGIN request** (+ changes to makefile)
8. **Blacklist** (persisted) **and limited amount of login attempts**
9. **Concurrent server** (multithreading)
10. **Bug fixes** (memory leaks)
11. **Client IP in blacklist (passing multiple argument to thread)**
12. **User defined IP, port and directoryName** (passing multiple argument to thread)
12. **Bug fixes** (login, directory, blacklist)

Of these, tasks 10, 11 and 12 were unplanned.