

# **TW-Mailer Extended**

**VERTS – BIF WS2021**

if20b089 – Jakob Friedl

if20b161 – Viktor Szűsz

# Client – Server Architektur

## Client:

Ein Client kann sich mit einem Server unter Angabe der IP-Adresse und des Ports verbinden und hat dann eine Auswahl von vorerst 2 Befehlen, mit denen er mit dem Server kommunizieren kann.

- LOGIN schickt mit den angegebenen Anmeldedaten eine Anfrage an den technikum-wien LDAP-Server.
- QUIT beendet das Programm auf Client-Seite

Sind die Credentials fehlerhaft, bzw. existiert der angegebene User nicht am LDAP Server, so schlägt die Authentifizierung fehl und es wird ein Counter für die IP des sich anmeldenden Users erhöht. Erreicht der Counter 3, so wird der IP-Adresse geblacklisted und der User kann sich für eine Minute nicht anmelden. Stimmen die Credentials loggt sich der User ein und kann die folgenden Commands verwenden:

- SEND sendet eine Mail, bestehend aus Sender, Empfänger, Betreff und Nachricht an den Server. Der Sender muss dabei nicht angegeben werden, sondern ist der aktuell eingeloggte LDAP-User
- LIST zeigt dem Client alle Mails des eingeloggten Users an
- READ zeigt dem Client den Inhalt einer bestimmten Mail des eingeloggten Users an
- DEL löscht eine bestimmte Mail des eingeloggten Users

Bei fehlerhaften Eingaben (Username oder Betreff zu lang, Username mit Sonderzeichen) muss der Client die Eingabe wiederholen, solange, bis diese den Anforderungen entspricht. Zusätzlich ist es nun durch Threads möglich, dass sich mehrere Clients gleichzeitig mit dem Server verbinden und mit diesem interagieren. Kritische Stellen wie das Index-File und andere File-Zugriffe wurden mit einem Mutex gelockt.

## Server:

Beim Starten des Servers muss ein Port angegeben werden, auf dem sich Clients anschließend mit dem Server verbinden können. Außerdem bedarf es der Angabe eines sogenannten Mail-Spool-Verzeichnisses. In diesem Verzeichnis wird für jeden Empfänger einer E-Mail ein Verzeichnis angelegt, in dem die einzelnen erhaltenen Mails gespeichert werden. Die Dateien haben dabei eine eindeutige fortlaufende Nummer, sodass auch bei parallelen Zugriffen die von Client gewählte Mail korrekt abgerufen werden kann. Eine neue Mail wird immer mit der nächst höheren Nummer identifiziert. Diese Nummer wird immer in einem Index-File im Ordner des Users gespeichert, sodass leicht ablesbar ist, wie die neue Mail heißen soll. Das Programm wurde seit dem Basic-Beispiel demensprechend erweitert, sodass das angegebene Mail-Spool Verzeichnis immer in einem bestimmten Ordner angelegt wird.

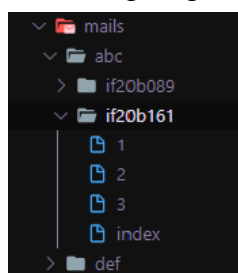


Abbildung 1 Mail-Spools (abc & def) im mails-Ordner mit den einzelnen Usern

Auf beiden Instanzen läuft eine Endlosschleife in der Befehle ausgetauscht werden. Diese Endlosschleife kann am Server nur über ein Tastenkürzel (z.B. CTRL+C) abgebrochen werden, auf dem Client auch mit dem QUIT-Befehl. Sendet nun der Client einen Befehl, so wird je nach Befehl am Server anders reagiert.

Um Mails versenden, lesen oder löschen zu können, muss sich der User zuerst anmelden. Dies geschieht über den LOGIN Befehl und der Angabe von Username und Passwort auf Clientseite. Natürlich wird das Passwort nicht offen auf der Konsole ausgegeben sondern versteckt. Daraufhin stellt der Server eine Verbindung mit dem internen LDAP-Server her und überprüft die Korrektheit der Anmeldedaten. Es existiert weiters ein Blacklist-Verzeichnis, in dem jede IP-Adresse, die sich mit der Server verbunden hat einen Ordner hat. Es gibt darin zwei Files: Counter und Timestamp. Im Counter-File wird mitgezählt, wie oft in Folge eine IP-Adresse falsche Credentials angibt. Erreicht diese Zahl 3, so wird im Timestamp-File die aktuelle UNIX-Zeit eingetragen, und der User kann sich erst wieder verbinden, wenn eine Minute vergangen ist.

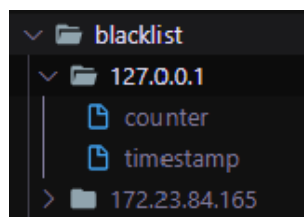


Abbildung 2 Blacklist

Nach dem SEND-Stichwort wird serverseitig auf die zu sendende Email gewartet. Für gewöhnlich antwortet der Server dann mit einem „OK“, das vom Client empfangen und angezeigt wird. In Fehlerfällen, wenn beispielsweise ein Benutzername angegeben wurde der nicht existiert, sendet der Server „ERR“ und es folgt ein Reset der Kommunikation, sprich der Client kann dann wieder einen der 5 Befehle auswählen. Beim LIST-Befehl öffnet der Server das User-Verzeichnis des aktuellen Users und sendet die Anzahl der Mails, sowie die Mail-ID und den Betreff der einzelnen Mails an den Client zurück. Der Client kann dann mit READ bzw. DEL unter Angabe der gewünschten Mail-ID den Inhalt einer Mail auslesen oder diese Löschen. QUIT bricht die Verbindung mit dem aktuellen Client ab.

## Technologien

- Visual Studio Code mit WSL (Ubuntu-20.04)
- C Programming Language
- LDAP Packages: libldap2-dev, ldap-utils

## Development Strategie

Ausgangspunkt des Programms war die Struktur des Client-Server-Beispiels. Es wurde dann versucht die Befehle schrittweise in folgender Reihenfolge zu implementieren:

1. QUIT
2. SEND
3. LIST
4. READ
5. DEL

Dies war von Vorteil, da nach SEND alle aufbauenden Befehle leicht testbar waren. Die Implementierung des SEND Befehls war jedoch im Vergleich zu allen anderen am Zeitaufwändigsten, da es Probleme mit dem Lesen und Schreiben auf Socket-Deskriptoren gab. Gelöst wurde dieses Problem durch Trial-and-Error, sowie gründliche Recherche. Zum Lesen

und Schreiben auf Sockets wurde die im „TCP/IP Programming“-PDF vorgestellten Funktionen „readline“ und „writen“ verwendet.

Im Rahmen des TW-Mailer Extended Beispiels wurde zunächst der bestehende Mail-Server mit der C pthread-library concurrent gemacht und generell verbessert. Das Öffnen, Schreiben und Löschen von Files wird mittels Mutexes gelockt, um gleichzeitigen Zugriff zu verhindern. Danach wurde der LOGIN-Befehl dermaßen implementiert, dass Daten vom Client zum Server gesendet werden können. Darauf folgte dann das Ausblenden des Passworts bei der Eingabe und im weiteren Sinne die LDAP-Authentifizierung. Dabei wurde die Struktur des LDAPSAMPLEINC-Beispiels nachgebaut. Nach dem Fertigstellen des LOGIN-Befehls wurden alle anderen Befehle so umgeändert, dass der angemeldete User nur seine eigenen Mails lesen und löschen kann. Dafür wurde der Username in einen string gespeichert, der an die einzelnen Befehle mitgegeben wird. Für die Blacklist wird dem LOGIN-Befehl die IP des Clients mitgegeben, damit diese gegebenenfalls gelockt werden kann.

Zuletzt wurden noch Fehler behoben, wie zum Beispiel, dass das Abbrechen (CTRL + C) während dem Schreiben der Mail-Message den Server in eine Endlosschleife versetzt. Außerdem wurde versucht, so viele Teile des Codes wie möglich in andere Files auszulagern, um die Lesbarkeit zu verbessern.

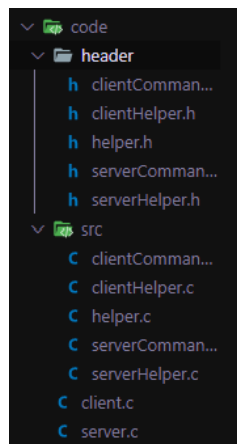


Abbildung 3 Projektstruktur