

# **TW-Mailer Basic**

**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 5 Befehlen, mit denen er mit dem Server kommunizieren kann.

- SEND sendet eine Mail, bestehend aus Sender, Empfänger, Betreff und Nachricht an den Server
- LIST zeigt dem Client alle Mails eines bestimmten Users an
- READ zeigt dem Client den Inhalt einer bestimmten Mail eines Users an
- DEL löscht eine bestimmte Mail eines Users
- QUIT beendet das Programm auf Client-Seite

## 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 einen eindeutigen Namen (UUID) und eine 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.

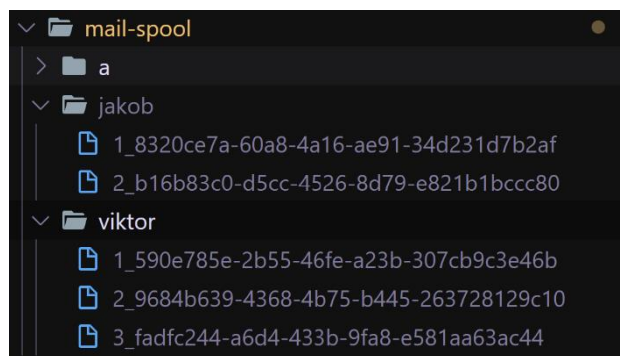


Abbildung 1 Aufbau des Mail-Spool-Directories

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. So wird nach dem SEND-Stichwort 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 ungültiger Benutzername angegeben wurde, 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 angegebene User-Verzeichnis 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 des Usernamens und 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
- Externe UUID Library: <http://e2fsprogs.sourceforge.net/>

# 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 Schreiben auf Sockets wurde die im „TCP/IP Programming“-PDF vorgestellte „writen“-Funktion verwendet. Zum Erhalten und Lesen der Daten die „recv“-Funktion der C socket-Library.