

Projektdokumentation: TW Mailer Pro

I. Client-Server-Architektur

Die Anwendung wurde basierend auf der Basic-Version grundlegend erweitert. Sie besteht weiterhin aus einem Server und einem Client, die Architektur des Servers wurde jedoch von **iterativ** auf gleichzeitig (**konkurrent**) umgestellt.

- **Server Start:** `./twmailer-server <port> <mail-spool-directoryname>`
- **Client Start:** `./twmailer-client <ip> <port>`

Verbindungsauftbau & Concurrency: Der TW Mailer-Server lauscht (`socket`, `bind`, `listen`) auf einem Port. Anstatt Clients nacheinander abzuarbeiten, erzeugt der Server nun bei jedem `accept()` mittels `fork()` einen Kindprozess.

- Der **Elternprozess** schließt die Verbindung sofort wieder und steht für neue Clients bereit.
- Der Kindprozess übernimmt die Kommunikation und hält die Verbindung offen, bis der Client `QUIT` sendet. Zombie-Prozesse werden durch Ignorieren des Signals `SIGCHLD` verhindert.

Session-Management: Der Server arbeitet nun zustandsbehaftet (Stateful). Nach dem Verbindungsauftbau muss sich der Client authentifizieren (`LOGIN`). Der Server speichert den Benutzernamen im Kontext des Kindprozesses. Nachfolgende Befehle (`SEND`, `LIST`, etc.) benötigen keine manuelle Eingabe des Benutzers mehr, sondern greifen auf diese Session-Daten zu.

II. Verwendete Technologien

Sprache: C (std=c99-gnu99)

Build: make & gcc (mit Flags `-lldap -llber` für LDAP)

Netzwerk: Berkeley Sockets API (TCP/IP Stream Sockets)

System & Concurrency: POSIX Standard (`unistd.h` für `fork`, `sys/wait.h`, `signal.h`)

Authentifizierung: OpenLDAP C-API (`ldap.h`) zur Überprüfung von Credentials gegen den FH-Server (`ldap.technikum-wien.at`).

Persistenz: Standard File-I/O (`mkdir`, `opendir`, `fopen`, `remove`, etc.), ebenfalls für Blacklist/IP.

III. Entwicklungsstrategie und Anpassungen

Die Entwicklung erfolgte in zwei Phasen. Phase 1 behob initiale Probleme der Basic-Version, Phase 2 implementierte die Pro-Features.

Phase 1: Stabilisierung (aus Basic-Version)

1. **Puffer-Desynchronisierung:** Eingabepuffer wurden auf LEN + 2 vergrößert, um \n Reste im Stream sicher abzufangen.
2. **Nachrichten-Sortierung:** Da das Dateisystem (readdir) unsortiert liefert, wurde eine Sortierlogik (get_sorted_messages) implementiert, um konsistente IDs für READ/DEL zu gewährleisten.
3. **Validierungs-Sync:** SEND liest nun immer den kompletten Body bis zum .\n, auch wenn der Empfänger ungültig ist, um das Protokoll nicht zu brechen.

Phase 2: Protokoll-Anpassungen für Pro

Das Protokoll wurde gemäß der Spezifikation "TW-Mailer Pro" gehärtet:

1. **Authentifizierung:** Einführung des Befehls LOGIN <user> <pass>. Nur authentifizierte User dürfen Befehle ausführen.
2. **Protokoll-Vereinfachung:** Entfernung der manuellen User-Parameter in SEND, LIST, READ und DEL. Der Server ermittelt den Sender/Nutzer nun automatisch aus der Session.
3. **Code-Restrukturierung:** Auslagerung von Protokoll-Konstanten (Buffer-Größen, Befehls-Strings) in eine gemeinsame common.h, um Inkonsistenzen zwischen Client und Server zu eliminieren (Reaktion auf Feedback).