TW-MAILER-EXTENDED PROTOKOLL

CLIENT ARCHITEKTUR

ÜBERSICHT FUNKTIONEN

```
char *loginUser();
void sendMessage(int create_socket, char *username);
void listMessage(int create_socket);
void readMessage(int create_socket);
void delMessage(int create_socket);
void getLines(int maxlen, char *prompt, char *target);
void getInput(int maxlen, char *prompt, char *target);
```

FUNKTIONSWEISE

Der Client kann sich mittels Localhost und Port zum Server verbinden, durch die eingebaute `fork()`-Funktion können sich mehrere Clients gleichzeitig zum Server verbinden. Mit dem Server verbunden muss sich der Client mit den richtigen Zugangsdaten (FH Technikum Logindaten) einloggen. Clients haben drei Versuche, bis sie auf die Blacklist kommen und deren Zugriff für eine Minute gesperrt wird. Loggt sich der/die User/in richtig an bekommt er/sie Zugriff auf alle Befehle (SEND, LIST, READ, DEL) und kann diese ausführen. Ruft der Client eines der Methoden auf, werden die notwendigen Parameter vom Client abgefragt und an den Server zur Bearbeitung geschickt.

ALLE BEFEHLE

LOGIN: Fragt den Client nach dem Usernamen und dem Passwort, diese werden dann über LDAP überprüft.
SEND: Speichert die geschickte Message im Sender und Receiver Ordner, falls es noch keinen Ordner gibt, wird ein neuer Ordner erstellt. Beim Sender wird dabei der Username verwendet, der anfangs vom Client zum Einloggen verwendet wird, genommen.
LIST: Listet alle Nachrichten und die Nachrichtenanzahl eines Users auf.
READ: Ermöglicht das Lesen einer bestimmten Nachricht.
DEL: Löscht eine bestimmte Nachricht.
QUIT: Beendet die Verbindung zum Server.

SERVER ARCHITEKTUR

ÜBERSICHT FUNKTIONEN

```
void *clientCommunication(void *data);
void signalHandler(int sig);

void mailHandler(int *current_socket, char buffer[]);
void loginUser(int *current_socket, char buffer[]);
void sendMessage(int *current_socket, char buffer[]);
void listMessage(int *current_socket, char buffer[]);
void readMessage(int *current_socket, char buffer[]);
void delMessage(int *current_socket, char buffer[]);

void updateBlackList();
int checkBlackList(char *userIP);
void blackListUser(char *userIP);
```

FUNKTIONSWEISE

Der Server bekommt die Anfragen vom Client und gibt die entsprechenden Informationen an den Client zurück.

Zuerst wird die Verbindung erstellt, dabei können auch mehrere Clients gleichzeitig mit dem Server verbunden sein. Dies wird durch die `fork()`-Funktion möglich gemacht.

Verbindet sich ein Client zum Server, muss sich der User einloggen. Die Authentifizierung erfolgt mit dem LDAP Server der FH Technikum. Zuerst wird eine Verbindung vom Server zum LDAP Server aufgebaut. Danach wird der User nach den Anmeldedaten gefragt und diese im Nachhinein dem LDAP Server geschickt. Der Client hat drei Login-Versuche, wird dies überschritten wird die IP-Adresse des Clients in die Blacklist hinzugefügt und nach einer Minute wieder gelöscht.

Sobald der Client mit den richtigen Daten eingeloggt ist, stehen ihm die restlichen Menü-Optionen (SEND, LIST, READ, DEL) zusätzlich zu den ersten beiden (LOGIN, QUIT) zur Verfügung und der Server kann die Anfragen vom Client bearbeiten. Dafür kriegt der Server die Parameter vom Client und schaut in der

```
void mailHandler(int *current_socket, char buffer[]);
```

-Funktion nach welche Methode aufgerufen werden soll.

VERWENDETE TECHNOLOGIEN

Der TW-Mailer-Extended wurde in einem WSL programmiert und kompiliert, da Linux spezifische Bibliotheken und Befehle zur Umsetzung benötigt wurden. Für dieses Projekt wurde die Kali Linux Umgebung benützt.

EINGEBUNDENE BIBLIOTHEKEN

Server	Client

`#include <sys/types.h>`
`#include <sys/stat.h>`
`#include <sys/socket.h>`
`#include <netinet/in.h>`
`#include <arpa/inet.h>`
`#include <unistd.h>`
`#include <stdlib.h>`
`#include <stdio.h>`
`#include <string.h>`
`#include <signal.h>`
`#include <dirent.h>`
`#include <fcntl.h>`
`#include <ldap.h>`
`#include <time.h>`
`#include <sys/wait.h>`
	`#include <sys/types.h>`
	`#include <sys/socket.h>`
	`#include <netinet/in.h>`
	`#include <arpa/inet.h>`
	`#include <unistd.h>`
	`#include <stdlib.h>`
	`#include <stdio.h>`
	`#include <string.h>`