Marashi Benard

TWMAILER EXTENDED

Client and Server Architecture

Mein Server ist von der Architektur her dem des Samples fast ident.

Der Server läuft auf localhost. SO_ REUSEADRR and SO_REUSEPORT sind beide aktiviert.

Der Client verbindet sich über TCP.

Der BUF ist auf 10240 Bytes gesetzt.

Die signalHandler-Funktion prüft auf abgebrochene Verbindungen.

Die Kommunikation verläuft über einen char-Array buffer mit der Größe BUF. Alle Daten werden im Format VAL1;VAL2;VAL3;…;VALN übergeben wobei die VALs für konkrete Daten stehen.

Bei Nachrichten vom Client an den Server ist VAL1 immer der Command und alle nachfolgenden VALs die Argumente.

Der Server kann ein einzelnes VAL ohne ; zurückschicken (OK oder ERR) oder mehrere VALs mit den jeweiligen konkreten Daten.

Forks, damit der Server concurrent wird

Login über LDAP.

Nach 3 falsche-Versuche wird der Client auf dem Blacklist (1 .txt File) geschrieben. Nach 60 Sekunden wieder gelöscht.

Used Technologies

Für die TCP Kommunikation habe ich, wie in der Angabe verlangt, die C Socket API verwendet.

Für die Übersetzung von int in Networkbyte data habe ich htons von arpa/inet.h verwendet.

Um meine Socket-Adresse und Port festzulegen habe ich die sockaddr_in von netinet/in.h verwendet.

Um auf die jeweils benötigten Ordner zuzugreifen habe ich dirent.h verwendet.

signal.h wurde für die Überprüfung, ob das Signal unterbrochen wurde, verwendet.

Für das Login wird der LDAP-Server von FH Technikum verwendet.

Development Strategy

Anfangs wollten wir einen string-Array and den Server übermitteln, jedoch stellte sich schnell heraus, dass dies eher problematisch sein würde. Deswegen habe ich stattdessen Daten im zuvor erwähnten Format in den buffer char-Array gespeichert. Nachdem die neue SEND Funktion an den Server übergeben werden konnte, habe ich die serverseitige Implementation geschrieben. Alle nachfolgenden Funktionen verliefen über dasselbe Prinzip.

Nachdem alle Commands einwandfrei gelaufen sind, habe ich angefangen uns auf Fehlermeldungen und mögliche Benutzerfehler zu konzentrieren. Diese habe ich, soweit uns bewusst ist, alle abgefangen.

Needed Adaptations

In der myserver.c Datei des Sample Servers hatte die Client Kommunikation nur 1 Argument, den void* data. Zudem mussten wir ein char* dirPath hinzufügen, um im richtigen Directory zu arbeiten.

Um effizient mit dem Buffer arbeiten zu können, mussten wir diesen in ein string-Array und string-

Vektoren nach der Übertragung einspeichern, wobei jeder ; ein Symbol für das Ende eines Strings war.

Um den LDAP-Server zu testen, mussten wir auch mit dem VPN des Technikums verbinden.