Vysoké učení technické v Brně
Fakulta informačních technologií

IMAP klient

18. novembra 2024 Christian Saloň (xsalon02)

Obsah

1 Úvod do problematiky

1.1 IMAP

Protokol IMAP[1] umožňuje klientovi manipulovať elektronickou poštou uloženou na serveri. Protokol IMAP zahŕňa operácie na vytváranie, mazanie a premenovanie schránok a na manipuláciu elektronickej pošty.

Protokol predpokladá spojenie prostredníctvom transportného protokolu TCP[2]. Ak je použitý protokol TCP, tak IMAP server počúva na porte 143.

Spojenie IMAP pozostáva z ustanovenia spojenia, uvítania od servera a zo interakcie medzi klientom a serverom. Tieto interakcie pozostávajú z príkazu, dát a odpovede zo servera. Všetky interakcie sú vo forme reťazcov, ktoré končia znakmi `CRLF`.

Klientský príkaz je prefixovaný tagom. Tag je krátky alfanumerický reťazec. Každý klientský príkaz začína iným tagom.

Odpovede poslané zo servera začínajúce `*` sa nazývajú untagged odpovede. Ak táto odpoveď bola poslaná po klientskom príkaze, tak to znamená že odpoveď nie je celá a bude odoslaná ďalšia odpoveď. Ak untagged odpoveď bola odoslaná samovoľne, tak tieto odpovede oznamujú zmenu stavu na serveri.

1.2 Internet Message Format

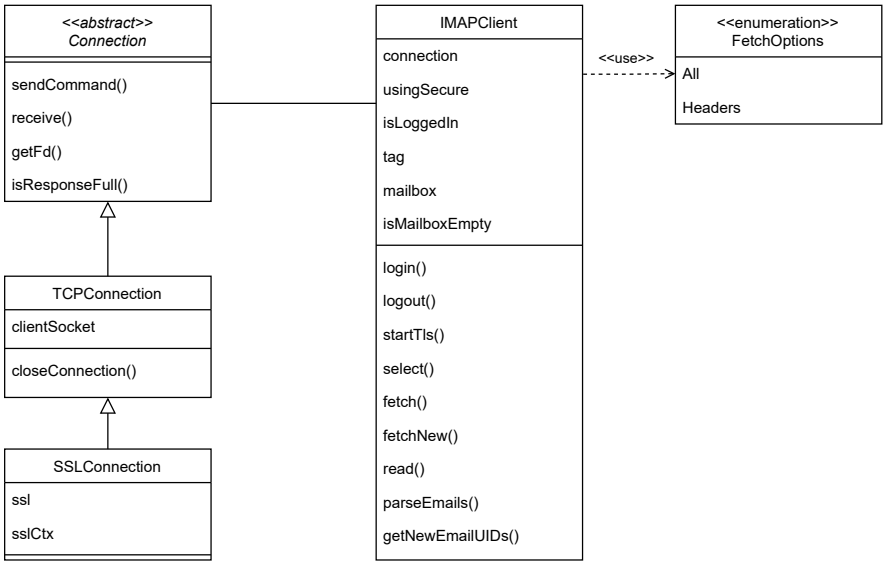
Štandard Internet Message Format[4] špecifikuje syntax pre textové správy v rámci elektronickej pošty. Tento štandard špecifikuje syntax iba pre textové správy a nie pre obrázky, audio alebo iné štrukturované dáta v rámci elektronickej pošty. Existujú rôzne rozšírenia, ako napríklad MIME[3], ktorý popisuje mechanizmus na posielanie takýchto dát.

1.3 SSL/TLS

TLS a SSL[5] sú kryptografické protokoly, ktoré slúžia na bezpečný prenos dát prostredníctvom siete. Protokol TLS je nástupcom protokolu SSL. Bezpečné TLS spojenie sa vytvorí pomocou TLS handshaku. Najprv klient pošle požiadavok o bezpečné spojenie, kde pošle zoznam podporovaných algoritmov. Následne server vyberie algoritmy, ktoré budú využívané počas komunikácie. Taktiež server pošle klientovi certifikát, ktorý klient si môže overiť. Následne sa vytvorí kľúč, ktorým sa šifrujú dáta, ktoré sú posielané v rámci spojenia. Takto je vytvorené bezpečné spojenie medzi klientom a serverom.

2 Implementácia aplikácie

Aplikácia vyžaduje štandard `c++20` a je závisla na knižnici `openssl`.



Obr. 1: Diagram tried

2.1 Trieda `TCPConnection`

Táto trieda slúži na vytvorenie TCP spojenia medzi klientom a serverom.

V konštruktore sa vytvorí socket na komunikácia medzi klientom a serverom.

Metóda `closeConnection()` ukončí spojenie. Unkončiť spojenie nie je možné v deštruktore, a to preto, lebo po zadaní príkazu `STARTTLS` je potrebné spojenie zabezpečiť a nie ukončiť.

Metóda `sendCommand()` pošle klientský IMAP príkaz na server. V tejto metóde sa čaká na odpoveď zo servera a to až kým nie je odpoveď celá. Odpoveď je celá, ak na začiatku posledného riadku odpovede je tag, ktorý bol použitý pri klientskom príkaze.

Metóda `receive()` príjme odpoveď na klientský príkaz. Príjmanie sa skončí, ak odpoveď končí `CRLF`.

2.2 Trieda `SSLConnection`

Táto trieda slúži na vytvorenie zabezpečeného spojenia medzi klientom a serverom.

Táto trieda disponuje dvomi konštruktormi. Prvý slúži na vytvorenie nového spojenie. Druhý slúži na prevod nešifrovaného spojenia na šifrované po zadaní príkazu `STARTTLS`.

Deštruktor ukončí spojenie medzi klientom a serverom.

Metóda `sendCommand()` pošle klientský IMAP príkaz na server. V tejto metóde sa čaká na odpoveď zo servera a to až kým nie je odpoveď celá. Odpoveď je celá, ak na začiatku posledného riadku odpovede je tag, ktorý bol použitý pri klientskom príkaze.

Metóda `receive()` príjme odpoveď na klientský príkaz. Príjmanie sa skončí, ak odpoveď končí `CRLF`.

2.3 Trieda `IMAPClient`

Táto trieda slúži na komunikáciu medzi klientom a IMAP serverom.

V konštruktore sa vytvorí spojenie medzi klientom a serverom. Následne sa príjme úvodná správa od servera.

V deštruktore sa pošle klientský príkaz `LOGOUT` ak je užívateľ prihlásený. Následne sa ukončí spojenie medzi klientom a serverom

Metóda `login()` pošle klientský príkaz `LOGIN` serveru. Parametre metódy sú prihlasovacie meno a heslo, ktoré sa pošlú príkazom `LOGIN`. Následne sa čaká na odpoveď a zistí sa, či prihlásenie prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu `"použitý_tag OK"`.

Metóda `logout()` pošle klientský príkaz `LOGOUT` serveru, ale iba v prípade, že užívateľ je prihlásený. Následne sa čaká na odpoveď a zistí sa, či odhlásenie prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu `"použitý_tag OK"`.

Metóda `startTls()` pošle klientský príkaz `STARTTLS` serveru, ale iba v prípade, že používané spojenie nie je šifrované. Následne sa čaká na odpoveď a zistí sa, či to prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu `"použitý_tag OK"`. Teraz nastáva fáza dohadovania parametrov šifrovaného spojenia, a to tak, že sa vytvorí nová inštancia triedy `SSLConnection`, ktorá toto zabezpečuje. Po dohode medzi klientom a serverom je možné pokračovať v komunikácií.

Metóda `select()` s parametrom názvu schránky vyberie túto schránku, ktorá bude aktívna, a to tak, že pošle klientský príkaz `SELECT` serveru. Následne sa čaká na odpoveď a zistí sa, či to prebehlo úspešne, a to tak, že

hľadáme v odpovedi sekvenciu "použitý_tag OK".

Metóda fetch() s parametrom či získať celý email alebo len hlavičky získa všetky emaily v aktívnej schránke, a to tak, že pošle klientský príkaz FETCH serveru. Následne sa čaká na odpoveď a zistí sa, či to prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu "použitý_tag OK". Metóda parseEmails() vráti emaily vo forme unordered_map, kde kľúč je vygenerovaný názov emailu a hodnota je obsah emailu. Názov emailu je vo formáte server_schránka_UID.eml.

Metóda fetchNew() funguje rovanko ako metóda fetch() len s tým rozdielom, že získa iba nové emaily a nie všetky.

Metóda getNewEmailUIDs() získa UID nových emailov, a to tak, že pošle klientský príkaz SEARCH NEW serveru. Príznak NEW vyhľadá správy, kotré majú príznak RECENT ale nemajú príznak SEEN. Následne sa čaká na odpoveď a zistí sa, či to prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu "použitý_tag OK".

Metóda read() označí nové správy aktívnej schránky ako prečítane, a to tak, že pošle klientský príkaz STORE selected_email_uids +FLAGS (\SEEN) serveru. Následne sa čaká na odpoveď a zistí sa, či to prebehlo úspešne, a to tak, že hľadáme v odpovedi sekvenciu "použitý_tag OK".

2.4 Súbor main.cpp

Vo funkcii main() sa spracujú argumenty programu a vytvorí sa inštancia triedy IMAPClient. Ak nie je použitý interaktívny režim, tak sa zavolajú metódy login(), select() a fetch() alebo fetchNew() inštancie triedy IMAPClient. Ak je použitý interaktívny režim, tak na základe vstupu užívateľa sa zavolá daná metóda inštancie triedy IMAPClient.

Ak je volaná metóda fetch() inštancie triedy IMAPClient, tak sa pred uložením emailov vymažú všetky emaily daného servera a schránky. To zebezpečí synchronizáciu medzi klientom a serverom.

Ak je volaná metóda fetchNew() inštancie triedy IMAPClient, tak sa pred uložením emailov nevymažú všetky emaily daného servera a schránky. Ak by sa vymazali, tak by vo výstupnom adresári boli iba nové emaily, a to nie je očakávané chovanie.

3 Informácie o programe

Program `imapcl`, ktorý umožnuje čítanie elektronickej pošty pomocou protokolu IMAP. Program po spustení stiahne správy uložené na serveri a uloží ich do zadaného adresára. Na štandardný výstup vypíše počet stiahnutých správ. Pomocou dodatočných parametrov je možné funkcionalitu meniť. V interaktívnom režime sa program pripojí k schránke a bude bežať až do ukončenia príkazom QUIT. Používateľ má možnosť sťahovať správy príkazom DOWNLOAD(NEW|ALL) [MAILDIR], čítať nové správy príkazom READNEW [MAILDIR].

3.1 Rozšírenia

Implementovaný interaktívny režim s podporou STARTTLS. Do interaktívneho režimu bol pridaný príkaz STARTTLS a príkaz LOGIN, ktorý autentizuje užívateľa s údajmi poskytnutých v autentizačnom súbore.

3.2 Obmedzenia

Program neumožnuje informovať užívateľa o prijatí novej správy. Taktiež na začiatku v interaktívnom režime nevypisuje počet správ v schránkach.

3.3 Príklad spustenia

Program sa preloží príkazom `make` a spustí sa príkazom `./imapcl server [-p port] [-T [-c certfile] [-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir [-i]`

4 Testovanie

Projekt bol preložený a testovaný lokálne v prostredí WSL na platforme Windows 11 aj na serveri merlin. Testovanie klienta bolo voči lokálnemu Dovecot serveru, serveru eva a serveru pobox.sk.

4.1 Testované prípady

Nevalidné argumenty

Očakávaný Výstup: Pomocná hláška na stderr a návratový kód != 0
Výstup: Pomocná hláška na stderr a návratový kód 1

Nesprávny formát autentikačného súboru

Očakávaný Výstup: Chybová hláška na stderr a návratový kód != 0
Výstup: Chybová hláška na stderr a návratový kód 1

Nesprávne autentikačné údaje

Očakávaný Výstup: Chybová hláška na stderr a návratový kód != 0
Výstup: Chybová hláška na stderr a návratový kód 1

Nešifrované stiahnutie emailov

Očakávaný Výstup: Stiahnutie emailov zo schránky, hláška na stdout a návratový kód 0
Výstup: Stiahnutie emailov zo schránky, hláška na stdout a návratový kód 0

Šifrované stiahnutie emailov

Očakávaný Výstup: Stiahnutie emailov zo schránky, hláška na stdout a návratový kód 0
Výstup: Stiahnutie emailov zo schránky, hláška na stdout a návratový kód 0

Stiahnutie emailov bez zadania schránky

Očakávaný Výstup: Stiahnutie emailov zo schránky INBOX, hláška na stdout a návratový kód 0
Výstup: Stiahnutie emailov zo schránky INBOX, hláška na stdout a návratový kód 0

Stiahnutie emailov zo schránky Important

Očakávaný Výstup: Stiahnutie emailov zo schránky Important, hláška na stdout a návratový kód 0
Výstup: Stiahnutie emailov zo schránky Important, hláška na stdout a návratový kód 0

Stiahnutie nových emailov

Očakávaný Výstup: Stiahnutie nových emailov, hláška na stdout a návratový kód 0
Výstup: Stiahnutie nových emailov, hláška na stdout a návratový kód 0

Stiahnutie hlavičok emailov

Očakávaný Výstup: Stiahnutie hlavičok emailov, hláška na stdout a návratový kód 0
Výstup: Stiahnutie hlavičok emailov, hláška na stdout a návratový kód 0

Príkaz `LOGIN` v interaktívnom režime

Očakávaný Výstup: Hláška na stdout
Výstup: Hláška na stdout

Príkaz `DOWNLOADALL` v interaktívnom režime

Očakávaný Výstup: Stiahnutie emailov zo schránky INBOX, hláška na stdout
Výstup: Stiahnutie emailov zo schránky INBOX, hláška na stdout

Príkaz `DOWNLOADALL Important` v interaktívnom režime

Očakávaný Výstup: Stiahnutie emailov zo schránky Important, hláška na stdout
Výstup: Stiahnutie emailov zo schránky Important, hláška na stdout

Príkaz `DOWNLOADNEW` v interaktívnom režime

Očakávaný Výstup: Stiahnutie nových emailov zo schránky INBOX, hláška na stdout
Výstup: Stiahnutie nových emailov zo schránky INBOX, hláška na stdout

Príkaz `DOWNLOADNEW Important` v interaktívnom režime

Očakávaný Výstup: Stiahnutie nových emailov zo schránky Important, hláška na stdout
Výstup: Stiahnutie nových emailov zo schránky Important, hláška na stdout

Príkaz `READNEW` v interaktívnom režime

Očakávaný Výstup: Označenie nových emailov zo schránky INBOX ako prečítané, hláška na stdout
Výstup: Označenie nových emailov zo schránky INBOX ako prečítané, hláška na stdout

Príkaz `READNEW Important` v interaktívnom režime

Očakávaný Výstup: Označenie nových emailov zo schránky Important ako prečítané, hláška na stdout
Výstup: Označenie nových emailov zo schránky Important ako prečítané, hláška na stdout

Príkaz `QUIT` v interaktívnom režime

Očakávaný Výstup: Ukončenie programu s návratovým kódom 0
Výstup: Ukončenie programu s návratovým kódom 0

Príkaz `STARTTLS` v interaktívnom režime

Očakávaný Výstup: Prechod z nešifrovanej komunikácie na šifrovanú
Výstup: Prechod z nešifrovanej komunikácie na šifrovanú

Literatúra

[1] Crispin, M.: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. RFC 3501, Marec 2003, doi:10.17487/RFC3501.
URL https://www.rfc-editor.org/info/rfc3501

[2] Eddy, W.: Transmission Control Protocol (TCP). RFC 9293, August 2022, doi:10.17487/RFC9293.
URL https://www.rfc-editor.org/info/rfc9293

[3] Freed, N.; Borenstein, D. N. S.: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996, doi:10.17487/RFC2045.
URL https://www.rfc-editor.org/info/rfc2045

[4] Resnick, P.: Internet Message Format. RFC 2822, Apríl 2001, doi:10.17487/RFC2822.
URL https://www.rfc-editor.org/info/rfc2822

[5] Wikipedia contributors: Transport Layer Security — Wikipedia, The Free Encyclopedia. 2024, [Online; videné 17. novembera 2024].
URL https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1257548947