Projekt - IMAP Klient s podporou TLS

Filip Botlo (xbotlo01)

18. November 2024

Obsah

1 Zadanie **2**

2 Protokol IMAP **2**
2.1 Základné charakteristiky IMAP 2
2.2 Fungovanie IMAP 2

3 Implementácia **3**
3.1 Struktúra programu 3
3.2 Spracovanie argumentov príkazového riadka 3
3.3 Pripojenie k IMAP serveru 4
3.3.1 Pripojenie bez šifrovania 4
3.3.2 Pripojenie so šifrovaním 4
3.4 Autentifikácia používateľa 4
3.4.1 Posielanie príkazov 5
3.5 Sťahovanie správ 5
3.6 Interaktívny režim 5
3.7 Správa chýb 5
3.8 Odhlásenie a uvoľnenie zdrojov 5
3.9 Hlavná funkcia programu 5

4 Preklad a spustenie **6**
4.1 Požiadavky na preklad a spustenie 6
4.2 Príklady vstupov a výstupov 6

5 Testovanie **6**

Literatúra **7**

1 Zadanie

Cieľom projektu bolo navrhnúť a implementovať IMAP klienta s podporou TLS, ktorý umožní pripojenie k IMAP serveru, autentifikáciu používateľa, získanie zoznamu e-mailových správ a ich spracovanie na základe zadaných parametrov. Program mal podporovať rôzne režimy vrátane interaktívneho režimu, kde používateľ môže zadávať príkazy priamo cez konzolu.

2 Protokol IMAP

Internet Message Access Protocol (IMAP) je štandardný sieťový protokol používaný na prístup k e-mailovým správam uloženým na serveri. Tento protokol umožňuje klientskym aplikáciám manipulovať so správami priamo na serveri, pričom sa zachováva konzistencia údajov naprieč viacerými zariadeniami.

2.1 Základné charakteristiky IMAP

IMAP je navrhnutý tak, aby poskytoval flexibilný a efektívny spôsob správy e-mailových správ. Medzi jeho kľúčové vlastnosti patria:

- **Synchronizácia správ**: Všetky e-maily sú uložené na serveri, čo umožňuje synchronizáciu medzi viacerými zariadeniami.
- **Prístup z rôznych zariadení**: Užívatelia môžu pristupovať k svojim správam z ktoréhokoľvek zariadenia, pričom sa všetky zmeny synchronizujú.
- **Offline režim**: Správy môžu byť stiahnuté na lokálne zariadenie pre offline prístup a zmeny sa synchronizujú po pripojení k internetu.
- **Správa zložiek na serveri**: Používateľ má možnosť vytvárať, upravovať a mazať zložky priamo na serveri.
- **Selektívne sťahovanie**: IMAP umožňuje stiahnuť len hlavičky správ alebo len určité časti správ, čo je užitočné pri pomalom pripojení.

2.2 Fungovanie IMAP

IMAP je klient-server protokol, kde e-mailový klient komunikuje so serverom pomocou štandardných príkazov a odpovedí. Proces prebieha nasledovne:

1. **Pripojenie k serveru**: Klient sa pripojí k IMAP serveru, pričom používa port 143 pre nezabezpečené pripojenie alebo port 993 pre zabezpečené pripojenie cez SSL/TLS.

2. **Autentifikácia**: Po pripojení je potrebné overiť identitu používateľa prostredníctvom prihlasovacích údajov (meno a heslo).

3. **Manipulácia so správami**: Používateľ môže prehliadať zoznam správ, čítať správy, presúvať ich medzi zložkami, označovať ich ako prečítané alebo ich mazať.

4. **Synchronizácia**: Zmeny vykonané na klientskom zariadení sa prenášajú na server, čím sa zabezpečuje aktualizácia na všetkých zariadeniach pripojených k rovnakému účtu.

5. **Ukončenie relácie**: Po ukončení práce s e-mailmi sa klient odpojí od servera.

3 Implementácia

3.1 Štruktúra programu

Program `imapcl` je implementovaný v jazyku C++ a pozostáva z funkcií a štruktúr na efektívnu komunikáciu s IMAP serverom.

- **Štruktúra** `IMAPConfig`: Slúži na uloženie všetkých konfiguračných parametrov programu, vrátane nastavení pripojenia, výberu poštovej schránky, režimu sťahovania správ (napr. iba hlavičky alebo nové) a nastavenia šifrovania.

- **Štruktúra** `IMAPConnection`: Uchováva všetky informácie o aktuálnom pripojení, ako sú stavové informácie, socket alebo SSL kontext pre šifrované spojenia.

- **Hlavné funkcie programu**: Implementujú logiku pre pripojenie k serveru, autentifikáciu používateľa, vykonávanie IMAP príkazov, sťahovanie e-mailových správ a správu chýb.

- **Funkcie na spracovanie chýb**: Tieto funkcie riadia spracovanie chýb v priebehu celého programu. Ak sa vyskytne chyba, generujú chybové hlásenie.

3.2 Spracovanie argumentov príkazového riadka

Program spracováva argumenty príkazového riadka pomocou knižnice `getopt`. Hlavné argumenty zahŕňajú:

- `-p port`: Definuje port servera. Ak nie je zadaný, použije sa predvolený port.
- `-T`: Aktivuje šifrovanie. Ak nie je zadané, pripojenie prebieha bez šifrovania.
- `-n`: Sťahuje len nové (neprečítané) správy.
- `-h`: Sťahuje len hlavičky správ namiesto celého obsahu.
- `-a auth_file`: Definuje cestu k autentifikačnému súboru obsahujúcemu prihlasovacie údaje.
- `-b MAILBOX`: Názov poštovej schránky (default je `INBOX`).
- `-o out_dir`: Určuje výstupný adresár, kde sa uložia stiahnuté správy.
- `-i`: Zapína interaktívny režim.

Po spracovaní argumentov program načíta konfiguráciu a použije ju pri inicializácii pripojenia k IMAP serveru.

3.3 Pripojenie k IMAP serveru

3.3.1 Pripojenie bez šifrovania

Funkcia `connectToServer` inicializuje pripojenie k IMAP serveru bez šifrovania. Používa systémové volania na vytvorenie socketu a spojenie so serverom cez zadaný port.

3.3.2 Pripojenie so šifrovaním

`connectToSecureServer` vytvára šifrované pripojenie pomocou TLS cez knižnicu OpenSSL. Najprv inicializuje potrebné SSL/TLS funkcie a vytvorí SSL kontext. Následne vytvorí zabezpečené BIO spojenie a nastaví adresu servera. Pomocou `BIO_do_connect` sa pripojí k serveru.

3.4 Autentifikácia používateľa

Po úspešnom pripojení program autentifikuje používateľa načítaním prihlasovacích údajov z autentifikačného súboru. Funkcia `performLogin` posiela príkaz `LOGIN` s prihlasovacími údajmi na server. Ak je autentifikácia úspešná, program pokračuje vo vykonávaní ďalších operácií.

3.4.1 Posielanie príkazov

Funkcia `sendMessage` zodpovedá za odoslanie príkazov na server. Každý príkaz je označený unikátnym ID správy.Program po odoslaní príkazu čaká na odpoveď zo servera.

3.5 Sťahovanie správ

Funkcia `downloadMessages` zabezpečuje sťahovanie e-mailových správ zo servera. Na vyhľadanie správ na serveri sa používajú príkazy `SEARCH` a `FETCH`.

3.6 Interaktívny režim

Interaktívny režim umožňuje používateľovi zadávať príkazy priamo cez konzolu. Po úspešnom pripojení a autentifikácii program čaká na zadanie príkazov, ako sú `DOWNLOADNEW`, `DOWNLOADALL`, a `READNEW`.

3.7 Správa chýb

Funkcia `handleError` slúži na spracovanie chýb. V prípade chyby funkcia nastaví kód chyby a zobrazí chybové hlásenie.

3.8 Odhlásenie a uvoľnenie zdrojov

Na konci behu programu alebo po ukončení interaktívneho režimu sa používateľ odhlási zo servera pomocou funkcie `logout`, ktorá odošle príkaz `LOGOUT`. Funkcia `disconnect` následne uvoľní všetky použité zdroje, ako sú sockety alebo SSL kontexty.

3.9 Hlavná funkcia programu

Hlavná funkcia programu spracuje argumenty príkazového riadka a vykoná operácie na základe zadaných parametrov. Ak je aktivovaný interaktívny režim, program prechádza do funkcie `runInteractive`. V opačnom prípade program pripojí k serveru, prihlási používateľa, vykoná stiahnutie správ a následne sa odhlási a ukončí pripojenie.

4 Preklad a spustenie

4.1 Požiadavky na preklad a spustenie

Na preklad a spustenie programu `imapcl` sú potrebné nasledujúce nástroje a knižnice:

- G++ kompilátor (minimálne verzia podporujúca `C++17`).
- Knižnica `openssl` pre šifrovanie cez TLS.
- Makefile na jednoduché spracovanie prekladu a čistenie súborov.

4.2 Príklady vstupov a výstupov

```
./imapcl eva.fit.vutbr.cz -n -o maildir -a auth_file -i
imapcl> DOWNLOADALL Drafts
Staženy 3 nové zprávy ze schránky Drafts.
imapcl> QUIT

./imapcl eva.fit.vutbr.cz -h -b Sent -o maildir -a auth_file
Stiahnute 80 sprav (iba hlavicky) zo schranky Sent.

./imapcl eva.fit.vutbr.cz -n -o maildir -a auth_file
Stiahnute 1 nove sprava zo schranky INBOX.
```

5 Testovanie

Testovanie na reálnej schránke

Program bol testovaný pomocou pripojenia k mojej vlastnej e-mailovej schránke na serveri `eva.fit.vutbr.cz`. Testovanie zahŕňalo:

- Pripojenie k serveru pomocou nešifrovaného a šifrovaného spojenia (TLS).
- Autentifikácia používateľa cez konfiguračný súbor s prihlasovacími údajmi.
- Sťahovanie nových správ (`-n`) a hlavičiek správ (`-h`).
- Sťahovanie všetkých správ zo špecifickej schránky (`Sent`, `Drafts`).
- Použitie interaktívneho režimu na zadávanie príkazov ako `DOWNLOADNEW`, `DOWNLOADALL`, a `QUIT`.

Testovanie chybových stavov

Na overenie správneho spracovania chybových stavov boli testované situácie ako:

- Pripojenie k neexistujúcemu serveru.
- Použitie neplatného konfiguračného súboru.
- Zadanie neznámych príkazov v interaktívnom režime (`haha`, `DOWLOADONE`).
- Nesprávne spracovanie odpovede od servera.

Zdroje

Literatúra

[1] Crispin, M. (2003). *RFC 3501: INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. Internet Engineering Task Force (IETF). Dostupné na: https://www.rfc-editor.org/rfc/rfc3501

[2] Resnick, P. (2008). *RFC 5322: Internet Message Format*. Internet Engineering Task Force (IETF). Dostupné na: https://www.rfc-editor.org/rfc/rfc5322

[3] Hall, B. (2003). *Beej's Guide to Network Programming*. Dostupné na: https://beej.us/guide/bgnet/

[4] OpenSSL Project. (n.d.). *OpenSSL Documentation*. Dostupné na: https://www.openssl.org/docs/

[5] Linux Programmer's Manual. *getopt_long(3)*. Dostupné na: https://man7.org/linux/man-pages/man3/getopt.3.html