VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

IMAPCL

PROJEKT DO PŘEDMĚTU ISA

AUTOR PRÁCE **JAN OSUSKÝ**
AUTHOR

,

TOULOUSE 2024

Obsah

Kapitola 1

Úvod

Projekt **imapcl** má za cíl vytvoření aplikace pro stahování elektronické pošty IMAP serveru prostřednictvím protokolu IMAP4rev1, definovaného v RFC 3501.

Aplikace **imapcl** po připojení ke zvolenému serveru (přes šifrovaný nebo nešifrovaný kanál podle volby uživatele) stáhne požadované zprávy a uloží je do předem definovaného výstupního adresáře, kde bude každá zpráva uložena jako samostatný soubor. Zprávy budou ukládány ve formátu Internet Message Format podle specifikace RFC 5322, která určuje standardizovaný způsob formátování emailových zpráv, včetně hlaviček, příjemců, odesílatele a těla zprávy.

Protokol RFC 3501

RFC 3501 je protokol umožňující e-mailovým klientům přístup k e-mailovým zprávám uloženým na poštovním serveru. IMAP ukládá zprávy na serveru, což umožňuje přístup ke schránce z více zařízení a šetří místo v lokálním úložišti, protože zprávy se stahují jen dočasně. Uživatelé mohou na serveru vytvářet, přejmenovávat, mazat a spravovat složky, což usnadňuje organizaci e-mailů, přičemž změny jsou synchronizovány na všech zařízeních. Na rozdíl od POP3, který stahuje všechny zprávy, IMAP umožňuje stahovat pouze záhlaví nebo tělo konkrétních zpráv, což šetří data. IMAP podporuje současný přístup více klientů ke schránce, kde každý vidí změny v reálném čase, což je výhodné pro sdílené účty. Protokol umožňuje synchronizaci, takže všechny změny provedené na jednom zařízení jsou viditelné i na ostatních.

Komunikace s IMAP serverem probíhá pomocí příkazů, jako například LOGIN pro autentizaci, SELECT nebo EXAMINE pro otevření schránky, FETCH pro stažení částí zpráv, STORE pro změnu příznaků zpráv, SEARCH pro vyhledávání a LOGOUT pro ukončení sezení. IMAP podporuje SSL/TLS šifrování (IMAPS), čímž je zajištěna bezpečnost připojení. Typický průběh IMAP sezení začíná připojením k serveru na portu 143 (IMAP) nebo 993 (IMAPS), následuje autentizace pomocí LOGIN, výběr schránky (např. INBOX), stahování záhlaví zpráv pomocí FETCH, stažení celé zprávy při otevření konkrétního e-mailu, správa zpráv příkazem STORE a odhlášení příkazem LOGOUT. Hlavní rozdíl mezi IMAP a POP3 spočívá v tom, že IMAP uchovává zprávy na serveru a umožňuje synchronizaci mezi zařízeními, zatímco POP3 zprávy stáhne a obvykle smaže ze serveru, což znamená, že zprávy jsou uloženy pouze lokálně. IMAP4rev1 je hojně využíván v e-mailových službách zaměřených na synchronizaci mezi zařízeními, tento projekt implementuje část tototo protokolu a to konkrétně zobrazení doručených zpráv [1].

Kapitola 2

Návrh

Při návrh aplikace jsem se rozhold pro rozdělení aplikace do několika souborů. Hlavní část programu s řídící logikou se bude nacházet v souboru **main.cpp** . V tomto souboru se bude také nacházet kompletní načítání vstupu z příkazové řádky stejně jako jejich parsování a ověření přítomnosti povinných argumentů.

Pro ustanovení připojení standartním i zabezpečeným připojením přes TSL, vzniknou soubory **connect.cpp** a **connect.h** . V těchto souborech proběhne i načtení certifikátů a dalších potřebných kroků pro ustanovení připojení k serveru.

Poslední dvojce souborů s názvy **imap.cpp** a **imap.h** budou sloužit pro komunikaci protokolem IMAP4rev1. Konkrétně pro přihlášení a ohlášení uživatele, dále pro vybrání požadované schránky, stažení emailů a také jejich parsování a uložení do požadované složky.

Kapitola 3

Implementace

Tato kapitola se bude věnovat některým zajímavým částem implementace a odůvodněním těchto rozhodnutí.

Při implementaci spojení se servrem byla využitá knihovna OpenSSl za pomocí tutoriálu na stránkách IBM zmíněného v zadání projektu[3]. Pro připojení byla využita knihovna `BIO` a to pro jak zabezpečené připojení přes port 993 tak také nezabezpečené přes port 143. Tato knihovna rovněž poskytla rozhraní pro práci s certifikáty. Pro udržení připojení a případně provední nového "handshaku"automaticky při vyžádání serverem je využit ukazatel `SSL` který má nastavený mód na `SSL_MODE_AUTO_RETRY`.

Při implementaci části programu provádějící komunikaci přes protokol IMAP, vznikly celkem 4 funkce pro jejich provední. Funkce `login` a `logout` souží k autentikaci. Pro provedení autentikace se pro stažení požadovaných emailů využije funkce `fetchMail`. V této funkci dojde k vybrání požadované schránky, vyhledání požadováných zpráv a jejich stažení popřípadně stažení jejich hlaviček v závislosti na požadavcích uživatele [2]. V případě, že uživatel využije parameter `-n` je použit příkaz `SEARCH UNSEEN`. V zadání projektu se píše, že parameter `-n` má stáhnout pouze nové zprávy. Co znamená nová zpráva je dost nejasné, nicméně většina dnešních emailových klientů nabízí ve svých filtrech možnost "nepřečtené"nikoliv nové a proto mi využití příkazu `UNSEEN` připadá jako nejlepší volba.

Testování

Program byl testován na serveru Merlin a to pro připojení k serverům `eva.fit.vutbr.cz` a `imap.stud.fit.vutbr.cz` a to až po stažení celkem 2500 emailů (veškerý obsah mého účtu). Zabezpečené i nezabezpečené připojení bylo otestováno díky serveru `eva.fit.vutbr.cz`. Během testování byly použity veškeré možné kombinace vstupních parametrů. Testování odalilo nedostatky implementace které byly odstraněny. Například, při stahování větších emailů ale také schránek s velkým množstvím emailů docházelo k nenačtění celého obsahu odpovědi během jednoho běhu `BIO_read`. Z tohoto důvodu bylo toto čtení předáno do cyklu, kde se konec zprávy ověřuje vyledáváním řetězce `"Axxx OK"`.

Při dalším testováni se i u tohoto postupu naskytla chyba a zhruba u každého tisícího emailu došlo k rozdělení zprávy mezi dvě čtení `BIO_read` právě mezi `"Axxx` a `OK"`. Proto byla podmínka ukončení rozšířena aby se předešlo těmto situacím.

```
if (strstr(response, "A00 OK") != nullptr || strstr(response, "OK Fetch") != nullptr
|| strstr(response, "Fetch completed") != nullptr) {
    fetch_completed = true;
```

```
}
```

Během testování bylo také odhaleno, že součástí uložených emailů jsou taktéž voláni protokolu IMAP jako `FETCH`. Ty byli následně z emailů odstraněny i když mohou být potenciálně užitečné.

Kapitola 4

Použití

```
./imapcl server [-p port] [-T [-c certfile]
[-C certaddr]] [-n] [-h] -a auth_file [-b MAILBOX] -o out_dir
```

Parametry

- **server**: Povinný parametr. Zadejte název serveru (IP adresa nebo doménové jméno) jako cílový zdroj.
- **-p port** *(volitelný)*: Určuje číslo portu na serveru. Doporučuje se vybrat výchozí hodnotu podle typu připojení (např. IMAP nebo IMAPS) a dle registrace u IANA.
- **-T** *(volitelný)*: Aktivuje šifrování (protokol IMAPS). Pokud tento parametr není uveden, použije se nešifrovaná varianta protokolu.
- **-c certfile** *(volitelný)*: Soubor s certifikáty, který se použije pro ověření platnosti SSL/TLS certifikátu, který předkládá server.
- **-C certaddr** *(volitelný)*: Adresář s certifikáty pro ověřování SSL/TLS certifikátu serveru. Pokud není specifikováno, použije se výchozí adresář `/etc/ssl/certs`.
- **-n** *(volitelný)*: Při aktivaci bude aplikace pracovat pouze s novými zprávami.
- **-h** *(volitelný)*: Při aktivaci budou stahovány pouze hlavičky zpráv, nikoli celé zprávy.
- **-a auth_file**: Povinný parametr. Soubor `auth_file` obsahuje informace pro autentizaci (příkaz LOGIN) a slouží k ověření přístupu k serveru.
- **-b MAILBOX** *(volitelný)*: Specifikuje název schránky na serveru, se kterou se bude pracovat. Výchozí hodnota je `INBOX`.
- **-o out_dir**: Povinný parametr. Určuje výstupní adresář, kam se uloží stažené zprávy.

Příklad použití

```
./imapcl mail.server.cz -p 993 -T -a auth_file -b INBOXMine -o /home/user/emails
```

Literatura

[1] Crispin, M. (2003). INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1 (IMAP4rev1). RFC 3501. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc3501section-6
[2] Resnick, P. (2008). Internet Message Format. RFC 5322. Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc5322
[3] IBM Developer. (n.d.). Using OpenSSL to secure your communications. Retrieved October 28, 2024, from https://developer.ibm.com/tutorials/l-openssl/