VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí

Klient POP3 s podporou TLS

23. října 2021 Pavel Bobčík (xbobci03)

Obsah

1 Úvod do problematiky **2**

2 Návrh aplikace **2**
2.1 Zpracování parametrů . . . 2
2.2 Komunikace klienta se serverem . . . 2

3 Implementace **2**
3.1 parser . . . 2
3.1.1 int checkArg(int argc, char* argv[]) . . . 2
3.1.2 string_code hashit(string const inString) . . . 3
3.1.3 Další funkce parser.cpp . . . 3
3.2 main . . . 3
3.2.1 Kontrola parametrů . . . 3
3.2.2 Navázání spojení . . . 3
3.2.3 Autorizace . . . 3
3.2.4 Stažení zpráv . . . 3
3.2.5 Ukončení spojení . . . 4
3.2.6 Pomocné funkce . . . 4

4 Použití programu **4**
4.1 Spuštění . . . 4
4.2 Popis parametrů . . . 4
4.3 Formát autentizačního souboru . . . 4

5 Literatura **5**

1 Úvod do problematiky

Klient POP3 zprostředkovává stahování elektronické pošty pomocí internotového protokolu POP nebo-li Post Office Protocol. Současná verze, tedy třetí (značená POP3), je standartizována v `RFC1939`. Protokol funguje na principu stahování zpráv ze serveru na lokální zařízení. Často dochází i k jejich smazání na serveru. Komunikace probíhá na základě požadavků ze strany klienta a odpovědí ze strany serveru. Základní verze protokolu POP3 je nešifrovaná, nicméně nám umožňuje navázat i šifrovanou komunikaci SSL/TLS.

2 Návrh aplikace

Aplikace se skládá ze dvou částí. První je zpracování parametrů na vstupu a druhou je samostatná komunikace klienta se serverem.

2.1 Zpracování parametrů

Zpracování parametrů je vypracováno v souboru `parser.cpp` a jeho hlavičkovém souboru `parser.hpp`. Problematika je řešena pomocí `switch` nad textovými řetězci reprezentujícími jednotlivé parametry. Jelikož C++ neumožňuje tuto operaci provést nad textovými řetězci, bylo využito operace enumeration, nebo-li `enum`, pro získání jejich číselné hodnoty. Funkce byla inspirována z odpovědi na Stack Overflow[1].

Dále se provede kontrola duplicity parametrů, správnosti zadání IPv4, vyplnění povinných parametrů, správné vyplněné přepínačů a vypsání nápovědy. Tyto údaje jsou následně přístupné pomocí funckí zvaných `getter`.

2.2 Komunikace klienta se serverem

Komunikace je vypracována v souborech `main.cpp` a `main.hpp`. Na počátku provede kontrolu vstupních parametrů. Následně dle uvedených parametrů provádí připojení k serveru pomocí nešifrované nebo šifrované komunikace. Toto připojení je ovlivněno přepínači -T a -S, nebo neuvedením ani jednoho. Poté již dochází ke komunikaci klienta se serverem pomocí odesílání požadavků, jimiž jsou například USER jméno a PASS heslo pro přihlášení. Server na tyto požadavky odpovídá pozitivně +OK nebo negativně -ERR.

Klient dle obdržené odpovědi ví, zdali se nadále jedná o platnou komunikaci a můžu pokračovat, nebo zdali došlo k nějaké chybě v komunikaci. Při úspěšné komunikaci klient stáhne jednotlivé zprávy, které si uloží do souborů ve složce uvedené v parametrech.

Komunikace se ukončí odesláním příkazu QUIT ze strany klienta.

3 Implementace

Projekt je naprogramován v jazyce C++. Rozdělen je na `parser.cpp` obsahující zpracování parametrů a `main.cpp` sloužící ke komunikaci. A to včetně jejich hlavičkových souborů. Jednotlivé soubory jsou rozděleny do funkcí. Ke komunikaci se serverem je využita knihovna `openssl/bio.h`.

3.1 parser

3.1.1 int checkArg(int argc, char* argv[])

Funkce pro kontrolu parametrů na vstupu. Jednotlivé parametry porovnává pomocí `switch`.

Jelikož C++ neumožňuje využít u `switch` textové hodnoty, byly tyto hodnoty převedeny do číselné podoby pomocí enumeration.

3.1.2 string_code hashit(string const inString)

Funkce k převodu textové reprezentace parametru na číselnou hodnotu enumeration. Na základě názvu vrátí hodnotu z enumeration.

Řešení bylo převzato z odpovědi na Stack Overflow.[1]

3.1.3 Další funkce parser.cpp

int checkIP()[4]
int mandatoryData()
string getServer()
a zbylé funkce `getter` pro zpřístupnění hodnot parametrů ze souboru `parser.cpp` v `main.cpp`.

3.2 main

3.2.1 Kontrola parametrů

Na počátku dojde ke kontrole parametrů zavoláním funkce `checkArg` ze souboru `parser.cpp`.

3.2.2 Navázání spojení

Dle zadaných přepínačů dojde k navázání nešifrovaného nebo šifrovaného spojení.

Pokud není uveden žádný přepínač, dojde k navázání nešifrovaného spojení pomocí funkce `connectPOP3`.

Při uvedení přepínače -T dojde k navázání šifrovaného spojení SSL pomocí funkce `connectSSL`. Dále dojde k nastavení certifikátů pomocí funkce `getCtx` a kontrole obdržených certifikátů od serveru funkcí `checkCert`.

Poslední možností je šifrované spojení TLS zahájené přepínačem -S. Funkce `connectTLS` napočátku naváže nešifrované spojení pomocí `connectPOP3`. Poté odešle požadavek STLS a v případě kladné odpovědi přejde na šifrované spojení SSL. Opět dojde k nastavení a kontrole certifikátů.

3.2.3 Autorizace

Po úspěšném připojení je klient požádán o autorizaci. Ta je provedena funkcí `login`. Autorizační údaje jsou získány ze souboru uvedeného v parametru pomocí funkce `getAuthInfo`.

3.2.4 Stažení zpráv

Pro stažení zprávy je potřeba získat informace o počtu zpráv pomocí funkce `getMsgCount`. Poté klient postupně žádá o zprávy. Z obdržené zprávy zjistí Message-ID, které uloží do seznamu stažených zpráv funkcí `saveMsgId` [3]. Pokud uživatel nastavil stažení pouze nových zpráv, tak dojde ke kontrole, zdali se Message-ID obdržené zprávy nevyskytuje mezi uloženými Message-ID funkcí `isNewMail`. Pokud ano, zpráva se neuloží a klient pokračuje ve vykonávání programu. Je-li nastaven přepínač -d, označí funkce `setMsgToDelete` zprávy ke smazání.

Po obdržení koncového znaku dojde k úpravě zprávy a uložení do souboru pomocí funkce `saveMsg`. Funkcí `getFileName`[5] je vytvořen název souboru podle předmětu zprávy, ze kterého jsou odstraněny znaky, jenž by mohly způsobovat problémy. V případě prázdného předmětu je pojmenován jako unknown. Pokud dojde k opakovanému stažení zprávy, nebo ke stažení zprávy se stejným předmětem, již se vyskytuje v souboru,

bude k názvu přidána inkrementovaná hodnota počtu souborů s identickým názvem. Soubory jsou ukládány do uvedného adresáře. Jedná-li se o neexistující adresář, dojde k jeho vytvoření. Kontrolu adresáře a jeho případné vytvoření zajišťuje funkce `checkOutputDir`[2].

3.2.5 Ukončení spojení

O ukončení spojední se stará funkce `quitMessage` a o následné uvolenění `closeBioAndEnd`.

3.2.6 Pomocné funkce

Mezi další funkce patří zaslání požadavku `sendMessage`, obdržení odpovědi `readMessage` a vypsání počtu stažených zpráv `outputMsg`.

4 Použití programu

4.1 Spuštění

./popcl <server> [-p <port>] [-T|-S [-c <certfile>] [-C <certaddr>]] [-d] [-n] -a <auth_file> -o <out_dir>

Pro nápovědu:

./popcl --help
./popcl -h

4.2 Popis parametrů

- Povinný parametr <server> (IP adresa, nebo doménové jméno) požadovaného zdroje.
- Volitelný parametr -p specifikuje číslo portu <port> na serveru.
- Parametr -T zapíná šifrování celé komunikace (pop3s), pokud není parametr uveden použije se nešifrovaná varianta protokolu.
- Parametr -S naváže nešifrované spojení se serverem a přejde na šifrovanou variantu protokolu.
- Volitelný parametr -c definuje soubor <certfile> s certifikáty.
- Volitelný parametr -C určuje adresář <certaddr>, ve kterém se mají vyhledávat certifikáty.
- Při použití parametru -d se zašle serveru příkaz pro smazání zpráv.
- Při použití parametru -n se bude pracovat (číst) pouze s novými zprávami.
- Povinný parametr -a <auth_file> obsahuje konfigurační soubor <auth_file>.
- Povinný parametr -o <out_dir> specifikuje výstupní adresář <out_dir>, do kterého má program stažené zprávy uložit.

4.3 Formát autentizačního souboru

```
username = jméno
password = heslo
```

5 Literatura

Rfc1939. IETF Datatracker [online]. Spojené státy americké: Dover Beach Consulting, 1996 [cit. 2021-10-23]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc1939

Rfc2595. IETF Datatracker [online]. Spojené státy americké: Innosoft International, 1999 [cit. 2021-10-23]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc2595

Rfc5322. IETF Datatracker [online]. Spojené státy americké: Qualcomm Incorporated, 2008 [cit. 2021-10-23]. Dostupné z: https://datatracker.ietf.org/doc/html/rfc5322

Secure programming with the OpenSSL API. IBM Developer [online]. Spojené státy americké: IBM, 2004 [cit. 2021-10-23]. Dostupné z: https://developer.ibm.com/tutorials/l-openssl/

Reference

[1] D.Shawley. Why the switch statement cannot be applied on strings? - answer. [online], 2013. [cit. 23. 10. 2021].

[2] Ingo Leonhardt. Portable way to check if directory exists [windows/linux, c] - answer. [online], 2013. [cit. 23. 10. 2021].

[3] Rico. std::ofstream, check if file exists before writing - answer. [online], 2013. [cit. 23. 10. 2021].

[4] Martin Stone. Splitting a c++ std::string using tokens, e.g.

[5] tutorialspoint.com. C++ regex library - regex_search. [online]. [cit. 23. 10. 2021].