

FAKULTA INFORMAČNÍCH
TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Klient POP3 s podporou TLS

15. listopadu 2021

Josef Krušina (xkrusi01)

Obsah

1	Úvod	2
2	POP3	2
3	Ovládání	2
4	Vlastní implementace	2
4.1	popcl.cpp	2
4.2	OpenSSLClass.cpp	2
4.3	ArgumentParser.cpp	2
4.4	ErrorHandling.cpp	2
5	Tok aplikace popcl	3
6	Funkce retrieveMail()	3
6.1	Implementace -n	3
7	Inspirace	3

1 Úvod

Dokumentace `manual.pdf` bude sloužit k bližšímu popisu mnou implementovaného klientu POP3.

2 POP3

POP3 protokol je používán pro stahování emailových zpráv ze serveru na zařízení klienta. Je pro něj vyhrazený TCP port 110. Služba je nešifrovaná, pro účely zabezpečení se může zašifrovat pomocí SSL/TLS.

3 Ovládání

Program se dá spustit pomocí příkazu

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

, kde pořadí parametrů je libovolné, až na `<server>`, u kterého se počítá, že bude vždy první. Je implementováno zpracování všech parametrů. Implicitní hodnota portu je rovna 110 pro ne-zabezpečené připojení pomocí SSL.

4 Vlastní implementace

Jako jazyk pro tento projekt jsem si zvolil C++, abych si více tento jazyk procvičil a donutil se více tvořit pomocí objektově orientovaném programování. Program `popcl` je složený ze čtyř C++ souborů. Dva zastupují třídy – `ArgumentParser.cpp` a `OpenSSLClass.cpp`. Soubor `ErrorHandling.cpp` obsahuje funkci pro chybovou hlášku. Posledním souborem je `popcl.cpp`, který plní funkci hlavního souboru.

4.1 `popcl.cpp`

Tento soubor inicializuje instance tříd a pomocí funkcí, schovaných za abstrakcí, plní výslednou chtěnou funkcionalitu programu.

4.2 `OpenSSLClass.cpp`

Obsahuje třídu `OpenSSLClass`, která má na starost inicializaci, otevření a uzavření spojení se serverem. Dále je schopna komunikovat se serverem pomocí protokolu POP3s.

4.3 `ArgumentParser.cpp`

Obsahuje třídu `ArgumentParser`, která zpracovává argumenty pomocí vlastně implementovaného parseru. Tyto argumenty přeposílá `popcl.cpp` třídě `OpenSSLClass`, která pomocí nich mění vlastní funkcionalitu.

4.4 `ErrorHandling.cpp`

Funkce `writeError()` pomocí parametrů vrátí požadovanou návratovou hodnotu a vypíše chybovou hlášku na standardní chybový výstup.

5 Tok aplikace popcl

Pomocí inicializované třídy `ArgumentParser` `arguments` se využitím metody `arguments.parseArguments(int argc, char* argv[])` zpracují všechny argumenty a jejich hodnoty se načtou do soukromých třídních proměnných, které jsou dále přístupné pomocí getterů. Případné chyby v sintaxi jsou hlášeny na standardní chybový výstup. Dále se rozhodne, jaké připojení uživatel vyžaduje a postupně se pomocí inicializované třídy `OpenSSLClass` `ossl` navazuje. Po úspěšném navázání spojení a handshake je uživatel přihlášen pomocí příkazů `USER` a `PASS`. Dále je vypsán počet zpráv ke stažení příkazem `STAT`. Ve funkci `ossl.retrieveMail(std::string directory, bool newMessagesOnly)` je provedeno stažení všech zpráv, funkci blíže popíši v další sekci. Pokud uživatel chtěl je provedeno smazání zpráv pomocí funkce `ossl.deleteMail()`, která v cyklu nad všemi zprávami volá `DEL`. Pro rozloučení a ukončení POP3 se serverem je zavoláno `QUIT`. Spojení SSL/TLS je následně také ukončeno.

6 Funkce `retrieveMail()`

Bohužel mi po implementaci parametru `-n` funkce výrazně zhoustla a znepráhlednila se. Pokusil jsem se ji řádně okomentovat, abych nastínil myšlenkový proces v daných pasážích kódu. Funkce kontroluje, zda jde otevřít složka určená pro ukládání zpráv. Tuto funkcionalitu jsem měl rovnou implementovat do třídy `ArgumentParser`, protože zde moc nedává smysl.

Stahování zpráv je založeno na dvou cyklech. Hlavním `for` cyklu, který proběhne tolikrát, kolik zpráv je na serveru dostupných ke stažení. A vedlejším `do while()` cyklu, který se opakuje dokud není splněno `isMessageEnd()`, které hlídá, zda byl přečtený konec zprávy. V tomto cyklu je zpráva ochuzena o odpověď serveru. Zprávu ukončující `\r\n.\r\n` je vyměněno za `\r\n` pomocí regex funkce `replace()`.

6.1 Implementace `-n`

Unikátnost ukládaných pouze nových, již nestážených zpráv je zaručena pomocí `Message-ID:` , které je použité ve jméně zprávy. Před zapisováním nových zpráv je vždy kontrolováno, zda již neexistuje takto pojmenovaný soubor. Pokud ano, tak se zpráva nezapisuje. Bohužel mi nefungovala implementace, kde bych toto kontroloval ihned po stažení hlavičky zprávy, a tak musím stahovat celou zprávu, než mohu kontrolovat unikátnost.

7 Inspirace

Většina případné inspirace, proběhla z příkladů užití funkcí na referenčních stránkách pro jazyk C++, jako je <https://www.cplusplus.com>. Dále jsem čerpal z referenčních zdrojů přiložených k zadání projektu.