Fakulta informačních technologií

Vysoké učení technické v Brně

# Klient POP3 s podporou TLS

Síťové aplikace a správa sítí

Jan Kubica (xkubic39) 2017/2018

# Obsah

# Úvod

POP (Post Office Protocol) je internetový protokol, který se používá pro stahování e-mailových zpráv ze vzdáleného serveru na klienta. Jedná se o aplikační protokol pracující přes TCP/IP připojení. V současnosti je používána zejména třetí verze (POP3), která byla standardizována v roce 1996 v RFC 1939. [5]

Program `popcl` byl vytvořen v rámci projektu do předmětu *síťové aplikace a správa sítí* na Fakultě informačních technologií Vysokého učení technického. Cílem projektu bylo napsat program `popcl`, který bude umožňovat čtení elektronické pošty skrze protokol POP3 (RFC 1939 s rozšířeními pop3s a POP3 STARTTLS - RFC 2595).

# Použití

## Instalace

Kompilace aplikace je zajištěna skrze přiložený `Makefile`.

## Spuštění

Zkompilovaný program se spouští přes vytvořený soubor `popcl`. Výstup klienta při bezchybné komunikaci začíná znakem '>', v opačném případě řetězcem `ERROR:`. Klient odpoví vždy jednořádkovým výstupem. Správný výstup pak může vypadat např.:

```
$ ./popcl pop3.centrum.cz -a auth_file.txt -o inbox -T
> 43 new messages have been downloaded to your inbox folder.
```

## Help

Při zadání parametru `-h` nebo `-help`, případně při spuštění bez parametrů, je vytištěna nápověda a program je ukončen s hodnotou `0`.

```
Program usage:
popcl <server> [-p <port>] [-T|-S [-c <certfile>] [-C <certaddr>]] [-d] [-n] -a
<auth_file> -o <out_dir>
     -p   port            - sets remote TCP port
     -T   pop3s           - sets pop3s secured connection
     -S   STSL            - connects with STLS (RFC 2595)
     -c   certfile        - path to certificate file
     -C   certfolder      - path to folder with certificates
     -d   delete          - deletes all messages from server
     -n   new messages    - works only with new messages
     -a   authfile        - path to authentication file
     -o   outdir          - folder where messages will be stored
```

# Rozdělení projektu

Pro přehlednost byl projekt systematicky rozdělen do více souborů:

- `src/args.cpp`
- `src/args.h`
- `src/connection.cpp`
- `src/connection.h`
- `src/error.h`
- `src/fileman.cpp`
- `src/fileman.h`
- `src/help.h`
- `src/pop3man.cpp`
- `src/pop3man.h`
- `Makefile`
- `popcl.cpp`

# Implementace

## Třídy

Funkční prvky projektu jsou rozděleny do jednotlivých tříd. Konkrétně se jedná o třídy:

`Arguments` – třída pro zpracování argumentů,
`Connection` – třída definující socketovou komunikaci klient – server a SSL/TSL zabezpečení,
`FileManager` – třída obsluhující práci se soubory,
`Pop3Manager` – třída, implementující dílčí kroky komunikace s POP3 serverem.

## OpenSSL

OpenSSL část projektu je implementována v souboru pop3man.cpp. Nebylo využito BIO socketů, funkce z knihovny OpenSSL staví nad klasickými BSD sockety.

## Error messages

```
    SUCCESS_EXIT = 0,
    BAD_LOGIN_OR_PW = -1,
    HOST_UNRESOLVED = -2,
    STLS_NOT_SUPPORTED = -5,
    FLAGS_INCOMPATIBLE = -11,
    DOUBLE_ARGUMENT = -12,
    ARGUMENT_ERROR = -13,
    REQUIRED_ARGUMENT = -14,
    INCOMPATIBLE_PORT = -15,
    UNKNOWN_ARGUMENT = -16,
    DIR_NOT_CREATED = -21,
    AUTH_FILE_ERR = -22,
    AUTH_SYNTAX_ERR = -23,
    BSD_ERROR = -31,
    SSL_ERROR = -32,
    CONNECTION_ERROR = -42,
    UNSPECIFIED_INTERNAL_ERROR = -99
```

# Individuální řešení

## Identifikace emailu

Pro správnou identifikaci každého emailu jsou při uložení získávány informace:

*messageUIDL* – unikátní číslo na straně serveru, které je získáno příkazem UIDL
*messageID* – parametr v hlavičce emailové zprávy, který je získán extrahováním z emailu

Tyto informace jsou poté pomocí funkce `bool actualizeTsvFile(std::string uidl, std::string mid)` zapsány do souboru `.maildir.tsv` ve stejné složce jako jsou jednotlivé emailové zprávy.

## Hashovací funkce

Parametr *Message-ID* je podle RFC 5322 pouze doporučený a nemusí tedy být v hlavičce emailu přítomen, stejně tak i podpora příkazu UIDL na POP3 serveru [1] [2]. Aby i přes tuto skutečnost byl zajištěn jednoznačný identifikátor každé zprávy, je hodnota *messageID* vygenerována pomocí hashovácí funkce *djb2*, jejíž autorem je Dan Bernstein [3]. Hashovací funkce vezme celou hlavičku jako řetězec znaků a převede jej na přiměřeně dlouhou posloupnost decimálních čísel. V hashovací funkci je dána počáteční hodnota jako prvočíslo hash = 5381, která je pomocí každého znaku z hlavičky a čísla 33 následně upravována. V závěru je jako výsledná hodnota *messageID* je použita kombinace hodnoty hashovací funkce a povinného parametru *From* z hlavičky emailové zprávy.

`hashvalue.fromvalue`

např. `3626010857947067422.neodpovidat@seznam.cz`

## Uložení souborů

Emailové zprávy jsou ukládány do složky specifikované pomocí parametru `-o`. Název souboru je dán podle hodnoty *messageID*.

## Formát souboru

Předpokládá se, že všechny emaily, které přicházející ze serveru, jsou nativně v platném formátu *Internet Message Format RFC 5322* [2]. Před uložením do cílové složky jsou z emailu ale odebrány zdvojené tečky, neboť tyto tečky nejsou původním obsahu emailové zprávy, nýbrž jsou přidávány do příchozího mailu k odlišení koncové tečky jako ukončujícího znaku pro zprávy od POP3 serveru.

## Soubor `.mail.tsv`

Soubor `.mail.tsv` se nachází po spuštění klienta ve výstupní složce určené parametrem `-o`. Tento soubor je využíván pro identifikaci nových příchozích emailů. Na každém řádku je uložena dvojice:

`messageUIDL\rmessageID`

pozn.: Pokud POP3 server nepodoruje příkaz UIDL, nachází se v souboru pouze hodnota messageID na každém řádku.

### Stažení nových zpráv

Pokud je klient spuštěn s parametrem `-n`, jsou do výstupní složky stažené pouze nové zprávy. Způsob vyhodnocení nového emailu je řešen několika způsoby. Pokud server rozpozná příkaz UIDL, je postupně pro všechny emaily na serveru zjištěna tato hodnota, která je porovnávána pomocí funkce `bool searchTsvFile(const char * uidl, const char * mid)` s informacemi v souboru `.mail.tsv`. Pokud server nezná příkaz UIDL, kontrolují se nové maily podle hodnoty *messageID* získané či vygenerované z hlaviček jednotlivých emailů přes příkaz TOP. Pokud server nepodoruje ani přikaz UIDL, ani TOP, dochází interně ke stažení celých emailů, které jsou uloženy jen v případě nenalezení stejné hodnoty *messageID* v `.mail.tsv` souboru.

# Omezení a rozšíření

- Projekt pro účely testování přizpůsoben pro server merlin.fit.vutbr.cz, jakožto referenčního stroje, a využívá funkce primárně funkce z knihovny SSL verze 1.0.2 [7].
- Při zadání špatného portu je nutno komunikaci ukončit manuálně příkazem `CTRL+C`. Nativně je pak port automaticky nastavován při šifrovaném spojení na standartní 995, nešifrovaném 110 [8].
- Argument `server` je při spuštění automaticky rozpoznán, jde-li o adresu IPv4, IPv6 nebo doménové jméno.

# Literatura

[1] *RFC 1939: Post Office Protocol - Version 3* [online]. [cit. 2017-11-18]. Dostupné z: https://www.ietf.org/rfc/rfc1939.txt

[2] *RFC 5322: Internet Message Format* [online]. [cit. 2017-11-18]. Dostupné z: https://tools.ietf.org/html/rfc5322

[3] *RFC 2595: Using TLS with IMAP, POP3 and ACAP* [online]. [cit. 2017-11-18]. Dostupné z: https://tools.ietf.org/html/rfc2595

[4] *Hash Functions* [online]. [cit. 2017-11-18]. Dostupné z: http://www.cse.yorku.ca/~oz/hash.html

[5] *Beej's Guide to Network Programming: 9.7. gethostbyname(), gethostbyaddr()* [online]. [cit. 2017-11-20]. Dostupné z: http://beej.us/guide/bgnet/output/html/multipage/gethostbynameman.html

[6] *Wikipedia: Post Office Protocol* [online]. [cit. 2017-11-18]. Dostupné z: https://cs.wikipedia.org/wiki/Post_Office_Protocol

[7] *OpenSSL - Manual Pages: ssl library 1.0.2* [online]. [cit. 2017-11-18]. Dostupné z: https://www.openssl.org/docs/man1.0.2/ssl/

[8] *Service Name and Transport Protocol Port Number Registry: iana.org* [online]. [cit. 2017-11-18]. Dostupné z: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[9] *Secure programming with the OpenSSL API: IBM - developerWorks* [online]. [cit. 2017-11-18]. Dostupné z: https://www.ibm.com/developerworks/library/l-openssl/