

Sítové aplikace a správa sítí
POP3 Server

Obsah

1	Úvod	2
2	Post Office Protocol	2
2.1	Komunikácia	2
2.2	Fázy komunikácie	2
3	Návrh aplikácie	2
4	Implementácia	2
4.1	Použité knižnice	3
4.2	Vonkajšie zdroje	3
4.3	Detaily implementácie	3
5	Použitie aplikácie	4
5.1	Spustenie	4
6	Záver	4

1 Úvod

Dokument v skratke vysvetľuje čo je to POP3 server, jeho fungovanie a následne popisuje návrh a implementáciu výslednej aplikácie.

Dokumentácia bola vytvorená pomocou L^AT_EXu.

2 Post Office Protocol

Post Office Protocol je internetový protokol, bežiaci na aplikačnej vrstve, ktorý sa používa na prijímanie elektronickej pošty zo vzdialeného serveru. Na pripojenie klienta k serveru sa využíva TCP/IP spojenie.

Dnes sa používa už jeho tretia verzia (POP3), ktorá je definovaná v RFC 1939[2].

2.1 Komunikácia

Protokol pre komunikáciu využíva TCP port 110. Príkazy sú 3 alebo 4 znaky dlhé, začínajú na začiatku riadku a nerozlišujú veľké a malé písmena. Vďaka ich jednoduchosti je možné tento protokol používať aj ručne. Za niektorými príkazmi môžu nasledovať argumenty. Riadky musia byť ukončené CRLF[3] a každá odpoveď serveru musí začínať buď +OK, alebo -ERR.

2.2 Fázy komunikácie

POP3 server má 3 fázy:

- autorizáčná - počas tejto fázy server čaká na prihlasovacie údaje od klienta, ten po ich odoslaní čaká na odpoveď serveru. Používané príkazy v tejto fáze sú APOP, USER, PASS a QUIT.
- transakčná - do tejto fázy sa server dostáva po úspešnej autorizácii užívateľa, počas nej sa deje všetka ďalšia komunikácia s klientom. Používané príkazy v tejto fáze sú STAT, LIST, UIDL, RETR, RSET, NOOP, DELE a QUIT.
- aktualizáčná - nasleduje transakčnú fázu, server sa do nej dostáva po tom čo klient zadá príkaz QUIT v transakčnej fáze. Počas nej sa na servery aplikujú všetky zmeny vyžiadané užívateľom napr. mazanie správy príkazom DELE.

3 Návrh aplikácie

Aplikácia `popser` má byť schopná pracovať s jedným klientom, nesmie spúšťať ďalšie procesy, musí využívať neblokujúci socket [1] a má vedieť pracovať nad jednou zložkou Maildir, ktorá obsahuje podzložky `/new` a `/cur`.

Má obsahovať kompletnú implementáciu podľa štandardu pre POP3 server (RFC 1939)[2]. Teda správne vykonať všetky 3 fázy komunikácie medzi klientom a serverom. Taktiež by mala pracovať len s IPv4.

Nesmie čítať obsah súborov v zložke `/cur` mimo autorizačnej fázy a príkazov TOP, či RETR. Musí počítat s tým, že proces môže byť ukončený a následne znovu spustený.

Chybový výstup by mala zapisovať na `stderr` a okrem parametru `-h` by na `stdout` nemala vypisovať vôbec nič.

4 Implementácia

Aplikácia je implementovaná v jazyku C/C++ so štandardom C++11 a používa niektoré objekty zo štandardných knižníc C++. Je testovaná na servere merlin, ktorý bol označený ako referenčný stroj a tiež na Ubuntu 16.04.

4.1 Použité knižnice

```
#include <unistd.h>
#include <pthread.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <iostream>
#include <cstring>
#include <dirent.h>
#include <mutex>
#include <vector>
#include <fcntl.h>
```

4.2 Vonkajšie zdroje

Kedže POP3 používa v autorizačnej fáze zašifrované heslo pomocou hashovacieho algoritmu MD5, ktorého vlastná implementácia by bola zbytočne zdĺhavá, tak som sa rozhodol použiť voľne dostupné riešenie z internetu[4], ktoré následne prekladám a využívam v aplikácii.

4.3 Detaily implementácie

Aplikácia nie je písana objektovo, ale používa niektoré objekty zo štandardných knižníc C++.

Aby sa nevytvárali nové procesy používa vlákna, ktoré vznikajú pri pripojení nového užívateľa a zanikajú spolu so spojením.

Pri behu programu vzniká jeden pomocný súbor `temp_moves.txt`, ktorý drží informácie o presunoch súborov od prvého spustenia. Tento súbor je následne použitý v prípade resetu aby sa všetko vrátilo do pôvodného stavu, okrem zmazaných súborov. Po resete tento pomocný súbor zaniká a vznikne až pri novom spustení serveru.

Medzi hlavné funkcie patria:

- *int decode_request(int client_socket_fd, char *data, int status, char *timestamp);* - určuje aký príkaz prišiel od klienta
- *int checkPassword(char *data);* - overuje správnosť zadaného hesla
- *int checkUser(char *data);* - overuje správnosť zadaného užívateľského mena
- *int apop_login(char *data, char *timestamp);* - overuje správnosť mena a hesla pri použití šifrovaného hesla
- *int move_new_to_cur();* - po autorizačnej časti presunie súbory zo zložky /new do /cur
- *int update(int client_socket_fd);* - po skončení transakčnej fázy vykonáva žiadané zmeny
- *int send_retr(char *data, int client_socket_fd);* - posiela klientovi zvolenú správu
- *int send_uidl(int number, int client_socket_fd, char *data);* - posiela klientovi odpoveď na dotaz UIDL
- *int send_list(int number, int client_socket_fd, char *data);* - posiela klientovi odpoveď na dotaz LIST

4.4 Rozšírenia

V aplikácii je implementované aj rozšírenie príkazu *TOP n m*. Tento príkaz pošle klientovi *m* riadkov, po hlavičke, zo správy *n*.

5 Použitie aplikácie

Aplikácia slúži ako zjednodušený POP3 server. Umožňuje komunikáciu s viacerými užívateľmi, ale pripojený môže byť stále len jeden. Ak nedostane žiadnu správu od klienta, tak po 10 minútach spojenie s klientom ukončí a uvoľní zdroje. Server svoju prácu ukončí pri zadaní signálu SIGINT.

5.1 Spustenie

Aplikácia sa spúšťa nasledovne:

```
./popser [-h] [-a PATH] [-c] [-p PORT] [-d PATH] [-r]
```

- h* - voliteľný parameter, vypíše nápovedu, ak je zadaný s ostatnými vypíše nápovedu a ukončí server
- a PATH* - povinný pre spustenie serveru, PATH obsahuje cestu do súboru s autorizačnými údajmi
- p PORT* - povinný pre spustenie serveru, PORT je číslo portu, na ktorom bude server pracovať
- d PATH* - povinný pre spustenie serveru, PATH obsahuje cestu do zložky Maildir
- c* - voliteľný, ak je zadaný tak sa ako autorizačná metóda používa USER/PASS, inak APOP
- r* - voliteľný, vráti všetko do stavu pred prvým spustením serveru, ak zadaný s ostatnými, tak sa najprv prevedie reset a následuje normálne fungovanie serveru

6 Záver

Program slúži ako jednoduchý POP3 server na prácu s elektronickou poštou. Prijíma príkazy od klienta, následne ich vyhodnocuje a odpovedá klientovi pomocou využitia neblokujúceho socketu. Preklad prebieha použitím prekladača g++ a na uľahčenie prekladu slúži priložený Makefile. Aplikácia bola vyvíjaná v operačnom systéme Linux, *Ubuntu 16.04*, a následne testovaná na referenčnom stroji *merlin*.

Reference

- [1] HALL, B. *Beej's Guide to Network Programming* [online]. Version 3.0.21. 2016. Dostupné na: <<http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html>>.
- [2] MYERS, J. a ROSE, M. *Post Office Protocol - Version 3* [online]. 1996. Dostupné na: <<https://tools.ietf.org/html/rfc1939>>.
- [3] P. RESNICK, E. *Internet Message Format* [online]. 2008. Dostupné na: <<https://tools.ietf.org/html/rfc5322>>.
- [4] THILO, F. *C++ md5 function* [online]. Dostupné na: <<http://www.zedwood.com/article/cpp-md5-function>>.