

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí
POP3 server

Obsah

1	Úvod	2
2	Dôležité pojmy	2
2.1	POP3	2
2.2	IMF	2
2.3	Maildir	2
3	Návrh a implementácia	3
3.1	Spracovanie argumentov	3
3.2	Pripojenie klientov	3
3.3	Spracovanie príkazov	3
3.4	Ukončenie signálom SIGINT	3
3.5	Použíte knižnice	3
4	Použitie programu	4
5	Záver	4

1 Úvod

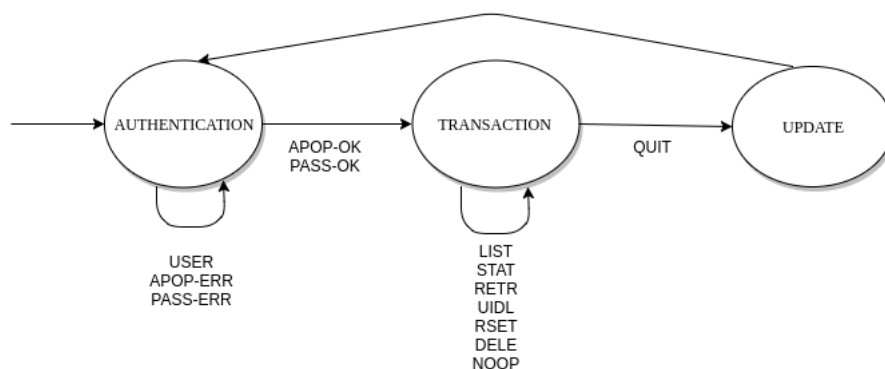
Táto dokumentácia popisuje program s názvom **popser** vytvorený ako projekt do predmetu Sít'ové aplikácie a správa sítí. Úlohou tohoto zadania bolo vytvoriť e-mailový server, ktorý pracuje nad protokolom POP3, spracováva príkazy klientov a poskytuje na ne odpovede. Program má implementovať celú špecifikáciu POP3(okrem sekcie č. 8) a má pracovať s e-mailmi vo formáte IMF. V nasledujúcich častiach sú popísané jednotlivé protokoly, popis riešenia jednotlivých problémov a ich implementácia.

2 Dôležité pojmy

Funkčnosť celej aplikácie je založená na niekoľkých protokoch a štandardoch. K pochopeniu zadania a k vypracovaniu projektu je potrebné ich najprv preštudovať a porozumieť.

2.1 POP3

Najdôležitejším protokolom v rámci celej implemetácie je protokol POP3(Post Office Protocol - Version 3) popísaný v RFC1939 [2]. Slúži na prístup k e-mailom na mailovom serveri ktorý podporuje tento protokol. Neponúka rozsiahle manipulačné operácie, zvyčajne klient sa pripojí k serveru, stiahne si maily a na serveri sa vymažú. Používa sa komunikácia klient-server. Klient po pripojení posielá príkazy serveru, server ich spracuje a poskytne odpoveď. Protokol popisuje presný formát správ a odpovedí. Odpoveď musí začať s +OK pri pozitívnej odpovedi, -ERR pri negatívnej. Pripojenie(session) prechádza niekoľkými stavmi. Obrázok 1 popisuje stavy spojenia a prechody medzi nimi.



Obr. 1: Stavový automat pripojenia POP3.

2.2 IMF

Internet message format je popísaný v RFC5322 [3]. Udáva formát súborov s ktorými bude náš server pracovať. Každý e-mail sa má skladať z hlavičky a tela správy. Každý riadok súboru má byť ukončený s dvojicou znakov \r\n(tj. CRLF).

2.3 Maildir

Formát Maildir [1] popisuje potrebnú adresárovú štruktúru. Priečinko, ktorý bude použitý ako Maildir musí obsahovať adresáre **new**, **cur** a **tmp**, poprípade môže obašhovať nejaké ďalšie súbory alebo adresáre. V rámci tejto implementácie sa používa iba new a cur, keďže s tmp pracujú rôzne SMTP servery. Tiež popisuje že jednotlivé e-mailové súbory musia mať jedinečný názov v rámci jedného Maildiru a premenovanie súborov pri presune z new do cur.

3 Návrh a implementácia

Program je implementovaný v jazyku C/C++ s použitím povolených knižníc potrebných pre vypracovanie zadania. Je rozdelený do dvoch modulov. Nie je použitý objektový návrh ale je vytvorená jedna trieda.

3.1 Spracovanie argumentov

Pre spracovanie argumentov bola vytvorená trieda `Arguments`, ktorá obsahuje privátne premenné pre každý prepínač a jednu metódu na kontrolu argumentov. Na začiatku sa kontroluje či bol zadaný prepínač `-h`. Ďalej sa kontroluje či bol program spustený iba so samotným prepínačom `-r`. Na koniec sa povienné a voliteľné parametre kontrolujú pomocou funkcie `getopts` ktorá vyhodnotí správnosť parametrov a ich hodnôt. Na začiatku `main` sa vytvorí objekt typu `Arguments` a zavolá sa metóda na spracovanie argumentov.

3.2 Pripojenie klientov

Sieťová komunikácia a pripojenie klientov je riešená pomocou BSD socketov. Keďže sa jedná o server ktorý je paralelný, čiže paralelne vybavuje každé pripojenie boli použité vlákna z knižnice `pthread.h`. Keď sa jeden klient pripojí akceptuje sa pripojenie, vytvorí sa socket cez ktorý komunikuje a ďalej jeho požiadavky sú spracované v rámci vlákna. Túto časť popisuje ďalšia kapitola.

3.3 Spracovanie príkazov

3.4 Ukončenie signálom SIGINT

Na zachytenie signálu je použitá funkcia `signal` z knižnice `<csignal>`. Pri zachytení signálu je nastavená globálna premenná, ktorá sa kontroluje v podmienkách cyklu pre prijímanie pripojení vo funkcií `main` a pre komunikáciu s klientom vo funkcií pre vlákno. Táto kontrola je možná kvôli použitiu neblokujúcich socketov, kde sa cyklí. Vlákno ukončí socket na ktorom komunikuje a vhodne sa ukončí. Funkcia `main` čaká na ukončenie každého vlákna. Táto časť je riešená pomocou globálnej premennej, kde sa ukladá aktuálny počet vlákien. Ak sa vytvorí vlákno tak premennú inkrementuje, pri ukončení dekrementuje. Funkcia `main` sleduje túto premennú a ak už nie je aktívne žiadne vlákno tak zatvorí pasívny socket a skončí program.

3.5 Použité knižnice

```
#include <iostream>
#include <cstdlib>
#include <string>
#include <string.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/select.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <ctype.h>
#include <fstream>
```

```
#include <fcntl.h>
#include <openssl/md5.h>
#include <csignal>
#include <locale>
#include <dirent.h>
#include <ctime>
#include <list>
#include <vector>
```

4 Použitie programu

Program má tri režimy behu:

1. `./popser -h`: Vypíše sa nápoveda, program sa správne ukončí.
2. `./popser -r`: Reset, Maildir sa vráti do stavu, ako keby server nebol nikdy spustený, program sa správne ukončí.
3. `./popser -p číslo_portu -a autentifikačný_súbor -d cesta_k_maildiru [-r] [-c]`: Server sa spustí, komunikuje na zadanom porte, pracuje so zadaným Maildirom a načíta užívateľské meno a heslo zo súboru `autentifikačný_súbor`. Pri zadaní parametru `-c` sa povolí autentifikácia typu USER/PASS, inak je povolená autentifikácia pomocou príkazu APOP. Parameter má rovnaký význam ako v príklade č. 2, čiže resetne server ale teraz sa neukončí ale beží ďalej.

5 Záver

Program bol spočiatku vyvíjaný na Ubuntu 16.04.03 LTS, neskôr však radšej som to presunul na školský server Merlin kvôli tomu že tu sa bude testovať aj odovzdaná verzia. Program je preložený prekladačom g++ a Makefile, ktorý je v odovzdaných súboroch.

Literatúra

- [1] Bernstein, D. J.: *Using maildir format*. [Online].
URL <http://cr.yp.to/proto/maildir.html>
- [2] Myers, J.; Mellon, C.; Rose, M.: *Post Office Protocol - Version 3*. 1996, [Online].
URL <http://www.ietf.org/rfc/rfc1939.txt>
- [3] P. Resnick, E.: *Internet Message Format*. 2008, [Online].
URL <http://tools.ietf.org/html/rfc5322>