

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu do předmětu ISA

## POP3 server

19. listopadu 2017

Autor: Filip Března, [xbrezn00@stud.fit.vutbr.cz](mailto:xbrezn00@stud.fit.vutbr.cz)

Fakulta informačních technologií v Brně

Vysoké Učení Technické v Brně

# Obsah

1	Úvod.....	3
2	Důležité pojmy.....	3
2.1	TCP protokol .....	3
2.2	BSD sockety .....	3
2.3	POP (Post Office Protocol).....	4
3	Návrh programu .....	5
4	Implementace .....	5
4.1	POP3 server .....	5
4.2	Rozšíření .....	6
4.3	Šifrování přihlášení pomocí MD5 hashe .....	6
4.4	Použité knihovny .....	6
5	Závěr .....	7

# 1 Úvod

Tato dokumentace, která vznikla k projektu do předmětu Sít'ové aplikace a správa sítí, pojednává o základních principech a implementaci POP3 serveru. Zaměřuje se nejen na popis této poštovní služby, ale především na protokol, který tato služba využívá. V závěru jsou shrnuty dosažené výsledky projektové implementace.

## 2 Důležité pojmy

Je třeba vysvětlit některé důležité pojmy, bez jejichž pochopení není možné navrhnout kýžený POP3 server. Následuje tedy krátký popis protokolu TCP a BSD schránek, jelikož byly využity při implementaci projektu. A následně již podrobnější obecná charakteristika přímo pro mailový Post Office Protocol.

### 2.1 TCP protokol

TCP (Transmission Control Protocol) pracuje na transportní vrstvě TCP/IP modelu, kde vzniká logické spojení mezi procesy. Načítá data z vrstvy aplikační a skládá je do jednotlivých paketů, které se dají chápat jako transportní moduly. TCP nepatří mezi nejrychlejší protokoly svého druhu, ale jeho velkou výhodou a důvodem využívání je schopnost zajistit spolehlivý přenos dat. To je docíleno vytvářením a rušením spojení a opakovaným přenosem ztracených, nedoručených paketů. [3]

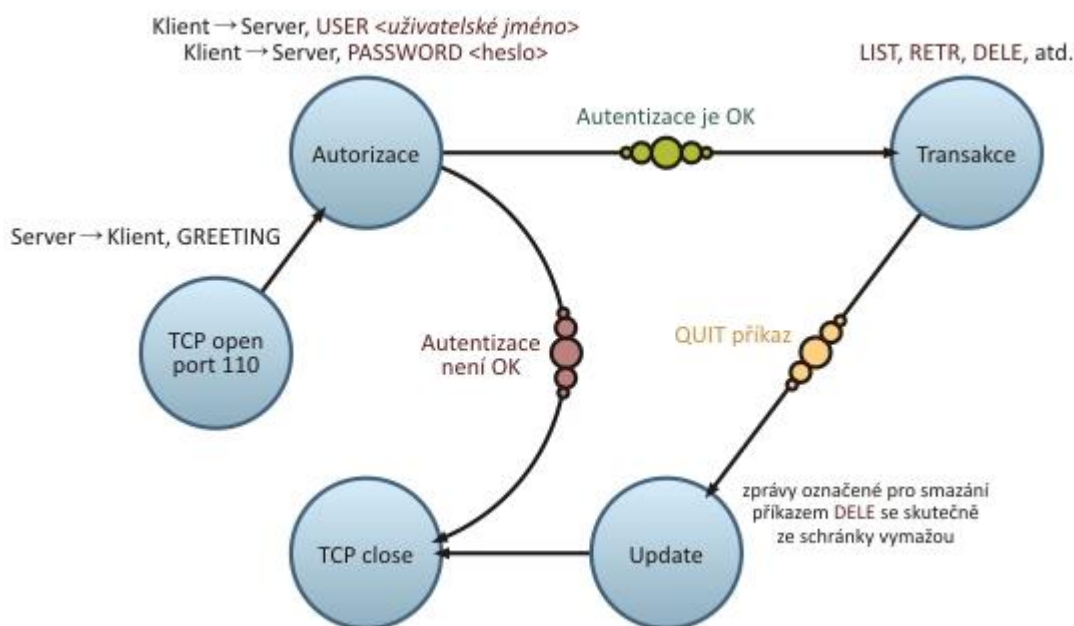
### 2.2 BSD sockety

BSD sockety, často nazývané jako BSD schránky, jsou datová struktura typu *záznam*, která nám umožňuje komunikaci po síti bez nutnosti vytváření nového transportního protokolu. Schránky jsou jednoznačně a unikátně identifikovány čtveřicí (*zdrojový port*, *zdrojová IP adresa*, *cílový port*, *cílová IP adresa*). Je možné je vidět jako přechodovou složku mezi vrstvou aplikační a transportní. Vytváří se a zároveň nastavují funkcí `socket()`. [3]

## 2.3 POP (Post Office Protocol)

I přestože je elektronická pošta ústupem, který zapříčinil tzv. *instant messaging*, může být označena za jednu z nejrozšířenějších internetových služeb, bez níž si mnozí nedokáží síťovou komunikaci představit. Také nelze popřít, že je jedním z důvodů, který odstartoval celosvětové rozšíření internetu.

Post Office Protocol s poslední nejrozšířenější verzí 3 označován jako POP3, slouží ke stahování e-mailových zpráv ze serveru na klienta. Standardně pracuje na TCP portu 110. Nevýhodou POP3 je, že značnou režii musí provádět sám klient, podstatná část práce a změn se tedy neděje přímo na straně serveru jako je tomu u protokolu IMAP. Jedná se však v některých situacích o dostačující jednodušší protokol, který umožňuje zabezpečení přihlášení pomocí MD5 hashe s využitím příkazu APOP.



Obrázek 1: Konečný automat protokolu POP3.

Zdroj: <http://docplayer.cz/1025735-Protokoly-aplikacni-vrstvy-dns-smtp-http-leos-bohac.html>

## 3 Návrh programu

POP3 server pracuje ve třech režimech, které popisuje konečný automat v obrázku 1. Prvním režimem je autorizace, která přijímá příkazy USER následovaný příkazem PASS pro nešifrovanou komunikaci a nebo příkaz APOP pro komunikaci šifrovanou. Pokud autentizace proběhne v pořádku, přechází se do stavu s názvem transakční, ten slouží pro řadu příkazů jako například RETR, RSET, UIDL, DELE, STAT atd. definovaných v RFC.[1] Proces končí pokud klient zadá příkaz QUIT, tehdy se přechází do stavu update, kde dojde k smazání zpráv označených značkou pro smazání a odpojení klienta.

## 4 Implementace

Program byl vytvořen v jazyce C++ se standardem C++14. Aplikace je vytvořena a cílena na operační systém Linux, na kterém byla i testována. Testy proběhly na školním serveru Merlin, který byl v zadání uveden jako referenční stroj, ale také na Ubuntu verze 14.04.5.

### 4.1 POP3 server

Běh programu začíná typicky ve funkci `main()`, následuje kontrola vstupních argumentů, ta se provádí ve funkci `checkArguments()`. Pokud je vše v pořádku, provede se instanciací třídy `pop3_server`, umístěné v hlavičkovém souboru `popser_class.h`. Tím vytvoříme objekt, s kterým budeme následně pracovat. Nyní následuje kontrola zda byl zadán parametr `-r`, pokud ano, zavolá se funkce `resetParameter()`, jejíž definice, stejně jako definice téměř každé následující funkce je uvedena v třídě `pop3_server`. V případě, že byl parametr `-r` zadán samostatně, program končí. V opačném případě se volá funkce `newToCur()`, která přesune mailové zprávy ze složky `new` do složky `cur`, umístěné v mailovém adresáři. Poté se vytvoří BSD schránky a je provedena režie kolem nich a čeká se na připojení klienta, které je provázáno odesláním uvítací hlášky s časovým razítkem a vytvářením vlákna jeho obsluhou ve funkci `clientCommunicationHandler()`, která řeší komunikaci mezi serverem a volá si k tomu pomocné funkce definované třídě `pop3_server`.

V případě, že se klient odpojí, dojde k uzavření schránek, s kterými pracoval a vlákno, které využíval také končí. V případě ukončení serveru signálem `SIGINT`, dojde k zavření všech otevřených socketů a vláken, stará se o to funkce `SIGINT_handler()`.

## 4.2 Rozšíření

Nepovinné rozšíření ve formě příkazu *TOP* bylo implementováno, a tedy může být součástí hodnocení. Tento příkaz umožňuje zasílání pouze hlavičky zprávy určené klientovi. Dalším parametrem příkazu je pak počet řádků obsahu, které chce klient také obdržet. V případě volby většího počtu řádků než je jejich maximální počet, pošle se celá zpráva.

## 4.3 Šifrování přihlášení pomocí MD5 hashe

Tvorba hashovacího algoritmu nebyla součástí požadované práce a MD5 hash byl převzat z veřejně dostupného zdroje od Franka Thila, za který jsem mu velmi vděčný. Prohlašuji tedy, že nejsem autorem této části práce a zdroj čerpání bude uveden v referencích v závěrečné kapitole.[2]

## 4.4 Použité knihovny

<code>#include &lt;stdint.h&gt;</code>	<code>#include &lt;locale&gt;</code>
<code>#include &lt;string.h&gt;</code>	<code>#include &lt;iostream&gt;</code>
<code>#include &lt;dirent.h&gt;</code>	<code>#include &lt;unistd.h&gt;</code>
<code>#include &lt;mutex&gt;</code>	<code>#include &lt;sys/socket.h&gt;</code>
<code>#include &lt;vector&gt;</code>	<code>#include &lt;arpa/inet.h&gt;</code>
<code>#include &lt;iostream&gt;</code>	<code>#include &lt;netinet/in.h&gt;</code>
<code>#include &lt;stdio.h&gt;</code>	<code>#include &lt;signal.h&gt;</code>
<code>#include &lt;stdlib.h&gt;</code>	<code>#include &lt;time.h&gt;</code>
<code>#include &lt;sys/stat.h&gt;</code>	<code>#include &lt;thread&gt;</code>
<code>#include &lt;sys/types.h&gt;</code>	<code>#include &lt;fcntl.h&gt;</code>
<code>#include &lt;algorithm&gt;</code>	

## 5 Závěr

POP3 server byl implementován včetně rozšíření a úspěšně přeložen i otestován na referenčním stroji Merlin. Program splňuje požadavky pro komunikaci s klientem a může být tedy za tímto účelem využíván.

## Metriky kódu

Počet souborů: 6

Počet řádků zdrojového textu pro POP3 server (vyjma MD5 hashe): 1853

Velikost spustitelného souboru: 495129 Bytů

## Reference

[1] **MYERS, J.** Post Office Protocol - Version 3. [Online]

Dostupné z: <https://tools.ietf.org/html/rfc1939>.

[2] **THILO, Frank.** C++ md5 function. [Online]

Dostupné z: <http://www.zedwood.com/article/cpp-md5-function>.

[3] **MATOUŠEK, Petr.** *Studijní opora předmětu Sítové aplikace a správa sítí*

*Kapitola 2 Programování sítí TCP/IP* [Online] Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FISA-IT%2Ftexts%2Fkapitola2.pdf&cid=12191>