



Síťové aplikace a správa sítí
Programování síťové služby
Klient SMTP pre odosielanie pošty

17. listopadu 2015

Autor: Jakub Pastuszek, xpastu00@stud.fit.vutbr.cz
Fakulta Informačních Technologii
Vysoké Učení Technické v Brně

Úvod

Cílen projektu je navrhnout konzolovou aplikaci jednoduchého SMTP klienta. Implementace aplikace musí být dle IETF standardu RFC 2821 [RFC2821]. Připojení na server probíhá prostřednictvím nešifrované komunikace. Program generuje výstup pouze v případě výskytu chyb(y).

Zadání

Vytvorte konzolovú aplikáciu implementujúcu jednoduchého SMTP klienta, ktorý je schopný pripojiť sa na SMTP server a následne odoslať poštu.

Návrh aplikace

Implementačním jazykem mého projektu je jazyk C++ s podporou objektového přístupu. Aplikace je rozdělena do několika samostatných objektů a modulů. Jeden objekt se stará o otevření soketu, další o práci se souborem a poslední, ale nejdůležitější objekt je pro práci s celým klientem, jako zasílání a přijímání zpráv do/od serveru, nebo výpis chybových hlášek a zpracování argumentů. Jako jediný modul se vyskytuje modul pro zpracování signálu, které může uživatel aplikaci zaslat k jejímu „násilnému“ ukončení, a které musí být náležitě aplikačně ošetřeny, aby bylo spojení šetrně ukončeno.

Popis implementace

SOCKET

Objekt „Socket“ se stará o otevření a také uzavření soketů. Sokety se mohou otevírat pro komunikaci na L3 vrstvě (síťová vrstva) TCP/IP¹ jak pro IPv4, tak i IPv6. Dále na L4 vrstvě (transportní vrstva) aplikace používá TCP protokol² pro spolehlivé doručování.

FILE

„File“ objekt je vytvořen pro práci se soubory obecně. V této aplikaci je použit pro otevření a postupně čtení vstupního souboru zadaného parametrem (viz [Návod na použití](#)).

CLIENT

Nejdůležitějším objektem je objekt „Client“, který se stará o celý chod aplikace. Nejprve jsou zpracovávány argumenty aplikace, kdy se kontroluje jejich úplnost a nastavují požadovaná data jako je IP adresa serveru, port, vstupní soubor, či délka uměle udržovaného otevřeného spojení. Dále je zpracován obsah souboru. V následujícím kroku je otevřen soket a nastaven port, toto již stačí pro připojení se k serveru, které je následně provedeno.

Po připojení se server „ohlašuje“, že může komunikovat a odesílá zprávu klientovi s kódem 220. V případě nenavázání spojení posílá 554 a aplikace se ukončuje. Dále již nastává komunikace se serverem řízená dle RFC 2821 [RFC2821].

EHLO

Nejprve klient zasílá Extended HELLO zprávu serveru (EHLO), která je ve tvaru (1) dle sekce 4.1.1.1 a server následně odpovídá zprávou s kódem 250, jako HELLO. V případě chyby se uzavírá spojení a ukončuje se aplikace. Dále server může poslat další upřesňující zprávy. Mezi ně patří např. PIPELINING, které aplikaci povolují řetězit zprávy za sebou – aplikace nemusí každou zprávou(řádek) posílat samostatně a čekat na potvrzení, nebo zpráva SIZE, která udává maximální délku zprávy, kterou může server přijmout, a další rozšíření (viz [RFC2821]).

MAIL

Dalším příkazem je MAIL FROM sekce 4.1.1.2, který udává z jaké emailové adresy budou zprávy odesílány (2). Odpověď od serveru je s kódem 250, pokud je odesílatel potvrzen. Když nastane chyba je zavolán RSET (5) pro obnovení výchozího stavu (stav viz níže).

RCPT

Příkaz RCPT TO je určen tomu, aby vytvořil jednotlivé příjemce zprávy (3), sekce 4.1.1.3. V případě nalezení příjemce odpovídá zprávou s kódem 250, v opačném případě s kódem 550, kde se chyba zaznamenává. Při jiných chybách není chyba zaznamenána, ale je zpracována.

¹ <https://cs.wikipedia.org/wiki/TCP/IP>

² https://cs.wikipedia.org/wiki/Transmission_Control_Protocol

DATA

Následujícím příkazem DATA řekneme serveru, že již další příjemci pošty nebudou a dále se bude odesílat obsah zprávy (4) dle sekce 4.1.1.4, na tuto zprávu server odpovídá zprávou s kódem 354 a požadovaným formátem zasílané zprávy. Zpráva musí být ukončená následujícím řetězcem v uvozovkách: „<CRLF>.<CRLF>“ (6).

Pokud není nalezen žádný platný příjemce, server zašle zprávu s kódem 554 a klient musí poslat zprávu RSET (5) sekce 4.1.1.5 a celé odesílání stornovat. Tato zpráva také musí být potvrzena serverem. Zpráva RSET může být serveru zaslána kdykoliv pro požadavek na zrušení dané zprávy, příjemců i odesílatele. Ještě než se pošle zpráva DATA, musí být ověřeno zda odesílaná zpráva nepřesahuje velikostní limit daný serverem, pokud ano, odesílání se zruší zprávou RSET.

Tento celý proces až na HELLO zprávu se může neomezeně opakovat, a to pro odesílání více zpráv od různých (nebo i stejného) odesílatele různým příjemcům.

QUIT

Poslední zprávou, kterou klient serveru zasílá je QUIT (7) sekce 4.1.1.8.

Po odeslání všech zpráv (vyčerpání všech řádků souboru) se klient ukončí. Ještě před ukončením musí proběhnout určité akce jako umělé udržování otevřeného spojení a následné ukončení spojení a uzavření socketu. Udržování otevřeného spojení probíhá pomocí uspání aplikace na dobu danou parametrem „sekund“, pokud je tento parametr větší než 300, tak se po každých uplynutých 5-ti minutách posílá serveru zpráva NOOP 4.1.1.9, která způsobí, že server s klientem neukončí spojení.

Pokud byla aplikace ukončená „násilně“ pomocí signálu zaslaného aplikaci, se aplikace řádně ukončí a to tak, že nejprve uzavře spojení se serverem. Avšak, pokud byla již na server odeslána zpráva DATA, musí se dokončit odesílání již započatého emailu a až pak může klient ukončit spojení. V případě ukončení aplikace pomocí signálu se neprovádí nastavené umělé udržování otevřeného spojení. Dále v případě výskytu chyb jako např. špatný příjemce, žádný příjemce aj. se chyby vypíší na standardní chybový výstup.

Příklady komunikace: zprávy odeslané klientem

[a] Správné odeslání zprávy

- (1) EHLO isa.local
- (2) MAIL FROM:<xpastu00@isa.local>
- (3) RCPT TO:<etela@isa.local>
- (4) DATA
- (6) Useful data<CRLF>.<CRLF>
- (7) QUIT

[b] Žádný platný příjemce zprávy

- (1) EHLO isa.local
- (2) MAIL FROM:<xpastu00@isa.local>
- (3) RCPT TO:<nikdo@isa.local>
- (4) DATA
- (5) RSET

[a] – Server na všechny zprávy odpověděl zprávou s kódem 250(OK) kromě zprávy DATA (4), na kterou odpověděl kódem 354

[b] – Server na zprávy EHLO (1) a MAIL FROM (2) odpověděl zprávou s kódem 250(OK), na RCPT TO (3) s neplatným příjemcem odpověděl 550 – uživatel nenalezen a následně po zaslání DATA (4) odpověděl 554 – žádný platný příjemce. Poté musel klient zrušit odesílání emailové zprávy zasláním RSET (5)

Návod na použití

./smtpklient [-a IP] [-p port] [-i soubor] [-w sekund] [-h]

- IP** - (nepovinný, výchozí hodnota je 127.0.0.1) IP adresa SMTP serveru (možnost zadat IPv4 i IPv6 adresu)
- port** - (nepovinný, výchozí hodnota je 25) port, na kterém SMTP server očekává přicházející spojení
- soubor** - (povinný parameter) cesta k souboru, ve kterém sa nacházejí zprávy pro odeslání
- sekund** - (nepovinný parametr, výchozí hodnota 0) po odeslání poslední zprávy se neukončí spojení okamžitě, ale klient bude uměle udržovat spojení otevřené po dobu specifikovanou tímto parametrem. Nejvyšší hodnota, kterou je možné zadat je jedna hodina.
- h** - vypíše nápovědu

Literatura

- [RFC2821] AT&T Laboratories, „Simple Mail Transfer Protocol“, RFC 2821, April 2001.
- [RFC1870] J. Klensin, WG Chair, „SMTP Service Extension for Message Size Declaration“, RFC 1870, November 1995
- [projekt] Ing. Holkovič [online], „Zadání projektu“, October 2015, Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/course-sl.php?id=0&item=54117&cpa=1>