



Dokumentace k projektu předmětu ISA

IRC bot s logováním přes SYSLOG protokol

30. listopadu 2014

Autor: Jan Havran, xhavra13@stud.fit.vutbr.cz

Obsah

| | | |
|----------|-------------------------------|----------|
| 1 | Úvod | 1 |
| 2 | Analýza | 2 |
| 2.1 | Zadání problému | 2 |
| 2.2 | IRC | 2 |
| 2.3 | SYSLOG | 3 |
| 3 | Návrh řešení | 3 |
| 3.1 | Zpracování zpráv | 3 |
| 3.1.1 | Využití vláken | 4 |
| 3.1.2 | Fronty | 4 |
| 3.2 | Odpojení od serveru | 4 |
| 4 | Ovládání aplikace | 4 |
| 4.1 | Vstupní parametry | 5 |
| 4.2 | Ukončení programu | 5 |
| 5 | Závěr | 6 |

1 Úvod

Tato práce se zabývá návrhem a implementací jednoduchého IRC robota, který pomocí SYSLOG protokolu loguje zprávy obsahující klíčová slova. Chování IRC robota lze předem ovlivnit předanými parametry programu.

Popis problému je rozvržen do několika částí. V kapitole 2 jsou popsány potřebné informace k použitým síťovým protokolům, tj. IRC a SYSLOG. Kapitola 3 se zabývá samotným návrhem aplikace, jako je zpracování zpráv, využití možných prostředků atd. Práce s programem, jako spuštění a ukončení, je popsána v kapitole 4.

2 Analýza

Tato kapitola se podrobně zabývá analýzou zadání, teoretickým rozбором komunikace mezi serverem a klientem a protokoly IRC ¹ a SYSLOG ².

2.1 Zadání problému

Cílem tohoto projektu je vytvoření aplikace v jazyce C, který s využitím BSD socketů bude komunikovat s IRC serverem na zadaném kanálu a pomocí SYSLOG protokolu bude dané zprávy logovat na příslušný server. Aplikace musí být spustitelná na Unixových systémech, minimálně na Linuxových distribucích.

2.2 IRC

IRC je jednoduchý protokol na TCP, který slouží ke komunikaci v chatovacích místnostech (channels). Zprávy jsou zasílány v textové podobě a jejich forma je velmi jednoduchá. Maximální délka jedné zprávy je 512 bytů[1].

IRC zpráva se může sestávat z několika částí. Validní zpráva má následující formát:
[:<prefix>]<command>[<params>]<CRLF>

- prefix
Zpravidla obsahuje doplňující informace ke zprávě. Volitelný parametr
- command
Příkaz/požadavek.
- params
Parametry příkazu. Poslední parametr může začínat znakem ':' a může obsahovat více slov. Maximální počet parametrů je 15.

Následuje seznam základních příkazů, které bude implementovaný program ovládat (syntaxe převzata z [1]):

- NICK <nickname>
Nastavení nebo změna přezdívky, **nickname**, na serveru.
- USER <username> <hostname> <servername> <realname>
Slouží k nastavení **username**, **hostname**, **servername** a **realname** (tento musí být poslední, jelikož může obsahovat i mezery) na serveru.
- JOIN <channel>
Připojení ke kanálu **channel**. Název kanálu včetně úvodního znaku #.
- PING <server>
Test spojení, zasílá **server**.
- PONG <daemon>
Odpověď na test spojení.
- PRIVMSG <receiver> <text>
Zaslání soukromé zprávy. Příjemce **receiver** může být jak uživatel, tak kanál serveru.

¹Internet Relay Chat, protokol je popsán v RFC 1459

²BSD syslog protokol je popsán v RFC 3164

- NOTICE `<nickname> <text>`
Podobně jako `PRIVMSG`, ale s tím rozdílem, že klient/server neodesílá odpovědi. Předchází se tak zacyklení aplikace.
- QUIT `[<message>]`
Odpojení od serveru. Lze zaslat i ukončovací zprávičku `message`.

2.3 SYSLOG

K logování je využit SYSLOG, který, narozdíl od IRC, využívá UDP protokol. Celková délka zprávy nesmí přesáhnout 1024 bytů[2]. Jedná se opět o jednoduchý textový protokol, který se také skládá z několika částí. Jsou to:

- PRI
Priorita (PRI) je ohraničena ostrými závorkami `'('` a `')'` a je vypočítáno jako `FACILITY * 8 + SEVERITY`. PRI se sestává ze tří až pěti znaků (včetně hraničních závorek).
- HEADER
Hlavička se skládá z časové značky - `TIMESTAMP` - a IP adresy nebo jména zařízení. Čas musí být ve formátu `'Mmm dd hh:mm:ss'`.
- MSG
Také MSG se skládá ze dvou částí: z části `TAG` (obsahuje jméno programu) a části `CONTENT` (obsah zprávy).

Dle zadání bude mít `CONTENT` podobu:

`<NICKNAME>: MESSAGE`

kde `NICKNAME` je nick odesílatele a `MESSAGE` tělo zprávy.

3 Návrh řešení

V této kapitole bude popsán návrh řešení kritických částí aplikace, jako je odpojení od serveru, zpracování zpráv nebo možnost využití více vláken.

3.1 Zpracování zpráv

Zpracování příchozích a odchozích je stěžejní část aplikace. Při procesu přijetí zpráv nastává několik komplikací. Zpráva může obsahovat několik různých parametrů, z toho jsou navíc některé volitelné. Dále není nikterak zaručeno, že jeden packet bude obsahovat právě jednu zprávu.

Pro přijetí zpráv byl zvolen konečný automat, díky němuž se celý proces zpracování značně zjednoduší. Lze tak relativně snadno odhalit špatný formát zpráv, zpracovat jednotlivé parametry příkazů a zkompletovat přijmuté packety do jednotlivých zpráv.

Aby aplikace mohla pružně reagovat na různé podněty (žádost o ukončení aplikace, přijímání a odesílání zpráv), byly některé činnosti rozděleny do samostatných vláken. Pro sdílení prostředků mezi vlákny aplikace byly využity globální struktury `front`.

3.1.1 Využití vláken

Síťové aplikace pro využití vláken přímo vybízí. V této aplikaci byla využita standardní Unixová vlákna `pthread`s. Po inicializaci programu a připojení se k serveru IRC a SYSLOG jsou vytvořena tři nová vlákna:

- Zapisující vlákno IRC
Jedná se o vlákno které čte ze sdílené fronty. Pokud je fronta prázdná, tak se čeká na signál oznamující naplnění fronty. Podly typu zprávy ve frontě buď zašle zprávu na IRC server, nebo se ukončí.
- Čtecí vlákno IRC
Toto vlákno čte zprávy z IRC serveru. Pokud je přijata nějaká zpráva, tak ji zpracuje a případně na ni reaguje. Pokud například přijde zpráva `PING`, tak naplní frontu pro zapisující IRC vlákno odpovídající zprávou `PONG`. Pokud přijde některá ze zpráv `NOTICE` nebo `PRIVMSG` a taková zpráva obsahuje hledaný řetězec, tak je naplněna fronta pro SYSLOG a následně je signalizováno naplnění této fronty. V případě chyby na socketu (odpojení), je vlákno ukončeno a ostatní vlákna jsou rovněž požádány o ukončení.
- Zapisující vlákno SYSLOG
Podobně jako zapisující IRC vlákno pracuje i vlákno pro odesílání dat na SYSLOG server. Je zde taktéž využita sdílená fronta zpráv.

3.1.2 Fronty

IRC robot využívá dva typy front. První typ fronty, popsany a vysvětleny v [3], slouží pro uchování hledaných řetězců (zadaných přes parametr programu, viz kapitola 4). Tato fronta nemůže být sdílena mezi více vlákny, neboť neobsahuje žádné synchronizační mechanismy. Proto byla vytvořena ještě jiná fronta, která tyto mechanismy obsahuje.

Fronta byla doplněna o mutexy sloužící pro synchronizaci zápisu dat mezi vlákny, a synchronizační zařízení `condition`. Před každou operací (naplnění fronty, vyprázdnění, atd.) je uzamčen mutex, a po provedení operace je opětovně odemčen. Po naplnění fronty (`QThreadUp`) novými daty je zaslán synchronizační signál. Tento signál je využit při vyprazdňování fronty (`QThreadFront`), kdy v případě prázdné fronty se čeká právě na tento signál.

3.2 Odpojení od serveru

Další důležitou událostí, kterou je nutné brát v potaz, je odpojení od serveru. To může nastat z několika příčin. Může se tak rozhodnout IRC server (klient například neodpovídá na `ping`), nebo také může být příčinou chyba spojení (například odpojení síťového kabelu). V obou případech chybu zaznamená čtecí vlákno IRC pomocí funkce `recv`. Pokud chyba opravdu nastane, tak příslušné vlákno pomocí sdílených front informuje ostatní vlákna o chybě a samo se ukončí. Ještě před ukončením logovacího vlákna je o chybě informován příslušný SYSLOG server.

4 Ovládání aplikace

Tato kapitola popisuje spuštění a ukončení aplikace. Jediným vstupem pro tohoto IRC robota jsou parametry z příkazové řádky. Pozdější předávání dat tedy není možné. Stejně tak není možné robota ukončit příkazem, jelikož ignoruje standardní vstup.

4.1 Vstupní parametry

Aplikaci lze předat hned několik parametrů. Pomocí nich je možné/nutné určit, na který IRC a SYSLOG server se připojit a do jakého kanálu vstoupit. Parametry pro spuštění aplikace: `ircbot <host>[:<port>] <channel> <syslog_server> [list]`

Význam parametrů je následující:

| | |
|----------------------------|-----------------------------------------|
| <code>host</code> | IRC server |
| <code>port</code> | číslo portu (default 6667) |
| <code>channel</code> | název IRC kanálu |
| <code>syslog_server</code> | SYSLOG server |
| <code>list</code> | seznam sledovaných slov (oddělovač ';') |

Příklad spuštění:

```
ircbot irc.freenode.net "#testfit" 127.0.0.1 "fit vutbr;lan;wan"
```

A nebo pomocí Makefile, s předpřipravenými parametry:

```
make run
```

4.2 Ukončení programu

Aplikace se ukončí v případě chyby nebo odpojení ze serveru. Jelikož aplikace nečte ze standardního vstupu (ani odjinud), tak je nutné aplikaci ukončit jinak. Nenásilnou formou je například zaslání **SIGINT** pomocí kláves `ctrl+c` (doporučeno). Aplikace odchytává právě tento signál. Po odchycení se odpojí od IRC serveru, informuje SYSLOG server a nenásilně se ukončí.

5 Závěr

Implementovaný IRC robot má funkcionalitu dle zadání. Při řešení komunikace s IRC a SYSLOG servery byly využity informace dostupné z příslušných RFC. Aplikace neobsahuje kompletní implementaci zmíněných protokolů, nýbrž jen základní příkazy postačující ke splnění zadání.

Aplikace byla otestována na 64 bitových Linuxových distribucích Debian, LinuxMint a referenčním virtuálním stroji s Ubuntu.

Reference

- [1] The Internet Engineering Task Force/RFC 1459. [Online]
<http://tools.ietf.org/html/rfc1459>
- [2] The Internet Engineering Task Force/RFC 3164. [Online]
<http://tools.ietf.org/html/rfc3164>
- [3] HONZÍK, Jan M.: *Algoritmy IAL: Studijní opora*. Fakulta informačních technologií VUT v Brně, 2012-09-16.