VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

IRC BOT S LOGOVÁNÍM SYSLOG

IRC BOT WITH SYSLOG

PROJEKOTVÁ DOKUMENTACE
PROJECT DOCUMENTATION

AUTOR PRÁCE **KATARÍNA GREŠOVÁ**
AUTHOR

BRNO 2017

Obsah

Kapitola 1

Úvod

Táto správa je dokumentáciou k projektu v predmete Sieťové aplikácie a správa sietí. Cieľom projektu bolo vytvoriť jednoduchého IRC bota s logovaním SYSLOG. V nasledujúcich kapitolách je stručný úvod do problematiky a následný popis samotnej implementácie.

Kapitola 2

Úvod do problematiky

Pre správne pochopenie konkrétnej implementácie je potrebné zoznámiť sa s teoretickými základmi, na ktorých bola postavená.

2.1 Internen Relay Chat protokol

Špecifikácia IRC protokolu, podľa ktorej sme pracovali, je uvedená v RFC 1459 [2]. IRC protokol je textovým protokolom, kde najjednoduchším klientom môže byť ľubovolný schránkový (en. socket) program, ktorý je schopný pripojiť sa na server.

Server poskytuje miesto, kde sa klienti môžu pripojiť a komunikovať medzi sebou, a miesto pre pripojenie ďalších serverov. Klientom je všetko propojené k serveru, čo nie je ďalší server.

Významnou časťou IRC sú kanály. Kanál je pomenovaná skupina jedného alebo viacerých klientov, kde všetci účastníci prijímajú správy adresované danému kanálu.

Servery a klienti komunikujú pomocou textových správ s predpísaným formátom, ktorého možný zápis pomocou ‘pseudo‘ BNF je na obrázku 2.1

```
<message>  ::= [':' <prefix> <SPACE> ] <command> <params> <crlf>
<prefix>   ::= <servername> | <nick> [ '!' <user> ] [ '@' <host> ]
<command>  ::= <letter> { <letter> } | <number> <number> <number>
<SPACE>    ::= ' ' { ' ' }
<params>   ::= <SPACE> [ ':' <trailing> | <middle> <params> ]

<middle>   ::= <Any *non-empty* sequence of octets not including SPACE
               or NUL or CR or LF, the first of which may not be ':'>
<trailing> ::= <Any, possibly *empty*, sequence of octets not including
                 NUL or CR or LF>

<crlf>     ::= CR LF
```

Obr. 2.1: ’pseudo’ BNF formátu komunikácie

2.2 BSD syslog protokol

Špecifikácia BSD syslog protokolu, podľa ktorej sme pracovali, je uvedená v RFC 3164 [1].Syslog protokol poskytuje možnosť posielať správy o udalostiach prostredníctvom IP siete na syslog servery. Hlavným princípom tohto protokolu je jeho jednoduchosť. Posielanie správy môže začať bez toho, aby prijímajúca strana bola nakonfigurovaná. Každý obsah IP paketu, ktorého cieľový UDP port je 514, je považovaný za syslog správu.

Celková dĺžka paketu musí byť nanajvýš 1024 bajtov. Formát správy má tri rozpoznateľné časti: PRI, HEADER a MSG. Časť PRI popisuju prioritu. Časť HEADER obsahuje časovú značku a adresu odosielateľa. Časť MSG obsahuje dodatočné informácie o procese, ktorý správu vygeneroval, a samotnú správu.

Kapitola 3

Implementácia

V tejto časti je popísaná konkrétna implementácia zadania. Program je implementovaný v jazyku C/C++.

3.1 Použité štruktúry

Na jednoduchšiu a prehladnejšiu prácu boli v implementácii definované dve štruktúry. *ParsedInput*, ktorá slúži na uloženie vstupných parametrov a *ParsedImg*, ktorá slúži na uloženie správy prijatej od serveru.

3.2 Komunikácia s IRC serverom

Naviazanie spojenia s IRC serverom prebieha vo funkcii *connectTo()*, kde sa vytvorí TCP spojenie s aplikáciou na zadanej adrese a porte. Samotná komunikácia so serverom prebieha vo funkcii *talkTo()*. Táto funkcia pošle IRC serveru správy NICK a USER, ktoré slúžia na registráciu klienta. Následná komunikácia je riešená pomocou cyklu *while*, ktorý je aktívny pokiaľ server posiela dáta. V rámci cyklu sa prijaté dáta rozdelia na jednotlivé správy (oddelené CRLF), a správa je rozparsovaná vo funkcii *parseLine()* na časti, ktoré sú uložené do štruktúry *ParsedMsg*. Podľa položky *command* je určené ďalšie správanie programu.

3.2.1 Príkaz 001

Číselný príkaz 001 s názvom `LIBIRC_RFC_RPL_WELCOME` je prívítanie registrovaného klienta. Po prijatí tohto príkazu program odosiela správu JOIN so zoznamom kanálov, ktoré boli zadané pri spustení programu.

3.2.2 Príkaz 353

Číselný príkaz 353 s názvom `RPL_NAMREPLY` obsahuje zoznam užívateľov, ktorý sa nachádzajú na určitom kanále. Takáto správa je prijatá po pripojení sa na daný kanál.

Informácie získané z tejto správy sú použité na prvotné naplnenie premennej *map<string, vector<string>> users*, ktorá udržuje aktuálne informácie o užívateľoch a o kanáloch, na ktoré sú prihlásení.

3.2.3 Príkazy začínajúce číslicou 4 alebo 5

Číselné príkazy začínajúce číslicou 4 alebo 5 oznamujú rôzne chyby. V prípade prijatia takejto správy, program odošle na IRC server QUIT správu a ukončí sa.

3.2.4 Príkaz PING

Príkaz PING overuje prítomnosť aktívneho klienta. Na túto správu program odpovedá správou s príkazom PONG, inak by spojenie bolo ukončené.

3.2.5 Príkazy NOTICE a PRIVMSG

Správy s príkazmi NOTICE a PRIVMSG obsahujú text určený užívateľovi alebo kanálu. Tento text je v programe kontrolovaný na prítomnosť kľúčových slov zadaných pri spustení. Ak sa ľubovolné kľúčové slovo nachádza v texte správy, je zavolaná funkcia *sendSyslog()*.

Ďalej je v texte správy s príkazom NOTICE hľadané slovo ERROR, ktoré značí, server preposiela nejaké chybové hlásenie. V takom prípade je program korektne ukončený.

V texte správy s príkazom PRIVMSG sa hľadajú kľúčové slová funkcií *?today* a *?msg*. V prípade prítomnosti kľúčového slova funkcie sú volané funkcie, ktoré implementujú požadovanú funkcionalitu (*handleTodayFunction()* a *handleMsgFunction()*). Program nehľadá kľúčové slová funkcií v správach s príkazom NOTICE, pretože dané funkcie posielajú nejakú správu na server a podľa špecifikácie sa na správy s príkazom NOTICE nemá odpovedať automaticky.

3.2.6 Príkaz JOIN

Správa s týmto príkazom oznamuje pripojenie sa užívateľa na kanál, prípadne na viacero kanálov. Na správne fungovanie funkcie *?msg* musí program túto informáciu zohľadniť. Je volaná funkcia *handleJoin()*, ktorá aktualizuje zoznam užívateľov a kanálov, na ktorých sú prihlásení. Ďalej táto funkcia skontroluje, či nie je uložená správa pre práve pripojeného užívateľa a prípadne ju odošle pomocou funkcie *sendBacklog()*.

3.2.7 Príkaz PART

Správa s príkazom PART oznamuje, že odosielajúci užívateľ sa odhlasuje z uvedených kanálov. Pre správne fungovanie funkcie *?msg* je túto informáciu zohľadniť a zo zoznamu kanálov užívateľa odstrániť tie uvedené v správe.

3.2.8 Príkaz KICK

Pomocou správy s príkazom KICK je možné násilne odhlásiť užívateľov z kanálov. Pre správne fungovanie funkcie *?msg*, program túto informáciu zohľadní a svoj zoznam užívateľov a ich kanálov aktualizuje.

3.2.9 Príkaz QUIT

Správa s príkazom QUIT oznamuje, že užívateľ ukončil spojenie s IRC serverom.

Program skontroluje, či správa neobsahuje jeho *nickname*, v tom prípade ukončí svoj beh, inak vo svojich záznamoch odstráni zo všetkých kanálov užívateľa, o ktorom správa hovorí.

3.2.10 Príkaz KILL

Príkaz KILL je použitý vtedy, keď sa objaví duplikátny záznam o užívateľskom mene, a je použitý na odstránenie oboch záznamov.

Obsluha tohto príkazu je zhodná s obsluhou príkazu QUIT.

3.2.11 Príkaz NICK

IRC protokol dovoľuje užívateľom meniť svoje užívateľské mená pomocou príkazu NICK. Po prijatí správy s týmto príkazom, program zavolá funkciu *handleNick()*, ktorá zmení meno užívateľa v ich zozname uloženom v programe.

3.3 Komunikácia so SYSLOG serverom

Komunikácia so SYSLOG serverom prebieha pomocou UDP s využitím schránok typu `SOCK_DGRAM`. Špecifikácia udáva, že správy zasielané na SYSLOG server nemajú prekročiť dĺžku 1024 bytov, čo však nebolo nutné explicitne kontrolovať, pretože posielame časti správ z IRC komunikácie, ktorá nepresahuje dĺžku 512 bytov. Úplná syslog správa obsahuje tri odlíšiteľné časti: PRI, HEADER a MSG.

3.3.1 Časť PRI

Táto časť obsahuje v ostrých zátvorkách číselnú hodnotu, ktorá reprezentuje hodnoty *Facility* a *Severity*. Zadanie špecifikuje, aby *Facility* bolo *local0* a aby *Severity* bolo *Informational*. Celková priorita sa vypočíta tak, že hodnota *Facilita* (*local0* odpovedá hodnota 16) sa vynásobí číslom 8 a k výsledku sa pripočíta hodnota *Severity* (*Informational* odpovedá hodnota 6). Podľa tohto postupu získame obsah časti PRI: <134>.

3.3.2 Časť HEADER

Časť HEADER syslog paketu obsahuje polia TIMESTAMP a HOSTNAME. Pole TIMESTAMP obsahuje aktuálny čas vo formáte “Mmm dd hh:mm:ss” a musí bez medzery nasledovať za časťou PRI. Pole HOSTNAME obsahuje identifikáciu odosielajúceho zariadenia a je od poľa TIMESTAMP oddelené práve jednou medzerou. Zadanie špecifikuje, aby sme ako identifikáciu použili IP adresu zariadenia, ktorú získavame z informácii o rozhraniach. Použitá je prvá IP adresa po loopkacku. Pole HOSTNAME oddeľuje od zvyšku paketu práve jedna medzera.

3.3.3 Časť MSG

Zvyšok paketu je považovaný za časť MSG. Táto časť obsahuje polia TAG a CONTENT. Hodnota poľa TAG je názov programu, ktorý správu vygeneroval (v našom prípade *isabot*) a od nasledujúcej časti je oddelená práve jednou medzerou. V poli CONTENT sa nachádza samotná správa, ktorá má mať podľa zadania formát <nickname>:<msg>.

Kapitola 4

Použitie

Spustiteľný program sa vytvorí príkazom *make*. Pre spustenie je potrebné zadať dva povinné parametre a prípadne špecifikovať voliteľné. Zoznam parametrov je nasledovný:

`HOST` názov IRC serveru

`PORT` číslo portu, na ktorom serve počúva (predvolený je 6667)

`CHANNELS` názov jedného kanálu (prípadne viacerých kanálov), na ktorý sa klient pripojí

`-s SYSLOG_SERVER` IP adresa logovacieho serveru

`-l HIGHLIGHT` zoznam kľúčových slov oddelených čiarkov

Formát spustenia je nasledovný:

```
./isabot HOST[:PORT] CHANNELS [-s SYSLOG_SERVER] [-l HIGHLIGHT] [-h|-help]
```

Konkrétny príklad spustenia:

```
./isabot irc.freenode.net:6667 "#ISAChannel,#IRC" -s 192.168.0.1 -l "ip,isa"
```

Kapitola 5

Závěr

Program bol implementovaný v jazyku C/C++ podľa špecifikácií uvedených v RFC 1459 [2] a RFC 3164 [1]. Testovanie prebehlo na systémoch Fedora 26 a CentOS 7.4.1708.

Literatúra

[1] C. Lonvick: The BSD syslog Protocol. RFC 3164, August 2001.
URL https://www.ietf.org/rfc/rfc3164.txt

[2] J. Oikarinen, D. Reed: Internet Relay Chat Protocol. RFC 1459, May 1993.
URL https://www.ietf.org/rfc/rfc1459.txt