VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

IRC BOT S LOGOVÁNÍM SYSLOG

IRC BOT WITH SYSLOG LOGGING

SÍŤOVÉ APLIKACE A SPRÁVA SÍTÍ
NETWORK APPLICATIONS AND NETWORK ADMINISTRATION

AUTOR **MILAN AUGUSTÍN**
AUTHOR

BRNO 2017

Obsah

Kapitola 1

Úvod

Úlohou je vytvoriť jednoduchého IRC bota s logovaním pomocou SYSLOG protocolu.

1.1 Internet Relay Chat protocol

Internet Relay Chat (IRC) je aplikačný protocol určený na textovú komunikáciu. IRC je postavené na TCP/IP protocole a využíva klient-server architektúru. Podporuje možnosť distribuovanej siete serverov, ktoré slúžia ako uzly kde sa môžu klienti (non-server aplikácie) pripojiť, do tohto IRC systému. Komunikácia prebieha hlavne medzi viacerými užívateľmi, skupinách, na serveri v miestnostiach nazývanymi kanály, ale je možnosť zasielania správ aj klient klientovi.

1.1.1 IRC správy

Komunikácia v IRC protocole je vo forme správ. Každá správa je ukončená dvojznakom `CR` a `LF` a môže obsahovať maximálne 512 znakov, vrátane dvoch ukončovacích znakov. Ďalej správa nesmie obsahovať znaky `NULL`, `CR` alebo `LF` mimo ukončovacieho dvojznaku a ako jediný biely znak sa považuje `SPACE` (0x20), ostatné znaky sú považované za ne-biele znaky.

Všetky správy musia spĺňať istý formát, inak budú odignorované serverom. A tento formát má svoju "pseudo" BNF notáciu[2]:

```
<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
```

1.2 Syslog protocol

Syslog je určený na logovanie správ z rôznych zariadení na sieti. Tieto správy môžu neskôr slúžiť na analýzu správania sa systému. Sú zasielané cez IP vrstvu s ľubovoľným UDP či TCP protokolom na server, tzv. kolektor, ktorý načúva na porte 514.

1.2.1 Formát syslog správy

Syslog správy majú jednoduchý formát, kde dĺžka správy nepresahuje 1024 znakov a pozostáva z troch častí PRI, HEADER a MSG.

Časť PRI nám určite priority správy. Je zobrazená číslom v desiatkovej sústave a je zakódovaná jednoduchým vzorcom `Facility * 8 + Severity`, kde hodnoty Facility a Severity získame z tabuľky, ktorá je uvedená v dokumente RFC 3164[1]. A to číslo je potom uvodené ostrými zátvorkami.

Ďalšia časť HEADER pozostáva z timestampu a hostname, prípadne IP adresu zariadenia, pokiaľ nepoznáme hostname. Timestamp má formát `Mmm dd hh:mm:ss`, kde mesiac je v anglickom jazyku s veľkým počiatočným písmenom a deň sa zapisuje na dva znaky, kde čísla dní 1-9 sa zapíšu ako medzera a číslovka. Čas je v 24-hodinovom formáte. Ďalej nasleduje jedna medzera rozdelujúca timestamp a hostname, kde hostname nesmie obsahovať medzery a môže to byť len hostname alebo IPv4/IPv6 adresa pôvodcu správy.

Posledná časť MSG pozostáva z častí tag a content. Tag obsahuje názov programu alebo process, ktorý správu vytvoril a nesmie presiahnuť dĺžku 32 znakov. Pri prekročení budú ďalšie znaky považované ako content časť. Inak časť tag sa ukončuje jedným zo znakov `[`, `:` alebo `SPACE` a zvyšné znaky potom patria do časti content.

Kapitola 2

Základné informácie o programe

Aplikácia tvorí jednoduchého IRC bota, ktorý sa napojí na server zadaný prvým argumentom s možnosťou zmeny implicitného portu. Po napojení, sa pripojí na zadané kanály v druhom parametre ako zoznam oddelený čiarkou. Ďalšie parametre sú voliteľné a sú určené pre logovanie správ. Parametrom `-l` a nasledujúcim zoznamom slov nastavíme bota tak aby kontroloval či sa v správach, odoslaných užívateľmi na kanál, nachádza nejaké slovo zo zoznamu, kde sa následne celá správa zaloguje na syslog server. Syslog server môžeme zmeniť z implicitného localhostu na inú adresu uvedenú za parametrom `-s`. Ďalej IRC bot ponúka dve funkcie, ktoré sú vyvolané, ak užívateľ odošle správu na kanál, ktorá začíná kľúčovým slovom `?today` alebo `?msg`.

2.1 Návod na použitie

Aplikácia sa spúšťa pomocou príkazu:

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

`HOST` - názov serveru alebo IPv4 adresu

`PORT` - číslo portu, na ktorom IRC server načúva (implicitne 6667)

`CHANNELS` - zoznam obsahujúci názov jedného alebo viacerých kanálov oddelených čiarkov a uvedených znakom `#` alebo `&`

`-s SYSLOG_SERVER` - je IPv4/IPv6 adresa logovacieho syslog serveru (implicitne 127.0.0.1)

`-l HIGHLIGHT` - zoznam obsahujúci kľúčové slová oddelené čiarkou

`-h, -help` - vypíše nápovedu

2.2 Funkcie IRC bota

Funkcie IRC bota sú dve:

?today - Užívateľ zašle správu na kanál, kde sa nachádza aj bot, ktorej obsahom je len `'?today'`. Bot zašle odpoveď na rovnaký kanál s lokálnym dátumom bota vo formáte `'dd.mm.yyyy'`.

?msg - Užívateľ zašle správu na kanál, kde sa nachádza aj bot, ktorej obsahom je `'?msg <nickname>:<message>'`. Bot zašle odpoveď na rovnaký kanál vo formáte `'<nickname>:<message>'`, pokiaľ je užívateľ prítomný v kanály. Ak nie je prítomný, bot si danú správu uloží a pri najbližšom pripojený užívateľa, pre ktorého bola správa určená, ju odošle.

2.3 Logovanie správ

Pokiaľ bol zadaný parameter `-l`, budú sa kontrolovať správy typu **PRIVMSG** alebo **NOTICE** o prítomnosť aspoň jedného kľučového slova zo zoznamu, ktoré je oddelené bielym znakom od zvyšku správy. Pri nájdení, sa celé znenie správy zaloguje na syslog server s nastaveniami pre facility local0 a pre severity Infomational. Formát správy vyzerá nasledovne: `'<134>Dec 24 18:00:00 192.168.1.135 isabot: vianoce:su tu'`.

Kapitola 3

Návrh aplikácie

Aplikácia bude mať konzolový charakter a bude rozdelená do dvoch hlavných častí. Prvá časť bude obsahovať potrebnú inicializáciu, kde sa spracujú argumenty programu, vytvorí spojenie s IRC serverom a prípadne aj so syslog serverom. Keď všetko prebehne v poriadku, nasledujú tri úvodné správy, pomocou ktorých sa prihlásime pod prezývkou do kanálov. Ďalej nasleduje hlavná časť programu, kde sa budú v nekonečnej smyčke spracovávať postupne prichádzajúce správy.

Kapitola 4

Popis implementácie

Aplikácia bola napísaná v jazyku C++. Boli využité časti kódu z predchádzajúcich projektov do predmetu IPK.

4.1 Trieda Client

Táto trieda bola prebraná z môjho predchádzajúceho projektu do predmetu IPK, ktorú som vylepšil pre tento projekt, kde som využil implementáciu BSD socketov pomocou funkcie `getaddrinfo()` z man-page. Touto funkciou som mal možnosť pripojenia cez IPv4 alebo IPv6 protokol bez ďalších problémov. V tomto projekte som pridal možnosť zmeny datagramu zo statického `SOCK_STREAM` na ľubovoľný vhodný protocol ako napríklad `SOCK_DGRAM`. Týmto spôsobom som vyriešil okrem pripojenia TCP aj UDP spojované sokety.

Výhodou tejto triedy je, že sa nemusím vôbec starať o správu TCP/UDP pripojenia a jediné zachovať postupnosť volania funkcií o ktoré sa postarajú intuitívne názvy funkcií.

4.2 Parsovanie správ

Na prijímanie a filtrovanie správ je potrebné vytvoriť nejaký parser. Parser je implementovaný v triede Controller, kde som si zvolil jednoduchšiu cestu pomocou regexov. Vytvoril som regex ktorý rozparsuje z gramatiky v sekcii 1.1.1 časť message a prefix. Z nich získam nick/servername, command a params. Po tomto rozparsovaní správy môžem rozhodnúť, ako s danou správou naložím, pretože mám k dispozícii rozhodujúci príkaz. Ďalej sa prejde do if-else časti, pretože C++ nepodporuje vo switchy stringy ako iné jazyky. V if-else časti sa odchytia potrebné správy a vyhodnotia sa podľa zadaného správania bota, vďaka príkazu získaného zo správy.

V tomto bode sa narazilo na problém, že regex knižnica v C++ nepodporuje rekurziu, ktorú môžeme vidieť v gramatike v časti params. Preto bolo potrebné vytvoriť jednoduchý automat, ktorý rozdelí jednotlivé parametre pre ďalšiu prácu s nimi. Získané parametre sú uložené vo fronte atribútu triedy Controller, pre jednoduchšiu manipuláciu cez ďalšie jej funkcie.

4.3 Funkcie a vyhľadávanie

Keďže používam regexy na parsovanie správ, rovnako som ich použil aj na vyhľadávanie kľúčových slov. Pri testovaní aplikácie som narazil na problém, keď sa do kľúčoveho slova

vložil nejaký znak, ktorý je riadiacim znakom v regexe. Následne pri pokuse vyhľadať to kľúčové slovo, aplikácia spadla. Na vyriešenie tohto problému som sa rozhodol použiť funkciu `string::find()`. Ďalej len kontrola na okraje kľúčového slova, či je ohraničené bielym znakom, prípadne začiatkom či koncom správy.

Pri funkcií **MSG** využívam hashovacie tabuľky na zaznamenanie príchodov a odchodov užívateľov (online/offline stav). Dokopy mám dve veľké tabuľky. V jednej ukladám informácie o užívateľoch (online/offline; true/false) a v druhej správy pre daného užívateľa (fronta). Nie som si istý či je to správny postup/riešenie, ale prišlo mi to veľmi jednoduché na použitie, keď som použil hashovaciu tabuľku v ktorej som si vyhľadal kanál. Každý kanál potom mal ako hodnotu ďalšiu hashovaciu tabuľku, v ktorej boli jednotliví užívatelia. A každý užívateľ mal svoju hotnotu. V prípade zaznamenávania stavu prítomnosti, jednoduchý boolean. A v prípade, keď prišlo vyvolanie funkcie **MSG**, uložila sa pripravená správa na odoslanie do fronty. Takto zachovám aj poradie správ, ktoré sa majú odoslať.

Literatúra

[1] Lonvick, C.: The BSD syslog Protocol. RFC 3164, RFC Editor, Aug 2001.
URL http://www.rfc-editor.org/rfc/rfc3164.txt

[2] Oikarinen, J.; Reed, D.: Internet Relay Chat Protocol. RFC 1459, RFC Editor, May 1993.
URL http://www.rfc-editor.org/rfc/rfc1459.txt