VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

IRC BOT SO SYSLOG ZAZNAMENÁVANÍM

IRC BOT WITH SYSLOG LOGGING

SÍŤOVÉ APLIKACE A SPRÁVA SÍTÍ
NETWORK APPLICATIONS AND NETWORK ADMINISTRATION

AUTOR **DÁVID BOLVANSKÝ**
AUTHOR

BRNO 2017

Obsah

Kapitola 1

Úvod

Úlohou projektu bolo navrhnúť, implementovať a otestovať **IRC** bota so **SYSLOG** zaznamenávaním. V nasledujúcich kapitolách sú opísané dôležité časti projektu.

V kapitole 2 prebieha úvodom do problematiky, ozrejmením nevyhnutých pojmov. Kapitola 3 sa zaoberá návrhom programu. V kapitole 4 je opísaná jeho samotná implementácia. Kapitola 5 obsahuje informácie o priebehu ladenia a testovania. Kapitola 6 poskytuje prehľad nad používaním programu. V kapitole 7 sú uvedené základné informácie o programe. Posledná kapitola zhrňuje získané vedomosti a skúsenosti z projektu.

Kapitola 2

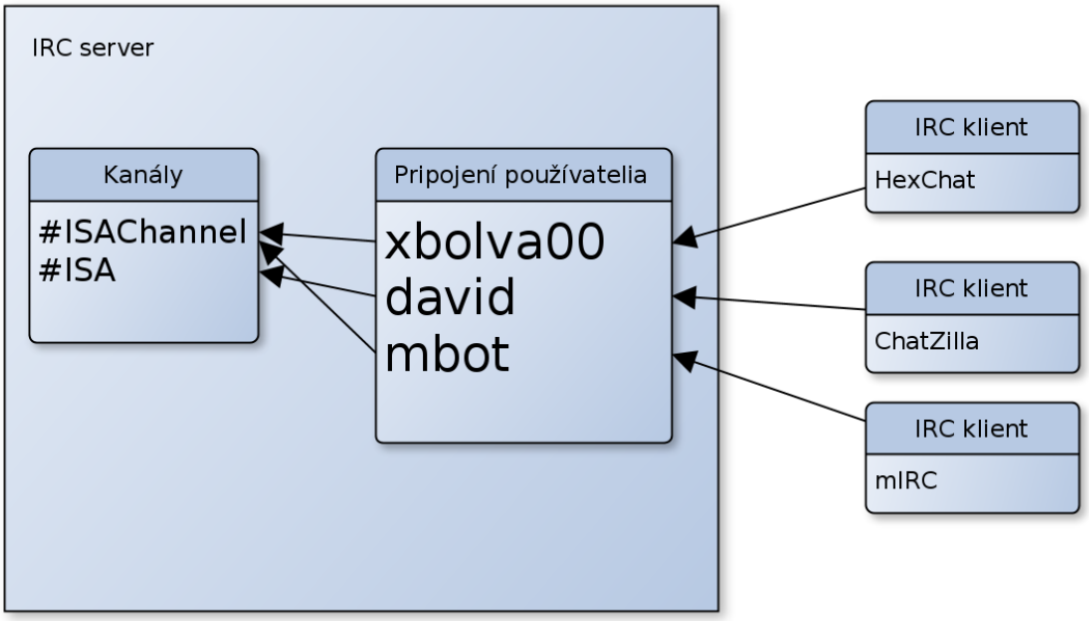
Súhrn pojmov

Táto kapitola obsahuje vysvetlenie jednotlivých pojmov súvisiacich s projektom.

2.1 IRC

Protokol **IRC** (Internet Relay Chat) je textovo založený aplikačný protokol určený na okamžitú textovú komunikáciu. Používa **TCP** a voliteľne aj **TLS**. Najznámejší port pre IRC je **6667**. Za jeho tvorcu je považovaný Jarkko Oikarinen. Protokol je opísaný v dokumente **RFC 1459** [2]. Primárne je určený na komunikáciu skupín v miestnostiach tzv. kanáloch ale umožňuje aj komunikáciou jedného používateľa s druhým pomocou súkromných správ.

IRC je založený na klient–server architektúre. IRC klient je program, pomocou ktorého sa používateľ pripája k IRC serveru a kanálu a komunikuje s ostatnými pripojenými používateľmi. Medzi najznámejšie klienty patrí mIRC, HexChat, ChatZilla. IRC klienti sa pripájajú na IRC servery. Jediná povolená sieťová konfigurácia je spanning tree. IRC servery sa môžu pripojiť na iné IRC servery a týmto spôsobom rozširujú IRC sieť. IRC servery zväčša nevyžadujú prihlásenie používateľov, ale je požadované nastavenie prezývky (nickname) pred pripojením na kanál. Prezývka nesmie mať viac ako 9 znakov.



Obr. 2.1: IRC klient server architektúra

Existujú rôzne typy používateľov IRC. Zakladateľ kanálu je používateľ, ktorý založil kanál. Na danom kanáli ma najvyššie práva a kompletnú správu kanálu. Operátor kanálu má čiastočné práva na správu kanálu získané od zakladateľa kanálu. Ostatní používatelia nemajú žiadne práva na akúkoľvek správu kanálu.

Kanály sú pomenované skupiny jedného alebo viac klientov. Všetci v tejto skupine prijímajú správy pre tento kanál. Kanál sa zakladá po pripojení prvého používateľa a zaniká po odchode posledného. Názvy kanálov začínajú znakmi @ alebo &, nesmú obsahovať znaky medzery a čiarky a ich maximálna dĺžka je 200 znakov.

2.1.1 IRC správy

Servery a klienti si odosielajú správy medzi sebou. Ak správa obsahuje platný príkaz, klient očakáva odpoveď. Každá IRC správa obsahuje tri hlavné časti. Prvou je voliteľný **prefix**, druhou je **príkaz** a tretia časť obsahuje **parametre príkazu**. Časti su oddelené jednou alebo viacerými medzerami. Správy sú zakončené pomocou sekvencie znakov **CLRF** („\r\n“). V sekcii 2.3.1 v RFC 1459 [2] je uvedený formát správy v BNF. IRC správa nesmie presiahnuť dĺžku 512 znakov (vrátane CLRF).

2.1.2 IRC príkazy

Významné IRC príkazy sú:

- **PRIVMSG** <msgtarget> <message>

odoslanie správy (<message>) na cieľ (<msgtarget>) (používateľ alebo kanál).

- **NOTICE** <msgtarget> <message>

odoslanie správy (<message>) na cieľ (<msgtarget>) (používateľ alebo kanál) automatické odpovede nesmú byť odosielané ako odpovede na NOTICE správy

- **JOIN** <channels> [<keys>]

pripojenie ku kanálom (<channels>)

- **PART** <channels> [<message>]

odchod z kanálov <channels>

- **KICK** <channel> <client> [<message>]

odstránenie klienta (<client>) z kanálu (<channel>)

- **NICK** <nickname> [<hopcount>]

zmena prezývky klienta

- **PING** <server1> [<server2>]

testovanie pripojenia k serveru

- **PONG** <server1> [<server2>]

odpoveď na PING príkaz

- **QUIT** [<message>]

odpojenie klienta od servera

2.2 SYSLOG

Syslog je protokol slúžiaci na zaznamenávanie správ. Protokol je opísaný v dokumente **RFC 3164** [1]. Typ architektúry je klient–server. Syslog správy je možné posielať cez **UDP aj TCP** protokoly. UDP port pridelený Syslogu je **514**, u TCP je to **6514**. Správy sa odkazujú na zariadenia (**Facility**), ako napr. auth, deamon, local0, atď. Správam sú tiež pridelené úrovne závažnosti (**Severity**)– Emergency, Alert, Critical, Informational, atď.

2.2.1 Syslog správy

Celý formát Syslog správy sa skladá z troch častí - **PRI**, **HEADER** a **MSG**. Celková dĺžka paketu nesmie presiahnuť 1024 bytov. **PRI** časť sa skladá z 3 až 5 znakov. Začína znakom „<“, nasledovaným číslom a znakom „>“. Číslo udáva prioritu, ktorá reprezentuje zariadenie/subsystém a mieru závažnosti. Túto hodnotu získame nasledovne:

PRI = Facility * 8 + Severity

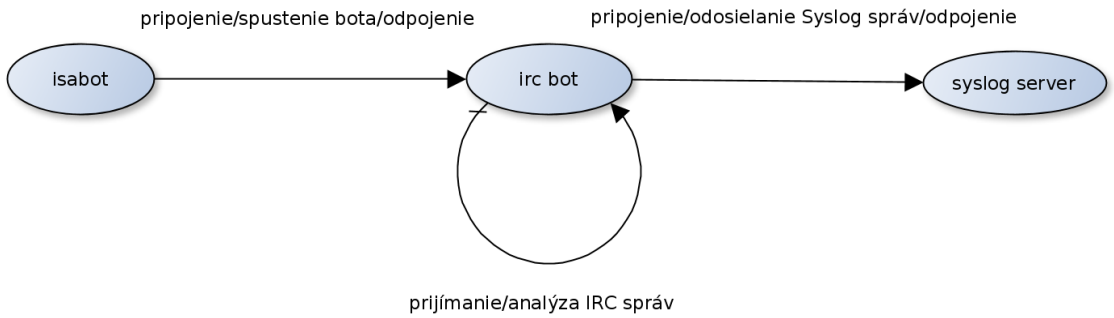
Druhá časť, **HEADER**, obsahuje položku **TIMESTAMP**, kde sa nachádza dátum, čas, názov hostiteľa. Položky su oddelené medzerou. Dátum a čas je uvedený vo formáte „**Mmm dd hh:mm:ss**“, kde Mmm je skrátený anglický názov mesiaca (Jan, Feb, atď.). Nasleduje **HOSTNAME** – ak je adresa hostiteľa neznáma, použije sa IP adresa odosielateľa. Nasleduje tretia časť zvaná **MSG**, ktorá siaha až do konca paketu. Na začiatku nájdeme **TAG**, t.j. informácia o procese/programe, ktorý správu odoslal na Syslog server. Maximálna dĺžka tohto identifikátora je 32 znakov. Prvý nealfanumerický znak značí koniec identifikátora a začiatok druhej časti, tzv. **CONTENT**, ktorý obsahuje text správy.

<PRI>TIMESTAMP HOSTNAME TAG CONTENT

Kapitola 3

Návrh programu

Nasledujúci obrázok popisuje návrh samotného programu, jeho jednotlivé časti/moduly.



Obr. 3.1: Návrh programu

Vstupný bod program je isabot, ktorý spracuje a skontroluje argumenty z terminálu. Ak nedôjde k chybe v tejto fáze, isabot požiada IRC bota o pripojenie k IRC serveru na daný kanál/kanály. Následne sa spustí samotné prijímanie správ a ich analýza. IRC bot sa pripojí na Syslog server, kde sa budú zaznamenávať IRC správy, ktoré obsahujú zadané kľúčové slová. V prípade špecifického textu v správe spúšťa IRC bot svoje dve funkcie pre výpis dátumu alebo odoslanie správy. Po prijatí SIGINT signálu isabot požiada IRC bota o odpojenie od IRC aj Syslog servera a program sa ukončí.

Kapitola 4

Implementácia programu

Program je napísaný v jazyku C++, v štandarde **C++14**. Sieťová komunikácia je implementovaná pomocou **BSD** schránok. Program podporuje **IPv4** aj **IPv6** adresy. IPv6 adresa musí byť zadaná spolu s portom, keďže oddelovač adresy a portu je znak dvojbodky a nebolo by možné pre program zistiť, či dvojbodka je súčasť adresy, alebo je oddelovačom adresy od portu. Jednotlivé logické celky su umiestnené v triedach.

V súbore isabot.cc je samotný vstup programu (funkcia main). V nej sa volá funkcia **parse_args**, ktorá spracováva vstup z terminálu. V prípade chyby je vyhodená výnimka **argument_exception** s popisom chyby, ktorý sa vypíše na štandardný chybový výstup a program sa ukončí s návratovým kódom 1. Následne sa nastavuje vlastná reakcia na prijatie SIGINT signálu – po prijatí tohto signálu IRC bot sa odpojí od IRC servera a program sa ukončí. Ak sa nastavenie vlastnej reakcie na signál SIGINT nepodarí, program sa ukončí s návratovým kódom 3. IRC bot sa pripája k IRC serveru a spúšťa svoj beh. V prípade chyby v sieťovej komunikácii (napr. nepodarí sa pripojiť k IRC serveru) je vyhodená výnimka **network_exception** s popisom chyby, ktorý sa vypíše na štandardný chybový výstup a program sa ukončí s návratovým kódom 2. Ak nedôjde k žiadnym chybám počas behu programu, program sa ukončí s návratovým kódom 0 po prijatí SIGINT signálu.

Súbor irc_bot.cc obsahuje triedu **irc_bot**, ktorá zapuzdruje kód súvisiaci s IRC botom. Nachádzajú sa tu funkcie pre pripojenie/odpojenie sa k IRC serveru, spustenie samotného behu IRC bota, kde sa prijímajú a analyzujú IRC správy.

Program načítava znaku po znaku pomocou funkcie recv z IRC servera a pripája znaky do reťazca. Po prijatí znaku sa následne kontroluje, či tento reťazec je zakončený sekvenciou znakov CLRF. Ak áno, vytvorí sa objekt triedy **irc_command** (implementovaná v irc_command.cc), ktorá ponúka metódy na zistenie typu príkazu a analýzu a detekciu častí IRC správ.

Pripojenie, odosielanie správ a odpojenie od Syslog servera je implementované v triede **syslog_server**, implementované v súbore syslog_server.cc. Ďalej je tu metóda na získanie časového razítka (timestamp), ktoré je súčasťou Syslog správy. Poslednou metódou v tejto triede je funkcia na získanie IP adresy odosielateľa, ktorá je taktiež súčasť Syslog správy.

Pomocné funkcie, ktoré sa používajú v rôznych miestach programu, sú implementované v súbore utils.cc. Je tu funkcia na prevod reťazca na pole (vektor) slov, slová sú v reťazci oddelené medzerami. Ďalej je tu funkcia na prevod pola (vektoru) reťazcov na jeden reťazec, medzi jednotlivé reťazce je pridaný znak čiarky.

4.0.1 Pripojenie ku IRC kanálu

Po pripojení k IRC serveru sa odošle nasledovná sekvencia IRC príkazov. Každý príkaz je zakončený sekvenciou znakov CLRF.

```
NICK xbolva00
USER xbolva00 xbolva00 xbolva00 :xbolva00
JOIN kanál, kanál
```

Ak dôjde k chybe počas tejto fázy (ban na IRC kanáli, kanál je neplatný, atď), program sa ukončí a o vyskytnutej chybe je používateľ informovaný správou na štandardný chybový výstup.

4.0.2 Analýza IRC správ

Nasledujúci algoritmus popisuje implementáciu analýzy IRC správ po jej prijatí z IRC servera.

Vstup: IRC správa zakončená CLRF
Výstup: Spracovaná IRC správa
if *príkaz PING* **then**
  pošli príkaz PONG s parametrom označujúcim server z PING správy
end
if *príkaz PRIVSMG alebo NOTICE* **then**
  zisti prezývku, kanál, text správy
  if *kľúčové slovo sa nachádza v slovách textu správy* **then**
    vytvor a odošli správu na Syslog server
  end
  if *príkaz PRIVSMG a správa obsahuje kanál* **then**
    if *text správy je „?today“* **then**
      pošli príkaz PRIVMSG na IRC server, text správy je aktuálny dátum na zariadení kde beží IRC bot vo formáte „dd.mm.yyyy“
    end
    if *text správy je vo formáte „?msg prezývka:správa“* **then**
      if *je používateľ s danou prezývkou pripojený na kanáli* **then**
        pošli príkaz PRIVMSG na IRC server, text správy je vo formáte „prezývka:správa“
      else
        ulož si správu do interného pola (vektora) správ na neskoršie odoslanie v prípade, že sa daný používateľ pripojený na kanál
      end
    end
  end
end
if *príkaz JOIN* **then**
  zisti prezývku, kanál, text správy
  zmen stav používateľa v interných záznamoch na stav pripojený
  ak existujú čakajúce správy pre tohto používateľa, odošli ich
end

Algoritmus 1: Algoritmus analýzy IRC správ

Vstup: IRC správa zakončená CLRF
Výstup: Spracovaná IRC správa
if *príkaz QUIT* **then**
    zisti prezývku, v interný štruktúrach pre každý kanál nájdi tohto používateľa a zmeň jeho stav na nepripojený
end
if *príkaz PART* **then**
    zisti prezývku, kanál
    v interný štruktúrach pre daný kanál/kanáli nájdi tohto používateľa a zmeň jeho stav na nepripojený
end
if *príkaz KICK* **then**
    zisti prezývku vyhodeného používateľa, kanál
    if *prezývka sa zhoduje s „xbolva00“* **then**
        vypíš informáciu o vyhodení IRC bota z kanálu, ukonči program
    else
        v interných štruktúrach pre daný kanál/kanáli nájdi tohto používateľa a zmeň jeho stav na nepripojený
    end
end
if *príkaz NICK* **then**
    zisti starú a novú prezývku
    v interných štruktúrach zmeň starú prezývku používateľa na novú
    ak existujú čakajúce správy pre novú prezývku používateľa, odošli ich
end
if *príkaz RPL_NAMREPLY (353)* **then**
    zisti zoznam prezývok aktuálne pripojených používateľov, kanál
    v interných štruktúrach pre daný kanál nájdi týchto používateľov a zmeň ich stav na pripojený
end
if *chyba obmedzujúca beh IRC bota* **then**
    zisti text správy informujúci o chybe
    vypíš text o chybe, ukonči program
end

Algoritmus 2: Algoritmus analýzy IRC správ - pokračovanie

Chyby obmedzujúca beh IRC bota sú napríklad:

- vyhodenie z kanála (KICK)
- neplatný kanál – ERR_NOSUCHCHANNEL (403)
- pripojenie na príliš veľa kanálov – ERR_TOOMANYCHANNELS (403)
- nemožnosť odosielania správ na kanál – ERR_CANNOTSENDTOCHAN (404)
- pripojenie sa na príliš veľa kanálov – ERR_TOOMANYCHANNELS (405)
- prezývku IRC bota („xbolva00“) už niekto používa na danom kanáli – ERR_NICKNAMEINUSE (433)
- kanál je plný – ERR_CHANNELISFULL (471)
- kanál len na pozvanie – ERR_INVITEONLYCHAN (473)
- ban na kanáli – ERR_BANNEDFROMCHAN (474)

4.0.3 Funkcie IRC bota

Ak sa v texte správy s príkazom **PRIVMSG** nachádza text „**?today**“, IRC bot odošle aktuálny čas vo formáte „**dd.mm.yyyy**“ na daný kanál získaný pomocou funkcie std::localtime. Ďalej, ak text správy sa zhoduje s formátom „**?msg prezývka:správa**“, program zistí, či používateľ s touto prezývkou je pripojený na danom kanáli. Ak áno, pošle mu správu na daný kanál. Ak nie, správu si uloží do asociatívneho kontajnera std::map[1], kde kľúčom je reťazec (std::string[2]) označujúci prezývku používateľa a hodnotou je vektor reťazcov (std::vector<std::string>[3]), ktoré reprezentujú jednotlivé čakajúce správy na odoslanie. Ďalej existuje nadradený asociatívny kontajner, kde kľúčom je názov kanála a hodnotou je vyššie spomínané asociatívny kontajner s používateľmi (na danom kanáli) a čakajúcimi správami na odoslanie pre týchto používateľov. Následne sa čaká, kým sa daný používateľ znova pripojí na kanál. Pri porovnávaní prezývok nezáleží na veľkosti písmen. Sledovanie pripojených používateľov prebieha nasledovne: po pripojení IRC bota mu IRC server zašle správu **RPL_NAMREPLY** (353) so zoznamom aktívnych používateľov na danom kanáli a pomocou tohto zoznamu si bot zostaví prvotný zoznam používateľov a nastaví ich stav na pripojený. Následne IRC bot sleduje príkaz **JOIN**, po pripojení používateľa sa záznam pridá do tohto zoznamu so stavom používateľa ako pripojený. IRC bot sleduje príkazy **KICK**, **PART**, **QUIT**. Pri príkaze KICK sa získa prezývka vyhodeného používateľa, ak je to prezývka IRC bota („xbolva00“), program sa ukončí, lebo došlo k obmedzeniu jeho funkcionality na danom kanáli. V ostatných prípadoch sa v zozname používateľov nájde daný používateľ s touto prezývkou a jeho stav sa zmení na nepripojený. V prípade príkazu PART si v zozname používateľov u kanálu, z ktorého používateľ odišiel, nájdeme tohto používateľa a taktiež zmeníme jeho stav na nepripojený. U príkazu QUIT sa zmení stav používateľa na nepripojený v zozname používateľov pre každý sledovaný kanál. IRC bot sleduje aj zmenu prezývok na kanáli pomocou príkazu **NICK**, ak dôjde k zmene, táto zmena je aplikovaná aj v zoznamoch používateľov, ktoré si spravuje IRC bot. V prípade, že má IRC bot uložené nejaké správy pre novú prezývku používateľa, odošle mu ich na kanál.

[1] http://en.cppreference.com/w/cpp/container/map
[2] http://en.cppreference.com/w/cpp/string/basic_string
[3] http://en.cppreference.com/w/cpp/container/vector

4.0.4 Odosielanie Syslog správ

V prípade, že nie je zadané kľúčového slovo, IRC bot sa k Syslog serveru nepripája, keďže nie je čo zaznamenávať. Ak je zadané jedno alebo viacero kľúčových slov oddelených čiarkou, postupuje sa nasledovne: ak sa kľúčové slovo nachádza v IRC správe, táto správa sa odošle na Syslog server, ktorý je definovaný prepínačom -s, inak sa použije adresa localhostu, t.j. „127.0.0.1“. Požiadavky v zadaní hovoria o tom, že zariadenie (Facility) má byť local0 (hodnota 16), a miera závažnosti nech je Informational (hodnota 6). V kapitole 2.2.1 je uvedený vzorec na výpočet hodnoty priority Syslog správy a v našom konkrétnom prípade je to: 16 * 8 + 6 = 134. PRI časť začína znakom „<“, nasledovaný číslom priority (v našom prípade **PRI = 134**) a znakom „>“. Na získanie dátumu a času sa používa funkcia **std::localtime**, získanú štruktúru s informáciami následne naformátujeme na formát dátumu a času, ktorý je spomenutý v kapitole 2.2.1. IP adresu odosielateľa zisťujeme pomocou funkcie get_ip_address z triedy syslog_server, ktorá bola popísaná vyššie. Získaný časový údaj a IP adresa odosielateľa sa uvedie do HEADER časti. Ako identifikátor procesu v MSG časti správy sa použije názov nášho programu, t.j. „isabot“. Nasleduje prezývka používateľa, ktorý danú IRC správu, ktorá sa zaznamenáva, napísal. Ďalším znakom je „:“, za ktorým je samotný text správy (**CONTENT**). Takto naformátovaná správa skladajúca sa z **PRI**, **HEADER** a **MSG** častí sa odošle Syslog server.

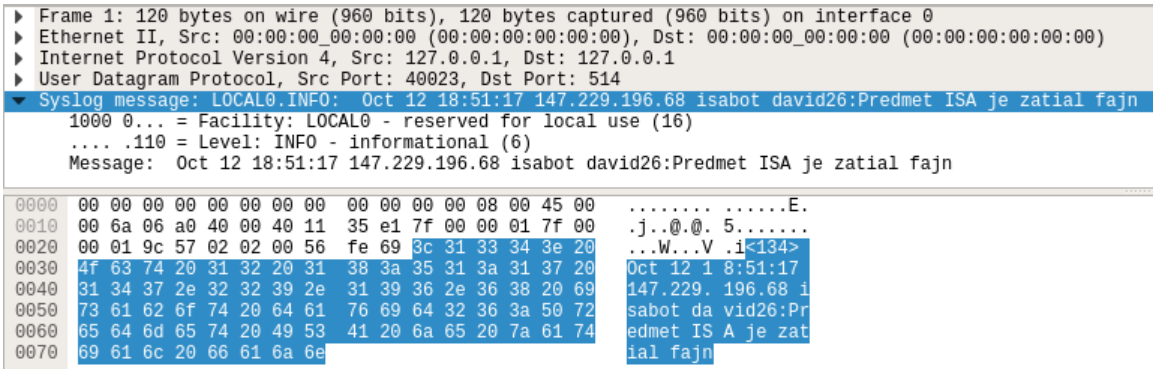
4.0.5 Odpojenie od IRC/Syslog servera

Po prijatí SIGINT signálu IRC bot sa odpojí od IRC servera pomocou príkazu QUIT a uzavrie pripojenie k IRC/Syslog serveru.

Kapitola 5

Ladenie a testovanie programu

Spoločne s programom za na účely testovania a ladenia používal IRC klient **HexChat** a voľne dostupný IRC bot napísaný v Pythone, ktorý vypisoval na štandardný výstup prijaté správy od IRC servera. Tieto informácie poslúžili na overenie správnosti formátov správ, či už u IRC alebo Syslog správ. Na ladenie chýb v kóde sa využívali pomocné výpisy, prípadne krokovanie cez nástroj gdb. Po pripojení nášho programu, HexChatu a Python IRC bota nasledovali testy funkcií bota, kde som v HexChate zadal, či už „?today“ alebo „?msg prezývka:správa“ a sledoval reakcie programu. Pre overenie reakcie bota na ban či vyhodenie, som sa s touto trojicou programov pripojil na kanál, na ktorom nikto nebol, a teda som sa tam stal správcom kanálu, čo znamenalo zisk najvyšších práv na správu kanálu. V HexChate som udelil ban/vyhodil bota z kanála, t.j. používateľa s prezývkou „xbolva00“ a sledoval ako sa program zachová, či sa správne ukončí, a pod. Testovanie Syslog správ prebiehalo tak, že som spustil Wireshark a odchytával som pakety na Loopbacku. Následne napísal nejaký text správy v HexChate, ktorý obsahoval nejaké z kľúčových slov a sledoval záznamy vo **Wiresharku** protokol Syslog. Záznamy som skontroloval na správnosť formátu a údajov v nich.



Obr. 5.1: Sledovanie Syslog správ vo Wiresharku

Kapitola 6

Návod na použitie

Program sa spúšťa cez terminál. Pri zadaní prepínača **-h/--help** sa vypíše informačný text o programe a jeho prepínačoch. V prípade neznámeho či chybne použitého prepínača (nesprávna/chýbajúca hodnota prepínača) alebo pri akejkoľvek chybe v sieťovej komunikácii sa program ukončí a o probléme informuje používateľa správou na štandardný chybový výstup.

```
Použitie: isabot HOST[:PORT] CHANNELS [-s SYSLOG_SERVER] [-l HIGHLIGHT] [-h|–help]
HOST je názov/IP adresa servera (napr. irc.freenode.net)
PORT je číslo portu (predvolené je 6667)
CHANNELS obsahuje jeden alebo viac kanálov (začínajú znakom # alebo &, oddelené sú čiarkou)
-s SYSLOG_SERVER je IP adresa Syslog servera
-l HIGHLIGHT je zoznam kľúčových slov oddelených čiarkou (napr. ip,tcp,udp,isa)
-h|--help zobrazenie informácii o programe a o prepínačoch
```

Program sa ukončuje pomocou **Ctrl-C**, resp. pomocou príkazu „**kill -INT <pid>**“ v termináli.

Kapitola 7

Informácie o programe

Program sa skladá z Makefile, ktorý slúži na zostavenie programu a nasledovných zdrojových súborov:

- isabot.cc
- isabot.h
- irc_bot.cc
- irc_bot.h
- irc_command.cc
- irc_command.h
- syslog_server.cc
- syslog_server.h
- utils.cc
- utils.h

Spolu sa jedná o 813 riadkov zdrojového textu. Veľkosť výsledného binárneho súboru je 91,2 kB (preložené s -02).

Kapitola 8

Záver

Projekt mal za cieľ vyskúšať si programovanie sieťovej služby. Bolo potrebné si naštudovať IRC a Syslog protokoly z RFC dokumentov a následne tieto získané znalosti aplikovať v implementácii samotného programu. Projekt overil nielen komplexne znalosti (analýza RFC dokumentov, práca s BSD schránkami, programovanie v C++, atď.) ale aj programatorské zručnosti - návrh, implementácia, ladenie a testovanie programu. Novozískané vedomosti z tohto projektu sa týkali hlavne programovania sieťových aplikácií a protokolov (IRC, Syslog), na čo sú, čo umožňujú a ako fungujú.

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