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 LOGGING

PROJEKT
PROJECT

AUTOR PRÁCE **JAKUB FAJKUS**
AUTHOR

BRNO 2017

Obsah

Kapitola 1

Úvod

Cílem této práce je vytvoření IRC bota, který bude připojený na dané IRC kanály pro účely monitorování zpráv uživatelů. Bot bude ve zprávách uživatelů vyhledávat zadaná klíčová slova. Zprávy obsahující tato klíčová slova budou zalogovány na syslog server.

Bot bude také poskytovat dvě jednoduché služby uživatelům na IRC kanále. První z nich je vypsání aktuálního data. Druhá služba spočívá v možnosti uložení zprávy pro uživatele, který aktuálně není přihlášený. Bot si tuto zprávu uloží a pošle ji uživateli ve chvíli, kdy se přihlásí.

Kapitola 2

Použité technologie

Tato kapitola stručně popíše použité technologie(syslog a RFC) z pohledu tohoto projektu.

2.1 Syslog

Syslog[1] je relativně jednoduchý textový protokol, popisující způsob posílání a ukládání zpráv(logů), které vznikají na různých zařízeních. Tento protokol definuje různé typy zařízení a způsoby, jakými se mezi těmito zařízeními šíří zprávy.

Syslog definuje 3 typy zařízení.

- Device - zařízení, které je schopno vytvářet zprávy
- Relay - zařízení, které může přijímat zprávy a přeposílat je dále
- Collector(syslog server) - zařízení, které přijímá zprávy, ale dále je nepřeposílá.

Syslog je převážně síťový protokol, pracující nad UDP protokolem. Je možné jej nakonfigurovat tak, aby pracoval pouze na jednom zařízení a všechny zprávy zapisoval do souboru.

Všechny syslog zprávy obsahují protokolem vyžadované informace a samotnou zprávu. Povinné informace syslog zprávy jsou:

- Zdroj zprávy
- Závažnost zprávy
- Datum a čas vzniku zprávy
- Doménové jméno nebo IP adresa zařízení,na kterém zpráva vznikla

2.2 IRC

IRC[2] je textový protokol pracující nad TCP protokolem. IRC je určený pro textovou výměnu zpráv mezi uživateli. IRC definuje architekturu, která je složena ze sítě serverů, které si mezi sebou předávají zprávy, a klientů, kteří zprávy vytvířejí a konzumují.

IRC definuje pojem *kanál*. Kanál je pojmenovaná skupina klientů, kteří obdrží všechny zprávy, které do kanálu přicházejí. Každý klient může být připojený do více kanálů. Každý kanál má svého operátora, kteý kanál spravuje.

Každý klient má svou přezdívku(nickname), která jej jednoznačně identifikuje a která musí být v celé IRC síti unikátní.

2.2.1 Příkazy IRC

IRC klient přijímá dva typy zpráv. První typ jsou zprávy, které server posílá sám od sebe, např. klient se přihlásil, do skupiny byla odeslaná nová zpráva. Druhým typem zpráv jsou odpovědi klientovi na jeho zprávy a případně chybové hlášení, např. výpis všech přihlášených klientů na kanále, nebo chyba v případě, že přezdívka, kterou používá klient, je již použita.

IRC definuje mnoho příkazů(zpráv), používaných pro komunikaci v IRC síti. Z pohledu naší aplikace je zajímavých pouze několik z nich. Jedná se o zprávy: USER, NICK, JOIN, PING, PONG, PRIVMSG, NOTICE, JOIN, PART, QUIT a KICK.

Zpráva **USER** se používa na počátku komunikace a oznamuje připojení nového klienta s danou přezdívkou a jménem. Server v tuto chvíli nekontroluje unikátnost přezdívky.

Zpráva **NICK** slouží k nastavení přezdívky klienta, nebo ke její změně, pokud už klient nějakou má. Server v tuto chvíli kontroluje unikátnost přezdívky. Pokud už je přezdívka použita, hlásí chybu a klienta nepřipojí.

Zasláním zprávy **JOIN** klient oznamuje, že se chce připojit k určeným kanálům. Pokud je připojen úspěšné, server pošle klienovi zprávu s přezdívkami všech klientů na kanále. V příadě jakékoli chyby server posílá klientovi příslušný kód chyby.

Server zasílá periodicky na všechny připojené klienty zprávy **PING**. Pomocí těchto zpráv zjišťuje, zda-li je klient stále aktivní. Pokud ano, klient odpovídá zprávou **PONG**. V případě, že klient neodpoví za určitý čas, server ukončí komunikaci s klientem.

K zasílání zpráv, které vytváří samotní uživatelé IRC sítě, slouží zprávy **PRIVMSG** a **NOTICE**. Tyto zprávy se liší v tom, že na zprávu typu **NOTICE** server negeneruje žádnou odpověd. Příklady odpovědí serveru na zprávy typu **PRIVMSG** mohou byt tyto: neexistující uživatel, nebo nedostačující práva k odeslání zprávy. Obě zprávy lze použít pro odeslání zprávy jak na kanál, tak samotnému uživateli.

Další zprávy, které server odesílá, souvisí s příhlašovaním a odhlašovaním klientů na kanál. V případě přihlášení nového klienta, server odesílá zprávu **JOIN**. Pokud klient opustí kanál, server odesílá zprávu **PART**. Poté, co se klient úplně odpojí od IRC sítě server odesílá zprávu **QUIT**. Operátor kanálu má také možnost odpojit klienta od kanálu. Pokud to udělá, server odesílá zprávu **KICK**.

Kapitola 3

Návrh aplikace

Aplikace je navržena jako konzolová aplikace pro prostředí UNIXových systémů. Aplikace pracuje v jednom vlákně, ve kterém se cyklicky načítají zprávy z IRC serveru. Aplikace je navržená částečně objektově. Komunikaci se serverem syslog zajišťuje BSD socket, pracující nad protokolem TCP. Zprávy na syslog server se odesílají rovněž pomocí socketu, tentokrát za použití UDP protokolu.

Základní části aplikace jsou: zpracování parametrů programu z příkazové řádky, inicializace spojení s IRC serverem, obsluha zpráv přijmaných z syslog serveru, logování na syslog server.

Kapitola 4

Popis implementace

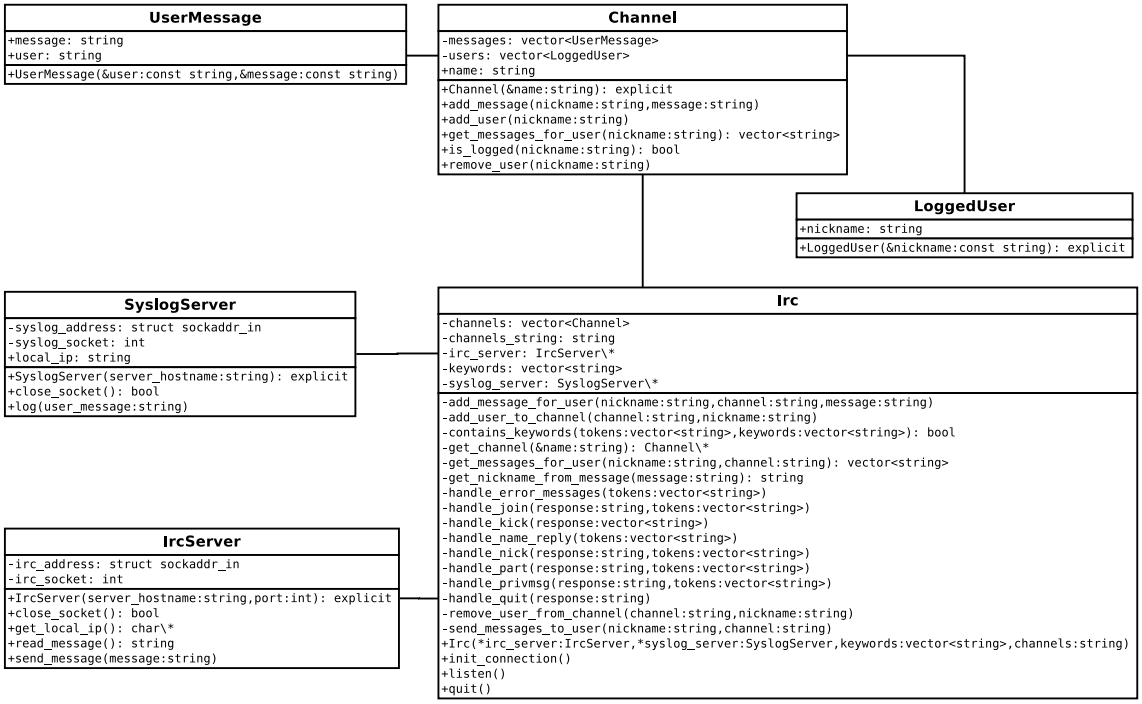
Logika aplikace je rozdělena do několika tříd a do souboru main.cpp. V souboru main.cpp je funkce `main()`, která zpracovává argumenty příkazové řádky a inicializuje ostatní částí aplikace.Tyto další části jsou rozděleny do několika tříd.

Třída `LoggedUser` reprezentuje přihlášeného uživatele.

Třída `Channel` reprezentuje IRC kanál. Kanál si udržuje seznam všech prihlášených uživatelů. Dále si udržuje instance třídy `UserMessage`, která reprezentuje zprávu, kterou nějaký uživatel uložil pro pozdější odeslání.

Třídy `SyslogServer` a `IrcServer` zajišťují práci s BSD sockety. Poskytují rozhraní pro odesíání zpráv na server. Třída `SyslogServer` navíc poskytuje rozhraní pro čtení zpráv.

Nejdůležitější třídou je `Irc`. Tato třída zajišťuje veškerou komunikaci a práci s IRC serverem. Využívá služeb tříd `SyslogServer` a `IrcServer`, které zajišťují nízkoúrovňovou práci se sockety. Je zde také umístěna nekonečná smyčka, ve které se načítají zprávy z IRC serveru, na které se následně reaguje.



Obrázek 4.1: Diagram tříd aplikace

Všechny třídy v případě jakékoli chyby, která nedovoluje správné fungování aplikace, vyvolávají výjimku typu `string` nebo `const char*`. Tyto výjimky jsou zachycovány ve funkci `main()`, která aplikaci řádně ukončí.

4.1 IRC komunikace

Po zapnutí aplikace proběhne inicializace spojení s IRC serverem. Tato inicializace spočívá v odeslání zpráv **USER**, **NICK** a **JOIN**. Tímto se připojí klient, nastaví se mu nickname a připojí se k zadaným kanálům. Poté už se spouští výše uvedená programová smyčka, ve které se načítají zprávy ze serveru.

Každá zpráva, přijatá ze serveru, je řetězec maximální délky 512 znaků. Každá zpráva se skládá z určitých polí, oddělených mezerami. Toho se využívá při zpracovávání zpráv. Zpráva je nejdříve rozdělena na seznam tokenů, které obsahují jednotlivé části zprávy. Podle hodnot v tomto poli se identifikuje druh zprávy. Většina zpráv má svou obslužnou metodu ve třídě `Irc`.

Pokud přijatá zpráva je typu **NOTICE**, nebo **PRIVMSG**, jsou v ní hledány klíčová slova. Zprávy obsahující tato klíčová slova jsou zalogovány na syslog server.

Kapitola 5

Informace o programu

Tyto informace se nacházejí i v souboru README, který je přiložený se zdrojovými kódy aplikace. Pro úplnost jsou i zde.

Aplikace využívající BSD sockety pro připojení k IRC a syslog serveru. Poskytuje možnost sledovat více IRC kanálů a monitorovat zprávy, které uživatelé píší. Tyto zprávy je možné logovat na syslog server, pokud obsahují klíčová slova, zadaná při spuštění bota. Bot také poskytuje uživatelům na IRC kanále dvě služby. První z nich je výpis aktualního data, který se aktivuje zasláním zprávy "?today". Druhou službou je možnost vytvořit a uložit zprávu pro uživatele, který není přihlášený. Bot si tuto zprávu uloží a odešle ji na kanál poté, co se daný uživatel přihlásí. Tato funkce se aktivuje zasláním zprávy "?msg NICKNAME:ZPRÁVA". Kde NICKNAME je nickname uživatele, kterému se má poslat zpráva a ZPRÁVA je samotná zpráva, která se má odeslat.

Následuje popis použití a příklady použití převzaté ze zadání projektu:

```
isabot HOST[:PORT] CHANNELS [-s SYSLOG_SERVER] [-l HIGHLIGHT] [-h|--help]
HOST je název serveru (např. irc.freenode.net)
PORT je číslo portu, na kterém server naslouchá (výchozí 6667)
CHANNELS obsahuje název jednoho či více kanálů, na které se klient připojí
(název kanálu je zadán včetně úvodního # nebo &; v případě více kanálů
jsou tyto kanály odděleny čárkou)
-s SYSLOG_SERVER je ip adresa logovacího (SYSLOG) serveru
-l HIGHLIGHT seznam klíčových slov oddělených čárkou,
např. "ip, tcp, udp, isa"
```

Příklady použití:

```
isabot irc.freenode.net:6667 "#ISAChannel,#IRC" -s 192.168.0.1 -l "ip,isa"
isabot irc.freenode.net "#ISAChannel,#IRC" -l "ip,isa" -s 127.0.0.1
isabot irc.freenode.net #ISAChannel -l "tcp,udp"
```

Literatura

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

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

Seznam obrázků