

Univerza v Mariboru
Fakulteta za elektrotehniko, računalništvo in informatiko

Seminarska naloga pri predmetu
Sistemska programska oprema

Internet Relay Chat

Miha Pirnat, 2007

Contents

1	Uvod	3
2	Protokoli	3
2.1	IRC	3
2.1.1	Prijava	4
2.2	Client-To-Client Protocol (CTCP)	6
2.3	IRC barve	6
2.4	Direct Client-to-Client	7
3	Slike	9
4	Implementacija	11
5	Literatura in viri	12
5.1	IRC in CTCP	12
5.2	IRC barve	12
5.3	DCC	12

1 Uvod

”Internet Relay Char (IRC) je oblika pogovora v živo na internetu. Namenjena je predvsem skupinski komunikaciji mnogih-z-mnogimi v pogovornih forumih, imenovanih kanali, omogoča pa tudi komunikacijo enega-z-enim z zasebnimi sporočili in podatkovne prenose preko Direct Client-to-Client.” (Wikipedia)

IRC je bil ustvarjen leta 1988 in njegova zlata leta so bila devetdeseta leta drugega tisočletja, ko so množice tudi pri nas dobile dostop do interneta. Po letu 2000 je njegova priljubljenost začela počasi upadati predvsem zaradi t.i. ’messengerjev’ (MSN/Live Messenger, ICQ, Yahoo, Skype...), ki omogočajo ’osebni imenik prijateljev’ ter pogovor z videom, zvokom in predvsem neumnimi animiranimi sličicami (smeškoti).

IRC strežniki so lahko povezani v mrežo, s topologijo drevesa, kjer si ”delijo” uporabnike in prenašajo (zaradi tega je v imenu beseda ’Relay’) sporočila uporabnikov ter dogodke (vstop/izstop iz kanala...). Kadar takšna mreža deluje dobro (ni velikih zakasnitev), za uporabnike ni pomembno na katerem strežniku so, dokler so na isti mreži.

2 Protokoli

2.1 IRC

Protokol je tekstovni, osnovna enota sporočila je vrstica. Kodiranje znakov ni predpisano.

Format vrstice, ki jo odjemalec pošlje strežniku kot ukaz je:
ukaz parameter1 ...

Če želimo, da zadnji parameter vsebuje več besed, na začetku parametra dodamo znak : in od tod naprej bo vse en sam parameter. Torej, da se pridružim kanalu #slovenia in jih pozdravim, napišem:

```
JOIN #slovenia
```

```
PRIVMSG #slovenia :Pozdravljeni vsi skupaj!
```

Format vrstice, ki jo odjemalec prejme od strežnika je
:izvor ukaz cilj parameter1 ...

Na primer, kdor bi že bil na kanalu #slovenia, bi moja dva ukaza videl kot:

```
:wolfey! wolfey@wolfey.si JOIN #slovenia
:wolfey! wolfey@wolfey.si PRIVMSG #slovenia :Pozdravljeni vsi skupaj!
```

Strežnik sme dovoliti sporočanje izvora samo strežnikom. Če bi torej jaz pred svoj ukaz napisal izvor, bi lahko po protokolu napisal samo enak izvor, kot ga strežnik sam navede za moja sporočila, strežnik pa v vsakem primeru moj izvor ignorira in zabeleži moje dejanske podatke, torej vzdevek!uporabnik@racunalnik.

Poleg ukazov strežnik sporoča tudi svoje odgovore v obliki številčnih kod. če kot wolfey pošljem ukaz ADMIN, ki izpiše podatke o strežniku, dobim naprimer

```
:ibm.wolfey.si 256 wolfey :Administrative info about ibm.wolfey.si
:ibm.wolfey.si 257 wolfey :irc@wolfey.si
:ibm.wolfey.si 258 wolfey :wolfey's testnet
:ibm.wolfey.si 259 wolfey :IRC Admins <irc@wolfey.si >
```

Te številke pomenijo:

256	RPL_ADMINME	Ime strežnika, ki odgovarja
257	RPL_ADMINLOC1	Prva vrstica lokacije strežnika
258	RPL_ADMINLOC2	Druga vrstica lokacije strežnika
259	RPL_ADMINEMAIL	Elektronska pošta administratorja

2.1.1 Prijava

Oglejmo si primer prijave odjemalca v strežnik. Ob povezavi dobim:

```
NOTICE AUTH :*** Checking Ident
NOTICE AUTH :*** No ident response
```

Na tem mestu strežnik priakuje moje podatke. Napiem:

```
USER wolfey localhost irc.wolfey.si :Moj osebni opis
NICK wolfey
```

Pri ukazu USER so parametri uporabniško ime (wolfey), naslov mojega računalnika (localhost) in strežnik, kamor se povezujem (irc.wolfey.si). Kot prej pri izvoru mojega sporočila, strežnik dejansko localhost ne upošteva niti

ne upošteva, kam se povezujem (irc.wolfey.si). Pri uporabniškem imenu pa sprejme podano uporabniško ime (wolfey), če na mojem računalniku ne teče storitev IDENT, sicer upošteva uporabniško ime, kot ga vrne ta storitev. Če uporabniško ime ni preverjeno z IDENT, je to na IRC strežniku označeno s predpono , torej wolfey.

Če mi strežnik odobri povezavo in je vzdevek (NICK) wolfey prost, dobim od mojega irc strežnika:

```
PING :3740745792
```

To je UnderNetov varnostni dodatek k IRC protokolu. Ker so bili problemi z lažnimi (spoofed) naslovi odjemalcev, zahteva UnderNet, da odjemalec vrne isto številko, kot jo je prejel. Odjemalci iz lažnih naslovov seveda te številke ne bodo videli in za njih je tu konec zgodbe. Jaz pa napišem:

```
PONG 3740745792
```

Zdaj me strežnik pozdravi z informacijami o sebi in mreži v katero je povezan:

```
:ibm.wolfey.si 001 wolfey :Welcome to the UnderNet IRC Network, wolfey
```

```
:ibm.wolfey.si 002 wolfey :Your host is ibm.wolfey.si, running version u2.10.12.12
```

```
:ibm.wolfey.si 003 wolfey :This server was created sob jan 12 2008 at 22:25:20  
CET
```

```
:ibm.wolfey.si 004 wolfey ibm.wolfey.si u2.10.12.12 dioswkgx AbiklmnopstU-  
vrD AbkloUv
```

```
:ibm.wolfey.si 005 wolfey WHOX WALLCHOPS WALLVOICES USERIP  
CPRIVMSG CNOTICE SILENCE=15 MODES=6 MAXCHANNELS=10  
MAXBANS=45 NICKLEN=12 :are supported by this server
```

```
:ibm.wolfey.si 005 wolfey MAXNICKLEN=15 TOPICLEN=160 AWAYLEN=160
```

```
KICKLEN=160 CHANNELLEN=200 MAXCHANNELLEN=200 CHANTYPES=#&
```

```
PREFIX=(ov)@+ STATUSMSG=@+ CHANMODES=b,AkU,l,impstrDd
```

```
CASEMAPPING=rfc1459 NETWORK=UnderNet :are supported by this  
server
```

```
:ibm.wolfey.si 251 wolfey :There are 1 users and 0 invisible on 1 servers
```

```
:ibm.wolfey.si 255 wolfey :I have 1 clients and 0 servers
```

```
:ibm.wolfey.si NOTICE wolfey :Highest connection count: 1 (1 clients)
```

```
:ibm.wolfey.si 422 wolfey :MOTD File is missing
```

```
:ibm.wolfey.si NOTICE wolfey :on 1 ca 1(4) ft 10(10)
```

Od tu naprej dejansko lahko sledi primer s katerim sem začel.

2.2 Client-To-Client Protocol (CTCP)

CTCP je protokol za komunikacijo med odjemalci. Ukaz se prenaša kot sporočilo (PRIVMSG), odgovor pa je vrnjen kot obvestilo (NOTICE).

Da odjemalec loči CTCP ukaz od navadnega sporočila, je na začetku in na koncu CTCP ukaza oziroma odgovora ASCII koda 1 (CTCP je torej "enkapsuliran" znotraj IRC protokola). Čeprav bi po protokolu navadno sporočilo lahko vsebovalo tudi CTCP ukaz, se v praksi običajno uporabnikova sporočila in CTCP ukazi pošiljajo kot ločena sporočila.

Primer CTCP ukaza in odgovora:

```
jaka PRIVMSG jana :\001 VERSION\001
```

```
jana NOTICE jaka :\001 VERSION :Moj super IRC odjemalec, verzija 1.95\001
```

\001 seveda pomeni, da je tam ascii znak s kodo 1.

Opomba: med znakom \001 in naslednjim znakom ni presledka, tu je presledke le zaradi preglednosti.

Poleg VERSION je pogost ukaz še PING, ki kot odgovor pričakuje stevilo milisekund od 1.1.1970 (kot besedilo) in iz tega odjemalec izračuna čas prenosa sporočila med dvema uporabnikoma.

Še najbolj pogost pa je ukaz ACTION, ki ga večina IRC odjemalcev ponudi uporabniku kot ukaz \ME (za trenutno "okno") oziroma \DESCRIBE cilj. Večina uporabnikov se tu verjetno niti ne zaveda, da gre za CTCP ukaz in ne IRC ukaz.

```
Primer: Jaka PRIVMSG Jana :\001 ACTION :ima rad Jano\001
```

bi Jana v običajnem IRC odjemalcu videla kot:

```
Jaka ima rad Jano
```

2.3 IRC barve

Barve so podobno kot CTCP ukazi enkapsulirane znotraj sporočila. Na začetku barvne kode je znak z ASCII kodo 3 (CTRL+C) temu pa sledi številka

barve (1-16) za pisavo in opsijsko za vejico še številka barve ozadja. Samo koda za barvo pomeni konec uporabe prejšne barve.

Barvna tabela:

1	#000000	Black
2	#000080	Navy Blue
3	#008000	Green
4	#FF0000	Red
5	#804040	Brown
6	#8000FF	Purple
7	#808000	Olive
8	#FFFF00	Yellow
9	#00FF00	Lime Green
10	#008080	Teal
11	#00FFFF	Aqua Light
12	#0000FF	Royal Blue
13	#FF00FF	Hot Pink
14	#808080	Dark Gray
15	#C0C0C0	Light Gray
16	#000000	White

Primer:

```
jana PRIVMSG jaka :\003 13Da, dragi?
```

```
.. bo vso sporočilo, ki ga Jana pošlje Jakatu pobarvalo v njeno najljubšo barvo (roza).
```

Opomba: med znakom \003 in naslednjim znakom ni presledka, tu je presledek le zaradi preglednosti.

2.4 Direct Client-to-Client

DCC je protokol, ki dodaja možnost neposredne povezave med odjemalci za zasebne pogovore, ki se ne prenašajo prek strežnika (DCC Chat), ter za izmenjavo datotek (DCC Send)

Oglejmo si, kako poteka prenos datotek. Pošiljatelj polje CTCP ukaz DCC SEND ime_datoteke ip vrata velikost_datoteke

```
Primer: :miha! miha@ibm.wolfey.si PRIVMSG wolfeytest :\001 DCC
```

SEND irccolors.png 2130706433 37649 49364\001

Tu se IP pač pošlje kot eno 32 bitno število. Po originalnem protokolu se je tu prejemnik ali povezal na ip:vrata ali pa ignoriral ponudbo. Kasneje so dodali tudi možnost nadaljevanja prenosa, ki je bil prekinjen. V tem primeri bi prejemnik recimo odgovoril:

DCC RESUME irccolors.png 21000

S tem bi zahteval nadaljevanje od znaka (byte) 21000 dalje. Če pošiljatelj to omogoča, bo odgovoril

DCC RESUME irccolors.png 21000

Sam prenos poteka po blokih, vsak blok mora prejemnik potrditi z 32 bitnim "network byte order" celim številom - pravilo debelega konca (Big Endian). V novejših odjemalcih običajno pošiljatelj pošilja vnaprej, ne da bi čakal na potrditev posameznega paketa - a konec koncev mora biti prej ali slej vseeno potrjeno.

To potrjevanje je rahlo odveč, saj to že počne TCP protokol. Različni odjemalci so uvedli svoje protokole, ki izboljšujejo prenos, vendar razen zgoraj opisanega nobeden ni splošno razširjen.

3 Slike





4 Implementacija

ctcp.h

CTCP ukazi

```
typedef struct {  
    const char* target;  
    const char* command;  
    const char* arguments;  
    struct CTCP *next;  
} CTCP;
```

format.h

Obarvano besedilo

```
typedef struct {  
    int foreground;  
    int background;  
    char* text;  
    struct ColoredText* next;  
} ColoredText;
```

irc.h

Sporočilo strežnika

```
typedef struct {  
    char* sender;  
    char* target;  
    char* command;  
    int argumentCount;  
    char** arguments;  
} ServerMessage;
```

stringutils.h

Pomožne funkcije za delo z besedilom

```
typedef struct {  
    char* text;  
    int offset;  
} TextBuffer;
```

5 Literatura in viri

5.1 IRC in CTCP

<http://en.wikipedia.org/wiki/IRC>
<http://www.irchelp.org/irchelp/rfc/>

5.2 IRC barve

<http://www.ircbeginner.com/ircinfo/colors.html>
<http://www.mirc.co.uk/help/color.txt>
<http://www.ircl.com/colorfaq.shtml>

5.3 DCC

http://en.wikipedia.org/wiki/Direct_Client-to-Client