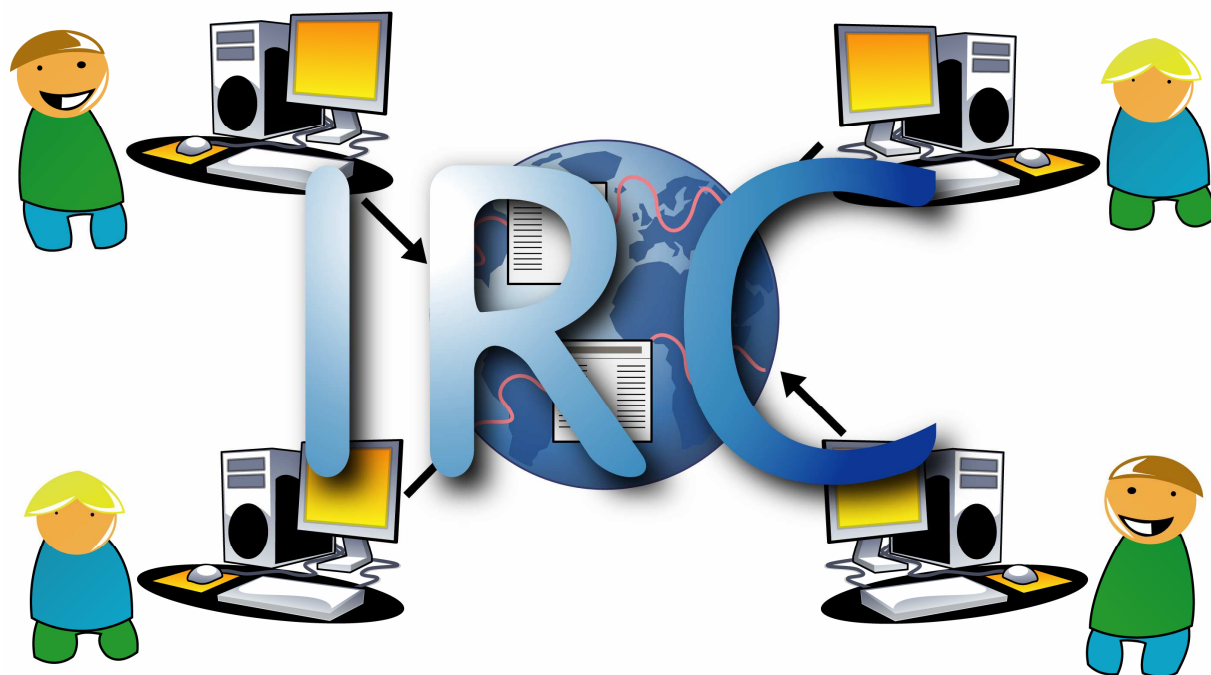


I R C

Internet Relay Chat



Aktualna, edytowalna wersja tego podręcznika jest dostępna w Wikibooks, bibliotece wolnych podręczników pod adresem <http://pl.wikibooks.org/wiki/IRC>

Całość tekstu jest objęta licencją GNU Free Documentation License.

Spis treści

<i>Czym jest IRC?</i>	3
<i>Podstawy korzystania z IRC-a</i>	4
<i>Początki</i>	4
<i>Najbardziej znane publiczne sieci IRC</i>	12
<i>Klienci</i>	16
<i>Serwery</i>	26
<i>Kanały</i>	30
<i>Użytkownicy</i>	38
<i>Netykieta i społeczności</i>	50
<i>Zaawansowane korzystanie z IRC-a</i>	53
<i>Usługi DCC i CTCP</i>	53
<i>Skrypty i boty</i>	60
<i>Protokół i jego odmiany</i>	71
<i>Dodatkowe serwisy</i>	78
<i>Terminologia</i>	90
<i>Autorzy</i>	99
<i>Spis ilustracji</i>	99
<i>GNU Free Documentation License</i>	101

Czym jest IRC?

IRC, a właściwie *Internet Relay Chat*, to usługa sieciowa pozwalająca na tekstową rozmowę z innymi użytkownikami "na żywo".

Usługa funkcjonuje na zasadzie połączenia klient-serwer. Oznacza to, że użytkownicy komunikują się między sobą za pośrednictwem **serwera**. Serwery ułożone są w **sieci** - sieć to grupa serwerów, która jest ze sobą połączona aby użytkownicy podłączeni do tych serwerów mogli się ze sobą komunikować, niezależnie od tego do którego serwera w sieci są przyłączeni.

Do korzystania z IRCa wymagany jest **klient** - program, który łączy się z serwerem i umożliwia prowadzenie rozmów. Istnieje wiele darmowych klientów dla każdego systemu operacyjnego, więc zdobycie takiego programu nie jest trudne.

Zwykle w sieci IRC użytkownicy nie używają swoich prawdziwych imion, lecz posługują się pseudonimami, czyli **nick** lub **nickname**.

Rozmowy prowadzone są na **kanalach**. Wiadomości wpisywane przez użytkowników obecnych na kanale są widoczne dla wszystkich, prowadzona jest wspólna rozmowa, jak na konferencji. Oprócz tego możliwe są rozmowy prywatne, jeden do jeden, a także przesyłanie plików.

IRC, w odróżnieniu od komunikatorów internetowych typu "Gadu-Gadu" jest nastawiony głównie na rozmowy w grupach i to stanowi jego istotę. Jego zaletą jest to, że jest to usługa nie związana z żadnym producentem oprogramowania czy firmą. Specyfikacja protokołu IRC jest całkowicie jawna. Oprogramowanie dla serwerów jest oparte na wolnych licencjach, dzięki czemu każdy kto dysponuje odpowiednią wiedzą i możliwościami sprzętowymi może sobie zestawić swoją własną sieć IRC.

IRC sprzyja tworzeniu spontanicznych **społeczności**, które mają swoje własne obyczaje, zasady i subkulturę. Uczestnictwo w tych społecznościach daje często poczucie określonej tożsamości, a w przypadku osób, które uzyskały wysoką pozycję w ich hierarchii poczucie władzy i satysfakcji.

Na świecie istnieje kilkanaście wielkich, publicznych sieci IRC i tysiące mniejszych, dostępnych publicznie lub tylko dla wybranych grup użytkowników.

Jak korzystać z podręcznika

Podręcznik jest nieco sztucznie podzielony na dwie sekcje, dla początkujących i dla zaawansowanych. W części dla początkujących założono, że jej czytelnik posiada jedynie elementarną wiedzę o internecie oraz nie wie zupełnie nic o IRC, protokołach internetowych i innych tego typu sprawach.

Osoby, które chcą używać IRC wyłącznie do prowadzenia rozmów i nie mają ambicji tworzenia czy zarządzania kanałami mogą w zasadzie ograniczyć swoją lekturę do rozdziału "Początki". Pobieżne przejrzanie rozdziału "Klienty" oraz "Najbardziej znane publiczne sieci IRC" pomoże im wybrać odpowiadający ich potrzebom program i sieć IRC. Aby uniknąć problemów typowych dla początkującego użytkownika IRC, być może warto zapoznać się jeszcze z rozdziałem "Netykieta i społeczności".

Osoby, które są zainteresowane bardziej świadomym korzystaniem z IRC powinny przeczytać wszystkie rozdziały z części "dla początkujących", gdzie bardziej szczegółowo omówione są najważniejsze komendy, przy pomocy których korzysta się z IRC oraz być może rozdział "Usługi DCC i CTCP", gdzie omówione jest m.in. jak przesyłać pliki. Jeśli zamierza się korzystać z sieci Undernet, DALnet, Freenode czy QuakeNet, warto się będzie zapoznać z rozdziałem "Dodatkowe serwisy". Rozdziały te zakładają podstawową wiedzę o IRC i ogólnie o protokołach internetowych, ich zrozumienie nie wymaga jednak zbyt zaawansowanej wiedzy na te tematy.

Rozdziały "Skrypty i boty" oraz "Protokół i jego odmiany" są pomyślane jako rodzaj wprowadzenia dla osób, które chciałyby głębiej wejść w zagadnienia techniczne związane z IRC. Można je potraktować jako wstęp do stania się "rasowym ircownikiem". Jakkolwiek nie ma tam szczegółowych instrukcji jak postawić własnego bota, czy jak postawić swoją własną sieć IRC, rozdziały te pomogą przynajmniej zorientować się jak się do tego zabrać.

W wielu miejscach tej "książki" rozproszone są także informacje o charakterze historycznym, opisujące rozwój IRC. Zwykle były one umieszczane głównie po to aby pomóc zrozumieć aktualny stan rozwoju tej usługi. Być może jednak będą też interesujące dla tych, którzy po prostu chcą coś wiedzieć o historii IRC.

Podstawy korzystania z IRC-a

Początki

IRC zraża do siebie ludzi, gdyż wiele osób uważa, że jest on bardzo skomplikowany w użyciu. Ten rozdział ma pomóc osobom, które chcą zacząć i wykazać, że nie jest to wcale takie trudne.

Aby zacząć użytkować IRCa należy:

- ściągnąć z internetu bądź zdobyć w jakiś inny sposób program do IRC, tzw. klienta a następnie go zainstalować na swoim komputerze; obecnie istnieje kilkadziesiąt różnych tego rodzaju programów pod wszystkie systemy operacyjne i część jest bezpłatna; więcej na ten temat w rozdziale "Klienty"
- nauczyć się kilku podstawowych komend, które zostaną omówione po krótko w tym rozdziale, a bardziej szczegółowo w dalszych

- wybrać sobie jedną z sieci IRC i zalogować się do jednego z jej serwerów; najbardziej znane publiczne sieci IRC są opisane w rozdziale "Najbardziej znane publiczne sieci IRC"
- po udanym zalogowaniu się, wejść na jeden z kanałów, które istnieją w tej sieci i zacząć po prostu rozmawiać z innymi; obsługa kanałów jest dokładniej opisana w rozdziale "Kanały".

Zaczynamy

Omówimy cały proces na przykładzie programu [Chatzilla](#) - ze względu na jego mały rozmiar, dostępność na wiele platform i bardzo łatwą instalację; nieistotne jest dla nas w tym momencie, czy jest lepszy czy gorszy od innych.

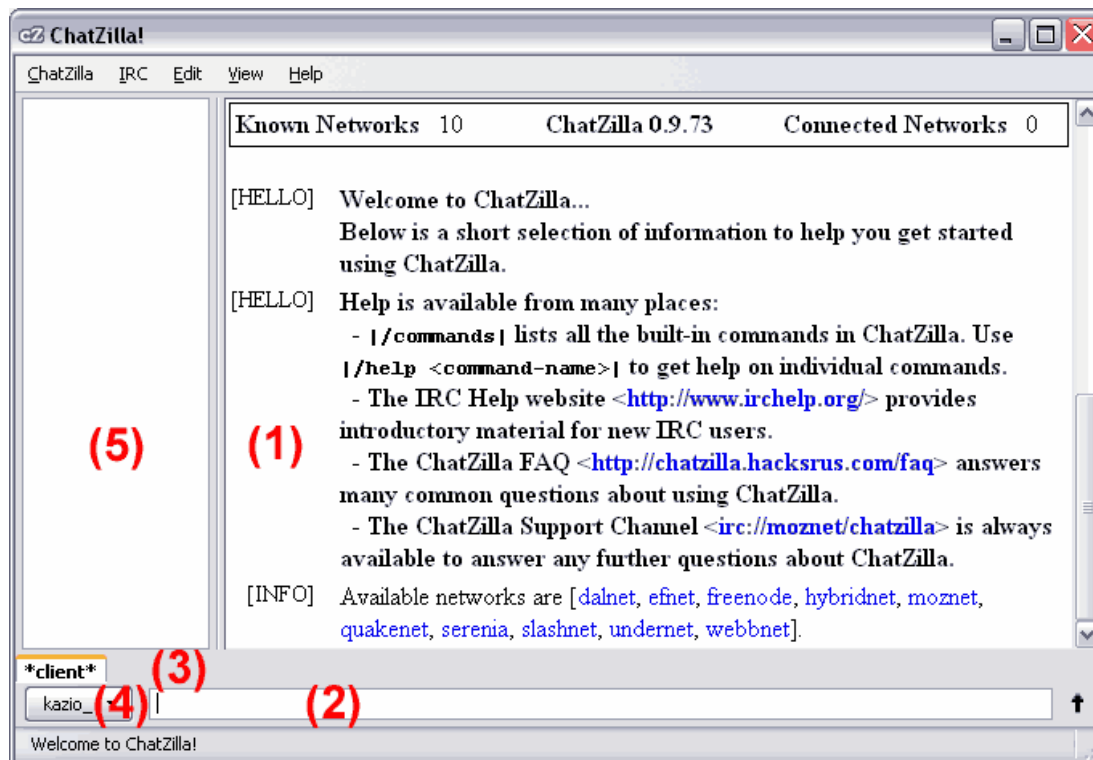
Chatzilla jest dodatkiem (pluginem) do przeglądarek [Mozilla/Firefox](#). Ich użytkownicy muszą jedynie wejść na stronę addons.mozilla.org i wyszukać ten dodatek. Na [stronie opisu Chatzilli](#) znajdują się dalsze informacje.

Nie będziemy się tutaj zbyt dokładnie zagłębiać w działanie samego programu Chatzilla, gdyż wszystkie klienty IRC obsługują się w sumie w dość podobny sposób, a ten został wybrany gdyż jest bezpłatny i stosunkowo prosty w obsłudze, więc na początek powinien wystarczyć. Później, w każdej chwili będziemy mogli popróbować też inne programy do IRC.

Opis zadania

Założmy, że naszym zadaniem jest wejść do sieci IRCnet na kanał o nazwie "#polska" i zamienić z siedzącymi tam ludźmi parę uwag, a później wybrać jedną osobę z tego kanału i porozmawiać z nią prywatnie na dowolny temat.

Krok 1: uruchamiamy program



Zrzut ekranu programu Chatzilla (1) - główne okno komunikatów, (2) - okno wprowadzania komend i komunikatów, (3) - lista paneli, (4) - klawisz z nickiem, (5) - okno listy użytkowników

Po uruchomieniu Chatzilli będziemy mieli taki widok jak na obrazku załączonym obok:

Duże białe pole u góry (1), to okno, w którym będą się pojawiały komunikaty przysyłane do nas przez ludzi lub serwer. Komunikaty te będą się przesuwaly od dołu do góry i w momencie, gdy okno się nimi zapełni, będą od góry "uciekały za ekran".

Zaraz pod dużym oknem przeznaczonym na komunikaty znajduje się małe pole (2) przeznaczone do wpisywania przez nas komunikatów oraz komend, które będą wysyłane do serwera i dalej przez serwer do ludzi, z którymi będziemy rozmawiać.

Między tymi dwoma oknami znajduje się belka (3), w której są klawisze umożliwiające przełączanie się do kolejnych paneli. Chatzilla będzie otwierała nowe panele, gdy będziemy wchodzić na kolejne kanały lub rozpoczynali rozmowy z pojedynczymi osobami. Pierwszy panel, ten który teraz widzimy jest przeznaczony do komunikowania się z serwerem.

Jak odróżnić komendy od komunikatów

To co wysyłamy do serwera używając pola 2 dzieli się na komunikaty i komendy. Komendy od komunikatów odróżnia się przez znak / (ukośnik) na początku linii wysyłanego tekstu. Serwer uważa, że gdy tego znaku nie ma to, co wysyłamy jest prostym komunikatem

przeznaczonym dla jakiejś osoby, a gdy ten znak jest to ma to interpretować jako jedną ze swoich komend.

W tej chwili nie jesteśmy jeszcze zalogowani do żadnego serwera, zatem zarówno nasze komendy jak i komunikaty będą trafiały w próżnię.

Komendy wysyłane na serwer są interpretowane i wykonywane przez serwer a nie przez nasz program. Stąd w gruncie rzeczy zestaw dostępnych komend nie zależy od naszego programu, lecz od tego co potrafi zrobić serwer. Wyjątkiem są tylko komendy, które służą do konfiguracji samego klienta oraz tzw. **aliasy** - czyli uproszczone komendy, które nasz program zamienia w niewidoczny dla nas sposób na bardziej złożone rozkazy zrozumiałe dla serwera.

uwaga techniczna

Od tej pory, będziemy trzymać się zasady, że wszystkie opisywane komendy będziemy poprzedzali znakiem / - którego przy próbach ich wykonania nie należy w żadnym wypadku pomijać. Trzeba koniecznie zapamiętać, że znak / musi być **naprawdę** pierwszy w komendzie, którą będziemy pisali - nie może być przed nim żadnego pustego miejsca ani innego znaku, nawet kropki. Należy też pamiętać, że po znaku / od razu zaczyna się komenda - nie należy po nim wpisywać żadnej spacji.

druga uwaga techniczna

Zarówno komendy jak i komunikaty są wysyłane do serwera po tym jak je wpisujemy do okna 2 i naciśniemy klawisz "enter", nie należy klawisza "enter" używać do rozpoczęcia następnej linii tekstu - IRC i tak pozwala na wysyłanie tylko jednej linii tekstu na raz.

Krok 2: łączymy się z serwerem

Aby połączyć się z serwerem sieci IRCnet z terenów środkowej i północnej Polski piszemy w oknie 2:

```
/server warszawa.irc.pl
```

i przyciskamy klawisz "enter". Możemy też spróbować serwerów: `krakow.irc.pl`, `lublin.irc.pl`, `poznan.irc.pl`, jeśli mieszkamy bliżej tych miast. Połączenie trwa jakiś czas - czasem nawet do 2-3 minut, w czasie których nic się na naszym ekranie nie pokazuje. Gdy wreszcie zostaniemy połączeni serwer wyśle do nas komunikat wyglądający następująco:

```
Network view for "warszawa.irc.pl" opened.  
Attempting to connect to "warszawa.irc.pl". Use /cancel to abort.  
Connecting to irc://warszawa.irc.pl/ (irc://warszawa.irc.pl/), attempt 1,  
next attempt in 15 seconds...  
Please wait while we process your connection.  
Welcome to the Internet Relay Network kazio_!~chatzilla@214.421.32.71  
Your host is warszawa.irc.pl, running version 2.11.1pl  
This server was created Tue Aug 2 2005 at 12:28:48 CEST  
There are 106591 users and 5 services on 41 serversm, 196 operators online,  
54183 channels formed  
=====  
- Porty na serwerze warszawa.irc.pl: 6661 - 6667.
```

```

- UWAGA: Zaleca sie uzywanie portow z zakresu 6661 - 6666.
-
- -----
-
- Inne serwery IRC w Polsce:
- - krakow.irc.pl (6661-6667)
- - lublin.irc.pl (6661-6667)
- - poznan.irc.pl (6661-6667)
- -----
-
- UWAGA klienti:
- Prosimy nie uzywac wiecej niz dwuch sesji jednoczesnie.
-
- Zglaszanie naduzyc: abuse@irc.pl
- Informacje dotyczace IRC: http://www.irc.pl
- Interfejs WWW do bazy K:linii: http://www.irc.pl/kline
-
- =====
- Admin zastrzega sobie prawo do usuniecia z serwera (K:line)
- dowolnej sesji bez podania przyczyn. Jesli Ci sie to bardzo
- nie podoba, to mozesz zmienic serwer, z ktorego korzystasz.
- =====
End of MOTD command.
User mode for kazio_ is now +i

```

Krok 3: zmieniamy naszego nicka

W trakcie logowania do serwera program wybrał dla nas wstępnie jakiegoś przypadkowego **nicka**, czyli pseudonim, pod którym będziemy rozpoznawani przez innych w sieci. Nick ten widać na samym, dolnym, prawym rogu ekranu w polu (3). Jeśli nie podoba się nam ten nick i chcemy zamienić go na jakiś inny piszemy w polu 2 np:

`/nick poczatkujacy29`

i po chwili serwer wyśle do nas komunikat, że nasz nick zmienił się na *poczatkujacy29*. Oczywiście, zamiast *poczatkujacy29* - możemy sobie wybrać przy pomocy tej komendy jakiś inny nick.

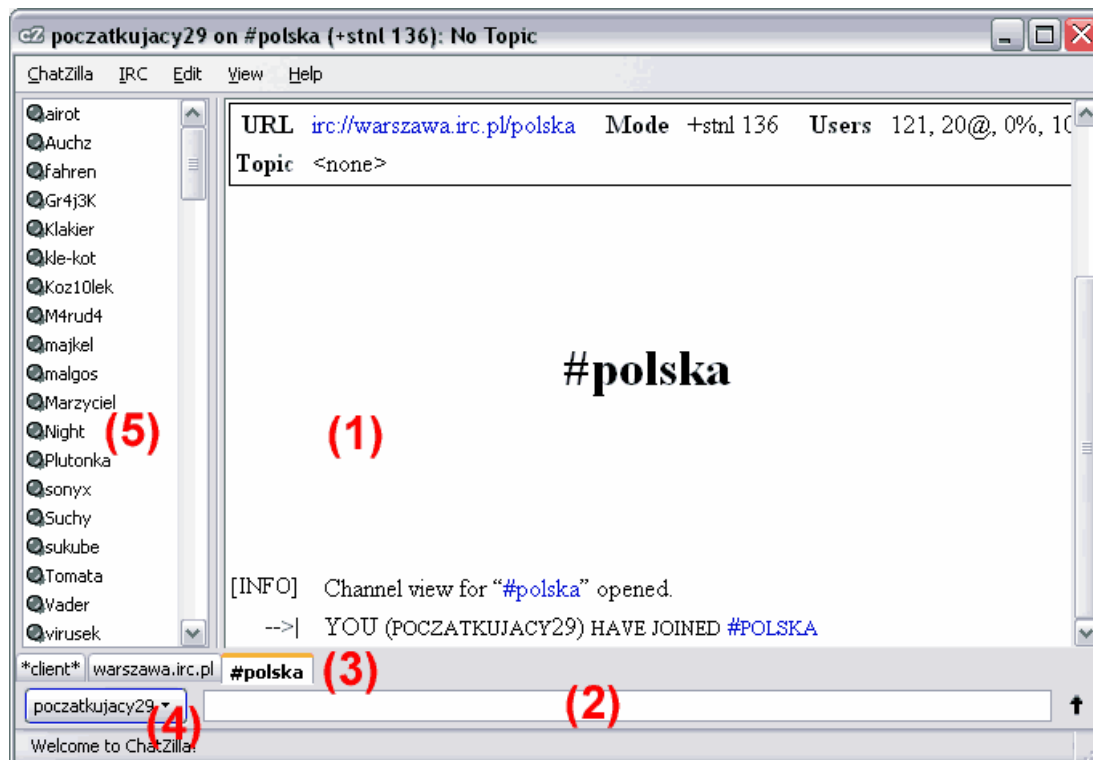
uwaga techniczna

W nicku nie może być polskich liter oraz niektórych znaków specjalnych takich jak "?" "*" "@" i parę innych.

druga uwaga techniczna

Wybrany przez nas nick może być już przez kogoś używany. Wówczas serwer nie pozwoli nam wykonać tej komendy i zamiast tego przyśle ostrzeżenie, że należy sobie wybrać jakiś inny nick.

Krok 4: wchodzimy na kanał



Zrzut ekranu programu Chatzilla otwartego na kanale #polska, sieci IRCnet

Aby wejść na kanał "#polska" piszemy w oknie 2:

```
/join #polska
```

W tym momencie Chatzilla otwiera nowy panel, w którym widać po lewej stronie listę nicków osób, które akurat na nim przebywają (5) a po prawej (1) okno, w którym pojawiają się wysyłane na ten kanał komunikaty.

uwaga techniczna

Osoby zaznaczone zielonymi kropkami na liście 5 to tzw. **opy** - czyli operatorzy kanału, którzy mają na nim pewne szczególne prawa, opisane dokładniej w rozdziale "[Kanały](#)".

druga uwaga techniczna

Nazwa prawie wszystkich kanałów zaczyna się od znaku "#". Istnieją też kanały zaczynające się od innych znaków specjalnych, ale początkowo się na nie raczej nie natkniemy.

Krok 5: gadamy sobie na kanale

Po pomyślnym wejściu na kanał możemy zacząć już na nim rozmawiać poprzez pisanie dowolnych tekstów w oknie 2 i przyciskanie po ich napisaniu klawisza "enter". Np:

Czesc, jestem Kaziu, witam wszystkich zgromadzonych (enter)

co spowoduje, że na kanale pokaże się tekst:

```
<poczatkujacy29>: Czesc, jestem Kaziu, witam wszystkich zgromadzonych
```

i tekst ten zobaczą w ten sam sposób wszyscy inni uczestnicy kanału.

Nasze przywitanie może, ale nie musi wywołać reakcję w postaci przywitania nas przez innych, np. zobaczymy:

```
<mila> Hej, milo, ze tu wpadles
```

albo:

```
<niegrzeczny> Spadaj baranie
```

bo raczej nie oczekujemy, że na IRC-u zawsze, wszyscy będą dla nas mili. W ten sam sposób możemy dalej kontynuować rozmowę na kanale, do czasu, aż postanowimy z niego wyjść, co możemy uczynić komendą:

/part #polska Do zobaczenia

co spowoduje zamknięcie panelu tego kanału, a na kanał zostanie wysłany komunikat, który zobaczą Ci co na nim zostali:

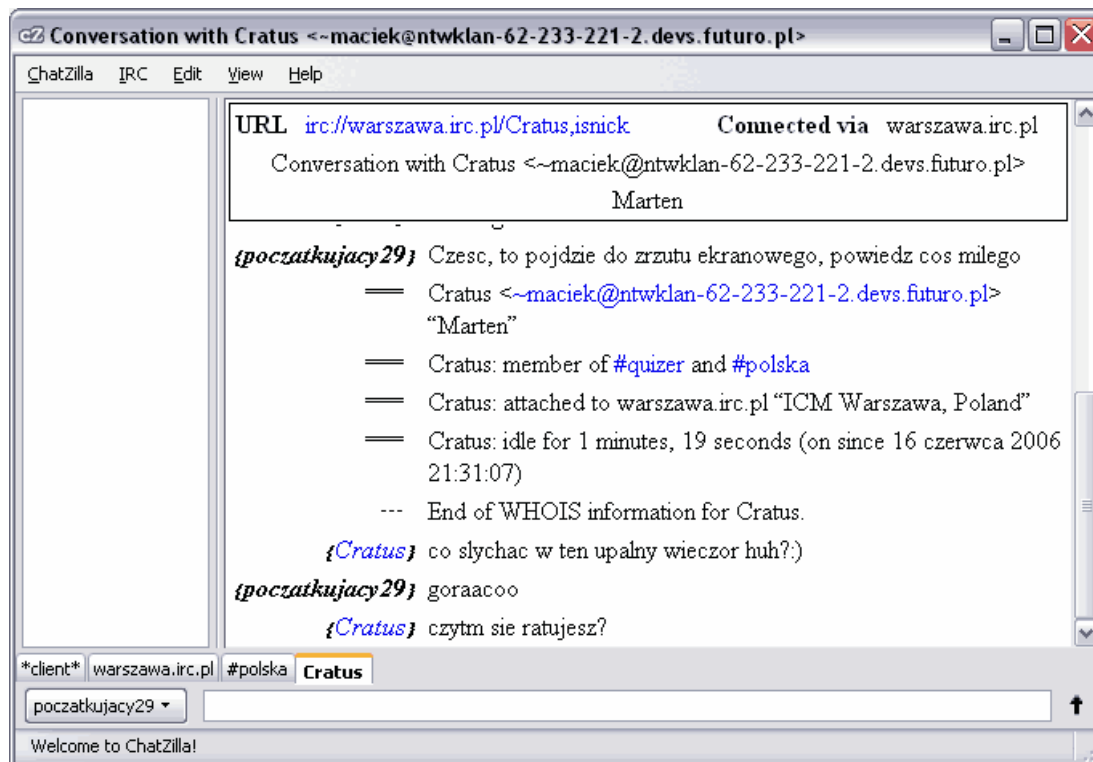
```
*<poczatkujacy29> left channel #polska (Do zobaczenia)
```

Oczywiście, zamiast "do zobaczenia" możemy napisać w komendzie **/part** coś innego.

uwaga techniczna

na różnych kanałach stosuje się różne sposoby kodowania polskich liter. Jeśli nasz program koduje je w inny sposób niż powszechnie przyjęty unikajmy pisania z użyciem "ą,ę,ś,ć,ó" itd. przynajmniej do czasu, aż nie dowiemy się jak dostosować nasz program do tego kodowania. Komunikaty osób, które piszą na tym kanale, stosując kodowanie UTF będą dla nas sprawiały wrażenie jakby zawierały niezrozumiałe "krzaki" - nie jest to wina tych osób ale nasza - bo nasz program nie jest jeszcze poprawnie skonfigurowany.

Krok 6: rozmawiamy prywatnie



Okno rozmowy prywatnej

Załóżmy, że do rozmowy prywatnej, poza kanałem, wybieramy sobie osobę o nicku *cratus*. Konwersację możemy zacząć przez wykonanie komendy:

/query cratus Hej, to pojdzie do zrzutu ekranowego, powiedz cos milego

co spowoduje uruchomienie kolejnego panelu u nas i prawdopodobnie osobnego okna lub panelu w programie IRC, który używa *cratus*. W nowootwartym oknie *cratusa* pokaże się tekst:

{poczatkujacy29} Hej, to pojdzie do zrzutu ekranowego, powiedz cos milego

Panel rozmowy prywatnej przypomina okno kanału, tyle, że nie zawiera listy użytkowników. W tym panelu możemy rozmawiać tak samo jak na kanale pisząc nasze komunikaty w okienku **2**, które się będą ukazywały w dużym oknie **1**. Tym razem będą one jednak trafiały wyłącznie do osoby o nicku *cratus* i do nikogo więcej.

Po zakończonej konwersacji możemy ją zakończyć klikając prawym klawiszem myszki na zakładkę panelu "cratus" i wybierając opcję "hide tab". W wielu innych program można zamknąć okno rozmowy prywatnej pisząc w polu komend:

/query

Cratus nie dostanie od serwera żadnego komunikatu informującego go o tym fakcie, wypada zatem wcześniej go pożegnać.

Krok 7: wychodzimy

Gdy już się nam znudzi zabawa powinniśmy poprawnie opuścić serwer. Możemy to zrobić po prostu wyłączając Chatzillę, ale może to skutkować tym, że jeśli mieliśmy w tym momencie włączone panele z kanałami nasz nick pozostanie się przez jakiś czas na liście i staniemy się tzw. **duchem**. Bardziej elegancko jest przed wyłączeniem programu rozłączyć się z serwerem. Można to zrobić komendą:

/quit Do widzenia

co spowoduje rozłączenie się z serwerem i wysłanie przez niego na wszystkie kanały, na których przebywaliśmy w momencie wykonywania tej komendy komunikatu:

```
* <poczatkujacy29> has left #polska (quit "Do widzenia")
```

Oczywiście, tak samo jak w przypadku komendy **/part** zamiast "Do widzenia" możemy napisać dowolny inny tekst.

Podsumowanie

Przekazane tu informacje, w zasadzie powinny wystarczyć do prowadzenia prostych rozmów za pomocą IRC. Zapoznaliśmy się tu z następującymi komendami:

server	połączenie się z określonym serwerem sieci IRC
nick	zmiana nicka
join	wejście na kanał
part	opuszczenie kanału
query	rozpoczęcie i zakończenie rozmowy prywatnej
quit	rozłączenie się z serwerem

IRC posiada jednak znacznie więcej możliwości, które są opisane w dalszych rozdziałach.

Najbardziej znane publiczne sieci IRC

Protokół i architekturę IRC wymyślił w 1988 roku [Jarkko Oikarinen](#) z Finlandii, w ramach swojej pracy doktorskiej o możliwości prowadzenia teledyskusji za pomocą Internetu.

Oikarinen uważał to tylko za rodzaj niezbyt dopracowanego przykładu jednego z możliwych sposobów prowadzenia takiej dyskusji.

Zabawa spodobała się jednak studentom na Uniwersytecie Oulu, którzy uruchomili pierwszy serwer IRC (tolsun oulu.fi, dziś już niestety nieczynny). Następnie rozesłali do zaprzyjaźnionych administratorów na całym świecie kod źródłowy serwera i klienta IRC.

Powstała w ten sposób pierwsza sieć IRC.

EFNet

Pierwsza sieć IRC pozwalała dołączać się dowolnym serwerom, co po pewnym czasie zaczęło sprawiać kłopoty. W 1990 roku większość administratorów zgodziła się na zmianę zasady otwartości. Pozostała garstka, której przewodził administrator serwera irc:eris.berkeley.edu, została odłączona, a reformatorzy nadali sobie nazwę EFNet (*Eris Free Network*). Odłączeni nazwali się ANet (Anarchy Net), jednak niedługo potem przestali istnieć. Sieć EFNet ma się dobrze do dziś i jest jedną z czterech największych sieci IRC na świecie. Pierwszy polski serwer IRC został przyłączony w 1993 właśnie do sieci EFNet. Obecnie w Polsce działa jeden serwer EFnetu: <irc://irc.efnet.pl>

Oficjalna strona sieci: <http://www.efnet.org/>

IRCnet

W 1996 roku operatorzy IRC nie mogli dojść do porozumienia w sprawie kierunku dalszego rozwoju sieci. Na to nałożyły się problemy z łączem Europa-Ameryka oraz prywatne animozje między administratorami. Skończyło się to tzw. "Wielkim Splitem" - większość serwerów europejskich i azjatyckich odłączyła się od EFNetu i argumentując, że kolebka IRC jest właśnie w Europie, przybrali nazwę "sieć IRC", czyli IRCnet. Z IRCnetu obecnie korzysta jednocześnie ok. 100 000 użytkowników - jest jedną z czterech największych sieci IRC na świecie, a także najpopularniejszą siecią IRC w Polsce. W Polsce włączone jest w nią 6 serwerów, z których 5: <irc://warszawa.irc.pl> , <irc://krakow.irc.pl> , <irc://lublin.irc.pl> , <irc://poznan.irc.pl> oraz <irc://open.pl.ircnet.net> służą bezpośrednio użytkownikom, zaś hub.irc.pl - łączy je wszystkie z resztą IRCnetu.

- Oficjalna strona IRCnetu: <http://www.ircnet.com/>
- Oficjalna, strona polskiej części IRCnetu: <http://www.irc.pl/>

Undernet

To historycznie druga z sieci ogólnosięciowych, powstała w 1993 roku, na początku jako sieć testowa dla programistów zajmujących się pisaniem oprogramowania dla serwerów IRC, potem szybko rozrosła się i jest teraz jedną z czterech największych. Sieć ta na początku była obsługiwana przez te same serwery IRC co serwery EFnetu, tylko na innych "niższych" portach z czego wywodzi się nazwa tej sieci. Później jednak potrzeba testowania pionierskich możliwości takich jak serwis "X" umożliwiający rejestrowanie kanałów

wymusiły zmianę całego protokołu IRC i postawienie sieci własnych serwerów. Aktualnie w Polsce nie ma żadnego serwera przyłączonego do sieci Undernet, można jednak próbować łączyć się z dowolnym, losowo wybranym serwerem Undernetu.

Oficjalna strona Undernetu: <http://www.undernet.org/>

DALnet, USANet i inne

Na bazie pomysłu Undernetu zaczęły od 1995 roku powstawać kolejne nie połączone ze sobą sieci, z których największą jest chyba DALnet. Mają one zwykle możliwość rejestrowania pseudonimów i kanałów oraz są odporniejsze na splity i lagi, nie cieszą się jednak takim powodzeniem jak IRCnet czy Undernet. Wiele z nich, jak USANet, ma charakter czysto lokalny. W DALnecie opracowano najbardziej popularny, mniej restrykcyjny od Undernetowego model dodatkowych serwisów, który później rozpowszechnił się w wielu innych sieciach IRC. W DALnecie też po raz pierwszy zastosowano system automatycznego przydzielania użytkowników do aktualnie najmniej obciążonego serwera. Jakkolwiek w DALnecie można się przyłączyć do dowolnego serwera, zalecane jest łączenie się przez główny adres sieci: <irc://irc.dal.net>, z którego jest się kierowanym do jednego z innych serwerów.

Oficjalna strona DALnetu: <http://www.dal.net/>

QuakeNet

QuakeNet to czwarta i najmłodsza z wielkich sieci IRC. Powstała w 1997 roku jako niewielka sieć dla graczy Quake i Quakeworld. Charakterystyczne dla QuakeNetu są pseudoboty serwisowe "Q" i "L" oraz nastawienie na graczy online. Podobnie jak Undernet i Freenode ma możliwość rejestracji nicków, posiada też stałe kanały służące kojarzeniu graczy i meczów w grach sieciowych.

- Oficjalna strona QuakeNetu: <http://www.quakenet.org/>
- Lista serwerów: <http://staff.quakenet.org/servers.phtml>

FreeNode

Freenode powstała w 2002 r. na bazie serwera *irc.openprojects.net*, istniejącego od 1998 r. FreeNode jest siecią skupiającą użytkowników i programistów wolnego oprogramowania, oraz osób związanych z projektami tworzenia wolnej treści takimi jak Wikipedia. Freenode działa na protokole zbliżonym do DALnetu, dokładniej opisanego Na FreeNode mieszczą się m.in. wszystkie kanały związane z polskimi projektami Fundacji Wikimedia: <irc://irc.freenode.net/#wikipedia-pl>, <irc://irc.freenode.net/#wikinews-pl> (goszczący polskie Wikibooks) itp. Podobnie jak z DALnetem z freenode należy się łączyć korzystając z jednego adresu: <irc://irc.freenode.net>, z którego jest się odsyłanym automatycznie do najmniej aktualnie obciążonego "faktycznego" serwera tej sieci.

Oficjalna strona freenode: <http://freenode.net>

Polskie sieci IRC

Pierwsze serwery IRC w Polsce połączone były do sieci EFnet. Po podziale sieci EFnet na EFnet i IRCnet polska część pozostała przy sieci IRCnet. Pierwszy serwer został założony przez Grzegorza Aksamita na Akademii Górniczo-Hutniczej w Krakowie w 1993 roku. Aktualnym koordynatorem polskiej części sieci IRCnet jest Piotr "Beeth" Kucharski.

Drugą pod względem średniej liczby użytkowników siecią IRC w Polsce jest EFnet, posiadający od grudnia 2000 roku uruchomiony przez ATM S.A. polski serwer. Wciąż trudno tam natrafić na kanały polskich społeczności, niemniej sieć ciągle się rozwija. EFnet jest szczególnie popularny w środowiskach uniwersyteckich oraz polonii z USA i Australii, których serwery po "Wielkim Splicie" pozostały w sieci EFnet. Irc.efnet.pl jest obecnie największym serwerem w kraju pod względem liczby użytkowników. Koordynatorem sieci Efneta w Polsce jest Andrzej Karpiński, były operator sieci IRCnet.

Oprócz tego istnieją rdzennie polskie publiczne sieci IRC nie powiązane z sieciami międzynarodowymi. Zalicza się do nich m.in:

- PolNet - prawdopodobnie największa tego rodzaju sieć, do ok. 3000 użytkowników. Działa na protokole IRCd-hybrid, z serwisami Anope.
- Crashnet - zwana także IRC7.pl - ok. 350 użytkowników w szczycie. Prowadzona jest przez jednego z programistów zaangażowanych w projekt IRCd-hybrid. Oferuje serwisy i szereg dodatkowych usług dla użytkowników.
- NPIRCS - powstała w 2005 roku. Dysponuje 6 serwerami, w różnych miastach Polski, średnio 100 użytkowników, w szczycie 700 i ciągle rośnie. Sieć korzysta z serwisów Anope. Serwery działają na oprogramowaniu UnrealIRCd.
- IRCX.net.pl - (dawniej pod nazwą i domeną ircx.eu.org) sieć założona na początku 2000 roku, średnio 60 użytkowników, w szczycie 120 - działa na protokole IRCX - 3 serwery. Główny kanał: #Polska
- IFNet - sieć założona w 2002 roku przez grupę administratorów i programistów - entuzjastów IRCd - nieznaną liczbą użytkowników - działa na irc2.11.x - 4 serwery - 1 w Niemczech i 3 w Polsce, flagowy kanał to #Polska. Nie posiada serwisów, zamiast nich jest tzw. indeks kanałów umożliwiający zarejestrowanie kanału.

Ponadto w wielu lokalnych sieciach, działających w krajach, gdzie żyją polskie społeczności, istnieją często polskie kanały, np.: #polska w Undernetcie, #polonia w DALnetcie i USAnecie.

Oficjalne strony WWW sieci "czysto-polskich"

- PolNet: <http://www.ircnet.pl>
- Strona IFNet: <http://www.ifnet.de>
- IRCX: <http://www.ircx.net.pl>
- NPIRCS: <http://www.npircs.pl>

Klienty

Klient to program pośredniczący między użytkownikiem i [serwerem sieci IRC](#), który musi być zainstalowany na komputerze osobistym użytkownika, lub na komputerze na którym ma on swoje konto. Sam klient nie jest programem realizującym usługę IRC - jego działanie bez dostępu do serwera nie ma żadnego sensu.

Klient pełni następujące podstawowe funkcje:

- łączy się z serwerem i dostarcza mu danych o użytkowniku, których serwer wymaga przy połączeniu
- przesyła do serwera komendy i komunikaty użytkownika
- porządkuje dane dochodzące z serwera umieszczając je w odpowiednich oknach lub panelach.

Oprócz tego wiele klientów zapewnia:

- ochronę użytkownika przed atakami innych użytkowników
- szyfrowanie danych poufnych
- automatyczną stabilizację połączenia z serwerem
- logowanie rozmów
- gromadzenie danych na temat użytkowników z którym się zetknęliśmy.

Aktualnie na rynku istnieje wiele klientów IRC na w zasadzie wszystkie systemy operacyjne. Część z tych programów jest bezpłatna i oparta na wolnych licencjach, za część (niestety często najlepszych) trzeba zapłacić.

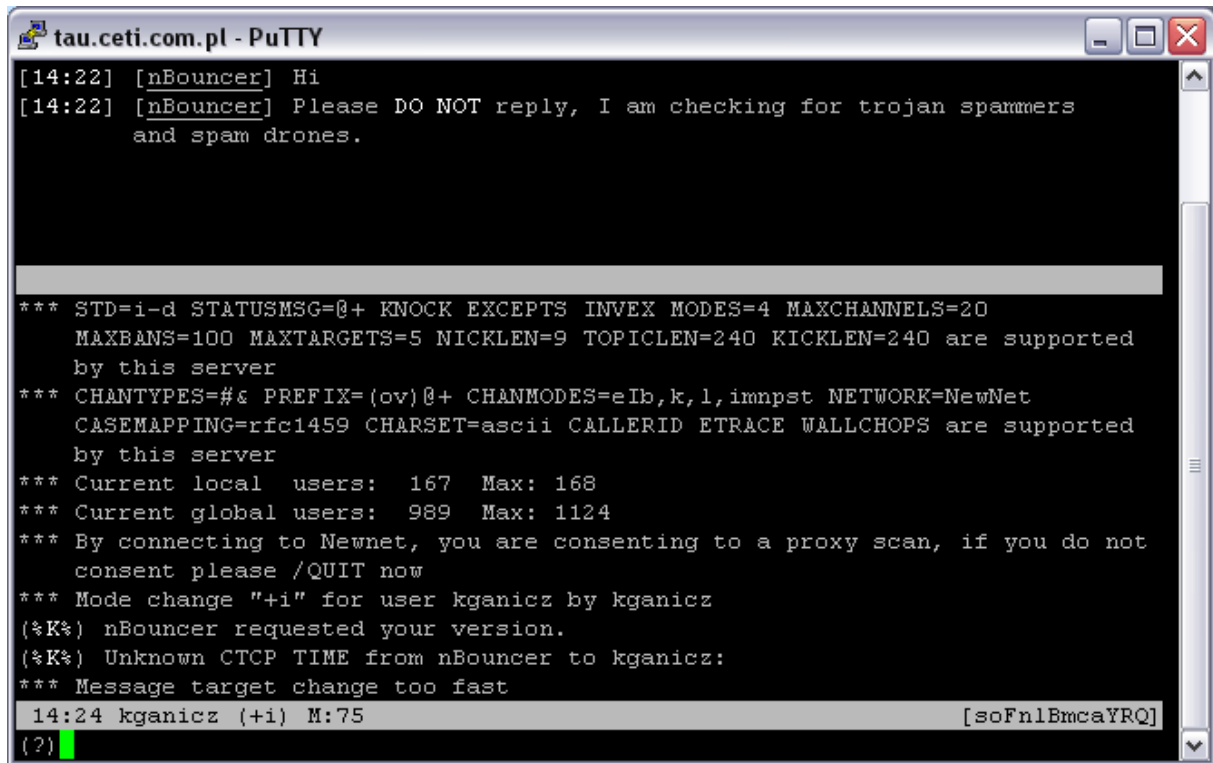
Generalnie klienty IRC można podzielić na:

- działające w trybie tekstowym - w których jest jedno okno komunikatów, które ew. może być dzielone na panele i jedno okno wprowadzania komunikatów i komend - tego typu programy mają jednak często bardzo dużo możliwości, z których korzystają zaawansowani użytkownicy - typowe przykłady to ircII, BitchX, czy irsii
- działające w środowisku graficznym - w którym nadchodzące z serwera dane są automatycznie porządkowane w oknach i w których część najbardziej popularnych komend można wydawać z użyciem myszki i rozwijalnych menu - są to zazwyczaj programy dla początkujących oraz tych, którzy cenią sobie wygodę i nie odczuwają potrzeby zagłębiania się w techniczne szczegóły - typowe przykłady to mIRC, Visual IRC czy pIRCH.
- programy "elementarne" - które koncentrują się na ułatwianiu rozmów czy ogólnie pracy "zwykłych" użytkowników, którzy nie mają ambicji administrowania kanałów czy toczenia "wojen" z innymi użytkownikami - typowy przykład to Chatzilla, Bersirc i Opera-Chat.
- programy "rozbudowane" - posiadające wiele dodatkowych modułów administracyjnych, umożliwiające automatyzowanie wielu czynności oraz modyfikowanie swojego działania poprzez dołączanie do nich skryptów - wiele z

tego rodzaju programów ma już pewne cechy botów - typowe przykłady to BitchX, mIRC, czy EPIC.

Omówienie najbardziej popularnych klientów

ircII



```
tau.ceti.com.pl - PuTTY
[14:22] [nBouncer] Hi
[14:22] [nBouncer] Please DO NOT reply, I am checking for trojan spammers
and spam drones.

*** STD=i-d STATUSMSG=@+ KNOCK EXCEPTS INVEX MODES=4 MAXCHANNELS=20
MAXBANS=100 MAXTARGETS=5 NICKLEN=9 TOPICLEN=240 KICKLEN=240 are supported
by this server
*** CHANTYPES=#& PREFIX=(ov)@+ CHANMODES=eIb,k,l,impst NETWORK=NewNet
CASEMAPPING=rfc1459 CHARSET=ascii CALLERID ETRACE WALLCHOPS are supported
by this server
*** Current local users: 167 Max: 168
*** Current global users: 989 Max: 1124
*** By connecting to Newnet, you are consenting to a proxy scan, if you do not
consent please /QUIT now
*** Mode change "+i" for user kganicz by kganicz
(%K%) nBouncer requested your version.
(%K%) Unknown CTCP TIME from nBouncer to kganicz:
*** Message target change too fast
14:24 kganicz (+i) M:75 [soFn1BmcaYRQ]
(?)
```

Typowy wygląd programu ircII, uruchomionego ze zdalnej konsoli

ircII to bezpośredni potomek programu "irc" napisanego jeszcze przez Jarko Oikarinena. Program jest przeznaczony do pracy na terminalu tekstowym bezpośrednio spod tzw. "shella" systemów typu "UNIX". Zachowuje on wygląd i funkcjonalność programu-klienta Oikarinena, ale dołączono do niego sporo dodatkowych funkcji oraz rozbudowany język do pisania skryptów, który stał się szybko standardem, dzięki czemu istnieją do niego setki skryptów rozszerzających jego możliwości, łącznie z możliwością przekształcenia go w rodzaj prymitywnego bota.

Jakkolwiek ircII jest wyjątkowo mało przyjazny dla użytkownika istnieją wciąż liczni jego zwolennicy, którzy nie "uznają nowych wynalazków". ircII jest dostępny na licencji BSD (*Berkeley Software Distribution*). ircII jest też bardzo często "standardowym" klientem IRC w środowisku Linuksa, dołączanym do większości jego dystrybucji. Na kodzie tego programu są oparte prawie wszystkie pozostałe programy-klienty IRC. Bezpośrednimi jego "mutacjami" są [BitchX](#) i EPIC, które posiadają bardziej od niego rozbudowane możliwości skryptowe i inne dodatkowe funkcje.

Oficjalna strona: <http://www.eterna.com.au/ircii/>

BitchX

```
21:31 Piom> w dyskusji do tego artykułu są moje uwagi, trochę porządku
21:31 Piom> o co chodzi teraz z svg?
21:32 tsca> no nie teraz, tylko w ogóle.
[*] SignOff adziura: #wikipedia.pl (Read error: 104 (Connection reset by peer))
21:40 BRAMKARZ> nie miał zaproszenia
[*] adziura [~adamdziur@80.51.161.2] has joined #wikipedia.pl
[*] Wiki33 [~wikipedys@80.51.161.2] has joined #wikipedia.pl
21:41 Wiki33> nie mam zaproszenia ;)
[*] SignOff Wikipedysta: #wikipedia.pl ☹
21:42 BRAMKARZ> "wikipedysta" też nie miał, i prosze...
21:42 Piom> co tu się dzieje, ale bałagan
21:42 Piom> z tymi dinkami
21:42 Piom> gadziomiednicznymi
21:42 Wiki33> bo to jeden i ten sam był ;)
21:43 matusz> jak będę miał jeszcze jedno dziecko, to nazwę je Wiktoria albo Wiktor
        wołał 'Wiki'
21:45 BRAMKARZ> :-)
21:48 adziura> i będzie jej czytał wikipajki?
21:49 Piom> :D
[*] Wpisz_nick [~5033a102@adi.ehost.pl] has joined #wikipedia.pl
[*] SignOff Wpisz_nick: #wikipedia.pl (Client Quit)
21:52 BRAMKARZ> nie miał zaproszenia
21:54 adziura> a skąd wiesz?
21:55 BRAMKARZ> było widac po oczach
21:55 Piom> co tu się dzieje?
21:55 Piom> czy w kółko trzeba powtarzać "nie miał zaproszenia" ?
21:56 tsca> a czy BRAMKARZ ma w ogóle zaproszenie?
21:56 adziura> ma te co zabrał tym co mieli ;)
21:57 BRAMKARZ> odstąpię niedrogo
[ ][tsca(+eiw)][Mail: 117] [#wikipedia(+n)]
[Lag 0] [0/1 N/10 I/0 V/0 F/0] [U:
[#wikipedia.pl]
```

typowy wygląd BitchX-a

[BitchX](#) był początkowo jednym z najlepszych skryptów "dla profesjonalistów" napisanym do EPICa, mutacji ircII. Ograniczenia funkcjonalności języka skryptowego ircII spowodowały jednak, że zdecydowano się napisać nowego klienta, w którym wszystkie funkcje dawnego skryptu zostały wbudowane integralnie w program, zaś pierwotny język skryptowy ircII został znacznie poszerzony i rozbudowany. BitchX jest dość trudnym w obsłudze programem, w którym nacisk położono nie tyle na funkcjonalność w czasie prowadzenia rozmów ile na zapewnianiu jego użytkownikowi bezpieczeństwa i ułatwianiu kontrolowania kanałów. Stosunkowo prosto daje się go rozbudować do dość zaawansowanego "quasi-bota" pilnującego kanału. Jest popularny wśród zaawansowanych użytkowników pracujących w środowisku Linux.

W rękach "amatora" jest często niebezpieczny, gdyż standardowo jest skonfigurowany do prowadzenia "wojny". Np: przy pomocy prostego skrótu klawiaturowego można go przypadkowo przełączyć w tryb "revenge", w którym wykopuje on i banuje z każdego kanału na który się nim wejdzie każdego użytkownika znajdującego się na "czarnej liście".

Oficjalna strona: <http://www.bitchx.org/>

Irssi

```
Status: Now using Mediawiki v1.4beta3; report technical issues in #mediawiki |
00:03 [ brion_away ] [ JeLuF ] [ oliverh ] [ Supachikn ]
00:03 [ britty ] [ jeronim_ ] [ Olrick ] [ TBSDYlaway ]
00:03 [ Broca ] [ johnfreez ] [ pakaran ] [ termie ]
00:03 [ bumm13 ] [ joshk ] [ Pechorin ] [ tindust ]
00:03 [ CotoideIAA ] [ kdehl ] [ pembertond ] [ tomg ]
00:03 [ chii ] [ KevinMarks ] [ pinkAnthEr ] [ TomSommer ]
00:03 [ Chuq ] [ Khalid ] [ pixlafk ] [ towo ]
00:03 [ cimon ] [ Kieff ] [ publunch ] [ TreyHarris ]
00:03 [ cky ] [ kmccoy ] [ puzzlet ] [ Tristan_ ]
00:03 [ CSB ] [ Kyrin ] [ Raul654 ] [ VerTiCal ]
00:03 [ CXI ] [ Laney ] [ rdnk ] [ voska ]
00:03 [ Cyber.james ] [ lilo ] [ rhobite ] [ waltz ]
00:03 [ Darkhalf ] [ lotek ] [ RichardP ] [ Wegge ]
00:03 [ Dave2 ] [ lotusleaf ] [ Riesz ] [ xcap ]
00:03 [ denelson83 ] [ Luigi30 ] [ robink ] [ yelyos ]
00:03 [ Derk ] [ Lyellin ] [ sam ] [ ZeroOne ]
00:03 [ dma ] [ Mackensen ] [ sannse_away ]
00:03 [ Doomgaze ] [ mark_sweep ] [ Sebbe ]
00:03 -!- Irssi: #wikipedia: Total of 130 nicks [2 ops, 0 halfops, 0 voices,
128 normal]
00:03 -!- Channel #wikipedia created Sat Feb 14 21:08:28 2004
[00:03] [joshk(+ei)] [18:#wikipedia(+cn)] [Act: 5,6,7,9,11,12,13,15]
[#wikipedia]
```

typowy wygląd irssi

[Irssi](#) to konkurencyjny wobec ircII i jego pochodnych program przeznaczony do tekstowego środowiska systemów operacyjnych podobnych do Linuksa. Jest jednak dostępny też do MS Windows i MacOs. Posiada on dość wygodny, jak na programy pracujące w środowisku tekstowym, interfejs i jest standardowo skonfigurowany w nie-agresywny sposób, co powoduje, że w niektórych dystrybucjach Linuksa używany jest zamiast ircII jako "standardowy" klient IRC. Dla jednych jego zaletą, a dla innych wadą, jest to, że korzysta z języka "perl" jako swojego języka skryptowego. Umożliwia to z jednej strony adaptowanie skryptów pisanych z myślą o innych zastosowaniach niż IRC i tworzenie na tej podstawie usługowych "quasi-botów" a z drugiej strony zestaw skryptów przeznaczonych specjalnie dla tego programu jest znacznie skromniejszy od tych dostępnych dla ircII i jego pochodnych.

Na Wikibooks dostępny jest [podręcznik dla Irssi](#).

Oficjalna strona: <http://www.irssi.org/>

mIRC

[mIRC](#) jest prawdopodobnie najbardziej popularnym klientem IRC napisanym specjalnie dla systemu MS Windows i prawdopodobnie najbardziej popularnym tego rodzaju programem w ogóle. W pewnym sensie jest takim samym standardem wśród klientów graficznych dla Windows jak ircII dla klientów tekstowych dla Linuksa. Stanowi on dość rozsądny kompromis między funkcjonalnością dla początkujących i średnio zaawansowanych. Rozpoczęcie z nim pracy jest bardzo proste, a mimo to posiada on wiele rozbudowanych możliwości, które można stopniowo poznawać w trakcie nabierania wprawy. Biblioteka skryptów do niego jest niemal tak samo obszerna jak do ircII i jego pochodnych. Jego język

skryptowy i dodatkowe opcje są nastawione głównie na średnio zaawansowanych użytkowników, którzy nad bezpieczeństwo przedkładają użyteczność i elementy zapewniające "dobrą zabawę". Typowym przykładem "co może" mIRC jest skrypt przekształcający ten program w odtwarzacz mp3, z możliwością wspólnego, jednoczesnego słuchania utworu przez użytkowników jednego kanału.

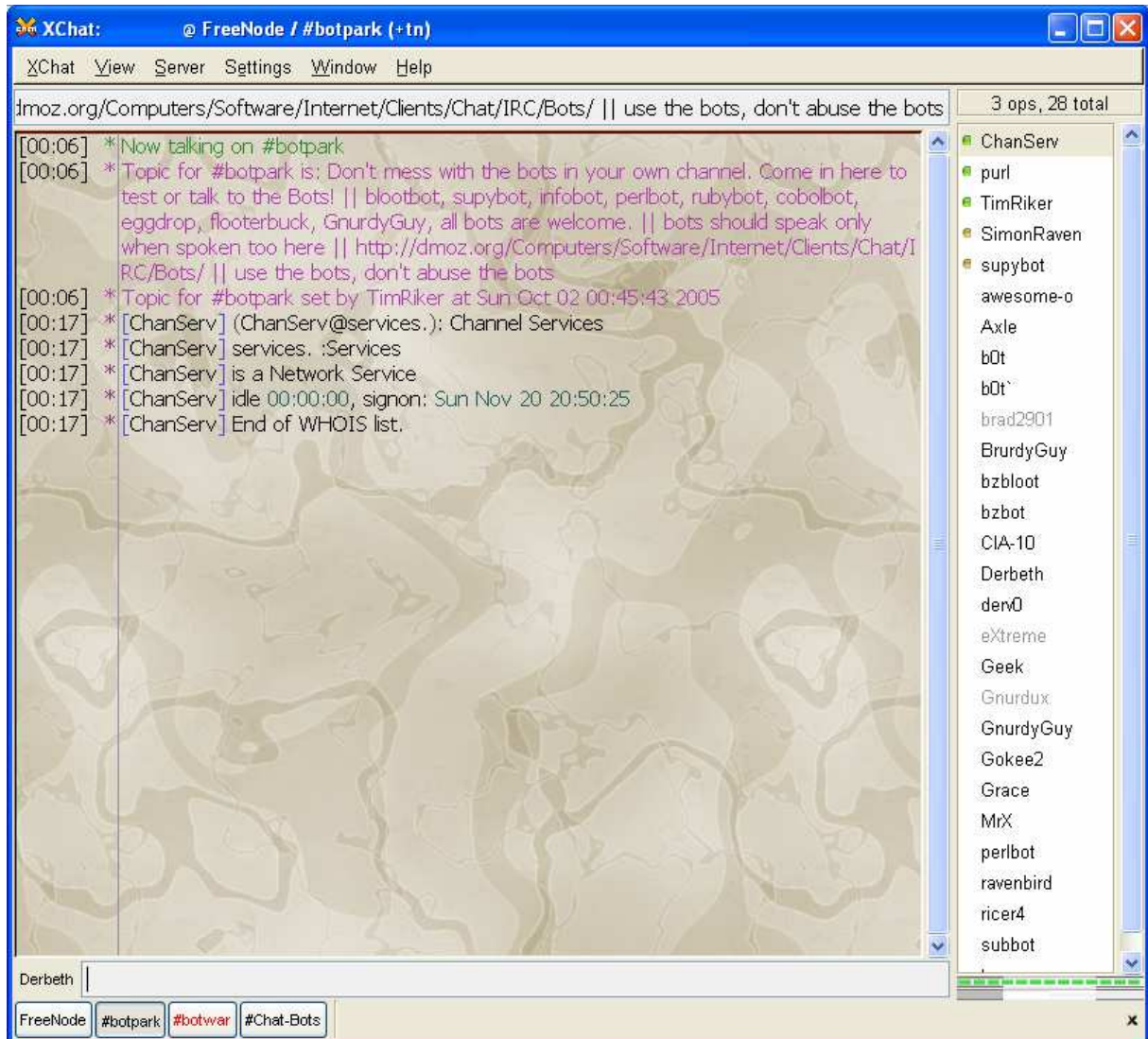
Bardziej zaawansowani użytkownicy krytykują go za "dziwny i nie przemyślany do końca" język skryptowy, który utrudnia pisanie do niego skryptów skutecznie nadzorujących kanały. Słabo chroni on także użytkownika przed atakami w czasie "wojen". W starciu z BitchXem praktycznie nie ma żadnych szans.

Jest programem płatnym, typu shareware. Spory odsetek ludzi używa jego wersje "spiratowane". Program jest krytykowany za umożliwienie zaśmiecania kanałów kolorowanymi tekstami. Umożliwia on bowiem łatwe wysyłanie komunikatów z kodami formatującymi, które są poprawnie interpretowane tylko przez ten program. Wiele skryptów do mIRCa wykorzystuje tę możliwość do tworzenia bibliotek kolorowych "floodów" (tekstów wysyłanych linijka po linijce i tworzących jakiś wzór lub rysunek), które potem można wysłać na kanał jedną komendą.

Oficjalna strona: <http://www.mirc.com/>

Dwa inne, popularne klienty IRC o zbliżonym do mIRCa typie interfejsu, choć o mniejszej funkcjonalności, ale za to bezpłatne i oparte na wolnych licencjach, to pIRCh i Visual IRC.

X-Chat



X-Chat, wersja pod MS Windows

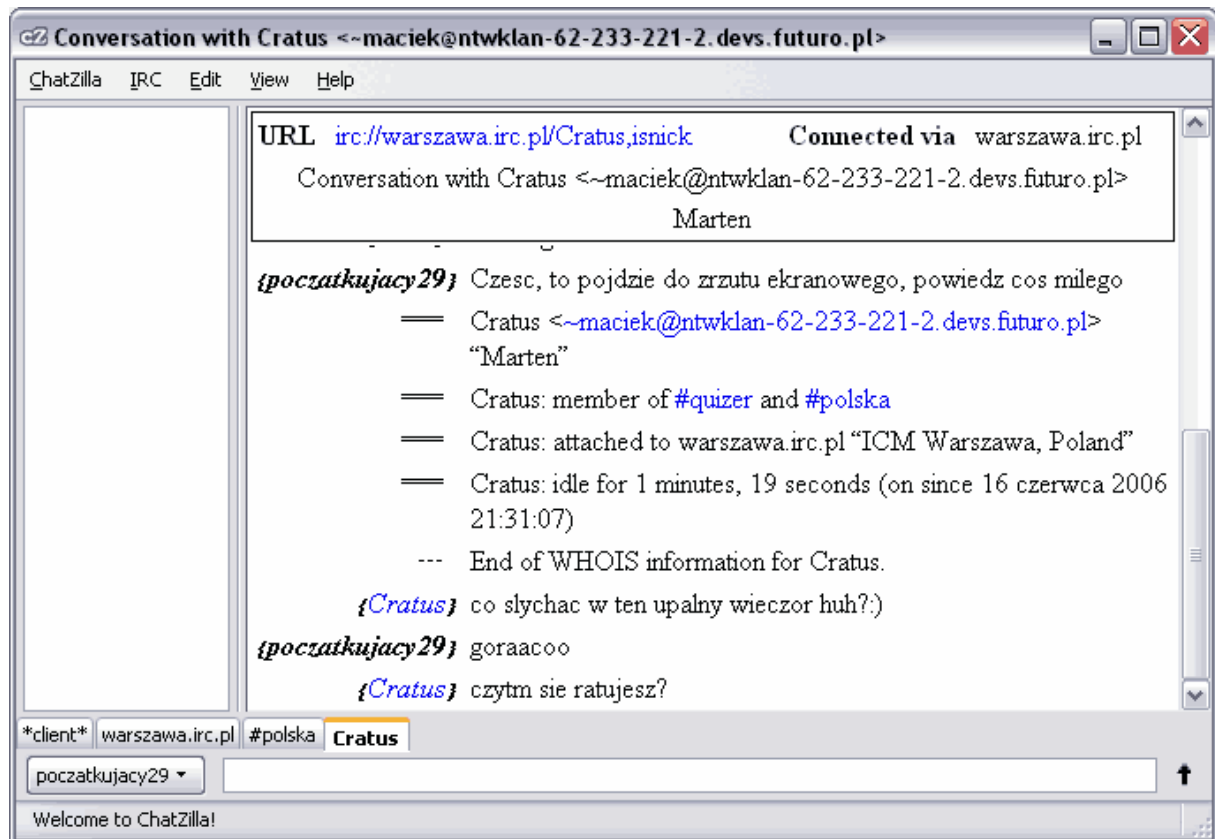
[X-Chat](#) powstał pierwotnie jako klient tekstowy do systemów typu Linux o zbliżonej funkcjonalności do irssi, ale posiadający możliwość dołączania skryptów w potencjalnie każdym języku programowania, dzięki stosowaniu mechanizmu wtyczek. W praktyce, w połowie 2006 r. istniały do niego wtyczki obsługujące, języki perl, python, tcl i ruby. Mimo tego faktyczna liczba skryptów dostępnych do tego programu jest znacznie mniejsza niż do ircII i pochodnych, a nawet do mIRC'a. Na bazie pierwotnego X-Chata, w ramach projektu "X-Chat-Aqua" rozwijane są też jego wersje do kilku środowisk graficznych.

X-Chat jako klient tekstowy nigdy nie osiągnął zbyt dużej popularności przegrywając z irssi wśród średnio zaawansowanych użytkowników i BitchXem wśród zaawansowanych. Jego wersje graficzne cieszą się dużą popularnością wśród średniozaawansowanych użytkowników, gdzie skutecznie konkuruje on z mIRCem i innymi mniej popularnymi klientami tego typu dzięki temu, że oferuje zbliżoną funkcjonalność i jednocześnie jest stale

rozwijany i bezpłatny. Problemem jest fakt, że wydanie dla Windows jest [shareware](#), choć dostępne są nieoficjalne (jednocześnie legalne), w pełni darmowe wersje.

Oficjalna strona: <http://www.xchat.org/>

Chatzilla



Chatzilla - wersja "wtyczkowa" do Firefoxa, pracująca pod MS Windows

[Chatzilla](#) jest przykładem prostego, podstawowego klienta dla początkujących, pracującego w środowisku graficznym. Swoją popularność zawdzięcza temu, że jest bezpłatnie dostępny dla niemal wszystkich systemów operacyjnych oraz temu, że jest dołączany do pakietu przeglądarki Mozilla, a użytkownicy Firefoxa mogą go sobie bardzo łatwo zainstalować w formie wtyczki do tego programu.

Chatzilla posiada dość prosty, niezbyt elegancki, choć dość funkcjonalny interfejs. Teoretycznie można do Chatzilli dodawać skrypty, jednak mogą one być pisane wyłącznie w języku JavaScript, który nie nadaje się zbyt do tego celu. Biblioteki skryptów do Chatzilli są więcej niż skromne.

Oficjalna strona: <http://www.mozilla.org/projects/rt-messaging/chatzilla/>

Podobnym produktem do Chatzilli jest IRC-klient integralnie wbudowany do pakietu przeglądarki "[Opera](#)".

Lista popularnych klientów

Kliknięcie na nazwę klienta, kieruje do jego oficjalnej strony, z której zazwyczaj można go ściągnąć.

	Data pierwszej edycji	Ostatnia stabilna wersja	Cena (w USD)	Licencja
BitchX	1996	1.1	bezpłatne	BSD
Bersirc	1999	2.2.14	bezpłatne	LGPL
ChatZilla	2001	0.9.73	bezpłatne	MPL/GPL/LGPL
EPIC	1994	epic4-2.4	bezpłatne	BSD
HydraIRC	2002	0.3.151	bezpłatne	własnościowe
IceChat	2002	5.50	bezpłatne	własnościowe
ircII	1990	20040820	bezpłatne	BSD
Ircle	?	3.1.2	Shareware, \$23	własnościowe
Irssi	1999	0.8.10	bezpłatne	GPL
Konversation	2002	0.19	bezpłatne	GPL
Kopete	2002	0.12.1	bezpłatne	GPL
KVirc	1999	3.2.0	bezpłatne	GPL
mIRC	1995	6.17	Shareware, \$20	własnościowe
Opera	?	8.54	bezpłatne	własnościowe
savIRC	2001	2.0	bezpłatne	GPL
Visual IRC	1995	2.0	bezpłatne	GPL
wmii	2006	1-rc2	bezpłatne	MIT/X Consortium License
WeeChat	2003	0.1.9	bezpłatne	GPL
X-Chat	2000	2.6.2	Shareware, \$20 dla MS Windows; bezpłatnie dla innych systemów operacyjnych	własnościowe dla MS Windows; GPL dla innych systemów operacyjnych
	Data pierwszej edycji	Ostatnia stabilna wersja	Cena w USD	licencja

- Źródło: [Comparison of IRC clients](#), artykuł z Wikipedii-en

Wsparcie systemów operacyjnych

Systemy operacyjne, w jakich może działać dany klient

	Windows	Mac OS X	Linux	BSD	Unix
BitchX	Częściowo	Tak	Tak	Tak	Tak
Bersirc	Tak	Tak	Nie	Tak	Tak
ChatZilla	Tak	Tak	Tak	Tak	Tak
HydraIRC	Tak	Nie	Nie	Nie	Nie
IceChat	Tak	Nie	Nie	Nie	Nie
ircII	Tak	Tak	Tak	Tak	Tak
Ircle	Nie	Tak	Nie	Nie	Nie
Irssi	Częściowo	Tak	Tak	Tak	Tak
Konversation	Nie	Tak	Tak	Tak	Tak
Kopete	Nie	Tak	Tak	Tak	Tak
KVirc	Tak	Tak	Tak	Tak	Tak
mIRC	Tak	Nie	Nie	Nie	Nie
Opera	Tak	Tak	Tak	Tak	Tak
savIRC	Tak	?	Tak	Tak	Tak
ScrollZ	Częściowo	Tak	Tak	Tak	Tak
Visual IRC	Tak	Nie	Nie	Nie	Nie
wwii	Nie	Tak	Tak	Tak	Tak
WeeChat	Nie	Tak	Tak	Tak	Tak
X-Chat	Tak	Tak	Tak	Tak	Tak
	Windows	Mac OS X	Linux	BSD	Unix

- Źródło: [Comparison of IRC clients, artykuł z Wikipedii-en](#)

Możliwości

Obsługa dodatkowych protokołów, komend, obsługa połączeń z więcej niż jednym serwerem, obsługa połączeń szyfrowanych (SSL i IPv6), obsługa kodowania polskich znaków (UTF-8), graficzny (GUI) lub tekstowy (UI) interfejs, możliwość dodawania skryptów.

	Multyserwerowość	DCC	Pasywne DCC / DCC serwer	SSL	IPv6	UTF-8	UI	GUI	skrypty
BitchX	Tak	Tak	Nie	Tak	Tak	Nie	Tak	Tak	poszerzona składnia skryptów IrcII
Bersirc	Tak	Częściowo	Nie	Nie	Nie	Nie	Nie	Tak	Nie
ChatZilla	Tak	Tak	Nie	Tak	Tak	Tak	Nie	Tak	JavaScript
HydraIRC	Tak	Tak	?	Nie	Nie	Nie	Nie	Tak	nie, ale można dodawać pluginy
ircII	Tak	Tak	Nie	Nie	Tak	Tak	Tak	Nie	własny język skryptowy, który stał się standardem
Irssi	Tak	Tak	Tak	Tak	Tak	Tak	Tak	Nie	skrypty w jęz. Perl
Konversation	Tak	Tak	Nie	Tak	Tak	Tak	Nie	Tak	Shell script
Kopete	Tak	Tak	?	Tak	Tak	Tak	Nie	Tak	{{No}}
KVirc	Tak	Tak	Częściowo	Tak	Tak	Tak	Nie	Tak	KVirc script, Perl
mIRC	Tak	Tak	Tak	Tak	Częściowo	Tak	Nie	Tak	mIRC script (podobny do IRCII)
Opera	Tak	Tak	?	?	?	Tak	Nie	Tak	Nie
savIRC	Tak	Tak	Tak	Tak	Tak	Tak	Nie	Tak	Tcl/Tk
ScrollZ	Tak	Tak	Nie	Tak	Tak	Tak	Tak	Nie	Poszerzony IrcII
Visual IRC	Tak	Tak	Częściowo	Nie	Nie	Nie	Nie	Tak	Versus
wwii	Tak	Nie	Nie	Nie	Nie	Tak	Nie	Nie	Nie
WeeChat	Tak	Tak	Częściowo	Tak	Tak	Tak	Tak	Nie	Perl, Python, Ruby, Lua
X-Chat	Tak	Tak	Częściowo	Tak	Tak	Tak	Tak	Tak	Perl, Python, Tcl, Ruby
	Multyserwerowość	DCC	Pasywne / Serwer DCC	SSL	IPv6	UTF-8	UI	GUI	Skrypty

- Źródło: [Comparison of IRC clients](#), artykuł z Wikipedii-en

Serwery

Serwer IRC to rodzaj programu realizującego na zdalnym komputerze usługę IRC. Serwer generalnie realizuje kilka podstawowych funkcji:

- przyjmuje i odbiera strumień informacji od wszystkich przyłączonych do niego aktualnie klientów
- jeśli pracuje w sieci wieloserwerowej, realizuje przekazywanie i przyjmowanie strumienia informacji z i do innych serwerów
- przechowuje i na bieżąco uaktualnia listę użytkowników i kanałów.

Oprogramowanie serwera IRC jest publicznie dostępne, teoretycznie każdy więc może postawić serwer i następnie, łącząc się z innymi serwerami, budować własną sieć IRC. Serwerami zarządzają tzw. **ircopi**, którzy z tego tytułu mają szczególne przywileje wewnątrz samej sieci. Jakkolwiek dostępne są wersje oprogramowania serwerów na wiele systemów operacyjnych, olbrzymia większość z nich działa pod systemami typu Unix (Linux, Irix itp). Aby działanie serwera miało sens musi on funkcjonować na odpowiednio szybkim komputerze ze stabilnym systemem operacyjnymi i być utrzymywany przez 24 godziny na dobę, 7 dni w tygodniu. Oprócz tego poprawne zainstalowanie, skonfigurowanie i utrzymanie serwera wymaga dużej wiedzy i dysponowania wolnym czasem.

W dużych sieciach IRC istnieją trzy rodzaje struktury połączeń serwerów:

- każdy serwer jest połączony z każdym innym - tworzą więc one gęstą sieć połączeń
- w sieci istnieje serwer "matka" zwany "hubem" - wszystkie inne serwery łączą się za pośrednictwem huba i nie kontaktują się bezpośrednio z sobą; użytkownicy nigdy nie łączą się bezpośrednio z hubem
- układ mieszany - sieć jest podzielona na podsieci; każda z podsieci posiada swojego huba, huby łączą się jednak na zasadzie każdy z każdym tworząc podstawowy szkielet systemu; niektóre huby funkcjonują też jako serwery dostępne bezpośrednio dla użytkowników.

Łączenie się z serwerem

Łączenie z serwerem realizuje się za pomocą komendy:

```
/server adres_serwera numer_portu
```

Adres_serwera jest typowym adresem domeny internetowej np: irc.freenode.org albo konkretnym numerem IP (np: 83.212.34.78). **Numer_portu** - określa port na którym serwer oczekuje na nadchodzące połączenia klientów. Standardowo jest to 6666 lub 6667, niektóre serwery sieci "dla wtajemniczonych" używają jednak nietypowych numerów portów, aby w ten prosty sposób wyeliminować większość niepożądanych gości.

Większość klientów IRC posiada w miarę aktualny spis serwerów większości najbardziej popularnych sieci, dzięki czemu nie trzeba ich szukać w internecie. Większość klientów IRC wykonuje automatycznie komendę `/server` do naszego ulubionego serwera. Wiele programów do IRC umożliwia też utworzenie listy serwerów naszej ulubionej sieci IRC i automatycznie próbuje się łączyć z kolejnymi, gdy z jakichś powodów nasz ulubiony serwer nie chce zaakceptować połączenia.

Zasady akceptacji połączenia są różne w różnych sieciach i na różnych serwerach. Zasady te ustalają operatorzy serwerów. Istnieją sieci dopuszczające tylko osoby łączące się z określonej puli adresów internetowych, sieci sprawdzające jakim klientem próbujemy się łączyć i odrzucające klienty uznane za niedobre czy wreszcie sieci nie pozwalające łączyć się z dynamicznych adresów IP.

Uzyskiwanie połączenia z serwerem trwa zwykle jakiś czas, niekiedy może to być nawet kilka minut. Po uzyskaniu połączenia wyświetla się zwykle komunikat główny serwera (tzw. motd *message of the day*), w którym znajdują się podstawowe informacje o nim, o liczbie istniejących aktualnie kanałów i użytkowników, o tym kto jest jego operatorem oraz często o podstawowych zasadach panujących na nim. Motd można w każdej chwili wyświetlić komendą:

`/motd`

Po uzyskaniu połączenia możemy wydawać dalsze komendy - takie jak wejście na kanał, wyświetlenie listy kanałów itp.

Rozłączenie się danym serwerem bez wyłączania naszego klienta można uzyskać wykonując komendę:

`/server`

bez żadnych parametrów. Rozłączenie następuje też automatycznie przy próbie połączenia się z innym serwerem, choć niektóre klienty IRC posiadają możliwość jednoczesnego łączenia z kilkoma serwerami. Wymaga to jednak zwykle wcześniejszego uruchomienia dodatkowego okna "serwerowego".

Lagi

Lag to zjawisko spowolnienia pracy serwera na skutek jego przeciążenia nadmierną liczbą użytkowników, lub problemów z łączami. Lag powoduje, że serwer przekazuje komunikaty użytkownikom z dużym opóźnieniem. W dużych, wieloserwerowych sieciach lag jest często różny, dla różnych par użytkowników zależnie od tego do których serwerów są oni przyłączeni. Lagi powodują, że komunikaty pojawiające się kolejno na kanale mogły być wysyłane w innej kolejności niż je widzimy, co często prowadzi do mniej czy bardziej zabawnych nieporozumień. Co więcej - kolejność komunikatów jest w tej sytuacji różna na różnych serwerach, stąd obraz kanału użytkowników połączonych do różnych serwerów może być inny.

Lagi mogą być czasami zaskakująco duże. Znane są przypadki z największych światowych sieci takich jak IRCnet, że dochodzą one nawet do 30-40 minut. Lag rzędu kilku sekund jest w zasadzie nieodczuwalny, jednak lag rzędu 1 minuty daje się już mocno we znaki i powoduje wrażenie jakby się rozmawiało z kimś przebywającym na innej planecie. Długie lagi, rzędu kilkunastu minut praktycznie uniemożliwiają już sensowną komunikację.

Aby sprawdzić jak się ma lag do określonego użytkownika można wydać komendę:

/ctcp ping nick

w wyniku czego otrzymuje się odpowiedź:

[nick PING reply]: x sec

W wielu klientach IRC komenda ta jest uproszczona (za pomocą aliasu) do:

/ping nick

Oprócz serwera, częściowo odpowiedzialne za lag może być też nasze własne łącze. Aby się przekonać, jaka część lagu wynika bezpośrednio z naszego łącza z serwerem, możemy wysłać "pinga" do siebie samego.

Jak radzić sobie z lagiem

Najpierw warto zawsze sprawdzić lag do samego siebie - jeśli jest duży warto poszukać innego serwera lub połączyć się ponownie z tym samym. Jeśli nasz własny lag jest mały, ale na kanale, na którym chcemy rozmawiać mamy duży lag do większości użytkowników możemy sprawdzić z jakich serwerów oni korzystają przy pomocy komendy **/whois** (opisanej w rozdziale użytkownicy) i o ile to możliwe przenieść się na serwer, z którego korzysta większość użytkowników danego kanału.

Splity

Split to sytuacja, w której dochodzi do rozłączenia się serwerów, najczęściej na skutek przepełnienia łącz między nimi lub uszkodzenia jednego z hubów. Split może dotyczyć jednego serwera, który samodzielnie odłącza od reszty sieci lub może powodować czasowe podzielenie się sieci na dwa lub więcej mniejszych fragmentów.

Split powoduje, że z kanału "znika" nagle duża grupa użytkowników, co jest szerzej opisane w następnym rozdziale. Split jest najczęściej poprzedzony okresem narastającego laga, stąd można powiedzieć, że lagi są często zwiastunami splitu. W trakcie splitu oprogramowanie serwerów stara się ponownie odzyskać łączność. Różne wersje serwerów mają różne procedury radzenia sobie ze splitami. Splity mogą trwać od kilku sekund do nawet wielu godzin i dni. Szczególnie uciążliwe są częste splity krótkotrwałe, gdyż powodują ciągłe rwanie się dyskusji i zaśmiecanie kanałów licznymi komunikatami o wejściach i wyjściach grup użytkowników z różnych "odsplitowanych" serwerów.

Radzenie sobie ze splitami

Krótkotrwałe, pojedyncze splity najrozsądniej jest po prostu przeczekać. Gdy split trwa dłużej i tak się składa, że większość użytkowników kanału z którymi chcemy porozmawiać znalazła się "po drugiej stronie" warto rozważyć przeniesienie się na inny serwer. Paniczne zmiany serwerów spowodowane krótkotrwałymi splitami pogłębiają tylko chaos. Ludzie przenoszą się z jednego serwera na drugi co dodatkowo je przeciąża i utrudnia im wyjście ze splitu. Na samym kanale, oprócz wyjść i wejść spowodowanych splitem pojawiają się wejścia i wyjścia spowodowane wędrownkami użytkowników po serwerach.

Odmiany serwerów z punktu widzenia użytkownika

Istnieje kilkanaście odmian oprogramowania serwerów, które różnią się różnymi szczegółami a także rodzajem obsługiwanego protokołu. Odmiany te są pochodnymi czterech podstawowych wersji serwerów wywodzących się z oprogramowania EFnetu, IRCnetu, DALnetu i Undernetu (patrz rozdział o sieciach IRC). Z punktu widzenia zwykłego użytkownika różnice między serwerami EFnetu i IRCnetu są prawie niezauważalne, jednak serwery oparte na oprogramowaniu Undernetu i zwłaszcza DALnetu funkcjonują wyraźnie inaczej.

Najłatwiej jest te różnice dostrzec przez porównanie serwerów DALnetu i IRCnetu:

- Z DALnetem łączymy się zawsze stosując adres **irc.dalnet.net** - serwer "matka" automatycznie przydzieli nas do odpowiedniego geograficznie lub aktualnie najmniej obciążonego serwera
- Z IRCnetem łączymy się bezpośrednio z wybranym samodzielnie z listy serwerem
- W DALnecie istnieją dodatkowe usługi wbudowane w serwer umożliwiające rejestrację nicka i kanału
- W IRCnecie musimy sobie radzić z pilnowaniem kanału i nicka sami
- DALnet inaczej obsługuje splity - w zarejestrowanych kanałach "porządek" po splicie jest dokonywany automatycznie przez specjalną usługę - w niezarejestrowanych po splicie wszyscy tracą status operatora

Ogólnie można powiedzieć, że serwery sieci podobnych do DALnetu są nastawione na utrzymywanie daleko idącego porządku po to, aby zapobiec "wojnom na kanale" i kłótniom użytkowników o ich nicki, natomiast serwery sieci podobnych do IRCnetu pozostawiają większą swobodę użytkownikom, co tworzy na nich rodzaj ircowego folkloru, który szerzej opisany jest w osobnym rozdziale.

Techniczne szczegóły działania oprogramowania serwerów i opis ich rozwoju oraz odmian znajduje się w rozdziale "[Protokół i jego odmiany](#)". Jest tam również mowa o zdarzających się czasem problemach z połączeniem się sieciami z powodu niezgodności klienta i serwera.

Szanuj swojego IRCopa

IRCop - czyli zwykle administrator serwera jest w sieci IRC rodzajem "boga". Toczenie z nim sporu na ogół mija się z celem, gdyż ma on zawsze narzędzia umożliwiające pozbycie się

uciążliwego użytkownika. IRCopowie ustalają też ogólne zasady i politykę na serwerze, które nie podlegają zwykle żadnej dyskusji. IRCopowie wykonują ciężką, społeczną pracę, za którą zwykle nikt im nie płaci. Zgłaszanie próśb o pomoc do IRCopa należy zatem zawsze wcześniej dobrze przemyśleć.

Do IRCopa można się zwrócić o pomoc w sytuacji gdy:

- kanał uległ trwałemu, wrogiemu przejęciu i nie ma wątpliwości do kogo on pierwotnie należał i co na nim powinno się dziać
- nastąpiła trwała desynchronizacja kanału na skutek splitu lub lagu
- IRCop założył [k-linię](#), która objęła nas mimo, że nie powinna.

Do IRCopa nie należy zwracać się o:

- pomoc w rozstrzygnięciu sporu między dwiema skłóconymi grupami operatorów kanału - IRCopowie z zasady nie mieszają się do "polityki kanałowej"
- pomoc w odzyskaniu utraconego nicka
- skargami na innych użytkowników

O tym jak znaleźć IRCopa można przeczytać w rozdziale "[Użytkownicy](#)"

Kanały

Kanały to podstawowa cecha IRC-a, która odróżnia go od innych usług umożliwiających chat w internecie. Olbrzymia większość rozmów w IRC odbywa się właśnie na kanałach.

Kanał w pewnym sensie przypomina salę obrad. Wchodząc do niego spotykamy pewną grupę osób, która już w nim jest i prowadzi jakąś rozmowę lub milczy. W odróżnieniu jednak od prawdziwych sal obrad, IRC pozwala na ich tworzenie w każdej chwili, bez ograniczeń co do ich liczby. Podobnie jak z obradami na żywo, istnieją kanały publiczne, na które może każdy wejść jak i "prywatne pokoje", do których dostęp jest ograniczony lub całkowicie zamknięty przed niepowołanymi.

Nazewnictwo kanału

Nazwy większości kanałów rozpoczynają się od znaku specjalnego "#". Niektóre protokoły IRC dopuszczają też "+", "&", "!" i "%". Kanały zaczynające się na "!", "&" mają zwykle szczególne znaczenie (np: są to kanały techniczne, szczególnie chronione, albo kanały pomocowe lub lokalne). Kanały specjalne standardowo nie pokazują się na zwykłej liście kanałów. Za znakiem specjalnym kanału znajduje się jego nazwa.

Większość protokołów IRC ogranicza długość nazwy kanału do 16 lub 32 znaków. W samej nazwie zakazane jest stosowanie znaków specjalnych: "?", "!", "*", "@", "|", "&", "~", "#", ""

(cudzysłów) - które są używane jako metaznaki w składni kluczy wyszukiwawczych, lub mają jakieś inne, szczególne zastosowania

Przykłady nazw:

```
#wikipedia
#wikipedia-pl
@ircops
?help
#wikipedia - przykładowa nazwa nieprawidłowa
```

Wyszukiwanie kanałów

Listę istniejących kanałów na danym serwerze można uzyskać przy pomocy komendy:

```
/list klucz wyszukiwania
```

Gdy użyjemy komendy:

```
/list
```

bez klucza wyszukiwania otrzymamy pełną listę wszystkich kanałów, oprócz kanałów z trybem "+s" (sekretnych). Generowanie takiej pełnej listy kończy się jednak zazwyczaj fatalnie - w dużych sieciach lista ta może liczyć kilkadziesiąt tysięcy pozycji i zanim ją całą otrzymamy serwer zerwie z nami połączenie na skutek przekroczenia dozwolonego limitu danych w jednostce czasu. Z tego względu lepiej jest użyć dobrze przemyślanego klucza wyszukiwania. Klucz ten może się składać z poszukiwanych fragmentów nazw kanału oraz znaków specjalnych takich jak "*", "!" i "?" zwanych metaznakami.

```
/list #wiki*
```

wyświetli listę wszystkich kanałów, których nazwa zaczyna się od "#wiki" - a zatem "#wikipedia", "#wikinews", "#wikimedia-pl itd.

```
/list #wiki*pl
```

wyświetli listę wszystkich kanałów, których nazwa zaczyna się od "#wiki" i kończy na "pl" a zatem "#wikipedia-pl", "#wikinews-pl", itd.

Lista kanałów zawiera zwykle jego nazwę, liczbę użytkowników, tryb i topic:

Przykład działania komendy **/list**

```
/list wiki*
warszawa.irc.pl Tue Jun 13 21:52:40 2006
#PartioWiki 18 public
#wikipedia 15 +int blowfish: de goeiste
#tuxboxwiki 10 public http://wiki.tuxbox.org - Das TuxBox Wiki-Projekt
```

```
#pukiwiki 7 +nt
#autowiki 5 public
#murowiki 5 +t Älkää peljästykö, Wikiä siirretään tuotantopalvelimelle
#wikilinux 4 +itn
/END of list
```

Tworzenie kanału i wchodzenie na już istniejący

Tworzenie kanału jest banalnie proste - o ile dany kanał jeszcze nie istnieje piszemy komendę:

```
/join #nazwa_kanału
```

co powoduje jednocześnie jego utworzenie na serwerze i od razu nasze wejście na ten kanał. Oczywiście jeśli kanał już istnieje komenda **/join** powoduje po prostu wejście na dany kanał, o ile jest on dla nas dostępny, lub uzyskanie informacji o odmowie wpuszczenia.

Zwykle tworzenie nowego kanału ma jakiś powód. Np: mamy grupę osób, którą informujemy w jakiś sposób, że tworzymy kanał na jakiś temat i czekamy aż przyjdą. Tworzenie kanału dla samego jego tworzenia jest zajęciem na ogół pozbawionym sensu. Gdy chcemy spotkać określoną grupę ludzi lub porozmawiać na jakiś określony temat lepiej jest poszukać już istniejącego kanału niż tworzyć nowy i czekać aż ktoś się pojawi.

Po wejściu na kanał, który wcześniej nie istniał jesteśmy, przynajmniej początkowo, na nim całkowicie sami. Wchodząc na nieistniejący kanał serwer daje nam na nim automatycznie przywileje operatora, którymi później możemy (lub nie) dzielić się z innymi wg naszego uznania.

Netykieta wchodzenia na kanał

Przy wchodzeniu na już istniejący kanał wypada powiedzieć "cześć". Zwykle dłuższe powitanie, czy przesadnie długie przedstawianie swojej osoby jest odbierane przez uczestników kanału jako dowód na tzw. lamerstwo. Na kanałach gdzie jest wielu użytkowników, ciągle przedstawianie się nowo-wchodzących jest zbyt uciążliwe. Po wejściu na kanał warto też po powiedzeniu "cześć" pomilczeć chwilę, aby zorientować się o czym aktualnie jest mowa. Przerwywanie rozmów innych lub wyskakiwanie "jak Filip z Konopi" jest zwykle negatywnie odbierane przez innych użytkowników.

Temat

Temat (ang. topic) to tekst przypisywany do kanału, który pojawia się po wejściu na niego, a w wielu klientach IRC jest on też automatycznie wstawiany do belki tytułowej okna, w którym jest wyświetlany dany kanał. Oprogramowanie serwerów IRC ogranicza zwykle długość tekstu tematu do 128 lub 256 znaków. Mogą to być zupełnie dowolne znaki.

W zależności od trybu działania kanału temat może zmieniać każdy użytkownik lub tylko użytkownicy ze statusem operatora kanału. Robi się to komendą:

`/topic #nazwa_kanału Tekst`

Opuszczanie kanału

Kanał można generalnie opuścić na 4 sposoby:

- dobrowolnie: poprzez napisanie komendy `/part #nazwa_kanału`, lub po prostu `/part` gdy aktualnie mamy aktywne okno z tym kanałem; podobny efekt daje też w większości popularnych klientów IRC zamknięcie okna z danym kanałem
- przymusowo: na skutek wykonania przez jednego z operatorów komendy `/kick`, która wyrzuca nas z kanału - po wyrzuceniu można wrócić komendą `/join #nazwa_kanału`, o ile nie jest się na tym kanale zbanowanym
- przymusowo lub dobrowolnie: na skutek zerwania łącza z serwerem
- przymusowo: na skutek splitu (zaniku łączności między serwerami IRC).

Uwagi na temat wychodzenia z kanału

Po zerwaniu łącza nasz nick jest widoczny przez jakiś czas na liście użytkowników wszystkich kanałów, na których się przebywało, co nazywane jest "zamienieniem się w ducha"; dlatego na ogół zaleca się nie wyłączać programu do IRC zanim nie opuści się wszystkich kanałów; częste wychodzenie z pozostawieniem "ducha" jest uważane za naruszenie netykiety IRCa i dowodem na ogólny brak wychowania, podobnie jak wychodzenie z kanału bez żadnego pożegnania.

Opuszczenie kanału przez ostatniego jego użytkownika powoduje, że kanał "znika" przynajmniej, do czasu aż ktoś ponownie na niego nie wejdzie. Serwer nie zapamiętuje historii kanału. Gdy kanał "znika", serwer całkowicie "zapomina" o jego dawnym istnieniu. Nie ma więc możliwości automatycznego odzyskania statusu operatora kanału, który się opuściło, nawet gdy wcześniej taki status się miało, w sytuacji, gdy ktoś w międzyczasie dany kanał ponownie utworzy. W niektórych sieciach IRC istnieją specjalne usługi, umożliwiające rejestrację kanału, takie jak np: "chanserv". W sieciach, w których tej usługi nie ma, jedynym sposobem "utrzymania" kanału jest pozostawienie na nim grupy wzajemnie wspierających się botów (programów pilnujących kanału), działających z kilku różnych serwerów.

Tryby kanału

Oprogramowanie serwerów IRC umożliwia ustawianie tzw. trybów kanału. Prawo ich zmiany posiadają wyłącznie operatorzy kanału. Tryb kanału zmienia się za pomocą komendy `/mode`:

`/mode #nazwa_kanału +parametry -parametry`

Parametry przyjmują postać jednoliterowych skrótów. Znak "+" oznacza włączenie trybu, znak "-" wyłączenie. Komenda umożliwia jednoczesną zmianę wszystkich parametrów na

raz lub tylko jednego. Użycie tej komendy bez żadnych parametrów wyświetla aktualny tryb kanału.

Lista trybów

- **publiczny** - jest to standardowy tryb kanału zaraz po jego utworzeniu - dozwolone jest na nim wówczas wszystko, jest on widoczny na liście kanałów, także dla tych którzy na nim nie przebywają, lista użytkowników kanału jest w pełni jawna
- **prywatny** - **+p** - lista osób na nim przebywających staje się "tajna" dla osób na nim nie przebywających - nie można np: wyświetlić listy osób przebywających na tym kanale za pomocą komendy **/whois**
- **sekretny** - **+s** - kanał jest niewidoczny w ogólnodostępnym spisie kanałów, ponadto po użyciu komendy **/whois nick** - informacja o tym, że ktoś przebywa na tym kanale jest ukrywana
- **No external messages** - **+n** - na kanał nie można wysyłać komunikatów komendą **/notice #kanał** gdy się na nim nie przebywa
- **zablokowany topic** - **+t** - topic mogą zmieniać wyłącznie operatorzy kanału
- **moderowany** - **+m** - na takim kanale mówić mogą tylko operatorzy oraz użytkownicy, którzy dostaną od operatorów głos, co jest zaznaczane na liście użytkowników kanału przez flagę "+"
- **na zaproszenie** - **+i** - na taki kanał można wejść tylko po zaproszeniu otrzymanym komendą **/invite #nazwa_kanału nick**, którą wydawać mogą operatorzy kanału
- **ograniczony** - **+l liczba użytkowników** - na kanale na raz może przebywać tylko określona liczba użytkowników; gdy zostanie osiągnięta kolejni użytkownicy nie mogą wejść na kanał do momentu aż ktoś go nie opuści
- **na hasło** - **+k hasło** - wejście na kanał wymaga podania hasła; robi się to komendą **/join #nazwa_kanału hasło**.

Przykłady

```
/mode #wikipedia +pstm-i
```

kanał #wikipedia stanie się prywatny, sekretny, z zablokowanym topicem, moderowany i jednocześnie przestanie być na zaproszenie (o ile wcześniej był)

```
/mode #wikipedia +l 120 +k kasztany_na_placu_Pigalle
```

kanał #wikipedia zostanie ograniczony do 120 użytkowników, a wejście będzie możliwe po podaniu hasła kasztany_na_placu_Pigalle

Uwagi

- tryb **+s** włącza jednocześnie tryb **+p**
- tryb **+i** włącza tryb **+k** i wice-versa
- na wielu serwerach standardowym trybem po utworzeniu kanału nie jest "publiczny" lecz "+n" lub "+nt"

Oprócz tego komenda **/mode** umożliwia też kontrolę nad użytkownikami.

Nadawanie i odbieranie statusu operatora

Każdy operator kanału może podzielić się swoją władzą, przez nadanie statusu operatora innemu użytkownikowi. Nadawanie statusu dokonuje się za pomocą komendy:

```
/mode #nazwa_kanału +o nick
```

Za pomocą jednej komendy można nadać na raz status operatora maksymalnie trzem użytkownikom:

```
/mode #nazwa_kanału +ooo nick1 nick2 nick3
```

Wiele klientów IRC posiada alias tej komendy, który przybiera zwykle postać **/op nick**, gdzie **/op** = **/mode #nazwa kanału +o**. Alias umożliwia jednak zwykle nadanie statusu tylko jednemu użytkownikowi.

Odbieranie statusu operatora robi się dokładnie tę samą komendą, tylko ze znakiem "-":

```
/mode #nazwa_kanału -o nick
```

podobnie można odebrać status operatora maksymalnie trzem użytkownikom:

```
/mode #nazwa_kanału -ooo nick1 nick2 nick3
```

Wiele klientów IRC posiada alias tej komendy, który przybiera zwykle postać **/deop nick**, gdzie **/deop** = **/mode #nazwa kanału -o**. Alias umożliwia jednak zwykle odebranie statusu tylko jednemu użytkownikowi.

Nadawanie i odbieranie statusu operatora można z sobą mieszać. Np:

```
/mode #nazwa_kanału -o+o-o nick1 nick2 nick3 = /mode #nazwa_kanału -oo+o  
nick1 nick3 nick2
```

odbierze status nick1 i nick 3 a nada go nick2. Tego typu "tricki" nie są możliwe do wykonania za pomocą aliasów **/op** i **deop**.

Uwagi

Użytkownik, który dostanie status operatora staje się technicznie równorzędny operatorowi który mu ten status nadał. Nie istnieje zatem hierarchia operatorów i nie ma znaczenia kto komu nadał status operatora. Nadanie tego statusu wiąże się zatem z ryzykiem, że osoba, która go dostała od nas może "nielojalnie" odebrać ten status nam.

Banowanie i wyrzucanie

Użytkownicy mogą być wyrzuceni z kanału za pomocą komendy:

/kick #nazwa_kanału nick powód.

Komenda ta może być wydawana tylko przez operatorów kanału. Użytkownik wykopany z kanału może w każdej chwili nań wrócić komendą **/join #nazwa_kanału**, chyba że zostanie zbanowany. W momencie wykopania serwer wysyła wyrzuconemu użytkownikowi komunikat o tym fakcie, z powodem podanym w komendzie **/kick**. Podawanie tekstu powodu jest opcjonalne. Operator może wykopać użytkownika bez wpisywania w komendzie powodu.

Banowanie, czyli zablokowanie możliwości wejścia na kanał dla określonego użytkownika lub grupy użytkowników dokonuje się komendą:

/mode #nazwa kanału +b nick!user@host powód

gdzie **nick!user@host** to klucz definiujący albo tylko jednego użytkownika, albo całą ich grupę. W kluczach tych można używać tych samych znaków, co przy kluczach wyszukiwania kanałów. Po założeniu bana, użytkownik, który pasuje do podanych kryteriów dostanie przy próbie wejścia na kanał komunikat, że jest zbanowany, kto go zbanował oraz powód zbanowania podany w komendzie **/mode +b**. Podawanie powodu jest opcjonalne. Przykłady:

/mode #wikipedia +b polimerek

spowoduje zbanowanie użytkownika o nicku polimerek. Jeśli jednak polimerek rozłączy się z siecią IRC i wejdzie powtórnie pod innym nickiem to będzie mógł w ten sposób ominąć ten ban. Wystarczy, że zmieni w nicku choć jedną literkę: Np ten ban nie zadziała na nicka "polimerek_"

/mode #wikipedia +b *@*.pl

spowoduje zbanowanie wszystkich użytkowników łączących się z serwerem z domeny .pl. Ominięcie tego rodzaju bana jest dużo trudniejsze. Wymaga albo dostępu do usługi **cloak** albo zdalnego połączenia się z komputerem nie znajdującym się w domenie .pl i uruchomieniem na nim klienta IRC.

/mode #wikipedia +b *oli*

spowoduje zbanowanie wszystkich, którzy mają w swoim nicku tekst "oli" - a zatem "polimerek_", "olinka" oraz "wolinek" nie będą mogli wejść na kanał.

/mode #wikipedia +b *!*@*

Spowoduje, że dokładnie wszyscy będą zbanowani - czyli nikt nie będzie mógł wejść na kanał!

Użycie komendy:

/mode #kanał +b

bez dalszych parametrów wyświetla listę aktualnych banów na danym kanale.

Przykład listy banów:

```
/mode #wikimedia +b
#wikimedia *!*@scrod.soc.lib.md.us set by Essjay!i=Essjay@wikimedia/Essjay
(Wed Jun 07 08:57:43)
#wikimedia *!n=cmaguina@201.230.235.* set by
Headcrab!n=aschenck@wikipedia/Linuxbeak
(Thu May 25 04:40:13)
#wikimedia *!*@adsl-68-89-130-209.dsl.snantx.swbell.net set by
Linuxbeak!n=aschenck@wikipedia/Linuxbeak (Tue May 23 03:24:35)
#wikimedia End of Channel Ban List
```

Serwery ograniczają zwykle liczbę możliwych banów, które można założyć na jednym kanale do pewnej liczby, zwykle 16 lub 32. Zdjęcie bana wymaga użycia dokładnie takiej samej komendy jak do zbanowania (z dokładnie tym samym kluczem), tyle że z parametrem **-b**:

```
/mode #wikipedia -b *oli* usuwa bana założonego komendą /mode #wikipedia +b *oli*
```

nawet drobna różnica w kluczu bana powoduje, że komenda **/mode #kanał -b** - nie zadziała.

Skutki lagów i splitów - czyli desynchronizacja kanałów

Dla osób, które znalazły się "po drugiej stronie", split wygląda jakbyśmy opuścili nagle kanał. Dla nas i osób które zostały się "po naszej stronie" splitu wygląda to z kolei tak, jakby to tamte osoby wyszły z kanału, a my na nim nadal przebywamy, tyle że w mniejszym gronie. Po zakończeniu splitu, nasz powrót na kanał wygląda tak samo jak wejście nań komendą **/join**. Jedyne co informuje, że nastąpił split a później jego zakończenie jest nagłe masowe opuszczenie lub wejście całej grupy użytkowników.

Split powoduje też desynchronizację kanału. Często w wyniku splitu część operatorów kanału zostaje się po jednej a druga część po drugiej stronie splitu. Jeśli operatorzy po jednej stronie zmienią topic, tryb kanału, odbiorą lub nadadzą komuś status operatora, po zakończeniu splitu serwery mają poważny problem jak scalić kanał.

Różne wersje serwerów radzą sobie z tym na różne sposoby. Najprostszym rozwiązaniem jest zapamiętywanie przez serwery stanu kanału sprzed splitu i doprowadzenie do jego powrotu przez cofnięcie efektów wszystkich komend, które zostały wykonane przez operatorów w czasie jego trwania. Serwery przywracają więc tryb kanału, odbierają statusu operatora, tym którzy go dostali w czasie splitu i nadają go na powrót tym którzy go w trakcie splitu utracili. Ten mechanizm działa jednak tylko przy pojedynczym splicie. Gdy następuje seria splitów i ponownych odtworzeń połączeń serwerów, serwery często nie nadążają z synchronizowaniem kanałów. W efekcie stan kanału, czyli jego tryby i lista operatorów mogą być po serii splitów zupełnie różne na różnych serwerach jednej sieci.

Podobne efekty desynchronizacji mogą też następować w czasie długich lagów. Lagom podlegają bowiem nie tylko komunikaty wysyłane na kanał ale także realizacja komend. Jeśli zatem istnieje długi lag między dwoma serwerami, operatorzy kanałów działający z jednego serwera mogą wykonywać komendy, które są sprzeczne z komendami operatorów działających z drugiego serwera, co w efekcie powoduje, że stan kanału na obu serwerach staje się różny.

Ponowne zsynchronizowanie kanału możliwe jest na trzy sposoby:

- użytkownicy kanału zgodnie, w jednym momencie decydują się na jego opuszczenie - jest to jednak dość ryzykowne, gdyż jest świetną okazją do jego przejęcia
- operatorzy działający z różnych serwerów wykonują w skoordynowany sposób serię komend, która uzgadnia stan kanału na wszystkich serwerach - wymaga to oczywiście wzajemnego zaufania operatorów do siebie
- na kanał wchodzi ircop, który czasowo odbiera wszystkim status operatora i przywraca odpowiedni stan kanału.

Użytkownicy

Użytkownikiem sieci IRC staje się każdy, kto zdoła się podłączyć swoim klientem do jednego z serwerów.

Serwery przechowują podstawowe informacje takie jak adres komputera, z którego użytkownik działa, jego nick, tryb działania, lista kanałów na których przebywa, czas wejścia na serwer i czas od wysłania do sieci ostatniego komunikatu. Informacje te są jawne i każdy użytkownik może się z nimi zapoznać wysyłając do serwera odpowiednią komendę-zapytanie. Użytkownicy nie mają bezpośredniego wpływu na te informacje (oprócz zmiany nicka) i nie mogą ich sami usuwać lub modyfikować.

Część informacji o użytkowniku jest przechowywana i udostępniana (lub nie) przez jego klienta IRC. Większość klientów pozwala w prosty sposób modyfikować te informacje i zezwalać innym użytkownikom na ich pobieranie lub nie pobieranie. Do informacji tych zalicza się:

- "username" - czyli nazwa użytkownika, a dokładnie to co występuje przed znakiem @ w adresie użytkownika, który łączy się z serwerem IRC; standardowo jest to nazwa konta użytkownika pobrana z jego systemu operacyjnego; można jednak tak skonfigurować klienta IRC aby podawał serwerowi w to miejsce dowolną, inną nazwę
- "version" - czyli informacja o używanym kliencie i systemie operacyjnym, w którym działa
- "time" - czas systemowy użytkownika

i kilka innych

Pozyskiwanie informacji o jednym użytkowniku

Podstawową komendą umożliwiającą pozyskanie informacji o danym użytkowniku jest:

```
/whois nick
```

Po wysłaniu tej komendy do serwera, zależnie od jego wersji otrzymuje się różną postać odpowiedzi, zawiera ona jednak zazwyczaj: adres komputera użytkownika, nazwę użytkownika (wpisaną w odpowiednim miejscu do klienta), informację o serwerze z którego korzysta, czasie zalogowania, czasie wysłania ostatniego komunikatu (*idle time*), nie-sekretnych kanałach na których przebywa oraz trybie przyznanym przez serwer. Przykładowy efekt działania:

```
/whois poli_  
poli_ is n=Poli_@wikipedia/Polimerek * Tomek G.  
poli_ on #wikimedia, #wikimedia-pl @#wikipedia-pl  
poli_ using clarke.freenode.net Fremont, California, US  
poli_ is identified to services  
poli_ has been idle 15mins 13secs, signed on Tue Jun 13 19:42:59  
poli_ End of /WHOIS list.
```

Whois wyświetla tylko informację przechowywaną na serwerze i nie powoduje kontaktowania się serwera z klientem badanej osoby. O fakcie wykonania komendy **/whois nick** sam badany nie jest w żaden sposób informowany przez serwer, stąd nie istnieje ryzyko, że dowie się on, że go w ten sposób "sprawdzamy". Większość informacji z whois jest automatycznie generowana przez serwer i użytkownik nie ma żadnego wpływu na jej kształt.

Część informacji jest jednak pobierana przez serwer z klienta w momencie, gdy się na niego logujemy (nick, username i adres internetowy naszego komputera) możemy ją więc zmodyfikować zmieniając ustawienia naszego klienta zanim się połączymy z serwerem. Po uzyskaniu połączenia, jedyny sposób dokonania zmian tych danych to rozłączyć się, zmienić ustawienia klienta i ponownie się zalogować do niego. Wyjątek stanowi tylko zmiana nicka.

Pozyskiwanie list użytkowników

Komenda **/who** daje możliwość wyświetlenia listy użytkowników spełniających określone kryterium. Jest to możliwe przy zastosowaniu klucza wyszukiwania, którego składnia jest taka sama jak we wcześniej opisanych komendach **/list**, **/mode +b** itp.

Np, jeśli chcemy wyświetlić listę wszystkich użytkowników, którzy działają spod adresów w domenie internetowej "pl" piszemy:

```
/who *.pl
```

W wyniku czego otrzymamy listę wyglądającą mniej więcej tak:

```
#koffice mike_ H n=mike@uf122.internetdsl.tpnet.pl :0 Michal Sciubidlo
```

```
#tto obi_1 H@ n=obi_1@nat.henet.pl :0 ask
#cakephp kabturek__ H i=famuhu@nat-6.aster.pl :0 kabturek
#worldwind m_k H i=Michal@dsx2.neoplus.adsl.tpnet.pl :0 Michal Kursa
#Azureus-Polish toudim5 H n=OzPZQiAN@mie3.1000lecie.pl :0 Azureus IRC
#ubuntu-pl Cyr4x H n=Cyr4x@cn.ramtel.pl :0 Cyr4x
```

Jeśli chcemy wyświetlić listę wszystkich użytkowników, których nick zaczyna się na "po" piszemy:

/who po*

Komenda:

/who *

wyświetli nam listę wszystkich użytkowników danej sieci IRC. W dużych sieciach, gdzie na raz przebywa tysiące użytkowników wyświetlanie listy wszystkich nie jest jednak najlepszym pomysłem. Zwykle serwer w czasie wysyłania do nas takiej listy rozłączy się w pewnym momencie, z powodu przekroczenia dozwolonego limitu danych w jednostce czasu.

Komenda:

/names #nazwa_kanału

wyświetla listę nicków wszystkich użytkowników tego kanału, razem z informacją o ich statusie na kanale. W przypadku kanałów publicznych (nie działających w trybie +s lub +p) listę tę można uzyskać bez potrzeby wchodzenia na kanał. W przypadku kanałów sekretnych i prywatnych komenda jest, zależnie od rodzaju serwera albo ignorowana, albo zwracana jest pusta lista użytkowników (sprawiająca wrażenie jakby kanał nie istniał) lub wysyłana jest informacja z odmową wykonania komendy. Dla kanałów prywatnych i sekretnych komenda ta zadziała dopiero po tym jak wejdziemy na taki kanał.

W pewnym sensie podobnie działa komenda:

/who #nazwa_kanału

wynik **/names** od **/who** różni się jednak tym, że **/names** wyświetla krótką listę nicków w jednej linii, a **/who** listę nicków, przypisanych do nich nazw i adresów ich komputerów, przy czym w jednej linii tekstu znajduje się informacja o jednym nicku.

Przykład rezultatu komendy **/names**:

```
/names #wikipedia-pl
#wikipedia-pl @tsca alx-pl @WarX @MChrist Shaqspeare @poli_ Roman_92
Jedyooo A_Bach Ark-pl
@Maire Catz Stimoroll @Derbeth Herr_Kriss Malin @szwedzki @Lukas3 @Ausir
@ChanServ DoZerPL TOR_CNR jackal @DonnerJack
#wikipedia-pl End of /NAMES list.
```


Przykład rezultatu komendy **/who #nazwa_kanału**:

```
#wikipedia-pl tsca H@ n=tsca@port10.ds8-hj.adsl.cybercity.dk :0
[[m:user:tsca]]
#wikipedia-pl alx-pl H n=alx-pl@j272315.upc-j.chello.nl :0 Aleksy Mazan
#wikipedia-pl WarX H@ n=fizyk@h12-243-260-56-static.e-wro.net.pl :0 Unknown
#wikipedia-pl MChrist H@ n=MonteChr@vir.mezi2-net.com :0 Christ
#wikipedia-pl Shaqspeare H n=kemalata@HSI-KBW-185-216-453-
034.hsi.kabelbw.de :0 ggg
#wikipedia-pl poli_ H@ n=Poli_@wikipedia/Polimerek :0 Tomek G.
#wikipedia-pl Roman_92 H i=Roman@c63-246.icpnet.pl :0 Roman Izdebski
#wikipedia-pl Jedyooo H n=jedyooo@chello153207213040.chello.pl :0 Jedyooo
#wikipedia-pl A_Bach H n=name@ako74.internetdsl.tpnet.pl :0 A_Bach
#wikipedia-pl Ark-pl H n=chatzill@diw12.neoplus.adsl.tpnet.pl :0 New Now
Know How
#wikipedia-pl Maire H@ n=eigann@bby52.neoplus.adsl.tpnet.pl :0 New Now Know
How
#wikipedia-pl Catz H i=adam@host-87-93-45-112.lanet.net.pl :0 Przemysław
Okoń
#wikipedia-pl Stimoroll G i=masq258@ropaktor.net :0 New Now Know How
#wikipedia-pl Derbeth G@ i=tomasz2@neptun.wzn.pl :0 Derbeth
#wikipedia-pl Herr_Kriss H i=java@bxo46.neoplus.adsl.tpnet.pl :0 moonspell
#wikipedia-pl Malin H n=Malin@priv5.isatel.com.pl :0 malin
#wikipedia-pl szwedzki H@ n=suidisz@dvc100.neoplus.adsl.tpnet.pl :0
szwedzki
#wikipedia-pl Lukas3 H@ n=chatzill@auu11.neoplus.adsl.tpnet.pl :0 New Now
Know How
#wikipedia-pl Ausir H@ n=Ausir@sk-44.mikolow.net :0 Ausir
#wikipedia-pl ChanServ H@ ChanServ@services. :0 Channel Services
#wikipedia-pl DoZerPL H i=dozer@NAT-7.autocom.pl :0 Damian
#wikipedia-pl TOR_CNR G i=tor@wikipedia/TOR :0 TOR
#wikipedia-pl jackal H i=jackal@rione.es :0 jackal
#wikipedia-pl DonnerJack H@ n=A_Bach@gw2-gts.polanet.pl :0 Lord Bachus
bocik
#wikipedia-pl End of /WHO list.
```

Prywatna komunikacja między użytkownikami

Oprócz możliwości rozmowy z innymi użytkownikami na kanałach IRC oferuje też możliwość prowadzenia z nimi bezpośrednich rozmów poza kanałami. Podstawową komendą służącą do tego celu jest:

/msg nick Tekst komunikatu

Tekst komunikatu wysłany tą komendą trafia bezpośrednio do danej osoby i nie widzą go inni użytkownicy. W zależności od konfiguracji klienta komunikat ten pojawia się w osobnym oknie lub w oknie aktualnie aktywnego kanału. Gdy pojawia się w aktywnym oknie kanału, odróżnia ją od tekstów widocznych dla wszystkich na kanale znak "->" za którym znajduje się nick osoby, która ten komunikat wysłała. Np po wysłaniu przez osobę o nicku "janek" komunikatu do kazia:

/msg kaziu Witam kolege

zobaczy on:

->*janek* Witam kolegę

Ze względu na to, że ciągle kontaktowanie się przez wypisywanie komendy **/msg nick komunikat** jest uciążliwe istnieje komenda:

/query nick

która automatycznie otwiera nowe okno lub panel w niemal każdym kliencie. Okno to przypomina swoją strukturą okno kanału, tyle że jest przeznaczone tylko dla dwóch osób. Pisząc w tym oknie nie ma już potrzeby poprzedzania komunikatów jakąkolwiek komendą. Gdy klient jest tak skonfigurowany, że osobne okno jest otwierane również po otrzymaniu komendy **/msg** różnica między **/query** i **/msg** jest nieodczuwalna. Jeśli jednak nie wiemy jak skonfigurowany jest klient naszego rozmówcy a chcemy, aby u niego otworzyło się osobne okno lub panel rozmowy prywatnej, należy użyć komendy **/query**.

W pewnym sensie podobnie do **/query** zadziałałoby utworzenie nowego kanału z trybem **"+si"** (sekretny i wymagający zaproszenia) i zaproszenie nań osoby z którą chcemy porozmawiać, jest to jednak w sumie bardziej kłopotliwe.

Wszystkie komunikaty wysyłane komendami **/msg** i **/query** przechodzą przez serwer, może je zatem podsłuchać administrator serwera. Ponadto są one tak samo zależne od splitów i lagów jak rozmowy prowadzone na kanałach. Jeśli zależy nam na szybkości i poufności bezpieczniej jest użyć komendy **/dcc chat**. Jej użycie jest dokładniej wyjaśnione w rozdziale ["Usługi DCC i CTCP"](#).

Zmiana nicka

Normalnie, zaraz po połączeniu się serwerem pobiera on z naszego klienta nasz nick, który wpisaliśmy do niego w ramach procedury jego konfiguracji. Jeśli jednak chcemy z jakiegoś powodu zmienić nick możemy to zrobić bez rozłączania się serwerem i zmiany konfiguracji klienta. Można to zrobić za pomocą komendy:

/nick nowy_nick

Aby komenda ta zadziałała **"nowy_nick"** musi być wolny, tzn. w danym momencie nie ma nikogo innego kto ten nick używa. Jeśli ktoś taki jest, dostajemy komunikat z serwera informujący nas o tym fakcie i musimy wybrać jakiś inny nick. Nick można zmienić w każdej chwili i tyle razy ile się chce. Częste zmiany nicka są jednak uciążliwe dla innych użytkowników kanału i dlatego za zbyt częste zmiany można nawet dostać bana.

W przypadku serwerów opartych na oprogramowaniu wywodzącym się z DALnetu możemy w momencie zmiany nicka dostać ostrzeżenie, że jest on już zarejestrowany, nawet jeśli aktualnie jego "właściciel" jest nieobecny. Nicka tego zwykle możemy mimo to używać, ale tylko do czasu pojawienia się jego prawowitego właściciela. Osoby, które go/ją znają mogą jednak odebrać negatywnie sam fakt posługiwania się cudzym nickiem, lepiej jest go więc po takim komunikacie zmienić. Sposoby rejestrowania nicka w sieciach o architekturze DALnetu są dokładnie opisane w rozdziale ["Dodatkowe serwisy"](#).

W sieciach opartych na oprogramowaniu IRCnetu nicki nie są rejestrowane. Jeśli jednak serwer przyznał nam tryb "secure" mamy całkowicie zablokowaną możliwość użycia komendy "/nick" i zmiana nicka wymaga rozłączenia się z siecią IRC. Jakkolwiek w IRCnetcie nicki nie są rejestrowane może zdarzyć się, że ktoś czuje się właścicielem danego nicka i jeśli go będziemy używać możemy się spodziewać agresywnych reakcji właściciela tego nicka i jego znajomych. Najlepiej jest w tej sytuacji zastosować zasadę "ustępowania głupiemu" i np. zmienić nasz nick o jedną literę lub dodać na jego końcu "_" lub inny tego rodzaju znak.

Tryby użytkowników

Użytkownikom IRC przydzielane są tzw. tryby działania. Tryby te dzielą się na globalne, związane z relacjami z serwerem i tryby kanałowe, związane z uprawnieniami na kanale. Jeden użytkownik może mieć różne tryby kanałowe na różnych kanałach nie może mieć jednak różnych trybów serwerowych, przynajmniej w ramach jednej sieci IRC.

Tryby globalne

Tryby globalne są zazwyczaj automatycznie przydzielane przez serwer w momencie łączenia się z nim. Zależnie od konfiguracji serwera tryby te można sobie w pewnym zakresie samodzielnie zmieniać lub nie. Jedni użytkownicy nie mogą zmieniać trybów globalnych innym. Wyjątkiem są tylko IRCopowie, którzy mogą to robić.

Do zmiany własnego trybu globalnego służy komenda:

/mode nick parametry

gdzie "parametry" to jednoliterowe skróty trybów poprzedzone znakiem "+" (włączenie) lub "-" (wyłączenie). Użycie tej komendy bez parametrów:

/mode nick

wyświetli nam informację o naszych aktualnych trybach globalnych.

Niestety skróty trybów globalnych stosowanych przez różne wersje oprogramowania serwerów różnią się dość zasadniczo. Do najważniejszych z punktu widzenia zwykłego użytkowników trybów zaliczyć można:

- tryb IRCopa: **+a** - daje on "superuprawnienia" tożsame z uprawnieniami administratora serwera, przydzielić może go tylko inny IRCop
- tryb *invisible*: **+i** - powoduje on, że nasz nick nie będzie wyświetlany na listach generowanych komendą **/who klucz** ani komendą **/names** adresowaną do kanału, na którym osoba ją wysyłająca nie przebywa; tryb ten nie chroni jednak przed wyświetleniem informacji o nas komendą **/whos nick** oraz **/names** w stosunku do kanału na którym przebywa osoba ją wysyłająca
- tryb *secure*: **+s** - istnieje ona na serwerach z oprogramowaniem wywodzącym się z IRCnetu; jest on zwykle przydzielany użytkownikom działającym z "niedzwolnych adresów" (np: z dynamicznych IP, adresów bez zwrotnego DNS, adresów spoza

określonego zakresu "zaufanych"); tryb ten powoduje, że nie użytkownik nie może wykonywać pewnych komend, zwykle "mode, nick i msg"

- tryb *silent*: **+s** - istnieje ona na serwerach z oprogramowaniem wywodzącym się z DALnetu i Undernetu i oznacza blokadę automatycznego wysyłania większości komunikatów systemowych serwera, co oszczędza łącza, a większość zwykłych użytkowników i tak nie czyta tych komunikatów
- tryb *registered*: **+r** - istnieje on na serwerach z oprogramowaniem wywodzącym się z DALnetu i Undernetu; jest on przydzielany użytkownikom, którzy zarejestrowali na serwerze swój nick lub adres swojego komputera; brak włączonego trybu **+r** daje zwykle podobny efekt jak tryb **+s** na serwerach IRC.

Tryby kanałowe

Istnieją trzy tryby kanałowe:

- zwykłego użytkownika
- użytkownika z prawem do głosu : **"v"** (ma to sens tylko na kanałach moderowanych)
- użytkownika z prawami operatora : **"o"**

Tryby kanałowe nadaje się komendą:

```
/mode #nazwa_kanału +o (lub +v) nick
```

jak to zostało dokładniej wyjaśnione w rozdziale "Kanały"

Sposób oznakowywania trybów na listach użytkowników

Listy użytkowników generowane komendami **/who** i **/names** zawierają specjalne znaki symbolizujące określone tryby, zarówno globalne jak i kanałowe.

Tryby są oznaczane różnie w różnych wersjach serwerów, ale większość z nich stosuje następującą symbolikę:

* - oznacza IRCopa

@ - oznacza operatora kanału

+ - oznacza użytkownika z prawem do głosu na kanale moderowanym

! - oznacza użytkownika "ducha", który przez określony, długi czas nie wysłał żadnego komunikatu na serwer

~ - (bezpośrednio przed adresem internetowym) oznacza użytkownika działającego z nierozpoznanego adresu lub niezarejestrowanego (z trybem **+s** w IRCnecie lub z trybem **-r** w DALnecie)

n/i= (bezpośrednio przed adresem internetowym) - oznacza użytkownika działającego w trybie normalnym (n) lub "niewidzialnym" (*invisible +i*)

G/H - oznacza odpowiednio użytkownika "uśpionego" komendą **/away** (G = *gone*) lub aktywnego (H - *here*)

Inne, podstawowe komendy

Oprócz podstawowych komend służących do obsługi kanałów, użytkowników i łączności z serwerem, istnieje też wiele innych dodatkowych komend, których znajomość nie jest konieczna, ale może czasem się przydać.

/Notice i /me

/Notice i /me to dodatkowe komendy służące do wysyłania komunikatów na kanał lub do konkretnego użytkownika.

Komenda

/me #kanał tekst komunikatu

wysyła na dany kanał komunikat, który wygląda nieco inaczej od zwykłego tekstu na kanale. Komunikat "me" jest wyróżniany znakiem "*" i zaczyna się od nicka osoby wysyłającej. Np komenda wysłana przez użytkownika o nicku "kazio":

/me #kanał wita kolegów

wyświetli się na tym kanale jako:

```
* kazio wita kolegów
```

gdy wysyłamy tę komendę z aktywnego okna danego kanału można pominąć w niej nazwę tego kanału. Oprócz wyróżnienia gwiazdką i nickiem wiele klientów wyświetla teksty komunikatów wysyłanych komendą **/me** innym kolorem niż zwykłe teksty lub pogrubione. Ponadto, na kanały działające w trybie publicznym można wysyłać w ten sposób komunikaty bez wchodzenia na nie.

Działanie komendy

/notice nick tekst komunikatu

jest zbliżone do komendy **/msg**, występują tu jednak trzy różnice:

- tekst komunikatu zawsze pokazuje się w aktualnie aktywnym oknie użytkownika do którego się ten tekst wysyła - nie uruchamia ona zatem u niego żadnego nowego okna
- tekst komunikatu jest wyróżniony znakiem "-" lub "*" i większość klientów wyróżnia go innym kolorem lub pogrubieniem podobnie jak to ma miejsce w przypadku komendy **/me**

Jest to więc w sumie komenda służąca do wysyłania, krótkich ale ważnych ostrzeżeń do innych użytkowników.

/away

Komenda:

/away powód

przełącza nas w stan "nieobecności". Stosujemy ją gdy pozostajemy połączeni z siecią IRC ale nie śledzimy na bieżąco tego co się tam dzieje, bo np: pracujemy lub wyszliśmy akurat zrobić sobie herbatę. W efekcie tej komendy:

- nasz klient wyśle na kanał komunikat informujący innych o naszym przejściu w stan spoczynku (tę opcję można w kliencie zwykle wyłączyć)
- gdy ktoś sprawdzi nas komendą **/whois nasz_nick** otrzyma informację o tym, że jesteśmy "away" z powodem podanym w tej komendzie
- na listach użytkowników generowanych komendami **/names** i **/who** przy naszym nicku będzie zaznaczone, że jesteśmy "away" (zwykle przy pomocy dużej litery "G" (od *gone*))
- gdy ktoś będzie chciał z nami porozmawiać prywatnie za pomocą komend **/msg**, **/query** a także będzie się z nami próbował łączyć komendami z rodziny **dcc** otrzyma informację, że jesteśmy "away" z podaniem powodu zamieszczonego w komendzie.

Powrót do stanu aktywnego wykonuje się komendą:

/away

bez żadnych parametrów.

Niektóre klienty IRC da się tak skonfigurować, aby automatycznie przenosiły nas w stan "away", gdy odpowiednio długo nie wykonujemy żadnych działań oraz automatycznie przywracały nas w stan aktywny gdy tylko podejmiemy jakieś działania. Niektóre klienty IRC posiadają też własne, niezależne od mechanizmów serwera metody obsługi "uśpienia" i "obudzenia", polegające np: na wysyłaniu ozdobnych, kolorowych tekstów na kanał i długich wyjaśnień w momencie, gdy ktoś do nas wysła komunikat prywatny. Kolorowane i przesadnie rozbudowane komunikaty "away" są jednak na większości kanałów uważane za naruszenie netykiety, a użytkownika stosującego je może nawet spotkać zbanowanie.

/whowas, /notify, /ignore, /ison

Komenda:

/whowas nick

działa z grubsza tak samo jak komenda **/whois** lecz dotyczy ona osoby, która opuściła już sieć IRC. Jej działanie jest silnie zależne od konfiguracji serwera. Niektóre serwery wogóle

nie obsługują tej komendy, gdyż "zapominają" wszelkie informacje o każdym użytkowniku zaraz po tym jak się rozłączy, inne z kolei pamiętają opuszczających je użytkowników przez kilka-kilkadziesiąt minut. W czasie gdy serwer jeszcze pamięta dane użytkownika wyświetli nam on je po wykonaniu tej komendy w sposób identyczny jak dla komendy **/whois** zastosowanej do użytkownika aktualnie podłączonego do sieci.

Komenda:

/notify nick

powoduje, że serwer będzie nas informował specjalnym komunikatem o każdym wejściu i wyjściu z sieci użytkownika posługującego się tym nickiem. Komenda **/notify** może być też użyta z kluczem wyszukiwania - z kluczami trzeba tu jednak postępować ostrożnie, gdyż klucz, którego kryteria spełni zbyt wielu użytkowników spowoduje, że serwer będzie nas zalewał komunikatami o wejściach i wyjściach setek użytkowników. Wiele klientów IRC posiada własne systemy obsługi komunikatów o pojawianiu się i odchodzeniu użytkowników, które są niezależne od mechanizmu "notify" serwera - działają one jednak zwykle tylko w stosunku do kanałów, na których aktualnie się znajdujemy.

Aby usunąć dany nick z naszej listy "notify" musimy wykonać komendę:

/notify -nick

lub

/notify -klucz

Komenda:

/ignore nick powod

powoduje, że serwer przestanie wysyłać do nas jakiegokolwiek komunikaty (zarówno prywatne jak i wysyłane na kanał) od użytkownika posługującego się tym nickiem, a także nie będziemy go widzieli na listach użytkowników. Można powiedzieć, że praktycznie przestanie on dla nas wogóle istnieć. Komenda ta będzie działać nawet wtedy, gdy ten użytkownik zmieni nicka. Zobaczymy go dopiero wtedy, gdy się rozłączy i przyłączy do sieci ponownie z innym nickiem i to dopiero po czasie, gdy serwer zdąży "zapomnieć" już o jego istnieniu. Sam ignorowany zostanie o tym automatycznie poinformowany, komunikatem typu **notice**.

Jest to najprostszy i najskuteczniejszy sposób na osoby, które nam się naprzykrzają lub uprawiają trolling na naszym ulubionym kanale. Warto tylko pamiętać, że komenda ta działa tylko w stosunku do nas. Inni użytkownicy nadal będą widzieli jego komunikaty. Użytkownik ten będzie miał zatem możliwość "gadania za naszymi plecami" a my nie będziemy mieli możliwości dowiedzenia się o tym.

Komenda **/ignore** nie działa na wszystkich rodzajach serwerów, działa jednak na większości. Wiele klientów IRC posiada własny system obsługi ignorowania nie lubianych przez nas użytkowników, niezależny od serwera. Serwerowy "ignore" jest jednak najskuteczniejszy gdyż powoduje, że komunikaty ignorowanego użytkownika wogóle nie są do nas wysyłane i nie obciążają naszego łącza. Filtrowanie komunikatów przez klienta IRC odbywa się zaś na naszym komputerze, komunikaty te fizycznie do nas docierają, tyle, że ich nie widzimy.

W komendzie **/ignore** można także zamiast nicka, stosować klucz wyszukiwania, taki sam jak we wszystkich innych komendach ircowych akceptujących klucze. Aby przywrócić łączność z danym użytkownikiem musimy wykonać komendę:

/ignore -nick (lub -klucz)

Wreszcie wykonanie komendy **/ignore** bez żadnych parametrów wyświetli nam listę naszych aktualnie ignorowanych nicków i kluczy.

Komenda:

/ison nick1 nick2 nick3 nick4 ...

sprawdza czy w sieci, w danym momencie są osoby o podanych nickach.

Przydatne komendy dotyczące serwera

Jeśli chcemy szybko znaleźć kontakt z jednym z administratorów serwera z którego korzystamy możemy wykonać następujące komendy:

/admin - zawiera zwykle albo listę administratorów/ircopów albo informacje kontaktowe do nich

/motd (*message of the day*) - zawiera "przesłanie" do użytkowników, które informuje z jakiego serwera się korzysta, do jakiej sieci jest przyłączony i jakie zasady tu panują, czasami listę administratorów oraz rozmaite inne rzeczy, które administrator uzna za właściwie tam umieścić.

/info - zawiera zazwyczaj podstawowe informacje o serwerze, sieci i administratorach

Jeśli chcemy uzyskać dane statystyczne o serwerze i sieci, z której aktualnie korzystamy warto spróbować następujących komend:

/stats - wyświetla podstawową statystykę serwera - zawartość odpowiedzi na tę komendę jest bardzo różna na różnych serwerach, gdyż decyduje o tym jego administrator

/users - wyświetla informację o liczbie użytkowników aktualnie korzystających z sieci i serwera i o liczbie istniejących kanałów z rozbiciem na ich rodzaje

/links - wyświetla listę serwerów, które aktualnie są połączone z naszym serwerem; Gdy nie ma splitów daje to po prostu listę wszystkich serwerów w danej sieci.

Specjalnym rodzajem komendy, jest komenda:

/raw tekst

Komenda ta wysyła na serwer tekst, który ten analizuje bezpośrednio w sposób zaprogramowany przez administratora. Daje to możliwość administratorowi tworzenia dodatkowych komend, które standardowo nie są zawarte w gotowym oprogramowaniu serwera. Listy tych specjalnych komend są często umieszczane na oficjalnych stronach WWW serwera lub sieci. Bardzo często implementowaną w ten sposób komendą jest:

/raw help

która np: wyświetla kompletną listę "normalnych" komend akceptowanych przez serwer.

Na do widzenia

Komenda:

/quit powód

powoduje w zależności od konfiguracji klienta albo rozłączenie się z serwerem i dalsze "jałowe" funkcjonowanie klienta lub rozłączenie się z serwerem i zamknięcie klienta. Tekst powodu wyświetli się w komunikacie o naszym wyjściu, który pojawi się na wszystkich kanałach, na których byliśmy w momencie wykonania tej komendy.

Wiele klientów jest tak skonfigurowana, że wykonuje tę komendę automatycznie w momencie ich wyłączenia. Mają one też możliwość ustalania standardowego tekstu pożegnania, który pokaże się na kanałach. Niektóre klienty mają jednak brzydki zwyczaj po prostu się wyłączać bez wysłania do serwera komendy "/quit". Powoduje to, że serwer "nie dowiaduje się" o tym, że się rozłączyliśmy i nasz nick jest wciąż widoczny na listach. Stajemy się tzw. "duchem".

Serwery pozbywają się duchów dzięki mechanizmowi zrywania połączenia z użytkownikami, którzy nie wykonali żadnej akcji przez określony czas. Czas od wykonania ostatniej akcji przez użytkownika nazywa się *idle time*. Mechanizm ten może też spowodować rozłączenie nas gdy nadal mamy włączonego klienta IRC, ale z niego nie korzystamy. Aby nas uchronić przed takim zdarzeniem wiele klientów wysyła na serwer od czasu do czasu (zwykle co 2 minuty) komendę:

/ping

na którą serwer odpowiada **pong** i w ten prosty sposób zapewnia, że serwer nie uzna nas za ducha. Niektóre serwery ignorują jednak tę komendę i liczą *idle time* od ostatniej akcji rzeczywiście podjętej przez użytkownika. Niektóre klienty IRC potrafią to ominąć poprzez zastąpienie wysyłania "pinga" wysyłaniem innej, nie robiącej niczego widocznego dla innych użytkowników komendy np: "/notice mój_własny_nick Hello".

Netykieta i społeczności

IRC jest dobrym miejscem na wyjście z "samotności" i znalezienie sobie grupy stałych przyjaciół. Wokół wielu popularnych kanałów, na które wchodzi często wiele osób wytwarza się zwykle rodzaj społeczności, która zazwyczaj tworzą swoją własną subkulturę i specyficzne obyczaje. Subkultura ta obejmuje zwykle obowiązek przestrzegania często niepisanych zasad netykiety, używania określonych klientów IRC z odpowiednimi skryptami, określonego sposobu zachowania i pisania. Przechodząc od kanału do kanału i z jednej sieci do drugiej, można się szybko zorientować, że tworzone przez nie subkultury różnią się znacząco między sobą, mają one jednak także wiele wspólnych cech.

Wiele dużych kanałów organizuje regularne spotkania na żywo, które dodatkowo integrują związane z nimi społeczności. Organizowane są nawet imprezy w rodzaju "wybór miss kanału", "wybór miss IRCa", "mikołajki IRCowe" itp. Uczestnictwo w społecznościach IRCowych wymaga jednak w gruncie rzeczy takich samych zdolności towarzyskich jak w normalnym życiu.

W subkulturach ircowych powstaje często rodzaj hierarchii jej uczestników. Miejsce w hierarchii zależy od zaangażowania w organizowanie i nadzór nad kanałem, inteligencji emocjonalnej oraz kwalifikacji technicznych. Stosowanie klienta IRC, który jest trudny w obsłudze, umiejętność stawiania dobrze skonfigurowanych botów, dostęp do szybkich komputerów ze stałym IP i stabilnym systemem operacyjnym podnoszą zwykle w oczach innych status danego ircownika.

Netykieta

Netykieta obowiązująca na większości kanałów IRC jest zbliżona do ogólnej netykiety internetowej. Mogą w nią jednak być też czasami włączone pewne dodatkowe zasady, których łamanie może się wiązać ze zbanowaniem i wykluczeniem z danej społeczności. Zwykle zanim zaczniesz się aktywnie uczestniczyć w społeczności danego kanału warto się zapoznać z jego stroną WWW, na której zwykle można znaleźć ogólny [FAQ](#) (zbiór często zadawanych pytań i odpowiedzi na nie) kanału, informację o bardziej znanych użytkownikach itp. Oczywiście nie wszystkie kanały posiadają swoje strony WWW, jednak spora część dużych ma. Informacja o stronie często jest podawana na topicu lub w formie komunikatu wysyłanego przez specjalnego bota.

Ogólnie w większości sieci i na większości kanałów obowiązują następujące zasady:

- Należy się ogólnie zachowywać kulturalnie - tzn. nie pisać tekstów obscenicznych, nie bluźnić, nie wywoływać niepotrzebnych kłótni, nie molestować innych użytkowników przez query lub dcc chat jeśli sobie tego nie życzą.
- Nie wolno zazwyczaj wysyłać na kanał tzw. floodu - czyli serii komunikatów wysyłanych automatycznie jedna po drugiej - do floodowania zalicza się też często

uporczywe zadawanie ciągle jednego pytania - jeśli nikt na to pytanie nie chce odpowiedzieć, to znaczy zapewne, że nikt na nie zna odpowiedzi.

- Nie wolno też zazwyczaj wysyłać przez DCC plików do innych użytkowników, bez wcześniejszego uzgodnienia tego z nimi.
- Nie wolno stosować skryptów, które zachowują się uciążliwie dla innych użytkowników - np. mają rozbudowany system away, wysyłający co jakiś czas na kanał automatyczny, rozbudowany komunikat.
- Należy się dostosować do panującej na kanale normy kodowania polskich liter, lub pisać bez nich i nie narzekać, że się ich nie widzi w komunikatach innych.
- Nie wolno pisać samymi WIELKIMI LITERAMI, gdyż jest to odpowiednik krzyku w normalnej rozmowie.
- Nie należy wykradać innym ich nicka i się potem pod nich podszywać.
- W wielu sieciach nie wolno stawiać bota lub zostawiać zalogowanego klienta 24 godziny na dobę, bez wcześniejszego uzgodnienia tego z administratorem serwera; na wielu kanałach boty i "wieczne" sesje, które nie wiadomo po co są pozostawione na nim, są banowane.

Emotikony, zabawowe skrypty, ASCII-arty itp

Emotikony

Ważną częścią subkultury IRCowej jest stosowanie różnych form pisanej ekspresji emocji wychodzącej poza typowy tekst. Na IRCu masowo są stosowane [emotikony](#), czyli złożone ze znaków ASCII wyrazy nastroju, w postaci uproszczonego obrazu "buźki" z odpowiednią "miną":

- :-) uśmiech,
- ;-) uśmiech z przymrużeniem oka,
- !-) uśmiech z szelmowskim przymrużeniem oka,
- :-> uśmiech słodki lub złośliwość,
- :-(smutek, zmartwienie,
- ;-(płacz,
- :-p pokazanie języka,
- :-P jw.,
- ;-p jw.,
- :-* pocałunek
- :-o zdziwienie,
- :-O jw.,
- :-D śmiech,
- :-/ grymas niezdecydowania/zniesmaczenia,
- :-l obojętność, niezdecydowanie.

Krótkie komunikaty słowne i jednolinijkowe ASCII-arty, specyficzne dla IRC

IRC stanowi też wdzięczne poletko dla jeszcze nieopierzonych hakerów, którzy mogą "poćwiczyć" radzenie sobie z systemami zabezpieczeń programów bez narażania się na skutki prawne swoich "niedozwolnych" czynów. Nadzór nad kanałami, stawianie botów, pisanie własnych skryptów, walki o przejmowanie kanałów mają często charakter hakerskiej rywalizacji, w której wygrywa sprytniejszy i bardziej kompetentny programista.

Jeśli chcielibyśmy znaleźć "rasowego hakera" na kanałach dużych, popularnych sieci IRC może się to jednak okazać dość kłopotliwe. Najbardziej hałaśliwi i zauważalni użytkownicy kanałów, którzy często posiadają w ich społecznościach silną pozycję, są często "pseudo-hakerami", którzy opanowali do perfekcji parę sztuczek i uzależnili się od IRC, tak że poświęcają mu prawie cały swój czas. Poza IRC ich umiejętności okazują się często bardzo mierne. Poszukiwanie "prawdziwych hakerów" na kanałach w rodzaju #hacked, #hack, czy #hakers w dużych sieciach jest od razu skazane na porażkę. Spotkamy tam raczej liczną grupę dzieciaków i pół-amatorów, którzy chcieliby zostać hakerami, ale nie wiedzą jak tego dokonać, lub brak im zdolności.

Hakerzy z "prawdziwego zdarzenia" często unikają rozgłosu. W dużych sieciach tworzą zamknięte, sekretne kanały, przenoszą się do mniejszych sieci, tworzą własne, lub rozmawiają z sobą za pomocą chatów wewnętrznych botnetów.

Zaawansowane korzystanie z IRC-a

Usługi DCC i CTCP

Zarówno DCC (*direct client to client*) jak i CTCP (*client to client protocol*) są metodami łączenia się z sobą dwóch klientów IRC z pominięciem serwera. Zadanie serwera w tym przypadku sprowadza się do udzielenia obu klientom informacji umożliwiającej zrealizowanie bezpośredniego połączenia, po czym serwer dalej nie uczestniczy w przekazywaniu danych. DCC jest formalnie jedną z dopuszczalnych, specyficznych form CTCP, rodzajem nie do końca określonego "podprotokołu".

CTCP

Wysyłając komendę:

```
/CTCP nick dowolny tekst
```

powodujemy, że "dowolny tekst" jest bezpośrednio, bez żadnej interpretacji i zmian, wysyłany do klienta danego "nicka" z jednoczesną informacją przez kogo i kiedy został wysłany. Reakcja klienta, do którego wysłaliśmy tę komendę zależy od jego konfiguracji.

Standardowo większość klientów IRC reaguje na następujące teksty wysyłane do nich komendą CTCP:

- **PING** - odpowiedzią jest różnica czasu od wysłania przez nas komendy do jej otrzymania - umożliwia więc oszacowanie lagu między dwoma klientami z uwzględnieniem czasu przesyłania informacji do i od serwera
- **VERSION** - odpowiedzią jest nazwa, wersja i czasem nazwa systemu operacyjnego klienta
- **TIME** - czas systemowy klienta
- **USERINFO** - ogólna informacja na temat klienta
- **FINGER** - zwraca pełną nazwę użytkownika klienta i jego *idle time* (czas od ostatniego odezwania się na IRCu)
- **SOURCE** - adres strony WWW lub publicznego FTP, z którego można pobrać klienta danego użytkownika
- **CLIENTINFO** - nazwa używanego klienta
- **SED** - zestawia i uruchamia szyfrowany transfer danych
- **DCC** - rozpoczyna transmisję DCC lub informuje o możliwości, lub jej braku nawiązania łączności DCC.

Oprócz tych standardowych komend, można swojego klienta "nauczyć" reagować na dowolne teksty wysyłane przez CTCP. Umożliwia to pisanie skryptów, które realizują rozmaite funkcje pod warunkiem, że zna się "sekretne hasła" wysyłane przez CTCP, na które one reagują jak na rodzaj komend. Niektóre rodzaje botów są standardowo sterowane poprzez komendy wysyłane przez CTCP.

Większość bardziej zaawansowanych klientów IRC umożliwia też spreparowania innych niż standardowe odpowiedzi na standardowe zapytania. Czasami jest to robione dla żartu, a czasem dla celowego wprowadzenia w błąd, stąd odpowiedzi na te zapytania należy zawsze traktować z dużą ostrożnością. Np: niektórzy konfigurują swojego klienta tak aby odpowiadał na `/CTCP PING` "pingnij się w głowę", albo na `/CTCP TIME` "u mnie jest zawsze taka jak trzeba". Inni z kolei ukrywają np: swoją wersję i rodzaj klienta, po to aby zyskać przewagę zaskoczenia w "wojnach ircowych". Typowy przykład to skonfigurowanie BitchXa tak aby "udawał", że jest mIRCem. mIRC jest klientem z którym stosunkowo łatwo wygrać wojnę, natomiast dobrze skonfigurowany BitchX jest "bardzo groźną bronią".

Można też tak skonfigurować swojego klienta, aby całkowicie ignorował zapytania CTCP.

DCC

DCC jest podzbiorem usług, które są inicjowane przez komendy wysyłane protokołem CTCP. W ich przypadku CTCP służy jednak tylko do zestawienia łącza między klientami. Po zestawieniu łącza transmisja i sterowanie odbywa się już całkowicie bez pośrednictwa serwera IRC, aż do czasu zerwania połączenia. Istotą DCC jest bezpośredni transfer plików albo tekstów między klientami. Protokół zestawienia łącza DCC wymaga posiadania stałego numeru IP z tzw. zwrotnym DNS. Stąd usług DCC nie da się realizować, gdy pracuje się spod dynamicznych lub niezarejestrowanych numerów IP.

Podstawowy mechanizm DCC został stworzony w ramach prac nad klientem ircII i nie planowano go jako usługi, która by miała być stosowana powszechnie i stanowić część ogólnej specyfikacji protokołu IRC, jednak z czasem stał się on standardem, który został adaptowany do większości bardziej rozbudowanych klientów IRC.

Usługi DCC uruchamia się za pomocą komendy:

/DCC typ argument adres port (1)

gdzie - **typ** to rodzaj połączenia, **argument** - informacja o tym co ma być przesyłane (np: nazwa pliku), **adres** - adres IP użytkownika, z którym chcemy się połączyć, **port** - numer portu, z którym chcemy się połączyć. Większość klientów IRC pozwala uprościć postać tej komendy do:

/DCC typ argument nick (2)

przy czym faktycznie wysyłana jest komenda (1) - tyle, że nasz klient pobiera automatycznie adres klienta z serwera IRC, zaś za **port** podstawia standardowy, najczęściej stosowany port danego typu połączenia.

Wysyłanie tej komendy uruchamia u docelowego użytkownika pojawienie się zapytania a akceptację. Połączenie akceptuje się odpowiednią, analogiczną komendą, tyle że skierowaną na adres klienta proszącego o zgodę na połączenie. Odmowa realizacji prośby o połączenie polega po prostu na jej zignorowaniu. Po akceptacji, o ile nie ma żadnych przeszkód technicznych połączenie jest zestawiane i realizowane. Wiele klientów pracujących w środowisku graficznym upraszcza życie ich użytkownikom wyświetlając okno dialogowe z pytaniem o zrealizowanie połączenia i wysyła odpowiednią komendę zwrotną po kliknięciu na "OK" w tym oknie. Można się zatem komendą zwrotną wogóle nie kłopotać.

W podstawowym standardzie DCC dopuszczalne są następujące typy połączeń:

- **CHAT** - realizowane jest sekwencyjne przesyłanie od klienta do klienta kolejnych linii tekstu - mówiąc prościej umożliwia to po prostu rozmowę prowadzoną z pominięciem serwera IRC - aczkolwiek może też być używane do innych celów
- **SEND** - realizowane jest ciągłe, strumieniowe przesyłanie danych od klienta do klienta - co w praktyce umożliwia przesyłanie plików.

DCC CHAT w praktyce

W najprostszym wariantcie klient posługujący się nickiem "kazio" wysyła do nicka "juzek" komendę:

/DCC CHAT juzek

na którą juzek odpowiada:

/DCC CHAT kazio

po czym juzio i kazio mogą między sobą wysyłać linie tekstu zakończone "enterem" mniej więcej tak jak w zwykłej rozmowie IRC, tyle, że w tym przypadku komunikaty nie przechodzą przez serwer IRC. Umożliwia to zatem nieco bezpieczniejszą rozmowę, która nie jest też zależna od lagów i splitów serwera.

Warto zwrócić uwagę, że stosując bardziej rozbudowaną wersję tej komendy:

/DCC CHAT adres port

można się połączyć z klientem, który nie jest zalogowany do tej samej sieci IRC co my. Ściśle biorąc klient ten może wogóle nie być zalogowany do żadnego serwera IRC - wystarczy, że jest włączony i akceptuje komendy DCC.

DCC CHAT oprócz prowadzenia zwykłych rozmów, jest też często stosowany przez skrypty i boty do zdalnego sterowania nimi, co jest dokładniej opisane w rozdziale "[Skrypty i boty](#)".

DCC SEND w praktyce

DCC SEND służy generalnie do wysyłania plików, choć może być również używane do wysyłania wszelkich danych strumieniowych (np: strumieni dźwięku i obrazu). Załóżmy, że "Kazio" chce wysłać plik "moje_foto.jpg" to "Juzka". Kazio rozpoczyna od wysłania do Juzka komendy:

/DCC SEND moje_foto.jpg Juzek

Juzek, po otrzymaniu pytania o zgodę na przyjęcie tego połączenia odpowiada:

/DCC ACCEPT moje_foto.jpg

Niektóre wersje klientów, niezgodnie z pierwotnym schematem DCC, zamiast "ACCEPT" oczekują komendy "GET", gdy zatem ACCEPT nie zadziała Juzek może spróbować:

/DCC GET moje_foto.jpg

Jeśli transfer danych zostanie z jakiegoś powodu przerwany, Kazio musi go ponownie uruchomić komendą:

/DCC SEND moje_foto.jpg Juzek

zaś Juzio może wtedy odpowiedzieć komendą:

/DCC RESUME moje_foto.jpg

dzięki czemu transfer pliku zacznie się od miejsca, w którym był poprzednio przerwany.

Pełna wersja komend "SEND", "ACCEPT", "GET" i "RESUME" (z podaniem adresu i portu klienta) umożliwia, tak samo jak w przypadku "CHAT" łączenie się z klientami, które aktualnie nie są zalogowane do naszej sieci IRC, lub wogóle żadnej, a tylko są włączone.

Uwaga techniczna

plik `moje_foto.jpg` musi być przed rozpoczęciem wysyłania umieszczony w katalogu, w którym standardowo będzie go oczekiwał klient. Katalog plików "do wysyłki" trzeba zwykle zdefiniować w ramach konfiguracji klienta.

Przesyłanie plików w klientach działających w środowisku graficznym

W przypadku, gdy Juzek i Kazio dysponują dobrymi klientami pracującymi w środowisku graficznym, wysyłanie i przyjmowanie plików odbywać się będzie zapewne za pomocą okien dialogowych, które uruchamia się wybierając odpowiednią pozycję z menu programu. Wówczas oboje mogą nie mieć świadomości, że ich klienci wysyłają komendy DCC - z całą pewnością to się jednak dzieje, tyle że w sposób dla nich niewidoczny. Poznanie podstawowej składni komend umożliwia jednak bardziej świadome posługiwanie się DCC i pozwala lepiej zrozumieć co się właściwie w czasie wymiany plików dzieje.

Szyfrowane DCC

Niektóre klienty IRC umożliwiają przesyłanie tekstów i plików za pomocą szyfrowanych protokołów, takich jak np [SSL](#). Obsługa szyfrowych połączeń jest jednak różna, w różnych klientach, stąd zwykle tego rodzaju komunikacja jest możliwa tylko w przypadku gdy obie strony używają te same lub przynajmniej kompatybilne klienty. Czasami problem niekompatybilności klientów daje się rozwiązać przez wybranie jednej, umówionej wcześniej metody szyfrowania i zainstalowanie odpowiedniego skryptu lub wtyczki rozszerzającej możliwości naszego klienta.

Jednym z rozwiązań, zaimplementowanym w m.in. programie kVirc oraz jako wtyczka mIRC jest poszerzenie protokołu DCC o komendy, które nazywają się tak samo jak oryginalne, z dodaną literą "S" na początku, jest to zatem odpowiednio: **SCHAT**, **SSEND**, **SACCEPT** itd.

Innym rozwiązaniem jest stosowanie normalnych komend z dodatkowym mechanizmem uzgadniania protokołu szyfrowania już po nawiązaniu połączenia. Po nawiązaniu połączenia DCC CHAT, klienty wymieniają automatycznie komunikaty informujące się nawzajem o przejściu na szyfrowany protokół, po czym dopiero następuje właściwa transmisja danych. W praktyce, aby móc zacząć stosować szyfrowane DCC należy się zapoznać z instrukcją jego użycia dostarczoną przez producenta swojego klienta IRC.

Zagadnieniem ustandaryzowania procedur szyfrowych w ramach protokołu DCC zajmuje się projekt DCC2 (<http://www.dcc2.org/>), który pracuje nad stworzeniem nowej wersji DCC.

Pasywne i aktywne DCC

Przy normalnym połączeniu DCC strona zgłaszająca chęć uzyskania połączenia komendą "SEND" lub "CHAT" staje się jej stroną aktywną, biorącą na siebie zestawienie łącza, czyli spełniającą dla drugiej strony funkcję serwera. Z tego powodu właśnie, nie posiadając stałego adresu IP nie można z tej usługi skorzystać. Ten sam problem stwarza korzystanie z adresu [NAT](#), czy [zapory sieciowej](#) nie ujawniającej rzeczywistych numerów IP znajdujących się za nią komputerów.

Aby choć częściowo obejść ten problem, niektóre klienty IRC posiadają mechanizmy pozwalające zestawić łącze w trybie pasywnym. Działa to zwykle tak, że nasz klient wysyła komendę CTCP, która po drugiej stronie uruchamia procedurę nawiązania połączenia DCC z nami, dzięki czemu, mimo, że to my inicjujemy połączenie, stajemy się jego bierną stroną. Oczywiście ten system może działać tylko wtedy, kiedy druga strona ma stały numer IP.

Niestety, podobnie jak to jest z szyfrowanym DCC różne programy do IRC mają różne mechanizmy pasywnego DCC i mechanizm ten nie jest zestandaryzowany. Do najczęściej spotykanych należą:

System 100-121

jest on stosowany w mIRC-u i paru innych klientach i jest chyba najbardziej popularny - polega on na wysłaniu komendy **/CTCP 100 nick** dla połączenia CHAT i **/CTCP 120 nick** dla połączenia SEND - wadą tego rozwiązania jest konieczność wcześniejszego uzgodnienia adresu i numeru portu dla połączenia strony pasywnej.

System RDCC

w systemie tym inicjujący wysyła komendę **/RDCC funkcja** gdzie "funkcja" to "c" dla "CHAT" i "s" dla "SEND", a przyjmujący odpowiada komendą **RDCC 0 numer_IP numer_portu** - dzięki czemu nasz klient dostaje informację na jakim porcie i spod jakiego numeru IP ma oczekiwać na połączenie.

Serwery plików

Istnieją dwa rozszerzenia DCC, które umożliwiają przekształcenie naszego klienta w rodzaj serwera plików działającego trochę podobnie jak serwery [FTP](#). Po uruchomieniu tej usługi nasz klient IRC będzie automatycznie wysyłał listy udostępnionych przez nas plików i umożliwiał ich ściąganie - w zależności od konfiguracji wszystkim, lub tylko wybranym.

Rozszerzenie DCC o nazwie **fSERVe** zostało pierwotnie zaadaptowane w programie mIRC - dla innych programów jest ono jednak dostępne w postaci skryptu lub wtyczki. Obsługuje ono do 5 połączeń na raz i działa na zasadzie kombinacji DCC CHAT i DCC SEND. Przez DCC CHAT wysyłane są listy plików i komendy uruchamiające transfer, zaś sam transfer plików odbywa się przez tradycyjny mechanizm DCC, z fSERVem jako stroną czynną. W zasadzie każdy właściciel mIRC-a, posiadający stały numer IP może w dość prosty sposób (poprzez okno dialogowe) uruchomić tę usługę. Słabość mechanizmów zabezpieczających mIRC-a powoduje jednak, że jest to potencjalnie ryzykowne przedsięwzięcie.

Drugim tego rodzaju rozszerzeniem jest skrypt **XDCC** pierwotnie napisany w 1994 r. do programu ircII. XDCC zostało dołączone do BitchXa i paru innych klientów obsługujących język skryptowy ircII. XDCC oferuje znacznie więcej możliwości niż fSERVe - może być

skonfigurowane tak, aby komendy wyświetlające listy plików i uruchamiające transfer mogły być wysyłane w formacie "/CTCP komenda", "/msg komenda" lub przez DCC CHAT. XDCC obsługuje też system katalogowy plików, po którym można "wędrować" przy pomocy komend znanych z [DOSa](#) (DIR, CD itp.) XDCC jest bardzo popularną metodą rozpowszechniania tzw. [warezu](#) - czyli bibliotek legalnie i nielegalnie rozprowadzanych programów i plików multimedialnych.

CTCP/DCC a bezpieczeństwo

Ze względu na to, że usługi CTCP i DCC odbywają się bezpośrednio między dwoma klientami rodzi to szereg potencjalnych niebezpieczeństw, co szczególnie dotyczy DCC.

Pomijając już sprawę, że przez DCC SEND można łatwo komuś posłać wirusa lub konia trojańskiego, sam mechanizm połączenia może być dość łatwo wykorzystany do włamania się do systemu operacyjnego komputera z zainstalowanym klientem. Szczególnie narażeni są na tego rodzaju ataki użytkownicy klientów posiadających słabe zabezpieczenia, zwłaszcza mIRCa i Chatzillii.

Inny rodzaj ryzyka, dotyczący zarówno CTCP jak i DCC to możliwość kierowania ataku na klienta, który popularnie nazywany jest **CTCP bomb** lub **DCC bomb**. Atak ten polega na wysyłaniu serii komunikatów CTCP lub DCC jeden po drugim, w krótkim czasie, co powoduje oślepienie atakowanego. Przy standardowej konfiguracji klientów stosowanych najczęściej przez osoby początkujące, klient reaguje automatycznie na każdą przysyłaną komendę CTCP lub DCC. Konieczność wysłania odpowiedzi na klikaset komend w przeciągu np. 10 sekund powoduje, że klient się zawiesza lub zrywa łączy z serwerem. Wiele skryptów "bojowych" dostępnych w sieci daje możliwość łatwego wysyłania bomb DCC i CTCP. Właściciel takiego skryptu zazwyczaj musi wystukać tylko komendę np: "/BOMB DCC CHAT nick" aby wypuścić ją do określonego użytkownika.

Szczególnie skuteczny jest atak w postaci wysyłania specjalnie spreparowanej serii plików przez DCC SEND do klientów pracujących w środowisku graficznym. Jeśli klient jest ustawiony w ten sposób, że automatycznie akceptuje wszystkie nadsyłane pliki, realizacja kilkudziesięciu transferów dużych plików na raz skutecznie blokuje łączy, o ile wcześniej nie spowoduje to zawieszenie się komputera. Gdy klient jest ustawiony w ten sposób, że wyświetla okno dialogowe z pytaniem o akceptację przyjęcia plików - wyświetlenie kilkudziesięciu takich okien na raz skutecznie zawiesza klienta lub nawet cały komputer ofiary.

Aby się zabezpieczyć przed niebezpieczeństwami związanymi z DCC i CTCP można:

- ustawić swojego klienta tak aby całkowicie ignorował wszelkie komendy DCC i CTCP - jest to metoda 100% skuteczna, tyle, że utrudnia nam samym z korzystania z tych usług
- zainstalować sobie skrypt, który posiada zabezpieczenie przed bombami - różne skrypty realizują to zabezpieczenie na różny sposób; do najczęściej spotykanych należy akceptowanie DCC i CTCP tylko od osób z "białej listy" oraz automatyczne

czasowe włączanie ignorowania komend DCC i CTCP, jeśli ktoś do nas wysyła np. więcej niż 3 takie komendy jedna po drugiej w krótkim czasie.

Skrypty i boty

Tematyka skryptów i botów IRC jest z sobą silnie powiązana, dlatego oba te zagadnienia są przedstawione w jednym rozdziale.

Skryptem nazywa się w IRC specjalne fragmenty kodu programowego, które nie mogą funkcjonować samodzielnie, lecz stanowią dodatek do klienta IRC, poprawiający jego funkcjonalność, zwiększający jego możliwości lub wykonujący automatycznie pewne czynności.

Najbardziej lapidarna definicja ircowego **бота** to "klient IRC, który służy do automatycznego wykonywania zaplanowanych czynności". Można powiedzieć, że zwykły klient IRC, zaopatrzony w skrypt, który automatycznie wykonuje określone czynności, w czasie gdy jego właściciel zostawi go zalogowanego do serwera staje się rodzajem prymitywnego bota. Na ogół jednak mówiąc o ircowym bocie ma się na myśli specjalny, samodzielny program, który nie służy do rozmawiania, lecz tylko prawie wyłącznie do wykonywania pewnych, automatycznych czynności, bez bezpośredniej ingerencji użytkownika. Obraz ten gmatwa dodatkowo fakt, że wiele uniwersalnych botów ircowych konfiguruje się poprzez dodawanie do nich odpowiednich skryptów, podobnie jak zwykłe klienty IRC.

Skrypty

Możliwość pisania skryptów do klientów IRC pojawiła się wraz z programem ircII. Większość innych klientów IRC, które posiadają wbudowane własne języki skryptowe jest w mniejszym lub większym stopniu wzorowana na języku ircII. Obecnie coraz więcej klientów IRC odchodzi od stosowania wewnętrznych języków skryptowych na rzecz możliwości dołączania do nich skryptów pisanych w uniwersalnych językach, działających także poza światem IRC.

Działanie skryptów jest oparte zwykle na:

- definiowaniu nowych **aliasów** czyli komend nie występujących oryginalnie w kliencie; w najprostszym wariacie aliasy te skracają nazwy istniejących normalnie komend (np: zamiast `/join #kanał` możemy pisać `/j #kanał`); w bardziej złożonym wykonują ich całe sekwencje, mające prowadzić do wykonania określonej akcji; w najbardziej skomplikowanym wariacie alias może uruchamiać cały program z wieloma instrukcjami warunkowymi, który realizuje określone zadanie; np: alias `/takeover #kanał` może uruchamiać program, który najpierw tworzy listę nicków operatorów kanału, a następnie wykonuje serię komend `/mode #kanał -ooo nick1 nick2 nick3`, aż do pozbawienia statusu operatora wszystkich oprócz siebie

- automatycznego reagowania na określone zdarzenia tzw **events** lub **remotes** w postaci wysłania określonego tekstu na kanał lub na prywatny chat, wejście i wyjście z kanału, zmiana nicka, zmiana trybu kanału lub użytkownika itp; proste reakcje programuje się stosunkowo łatwo, bardziej złożone wymagają posiadania zdolności programistycznych i dobrej znajomości protokołu IRC
- definiowaniu wygodnych **skrótów klawiaturowych** dla najczęściej wykonywanych komend i aliasów
- w przypadku klientów działających w środowisku graficznym - dodawanie nowych **rozwijalnych menu** i zmieniania **wyglądu okien**.

Jakkolwiek skrypty można pisać samemu, to jednak w internecie można bez większego trudu znaleźć obszerne biblioteki skryptów, liczące tysiące pozycji, przynajmniej do najbardziej popularnych klientów IRC. Na ogół można więc bez trudu znaleźć gotowy skrypt, który realizuje te funkcje, które są nam akurat potrzebne. Zwykle aby znaleźć bibliotekę skryptów do danego klienta IRC wystarczy w Google' u (<http://www.google.pl>) lub innej wyszukiwarce wpisać hasło "nazwa_klienta scripts" aby uzyskać ich długą listę.

Przy pobieraniu skryptów z sieci należy uważać na czasami dołączane do nich wirusy i trojany. Najlepiej pobierać je tylko ze sprawdzonych bibliotek skryptów - polecanych na oficjalnych stronach twórców klientów IRC. Z drugiej jednak strony, wiele atrakcyjnych skryptów "bojowych" jest często niedostępnych w oficjalnych bibliotekach, gdyż mają one "zakazane" funkcje, takie jak możliwość wysyłania floodów jedną komendą, czy narzędzia do bezpośredniego ataku na system operacyjny "wroga". Tego rodzaju skrypty są często rozprowadzane przez samych użytkowników lub można je znaleźć na zakamuflowanych serwerach ftp lub XDCC.

Po ściągnięciu z sieci i rozpakowaniu skryptu, należy go zwykle umieścić w odpowiednim katalogu i uruchomić w sposób opisany w instrukcji. W przypadku ircII i wszystkich jego mutacji skrypty ładuje się komendą **/load nazwa_pliku**. Aby tego nie robić za każdym razem po włączeniu programu można tę komendę na stałe dodać do pliku konfiguracyjnego programu (w ircII jest to plik **.irc**). W programach w środowisku graficznym komenda **/load** też często działa tak samo jak w ircII. Najczęściej jednak rozbudowane skrypty do programów pracujących w środowisku graficznym są rozprowadzane w formie samoinstalującego się pliku.

Skrypty do programów pracujących w środowisku tekstowym zwykle wyświetlają na początku swojego działania ASCII-art z informacją o autorach i rodzajem skróconej instrukcji obsługi. Rozbudowane skrypty do programów pracujących w środowisku graficznym zwykle modyfikują menu programu, zmieniają nieco wygląd okien i "witają" użytkownika oknem z jakąś grafiką.

Skrypty do IRC można ogólnie podzielić na:

- **poszerzające funkcjonalność** - dodające listę aliasów ułatwiającą wysyłanie komend, narzędzia do pracy z funkcjami dostępnymi w określonej sieci itp.

- **zabawowe** - są to skrypty wysyłające automatyczne, zabawne komunikaty na kanał, skrypty z bibliotekami ANSI i ASCII-artów, skrypty do przeprowadzania quizów, gier, ułatwiających flirtowanie itp.
- **serwery DCC i baz danych** - są to skrypty które ułatwiają postawienie DCC serwera lub udostępniające określone bazy danych - np. gromadzące i udostępniające listy użytkowników kanału wraz z datami ich ostatniej bytności na kanale - tego rodzaju skrypty najczęściej są dołączane do botów - można jednak znaleźć też tego rodzaju skrypty do zwykłych klientów IRC
- **obronne i bojowe** - są to skrypty służące do obrony przed atakami innych użytkowników i/lub posiadające funkcje służące do atakowania innych
- **pakiety uniwersalne** - są to najpopularniejsze skrypty, które zawierają w sobie jak najwięcej udogodnień - od poszerzania funkcjonalności, przez elementy zabawowe po złożone funkcje obronne i bojowe.

Słynne skrypty

Poniżej opisane są w skrócie, wybrane "legendarne" skrypty do ircII i ich klonów oraz do mIRCa:

LiCe był prawdopodobnie pierwszym w historii uniwersalnym skryptem do klienta IRC. Nie wiadomo dokładnie kiedy powstał, ale pierwsze jego wersje istniały już na pewno w 1992 r. Pierwotnie był stworzony do "czystego" ircII, potem rozwijano go głównie do EPICa. W dużym stopniu swoją popularność zawdzięcza temu, że był od początku dostępny na licencji GPL, podczas gdy inne skrypty nie posiadały określonej licencji. Za jego głównego twórcę uchodzi anonimowy programista używający nicka srfrog. LiCe posiada w zasadzie wszystko, czego można by oczekiwać od skryptu: ochronę przed atakami i floodami, gromadzenie "białej" i "czarnej" listy nicków, narzędzia do ochrony i nadzorowania kanałów, wiele użytecznych aliasów i skrótów klawiaturowych oraz inne udogodnienia. Mimo, że posiada pewne narzędzia "ofensywne" jest to raczej skrypt dla ludzi nastawionych pokojowo. Wiele późniejszych skryptów tego rodzaju było w dużym stopniu wzorowanych na LiCe. LiCe jest wciąż popularny, zwłaszcza wśród osób przywiązanych do EPICa.

BitchX, który obecnie jest osobnym klientem IRC pierwotnie był najbardziej popularnym "bojowym" skryptem do ircII i EPICa. Był on nastawiony nie tyle na ułatwienie użytkownikowi życia, lecz na skuteczne prowadzenie wojen i kontrolowanie kanałów. To prawdopodobnie ten skrypt wprowadził takie pojęcia jak "massdeop", "masskick" i podobne. Nie wiadomo dokładnie, kiedy powstała pierwsza wersja skryptu, ale przed 1996 r. był on już bardzo popularny. Za jego autorów uważa się zwykle "Trench" i "HappyCrappy". Od 1996 r. "panasync" (Colten Edwards) zaczął przekształcać ten skrypt w osobnego klienta IRC, poprzez dodanie większości jego funkcji do kodu EPICa.

7thSphere to legendarny, piracki skrypt "bojowy" do mIRCa, który oprócz typowych aliasów i komend dla skryptu służącego do "walki" posiadał też wbudowane małe, zewnętrzne programiki, które można było używać niezależnie od skryptu i mIRCa. Programikami tymi można było próbować zawiesić system operacyjny przeciwnika, takie jak *portfuck* (wysyłanie na przypadkowe porty zgłoszeń uruchomienia rozmaitych usług), *pestilence* (zawieszający atak na port 195 Windowsów 95), *assault* (floodujący ICMP ECHO

oraz UDP) i kilka innych tego rodzaju "zabawek". Skrypt ten powstał ok 1996 r. i był dostępny ze strony <http://www.7thSphere.com>. Szacuje się, że pobrało go i używało nie mniej niż milion ludzi. Skrypt ten został tak zniechęcony, że za jego używanie na niektórych serwerach dostawało się od razu k-linię. Jego ostatnia wersja (3.0), wypuszczona w 2000 r. została wykorzystana do utworzenia jednej z największych w historii sieci komputerów zarażonych [trojanem](#) (rodzajem programu działającego podobnie do wirusa, który przejmuje kontrolę nad zarażonym komputerem). Nie jest do końca jasne, czy trojan był dołączony przez autorów skryptu, czy też ktoś rozpowszechnił jego dystrybucję z załączonym wirusem, w każdym razie spowodowało to, że skrypt stracił reputację, a 7thSphere.com zniknęła z rynku. Do dziś istnieją strony, które proponują do ściągnięcia ten skrypt, jest to jednak zawsze obciążone ryzykiem zainstalowania sobie wirusa.

ircN to prawdopodobnie najbardziej przemyślany skrypt do mIRCa, o charakterystyce zbliżonej do LiCe, tzn. nastawiony głównie na funkcjonalność i bezpieczeństwo. W porównaniu do innych popularnych skryptów do mIRCa posiada stosunkowo mało "wodotrysków", ale ma za to wiele pożytecznych narzędzi przydatnych przy pilnowaniu kanału, pracy z serwisami itp. Dostępny jest na licencji GPL, rozwijany jest przez klikunastoosobowy zespół programistów. Jest szczególnie popularny wśród użytkowników EFnetu, używa go nawet kilku ircopów.

eXtreme to solidnie napisany skrypt do mIRCa, przeznaczony dla osób nie mających ambicji stałego zarządzania dużymi kanałami. Posiada dość dobrą podstawową funkcjonalność ale najbardziej jest znany z dużej liczby "wodotrysków" takich jak kanałowy mp3 player, rozbudowana książka adresowa, efektowne przywitania itp. Był to prawdopodobnie pierwszy w historii skrypt z obsługą mp3 playera.

noName to kolejny solidnie napisany skrypt do mIRCa, posiadający wyjątkowo dużo "wodotrysków", jednak dość przydatnych i przejrzyste przygotowanych. Są to m.in: obsługa kanałów RSS, wykonywanie wykresów działania sieci (statystyka lagów i splitów), dobry system ostrzegania o splitach, obsługę serwisów "Q" i "L" QuakeNetu. Jest szczególnie popularny właśnie wśród użytkowników QuakeNetu i jego "mroczny klimat" jest dobrze dopasowany do gustów graczy Quake'a.

Krótką lista bibliotek skryptów

- <http://scripts.epicsol.org/list.php> - Skrypty do ircII i Epica
- <http://www.irchelp.org/irchelp/script/> - Wybrane skrypty do ircII
- <http://www.linuxirc.com/> - obszerna biblioteka skryptów do XChata, Irsii, Eggdropa i innych klientów IRC pod Linuxa
- <http://scripts.bitchx.org/> - biblioteka skryptów do BitchXa
- <http://scripts.xchat.org/cgi-bin/disp> - Obszerna biblioteka skryptów do XChata (głównie skrypty zabawowe i poszerzające funkcjonalność)
- <http://www.scriptheaven.net/> - biblioteka skryptów do różnych klientów, głównie do mIRCa
- <http://www.mircscripts.org/> - obszerna bibliotek skryptów do mIRCa
- <http://www.visualscripts.com/scripts> - biblioteka skryptów do Visual IRC

Boty

Boty są niemal tak samo stare jak sama sieć IRC. Prawdopodobnie pierwszym botem ircowym był GM napisany przez Grega Lindahla, który służył do prowadzenia przez IRC gry typu M.U.L.E *Hunt the Wumpus*. Współcześnie boty służą przede wszystkim do nadzorowania kanałów w sieciach, które nie oferują serwisu ich rejestrowania. W sieciach z serwisami można jednak spotkać boty oferujące rozmaite dodatkowe usługi, takie jak tworzenie baz danych użytkowników, zbieranie statystyk kanałowych, prowadzenie gier i quizów, dostarczanie rozmaitych informacji, prowadzenie sztucznych konwersacji i wiele innych usług.

Zanim podejmie się decyzję o postawieniu własnego bota warto się zastanowić nad sensownością takiego działania. Bot z zasady jest zalogowany do sieci 24 godziny na dobę. Zakładając, że przeciętny użytkownik IRCa spędza na nim 2-3 godziny na dobę, jeden bot obciąża serwer i łączy w sposób odpowiadający 8-10 zwykłym użytkownikom. Stąd wielu administratorów serwerów IRC zwalcza aktywnie stawianie zbędnych botów. W niektórych sieciach istnieje obowiązek rejestrowania botów, co wymaga dobrego uzasadnienia ich postawienia, w innych z kolei użytkownicy mogą donosić ircopom o zbędnych botach i są one następnie "likwidowane" poprzez założenie dla nich [k-](#) lub [g-linii](#). Wreszcie istnieją sieci, w których stawianie botów jest całkowicie zabronione.

Do postawienia bota trzeba mieć odpowiednie warunki. Zaliczają się do nich:

- posiadanie stabilnego i dobrze skonfigurowanego systemu operacyjnego (MS Windows raczej odpada)
- posiadanie stabilnego łącza ze stałym numerem IP i zwrotnym DNS
- odpowiednie kwalifikacje programistyczne i dużo wolnego czasu na poprawne skonfigurowanie i nadzorowanie bota.

Boty od strony funkcji, jakie spełniają, można podzielić na:

- **infoboty** - gromadzące i później udostępniające rozmaite informacje, np. boty zbierające i udostępniające dane statystyczne na temat kanału, boty tworzące bazy danych użytkowników itp.
- **boty usługowo-specjalistyczne** - które zazwyczaj są rodzajem łącznika między rozbudowanymi programami a określonym kanałem IRC; typowy przykład to bot Eliza poprzez którego można prowadzić konwersację ze słynnym programem [Eliza](#) udającym zwykłą konwersację
- **boty "rozrywkowe"** - boty do przeprowadzania quizów, gier, boty z bibliotekami żartów itp.
- **boty "serwery plików"** - spełniające rolę ircowych serwerów ftp - udostępniające publicznie lub tylko dla wybranych zasoby plików multimedialnych lub warezu.
- **boty "nadzorcy"** - najczęściej spotykane - służą do ochrony i pilnowania porządku na kanałach

Infoboty

Prawozorem wszystkich infobotów był bot o nazwie **url**, który gromadził adresy stron WWW a następnie podawał je na kanał po zadaniu na kanale odpowiedniego pytania. Np: chcąc aby **url** zapamiętał adres Wikipedii należało napisać:

wikipedia is <http://pl.wikipedia.org>

na co bot odpowiadał:

The Infobot at this point silently remembers the statement

po czym przy każdym pojawieniu się w czyjejs wypowiedzi słowa "wikipedia" ze znakiem zapytania otrzymuje się odpowiedź:

somebody said wikipedia is <http://pl.wikipedia.org/>

Bardzo często spotykanymi infobotami są boty gromadzące informacje o użytkownikach kanału, zwane **seenami** (od *seen* - *widziałem*). Rejestrują one wejścia wszystkich użytkowników oraz to kiedy ostatni raz coś powiedzieli na kanale. Później można im zadawać pytania pisząc na kanale:

!seen Nick

na co się otrzymuje odpowiedź w rodzaju:

Nick was here 7 hrs 31 min ago

albo

I don't know nick

albo

Nick is here now, don't U see it?

Jednym z najbardziej rozwiniętych infobotów jest **blootbot** rozwijany przez zespół pod kierownictwem Tima Rikera. Bot ten w zależności od konfiguracji może być nauczony gromadzenia wielu różnych danych oraz automatycznego przetwarzania prostych pytań na zapytania [SQL](#) kierowane do określonej bazy danych MySQL, na podstawie czego formułuje on określoną odpowiedź w języku naturalnym.

Era Eggdropa

Początkowo (mniej więcej do 1994 r.) większość botów nadzorujących kanały IRC była unikalnymi programami pisanymi w różnych językach programowania. Każdy z botów posiadał własne cechy, które odróżniały go od innych. Napisanie ciekawego bota stanowiło rodzaj nobilitacji w świecie programistów interesujących się IRC. Era "indywidualnych" botów-nadzorców dość szybko jednak się skończyła, gdyż okazało się, że prawie wszystkie

one wykonują bardzo podobne zadania. Wydawało się więc sensowne napisanie uniwersalnego programu, który można by dowolnie konfigurować w taki sposób aby wykonywał on wszystkie niezbędne akcje oczekiwane od tego rodzaju bota.

Najbardziej popularnym tego rodzaju programem stał się **Eggdrop** napisany przez Robeya Pointera i Jamiego Rishawa w 1993 r. Jest to program napisany w języku [C](#), który umożliwia:

- konfigurowanie go na różne sposoby na różnych kanałach
- ochronę kanału - poprzez stworzenie listy "przyjaciół i wrogów", odzyskiwanie statusu opa po splicie i rozłączeniu, możliwość zablokowania kanału i tworzenie list banów, które się uaktywniają dopiero po tym jak ktoś z listy zbanowanych pojawi się na kanale
- tworzenie botnetów z kilku botów i możliwość tworzenia w nich wewnętrznych botlinii
- dodawanie rozmaitych dodatkowych funkcji (np: podobnych w działaniu do infobotów) poprzez możliwość ładowania skryptów pisanych w języku tcl - skrypty pierwotnie pisane do Eggdropa można też używać w klientach IRC obsługujących język tcl.

Eggdrop bardzo szybko stał się niezwykle popularny i utworzone z Eggdropów botlinie można było spotkać pod koniec lat 90. XX w. na większości dużych kanałów w IRCnecie, EFnecie i innych sieciach nie posiadających serwisu rejestracji kanałów. Ze względu na to, że poprawne skonfigurowanie botnetu złożonego z Eggdropów nie jest takie proste, zaczęły do niego powstawać gotowe zestawy plików konfiguracyjnych i pakiety skryptów tcl, o najprzeróżniejszych nazwach i zastosowaniach. Użycie tego samego pakietu konfiguracyjnego we wszystkich Eggdropach nadzorujących dany kanał gwarantowało, że nie robił się na nim bałagan, polegający na przypadkowo zaczętych na skutek niezgodności konfiguracji bitwach między botami. Często właściciele Eggdropów mających kontrolować dany kanał uzgadniali konstrukcję takiego pakietu i utrzymywali ją w tajemnicy, aby utrudnić konkurencji przejęcie kanału.

Jednym z najbardziej znanych tego rodzaju zestawów był **Bear** autorstwa Macieja Freudenheima, przy pomocy którego nadzorowano kanał #AGH i #plug na IRCnecie. Eggdrop z **Bearem** był jednak zbyt powolny i czasami zdarzało mu się przegrywać bitwy nawet z dobrze skonfigurowanym BitchXem. Stąd zdecydowano dodać do kodu Eggdropa większość dodatkowych funkcji z pakietu Beara i w ten sposób ok. 1999 r. powstał nowy program o nazwie **Void**. Void nie był początkowo dostępny publicznie, lecz był rozpowszechniany wśród zaufanych znajomych Freudenheima, którzy nadzorowali największe polskie kanały w IRCnecie. W końcu jednak został upubliczniony i stał się popularny nie tylko wśród polskich użytkowników IRC, ale na całym świecie. W 2002 r. znaleziono w nim błąd krytyczny pozwalający na przejęcie całego botnetu opartego na Voidach, co spowodowało, że program ten stracił nimb "niezwycięzonego". Później powstało wiele podobnych pakietów konfiguracyjnych do Eggdropa, **aVersE**, **diVerse**, **Abuse 1**, **Abuse 2**, **DetoX** i kilkanaście innych, z których każdy ma rozmaite wady i zalety.

Inne boty uniwersalne, mniej lub bardziej podobne do Eggdropa to:

- **Winbot** - mutacja Eggdropa działająca pod systemem MS Windows - zawodna tak samo jak cały MS Windows
- **ghost** - tajemnicza mutacja Eggdropa - nastawiona na kontrolę kanału - podobna nieco do Voida
- **EnergyMech** - napisany w C o podobnej funkcjonalności co Eggdrop - nie ma jednak możliwości zestawiania hierarchicznego botnetu, a tylko wewnątrz DCC chat, bez możliwości centralnego sterowania całością - nie ma też możliwości dynamicznego włączania i wyłączania skryptów
- **Suppybot** - napisany w języku python - posiada szereg typowych funkcji nadzorczych, ale jest bardziej nastawiony na przyjazność i łatwość w obsłudze oraz możliwość dodawania wtyczek niż na kwestie bezpieczeństwa - działa w każdym systemie operacyjnym, w którym da się zainstalować kompilator pythona.

Lista popularnych botów IRC

[blootbot](#)

infobot o szerokich możliwościach współpracy z MySQL

[Botnix](#)

napisany w perlu bot nadzorca przystosowany do pracy z IPv6, posiada m.in. moduł pozwalający przekształcić go w kлона Elizy - nie obsługuje botnetów

[ComBOT](#)

rodzaj infobota umożliwiającego przeszukiwanie stron WWW przez komendy wydawne na kanale IRC lub na query

[Eggdrop](#)

omówiony już wyżej

[Darkbot](#)

gadatliwy bot, potrafiący prowadzić rodzaj konwersacji, odpowiadać na pytania zebrane w bibliotece FAQów, posiada także elementarne funkcje potrzebne do nadzorowania kanału - napisany w C - działa pod Windowsami i pod Linuxem

[EnergyMech](#)

opisany wyżej

[InfoBot](#)

oryginalny infobot opisany wyżej

[Mozbot](#)

napisany w Perlu - bot usługowy stworzony na potrzeby społeczności projektów Mozilli - przekazuje strumień informacji z bugzilli, potrafi na różne sposoby przeszukiwać strony WWW i przekazywać na kanał rezultaty oraz posiada podstawowe funkcje potrzebne do nadzorowania kanału

[YB-Bot](#)

podobny do Darkbota - posiada funkcję seen, odpowiadanie na pytania z listy FAQ oraz podstawowe narzędzia do nadzoru nad kanałem

[WinBot](#)

opisany już wyżej klon Eggdropa pod MS Windows

[Supybot](#)

opisany już wyżej bot o funkcjonalności zbliżonej do Eggdropa

[Ghost](#)

już opisany wyżej bojowy klon Eggdropa

Źródło: Artykuł [IRC bot](#) w anglojęzycznej Wikipedii

Miejsca skąd można pobrać popularne boty:

- ajuzo.com
- download.forum.irc.pl

Botnety i nadzór nad kanałami

Do postawienia Eggdropa trzeba mieć dostęp do tzw. konta shelowego na komputerze zaopatrzone w jakiś system operacyjny podobny do Uniksa (np: Linuksa albo FreeBSD), który jest włączony 24 godziny na dobę do stałego łącza internetowego, oraz pewien zasób podstawowych umiejętności pracy na koncie shelowym. Jeden Eggdrop, czy nawet jeden Void zazwyczaj nie wystarcza jednak do skutecznego kontrolowania kanału, ze względu na splity oraz możliwość zrywania się łącza między botem i serwerem IRC. Z tego względu na kanale musi zwykle przebywać grupa botów, najlepiej przyłączona przez różne serwery danej sieci IRC, które powinny z sobą współpracować, nawzajem się wspierając.

Problemy z nieskoordynowanymi grupami botów

Tego typu grupy botów powstają zazwyczaj spontanicznie - odpowiednio zaawansowani technicznie użytkownicy kanału stopniowo wstawiają na kanał kolejne boty, czasami z powodów praktycznych, a czasami z powodów czysto ambicjonalnych, gdyż posiadanie własnego bota jest rodzajem ircowej nobilitacji. Każdy nowy bot musi być odpowiednio skonfigurowany, aby nie "gryzł" się z innymi. Np: musi mieć dopisane do "listy przyjaciół" wszystkie wcześniejsze boty, nie może forsować innego trybu kanału, niż uzgodniony wcześniej itp. Jednak, nawet przy dobrym zegraniu botów, gdy są one pozostawione same sobie dochodzi wcześniej czy później do samozapętających się konfliktów. Typowy scenariusz konfliktu wygląda tak:

- jeden z botów dopisuje z jakiegoś powodu kogoś do "czarnej listy" (np: dlatego, że rozrabiał na innym kanale)
- drugi bot o tym "nie wie" i gdy pierwszy założy bana po wejściu osoby z "czarnej listy" drugi zdejmie tego bana, bo nie ma danej osoby na swojej "czarnej liście"
- pierwszy ponownie założy tego bana, drugi go zdejmie i tak w kółko, co uniemożliwi normalną rozmowę na kanale, gdyż kanał będzie wyglądał następująco:

```
bot1 has banned !*.hostxxx.neostrada.adsl.pl
bot2 has changed channel mode to -b !*.hostxxx.neostrada.adsl.pl
bot1 has banned !*.hostxxx.neostrada.adsl.pl
bot2 has changed channel mode to -b !*.hostxxx.neostrada.adsl.pl
bot1 has banned !*.hostxxx.neostrada.adsl.pl
bot2 has changed channel mode to -b !*.hostxxx.neostrada.adsl.pl
bot1 has banned !*.hostxxx.neostrada.adsl.pl
bot2 has changed channel mode to -b !*.hostxxx.neostrada.adsl.pl
bot1 has banned !*.hostxxx.neostrada.adsl.pl
bot2 has changed channel mode to -b !*.hostxxx.neostrada.adsl.pl
itd...
```

- jeśli na kanale nie będzie właścicieli botów, którzy ręcznie uzgodnią "czarną listę" na obu botach, jedynym rozwiązaniem będzie wykopać i zbanować jednego z tych dwóch botów; to jednak często pogorszy jeszcze sytuację, gdyż kolejne boty mają dopisanego tego bota do "białej listy", będą więc go chroniły
- w rezultacie boty zaczną nawzajem wykopywać się kanału, banować, zabierać sobie opa, aż wreszcie na "placu boju" zostanie się jeden.

Stawianie botnetów

Aby zapobiec tego rodzaju kłopotom boty muszą między sobą automatycznie wymieniać podstawowe informacje, takie jak białe i czarne listy, forsowanie określonego topicu, listy operatorów botów itd. W przypadku Eggdropów rozwiązano ten problem pisząc specjalny skrypt tcl o nazwie "botnet", który umożliwił zestawianie grup tych botów w sieci, w których wymieniają one między sobą podstawowe pliki konfiguracyjne. Sieci te konstruowane są albo poprzez stale utrzymywane połączenie DCC CHAT albo w bardziej zaawansowanych przypadkach połączenia [telnet](#) lub nawet [ssh](#). Połączenia te zwane są **botliniami** i tworzą one rodzaj wewnętrznego chatu całego botnetu. Chat ten jest głównie używany do automatycznej wymiany podstawowych danych konfiguracyjnych, ale może też być stosowany do prowadzenia rozmów między osobami, które zostały do botlinii dopuszczone.

Wewnątrz botnetu istnieją flagi hierarchizujące boty. Np: botnet można tak zestawić, aby jeden, główny bot gromadził czarną i białą listę od pozostałych botów, a następnie zwrotnie uzgadniał ją z nimi. Można jedna ustalić, że np: białe listy będą różne na każdym z botów, zaś czarna będzie narzucana przez głównego bota. Wewnątrz botnetu odpowiednie flagi nadaje się też osobom dopuszczonym do poszczególnych botów. Np: część osób może sobie tylko "pogadać" w botlinii, inni mogą w pewnym zakresie sterować jednym botem, ale nie mają prawa ingerowania w działanie pozostałych, wreszcie są też tacy, którzy otrzymują flagi "globalne" pozwalające sterować całą botlinią.

Botnety mogą posiadać rozmaite topologie:

- **Rozproszoną** - w której nie występuje wyróżniony główny bot, lecz boty wymieniają stale między sobą dane i pewna grupa użytkowników (zwykle właścicieli botów) otrzymuje jednakowe uprawnienia sterowania całością - jeśli boty są dobrze skonfigurowane (np: w oparciu o jeden pakiet konfiguracyjny) i żaden z ich właścicieli nie zrobi czegoś "głupiego", to taki botnet jest całkiem efektywny w nadzorowaniu kanału - niestety w praktyce często się zdarza, że któryś z właścicieli botów, zrobi coś, co zdesynchronizuje cały botnet.
- **Scentralizowaną** - w której występuje główny bot, zwany **hubem**, który narzuca wszelkie "reguły gry" pozostałym botom. Całym botnetem zarządza się z huba. Taki botnet jest dość łatwo skonfigurować, ale można go całkowicie zniszczyć przez atak na huba. Z tego powodu często hub nie jest wogóle logowany do serwera, dzięki czemu jest "ukryty", nie można go zaatakować z poziomu IRC, a jego wytropienie i zniszczenie wymaga wysłedzenia, na jakim koncie jest umieszczony i dokonanie włamania na to konto.
- **Mieszana** - w której występują boty "węzłowe" i boty częściowo ubezpiełnowolnione, przy czym węzłowe mogą z kolei być "ubezpiełnowolnione"

przez boty węzłowe "wyższego rzędu", w wyniku czego tworzy się złożony, częściowo zhierarchizowany graf, nie posiadający jednak łatwo wyróżnialnego centralnego punktu. Tego typu botnety jest niezwykle ciężko skonstruować w sposób zapobiegający konfliktom - gdy jednak to się uda - botnet taki jest bardzo trudny do pokonania, gdyż osobom z zewnątrz bardzo trudno jest się zorientować w jego strukturze i znaleźć w nim słabe punkty.

Ogólnie "przepis" na zorganizowanie skutecznego botnetu na dużym kanale w IRCnecie, na którym bywa ponad 100 osób na raz może wyglądać tak:

- Stawiamy scentralizowany botnet z ukrytym hubem - oparty na jednym z klonów Eggdropa np: diVersE, składający się z 8 botów - po jednym na każdym polskim serwerze IRCnetu + 3 na różnych zagranicznych + hub.
- Na kanale akceptujemy inne boty - które włączamy do botnetu pod warunkiem zgody na kontrolowanie ich whitelisty i shitlisty, pozostawiając resztę spraw ich właścicielom.
- Na kanale akceptujemy też wspierające botnet sesje BitchXa - pod warunkiem, że są one ustawione w niegresywny tryb.

Spółeczny aspekt botnetów

Jak widać, kontrolowanie kanałów przy pomocy botnetów to bardzo złożona sprawa, angażująca wiele osób i wiele wysiłku. Wydawałoby się, że w tej sytuacji sieci z centralnym [serwisem](#) rejestrowania kanałów w rodzaju **chanserva** - powinny już dawno całkowicie zdominować IRC. Zdaniem zwolenników sieci "dzikich", bez możliwości rejestrowania kanałów, mają one tę przewagę, że powstaje w nich swoista subkultura ircowa, której bardzo ważnym aspektem jest "walka o utrzymanie kanału" i budowanie botnetów.

Botnety oprócz funkcji pilnowania kanałów mają szereg aspektów społecznych - można je rozbudowywać nie tylko w oparciu o boty z danego kanału, lecz także tworzyć botnety międzykanałowe, a nawet między różnymi sieciami IRC - celem takich rozbudowanych botnetów nie jest już zarządzanie kanałami lecz tworzenie rodzaju "podsieci", dostępnej dla grona "wybranych" - czyli tych, którzy potrafią postawić, poprawnie skonfigurować i utrzymać bota. Botlinie w tego rodzaju botnetach - liczących czasami setki botów, stają się rodzajem "wewnętrznego kręgu IRCa", do którego dostęp "nobiletuje", daje poczucie wyższości i władzy. Dla osób z dostępem do botlinii dużych botnetów rozmawianie na zwykłych kanałach IRC staje się "lamerstwem" i "stratą czasu", zaś sieci, w których chanserv załatwia sprawę pilnowania kanału, bez korowodów ze stawianiem botów - to "zabijanie ducha prawdziwego IRCa".

Botnety oparte na trojanach

Botnety tego rodzaju służą do zmasowanego ataku na określone kanały, serwery lub nawet całe sieci IRC. Botnety tego rodzaju powstają w następujący sposób:

- Do pliku dystrybucyjnego klienta IRC, skryptu lub bota dodaje się trojana i tak spreparowany plik dystrybuje się w sieci. Czasami trojany tworzące botnety są też dodawane do programów nie mających żadnego związku z IRC.
- Trojan ten po zainstalowaniu na komputerze ofiary uruchamia prymitywnego bota IRC, który loguje się do wybranej przez twórcę trojana sieci i wysyła do niego ustalony komunikat ctcp, że jest już gotowy do pracy, po czym zostaje włączony do botnetu.
- Właściciel komputera, z którego został uruchomiony bot-trojan nie ma zazwyczaj zielonego o tym pojęcia.
- Osoba kontrolująca botnet ma zatem dostęp do dużej liczby botów, którymi na określoną komendę może wejść na dowolny kanał i próbować go przejąć.
- Często tego rodzaju botnety są też wykorzystywane do "oślepienia" serwerów, w celu spowodowania splitu i przejęcia danego kanału "zwykłymi" botami.
- Najgorszą rzeczą, jaką robią tego rodzaju botnety, jest atakowanie nie tyle użytkowników czy serwerów IRC, lecz innych poza-IRCowych celów, np: określonych serwerów znienawidzonych przez siebie instytucji czy firm, najczęściej przez zmasowany atak typu [DDoS](#) (*Distributed Denial of Service attack*); w tym przypadku IRC służy wyłącznie do wygodnego koordynowania całego ataku.

Z poziomu pojedynczych użytkowników IRC obrona przed tego rodzaju botnetami może polegać wyłącznie na chronieniu swojego komputera przed zainstalowaniem na nim trojana, oraz informowanie IRCopów w momencie, gdy zauważy się tego rodzaju botnet. Botnety trojanowe siedzące na kanałach jest stosunkowo łatwo rozpoznać, gdyż nicki poszczególnych botów są tworzone automatycznie wg jakiegoś algorytmu - ponadto wszystkie one milczą jak zakłute lub prowadzą nienaturalne, automatyczne rozmowy. Przykłady zrzutów ekranowych zainfekowanych kanałów można zobaczyć na stronie programu SwatIt (<http://swatit.org/bots/gallery.html>). Sami IRCopowie stale nadzorują swoje serwery w poszukiwaniu tego rodzaju botnetów i starają się zlokalizować źródło ich dowodzenia, a następnie zgłaszają do firmy dostarczającej internet właścicielowi botnetu poważne naruszenie netykiety.

Protokół i jego odmiany

Z technicznego punktu widzenia IRC jest jednym z protokołów internetowych. Jest on nawet wpisany do typów [MIME](#) i specyfikacji [URI](#), dzięki czemu można np. tworzyć na stronach WWW linki w postaci: <irc://adres.serwera/#kanał>, po których kliknięciu uruchamia się klient IRC (o ile jest poprawnie zainstalowany) i automatycznie wchodzi na dany kanał.

IRC jest o tyle dziwnym protokołem, że jego oficjalnych specyfikacji zawartych w odpowiednich dokumentach RFC ([RFC 1459](#), [RFC 2810](#), [RFC 2811](#), [RFC 2812](#), [RFC 2813](#)) nikt dokładnie nie przestrzega, tzn. aktualnie ani większość oprogramowania tworzącego serwery, ani większość klientów IRC nie stosuje się dokładnie do odpowiednich RFC, co powoduje, że w praktyce w sieciach IRC występuje wiele jego mutacji.

Protokół IRC jest otwartym protokołem tzw. warstwy aplikacyjnej [TCP](#) z możliwością stosowania też połączeń [SSL](#). Jest to protokół typu "plaintext", tzn. podobnie jak to jest w przypadku protokołu "mail", teoretycznie można go stosować przy pomocy każdego rodzaju programu, który posiada zdolność przesyłania strumienia bajtów do serwera. Np: można by, znając na pamięć odpowiedni ciąg komend i znaków, połączyć się z serwerem IRC przy pomocy dowolnego programu do [telnetu](#), choć oczywiście w praktyce taki sposób korzystania z IRC byłby szalenie uciążliwy. Protokół IRC opiera się na nieco tylko zmodyfikowanym zestawie znaków [ASCII](#), co powoduje, że stosowanie znaków diakrytycznych (np. tzw. pliterek) wymaga stosowania po stronie klienta różnego rodzaju tricków, np kodowania jednego znaku diakrytycznego dwoma znakami ASCII.

O ile protokół komunikacji między serwerem i klientem jest w zasadzie zbliżony we wszystkich odmianach IRCd poczynając od jego wersji 2.8, o tyle protokół komunikacji między serwerami różni się dość zasadniczo w różnych mutacjach IRCd. Najczęściej stosowane są TS5 (*Timestamp 5*), P10 i ND/CD (*Nick Delay/Channel Delay*), choć wiele sieci celowo implementuje protokoły zupełnie nietypowe (np: QuakeNet) i nawet nie publikuje ich dokładnej specyfikacji, aby utrudnić potencjalne ataki na serwery. Powoduje to, że różne mutacje serwerów IRC są zwykle z sobą niekompatybilne, choć niektóre z nich (np: IRC-hybrid) można przy pomocy specjalnych modułów dostosować do określonego protokołu. Istnieją też tzw. serwery mostkujące, umożliwiające w ograniczonym stopniu zestawianie sieci z różnych mutacji IRCd, pozostawiają one jednak wiele do życzenia i nie są w praktyce stosowane.

Ze względu na to, że oficjalne specyfikacje protokołu i tak nie są przestrzegane, aby omówić różnice między różnymi jego wariantami spotykanymi w istniejących sieciach, najlepiej jest opisać możliwości tzw. IRC demonów (w skrócie **IRCd**) - programów, które pełnią rolę serwerów IRC.

IRCd - historia i współczesność

Pierwotny demon serwerowy IRC został napisany przez Jarkko Oikarinena w 1989 r. Znany jest pod nazwą **ircd**. Pierwotna wersja ircd miała liczne wady, często się zawieszała i nie miała opracowanego skutecznego systemu wymiany danych między serwerami, na skutek czego niemożliwe było skonstruowanie sieci liczącej więcej niż 3 serwery. Właściwy rozwój IRC zaczął się dopiero od **IRCd 2.2**, który został napisany przez Markku Savela z pomocą Oikarinena. IRC z tamtych czasów był pod wieloma względami zupełnie różny od tego, który znamy obecnie. Np: kanały nie miały nazw tylko numery a wyboru kanału dokonywano na podstawie jego topicu, który wyświetlał się po użyciu komendy list. Długość nicków była ograniczona do 3 znaków. Później, używanie 3-literowych nicków, gdy dłuższe już były dozwolone, było symbolem tego, że należy się do starej elity pionierów IRC.

Wersje od 2.3 do 2.5 były rozwijane przez ciągle spierających się z sobą właścicieli spontanicznie powstających serwerów IRC, co powodowało chaos w kodzie i ciągle zmiany kierunku rozwoju tego oprogramowania. W tym czasie istniała w zasadzie tylko jednak sieć IRC, która miała charakter zupełnie otwarty, to znaczy każdy kto dał radę postawić serwer mógł się przyłączyć nim do pozostałych. Spowodowało to jednak, że topologia sieci była całkowicie chaotyczna i wiele osób stawiało serwery tylko po to, aby uzyskać status IRCopa i

móc przejąć tym sposobem jakiś kanał. W 1990 r. powstała organizacja o nazwie **EFnet**, która ustanowiła pierwsze reguły określające zasady przyłączania kolejnych serwerów do sieci i narzucająca rodzaj kodeksu IRCopa, zakazującego mu pewnych działań. Grupa serwerów, które znalazły się poza EFnetem utworzyła **A-net** (Anarchy net), która jednak z powodu kompletnego chaosu i sporów, wkrótce zanikła.

Wersja **IRCd 2.5** została gruntownie uporządkowana już w ramach EFnetu. Dodano w niej możliwość tworzenia kanałów posiadających nazwy tekstowe (pierwotnie jako znaku kanału używano "+" a nie "#") i wydłużono nazwy nicków do 8 znaków. Wzrastająca popularność EFnetu powodowała jednak, że sieć ta bardzo cierpiała na częste splitsy i trudności z zalogowaniem się do serwerów, gdyż tempo powstawania kolejnych serwerów było wolniejsze od wzrostu użytkowników.

W trakcie pracy nad **IRCd 2.7** w 1992 r., część właścicieli serwerów EFnetu zdecydowała się stworzyć odrębną sieć, początkowo tylko po to aby testować w niej nowe rozwiązania techniczne, nazbyt nowatorskie aby je od razu zaimplementować w EFnetcie. Testowy IRCd uzyskał nazwę **icru** i był od tej pory rozwijany niezależnie. Ze względu na to, że nowa sieć zyskała sporą popularność, zdecydowano się jej nie zamykać po zakończeniu prac na IRCd 2.7. Undernet do dzisiaj ma charakter sieci "dla wtajemniczonych", w której testuje się rozmaite, techniczne nowinki. M.in. to właśnie w tej sieci wprowadzono po raz pierwszy [serwis](#) umożliwiające rejestrację kanałów i nicków, które w odróżnieniu od innych sieci nie są oparte na osobnym module dołączanym do głównego oprogramowania, lecz są weń integralnie włączone. W icru po raz pierwszy zaimplementowano też bramkę zwaną **CSservice** - umożliwiającą zarządzanie serwisami rejestrującym przez stronę WWW.

IRCd 2.8 był przełomową wersją demona, gdyż poza Undernetem, został on zaakceptowany przez wszystkie większe sieci IRC i prawie wszystkie późniejsze wersje IRCd wywodzą się z 2.8 i są z nią zazwyczaj wstecznie zgodne. Lista komend dopuszczalna w IRCd 2.8 stała się swoistym kanonem, który jest zaimplementowany we wszystkich klientach IRC. W ramach IRCd 2.8 wprowadzono dłuższe, 16-znakowe nazwy kanałów i nicków, przyjęto znak "#" jako wyróżnik zwykłych kanałów "!" dla kanałów szczególnie chronionych i "&" dla kanałów lokalnych - funkcjonujących tylko w ramach jednego serwera.

W 1996 nastąpił tzw. wielki split, który spowodował podzielenie się EFnetu na sieć głównie amerykańską (zwaną dalej EFnetem) oraz "resztę świata" (głównie Europę), która nazwała się **IRCnetem**, nawiązując tym do pierwotnej idei Oikarinena. Obie te sieci uważają się za bezpośrednie potomkinie sieci założonej przez Oikarinena i odsadzają od czci drugą stronę, choć obecnie te stare spory straciły już na ostrości. Źródłem sporu były z jednej strony kontrowersje wokół dalszego rozwoju IRCd, a z drugiej strony dyskusje na temat kodeksu dla IRCopów i ustalenia ściślejszych zasad przyłączania nowych serwerów.

W ramach IRCnetu ustalono dość ściśle regulaminy dla właścicieli serwerów, oraz podjęto prace nad ustanowieniem "oficjalnej" specyfikacji protokołu IRC, co zaowocowało wydaniem serii dokumentów RFC, na podstawie których działa IRCnet. Pozostałe duże sieci IRC ignorują jednak RFC IRCnetu. W ramach IRCnetu ustanowiono oficjalną grupę programistów, którzy rozwijają w kontrolowany sposób IRCd, wypuszczając kolejne jego oficjalne wersje, przy czym wszyscy właściciele serwerów mają w tej sieci obowiązek

unacześniania swojej wersji oprogramowania pod groźbą odłączenia. W ramach IRCnetu rozwijana jest tzw. główna linia IRCd, która jest numerowana poczynając od 2.8, a więc 2.9, 2.10, 2.11 itd. Ta linia IRCd ma ściśle zdefiniowany protokół komunikacji między serwerami, który uniemożliwia przyłączanie do nich serwerów spoza tej serii.

W EFnetcie pozostawiono większą swobodę działania programistów. Powstały tam dwie odmiany IRCd 2.8 - 2.8+CS i 2.8+th, które funkcjonowały równolegle. Później, włączono do kodu 2.8-th pewne fragmenty CS i powstał w ten sposób **IRCd-hybrid** - który jest nadal rozwijany przez duży zespół programistów. IRCd-hybrid posiada nieco inny schemat flag niż "kanoniczny IRCd 2.8", umożliwia zaimplementowanie flagi tzw "półopa" oraz posiada możliwość dołączania modułów zmieniających protokół komunikacji między serwerami oraz dodawanie do sieci różnych wariantów serwisów dodatkowych. IRCd-hybrid, mimo że wywodzi się z EFnetu, rozwija się w oderwaniu od konkretnej sieci, próbując ustanowić ponadsieciowy standard. Jego mutacje z różnymi modułami komunikacyjnymi i serwisowym są rozwijane w ramach samych sieci. M.in:

- **IRCd-rathbox** - obecnie zalecany IRCd EFnetu
- **IRCd-hyperion** - na którym działa **Freenode**
- **IRCd-bahamout** - na którym działa **DALnet**

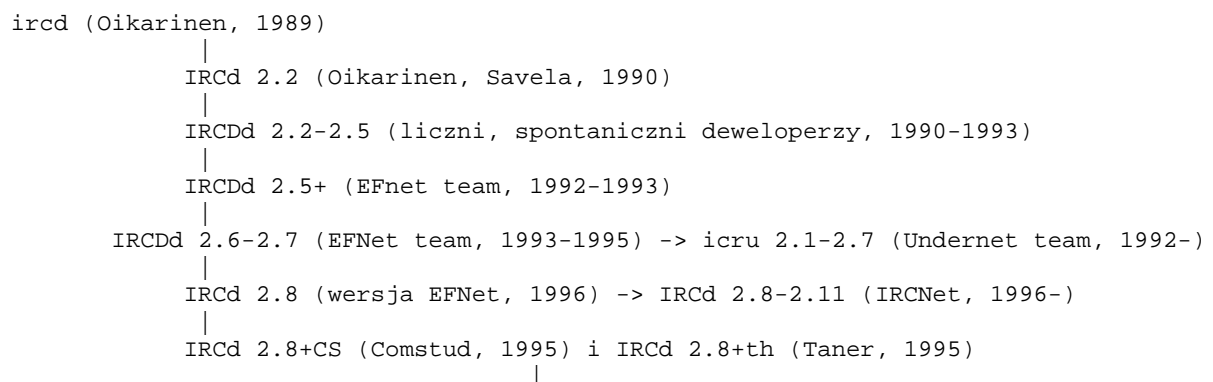
i wiele innych

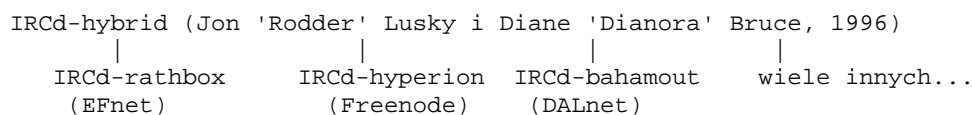
Oprócz tych dwóch głównych historycznych linii rozwojowych IRCd, które wszystkie są oparte na licencji GPL, istnieją też demony IRC, które były tworzone od zera bez korzystania z kodu objętego tą licencją. Zalicza się do nich m.in:

- **UnrealIRCd** i **QuakeIRCd** - na których działa **QuakeNET**
- **IRCX** - na którym działa korporacyjna sieć IRC **Microsoftu** i którą Microsoft oferuje też w wolnej sprzedaży
- **ConferenceRoom** produkcji **Webmaster Inc** - komercyjna wersja IRC przeznaczona głównie do używania przez bramki WWW oparte na apletach Java, sprzedawana razem z narzędziami do tworzenia profesjonalnie wyglądających bramek IRC

i wiele innych.

Schemat rozwojowy IRCd wywodzącego się z pierwotnego demona Oikarinena można przedstawić następująco:





IRCD a klienty

Ze względu na to, że generalnie różne mutacje IRCD stosują nieco odmienne protokoły IRC, może się zdarzyć, że Twój klient będzie miał kłopoty ze współpracą z daną siecią. Ze względu na mnogość IRCD i klientów trudno jest dokładnie podać, które klienty nie współpracują z którymi sieciami. W przypadku wyboru danego klienta do danej sieci warto w pierwszej kolejności poczytać, co o tym piszą na jej stronie WWW. Oprócz tego do niekompatybilnych klientów IRC istnieją często specjalne łątki lub skrypty, które dostosowują ich działanie do danej sieci.

Ogólnie można powiedzieć, że:

- W zasadzie wszystkie aktualne wersje popularnych klientów IRC obsługują sieci, których IRCD wywodzą się z wersji 2.8 lub są zgodne z IRCD-hybrid.
- Część popularnych klientów ma problemy z łączeniem się z Undernetem - ze względu na jego niezgodność z IRCD 2.8 - do niektórych z nich są jednak dostępne łątki.
- Niektóre rodzaje serwerów tworzonych od zera wymagają dodania do klienta specjalnie przygotowanej łątki lub skryptu, bez którego z serwerem nie da się połączyć.

Rada

pomocą w ustaleniu czy nasz klient będzie kompatybilny z daną siecią służyć może porównanie tabeli w rozdziale [Klienty](#) i tabeli na końcu tego rozdziału.

Mając już kompatybilnego z daną siecią klienta możemy mimo to natknąć się na problem z połączeniem z serwerem z przyczyn niezależnych od tej kompatybilności. Najczęściej spotykane problemy to:

- **Zły nasz adres:** serwer odpowiada: *You are not authorized to use this server*. Adres może być "zły" z powodu:
 - jest z zakresu niedozwolonego dla danego serwera (np: serwer warszawa.irc.pl akceptuje tylko adresy z centralnej i północnej Polski); rozwiązanie: poszukać właściwego dla nas serwera danej sieci (np: na jej stronie WWW)
 - adres nie ma zwrotnego DNS - niektóre serwery odrzucają połączenia z tego rodzaju adresów, aczkolwiek jest to coraz rzadsze, ze względu na rozpowszechnienie się adresów NAT i dynamicznych, które z zasady nie mają zwrotnego DNS; z naszej strony nie da się tego problemu rozwiązać, chyba że zmienimy dostawcę internetu lub poprosimy obecnego o przyznanie nam stałego numeru IP ze zwrotnym DNS, co jednak może być niewykonalne dla niego

- mamy adres [Ipv6](#) - a IRCd serwera nie potrafi rozpoznawać tego rodzaju adresów; należy albo poszukać serwera danej sieci, który radzi sobie z adresami Ipv6, albo zmienić naszego dostawcy internetu lub poprosić go o przyznanie nam numeru Ipv4, co może się jednak okazać dla niego niewykonalne.
- **Zły adres serwera:** nasz klient pisze: *Unable to resolve IRC server name*
 - wpisaliśmy po prostu zły adres serwera do komendy `"/server"`; trzeba go wpisać starannie jeszcze raz
 - serwer używa adresu Ipv6 a nasz klient lub pierwotny serwer DNS naszego dostawcy internetu nie umie obsługiwać tego rodzaju adresów; sprawdzić czy nasz klient obsługuje Ipv6, jeśli tak to jedyna rada zmienić dostawcy internetu lub poszukać innego serwera danej sieci
 - po prostu nie mamy połączenia z internetem; rozwiązanie: sprawdzić kable, sprawdzić konfigurację sieciową naszego komputera; sprawdzić czy ostatecznie zapłaciliśmy rachunek za internet itp.
- **Problemy z łącznością:** nasz klient pisze "Connection timed out". Dzieje się tak wtedy, kiedy klient znajduje serwer, wysyła do niego prośbę o połączenie i nie dostaje żadnej odpowiedzi
 - łącze między nami i serwerem jest zbyt powolne albo nie można go skutecznie zestawić; można poszukać innego serwera sieci, który może ma lepsze połączenie z nami
 - nasz dostawca internetu zablokował w ogólnej [zaporze sieciowej](#) możliwość korzystania z portów w typowym zakresie dla IRC (6660-9); należy się o to spytać administratora i ew. zmienić dostawcy internetu - niektóre serwery dla tego rodzaju użytkowników udostępniają nietypowe porty, o czym można zwykle przeczytać na stronie WWW danej sieci
 - sami zablokowaliśmy sobie możliwość korzystania z IRC przez np: restrykcyjne ustawienia naszej prywatnej zapory sieciowej; należy wtedy po prostu zmienić te ustawienia.

Lista znanych IRCd

IRCd	Twórcy/ Obsługiwana sieć	Źródło	Język programowania	Pierwsze wydanie	Ostatnie wydanie	Licencja udostępniana
Bahamut	DALnet	DreamForge i Hybrid	C	2002	bahamut-1.8(03)	GPL
chaosircd	Roman Senn	napisane od zera	C	2003	?	GPL
ConferenceRoom	WebMaster Incorporated	Od zera	C++	1996	ConferenceRoom/3.0	komercyjny
dancer-ircd	Freenode (obecnie porzucony)	Hybrid 6	C	?	(zastąpiony przez hyperion-ircd)	GPL
DreamForge IRCd	DALnet (obecnie porzucony)	?	C	?	(Wymieniony na bahamuta)	GPL
Microsoft Exchange	Microsoft	IRCX	C (Theorised)	1997	(~6.0.6249.0 / Service Pack 3 ?)	komercyjny
hyperion-ircd	Freenode	hybrid6-dancer-ircd	C	2002	hyperion-1.0.2(230)	GPL
ignitionServer	The Ignition Project	Pure-IRCD (VB6 version)	Visual Basic	2004	ignitionServer-0.3.6-P1.20050515	GPL
InspIRCd	ChatSpike , Brain, Craig, et al.	Od zera	C++	2002	InspIRCd-1.0.4	GPL
IRCD	IRCnet	oryginalny ircd	C	1988	irc2.11.1p1	GPL
IRCD-Hybrid	IRCD-Hybrid Development Team	irc2.8 (IRCD)	C	?	ircd-hybrid-7.2.1	?
ircd-ratbox	EFnet	2.8/hybrid	C	?	ircd-ratbox-2.1.8	GPL
ircu	Undernet Coder Committee	irc2.7 (IRCD)	C	1993	ircu2.10.12.04	GPL
IRCXpro Server	IRCXpro, Microsoft	IRCPlus	Visual Basic	?	?	komercyjny
Nefarious IRCu	Evilnet Development, AfterNET	ircu2	C	2004	u2.10.11.07+Nefarious(0.4.0)+[1.347]	GPL
PTlink IRC Server	PTlink IRC Software	ircd-hybrid6	C	2001 (v.6)	Hybrid6/PTlink6.19.6	GPL
QuakeIRCd	QuakeNet development team	UnrealIRCd	C	1997	QuakeIRCd-1.8	GPL
UnrealIRCd	UnrealIRCd Team	DreamForge	C	1999	Unreal3.2.5	GPL
UltimateIRCd	ShadowRealm Creations	DreamForge	C	?	UltimateIRCd(Tsunami)-3.0(01)	GPL

Źródło: Wikipedia-en: [Comparison of IRCd](#)

Dodatkowe serwisy

Serwisami nazywa się w sieciach IRC specjalne, dodatkowe usługi, które umożliwiają między innymi rejestrowanie i nadzorowanie kanałów bez potrzeby stawiania botów, rejestrowanie własnego nicka, zostawianie komunikatów innym zarejestrowanym użytkownikom, pozyskiwanie pomocy i wiele innych rzeczy.

Serwisy, zwłaszcza nadzorujące kanały, często są mylone z botami, gdyż z punktu widzenia użytkownika zachowują się one podobnie jak boty. Podobieństwo to jest jednak pozorne, gdyż serwisy nie łączą się jak klienci z serwerem, lecz działają bezpośrednio na nim. Niektóre IRCd (oprogramowanie serwerów) mają serwisy wbudowane bezpośrednio w ich kod, z kolei inne współpracują z niezależnymi modułami serwisowymi pracującymi jako oddzielne procesy. Do najbardziej znanych tego rodzaju pakietów należą: Anopa, srvx, Atheme, hybserv, Epona i IRC Services 5.

X Undernetu

Pierwszą siecią z zaimplementowanym serwisem był Undernet, gdzie w oprogramowanie **ircu** włączono integralnie usługę o nazwie **X**. X jest rodzajem "pseudobota", który jest stale obecny w sieci i który posiada uprawnienia **ircopa**.

W odróżnieniu od innych tego rodzaju serwisów, X służy tylko do rejestracji i ochrony kanału i nie można za jego pomocą rejestrować nicka. Zamiast tego, aby zacząć korzystać z X trzeba się zarejestrować na specjalnej stronie **CSservice** (<http://cservice.undernet.org/>) i następnie używać tego **username** w trakcie logowania do Undernetu. W trakcie rejestracji trzeba podać wiele danych osobowych (łącznie z adresem zamieszkania i numerem telefonicznym) co służy wykluczeniu możliwości wielokrotnego rejestrowania. Aby użyć X do rejestracji kanału i jego pilnowania, trzeba najpierw samemu założyć ten kanał i przez minimum 100 dn go utrzymywać i znaleźć po tym czasie 10 innych zarejestrowanych osób, które poprą "oficjalne" ustanowienie danego kanału. Ustanowienie to jest możliwe po złożeniu specjalnego "podania" w CSservis, które jest rozpatrywane przez *Channel Service Committee*, po którego zgodzie dopiero można zacząć zarządzać kanałem za pomocą serwisu X. Jeden zarejestrowany użytkownik może rejestrować tylko jeden kanał. *Channel Service Committee* jest raz do roku wybierany przez wszystkich zarejestrowanych użytkowników Undernetu.

Serwis X można obsługiwać z poziomu IRC poprzez wysyłanie odpowiednich komend przez **/msg** lub poza IRC poprzez stronę CSservice. Osoba która zarejestrowała kanał, dostaje automatycznie najwyższy poziom uprawnień tzw. 500 (*Menedżer kanału*) i następnie może według własnego uznania przydzielać innym użytkownikom kanału różne, niższe poziomy uprawnień: 450 (*Zaufany Administrator Kanału*), 400 (*Administrator Listy Accessów*), 200, (*Operator Listy Userów*), 100 (*Operator Kanału*), 75 (*Nowy Operator Kanału*), 50 (*Regularny uczestnik kanału*), 1 (*początkujący uczestnik kanału*), 0 (*wszyscy pozostali*).

W zależności od poziomu uprawnień uczestnicy kanału mają dostęp do określonego zestawu komend X. **1 i 50** mogą w zasadzie tylko wykonać komendy, dzięki którym uzyskują pewne informacje o kanale i jego użytkownikach:

/msg X STATUS #kanał

wyświetli szereg informacji o danym kanale

/msg X SHOWCOMMANDS #kanał

pokaże listę komend, do których jesteśmy na danym kanale uprawnieni

/msg X ACCESS #kanał -min 75

pokaże listę użytkowników, którzy mają poziom minimum 75.

Użytkownik mający **poziom 75** uzyskuje możliwość wykopywania z kanału i zmiany topicu:

/msg KICK #kanał nick

/msg TOPIC #kanał tekst nowego topicu

Użytkownicy **poziomu 100** otrzymują prawo do stosowania komend umożliwiających bezpośrednio zarządzanie kanałem. Są to m.in:

/msg X OP #kanał nick1, nick2, nick3

nadanie statusu opa na na danym kanale określonym użytkownikom

/msg X DEOP #kanał nick1, nick2, nick3

zabiera opa

/msg X BAN #kanał klucz powód czas

zakłada bana na określony czas z określonego powodu użytkownikowi spełniającemu określony klucz wyszukiwania

/msg X UNBAN #kanał klucz

zdejmuje bana.

Użytkownicy **poziomu 200 i 400** uzyskują możliwość nadawania i zmieniania innym statusu niższego od własnego przy pomocy komendy:

/msg X ADDUSER #kanał nick poziom

rejestracja i nadanie określonego poziomu użytkownikowi

/msg X MODINFO #kanał Access nick poziom

zmiana poziomu danego użytkownika.

Użytkownicy **poziomu 450** uzyskują możliwość przywoływania i wycofywania X-a z kanału:

/msg X JOIN #kanał

przywołanie X-a na dany kanał

/msg X PART #kanał

wycofanie X-a z kanału.

Dzięki temu, że X ma status IRCopa kanałem można sterować gdy X nie "siedzi" na kanale - użytkownicy są wtedy opowani, deopowani, wykopywani i banowani w sposób sprawiający

wrażenie jakby to robił sam serwer IRC. Trwanie Xa "na posterunku" powoduje jednak, że kanał cały czas istnieje, nawet wtedy, gdy wszyscy go akurat opuścili.

Wreszcie użytkownicy **poziomu 500** mają pełen dostęp do komendy "SET", która umożliwia zmianę konfiguracji kanału na X. Np:

```
/msg X SET #kanał StrictOp ON
```

powoduje, że status opa na kanale mogą mieć tylko użytkownicy posiadający na X-ie 50 lub wyżej, pozostałym, gdy go przypadkiem dostaną X automatycznie go odbierze.

Komenda SET posiada dziesiątki opcji, które można poznawać stopniowo w miarę potrzeb. Zawsze można też je wyświetlić przy pomocy komendy:

```
/msg X HELP SET
```

Najczęściej spotykany schemat serwisowy

Pionierski system serwisowy Undernetu jest przez wiele osób uważany za nadmiernie sformalizowany i trudny do praktycznego stosowania. Stanowił on jednak rodzaj wzorca dla późniejszych serwisów działających w innych sieciach. Najczęściej spotykany obecnie system serwisowy wywodzi się tego, który opracowano dla sieci **DALnet**. Systemy serwisowe są czasem wbudowane bezpośrednio w kod serwera (np: DALnetowy **behemout** i **hyperion** działający we freenode) lub działają w postaci osobnego pakietu (Anope, srvx, Atheme, hybserv, Epona, IRC Services 5). Jakkolwiek systemy te różnią się w niektórych szczegółach, ich ogólny schemat jest w zasadzie bardzo zbliżony.

Na schemat ten składają się zawsze minimum 2 pseudoboty:

- **NickServ** - służący do rejestrowania nicków
- **ChanServ** - działający z grubsza tak jak Undernetowy X, tylko bez konieczności przechodzenia przez biurokratyczny proces rejestrowania nowego kanału; na ogół przy pomocy ChanServu można po prostu zarejestrować każdy kanał, którego nikt wcześniej jeszcze nie zarejestrował.

Oprócz tego, w niektórych wersjach tego rodzaju serwisów dostępne są też:

- **MemoServ** - służący do zostawiania powiadomień dla innych
- **StatServ** - gromadzi i udostępnia rozmaite statystyki na temat kanału, serwera i całej sieci
- **SeenServ** - gromadzi informacje o tym, kiedy dany nick ostatni raz był na serwerze, danym kanale i kiedy ostatni raz się odezwał
- **HelpServ** - system pomocy dla początkujących
- **OperServ** - wsparcie dla IRCopów, niedostępne dla innych użytkowników
- **BotServ** - umożliwiający rejestrowanie prywatnych botów stawianych na kanale.

Uwagi techniczne

Aby się zorientować, z jaką wersją serwisu ma się do czynienia w danej sieci, najprościej jest użyć komendy **/msg HelpServ HELP**; oprócz tego zwykle każdy z pseudobotów ma swój własny system pomocy, który można zobaczyć przy pomocy komendy **/msg bot HELP** gdzie pod "bot" należy wstawić nazwę danego pseudobota. Komenda **/msg bot HELP komenda** wyświetla szczegółową informację o danej komendzie. Dla ułatwienia pracy z pseudobotem najlepiej jest użyć komendy **/query bot** i następnie "rozmawiać" z nim w osobnym oknie. Wówczas można w podanych poniżej przykładach komend pominąć ich część **"/msg bot"**.

Instrukcja obsługi serwisów freenode

Ze względu na to, że liczba komend obsługiwanych przez wszystkie te serwisy jest zawrotna i różna dla różnych odmian pakietów serwisowych, zapoznamy się z jego działaniem na przykładzie sieci **freenode** (http://freenode.net/irc_servers.shtml).

NickServ

Zaczynamy od **zarejestrowania się** w NickServie, bez czego nie będziemy mogli dalej działać. Robi się to komendą:

/msg NickServ REGISTER hasło - gdzie hasło = jakieś wybrane przez nas hasło.

Później można (ale nie trzeba koniecznie) umieścić w NameServie rozmaite informacje o sobie przy pomocy komendy "SET". Np:

/msg NickServ SET EMAIL adres@e-mail

możemy umieścić kontaktowy e-mail, który będzie widoczny, gdy ktoś wykona komendę:

/msg NickServ INFO nick

Gdy użyjemy komendy:

/msg NickServ SECURE ON w połączeniu z **/msg NameServ SET KILL ON**

użytkownik, który pod Twoją nieobecność pojawi się z Twoim nickiem w sieci, zostanie ostrzeżony przez NickServa, że nick jest zarejestrowany i gdy w przeciągu minuty nie zmieni nicka na jakiś inny zostanie odłączony od sieci. Trzeba pamiętać, że gdy sam/sama będziesz się logował/a do sieci, będziesz musiał/a zarejestrować się w NickServie i będziesz mieć na to również tylko minutę. Rejestrowanie wykonuje się komendą:

/msg NickServ IDENTIFY hasło gdzie hasło = wybrane przez siebie wcześniej hasło.

Gdy zrezygnujesz z ustawienia "KILL ON", NickServ będzie tylko ostrzegał użytkownika stosującego twój nick, że jest on zarejestrowany, ale nie będzie go usuwał z serwera. W momencie gdy ty zalogujesz się na serwer i będziesz chciał/a odzyskać swój nick, będziesz mógł/mogła to zrobić przy pomocy komendy:

/msg NickServ RECOVER nick hasło

co da "złodziejowi" nicka minutę na zmianę hasła, albo zostanie on/ona odłączony/a od serwera.

Pozostałe komendy NickServa

ACCESS nick adres albo klucz - modyfikuje listę adresów (lub ich kluczy), z której będziemy automatycznie rozpoznawani jako zarejestrowany użytkownik - po zdefiniowaniu "accessu", gdy będziesz działał/a spod adresu zgodnego z kluczem użytym w tej komendzie, nie będziesz musiał/a przy każdym logowaniu do sieci używać komendy IDENTIFY.

LIST klucz - wyświetla listę zarejestrowanych użytkowników spełniających kryteria podane w kluczu.

GHOST nick - usuwa naszego własnego "ducha" (zawieszoną sesję), jeśli po przypadkowym odłączeniu się od sieci zdarzyło nam się go zostawić na serwerze. Jest to przydatne, gdyż w momencie gdy "duch" ciągle jest na serwerze nie można działać spod własnego nicka, gdyż on go blokuje.

LINK nick1 nick2 - jeśli masz zarejestrowane dwa różne nicki możesz w ten sposób je z sobą "połączyć" dzięki czemu będzie je można obsługiwać przy pomocy jednego hasła i będą miały one wspólne ustawienia tworzone komendą SET.

UNLINK - znosi wcześniej ustawione połączenie dwóch zarejestrowanych nicków.

ChanServ: rejestracja kanału

Gdy już załatwiłeś/aś sprawy z NickServem, możesz przystąpić do rejestracji nowego kanału. Najpierw jednak warto sprawdzić, czy ktoś wcześniej już nie zarejestrował kanału o tej samej nazwie. Najprościej jest to zrobić komendą:

/msg ChanServ INFO #kanał

Jeśli kanał jest zarejestrowany otrzymasz informację przez kogo i kiedy, oraz być może opis tego kanału. Jeśli nie jest, ChanServ odpowie:

The channel [#kanał] is not registered.

Aby zarejestrować kanał należy wykonać komendę:

/msg ChanServ REGISTER #kanał hasło

Aby później móc zarządzać kanałem trzeba przy każdym zalogowaniu się do sieci wykonać komendę:

/msg ChanServ IDENTIFY #kanał hasło

nawet jeśli wcześniej rozpoznał cię już NameServ. Jeśli kanał przestał być potrzebny można go wyrejestrować komendą:

/msg ChanServ DROP #kanał hasło

W większości sieci, w tym też w omawianym freenode, rejestrowanie kanału, z którego się nie będzie aktywnie korzystać jest niemile widziane. W wielu sieciach, kanał nieużywany przez określony czas (np. 120 dni w freenode) jest automatycznie wyrejestrowywany bez powiadamiania o tym fakcie jego właściciela. Po wyrejestrowaniu kanał może zarejestrować ktoś inny.

ChanServ: konfiguracja kanału

Kanał konfiguruje się za pomocą komendy SET, do której ma się dostęp, gdy jest się właścicielem danego kanału. Ogólna składnia komendy SET:

/msg ChanServ SET #kanał OPCJA parametr

Do najbardziej przydatnych opcji należy:

PASSWORD hasło - ustawia hasło potrzebne do potwierdzenia, że jest się właścicielem kanału.

SECURE ON - włącza ochronę kanału - zmiany trybów i innych ustawień kanału będą mogły dokonywać tylko osoby posiadające odpowiedni poziom uprawnień.

SECUREOPS ON - powoduje, że tylko użytkownicy z poziomem uprawnień **cmdops** (patrz następny podrozdział) będą mogli mieć opa na kanale - pozostałym ChanServ będzie go automatycznie odbierał.

TOPICLOCK ON - powoduje, że topic kanału będzie można zmieniać tylko przy pomocy ChanServa.

GUARD ON - powoduje, że ChanServ wchodzi na kanał i stale na nim "siedzi".

MLOCK ON - powoduje, że ChanServ nie będzie pozwalał zmienić trybu kanału.

Opcje, które włącza się przez parametr "ON" można wyłączyć podając tę samą komendę z parametrem "OFF".

Inne, mniej ważne opcje

URL adres strony www - umieszcza w bazie danych adres strony WWW związanej z kanałem, który się wyświetli gdy ktoś wykona komendę LIST.

EMAIL adres@e-mail - umieszcza w bazie danych e-mail kontaktowy dla danego kanału.

ALTERNATE nick - dodaje alternatywną osobę kontaktową do obsługi kanału.

PRIVATE ON - kanał nie będzie pokazywany na liście generowanej komendą LIST.

ENTRYMSG wiadomość - każdy wchodzący na kanał będzie dostawał **notice** o określonej tą komendą treści.

SPLITOPS OFF - ChanServ nie będzie automatycznie odbierał opa wszystkim, którzy go dostali od serwera po zakończeniu splitu - jest to jedyna opcja, która jest ustawiona standardowo z parametrem "ON".

VERBOSE ON - ChanServ będzie informował wszystkich ze statusem **chanop** o zmianach na liście zarejestrowanych użytkowników kanału.

ChanServ: nadawanie uprawnień użytkownikownikom

Zaufanych użytkowników kanału, którym chce się dać określone uprawnienia do kontrolowania swojego kanału poprzez ChanServa można dopisać komendą:

```
/msg ChanServ ACCESS #kanał ADD nick (lub klucz) poziom_uprawnień
```

gdzie poziom_uprawnień to liczba od -1 do 100. Standardowa hierachia poziomów uprawnień wygląda następująco:

-1 **autodeop** - użytkownik z tym poziomem będzie automatycznie deopowany po wejściu na kanał.

OFF **autovoice** - użytkownik z tym poziomem będzie automatycznie dostawał flagę voice (+) po wejściu na kanał; poziom ten jest standardowo niezdefiniowany.

OFF **autoop** - użytkownik będzie automatycznie opowany po wejściu na kanał; standardowo ChanServ nikogo automatycznie nie opuje - dopiero po ustaleniu wartości poziomu uprawnień dla funkcji autoop będzie to czynił w stosunku do użytkowników posiadających ten poziom.

5 **cmdinvite** - użytkownik będzie miał prawo używać komendy INVITE w ChanServie.

8 **cmdvoice** - użytkownik będzie miał prawo używać komendy VOICE w ChanServie.

10 **cmdop** - użytkownik będzie miał prawo używać komendy OP ChanServa.

10 **cmdunban** - użytkownik będzie miał prawo używać komendy UNBAN ChanServa.

10 **topic** - użytkownik będzie miał prawo zmieniać topic kanału, gdy w Chanservie jest ustawiona opcja TOPICLOCK; gdy TOPICLOCK jest wyłączony to ten poziom uprawnień oczywiście nie ma żadnego praktycznego sensu.

15 **autoreem** - użytkownik będzie miał prawo używać komendy AUTOREM ChanServa.

20 **cmdclear** - użytkownik będzie miał prawo używać komendy CLEAR ChanServa.

25 **set** - użytkownik będzie miał prawo używać komendy SET ChanServa

30 **access** - użytkownik będzie miał prawo nadawać poziom uprawnień komendą ACCESS innym użytkownikom.

50 **level** - użytkownik będzie miał prawo używać komendy LEVEL ChanServa.

100 - użytkownik będzie miał takie same prawa jak właściciel kanału.

Standardowy schemat poziomów uprawnień można zmieniać komendą:

```
/msg ChanServ LEVEL #kanał SET zakres_uprawnień poziom_uprawnień
```

gdzie zakres_uprawnień to jego nazwa zaznaczona wyżej wytłuszczeniem. Na przykład komenda:

```
/msg ChanServ LEVEL #xyz SET autoop 10
```

ustawi poziom uprawnień potrzebny do tego aby ChanServ automatycznie opował użytkownika po wejściu na kanał #xyz, na wartość 10. Z komendy LEVEL SET trzeba korzystać bardzo ostrożnie. Na przykład ustawienie:

```
/msg ChanServ LEVEL #xyz SET set -1
```

spowoduje, że dokładnie każdy będzie mógł zmienić ustawienia kanału #xyz w ChanServie.

Inne komendy ChanServa

AUTOREM #kanał (ADD, DEL lub LIST) nick lub klucz

dodaje "ADD", zdejmuje "DEL" lub pokazuje listę (LIST) użytkowników, którzy są automatycznie wykopywani i banowani na kanale.

LIST klucz

wyświetla listę zarejestrowanych kanałów spełniających warunki podane w kluczu.

GETKEY #kanał

podaje hasło potrzebne do wejścia na kanał zabezpieczony hasłem (pracujący w trybie +k).

INVITE #kanał nick

ChanServ wysyła zaproszenie do wejścia na kanał pracujący w trybie +i; gdy nick jest pominięty zapraszany jest wysyłający tę komendę.

OP #kanał nick

ChanServ opuje osobę o danym nicku; gdy nick jest pominięty zapraszany jest wysyłający tę komendę.

VOICE #kanał nick

ChanServ nadaje status +v osobie o danym nicku; gdy nick jest pominięty status +v dostaje wysyłający tę komendę.

UNBAN #kanał

likwiduje całą listę banów danego kanału.

Q i L QuakeNetu

QuakeNet ma dość nietypowy, jedyny w swoim rodzaju system serwisowy. W jego skład wchodzi obecnie kilkanaście pseudobotów, z których najważniejsze są jednak **Q** i **L**. W QuakeNecie, podobnie jak w Undernecie nie ma możliwości bezpośredniego rejestrowania nicków, jednak użycie **Q** lub **S** jest pośrednio związane z rejestracją w nim swojego *username* i klucza adresu IP, spod którego będzie się rozpoznawanym przez te pseudoboty. Oba pseudoboty pełnią rolę nadzorców kanału, z tymże **L** jest wersją uproszczoną serwisu, przeznaczoną do pracy na niewielkich kanałach, a **Q** wersją pełną, przeznaczoną do obsługi dużych kanałów.

Rejestracja i obsługa swojego konta na Q

Pracę z oboma pseudobotami rozpoczyna się od zarejestrowania się na **Q**, gdyż **L** korzysta z danych rejestrowanych gromadzonych przez **Q**. Robi się to za pomocą komendy:

```
/msg Q HELLO adres@e-mail
```

Q wysyła na podany adres e-mail z hasłem, który trzeba użyć w kolejnej komendzie:

/msg Q@CServe.quakenet.org AUTH username hasło

Na co Q odpowiada:

Q(TheQBot@CServe.quakenet.org)- You have authed as userid: xxxx nick: yyyy

I od tej pory można już używać serwisu Q i L. Gdybyś chciał w przyszłości sprawdzić, czy jesteś zarejestrowany możesz zrobić to komendą:

/msg Q whoami

Aby zmienić swoje hasło piszesz:

/msg Q@CServe.quakenet.org NEWPASS stare_hasło nowe_hasło nowe_hasło
(nowe hasło trzeba powtórzyć dwa razy)

Można też zmienić swój adres e-mail, który się będzie stosowało do kontaktu z Q:

/msg Q@CServe.quakenet.org EMAIL hasło nowy@adres

W przypadku gdy zapomnisz swoje hasło możesz napisać:

/msg Q REQUESTPASSWORD twój@adres

przy czym "twój@adres" musi być tym, spod którego jesteś w Q zarejestrowany/a - hasło zostanie wysłane e-mailem na podany adres.

Praca z L

Aby wezwać L na kanał należy wykonać komendę:

/msg R REQUESTBOT #kanał

Aby ta komenda zadziałała trzeba spełnić następujące warunki:

- być zarejestrowanym na Q pod nickiem spod którego się tę komendę wysyła
- mieć opa na na kanale
- na kanale musi być w momencie wezwania minimum jeszcze 3 innych użytkowników, którzy nie muszą być zarejestrowani w Q.

Komenda ta nie zadziała od razu - L pojawi się na kanale po od kilku godzin do nawet kilku dni od zgłoszenia. Ponawianie od czasu do czasu tej komendy może trochę przyspieszyć pojawienie się L, ale nie należy wysyłać tego zgłoszenia zbyt często. L jest wycofywany z kanału, gdy nikt na niego nie wchodził przez 20 dni, lub gdy ktoś z operatorów tego kanału poważnie naruszył zasady netykiety QuakeNetu. Nie ma natomiast możliwość wycofania L z kanału samodzielnie.

Gdy L pojawi się na kanale oznacza to, że rejestracja kanału została zaakceptowana. L nie stosuje schematu poziomów uprawnień jak X i ChanServ, lecz przy rejestracji użytkowników kanału przyznaje się im określone flagi wewnętrzne. Są to: **+n** - właściciel (owner), **+m** mistrz (master), **+o** - użytkownik, który ma prawo mieć opę na kanale, **+v** - użytkownik, który ma prawo mieć voice na kanale, **+a** - powoduje, że gdy ktoś ma +o dostaje opę automatycznie po wejściu na kanał, a gdy ma +v dostaje automatycznie voice.

Nowych użytkowników dodaje się komendą:

/msg L ADDUSER #kanał nick

który standardowo nadaje temu użytkownikowi flagę "+ao"; dodawany użytkownik nie musi mieć konta na Q, ale musi być w momencie wykonywania tej komendy obecny na kanale;

/msg L ADDUSER #channel #Qkonto

to samo co wyżej, tylko z wykorzystaniem danych z konta osoby zarejestrowanej wcześniej na Q. Użytkownik zarejestrowany na Q nie musi być obecny na kanale w momencie wykonywania tej komendy. L akceptuje rejestrację maksymalnie 30 użytkowników na kanale, gdy będziemy chcieli zarejestrować ich więcej, będziemy musieli wezwać na kanał Q.

Komenda:

/msg L CHANLEV #kanał

bez parametrów wyświetla pełną listę zarejestrowanych użytkowników. Służy ona także do zmieniania flagi wcześniej zarejestrowanych użytkowników:

/msg L CHANLEV #kanał nick +(-)flagi

odpowiednio dodaje (+) lub odejmuje (-) flagę danemu użytkownikowi zarejestrowanemu pod danym nickiem.

/msg L CHANLEV #kanał #Qkonto +(-)flagi

działa tak samo dla użytkowników zarejestrowanych w Q.

Komenda:

/msg L REMOVEUSER #kanał nick lub **/msg L REMOVEUSER #kanał #Qkonto**

usuwa rejestrację użytkownika na kanale.

Inne przydatne komendy L:

Dostępne dla wszystkich:

HELP - system pomocy.

WHOIS - informacja o użytkowniku.

VERSION - podaje wersję L.

Dostępne, gdy ma się flagę +o na L:

INVITE - L - wyśle zaproszenie na kanał.

OP - L - da opa na kanale.

VOICE - L - da +v na kanale.

Dostępne, gdy ma się flagę +m lub +n na L:

'CLEARCHAN - likwiduje wszystkie aktualne tryby kanału.

RECOVER - służy do odzyskiwania kanału po przejściu go przez kogoś. Deopuje wszystkich aktualnych opów, likwiduje tryb kanału +i i +k i likwiduje wszystkie bany założone na kanale.

UNBANALL - usuwa wszystkie bany z kanału.

DEOPALL - odbiera wszystkim opa na kanale.

WELCOME - zmienia tekst powitalny kanału.

Praca z Q

Q można wezwać na kanał w momencie, gdy jest już na nim L z listą co najmniej 15 zarejestrowanych użytkowników. Może to zrobić tylko właściciel danego kanału. Oprócz tego trzeba spełnić parę innych wymagań, które nie są publicznie ujawniane i o których decyduje specjalna komisja zajmująca się Q. Q wzywa się na kanał dokładnie tak samo jak L:

/msg R REQUESTBOT #kanał

Działa to podobnie jak z L tzn. zastąpi on L na kanale w kilka godzin do kilku dni od wezwania, przy czym to się nie musi stać, gdy komitet uzna, że Q na danym kanale nie jest jeszcze potrzebny. Negatywnej decyzji komitetu nie wolno komentować publicznie ani domagać się wyjaśnienia jej powodów, pod groźbą wycofania z kanału także L i zablokowania konta na Q. W momencie zamiany L na Q na kanale, wszystkie ustawienia kanału i lista jego zaufanych użytkowników jest automatycznie przenoszona z L na Q.

Podstawowy zestaw komend Q jest podobny do L. Część z nich posiada jednak większą funkcjonalność. M.in. na Q można dodać użytkownika kanału z flagą **+b** (permanently zbanowany - użytkownik po wejściu na kanał jest natychmiast wykopywany i banowany) oraz **+t** (uprawniony do zmiany topicu kanału). Oprócz tego Q ma wiele komend do zarządzania kanałem, których nie ma L. Są to m.in:

AUTOLIMIT - ustala maksymalną liczbę osób na kanale.

BAN - zakłada permanenty ban.

BANCLEAR - czyści listę permanentnych banów na Q.

BANDEL - kasuje jeden, określony ban.

BANLIST - wyświetla listę założonych banów.

CHANFLAGS - ustawia forsowanie określonego trybu pracy kanału (flag kanałowych).

SETTOPIC - ustala topic na kanale, którego nikt nie może zmienić z poziomu IRC.

Lista pakietów serwisowych

Anope (<http://www.anope.org>)

osobny pakiet serwisowy, powstały na bazie Epony w 2005 r. początkowo tylko w celu naprawy niektórych błędów Epony. Obecnie rozwija się samodzielnie. W jego skład wchodzi: ChanServ, NickServ, OperServ, MemoServ, HelpServ, BotServ i HostServ. Dane gromadzone przez serwis są w bazie MySQL.

Atheme (<http://www.atheme.org/>)

minimalistyczna wersja pakietu zawierająca uproszczone wersje NickServa, ChanServa, MemoServa i OperServa.

Epona (<http://www.epona.org>)

najbardziej rozpowszechniona wersja "standardowa" pakietu serwisowego, która jest zgodna z bahamoutem DALnetu, zawierająca OperServa, ChanServa, NickServa, MemoServa i BotServa.

Hybserv2 (<http://www.hybserv.net/>)

wersja przeznaczona do pracy z ircd-hybrid rozwijana jako część projektu hybrid - zawiera NickServa, ChanServa, MemoServa i OperServa.

IRC Services (<http://www.ircservices.esper.net>)

kolejna wersja serwisu pretendująca do roli "wersji standardowej" - zawiera NickServa, ChanServa, MemoServa i OperServa.

PTlink (<http://software.pt-link.net/>)

jedna z nielicznych wersji serwisów działająca też pod MS Windows, używana w kilku mniejszych sieciach IRC.

QuakeNet services (<http://www.quakenet.org/>)

opisane już powyżej.

Ratbox services (<http://services.ircd-ratbox.org/>)

aktualnie stosowany w EFnecie - składa się tylko z jednego pseudobota "chanfixa", który służy do odzyskiwania kanałów i nicków w sytuacjach konfliktowych.

srvx (<http://www.srvx.net/>)

minimalistyczna wersja pakietu serwisowego, obsługująca NickServa, ChanServa, HelpServa i OperServa - stworzona jako alternatywa dla serwisu "X" dla icru - stosowanego w Undernecie.

CSservice (<http://cservice.undernet.org/>)

oficjalne wsparcie dla serwisu "X" Undernetu.

Terminologia

A

alias

forma komendy IRC, którą klient przekształca w pełną postać komendy wysyłanej do serwera, co skraca czas ich wypisywania; np: "/j #kanał" może być aliasem komendy "/join #kanał".

Anope

jeden z pakietów oferujących serwisy dodatkowe w IRC.

Athema

inny pakiet serwisowy do IRC.

B

ban

zablokowanie wstępu na kanał dla określonego użytkownika lub grupy użytkowników.

bahamout-IRCD

wersja oprogramowania serwerów IRC stosowana w sieci DALnet.

BitchX

"bojowy" klient IRC dla zaawansowanych użytkowników.

bot

wyspecjalizowany program, który łączy się z serwerem IRC tak jak zwykły klient, ale który nie służy do prowadzenia rozmów lecz do automatycznego realizowania określonych funkcji - np: pilnowania porządku na kanale, rejestrowania jego użytkowników itp.

bomba

metoda oślepienia przeciwnika na IRCu poprzez wysyłanie do jego klienta serii powtarzających się komend CTCP, DCC lub msg, które powodują zawieszenie się jego komputera lub zapchanie jego łącza powodującego zerwanie jego łączności z serwerem.

botlinia

niektóre programy realizujące funkcję bota (Np: eggdropp) posiadają wewnętrzny chat, który jest na stałe połączony z innymi botami podobnego rodzaju działającymi na tym samym lub na innych kanałach a nawet w innych sieciach IRC; tworzy to wewnątrz kanał porozumiewania się widoczny tylko dla operatorów botów.

botnet

sieć botów tworząca hierarchiczną sieć, umożliwiającą skuteczne nadzorowanie kanału z konsoli jednego z botów.

C

chanserv

rodzaj usługi, popularnej w wielu sieciach IRC, która umożliwia rejestrację kanałów a następnie "pilnuje" kanału, dzięki czemu nie istnieje ryzyko trwałego przejęcia kanału (patrz takeover i nameserv).

chanfix

usługa serwisowa dostępna EFnetcie, przy pomocy której można rozstrzygać spory o prawa własności kanałów.

chatzilla

popularny klient IRC - część pakietu Mozilli.

cloak, cloacking

specjalna usługa dostępna w niektórych sieciach, umożliwiająca ukrycie swojego rzeczywistego adresu przed innymi użytkownikami irc i zastąpienie go określonym tekstem np: "wikipedia/<nick>" Patrz [meta:IRC channel cloaks](#).

CSservice

oficjalna bramka WWW do obsługi serwisu "X" działającego w Undernetcie.

ctcp

client-to-client protocol - zbiór komend umożliwiających bezpośrednio pobieranie wybranych informacji z klienta innego użytkownika - umożliwia, uzyskiwanie informacji o nazwie klienta, jego wersji, czasie systemowym, lagu itp.

D

DALnet

Duża sieć IRC działająca w USA, w której powstał schemat dodatkowych serwisów stosowany później w wielu innych sieciach.

dcc

Direct Client-to-Client - zbiór usług związanych z bezpośrednim kontaktem dwóch klientów z pominięciem serwera IRC realizowany przez większość programów do IRC, umożliwiający chat z pominięciem serwera, przesyłanie plików, sprawdzanie laga itp.

deop

odebrania opa użytkownikowi.

duch (ghost)

porzucona, martwa sesja IRC; powstaje na skutek nagłego, nieporownego zerwania łączności z serwerem; użytkownik jest widoczny na kanale, mimo że faktycznie jest rozłączony.

E

EFnet

największa i najstarsza sieć publiczna IRC, działająca głównie w USA.

Eggdrop

najbardziej popularny program służący do stawiania botów.

Epona

pakiet serwisów dodatkowych używany w niektórych sieciach IRC.

event

dowolne zdarzenie które miało miejsce w sieci IRC, pojęcie stosowane przy pisaniu skryptów; *eventami* są np: komunikaty użytkowników wysyłane na kanał, wyjście i wejście na kanał, zmiana trybu kanału lub użytkownika itp.

eXtreme

popularny uniwersalny skrypt do mIRC'a, znany z licznych "wodotrysków", takich jak np: wbudowany mp3 player.

F

flood

wysyłanie na kanał lub bezpośrednio do określonego klienta dużych ilości danych na raz; czasami wykonywane omyłkowo przez początkujących użytkowników a czasami celowo jako rodzaj wandalizmu lub w celu "oślepienia" użytkowników z flagą opa w czasie dokonywania takeoveru.

freenode

jedna z dużych, publicznych sieci IRC; przeznaczona dla osób związanych z projektami wolnego oprogramowania i wolnej treści.

flaga

oznakowanie grup użytkowników kanału posiadających określone przywileje, widoczne na liście użytkowników - "@" - op, "+" - voice' wewnątrz botlinii i niektórych serwisów również występują flagi użytkowników tyle, że inne.

fSERV

rozszerzenie DCC umożliwiające stawianie prymitywnych bibliotek plików do pobrania, zaimplementowana po raz pierwszy w mIRC'u.

G

g-line

to samo co [k-line](#), tyle że nie jest zakładane na jednym serwerze lecz działa globalnie na wszystkich serwerach danej sieci IRC.

ghost

patrz [duch](#).

H

hub

węzłowy serwer IRC, do którego nie można się zalogować i którego zadaniem jest tylko łączyć inne serwery; w botnetach jest to węzłowy bot, do którego są przyłączone wszystkie pozostałe boty

hybrid-IRCD

wersja oprogramowania do serwerów IRC, powstała w EFnecie, z której wywodzi się wiele innych wersji tego oprogramowania, obecnie jest rozwijana jako niezależny projekt.

hybserv2

projekt stworzenia uniwersalnego pakietu serwisowego do współpracy z IRCd-hybrid.

hyperion-IRCD

wersja oprogramowania serwerów IRC, na której działa sieć freenode.

I

infobot

bot, który nie jest przeznaczony do pilnowania kanału lecz do udostępniania przez pytania zadawane na kanale określonych informacji.

ircop

użytkownik posiadający szczególne uprawnienia - ma automatycznie status operatora na wszystkich kanałach irc, może zakładać k-linię i tworzyć bany działające na całych grupach kanałów - status ircopa może nadać wyłącznie operator serwera; zwykle ircopami są operatorzy serwerów; w niektórych sieciach status ircopa mają boty oferujące dodatkowe usługi - patrz nickserv i chanserv

idle time

czas jaki minął od ostatniej akcji danego użytkownika; zwykle serwery po przekroczeniu pewnej, granicznej wartości "idle" uznają użytkownika za "ducha" (zawieszoną sesję) i zrywają z nim łączność.

identd

usługa serwerowa, umożliwiająca serwerowi rozpoznawanie numeru IP logującego się użytkownika, oraz przynajmniej na tej podstawie określoną flagę globalną.

IRCD

skrót od *IRC daemon* - podstawowe oprogramowanie serwera IRC, które realizuje prawie wszystkie jego funkcje, oprócz (w niektórych przypadkach) - serwisów.

ircii

najstarszy klient do IRC, stanowiący wzorzec dla pozostałych, przeznaczony głównie do pracy w systemach opartych na architekturze Unixa.

ircN

popularny skrypt uniwersalny do programu mIRC o charakterystyce zbliżonej do LiCe.

irssi

popularny, względnie łatwy w obsłudze klient do IRC, przeznaczony głównie do pracy w systemach opartych na architekturze Unixa.

IRC Services

popularny pakiet serwisowy do sieci IRC, starający się być standardem w tym zakresie.

J

join

podstawowa komenda IRC powodująca wejście na dany kanał.

K

kanał

miejsce prowadzenia rozmów na IRCu. Kanały oznacza się znakiem hasza (#) na przykład: *#wikipedia*, *#wikinews-pl* - osoby przebywające na raz na jednym kanale widzą wszystkie komunikaty, wysyłane przez uczestników kanału na ten kanał.

k-line

trwały ban dla określonych zakresów numerów IP uniemożliwiający łączenie się z daną siecią IRC - zakładane przez operatorów serwerów dla grup użytkowników nie stosujących się do podstawowych zasad danej sieci

klient

program umożliwiający rozmowy na IRCu, łączący się z serwerem.

kick

wykopanie użytkownika z kanału

klucz

→ patrz maska

L

L

pseudobot stanowiący część pakietu serwisowego QuakeNetu - uproszczona wersja Q pełniąca rolę nadzorcy niewielkich kanałów.

lag

zauważalne spowolnienie przepływu informacji między serwerem i klientem lub między dwoma klientami; zdarza się, że osiąga nawet kilka minut; powoduje czasami nieporozumienia w czasie rozmów na kanale - komenda, która umożliwia zbadanie lagu /ctcp ping <nazwa kanału> lub <nick usera>.

lamer

użytkownik niedouczony, który zachowuje się nieodpowiedzialnie lub zadaje dużo niepotrzebnych pytań, na które odpowiedź można uzyskać w podręczniku do IRC.

LiCe

popularny skrypt uniwersalny do programu ircII i pochodnych, stanowiący wzorzec dla innych tego rodzaju skryptów.

M

maska

ciąg metaznaków umożliwiający wybór grupy użytkowników wg określonego kryterium; stosowane najczęściej w komendach: ban, whois i k-line; metaznaki dozwolone w maskach ircowych są zgodne ze standardem unixa.

massdeop, massban, masskick itd.

masowe odbieranie przywilejów lub banowanie i wykopywanie użytkowników z kanału, zazwyczaj w celu przejęcia kanału, często wykonywane przez wyspecjalizowane w tym boty lub "warskrypty".

mIRC

najbardziej popularny klient IRC pod MS Windows.

memoserv

jeden z serwisów serwerowych - rodzaj usługi, która umożliwia pozostawianie krótkich notatek zarejestrowanym użytkownikom.

msg wysyłanie komunikatu do określonego użytkownika, niewidoczne na kanale.

motd komunikat ustawiany przez operatora serwera, który widzi każdy użytkownik logujący się na dany serwer; później można go obejrzeć przy pomocy komendy /motd.

N

nick pseudonim - nazwa, pod jaką użytkownik jest widziany na IRCu.

nameserv jeden z serwisów serwerowych - specjalna usługa dostępna w niektórych sieciach umożliwiająca rejestrowanie swojego nicka, dzięki czemu jest pewność, że nikt nam go nie odbierze.

nickcollision kolizja nicków - sytuacja zdarzająca się po splicie - polegająca na tym, że w sieci pojawia się nagle dwóch użytkowników z tym samym nickiem - różne protokoły IRC w różny sposób radzą sobie z tym problemem; zawsze jednak na skutek kolizji jeden z użytkowników jest po prostu odłączany od sieci - nickcollision jest czasem wykorzystywane do przejęcia kanałów (patrz takeover).

notice krótki komunikat wysyłany do określonego użytkownika - od msg różni się tym, że w większości klientów pokazuje się on w aktywnym, a nie nowoutworzonym oknie i jest specjalnie wyróżniany kolorem lub czcionką.

noName popularny, zwłaszcza wśród użytkowników QuakeNetu uniwersalny skrypt do mIRCa.

O

operator (op) użytkownik mający na danym kanale specjalne prawa - daje możliwości banowania i wykopywania innych użytkowników z kanału, a także nadawanie przywileju opa innym użytkownikom; flaga "@" - patrz też "ircop".

P

protectskrypt skrypt, którego działanie polega głównie na ochronie użytkownika przed atakami innych użytkowników.

Q

Q

podstawowy psuedobot serwisowy QuakeNetu; pełni rolę nadzorcy dużych kanałów oraz umożliwia rejestrację użytkowników tej sieci.

QuakeNet

jedna z największych, publicznych sieci IRC, używana głównie przez graczy Quake i innych gier komputerowych.

query

to samo co msg, tylko z funkcją uruchomienia osobnego okna do rozmowy.

R

remotes

inna nazwa na **event**

revenge

tryb działania niektórych skryptów, botów i klientów polegający na automatycznym wykopywaniu i banowaniu użytkowników z "czarnej listy" tworzonej automatycznie lub ręcznie.

rathbox-IRCD

wersja oprogramowania serwerów IRC na której działa sieć EFnet.

S

serwer

komputer, do którego trzeba się podłączyć przy pomocy klienta, by móc rozmawiać na IRCu. Serwer należy do sieci i zazwyczaj ma swoją nazwę, na przykład *irc.wikimedia.org*. Aby połączyć się z usługą IRC zazwyczaj korzysta się z [portu](#) 6667.

serwis

dodatkowa usługa dostarczana przez serwery IRC w niektórych sieciach, widoczna dla użytkownika jako rodzaj bota (ChanServ, NameServ, MemoServ, OperServ itp.)

split

pełna desynchronizacja jednego lub więcej serwerów powodująca czasowe podzielenie się sieci na kilka podsieci; ma miejsce zazwyczaj gdy serwery są przeciążone nadmierną liczbą użytkowników lub występuje przeciążenie łącz między nimi; na kanale widać to w postaci masowego zniknięcia grupy użytkowników

skrypt

specjalny program pisany w wewnętrznym języku klienta lub bota IRC, poszerzający w jakiś sposób jego możliwości

shitlist

"czarna" lista użytkowników gromadzona ręcznie lub automatycznie przez skrypty i boty; po wejściu na kanał kogoś z shitlisty, jest on natychmiast banowany i wykopywany z kanału

T

takeover

przejęcie władzy nad kanałem - np. poprzez wyłudzenie flagi opa od jednego z operatorów kanału a następnie szybki, masowy deop dotychczasowych operatorów

topic

tekst przypisywany do kanału, który jest wyświetlany wchodzącym na kanał użytkownikom, a w wielu klientach IRC jest on automatycznie wstawiany do belki okna, w którym jest wyświetlany dany kanał

U

Undernet

sieć IRC, która pierwotnie powstała jako testowa odnoga EFnetu; w której po raz pierwszy zaimplementowano system serwisowy rejestracji kanałów; obecnie istnieje zupełnie niezależnie.

username

to co występuje przed znakiem @ w adresie użytkownika, który łączy się z serwerem IRC; standardowo jest to nazwa konta użytkownika pobrana z jego systemu operacyjnego.

V

voice

flaga umożliwiająca zabieranie głosu na kanale moderowanym

VoiD

bot "obronno-zaczepny" powstały na bazie kod Eggdropa; pod koniec lat 90. XX w. uważany był za niezwyciężonego.

W

warskrypt

skrypt pisany w języku wewnętrznym klienta irc, z wbudowanymi funkcjami przydatnymi przy toczeniu wojny o przejęcie i kontrolowanie kanału

whois

komenda umożliwiająca uzyskanie informacji o danym użytkowniku takich jak adres spod którego pisze, imię, nazwisko, rodzaj używanego klienta, czas zalogowania itp - umożliwia też uzyskanie listy użytkowników spełniających określone kryterium (np: listę wszystkich użytkowników danego kanału)

whitelist

"biała lista" zaufanych użytkowników gromadzona ręcznie lub automatycznie przez boty i skryptu; osoby znajdujące się na białej liście są chronione - np. dostają automatycznie opa po wejściu na kanał i ktoś, kto odbierze opa lub wykopie z kanału kogoś z białej listy jest sam wykopywany i banowany

wielki split

podział sieci EFnet na Efnet i IRCnet, który miał miejsce w 1996 r.

X

X

pseodobot serwisowy sieci Undernet; pełni jednocześnie rolę nadzorcy kanałów i rejestratora użytkowników sieci.

X-Chat

popularny, uniwersalny klient IRC dostępny do wielu systemów operacyjnych.

XDCC

rozszerzenie DCC umożliwiające tworzenie bibliotek plików działających podobnie do poza-IRCowej usługi publicznych FTP.

Y

Z

7

7thSphere

legendarny skrypt bojowy do mIRC'a, posiadający wiele "pirackich" podprogramów, którymi można było m.in skutecznie zawieszać komputery z MS Windows 95.

Autorzy

Użytkownicy, którzy mieli istotny wkład w powstanie tego podręcznika:

- [Polimerek](#)

Spis ilustracji

- Okładka: autor Polimerek, licencja GFDL i Creative Commons-Attribution 1.0, 2.0 i 2.5
- Strony 6, 9, 11, 22: zrzuty ekranu z programu Chatzilla, licencja GPL, LGPL i MPL
- Strona 17: zrzut ekranu z programów ircII i PuTTY, licencja MIT/BSD
- Strona 18: zrzut ekranu z programu BitchX, licencja BSD
- Strona 19: zrzut ekranu z programu irssi, licencja GPL
- Strona 21: zrzut ekranu z programu XChat z tłem autorstwa Bowiego J. Poaga, licencja GPL

Bibliografia

Bibliografia

Źródła polskojęzyczne

1. Grzegorz Popczyk, Marek Wyszynski, *IRC w systemach UNIX i Windows*, Wydawnictwo Helion, 1997, [ISBN 83-86718-73-0](#)
2. Marianne Pyra, *IRC*, Wydawnictwo Helion, 1997, ISBN: 83-86718-65-X
3. [Różni autorzy, dokumentacja zawarta na stronie www.irc.pl](#)
4. Artykuł [IRC](#) w polskiej Wikipedii
5. [Strona Michu o skryptach i botach](#)
6. [Strona Irc4All](#)
7. [strona mirc.pl](#)
8. [Robert Kopaczewski, Boty, czyli ircowi strażnicy, czasopismo INTERNET, 10/2002](#)
9. [Bartosz Kulis, Co o IRC-owym bocie wiedzieć należy?, PC Centre, 4 stycznia 2004](#)
10. [Opis komend Undernetowego serwisu X na stronie kanału #polonia](#)

Źródła anglojęzyczne

1. [Nicolas Pioch, Owe Rasmussen, Michelle A. Hoyle, Joseph Lo, A Short IRC Primer, 1997](#)
2. [Ronald van Loon, IRC Tutorial, 2004](#)
3. [Anonimowi autorzy, The IRC Command Cosmos, an advanced guide to IRC commands, Edition 2.2, August 2002](#)
4. Khaled Mardam Bey, Help do programu mIRC, ver. 6.1, 2003
5. [RFC 2813](#), [RFC 2812](#), [RFC 2811](#), [RFC 2810](#), [RFC 1459](#)
6. [Liczni autorzy, artykuły w kategorii "IRC" w Wikipedii-en](#)
7. [Klaus Zeuge, Troy Rollo, Ben Mesander, The Client-To-Client Protocol, 1994](#)
8. [Liczni autorzy, artykuł "Direct Client-to-Client" z Wikipedii-en](#)
9. [An Introduction to IRC Bots z coste.com](#)
10. [Real hacker start their own IRC networks, tekst na stronie everything2.com](#)
11. [David Brumley, Tracking Hackers on IRC, 2000](#)
12. [JoJo, Selecting A Script, tekst na stronie irchelp.org](#)
13. [Opis skryptu 7thSphere na stronie everything2.com](#)
14. [Curve, Just What Is a Botnet?, The DALnet Ezine](#)
15. [Mandar Mirashi - Mmmm, The History of Undernet, sierpień 1993](#)
16. [Oficjalna strona Undernet CService](#)
17. Anonimowi autorzy, Pomoc on-line serwisów freenode
18. [Oficjalna strona QuakeNetu](#)

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of

mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

