

Messaggistica istantanea (instant messaging) «

37.1	Messaggi sul terminale Unix	1675
37.1.1	Accesso al proprio terminale	1675
37.1.2	Comunicazione diretta attraverso la rete	1677
37.1.3	Invio di un messaggio circolare	1678
37.2	IRC	1679
37.2.1	Canali, utenti e operatori	1679
37.2.2	Divisione e ricongiunzione di reti IRC	1680
37.2.3	Comportamenti spiacevoli	1680
37.2.4	Dal lato del server	1681
37.2.5	Dal lato del cliente	1683
37.2.6	Utilizzo di massa di un cliente IRC	1686
37.3	ICQ: «I-see-you»	1687
37.3.1	Licq	1688
37.3.2	Pidgin	1689
37.4	Abbreviazioni di Internet	1690
37.5	Riferimenti	1691

in.talkd 1677 irc 1683 ircd 1683 ircd.conf 1681 1681
 ircd.motd 1681 licq 1688 mesg 1675 pidgin 1689
 rpc.rwalld 1678 rwall 1678 rwalld 1678 talk 1677
 talkd 1677 tkirc 1683 wall 1675 write 1675 ytalk 1677

La messaggistica istantanea è diventato un concetto importante, tanto da assumere un acronimo diffuso: «IM», ovvero *Instant messaging*. Nei primi anni 2000 è proliferata l'offerta di servizi di messaggistica istantanea, soprattutto per assicurare l'accesso di un pubblico importante ai «portali» di comunicazione. Attualmente questi servizi si attuano generalmente attraverso applicazioni gestibili con un comune navigatore ipertestuale e l'uso di protocolli e programmi specifici è sempre meno diffuso.

In questo capitolo si considerano i tipi tradizionali di messaggistica istantanea e ICQ (*I seek you*), ma ciò soltanto a titolo esemplificativo del problema, perché tutti sono destinati a un progressivo abbandono.

37.1 Messaggi sul terminale Unix

Il modo normale di inviare un messaggio a una persona è quello di utilizzare la posta elettronica. In alternativa, sui sistemi Unix, quando si desidera aprire una comunicazione istantanea può essere conveniente l'uso di programmi come `'talk'`, ammesso che il sistema di destinazione sia predisposto per questo.

Il tipo di comunicazione che utilizza programmi come `'talk'` e simili, parte dal presupposto che si possa «scrivere» sul file di dispositivo corrispondente al terminale utilizzato dall'utente destinatario.

37.1.1 Accesso al proprio terminale

Quando si accede normalmente attraverso un terminale a caratteri, il dispositivo corrispondente dovrebbe appartenere all'utente che lo sta utilizzando e anche al gruppo `'tty'`. Ciò dovrebbe avvenire automaticamente per opera del programma `'login'`. Nel caso dell'utente `'tizio'` che sta utilizzando la seconda console virtuale di un sistema GNU/Linux, si dovrebbero osservare le caratteristiche seguenti.

```
$ ls -l /dev/tty2 [Invio]
```

```
crw--w---- 1 tizio tty 4, 2 dic 31 10:38 /dev/tty2
```

L'utente che utilizza il terminale dovrebbe avere i permessi di lettura e scrittura, inoltre, dovrebbe essere concesso al gruppo il permesso di scrittura. Con questa convenzione, un programma che sia stato avviato con i privilegi del gruppo `'tty'` avrebbe la possibilità di scrivere su questo file di dispositivo.

Scrivere sul file di dispositivo di un terminale significa andare a pasticciare lo schermo su cui sta lavorando presumibilmente un utente. Esistendo questa possibilità, cioè che processi estranei possano aggiungere informazioni allo schermo del terminale che si sta utilizzando, la maggior parte degli applicativi prevede un comando che riscrive il contenuto dello schermo (di solito si ottiene con la combinazione di tasti [*Ctrl I*]). Tuttavia, gli utenti potrebbero desiderare di limitare questa possibilità, eliminando il permesso di scrittura per il gruppo `'tty'` per il terminale che si sta utilizzando.

Per controllare il permesso di scrittura per il gruppo `'tty'` del dispositivo corrispondente al proprio terminale attivo, si può usare anche un programma molto semplice: `'mesg'`.¹

```
mesg [y|n]
```

Il fatto di togliere il permesso di scrittura per il gruppo `'tty'` al dispositivo del terminale, non è una garanzia che nessuno possa scrivervi. Un processo con i privilegi dell'utente `'root'` potrebbe farlo ugualmente. Tuttavia, si tratta di una convenzione che generalmente viene rispettata.

Opzione	Descrizione
y	Permette agli altri utenti di scrivere sul proprio terminale (aggiunge il permesso di scrittura al gruppo <code>'tty'</code>).
n	Impedisce agli altri utenti di scrivere sul proprio terminale (toglie il permesso di scrittura al gruppo <code>'tty'</code>).
	Se l'opzione non viene specificata, si ottiene la visualizzazione dello stato attuale.

Per scrivere sullo schermo di un altro utente collegato allo stesso elaboratore locale, si usano comunemente i programmi `'write'`² e `'wall'`.³

```
write utente [terminale] [< file_messaggio]
```

Il programma `'write'` rappresenta il sistema primordiale per inviare un messaggio a un altro utente che utilizza un terminale dello stesso sistema locale. Il messaggio viene atteso dallo standard input e viene scritto nel dispositivo dell'utente destinatario quando questo viene concluso con un codice di EOF (che di solito si ottiene con la combinazione [*Ctrl d*]).

Dal momento che il programma `'write'` non è destinato all'invio di messaggi attraverso la rete, il nome dell'utente va indicato in modo semplice, senza specificare il nodo. Il dispositivo del terminale può essere specificato e in tal caso si può indicare il percorso assoluto (`'/dev/tty*'`) oppure solo il nome finale. Se il terminale non viene precisato, `'write'` cerca di determinarlo da solo.

```
wall messaggio
```

```
wall [< file_messaggio]
```

Il programma `'wall'` è una variante di `'write'`, dove il messaggio viene inviato a tutti i terminali attivi. Il messaggio può essere fornito anche attraverso la riga di comando.

Per poter scrivere sul dispositivo dell'utente destinatario, secondo le convenzioni, `'write'` e `'wall'`, devono avere i privilegi del gruppo `'tty'`, per cui viene installato comunemente con il bit SGID attivato, appartenendo al gruppo `'tty'`.

```
# chown root:tty /usr/bin/write [Invio]
# chmod g+s /usr/bin/write [Invio]
# chown root:tty /usr/bin/wall [Invio]
# chmod g+s /usr/bin/wall [Invio]
```

Dal momento che quando si invia un messaggio, si presume che il proprio corrispondente voglia rispondere, `'write'` e `'wall'` non inviano il messaggio se il proprio terminale non ammette la risposta, cioè se i permessi del proprio file di dispositivo non lo consentono.

37.1.2 Comunicazione diretta attraverso la rete

Per entrare in comunicazione diretta con un utente che sta utilizzando un terminale o una console di un certo nodo raggiungibile attraverso la rete, si può utilizzare il servizio `'talk'` gestito attraverso il demone `'talkd'`⁴.

In tal caso, è il demone `'talkd'` (o meglio, `'in.talkd'`) del nodo destinatario, a occuparsi di scrivere sul dispositivo del terminale. Generalmente, questo programma viene avviato dal supervisore dei servizi di rete con i privilegi dell'utente `'root'`, cosa che gli permetterebbe di scavalcare qualunque limitazione di accesso ai dispositivi di terminale. Tuttavia, è il demone stesso che cerca di rispettare le convenzioni, evitando di scrivere se manca il permesso di scrittura per il gruppo `'tty'`.

```
in.talkd
```

Il demone `'in.talkd'` è gestito dal supervisore dei servizi di rete e controllato attraverso il filtro del TCP wrapper. Nell'esempio seguente, viene mostrata la riga di `'/etc/inetd.conf'` in cui si dichiara il suo possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
...
talk dgram udp wait root /usr/sbin/tcpd in.talkd
...
```

Dal lato cliente, il programma `'talk'` permette di entrare in comunicazione con una persona che sta utilizzando un nodo all'interno della rete:

```
talk utente [@nodo] [terminale]
```

Il nome dell'utente può essere espresso identificando anche il nodo all'interno del quale è, o dovrebbe essere connesso: `utente@nodo`. Se l'utente con cui si vuole comunicare è connesso su più terminali all'interno dello stesso nodo, è possibile specificare il nome del terminale nella forma `'ttyxx'`. Quando si è chiamati attraverso `'talk'`, sullo schermo del terminale appare un messaggio simile a quello seguente:

```
Message from Talk_Daemon@localhost at 11:31 ...
talk: connection requested by tizio@dinkel.brot.dg.
talk: respond with: talk tizio@dinkel.brot.dg
```

In questo caso si tratta dell'utente `'tizio'` che cerca di contattarci; nel messaggio viene suggerito anche il modo corretto di rispondere. Evidentemente, l'utente che vuole rispondere deve sospendere la propria attività, per avviare a sua volta una copia del programma `'talk'`.

Quando la comunicazione si instaura, viene utilizzato uno schermo suddiviso in due finestre per distinguere i messaggi: nella parte superiore si vedono quelli inviati, mentre nella parte inferiore appaiono quelli ricevuti.

Figura 37.5. Comunicazione attraverso `'talk'`.

```
[Connection established]
Io sto bene, grazie

|-----|

Ciao caio, come stai?
```



Durante la comunicazione, lo schermo può essere riscritto utilizzando la combinazione [Ctrl I]. La comunicazione può essere terminata da uno qualunque dei due interlocutori utilizzando il carattere di interruzione che di norma è [Ctrl c]. Segue la descrizione di alcuni esempi.

```
• $ talk tizio [Invio]
```

Cerca di contattare l'utente 'tizio' nello stesso sistema locale.

```
• $ talk tizio@dinkel.brot.dg [Invio]
```

Cerca di contattare l'utente 'tizio' presso *dinkel.brot.dg*.

```
• $ talk tizio@dinkel.brot.dg tty2 [Invio]
```

Cerca di contattare l'utente 'tizio' presso *dinkel.brot.dg*, al terminale 'tty2' (si tratta probabilmente della seconda console virtuale).

Oltre al programma 'talk' tradizionale, è disponibile comunemente anche 'ytalk'⁵ che consente la comunicazione tra più di due soli utenti:

```
ytalk [-x] utente...
```

Il suo funzionamento è simile a 'talk' e può anche comunicare con utenti che usano lo stesso 'talk'. L'utente può essere specificato in diversi modi:

<i>nome</i>	un utente connesso presso lo stesso elaboratore locale;
<i>nome@nodo</i>	un utente connesso presso un altro elaboratore;
<i>nome#terminale</i>	un utente connesso presso lo stesso elaboratore locale attraverso un terminale determinato;
<i>nome#terminale@nodo</i>	un utente connesso presso un altro elaboratore, su un terminale determinato.

Durante la comunicazione, è possibile richiamare un menù di funzioni premendo il tasto [Esc].

Il programma 'ytalk' è più complesso rispetto al solito 'talk', tanto che è previsto l'uso di file di configurazione: '/etc/ytalkrc' per le impostazioni generali e '~/.ytalkrc' per la personalizzazione da parte di ogni utente. Eventualmente si possono approfondire le altre caratteristiche consultando la sua pagina di manuale: *ytalk(1)*.

37.1.3 Invio di un messaggio circolare

Se quello che si desidera è l'invio di un messaggio circolare senza la necessità di avere un colloquio con gli utenti destinatari, si può usare Rwall.⁶ Il sistema si basa sulle RPC, di conseguenza, è necessario che i nodi destinatari di questo messaggio abbiano in funzione il Portmapper, oltre al demone particolare che si occupa di questo.

Rwall si compone in particolare di un demone, 'rpc.rwalld', oppure solo 'rwalld', il quale si avvia normalmente senza argomenti, di solito attraverso la procedura di inizializzazione del sistema, in modo indipendente dal supervisore dei servizi di rete.

Il programma cliente che serve per sfruttare il servizio è 'rwall', il quale si utilizza con la sintassi seguente:

```
rwall nodo_remoto [file]
```

Il programma 'rwall' consente di inviare un messaggio, eventualmente già preparato in un file, a tutti gli utenti di un nodo remoto determinato. Se non viene fornito il nome di un file contenente il messaggio da inviare, questo messaggio può essere inserito attraverso la tastiera del terminale da cui si avvia il programma. Per termi-

nare l'inserimento si utilizza il codice di EOF che di solito si ottiene premendo la combinazione [Ctrl d].

37.2 IRC

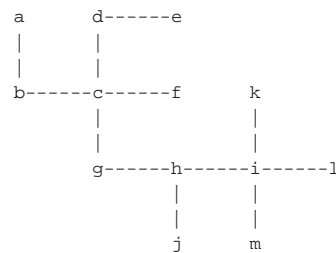
IRC è un sistema di comunicazione in tempo reale per discussioni pubbliche, o private, in forma scritta. Di per sé, IRC è l'evoluzione della comunicazione attraverso 'talk' (sezione 37.1).

Lo scopo di IRC, ovvero la realizzazione di un sistema di discussione pubblica a livello globale, richiede un'infrastruttura composta dai server IRC articolati in modo da formare una «rete» IRC.

Ragionando in piccolo, si può pensare alla realizzazione di un server IRC singolo, presso il quale si devono connettere tutte le persone che vogliono instaurare una forma di discussione qualunque. La distanza non è necessariamente un problema per chi si connette; tuttavia, diventa un problema la quantità di connessioni che verrebbero a essere aperte in modo simultaneo. Nella realtà, queste connessioni possono essere molto numerose (diverse migliaia), soprattutto a causa della filosofia di IRC per la quale l'organizzazione dei canali di discussione è libera, rendendo indispensabile la presenza di un'infrastruttura che sia in grado di recepire tale massa di utenze.

Si parla di reti IRC, a indicare i gruppi di elaboratori che gestiscono assieme gli stessi canali di comunicazione. Tali reti sono composte secondo una struttura ad albero, dove esiste un solo percorso possibile tra due nodi. Naturalmente, queste reti IRC si inseriscono praticamente sulla rete Internet, sfruttando il protocollo TCP per il transito delle informazioni.

Figura 37.7. Rete di server IRC.



L'organizzazione della rete IRC è importante per fare in modo che transitino al suo interno solo le informazioni che sono indispensabili, dal momento che il volume di messaggi gestiti è enorme.

A livello di rete IRC si può individuare una persona con un ruolo speciale: l'operatore IRC. L'operatore IRC è l'amministratore di uno o più server IRC, nel senso che può impartire a questi dei comandi speciali, relativi al loro funzionamento.

37.2.1 Canali, utenti e operatori

In una rete IRC, le comunicazioni avvengono all'interno di **canali** creati dinamicamente; gli utenti della rete IRC sono individuati in base a un nominativo, definito *nick*. Non esiste una regola nell'uso dei nominativi di identificazione degli utenti e nell'organizzazione dei canali di comunicazione: l'utente che si presenta nella rete IRC chiede di usare un nominativo e lo ottiene se questo non è già utilizzato; l'utente che chiede di accedere a un canale di comunicazione che non esiste, lo crea automaticamente e ne diventa il suo **operatore**.

Naturalmente, un utente che cerca di accedere a una rete IRC lo fa connettendosi a un server IRC di quella rete; ma questo server può definire una sua politica di accessi, per cui l'utente in questione potrebbe anche non essere ammesso ad accedere.

È importante comprendere la filosofia di IRC per ciò che riguarda i canali: questi vengono creati automaticamente nel momento in cui vengono richiesti per la prima volta; quindi scompaiono nel momento in cui non ci sono più utenti collegati al loro interno. È importante

anche chiarire il senso dell'operatore: si tratta dell'utente che crea inizialmente il canale, ovvero dell'utente che riceve questo privilegio da un altro operatore. L'operatore, noto anche con l'abbreviazione di «oper», oppure solo «op», ha la possibilità di stabilire la modalità di funzionamento del canale e può anche allontanare altri utenti dal canale stesso. Segue l'elenco delle modalità più importanti di un canale che sono controllate dall'operatore:

- si può accedere al canale a richiesta, oppure solo a seguito di un invito;
- si può specificare una parola d'ordine per l'accesso al canale;
- si può specificare il numero massimo di accessi, oltre l'operatore;
- si può rendere il canale moderato, per cui in pratica scrive solo l'operatore e gli utenti da lui autorizzati;
- si può bloccare la scrittura nel canale;
- si possono concedere i privilegi di operatore anche a un altro utente;
- si può rendere il canale privato, nel senso che non ne viene pubblicizzata la presenza;
- si può rendere il canale segreto, nel senso che non lo si vuole fare apparire nell'elenco dei canali presenti.⁷

Oltre al controllo sul funzionamento del canale, l'operatore può intervenire in modo privilegiato:

- può specificare il fatto che si tratti di un canale a tema;
- può consentire a un utente di scrivere in un canale moderato;
- può allontanare un utente o gruppi di utenti;
- può concedere un'eccezione nel caso di un canale che richieda l'invito.

Ogni utente, tra le altre cose, ha la possibilità di configurare il proprio accesso al canale in modo da rendersi parzialmente invisibile.

37.2.2 Divisione e ricongiunzione di reti IRC

« Una rete IRC può essere spezzata nel momento in cui un nodo che non è terminale cessa di funzionare per qualche ragione, oppure quando viene dato espressamente questo ordine da un operatore IRC. In questa situazione si formano due reti, in cui continuano a funzionare i canali per quanto possibile. Naturalmente, gli utenti che accedono a una di queste due reti risultano isolati rispetto all'altra rete.

La divisione della rete provoca quindi una crisi temporanea che alla fine si riassetta in qualche modo più o meno automatico. Il vero problema nasce nel momento in cui le reti vengono riunite: i canali con lo stesso nome vengono fusi assieme, riunendo gli utenti. Questa riunione può creare un po' di scompiglio, considerando che la modalità di funzionamento dei canali viene riadattata in modo da armonizzare le eventuali incompatibilità e che gli operatori vengono a sommarsi.

37.2.3 Comportamenti spiacevoli

« IRC è un sistema di comunicazione in cui gli utenti sono presenti simultaneamente nel momento in cui scrivono e leggono i messaggi. Nelle discussioni più o meno pubbliche come queste è comune il fatto che chi non sa stare alle regole di una discussione civile decida invece di esprimersi attraverso il dispetto, con la pretesa di dimostrare così la propria superiorità.

Queste situazioni sono così comuni che ne derivano dei termini standard il cui significato dovrebbe essere conosciuto:

- *bot* è un programma cliente automatico che funziona in modo autonomo (robot), senza un utente che sta comunicando effettivamente;

- *cloner* è un utente che sta utilizzando presumibilmente più programmi clienti, ognuno dei quali è un *clone* in questo contesto;
- *flooder* è colui che inonda in qualche modo un utente allo scopo di allontanarlo dalla comunicazione.

Il *bot*, ovvero il programma che usa IRC da solo, è il mezzo attraverso cui si compiono degli attacchi, altrimenti non ci sarebbe bisogno di un programma automatico, dato che IRC è fatta per comunicare tra esseri umani.

Il fatto di utilizzare diversi programmi clienti, mentre ne basterebbe uno solo per comunicare anche su più canali, può rappresentare l'intenzione di fare qualcosa di più della semplice comunicazione.

37.2.4 Dal lato del servente

« La realizzazione di un servente IRC isolato è un'operazione relativamente semplice, limitando il problema alla definizione di una politica di accessi al servizio. Qui non viene mostrato in che modo organizzare invece una vera rete IRC, che evidentemente è un problema più impegnativo.

Ircd⁸ è il servente IRC tipico dei sistemi Unix. In generale sono essenziali solo due file: l'eseguibile *ircd* e il file di configurazione *ircd.conf*, che in un sistema GNU dovrebbe trovarsi nella directory */etc/ircd/*.

Ircd può essere avviato in modo autonomo, senza l'intervento del supervisore dei servizi di rete, oppure sotto il suo controllo. Nel secondo caso, per quanto riguarda Inetd, si deve provvedere a sistemare il file */etc/inetd.conf* aggiungendo la riga seguente:

```
...
ircd stream tcp wait irc /usr/sbin/ircd ircd -i
...
```

Come si può osservare dall'esempio, conviene avviare l'eseguibile *ircd* usando i privilegi di un utente fittizio definito appositamente per la gestione del servizio IRC; in questo caso si tratta di *irc*. Inoltre, si fa riferimento alla porta TCP attraverso la denominazione *ircd*, la quale, secondo il file */etc/services*, corrisponde normalmente al numero 6667:

```
...
ircd          6667/tcp    # Internet Relay Chat
ircd          6667/udp    # Internet Relay Chat
...
```

Si intende che si tratta di una porta non privilegiata, giustificando la scelta di usare un utente fittizio diverso da *root* per avviare *ircd*.

Il demone *ircd* può essere configurato in modo da gestire autonomamente il protocollo IDENT e altri sistemi di controllo. In questo senso, generalmente non viene inserito il controllo del TCP wrapper.

37.2.4.1 Messaggio del giorno

« Nel momento di una nuova connessione al servizio IRC, il servente mostra il messaggio del giorno. In un sistema GNU/Linux, questo messaggio potrebbe essere contenuto nel file */etc/ircd/ircd.motd* (si tratta di un file di testo normale). In generale è importante predisporre questo file in modo da mostrare le notizie essenziali che si vogliono far conoscere agli utenti IRC, soprattutto per ciò che riguarda le regole di comportamento richieste.

37.2.4.2 Configurazione

« La configurazione può essere molto semplice per la realizzazione di un servente IRC interno, per una rete che non può essere raggiunta dall'esterno, ma ovviamente le cose cambiano nel momento in cui si vuole realizzare una rete IRC. Qui vengono mostrati solo alcuni elementi della configurazione, utili per realizzare un servente singolo, senza problemi di accesso.

Il file di configurazione è un file di testo normale, dove le righe che iniziano con il simbolo '#' sono commenti e le righe vuote o bianche vengono ignorate. Le direttive hanno una forma un po' strana, dove tutto inizia con una lettera che descrive il tipo di informazione che viene fornita dalla direttiva:

```
x : informazione_1 : informazione_2 : ... : informazione_n
```

In generale si dovrebbe disporre di un file di configurazione di partenza commentato adeguatamente, con tutti gli esempi di queste direttive (anche se mostrate solo come commenti). Qui vengono descritte alcune direttive essenziali per la realizzazione di un server IRC locale e isolato.

Una cosa da considerare nel caso il file contenga direttive che devono essere elaborate secondo un ordine preciso è il fatto che il file viene letto in ordine inverso, ovvero vengono lette prima le ultime direttive.

M

```
M : nome_del_servente : * : descrizione : porta : numero_servente
```

Questa direttiva serve a definire il nome a dominio del server, la descrizione del servizio IRC, la porta in cui resta in ascolto il server e il numero di ordine nella rete IRC. Questo ultimo numero è un intero che va da 1 a 64 e va stabilito in base alla gerarchia di una rete IRC; se si tratta dell'unico server, deve essere necessariamente indicato il numero uno, come si vede nell'esempio seguente:

```
M:dinkel.brot.dg:*:Mia IRC:6667:1
```

Nel caso in cui il demone 'ircd' venga utilizzato attraverso il controllo del supervisore dei servizi di rete, potrebbe essere necessario indicare una porta diversa da quella standard, per non interferire proprio con il supervisore stesso che già apre quella porta. Per esempio:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
```

È da considerare il fatto che un demone 'ircd' compilato espressamente per l'utilizzo attraverso il supervisore dei servizi di rete potrebbe non essere in grado di funzionare in modo autonomo, in ogni caso.

A

```
A : riga_1 : riga_2 : ... : riga_n
```

Si tratta della direttiva con cui si definiscono delle informazioni amministrative, elencate con il comando '/admin'. In pratica viene mostrato il contenuto dei campi in righe differenti. Si osservi l'esempio seguente che dovrebbe essere sufficientemente intuitivo:

```
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
```

I

```
I : maschera_ip : parola_d'ordine : maschera_dominio : : classe
```

Questa direttiva stabilisce i limiti di accesso al servizio in base a una maschera IP e a una maschera del nome a dominio; queste maschere si riferiscono ovviamente ai nodi che accedono come clienti. Le maschere in questione si realizzano facilmente utilizzando il simbolo '*' come variabile indefinita. In generale, l'esempio seguente consente qualsiasi accesso:

```
I:*:*:*:1
```

Il campo finale, riferito alla classe, deriva dalla definizione delle classi attraverso le direttive 'Y' che qui non vengono descritte, non essendo indispensabili. In ogni caso, il numero uno rappresenta tutte le classi possibili simultaneamente.

Il campo centrale riservato a una parola d'ordine serve a consentire l'accesso solo attraverso l'indicazione di questa. Tuttavia, a seconda di come è stato compilato il demone 'ircd', questa potrebbe dover essere inserita in modo cifrato. In tal caso dovrebbe anche essere presente un programma apposito per generare tali parole d'ordine cifrate.

K

```
K : maschera_nodo : motivazione : maschera_utente
```

Questa direttiva, non obbligatoria, consente di escludere esplicitamente una combinazione di nodi e di utenti che tentano di accedere da questi nodi. Le maschere in questione si realizzano con l'uso del carattere '*', con cui si rappresenta la solita stringa indefinita. In particolare, il nodo può essere indicato per nome (a dominio) oppure per numero IP. L'esempio seguente esclude gli utenti il cui nome inizia per 'dan' e accedono dalla rete *.brot.dg:

```
K:*.brot.dg:Accesso sospeso per un mese:dan*
```

Per concludere la descrizione della configurazione, l'esempio seguente mostra il caso di una configurazione minima, con le sole direttive indispensabili:

```
M:dinkel.brot.dg:*:Mia IRC:8005:1
A:Mia IRC:Servente IRC:Amministratore <root@dinkel.brot.dg>
I:*:*:*:1
```

37.2.4.3 Avvio del demone

```
ircd [opzioni]...
```

Il demone 'ircd' può funzionare in due modi diversi: legato al supervisore dei servizi di rete, oppure indipendentemente da questo. Nel primo caso si utilizza l'opzione '-i' e nel file '/etc/inetd.conf' non si inserisce il controllo di 'tcpd', perché si creerebbero dei problemi a causa dell'uso del protocollo IDENT:

```
...
ircd stream tcp wait irc /usr/sbin/ircd ircd -i
...
```

Diversamente, il demone può essere avviato come un comando normale, senza nemmeno dover aggiungere la richiesta esplicita di funzionamento sullo sfondo. In effetti, dal momento che si utilizza normalmente una porta TCP non privilegiata, ogni utente comune può, teoricamente, avviare questo tipo di servizio. Segue l'elenco di alcune opzioni della riga di comando di 'ircd'.

Opzione	Descrizione
-t	Fa in modo che il demone funzioni in primo piano, emettendo tutte le sue informazioni diagnostiche attraverso lo standard output.
-xn	Definisce il livello diagnostico richiesto: maggiore è il valore n, maggiore è la quantità di informazioni che si ottengono.
-i	Stabilisce che il demone è sotto il controllo del supervisore dei servizi di rete.
-f file_di_configurazione	Stabilisce espressamente da quale file trarre la configurazione.
-c	Si usa questa opzione quando si avvia il demone attraverso uno script della procedura di inizializzazione del sistema, per cui è necessario che il demone stesso si sganci dallo script e diventi un processo dipendente direttamente da Init.

37.2.5 Dal lato del cliente

Il compito di un programma cliente IRC è quello di consentire la comunicazione effettiva tra l'utente umano e il server IRC. La prima cosa che avviene è la **registrazione**, attraverso la quale l'utente ottie-

ne l'accesso al servizio assieme alla definizione del proprio nominativo. Una volta instaurata la connessione, l'utente ha la possibilità di unirsi a uno o più canali di discussione, creandoli automaticamente se questi non sono già presenti.

Qui si considerano solo due programmi, ircII e Tkirc, dove il secondo è solo un programma frontale che si avvale in pratica del primo per la comunicazione effettiva.

ircII⁹ è il programma cliente standard per comunicare con IRC. Si utilizza attraverso un terminale a caratteri normale, dove lo schermo è diviso in due parti: quella superiore per mostrare i messaggi che scorrono verso l'alto; quella inferiore che è semplicemente la riga da cui si impartiscono i comandi. Il programma eseguibile è 'irc' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del server IRC:

```
$ irc tizio dinkel.brot.dg [Invio]
```

```
*** Welcome to the Internet Relay Network tizio (from dinkel.brot.dg)
*** /etc/irc/script/local V0.5 for Debian finished. Welcome to ircII.
*** If you have not already done so, please read the new user
*** information with
+/HELP NEWUSER
*** Your host is dinkel.brot.dg, running version u2.10.07.0
*** This server was created Fri Dec 17 1999 at 19: 54:56 CST
*** umodes available dioswkg, channel modes available biklmpstv
*** There are 1 users and 0 invisible on 1 servers
*** This server has 1 clients and 0 servers connected
*** Highest connection count: 1 (1 clients)
*** - dinkel.brot.dg Message of the Day -
*** - 16/3/2001 20:44
*** - Benvenuto presso irc.brot.dg
*** -
*** on 1 ca 1(2) ft 10(10)
```

```
[1] 20:45 tizio * type /help for help
```

In questo caso, il messaggio del giorno è soltanto «Benvenuto presso irc.brot.dg», visibile in basso; il resto è stato generato automaticamente dal server. La riga contenente la stringa

```
[1] 20:45 tizio * type /help for help
```

è la linea di demarcazione tra la parte superiore contenente i messaggi e la parte inferiore riservata ai comandi dell'utente. Come si può vedere, viene suggerito l'uso del comando '/help' per richiamare l'elenco dei comandi disponibili.

Se si impartisce il comando '/help', come suggerito, si passa a un contesto differente, in cui si possono ottenere informazioni dettagliate su questo o quel comando:

```
/help [Invio]
```

!	:	abort	admin	alias
assign	away	basics	beep	bind
brick	bye	cd	channel	clear
commands	comment	connect	ctcp	date
dcc	deop	describe	die	digraph
dmsg	dquery	echo	encrypt	etiquette
eval	exec	exit	expressions	flush
foreach	help	history	hook	icb
if	ignore	info	input	intro
invite	ircii	ison	join	kick
kill	lastlog	leave	links	list
load	lusers	me	menus	mload
mode	motd	msg	names	news
newuser	nick	note	notice	notify
on	oper	parsekey	part	ping
query	quit	quote	rbind	redirect
rehash	restart	rules	save	say
send	sendline	server	servlist	set
signoff	sleep	squery	squit	stats
summon	time	timer	topic	trace
type	userhost	users	version	wait
wallops	which	while	who	whois
whowas	window	xecho	xtype	

```
[1] 20:56 daniele * type /help for help
```

Help?

Si può osservare dalla figura che, nella riga di comando, appare un invito che prima non era presente: 'help?', a significare che si può indicare il nome di un comando di quelli elencati per conoscerne la sintassi. Per esempio:

```
Help? help [Invio]
```

```
*** Help on help
Usage: HELP [<command> [<subcommands>]]
Shows help on the given command. The help documentation
is set up in a hierarchical fashion. That means that
certain help topics have sub-topics under them. For
example, doing
HELP ADMIN
gives help on the admin command, while:
HELP SET
gives help on the set command and also displays a list of
sub-topics for SET. To get help on the subtopics, you
would do:
HELP SET <subtopic>
where <subtopic> is one of the subtopics. If you are
using the built in help, then you need only type the
subtopic name. The input prompt will indicate what help
level you are on. Hitting return will move you up one
level.

At any time, you can specify a ? to get a list of
subtopics without the associated help file, for example:
HELP ?
gives a list of all main help topics. The following:
HELP BIND ?
gives the list of all BIND subtopics. If you use a ? with
[1] 21:00 daniele * type /help for help
*** Hit any key for more, 'q' to quit ***
```

Come si vede, se non c'è abbastanza spazio per visualizzare tutto il testo disponibile, basta digitare un carattere qualunque per vedere la pagina successiva, oppure basta inserire la lettera 'q' per terminare.

Alla fine della navigazione nella guida interna, basta premere il tasto [Invio] senza specificare il nome di alcun comando per ritornare alla modalità di funzionamento normale, dove non appare alcun invito.

```
Help? [Invio]
```

I comandi impartiti a ircII sono preceduti dal simbolo '/', per distinguerli dal testo dei messaggi che invece vanno inviati al canale di discussione.

Generalmente, quando ci si trova di fronte all'invito normale, è possibile richiamare i comandi precedenti scorrendo con i tasti [freccia-su] e [freccia-giù].

Si conclude il funzionamento di ircII con il comando '/quit'.

Tkirc¹⁰ è un programma frontale per ircII. Il programma eseguibile è 'tkirc' e si avvia in maniera molto semplice, come nell'esempio seguente, dove viene specificato il nominativo desiderato e l'indirizzo del server IRC:

```
$ tkirc tizio dinkel.brot.dg [Invio]
```

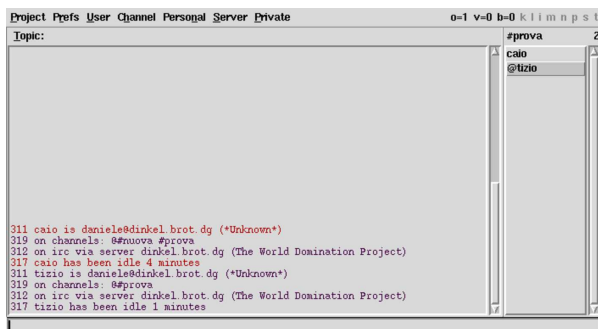
Figura 37.22. Schermata iniziale all'avvio di Tkirc.

```
Project Prefs User Channel Personal Server Private no channel
Topic:
*** Connecting to port 6667 of server localhost
001 Welcome to the Internet Relay Network tizio (from dinkel.brot.dg)
*** /etc/irc/script/local V0.5 for Debian finished. Welcome to ircII.
002 Your host is dinkel.brot.dg, running version u2.10.07.0
003 This server was created Fri Dec 17 1999 at 19: 54:56 CST
004 umodes available dioswkg, channel modes available biklmpstv
251 There are 1 users and 0 invisible on 1 servers
255 This server has 1 clients and 0 servers connected
*** Highest connection count: 1 (1 clients)
375 - dinkel.brot.dg Message of the Day -
372 - 16/3/2001 20:44
372 - Benvenuto presso irc.brot.dg
372 -
*** on 1 ca 1(2) ft 10(10)
*** notification method: 'notify'
```

Utilizzando il menù a tendina, è possibile ottenere un'altra finestra con la quale comunicare in un altro canale. Si utilizza precisamente la voce *New window* dal menù *Project*.

Nella colonna destra, vengono elencati gli utenti che partecipano al canale con cui si sta comunicando. Con un clic doppio del mouse si ottengono le informazioni su di loro, come si vede nella figura 37.23.

Figura 37.23. Informazioni sugli utenti collegati allo stesso canale.



37.2.6 Utilizzo di massima di un cliente IRC

Generalmente, prima di entrare in un canale si può avere l'interesse di visualizzare l'elenco di quelli disponibili. Questo si ottiene con il comando `/list`. Per esempio, con `ircII`:

```
/list [Invio]
```

```
*** Channel  Users  Topic
*** #prova   1
*** #pippo   3
```

Come si vede, il nome di un canale inizia con il carattere '#' per convenzione. In alternativa, il nome di un canale può iniziare anche per '&', ma in tal caso si tratta di un canale che riguarda esclusivamente il server al quale si è connessi, per cui non si diffonde agli altri server della stessa rete IRC.

Nello stesso modo, può essere utile visualizzare l'elenco degli utenti collegati. Questo si ottiene con il comando `/names`, che va usato comunque con parsimonia, considerando che una rete IRC «normale» è sempre molto affollata.

```
/names [Invio]
```

```
Pub: #prova      tizio @daniele
Pub: #pippo      caio @sempronio
```

Nell'elenco degli utenti, gli operatori di canale sono evidenziati dal prefisso '@'. Eventualmente, se si vede il simbolo '*' come prefisso, si tratta di un operatore IRC.

Il programma cliente che si utilizza potrebbe attribuire automaticamente il nominativo per accedere alla rete IRC, sfruttando presumibilmente il nominativo utente usato per accedere al proprio elaboratore. Se il nome in questione non è compatibile, eventualmente perché già utilizzato, è il programma cliente stesso che richiede di indicare un altro nominativo. In ogni caso, è possibile cambiare il proprio nome attraverso il comando `/nick`:

```
/nick pinco [Invio]
```

L'esempio mostra il caso in cui l'utente desidera usare il nome `pinco`, ammesso che questo non sia già utilizzato nella rete IRC in cui si è connessi.

Il nominativo usato all'interno di una rete IRC non può essere più lungo di nove caratteri.

Ci si aggrega a un canale con il comando `/join`. Se il canale indicato non esiste ancora, viene creato per l'occasione e l'utente che lo crea ne diventa l'operatore.

```
/join #prova [Invio]
```

L'esempio mostra il caso in cui ci si voglia aggregare al canale `#prova`. È importante ricordare che è necessario il prefisso davanti al nome, come si vede dall'esempio.

Quando ci si trova in un canale, ciò che si digita senza il prefisso '/', viene trasmesso al canale stesso:

```
ciao a tutti! [Invio]
```

Come ci si unisce a un canale, ci si può allontanare. Questo si ottiene con il comando `/leave`:

```
/leave #prova [Invio]
```

Segue il riepilogo di alcuni comandi essenziali per l'uso di un cliente IRC.

Comando	Descrizione
<code>/list [opzioni]</code>	Elenca i canali presenti nella rete IRC.
<code>/names [opzioni] [canale]</code>	Elenca gli utenti presenti nella rete IRC, oppure solo quelli presenti in un canale particolare.
<code>/nick nome</code>	Consente di modificare, o di stabilire, il proprio nominativo nell'ambito della rete IRC.
<code>/who canale</code>	Consente di elencare gli utenti che sono presenti nel canale indicato.
<code>/whois nome [,nome]...</code>	Consente di elencare le informazioni disponibili sugli utenti elencati. I nomi possono essere anche composti con caratteri jolly, ovvero con l'uso dell'asterisco per indicare una stringa qualunque.
<code>/join canale</code>	Consente di entrare in un canale.
<code>/msg nome messaggio</code>	Consente di inviare un messaggio esclusivamente all'utente indicato.
<code>/dcc chat nome</code>	Invia all'utente indicato una richiesta per instaurare una connessione privilegiata tra i due. Se l'altro utente risponde con lo stesso comando, si ottiene questa connessione. Per comunicare in modo privato, i due usano il comando <code>'msg =nome ...'</code> .
<code>/msg =nome messaggio</code>	Invia un messaggio esclusivamente all'utente indicato, che precedentemente è stato collegato con un comando <code>'dcc chat'</code> .
<code>/quit [messaggio]</code>	Chiude il funzionamento del programma cliente, ma prima si allontana dal canale, se necessario, inviando eventualmente il messaggio indicato.

37.3 ICQ: «I-see-you»

ICQ è un sistema di messaggistica istantanea, originariamente di Mirabilis, che gestisce i server ICQ. Sono disponibili diversi programmi clienti per accedere al servizio, anche nell'ambito del software libero.

Attraverso ICQ un utente si registra presso un server, specificando una parola d'ordine. Il server assegna all'utente un numero, definito UIN, ovvero *Universal Internet number*, e da quel momento si stabilisce l'abbinamento tra UIN e parola d'ordine. Successivamente l'utente può abbinare a questo numero qualche informazione in più su di sé.

L'utente si collega al server ICQ quando desidera annunciare la sua presenza nella rete. Il server ICQ accetta l'utente dopo aver confrontato il numero UIN con la parola d'ordine stabilita originariamente.

Quando un utente ICQ cerca un contatto con un altro utente, può fare una ricerca in base al numero UIN e poche altre informazioni.

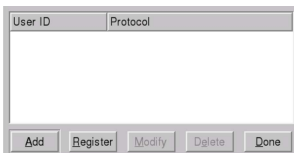
Per ottenere un numero UIN è possibile eseguire una registrazione presso <https://www.icq.com/join>, oppure ci si può affidare alle funzioni del proprio programma cliente; inoltre, una volta ottenuto il numero UIN, si può anche accedere a <http://www.icq.com/>, dove sono disponibili altri servizi.

37.3.1 Licq

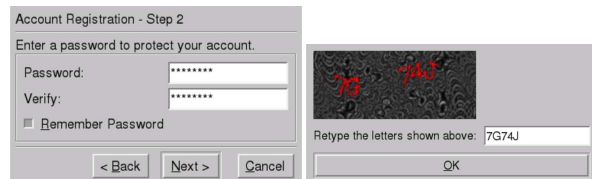
Licq¹¹ è un programma cliente per il servizio ICQ. Si utilizza attraverso l'eseguibile 'licq' che di norma si avvia senza argomenti:

```
licq [opzioni]
```

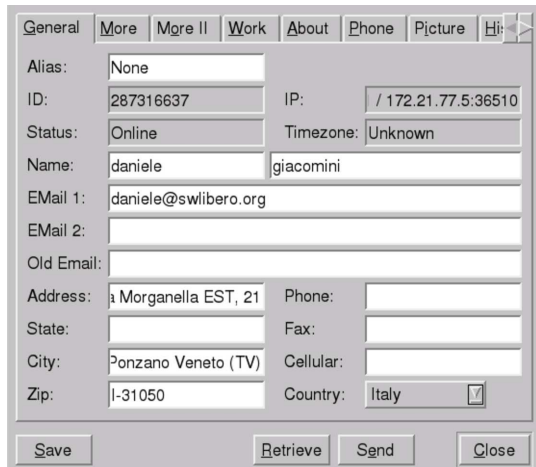
La prima volta che si avvia viene proposta la registrazione presso un server ICQ, in modo da ottenere un numero UIN; eventualmente si ottiene la stessa maschera dalla voce *Owner Manager* del menù *System Functions*:



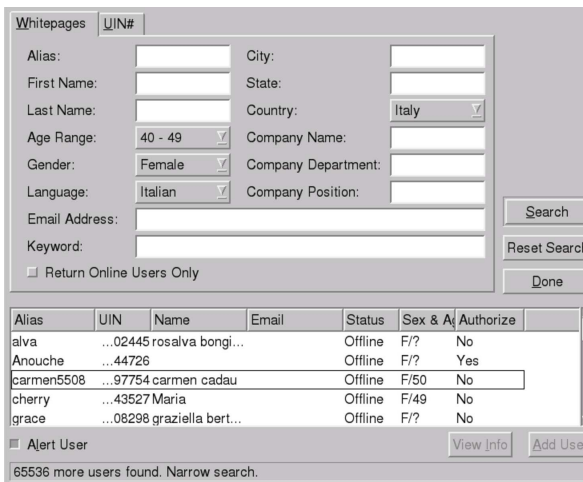
Se si seleziona la richiesta di un nuovo UIN (con il pulsante grafico **REGISTER**), appare successivamente la richiesta di inserimento della parola d'ordine, la quale deve essere al massimo di **otto caratteri** e non può essere più corta di sei:



Se tutto procede come previsto, si ottiene il numero UIN e si può poi continuare compilando le informazioni personali che si intendono rendere pubbliche; eventualmente si ottiene la stessa maschera dalla voce *Info* del menù *System Functions*:



Per cercare una persona, si seleziona la voce *Add User*, dal menù *User Functions*. È possibile specificare direttamente il numero UIN, oppure si può fare una ricerca in base al soprannome, al nome, al cognome, o all'indirizzo di posta elettronica (sempre che questi dati siano stati annotati dalla persona cercata):



Se si ha fortuna, si ottiene un elenco di contatti (non tutti i numeri UIN corrispondono effettivamente a persone reali), dal quale è possibile selezionare chi aggiungere al proprio elenco. Successivamente è possibile tentare di comunicare con questi, oppure è possibile sapere quando sono collegati alla rete anche loro.

Per mostrare la propria presenza attiva nella rete, bisogna selezionare la voce *Online*, dal menù *Status*.

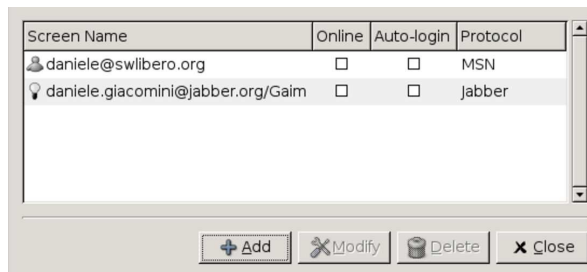
37.3.2 Pidgin

Pidgin,¹² noto originariamente con il nome Gaim, è un cliente generico per diversi sistemi di messaggistica istantanea, tra cui anche ICQ. Si avvia attraverso l'eseguibile 'pidgin' e di norma non si utilizzano opzioni:

```
pidgin [opzioni]
```



Il programma potrebbe presentarsi con diverse finestre; in particolare quella dell'elenco delle utenze configurate, che inizialmente potrebbe essere vuota (in questo caso, invece, appaiono già due utenze: una per il sistema di MSN e l'altra per Jabber). Eventualmente, si ottiene tale elenco selezionando il pulsante **ACCOUNTS**:



Per aggiungere un'utenza di ICQ si procede selezionando il pulsante grafico **ADD**. Si ottiene la maschera che appare nella figura successiva, già compilata per il numero UIN 287316637:

Dopo la conferma, l'elenco delle utenze contiene quella appena inserita:

Screen Name	Online	Auto-login	Protocol
daniele@swlibero.org	<input type="checkbox"/>	<input type="checkbox"/>	MSN
daniele.giacomini@jabber.org/Gaim	<input type="checkbox"/>	<input type="checkbox"/>	Jabber
287316637	<input type="checkbox"/>	<input type="checkbox"/>	AIM/ICQ

Per accedere al servizio di ICQ, basta fare un clic sulla casella *Online*; se nel momento dell'inserimento dell'utenza non è stato richiesto di memorizzare la parola d'ordine, questa va indicata contestualmente, in una mascherina che appare appositamente.

Si osservi che con Pidgin non è possibile registrare una nuova utenza per il sistema di ICQ; quindi, questa va creata, eventualmente, con i programmi specifici per tale protocollo.

37.4 Abbreviazioni di Internet

Spesso, quando si usa la posta elettronica, o altri sistemi di comunicazione testuale, si vedono usare delle sigle, il cui significato a volte sfugge. Storicamente, l'uso di sigle speciali per fare riferimento a concetti ben definiti deriva dalla telegrafia, prima su filo, poi senza filo. Questo ha prodotto il famoso codice «Q» standardizzato attraverso convenzioni internazionali.

La comunicazione odierna non ha più bisogno di abbreviare i messaggi e le abbreviazioni servono solo a creare un gergo che esclude in qualche modo chi non lo conosce. Sotto questo aspetto, non è cortese l'uso di abbreviazioni. Tuttavia, c'è chi non può proprio farne a meno, per cui diventa necessario avere un promemoria per queste cose. La tabella successiva riporta l'elenco delle abbreviazioni più comuni, assieme al loro significato originale (in inglese):

Acronimo	Significato	Acronimo	Significato
AFAICT	As Far As I Can Tell	AFAIK	As Far As I Know
AFK	Away From Keyboard	ASAP	As Soon As Possible
B4	Before	BBL	Be Back Later
BRB	Be Right Back	BTW	By The Way
CUL	See You Later	EOF	End Of File
FAQ	Frequently Asked Question	FOC	Free Of Charge
GA	Go Ahead	HHOJ	Ha Ha, Only Joking
HHOS	Ha Ha, Only Serious	IMBO	In My Bloody Opinion
IME	In My Experience	IMHO	In My Humble Opinion
IMO	In My Opinion	IOW	In Other Words
IRL	In Real Life	ISTM	It Seems To Me
ITRW	In The Real World	JAM	Just A Minute

Acronimo	Significato	Acronimo	Significato
L8R	Later	MUD	Multi User Dungeon
MUG	Multi User Game	OAD	Over And Over
OBTW	Oh, By The Way	OIC	Oh, I See
OMG	Oh My God	OTOH	On The Other Hand
ROFL	Rolls On Floor Laughing	RSN	Real Soon Now
RTFAQ	Read The FAQ	RTFM	Read The Fucking Manual
RUOK	Are You OK	TIA	Thanks In Advance
TNX	Thanks	TTYL	Talk To You Later
TVM	Thanks Very Much	WTH	What The Hell
YHM	You Have Mail		

37.5 Riferimenti

- *Internet Relay Chat (IRC) help*, <http://www.irchelp.org/>
- David Caraballo, Joseph Lo, *The IRC prelude*, <http://www.irchelp.org/irchelp/new2irc.html>
- *ICQ*, <http://www.icq.com>
- *Licq*, <http://licq.sourceforge.net/>
- *Pidgin*, <http://pidgin.im>
- *CenterICQ*, <http://thekonst.net/centericq>
- *aMSN*, <http://sourceforge.net/projects/amsl/>
- *LMME*, <http://sourceforge.net/projects/lmme>
- *Jabber*, <http://www.jabber.org>
- *Cabber*, <http://cabber.sourceforge.net>
- *Yahoo*, <http://www.yahoo.com>

¹ **Sysvinit** GNU GPL

² **Write** UCB BSD

³ **Wall** UCB BSD

⁴ **Talk** UCB BSD

⁵ **ytalk** software libero con licenza speciale

⁶ **Rwall** UCB BSD

⁷ In generale un canale può essere privato, segreto oppure pubblico.

⁸ **Ircd** GNU GPL con residui UCB BSD

⁹ **ircII** software libero con licenza speciale

¹⁰ **Tkirc** GNU GPL

¹¹ **Licq** GNU GPL

¹² **Pidgin** GNU GPL

