

# Posta elettronica



39.1	Servizio di rete e servizio di consegna locale .....	4097
39.2	Uso della posta elettronica .....	4098
39.2.1	Elementi di intestazione .....	4099
39.2.2	Risposta, proseguimento e riservatezza .....	4101
39.3	MTA tradizionale (Sendmail) .....	4103
39.3.1	Alias .....	4104
39.3.2	Coda dei messaggi .....	4106
39.3.3	Rinvio .....	4106
39.4	Recapito della posta elettronica: la variabile «MAIL»	4107
39.5	Mail user agent .....	4108
39.5.1	Ricezione e invio dei messaggi da parte del MUA	4108
39.5.2	Cartelle e formato dei dati .....	4109
39.6	Invio di messaggi attraverso un MTA compatibile con Sendmail .....	4110
39.7	Mailx .....	4113
39.7.1	Avvio e funzionamento .....	4114
39.7.2	Configurazione di Mailx .....	4119
39.7.3	Nail .....	4122
39.8	Mutt .....	4123
39.9	Configurazione compatibile tra Mailx, Nail e Mutt ...	4135
39.10	Ricerche nei file delle cartelle di messaggi .....	4138

39.11	Messaggi giunti presso recapiti remoti .....	4141
39.11.1	IMAP toolkit: ipop3d, ipop2d, imapd .....	4143
39.11.2	Popclient .....	4144
39.11.3	Fetchmail .....	4147
39.11.4	Gestione remota della posta elettronica .....	4154
39.12	Messaggi, allegati ed estensioni MIME .....	4155
39.12.1	Allegati .....	4156
39.12.2	Uuencode .....	4157
39.12.3	Involucro MIME .....	4161
39.12.4	Messaggi contenenti più parti MIME .....	4164
39.12.5	Sistemazione manuale di un allegato MIME ....	4168
39.12.6	Mpack .....	4174
39.13	Gestione della posta elettronica in generale .....	4177
39.13.1	Composizione di un messaggio .....	4179
39.13.2	Messaggi contraffatti e punto di iniezione .....	4184
39.13.3	Identificazione della destinazione .....	4186
39.13.4	Misure di sicurezza .....	4187
39.13.5	Referente per l'amministrazione del servizio ...	4188
39.14	Pratica manuale con i protocolli .....	4188
39.14.1	SMTP attraverso un cliente TELNET .....	4189
39.14.2	POP3 attraverso un cliente TELNET .....	4191
39.14.3	POP3s attraverso un cliente TELNET-SSL .....	4194
39.14.4	Script per l'invio di un messaggio attraverso Telnet	
	4196	

<b>39.15</b>	<b>Procmail</b>	.....	<b>4199</b>
<b>39.15.1</b>	<b>Configurazione di partenza e verifica del funzionamento</b>	.....	<b>4201</b>
<b>39.15.2</b>	<b>Attivazione di Procmail</b>	.....	<b>4203</b>
<b>39.15.3</b>	<b>Esempi semplici di configurazione</b>	.....	<b>4204</b>
<b>39.16</b>	<b>SpamAssassin</b>	.....	<b>4209</b>
<b>39.16.1</b>	<b>Configurazione di SpamAssassin</b>	.....	<b>4209</b>
<b>39.16.2</b>	<b>Cosa fa SpamAssassin</b>	.....	<b>4210</b>
<b>39.16.3</b>	<b>Filtrare i messaggi automaticamente</b>	.....	<b>4215</b>
<b>39.16.4</b>	<b>Autoapprendimento</b>	.....	<b>4216</b>
<b>39.17</b>	<b>Liste di posta elettronica</b>	.....	<b>4217</b>
<b>39.17.1</b>	<b>Lista elementare</b>	.....	<b>4218</b>
<b>39.17.2</b>	<b>Mailman</b>	.....	<b>4220</b>
<b>39.18</b>	<b>Riferimenti</b>	.....	<b>4233</b>
	<b>.fetchmailrc</b>	<b>4147</b>	<b>.forward</b> <b>4106</b>
	<b>.mailrc</b>	<b>4113</b>	<b>4122</b>
	<b>.poprc</b>	<b>4144</b>	<b>.procmailrc</b> <b>4199</b> <b>4215</b>
	<b>aliases</b>	<b>4104</b>	
	<b>fetchmail</b>	<b>4147</b>	<b>grepmail</b> <b>4138</b>
	<b>imapd</b>	<b>4143</b>	<b>ipop2d</b> <b>4143</b>
	<b>ipop3d</b>	<b>4143</b>	<b>mail</b> <b>4113</b>
	<b>mail.rc</b>	<b>4113</b>	<b>mailq</b> <b>4106</b>
	<b>mmsitepass</b>	<b>4221</b>	<b>mm_cfg.py</b> <b>4221</b>
	<b>mpack</b>	<b>4174</b>	<b>mutt</b> <b>4123</b>
	<b>nail</b>	<b>4122</b>	<b>nail.rc</b> <b>4122</b>
	<b>newaliases</b>	<b>4104</b>	<b>newlist</b> <b>4222</b>
	<b>procmail</b>	<b>4199</b>	<b>rmlist</b> <b>4222</b>
	<b>sa-learn</b>	<b>4216</b>	<b>sa-update</b>
	<b>4209</b>	<b>sendmail</b> <b>4103</b>	<b>spamassassin</b> <b>4209</b>
	<b>uuencode</b>	<b>4157</b>	
	<b>\$MAIL</b>	<b>4107</b> <b>4108</b>	

La gestione della posta elettronica è diventato, nel tempo, un problema complesso e impegnativo. In generale non conviene mettere in funzione un proprio server di posta elettronica, se non per lo scopo di comprenderne il funzionamento e le problematiche connesse. Eventualmente, la realizzazione di un servizio di gestione della posta elettronica si può giustificare per assistere un sistema che deve poter spedire messaggi, automaticamente, come un applicativo gestionale dal quale poter spedire le fatture senza complicazioni, ma non per riceverli.

Generalmente, l'invio di messaggi di posta elettronica (*email*) si basa su un MTA (*Mail transfer agent*) locale che, quando riceve una richiesta di invio di un messaggio, si occupa di mettersi in contatto con un suo collega presso l'indirizzo di destinazione, o se necessario in una destinazione intermedia, che si prenda cura di consegnare il messaggio o di inoltrarlo. Tutto quanto sembra molto semplice a dirsi, in realtà la configurazione di un MTA potrebbe essere molto complessa.

Spesso, in presenza di una rete locale, il funzionamento corretto dell'MTA richiede la predisposizione di un servizio di risoluzione dei nomi locale. A tale proposito conviene consultare il capitolo [33](#).

L'invio di messaggi di posta elettronica avviene solitamente attraverso l'uso di un programma adatto alla loro composizione, che poi si mette in comunicazione con l'MTA per l'inoltro del messaggio. Più precisamente, un messaggio inviato a un utente dell'elaboratore locale non richiede alcun MTA, mentre l'invio a un altro elaboratore richiede almeno la presenza di un MTA presso l'indirizzo di

destinazione.

Storicamente, l'MTA più diffuso nei sistemi Unix è stato Sendmail; <sup>1</sup> tuttavia, è sempre più comune l'uso di MTA alternativi, meno complicati, pur mantenendo un certo grado di compatibilità con quello tradizionale.

## 39.1 Servizio di rete e servizio di consegna locale

La posta elettronica non è semplicemente un servizio di rete che si attua attraverso un protocollo (SMTP). Il servizio di rete permette il trasferimento dei messaggi, ma l'MTA ha anche il compito di recapitarli ai destinatari, in forma di file. «

In questo senso, il meccanismo può sembrare un po' confuso all'inizio, trattandosi effettivamente di un sistema piuttosto complicato. In un sistema composto da un elaboratore isolato, anche se provvisto di terminali più o meno decentrati, non c'è alcun bisogno di fare viaggiare messaggi attraverso una rete, è sufficiente che questi vengano semplicemente messi a disposizione dell'utente destinatario in uno o più file. In tal caso, chi si occupa di attuare questo sistema è un MDA, ovvero *Mail delivery agent*.

Quando invece si deve inviare un messaggio attraverso la rete, perché l'indirizzo del destinatario si trova in un nodo differente, si utilizza il protocollo SMTP, per contattare presso la destinazione un server SMTP. Questo server ha il compito di recapitare la posta elettronica (presumibilmente presso il proprio sistema locale). Quindi, lo scopo del server SMTP è quello di recapitare i messaggi.

La trasmissione di un messaggio che richiede la connessione con il server remoto della destinazione, non fa capo ad alcun servi-

zio di rete nell'ambito locale. Questa connessione potrebbe essere instaurata direttamente dal programma che si utilizza per scrivere il messaggio da trasmettere, oppure, come succede di solito, da un altro programma specifico, che in più si preoccupa di ritentare l'invio del messaggio se per qualche motivo le cose non funzionano subito, rinviandolo eventualmente all'origine se non c'è modo di recapitarlo.

L'MTA ha generalmente questi tre ruoli fondamentali: l'attivazione del servizio SMTP, per la ricezione di messaggi dall'esterno; la gestione della trasmissione di questi, assieme a una coda per ciò che non può essere trasmesso immediatamente; la consegna locale dei messaggi ricevuti attraverso il protocollo SMTP oppure attraverso lo stesso sistema locale. Quindi, in generale, un MTA integra anche le funzioni di un MDA.

## 39.2 Uso della posta elettronica

«

La posta elettronica, o *email*, è un modo di comunicare messaggi che richiede la conoscenza di alcune convenzioni. Ciò, sia per evitare malintesi, sia per eliminare le perdite di tempo.

Un messaggio di posta elettronica è formato fondamentalmente da una «busta» e dal suo contenuto. La busta è rappresentata da tutte le informazioni necessarie a recapitare il messaggio, mentre il contenuto è composto generalmente da un testo ASCII puro e semplice. Tutte le volte che il testo è composto in modo diverso, si aggiungono dei requisiti nei programmi da utilizzare per la sua lettura; in pratica, si rischia di creare un problema in più al destinatario del messaggio.

In generale, un messaggio di posta elettronica può contenere uno o più *allegati*, conosciuti frequentemente come *attachment*. L'allegato

permette di incorporare in un messaggio un file che poi, attraverso strumenti opportuni, può essere estrapolato correttamente, riproducendo esattamente il file originale. Ciò che si deve evitare di fare in generale è l'invio del messaggio come un allegato. Questo, purtroppo, capita frequentemente quando si usano programmi per la composizione di messaggi di posta elettronica che permettono di introdurre elementi di composizione del testo (*Rich Text*). Quando si usano programmi grafici di scrittura per i messaggi di posta elettronica è bene controllare la configurazione per disabilitare l'inserimento di codici di composizione.

Le varie estensioni al codice ASCII hanno portato alla definizione di un gran numero di codifiche differenti. Spesso è sufficiente configurare il proprio programma di composizione dei messaggi di posta elettronica in modo da utilizzare la codifica UTF-8, per poter scrivere correttamente con qualunque lingua. Tuttavia, anche la scelta di una codifica come questa, che richiede l'utilizzo di 8 bit invece dei 7 bit tradizionali dell'ASCII, può costituire un problema per qualcuno. In tal senso, quando si scrive in italiano, può essere cortese l'uso di apostrofi alla fine delle vocali che avrebbero dovuto essere accentate.

### 39.2.1 Elementi di intestazione

Un messaggio di posta elettronica si compone inizialmente di una serie di indicazioni, tra cui le più importanti sono quelle che servono a recapitarlo al destinatario. L'uso corretto di questi elementi di intestazione è importante, non solo perché il messaggio raggiunga il destinatario o i destinatari, ma anche per chiarire loro il contesto del messaggio e le persone coinvolte.



Campo	Descrizione
To:	<p>Il campo ‘<b>To:</b>’ viene utilizzato per definire i destinatari del messaggio. Quando si tratta di più di uno, convenzionalmente, i vari indirizzi vengono separati attraverso una virgola.</p> <p>L’indirizzo del destinatario va indicato secondo le regole consentite dagli MTA interessati; generalmente è ammissibile una delle tre forme seguenti:</p> <p><i>utente@nodo</i></p> <p><i>nominativo_completo &lt;utente@nodo&gt;</i></p> <p><i>"nominativo_completo" &lt;utente@nodo&gt;</i></p>
Cc:	<p>Il campo ‘<b>Cc:</b>’ viene utilizzato per definire i destinatari del messaggio in <i>copia carbone</i>. Nella corrispondenza normale, almeno in Italia, si utilizza la definizione «per conoscenza», intendendo che questi destinatari non sono chiamati in causa direttamente.</p> <p>In pratica, si utilizza il campo ‘<b>Cc:</b>’ per recapitare una copia del messaggio a dei corrispondenti dai quali non si attende una risposta, ma che è importante siano a conoscenza di queste informazioni.</p>
Bcc:	<p>Il campo ‘<b>Bcc:</b>’ viene utilizzato per definire i destinatari del messaggio a cui deve essere inviata una copia carbone nascosta (<i>Blind carbon copy</i>). La differenza rispetto alla copia carbone normale sta nel fatto che i destinatari non vengono messi a conoscenza della presenza di queste copie aggiuntive.</p>
From:	<p>Il campo ‘<b>From:</b>’ viene utilizzato per definire l’indirizzo del mittente. Non si tratta necessariamente del nome utilizzato dall’utente nel momento in cui compone il messaggio, ma di quello al quale ci si aspetta di ricevere una risposta.</p>

Campo	Descrizione
Reply-To:	<p>Il campo '<b>Reply-To:</b>' viene utilizzato per indicare un indirizzo a cui si invita a inviare un'eventuale risposta. Viene utilizzato in situazioni particolari, quando per questo non si intende usare il campo '<b>From:</b>'. Tipicamente viene aggiunto nei messaggi trasmessi da un sistema che gestisce le liste di posta elettronica (<i>mailing-list</i>) quando si vuole lasciare l'indicazione del mittente effettivo, guidando la risposta verso la stessa lista.</p>
Subject:	<p>Il campo '<b>Subject:</b>' serve a indicare l'oggetto del messaggio. Anche se nella corrispondenza normale l'oggetto viene usato solo nelle comunicazioni formali, nella posta elettronica è opportuno aggiungere sempre questa indicazione.</p> <p>Un oggetto chiaro permette al destinatario di capire immediatamente il contesto per il quale viene contattato. Inoltre, quando da un messaggio si genera una catena di risposte (cioè un <i>thread</i>), è importante l'oggetto scelto inizialmente.</p>

### 39.2.2 Risposta, proseguimento e riservatezza

La risposta a un messaggio viene inviata normalmente al mittente ('**From:**'), a tutti i destinatari normali ('**To:**') e a tutti quelli cui è stato inviato il messaggio per conoscenza ('**Cc:**'). Questa operazione viene fatta solitamente in modo automatico dal programma utilizzato per leggere e comporre i messaggi: il *Mail user agent* (MUA). È importante però fare attenzione sempre che ciò corrisponda alla propria volontà, o che le circostanze siano appropriate. Infatti, se sono coinvolte diverse persone in una corrispondenza, è probabile che si giunga a un punto in cui non abbia più significato continuare a «im-



portunarle» quando la catena di risposte è degenerata in un contesto differente.

I programmi MUA comuni aggiungono la sigla **‘Re:’** davanti all’oggetto, a meno che non inizi già in questo modo, quando si risponde a un messaggio precedente. Il problema sta nel fatto che non sempre viene rispettata questa convenzione, per cui se ne trovano altri che aggiungono qualcosa di diverso, come **‘R:’**. Così, se la catena di risposte prosegue si rischia di arrivare ad avere un oggetto formato da una serie di **‘Re: R: Re: R:’**.

Quando il messaggio a cui si risponde contiene l’indicazione del campo **‘Reply-To:’**, si pone un problema in più: la scelta corretta del destinatario. Infatti, la risposta va inviata all’indirizzo **‘Reply-To:’** solo se perfettamente conforme al contesto. Si è già accennato al fatto che questo campo viene aggiunto dai programmi di gestione delle liste di posta elettronica. In questa situazione è molto diverso inviare una risposta alla lista o soltanto al mittente originario del messaggio.

Quando si vuole rinviare a un altro indirizzo (*forward* in inglese), si dice che questo viene fatto *proseguire* (così come avviene nella posta normale quando l’indirizzo di destinazione non è più valido per qualche motivo). Per farlo si incorpora il messaggio originale con alcune informazioni sul mittente e sul destinatario originale, permettendo di aggiungere qualche commento aggiuntivo.

Quando si riceve un messaggio, così come accade nella corrispondenza normale, occorre un po’ di attenzione se si pensa di divulgarne il contenuto ad altri. Evidentemente dipende dalle circostanze; in caso di dubbio occorre almeno chiedere il consenso della persona che

lo ha scritto.

### 39.3 MTA tradizionale (Sendmail)

Sendmail costituisce il capostipite degli MTA, tanto che a volte si confonde il suo nome con il concetto di MTA stesso. Sendmail aveva la caratteristica di essere estremamente versatile, ma con una configurazione di eccezionale complessità che era spesso la causa della sua vulnerabilità. Oggi è bene evitare di utilizzare Sendmail come MTA, tuttavia è opportuno conoscere le sue caratteristiche generali, perché altri MTA hanno seguito alcune sue convenzioni.

A seconda delle opzioni con cui viene avviato l'eseguibile '**sendmail**', si ottiene un demone in ascolto della porta SMTP (25), oppure si ottiene la trasmissione di un messaggio fornito attraverso lo standard input, oppure si hanno altre funzioni accessorie.

```
sendmail [opzioni]
```

Per l'attivazione del servizio SMTP, viene avviato normalmente come demone indipendente dal supervisore dei servizi di rete, aggiungendo così l'opzione '**-bd**'. Naturalmente, si tratta solitamente di un'operazione che viene fatta dalla stessa procedura di inizializzazione del sistema. Ecco come potrebbe apparire la riga che avvia '**sendmail**' in uno script del genere:<sup>2</sup>

```
/usr/lib/sendmail -bd
```

Per l'invio di un messaggio, è sufficiente avviare '**sendmail**', fornendogli questo attraverso lo standard input, avendo cura di separare con una riga vuota l'intestazione dal testo. Segue un esempio di questo tipo di utilizzo:.

```
$ cat | /usr/lib/sendmail tizio@dinkel.brot.dg [Invio]
```

**From:** caio@roggen.brot.dg [Invio]

**Subject:** ciao ciao [Invio]

[Invio]

**Ciao Tizio.** [Invio]

**Quanto tempo che non ci si sente!** [Invio]

[Ctrl d]

Questa forma di utilizzo dell'eseguibile **'sendmail'** può essere utile per realizzare uno script con qualche informazione definita in modo automatico.

Per questo tipo di utilizzo, è fondamentale la riga vuota (vuota e non solo bianca) prima del testo del messaggio.

### 39.3.1 Alias

«

Attraverso il file **'/etc/aliases'** è possibile configurare una serie di alias per facilitare l'invio di messaggi di posta elettronica. Gli alias stabiliscono a chi, effettivamente, debbano essere recapitati i messaggi.

In generale, non è conveniente che l'utente **'root'** possa ricevere dei messaggi, per questo, un alias potrebbe rimandare la sua posta elettronica verso il recapito corrispondente all'utente comune riferito a quella stessa persona. Inoltre, è importante che gli utenti fittizi

(**'bin'**, **'daemon'**, ecc.) non possano ricevere messaggi: prima di tutto non esistono tali persone, inoltre ciò potrebbe servire per sfruttare qualche carenza nel sistema di sicurezza dell'elaboratore locale. Infine, è molto importante che vengano definiti degli alias usati comunemente per identificare il responsabile del servizio SMTP presso il nodo locale.

L'esempio seguente mostra il file `/etc/aliases` tipico, in cui si dichiarano gli alias del responsabile del servizio (**'postmaster'**), gli alias degli utenti di sistema e infine l'alias dell'utente **'root'**.

```
postmaster: root
root: tizio

shutdown: root
reboot: root
admin: root
daemon: root
bin: root
sys: root
...
abuse: root
security: root

mailer-daemon: postmaster
```

L'MTA potrebbe non essere in grado di leggere direttamente questo file, richiedendo una sorta di compilazione. Sendmail prevede il comando **'newaliases'** con cui si produce o si aggiorna il file `/etc/aliases.db`. Spesso, gli MTA compatibili con Sendmail dispongono di un comando **'newaliases'** fasullo (privo di effetto), in quanto si servono direttamente dell'elenco testuale di partenza

senza bisogno di trasformarlo.

### 39.3.2 Coda dei messaggi

«

Quando l'MTA viene avviato per ottenere l'invio di un messaggio, questo utilizza normalmente una o più directory collocate da qualche parte sotto `/var/spool/`, per accodare ciò che non può essere trasmesso immediatamente. Per sapere se un messaggio è stato inviato effettivamente si utilizza normalmente il comando `'mailq'`:

```
$ mailq [Invio]
```

```

                Mail Queue (2 requests)
--Q-ID-- --Size-- -Priority- --Q-Time-- --Sender/Recipient--
VAA03244      16      30065 Sep 12 21:01 root
                (Deferred: No route to host)
                                daniele@weizen.mehl.dg
VAA03507      10      30066 Sep 12 21:09 root
                (Deferred: Connection refused by weizen.mehl.dg.)
                                root@weizen.mehl.dg
```

L'uso di `'mailq'` è molto importante per verificare che i messaggi siano stati inviati, specialmente quando si utilizza un collegamento saltuario alla linea esterna.

### 39.3.3 Rinvio

«

Il file `'~/ .forward'` può essere preparato da un utente (nella propria directory personale) per informare il sistema di consegna locale della posta elettronica (MDA) di fare proseguire (rinviare) i messaggi verso altri indirizzi. Il file si compone di una o più righe, ognuna contenente un indirizzo di posta elettronica alternativo; i messaggi giunti per l'utente in questione vengono fatti proseguire verso tutti gli indirizzi elencati in questo file.

```
daniele@dinkel.brot.dg
```

L'esempio mostra semplicemente che tutti messaggi di posta elettronica ricevuti dall'utente a cui appartiene la directory personale in cui si trova il file, devono essere rispediti all'indirizzo *daniele@dinkel.brot.dg*.

È importante chiarire che **non** rimane copia dei messaggi per l'utente in questione. Si presume che questo utente riceva la posta elettronica attraverso uno degli indirizzi elencati nel file `~/ .forward`.

## 39.4 Recapito della posta elettronica: la variabile «MAIL»

La posta elettronica viene recapitata normalmente all'interno di un file di testo unico, appartenente all'utente destinatario. Generalmente, si distinguono due possibilità sulla collocazione di tale file: la directory `/var/mail/` (o anche `/var/spool/mail/`) e un file particolare nella directory personale dell'utente.

Sendmail e altri programmi simili, utilizzano il primo modo, secondo la configurazione predefinita, dove ogni utente ha un proprio file con un nome che corrisponde a quello dell'utenza.

I programmi utilizzati per leggere la posta elettronica devono sapere dove trovarla; in generale si utilizza la convenzione della variabile di ambiente **MAIL**, la quale serve a definire il percorso assoluto del file di destinazione dei messaggi.

Di solito, nel profilo di configurazione della shell appare un'istruzione simile a quella seguente, dove si definisce l'uso di un file, il cui nome corrisponde a quello dell'utente destinatario, nella directo-

ry `‘/var/mail/’` (si fa riferimento a una shell derivata da quella di Bourne).

```
MAIL="/var/mail/$USER"  
export MAIL
```

L'esempio seguente ipotizza invece un recapito presso la directory personale dell'utente:

```
MAIL="$HOME/mail/inbox"  
export MAIL
```

## 39.5 Mail user agent

«

Per scrivere, inviare e leggere i messaggi di posta elettronica si utilizza normalmente un programma apposito, detto MUA o *Mail user agent*. Programmi di questo tipo se ne possono trovare in grande quantità, ma difficilmente questi sono compatibili tra loro.

Il MUA storicamente più importante e quasi sempre presente nei sistemi Unix è Berkeley Mail, ovvero Mailx.

### 39.5.1 Ricezione e invio dei messaggi da parte del MUA

«

La ricezione dei messaggi in un sistema Unix avviene principalmente leggendo il file usato per il recapito di questi nel sistema locale, ovvero il file indicato nella variabile di ambiente *MAIL*. Questo è ciò che si limita a fare un programma come Mailx, mentre altri programmi più sofisticati possono prelevare la posta direttamente da caselle remote attraverso i protocolli POP3 (a volte anche POP2) e IMAP.

Per l'invio dei messaggi, il programma MUA di un sistema Unix ha a disposizione due possibilità. La più semplice è l'utilizzo dell'eseguibile `‘sendmail’` (inteso come MDA locale), a cui viene passato

il messaggio attraverso lo standard input, dove poi è questo secondo programma che provvede da solo al recapito locale o all'invio ad altra destinazione attraverso il protocollo SMTP; la seconda possibilità consiste invece nell'accedere direttamente a un server SMTP.

Tanto per fare un esempio, Mailx è quel tipo di programma che si avvale dell'MDA locale per spedire i messaggi, mentre i programmi più sofisticati si avvalgono direttamente del protocollo SMTP. La differenza tra i due approcci è importante: se non si vuole gestire la posta elettronica localmente, ma si ha una casella di posta remota (come quando si fa un contratto con un ISP), si può fare affidamento esclusivamente su un server SMTP remoto (offerto da quello stesso ISP). Volendo invece utilizzare Mailx, o programmi simili, si è costretti a installare un MTA locale.

### 39.5.2 Cartelle e formato dei dati

Un programma MUA comune consente di organizzare i messaggi ricevuti e le copie di quelli trasmessi all'interno di *cartelle*. Queste cartelle possono essere delle directory contenenti i messaggi sotto forma di file differenti, oppure possono essere dei file singoli, a cui spesso si affiancano altri file contenenti dei riferimenti ai vari messaggi interni.

La forma tradizionale di queste cartelle è quella conosciuta con il nome *mailbox*, corrispondente in pratica a quella del file usato per il recapito dei messaggi locali, come indicato dalla variabile di ambiente *MAIL*. La gestione di cartelle in formato *mailbox* ha lo svantaggio di non offrire un metodo efficace per l'accesso simultaneo da parte di più programmi, tuttavia la corrispondenza è qualcosa di personale e difficilmente si utilizzano due o più programmi simultaneamente.

Nella situazione più semplice, il programma MUA gestisce le cartelle dei messaggi nel formato *mailbox*, in una directory, senza aggiungere altri file (riconoscendo tutti i file della directory come cartelle di messaggi). Eventualmente, alcune cartelle significative possono essere identificate dal programma MUA con un nome particolare, differente dal nome reale del file corrispondente. Per esempio, una di queste cartelle potrebbe chiamarsi «messaggi trasmessi» ed essere abbinata al file `'sentbox'`.

Sono pochi i programmi che ancora oggi si limitano all'uso del formato *mailbox*, senza associare degli indici, riconoscendo come cartelle tutti i file contenuti in una directory stabilita, ma sono solo questi che consentono di usare la posta elettronica sia con Mailx, sia con altri programmi compatibili.

## 39.6 Invio di messaggi attraverso un MTA compatibile con Sendmail

«

È già stato mostrato brevemente come inviare un messaggio molto semplice attraverso l'uso dell'eseguibile `'/usr/lib/sendmail'`, di Sendmail o di un altro MTA che ne conservi la compatibilità. Questa forma di invio dei messaggi diventa molto importante per programmi molto semplici che hanno la necessità di inviare delle informazioni in forma di messaggi di posta elettronica, senza potersi servire di un MUA particolare. Il modello sintattico seguente mostra come strutturare un file contenente un messaggio di posta elettronica da inviare in questo modo:

```

To: [ nominativo_del_destinatario ] <indirizzo_di_posta_elettronica_del_destinatario>
From: [ nominativo_del_mittente ] <indirizzo_di_posta_elettronica_del_mittente>
[ Cc: [ nominativo_destinatario_in_copia ] <indirizzo_destinatario_in_copia> [ , ←
↪ [ nominativo_destinatario_in_copia ] <indirizzo_destinatario_in_copia> ] ... ]
[ Bcc: [ nominativo_destinatario_anonimo ] <indirizzo_destinatario_anonimo> [ , ←
↪ [ nominativo_destinatario_anonimo ] <indirizzo_destinatario_anonimo> ] ... ]
[ altri_campi_particolari ]
...
[ Subject: oggetto ]

testo_del_messaggio
...
...

```

Un file del genere, potrebbe assomigliare all'esempio seguente:

```

To: Tizio <tizio@dinkel.brot.dg>
From: Caio <caio@roggen.brot.dg>
Subject: ciao ciao

Ciao Tizio.
Quanto tempo che non ci si sente!

```

Se questo file viene chiamato 'lettera', lo si può spedire in modo molto semplice così:

```
$ cat lettera | /usr/lib/sendmail -t [Invio]
```

In questo modo, con l'opzione '-t', si ottiene di far leggere l'indirizzo del destinatario dal file stesso.

L'invio di messaggi attraverso questo meccanismo diventa ancora più interessante quando avviene all'interno di uno script di shell. Il modello seguente fa riferimento all'uso di una shell standard per in-

viare all'utente '**root**' un rapporto su quanto svolto da un certo tipo di elaborazione; si può osservare che i comandi che costruiscono il messaggio vengono racchiusi tra parentesi tonde, per poter convogliare il loro flussi standard di uscita in modo complessivo verso `"/usr/lib/sendmail"`:

```
#!/bin/sh
(
echo "To: <root@localhost>"
echo "From: nome_di_comodo <root>"
echo "Subject: oggetto"
echo ""
echo "Il giorno `date` e\` stato eseguito il comando"
echo "comando che ha dato questo responso:"
comando_che_esegue_qualcosa
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Si intuisce che uno script realizzato secondo uno schema simile a quello appena mostrato, potrebbe essere avviato dal sistema Cron per svolgere automaticamente delle funzioni, avvisando convenientemente dell'esito l'amministratore del sistema.

Se non fosse chiaro, ecco come si potrebbe inviare all'amministratore il risultato del comando `'ls -l /'`:

```
#!/bin/sh
(
echo "To: <root@localhost>"
echo "From: ls <root>"
echo "Subject: oggetto"
echo ""
echo "Il giorno `date` e\` stato eseguito il comando"
echo "\"ls -l /\` che ha dato questo responso:"
ls -l /
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Nella sezione [39.14.4](#) viene mostrato come realizzare uno script che si avvale di Telnet per contattare un server SMTP in modo diretto.

## 39.7 Mailx

Mailx<sup>3</sup> è il programma standard di gestione della posta elettronica, originariamente parte dello Unix BSD. Si tratta di un programma piuttosto scomodo da usare, ma rappresenta lo standard ed è quasi indispensabile la sua presenza. «

L'eseguibile `'mail'` prevede due file di configurazione, uno generale per tutto il sistema e uno particolare per ogni utente. Si tratta rispettivamente di `'/etc/mail.rc'` e `'~/mailrc'`.

Nella sua semplicità, Mailx è comunque un programma ricco di opzioni e di comandi per l'utilizzo interattivo. Tuttavia, di solito, è apprezzato solo nelle situazioni di emergenza, per cui è raro che venga sfruttato al massimo delle sue possibilità.

Per l'invio della posta, Mailx utilizza l'eseguibile `'sendmail'`, passandogli le informazioni attraverso la riga di comando e lo standard input. Per la lettura dei messaggi ricevuti, Mailx legge il file

specificato dalla variabile di ambiente *MAIL*; inoltre, generalmente salva i messaggi letti e non cancellati nel file ‘~/mbox’ (nella directory personale dell’utente).

### 39.7.1 Avvio e funzionamento

«

Il programma ‘**mail**’ è l’eseguibile di Mailx. Con la sua semplicità ha il vantaggio di poter utilizzare lo standard input come fonte per un testo da inviare. Di conseguenza, è ottimo per l’utilizzo all’interno di script, anche se per questo si potrebbe richiamare direttamente l’eseguibile ‘**sendmail**’. La sintassi della riga di comando è molto semplice:

```
mail [opzioni] [destinatario...]
```

Segue la descrizione di alcune opzioni.

Opzione	Descrizione
-v	Visualizza un maggior numero di informazioni.
-i	Ignora i segnali di interruzione.
-I	Forza un funzionamento interattivo.
-n	Non legge il file ‘/etc/mail.rc’ quando viene avviato.
-N	Inibisce la visualizzazione delle intestazioni dei messaggi quando viene letta o modificata la cartella della posta.
-s <i>oggetto</i>	Permette di definire l’oggetto già nella riga di comando (se si intendono utilizzare spazi, l’oggetto deve essere racchiuso tra virgolette).

Opzione	Descrizione
-c <i>elenco_destinatari</i>	Permette di definire un elenco di destinatari di una copia del documento (copia carbone). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
-b <i>elenco_destinatari</i>	Permette di definire un elenco di destinatari di una copia carbone che non vengono menzionati nell'intestazione del documento ( <i>blind carbon copy</i> ). L'elenco degli indirizzi di destinazione è fatto utilizzando la virgola come simbolo di separazione.
-f <i>cartella_della_posta</i>	Permette di leggere la posta contenuta all'interno di un file determinato.

Il programma **'mail'**, se avviato allo scopo di leggere la posta, mostra un elenco dei messaggi presenti e attende che gli vengano impartiti dei comandi in modo interattivo. Per questo mostra un invito (*prompt*), formato dal simbolo **'&'**.

Ognuno di questi comandi ha un nome, ma spesso può essere abbreviato alla sola iniziale. L'elenco di questi comandi è molto lungo e può essere letto dalla documentazione interna, *mailx(1)*. Qui viene descritto solo l'utilizzo più comune, con i comandi relativi.

## Invio della posta

Per inviare della posta a una o più persone, è sufficiente avviare **'mail'** utilizzando come argomento gli indirizzi di destinazione delle persone da raggiungere. Per concludere l'inserimento del testo, generalmente è sufficiente inserire un punto ( **'.'** ) all'inizio di una riga nuova, oppure è possibile inviare il codice di

EOF: [*Ctrl d*]. Si osservi l'esempio seguente, in cui si invia un messaggio molto semplice all'indirizzo *tizio@dinkel.brot.dg*:

```
$ mail tizio@dinkel.brot.dg [Invio]
```

```
Subject: Vado in ferie [Invio]
```

```
Ciao Tizio, [Invio]
```

```
ti scrivo solo per avisarti che parto per una settimana [Invio]
```

```
e durante tale periodo non potrò leggere la posta. [Invio]
```

```
A presto, [Invio]
```

```
Caio [Invio]
```

```
. [Invio]
```

```
Cc: [Invio]
```

Durante l'inserimento del messaggio è possibile impartire dei comandi speciali, definiti attraverso delle sequenze di escape, rappresentate da una tilde ('~') seguita dal comando vero e proprio. Attraverso queste sequenze di escape è possibile aggiungere indirizzi ai destinatari in copia carbone, o in copia carbone nascosta, è possibile importare un file, cambiare l'oggetto del messaggio... In particolare, è possibile anche passare alla scrittura del testo attraverso un programma visuale più comodo (come VI o altro, a seconda della configurazione).

## Letture della posta ricevuta

Per controllare la cartella della posta ricevuta e per leggere eventualmente i messaggi, è sufficiente avviare 'mail' senza argomenti. Il programma 'mail' visualizza un elenco numerato delle

descrizioni dell'oggetto di ogni lettera ricevuta. Una volta avviato 'mail', questo presenta il suo invito rappresentato da una e-commerce ('&'), dal quale è possibile dare dei comandi. In particolare, è possibile inserire il numero del messaggio che si vuole leggere. Per leggere il successivo è sufficiente premere il tasto [+], mentre per rileggere quello precedente è sufficiente premere il tasto [-]. Segue un esempio di lettura di un messaggio.

\$ **mail** [Invio]

```
Mail version 8.1.2 01/15/2001.  Type ? for help.
```

```
"/home/tizio/mail/inbox": 6 messages
```

```
>  1 root@dinkel.brot.  Thu Mar 28 22:02 22/845 Debconf: OpenLDAP
   2 caio@dinkel.brot.  Sat Aug 24 09:23 15/484 Vado in ferie
```

& **2** [Invio]

```
Message 2:
```

```
From caio@dinkel.brot.dg Sat Aug 24 09:23:39 2002
```

```
To: tizio@dinkel.brot.dg
```

```
Subject: Vado in ferie
```

```
From: caio@dinkel.brot.dg
```

```
Date: Sat, 24 Aug 2002 09:23:39 +0200
```

```
Ciao Tizio,
```

```
ti scrivo solo per avvisarti che parto per una settimana
e durante tale periodo non potrò leggere la posta.
```

```
A presto,
```

```
Caio
```

& **q** [Invio]

```
Saved 1 message in /home/tizio/mbox
```

```
Held 1 message in /home/tizio/mail/inbox
```

## Gestione della posta ricevuta

Dopo aver letto un messaggio, lo si può cancellare con il comando **'delete'** (**'d'**) o si può rispondere con il comando **'reply'** (**'r'**). La cancellazione della posta non è irreversibile. Di solito si possono recuperare dei messaggi attraverso il comando **'undelete'** (**'u'**); però i messaggi cancellati risultano di fatto invisibili.

Si distinguono due tipi di risposta che fanno riferimento a due comandi simili: **'reply'** (**'r'**) e **'Replay'** (**'R'**). Nel primo caso la risposta viene inviata al mittente e a tutto l'elenco dei destinatari del messaggio di origine, mentre nel secondo la risposta va esclusivamente al mittente del messaggio di origine.

## Gruppi di messaggi

Alcuni comandi di **'mail'** accettano l'indicazione di gruppi di messaggi. Per esempio, **'delete 1 5'** cancella i messaggi numero uno e numero cinque, **'delete 1-5'** cancella i messaggi dal numero uno al numero cinque. L'asterisco (**'\*'**) viene utilizzato per identificare tutti i messaggi, mentre il simbolo **'\$'** rappresenta l'ultimo messaggio. Un caso tipico di utilizzo dell'asterisco come gruppo totale dei messaggi è il seguente: **'top \*'** che permette così di visualizzare le prime righe di tutti i messaggi ricevuti.

## Conclusione dell'elaborazione della posta

Per concludere la sessione di lavoro con **'mail'** è sufficiente utilizzare il comando **'quit'** (**'q'**). Di solito, salvo intervenire nella configurazione, la posta letta (e non segnata per la cancellazione)

viene trasferita nel file `~/mbox`, mentre quella non letta rimane nella cartella originale.

### 39.7.2 Configurazione di Mailx

Si è già accennato al fatto che Mailx utilizzi due file di configurazione: `/etc/mail.rc` per tutto il sistema e `~/mailrc` per le particolarità di ogni utente. Le direttive di questo file sono gli stessi comandi che possono essere impartiti a `mail` durante il suo funzionamento interattivo.

In generale, si utilizzano prevalentemente i comandi `set` e `unset`, i quali permettono l'attivazione o la disattivazione di alcune modalità di funzionamento, consentendo anche la definizione di alcune opzioni che prevedono l'indicazione di un'informazione precisa. Segue la descrizione di alcune modalità di funzionamento controllate dai comandi `set` e `unset`.

Direttiva	Descrizione
<pre>set append unset append</pre>	L'attivazione di questa modalità fa sì che i messaggi salvati nel file <code>~/mbox</code> siano aggiunti in coda, invece che inseriti all'inizio.
<pre>set ask unset ask set asksub unset asksub</pre>	L'attivazione di questa modalità fa sì che sia richiesta l'indicazione dell'oggetto prima di consentire l'inserimento del testo del messaggio.

Direttiva	Descrizione
<pre>set askcc unset askcc</pre>	L'attivazione di questa modalità fa sì che sia richiesta l'indicazione di destinatari aggiuntivi in copia carbone alla fine dell'inserimento del messaggio.
<pre>set askbcc unset askbcc</pre>	L'attivazione di questa modalità fa sì che sia richiesta l'indicazione di destinatari aggiuntivi in copia carbone nascosta (' <b>bcc</b> ') alla fine dell'inserimento del messaggio.
<pre>set dot unset dot</pre>	L'attivazione di questa modalità fa sì che sia consentito l'uso di un punto isolato per terminare l'inserimento di un messaggio.
<pre>set hold unset hold</pre>	L'attivazione di questa modalità fa sì che i messaggi letti nella cartella siano conservati (senza trasferirli in '~/ mbox'), se questi non vengono cancellati esplicitamente.
<pre>set ignoreeof unset ignoreeof</pre>	L'attivazione di questa modalità fa sì che non sia permesso l'uso del codice di EOF ([ <i>Ctrl d</i> ]) per terminare l'inserimento di un messaggio.

Segue la descrizione di altre opzioni.

Direttiva	Descrizione
<pre>set EDITOR=<i>programma</i></pre>	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape '~e'.

Direttiva	Descrizione
set VISUAL= <i>programma</i>	Permette di definire il percorso assoluto del programma che si vuole utilizzare per la modifica del testo di un messaggio, quando viene richiesto espressamente durante il suo inserimento, attraverso la sequenza di escape ' <b>~v</b> '.
set PAGER= <i>programma</i>	Permette di definire il percorso assoluto del programma che si vuole utilizzare per scorrere il contenuto di un messaggio. Perché funzioni correttamente, occorre definire anche l'opzione ' <b>crt</b> '.
set crt= <i>n-righe</i>	Permette di definire il numero di righe di altezza dello schermo, in modo da poter gestire correttamente il programma di impaginazione visuale (' <b>more</b> ' o ' <b>less</b> ').
set MBOX= <i>percorso</i>	Permette di definire il percorso assoluto del file da utilizzare per salvare i messaggi, al posto di ' <b>~/mbox</b> '.
set record= <i>percorso</i>	Permette di definire il percorso assoluto di un file da utilizzare per salvare una copia dei messaggi che vengono inviati.
set folder= <i>percorso</i>	Permette di definire il percorso assoluto di una directory contenenti file corrispondenti a cartelle di messaggi.

Segue la descrizione di alcuni esempi.

```
set append dot save asksub
```

Quello che si vede sopra è il contenuto del file di configurazione generale tipico (il file '**/etc/mail.rc**').

```
set MBOX=/home/tizio/mail/ricevuta
set record=/home/tizio/mail/spedita
set folder=/home/tizio/mail
```

L'esempio si riferisce a un file di configurazione personale, ovvero `~/mailrc`, dove l'utente vuole gestire la sua posta nella directory `~/mail/` (si tratta dell'utente `tizio`), dove possono trovarsi anche altri file intesi come cartelle di messaggi.

```
set MBOX="$HOME/mail/ricevuta"
set record="$HOME/mail/spedita"
set folder="$HOME/mail"
```

Questo esempio produce lo stesso risultato di quello precedente, con la differenza che i percorsi includono la variabile di ambiente *HOME*, la quale si espande nella directory personale dell'utente; in questo modo, tale configurazione potrebbe anche essere generalizzata e inserita nel file `/etc/mail.rc`.

### 39.7.3 Nail

«

Nail<sup>4</sup> è un programma funzionalmente simile a Mailx, ma in più consente l'uso di allegati MIME ed è in grado di servirsi direttamente di un servente SMTP per l'invio dei messaggi.

Anche la configurazione è compatibile con quella di Mailx, tanto che viene utilizzato lo stesso file `~/mailrc` per gli utenti, mentre la configurazione generale è contenuta nel file `/etc/nail.rc` per sicurezza.

## 39.8 Mutt

Mutt<sup>5</sup> è un programma per la gestione della posta per terminali a caratteri, più amichevole rispetto a Mailx e simili. Per l'invio dei messaggi, Mutt utilizza `/usr/sbin/sendmail`, oppure può avvalersi di un programma differente, ma con lo stesso comportamento, purché specificato nella configurazione. In pratica, Mutt non gestisce da solo il protocollo SMTP.

Mutt si compone dell'eseguibile `mutt`, il quale di solito si avvia senza argomenti, e prevede la presenza di diversi file di configurazione; in particolare `/etc/Muttrc` per tutto il sistema e `~/ .muttrc` (o `~/ .mutt/muttrc`) per le particolarità dei singoli utenti.

Una caratteristica molto importante di Mutt è la capacità di gestire formati differenti per le cartelle di posta elettronica. In particolare, il formato predefinito è attualmente il tipo *mailbox*, il quale consente un utilizzo simultaneo ad altri MUA tradizionali.

La configurazione di Mutt prevede direttive di vari tipi; in particolare si distinguono quelle che servono a definire delle «variabili», perché iniziano con la parola chiave `set`. La tabella seguente descrive alcune di queste direttive che vale la pena di conoscere per modificare l'impostazione predefinita della configurazione. Si osservi che Mutt può utilizzare direttamente i protocolli POP3 e IMAP, ma la configurazione relativa non viene mostrata.

Tabella 39.19. Alcune direttive di configurazione di Mutt.

Direttiva	Descrizione
<pre>set mbox_type="mbox   MMDF   MH   Maildir"</pre>	Definisce il tipo di cartelle di posta. Quello tradizionale è indicato attraverso la parola chiave <b>'mbox'</b> .
<pre>set spoolfile="<i>file</i>"</pre>	Definisce il percorso che identifica il file contenente i messaggi di posta in ingresso. In mancanza di questa indicazione, Mutt utilizza il contenuto della variabile di ambiente <b>MAIL</b> .
<pre>set mbox="<i>file</i>"</pre>	Definisce il percorso che identifica il file in cui vanno collocati i messaggi letti. In mancanza di questa indicazione, Mutt utilizza il file <b>'~/mbox'</b> .
<pre>set record="<i>file</i>"</pre>	Definisce il percorso che identifica il file in cui vanno collocati i messaggi inviati.
<pre>set postponed="<i>file</i>"</pre>	Definisce il percorso che identifica il file in cui vanno collocati i messaggi sospesi (da completare o inviare in seguito).

Direttiva	Descrizione
<pre>set folder="<i>directory</i>"</pre>	Definisce il percorso che identifica una directory in cui cercare le cartelle di posta. In mancanza di questa indicazione, Mutt utilizza la directory ‘~/Mail/’.
<pre>set signature="<i>file</i>" set signature="<i>comando</i>  "</pre>	Definisce il percorso che identifica un file il cui contenuto va aggiunto automaticamente in coda ai messaggi da inviare, come «firma». In mancanza di questa indicazione, Mutt utilizza il file ‘~/signature’. Come si vede dal modello sintattico, se il file termina con una barra verticale (‘ ’), si intende trattarsi dello standard output di un comando, da usare per ottenere qualcosa di dinamico.

Direttiva	Descrizione
<pre>set editor="<i>comando</i>"</pre>	Definisce il programma da usare per la creazione e la modifica di file di testo; principalmente per scrivere e modificare i messaggi di posta elettronica da inviare. Se non è indicato, si fa riferimento alle variabili di ambiente <b><i>VISUAL</i></b> , <b><i>EDITOR</i></b> , o in mancanza al programma <code>/usr/bin/editor</code> .
<pre>set attribution="<i>stringa</i>"</pre>	Definisce la stringa da inserire prima di un testo citato. In mancanza di questa indicazione si usa la stringa: <code>'On %d, %n wrote:'</code> . Si possono usare le sequenze descritte in parte nella tabella 39.20.

Direttiva	Descrizione
<pre>set indent_string="stringa"</pre>	Definisce la stringa da usare per evidenziare il testo citato del messaggio a cui si risponde. In mancanza di questa indicazione si usa il simbolo di maggiore seguito da uno spazio: '> '. Si possono usare le sequenze descritte in parte nella tabella 39.20.
<pre>set use_from="yes   no"</pre>	Abilita o disabilita l'inserimento automatico del nominativo utente nel campo 'From:'. Al posto di abilitare questa funzionalità, si può usare la direttiva 'my_hdr' per definire il campo 'From:' in modo preciso.
<pre>my_hdr nome : valore</pre>	Dichiara un campo particolare dell'intestazione, con il valore da assegnare (si usa preferibilmente nella configurazione personalizzata del singolo utente).

Direttiva	Descrizione
<code>my_hdr From: <i>nome_utente</i> &lt;indirizzo&gt;</code>	Dichiara in modo preciso il campo ' <b>From:</b> ' (conviene usare questa dichiarazione soltanto nella configurazione personalizzata del singolo utente).

Tabella 39.20. Alcune sequenze speciali che vengono sostituite da Mutt all'interno delle stringhe.

Macro	Risultato
<code>%a</code>	Indirizzo dell'autore del messaggio.
<code>%d</code>	Data e orario del messaggio dal punto di vista del mittente.
<code>%D</code>	Data e orario del messaggio dal punto di vista locale.
<code>%f</code>	Contenuto del campo ' <b>From:</b> '.
<code>%n</code>	Nome dell'autore, o in mancanza si fa riferimento all'indirizzo di posta elettronica dello stesso.
<code>%s</code>	Oggetto del messaggio.
<code>%t</code>	Contenuto del campo ' <b>To:</b> '.

Avviando l'eseguibile '**mutt**' la prima volta, è probabile che si veda la richiesta di creare la directory da usare per contenere le cartelle di posta; quindi si accede normalmente all'elenco dei messaggi disponibili nella cartella di posta in entrata, come si vede nella figura 39.21.

Figura 39.21. Aspetto di Mutt all'avvio.

```

q:Quit  d:Del  u:Undel  s:Save  m:Mail  r:Reply  g:Group  ?:Help
 1      Apr 26 Fulvio Ferroni  ( 31) Re: Nano OK
 2      Apr 27 Tizio Tizi     (  4) Bla bla bla

```

```

---Mutt: ~/mail/mbox [Msgs:2 Post:2 3.4K]-(threads/date)--(all)-

```

Il funzionamento di Mutt dipende dalla localizzazione, pertanto alcune risposte da dare alle domande che vengono proposte richiedono lettere differenti a seconda di questa. La figura mostra in particolare il funzionamento per le convenzioni della lingua inglese, dove si vede la presenza di due soli messaggi.

Quando Mutt si trova in una condizione del genere, ovvero quando mostra l'elenco di messaggi contenuto in una certa cartella (la figura mostra la cartella corrispondente al file `~/mail/mbox`), si dice che è in modalità «indice». Durante questa modalità di funzionamento, possono essere impartiti dei comandi, costituiti generalmente da lettere singole, una piccola parte dei quali viene riassunta sulla prima riga dello schermo. La tabella 39.22 descrive brevemente parte dei comandi validi quando appare un elenco di messaggi. Si osservi che la maggior parte dei comandi richiede poi una conferma o l'indicazione di altri dati, attraverso messaggi che appaiono nell'ultima riga dello schermo.

Tabella 39.22. Alcuni comandi validi quando si sta scorrendo un elenco di messaggi.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[ <i>m</i> ]	<i>mail</i>	Richiede di scrivere un messaggio di posta elettronica. Se sono disponibili messaggi rimasti in sospeso, viene richiesto se si intendono riprendere.
[ <i>r</i> ]	<i>reply</i>	Risponde al mittente del messaggio evidenziato.
[ <i>b</i> ]	<i>bounce</i>	Invia una copia del messaggio a un altro indirizzo.
[ <i>f</i> ]	<i>forward</i>	Rinvia una copia del messaggio a un altro indirizzo.
[ <i>g</i> ]	<i>group</i>	Risponde al mittente e a tutti i destinatari del messaggio evidenziato.
[ <i>L</i> ]	<i>list</i>	Risponde all'indirizzo che sembra appartenere a una lista di posta elettronica, indicato nel messaggio evidenziato.
[ <i>c</i> ]	<i>change</i>	Passa a un'altra cartella di messaggi. Viene richiesto di indicare il nome della cartella, oppure è possibile selezionarla da un elenco.
[ <i>Esc</i> ][ <i>c</i> ]	<i>change</i>	Passa a un'altra cartella di messaggi, ma in sola lettura.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[C]	<i>copy</i>	Copia il messaggio corrente in un'altra cartella di posta.
[d]	<i>delete</i>	Cancella il messaggio corrente.
[u]	<i>undelete</i>	Toglie la richiesta di cancellazione al messaggio corrente.
[o]	<i>order</i>	Cambia il metodo di riordino dei messaggi.
[O]	<i>order</i>	Inverte l'ordine dei messaggi (in base al tipo di ordinamento attuale).
[q]	<i>quit</i>	Salva le modifiche e conclude il funzionamento di Mutt.
[x]	<i>exit</i>	Annulla le modifiche e termina il funzionamento.
[Invio]		Visualizza il messaggio selezionato.
[v]	<i>view</i>	Visualizza gli allegati.
[/]		Cerca una stringa (da inserire subito dopo), tra i dati che si vedono nell'elenco.
[p]	<i>print</i>	Stampa il messaggio selezionato.
[Ctrl l]		Ridisegna lo schermo.

Come si vede dalla tabella 39.22, per inviare un messaggio si comincia dal premere il tasto [m] (*mail*); viene richiesto di inserire l'indirizzo di destinazione e l'oggetto, quindi si passa all'inserimento del testo del messaggio, attraverso un programma per la modifica

di file di testo. Al termine della stesura del testo, lo si deve salvare e quindi è necessario uscire da quel programma, per ritornare sotto il controllo di Mutt, il quale potrebbe mostrare una schermata simile a quella seguente:

Figura 39.23. Aspetto di Mutt dopo l'inserimento di un messaggio e prima del suo invio.

```

y:Send q:Abort t:To c:CC s:Subj a:Attach file d:Descrip ?:Help
  From:
    To: daniele@dinkel.brot.dg
    Cc:
    Bcc:
  Subject: ciao
Reply-To:
  Fcc: ~/mail/sentbox
  Mix: <no chain defined>
Security: Clear

-- Attachments
- I 1 /tmp/mutt-dinkel-3562-24 [text/plain, 8bit, 0.1K]

-- Mutt: Compose [Approx. msg size: 0.1K Atts: 1]-----

```

Come si può vedere, non appare più il corpo del messaggio, che invece viene indicato come allegato. Per tornare alla modifica del messaggio basta premere la lettera [*e*] (*edit*), per spedire il messaggio si usa la lettera [*y*], mentre per completare altri campi dell'intestazione si usano comandi simili. La tabella 39.24 riepiloga i comandi più importanti, validi in questo contesto.

Tabella 39.24. Alcuni comandi validi quando si sta componendo un messaggio di posta elettronica.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[ <i>e</i> ]	<i>edit</i>	Torna alla modifica del messaggio.
[ <i>q</i> ]	<i>quit</i>	Annulla il messaggio e torna alla situazione precedente all'inserimento, con la possibilità di mantenere in sospeso il messaggio.
[ <i>t</i> ]	<i>to</i>	Modifica il destinatario.
[ <i>Esc</i> ][ <i>f</i> ]	<i>from</i>	Modifica il campo ' <b>From:</b> '.
[ <i>c</i> ]	<i>cc</i>	Inserisce o modifica il campo ' <b>Cc:</b> '.
[ <i>b</i> ]	<i>bcc</i>	Inserisce o modifica il campo ' <b>Bcc:</b> '.
[ <i>f</i> ]	<i>fcc</i>	Inserisce o modifica il campo ' <b>Fcc:</b> ', ovvero l'indicazione del file in cui salvare il messaggio, una volta spedito.
[ <i>s</i> ]	<i>subject</i>	Inserisce o modifica l'oggetto.
[ <i>r</i> ]	<i>reply-to</i>	Inserisce o modifica il campo ' <b>Reply-To:</b> '.
[ <i>a</i> ]	<i>append</i>	Allega un file al messaggio.
[ <i>D</i> ]	<i>delete</i>	Elimina l'allegato o il messaggio selezionato.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[ <i>d</i> ]	<i>description</i>	Modifica la descrizione del messaggio o dell'allegato evidenziato.
[ <i>y</i> ]	<i>yes</i>	Invia il messaggio.
[ <i>P</i> ]	<i>postpone</i>	Sospende il messaggio, conservandolo per un secondo momento.
[ <i>Ctrl l</i> ]		Ridisegna lo schermo.

Da un elenco di messaggi si passa alla visualizzazione di quello selezionato premendo semplicemente [*Invio*]; durante la visualizzazione di un messaggio, è possibile rispondere allo stesso premendo il tasto [*r*], oppure fare altre cose come descritto nella tabella 39.25.

Tabella 39.25. Alcuni comandi validi quando si sta visualizzando un messaggio.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[ <i>q</i> ]	<i>quit</i>	Annulla e torna alla situazione precedente.
[ <i>r</i> ]	<i>reply</i>	Risponde al mittente del messaggio visualizzato.

Tasto, sequenza o combinazione di tasti	Termine mnemonico	Descrizione
[g]	<i>group</i>	Risponde al mittente e a tutti i destinatari del messaggio visualizzato.
[L]	<i>list</i>	Risponde all'indirizzo che sembra appartenere a una lista di posta elettronica, indicato nel messaggio visualizzato.
[b]	<i>bounce</i>	Invia una copia del messaggio a un altro indirizzo.
[f]	<i>forward</i>	Rinvia una copia del messaggio a un altro indirizzo.
[h]	<i>header</i>	Mostra l'intestazione completa del messaggio, o ritorna all'intestazione ridotta.
[p]	<i>print</i>	Stampa il messaggio visualizzato.
[Ctrl l]		Ridisegna lo schermo.

## 39.9 Configurazione compatibile tra Mailx, Nail e Mutt

In questa sezione si vuole mostrare in che modo si possono configurare Mailx, Nail e Mutt, per consentire il loro utilizzo in modo indifferente, sulle stesse cartelle di messaggi.

Per prima cosa si deve decidere in quale directory devono essere contenuti i file, in formato *mailbox*, delle cartelle. Si suppone di usare la directory `~/mail/` per tutti gli utenti del sistema, stabilendo

anche che la posta in ingresso viene consegnata nel file ‘~/mail/inbox’.

In generale, per informare della presenza della cartella dei messaggi in ingresso basta impostare la variabile di ambiente **MAIL**. Per intervenire su tutti gli utenti si può intervenire nel file ‘/etc/profile’ (nel caso di una shell compatibile con quella di Bourne), come in questo esempio:

```
MAIL="$HOME/mail/inbox"
export MAIL
```

Naturalmente, si deve provvedere a configurare anche il sistema di consegna locale dei messaggi, in modo che funzioni così, altrimenti la posta potrebbe risultare inserita in file all’interno della directory ‘/var/mail/’, o ‘/var/spool/mail/’, nonostante tutte le buone intenzioni.

Il passo successivo è la definizione di alcune cartelle, più o meno standard. Per esempio è necessario stabilire la collocazione della posta inviata, di quella che è in coda e di quella che è stata solo abbozzata (iniziata ma non completata). Si potrebbe stabilire questa associazione:

Cartella	File corrispondente
posta in ingresso	‘~/mail/inbox’
posta in uscita o in coda per l’invio	‘~/mail/outbox’
posta spedita	‘~/mail/sentbox’
posta letta	‘~/mail/readbox’
bozze di messaggi da trasmettere	‘~/mail/draftbox’
messaggi in attesa di essere eliminati	‘~/mail/trash’

Non tutti i programmi che si intendono utilizzare richiedono così

tante cartelle, ma almeno sono in grado di accedervi.

Si può stabilire anche l'uso di un file contenente una «firma», ovvero alcune righe da accodare a tutti i messaggi che vengono trasmessi. Per esempio, si può stabilire che debba trattarsi del contenuto del file `~/signature`.

Segue la porzione di configurazione da usare sia per il file `/etc/mail.rc`, sia per `/etc/nail.rc`, in favore di Mailx e di Nail:

```
set append
set folder="$HOME/mail"
set MBOX="$HOME/mail/readbox"
set record="$HOME/mail/sentbox"
```

In questo modo, Mailx e Nail traggono la posta in ingresso dal file `~/mail/inbox`, perché così è annotato nella variabile di ambiente **MAIL**; inoltre i messaggi letti e quelli trasmessi vengono inseriti correttamente nelle cartelle previste. L'accesso alle altre cartelle di messaggi risulta comunque facilitato perché è stata indicata la directory `~/mail/` in modo predefinito.

Nel caso particolare di Nail, si può aggiungere anche l'indicazione del file da usare come firma:

```
set signature="$HOME/.signature"
```

Per quanto riguarda Mutt, si può intervenire nel file `/etc/Muttrc`:

```
set mbox_type="mbox"
set record="~/mail/sentbox"
set spoolfile="~/mail/inbox"
set mbox="~/mail/readbox"
set postponed="~/mail/draftbox"
set folder="~/mail/"
```

Eventualmente, se si vuole evitare che Mutt sposti la posta letta in

modo automatico nella cartella relativa, è sufficiente indicare per questo la stessa cartella dei messaggi in ingresso:

```
set mbox_type="mbox"  
set record="~/mail/sentbox"  
set spoolfile="~/mail/inbox"  
set mbox="~/mail/inbox"  
set postponed="~/mail/draftbox"  
set folder="~/mail/"
```

## 39.10 Ricerche nei file delle cartelle di messaggi

«

I file delle cartelle di posta elettronica in formato *mailbox*, sono file di testo organizzati secondo una certa struttura. All'interno di questi file è possibile eseguire delle ricerche con Grep, ma il vero problema è quello di identificare il messaggio che contiene la stringa o l'espressione cercata. Per questo conviene usare invece Grepmail,<sup>6</sup> ovvero un programma Perl che restituisce il messaggio intero e non soltanto la riga che corrisponde al modello di ricerca.

Grepmail non si limita a questo, consentendo anche una ricerca selettiva nel corpo dei messaggi, nell'oggetto, escludendo eventualmente gli allegati. Il suo utilizzo più semplice è quello rappresentato dall'esempio seguente:

```
$ grepmail "Tizi[oa]" ~/mail/sentbox | less [Invio]
```

In questo caso si cercano tutti i messaggi contenuti nel file '~/mail/sentbox' che corrispondono in qualche modo con l'espressione regolare 'Tizi[oa]'. Con l'ausilio di 'less', si scorrono facilmente sullo schermo.

Trattandosi di un programma scritto in Perl, le espressioni regolari che si possono utilizzare devono avere le caratteristiche di questo linguaggio di programmazione.

```
grepmail [opzioni] [-e] espressione_regolare [file_cartella_messaggi] ...
```

Il modello sintattico mostra due particolarità: l'espressione regolare può essere indicata da sola oppure come argomento dell'opzione **'-e'**; i file delle cartelle dei messaggi possono essere forniti come argomenti finali della riga di comando, ma in loro mancanza, viene letto lo standard input. La tabella 39.32 riepiloga le altre opzioni più importanti.

Tabella 39.32. Opzioni più importanti di Grepmail.

Opzione	Descrizione
-b	Esegue la ricerca esclusivamente nel corpo dei messaggi.
-h	Esegue la ricerca esclusivamente nell'intestazione dei messaggi.
-i	Non distingue tra lettere maiuscole e minuscole.
-l	Emette solo il nome del file contenente i messaggi corrispondenti.
-M	Ignora gli allegati MIME di tipo binario.
-R	Cerca ricorsivamente nelle sottodirectory.
-v	Cerca i messaggi che <b>non</b> corrispondono al modello.

Opzione	Descrizione
<code>-d today   yesterday</code>	Seleziona solo i messaggi di oggi o di ieri.
<code>-d mm / gg / aaaa</code>	Seleziona solo i messaggi di una certa data.
<code>-d { n days ago   n weeks ago }</code>	Seleziona solo i messaggi di <i>n</i> giorni o settimane fa.
<code>-d { before   after ↵ ↶   since } data</code>	I messaggi più vecchi, più recenti, o a partire da una data di riferimento.
<code>-d between data and data</code>	Seleziona solo i messaggi compresi tra due date.
<code>-e espressione_regolare</code>	Dichiara espressamente il modello di ricerca.

Vengono mostrati solo alcuni esempi.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ↵
↶ ~/mail/* | less [Invio]
```

Cerca tutti i messaggi nella directory ‘~/mail/’ che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg*. Il risultato viene fatto scorrere con l’aiuto di ‘less’.

```
$ grepmail -h -i "From: .*pinco@dinkel.brot.dg" ↵
↶ ~/mail/* > pinco [Invio]
```

```
$ grepmail -h -i -v "From: .*pinco@dinkel.brot.dg" ↵
↶ ~/mail/* > altri [Invio]
```

I due comandi servono a estrarre tutti i messaggi provenienti presumibilmente da *pinco@dinkel.brot.dg*, per generare il file ‘pinco’, mettendo tutto il resto in un file denominato ‘altri’.

```
$ grepmail -h -d "since 7 days ago" -i ↵  
↵      -e "From: .*pinco@dinkel.brot.dg" ~/mail/* ↵  
↵      | less [Invio]
```

Cerca tutti i messaggi nella directory ‘~/mail/’ che sono stati inviati presumibilmente da *pinco@dinkel.brot.dg* entro gli ultimi sette giorni. Il risultato viene fatto scorrere con l’aiuto di ‘**less**’.

## 39.11 Messaggi giunti presso recapiti remoti

I messaggi di posta elettronica non vengono sempre recapitati presso l’elaboratore che si utilizza abitualmente. Per trasferire la posta da un recapito a un altro, si usa solitamente il protocollo POP3 (a volte POP2) oppure IMAP. Come si può immaginare, si tratta di un servizio che deve essere gestito da un demone.

Il modo con cui vengono scaricati messaggi e inseriti nel sistema locale ha dei risvolti importanti. Infatti, questi messaggi possono essere scaricati in un file locale, corrispondente di norma alla cartella della posta in ingresso dell’utente, il quale può leggerla attraverso Mailx o un altro programma che sfrutta lo stesso meccanismo. In alternativa, i messaggi potrebbero essere inseriti nel sistema locale attraverso un servizio SMTP.

Dei protocolli principali utilizzati per il prelievo e per l’invio dei messaggi, esistono delle «varianti» che prevedono una comunicazione cifrata. In realtà, si tratta degli stessi protocolli, che però si inseriscono a loro volta nel protocollo SSL, pertanto si utilizzano le sigle POP3s, IMAPs e sSMTP per identificarli. Si veda eventualmente la sezione [44.4](#) a proposito di SSL/TLS.

Ricapitolando, i messaggi di posta elettronica prelevati da un recapito remoto, possono essere:

1. scaricati in un file locale che rappresenta la cartella della posta in ingresso dell'utente per cui si svolge l'operazione;
2. inviati nuovamente attraverso l'MDA locale;
3. inviati nuovamente attraverso un server SMTP locale, o comunque uno più «vicino».

Ognuna delle scelte possibili ha dei vantaggi e degli svantaggi. Il primo tipo di operazione, non richiede la presenza di un server SMTP locale e nemmeno di un MDA, cioè di un *Mail delivery agent*, per la consegna locale dei messaggi. Così si presta perfettamente all'uso presso nodi isolati che possono connettersi a Internet solo saltuariamente. Il secondo tipo di operazione richiede la presenza di un MDA, composto generalmente da un programma in grado di ricevere i messaggi attraverso lo standard input, il quale poi sia in grado di recapitarli localmente o eventualmente di farli proseguire altrove attraverso gli alias e i *forward*. Il vantaggio di questa seconda scelta è che per attuarla potrebbe non essere necessario un servizio SMTP locale. L'ultimo caso richiede invece che localmente sia presente un MTA completo, in grado di ricevere le connessioni SMTP.

I motivi per cui non si riceve la posta direttamente nel nodo locale, possono essere vari: la connessione con l'esterno potrebbe essere discontinua; il sistema remoto presso cui giunge la posta per qualche motivo, potrebbe avere delle politiche che impediscono il proseguimento dei messaggi (il *forward*); il sistema locale potrebbe essere irraggiungibile dall'esterno a causa delle politiche di sicurezza adottate, per cui, la posta elettronica potrebbe non essere trasferita local-

mente, lasciando l'onere a ogni nodo di prelevarsela da un server principale.

Negli ultimi due tipi di trasferimento, il programma che lo fa interviene come se fosse un MTA vero e proprio. In tal senso, potrebbe essere attivato periodicamente attraverso il sistema Cron, a intervalli brevi, oppure come un demone.

Il prelievo della posta remota è un'operazione personale dell'utente che ha l'accesso presso il sistema remoto. Il programma che si usa per accedere a uno di questi servizi che lo permettono, deve identificarsi in qualche modo; di solito si tratta di fornire l'identità dell'utente remoto e la parola d'ordine. Il fatto di lasciare viaggiare la parola d'ordine in chiaro, attraverso la rete, è un problema da non trascurare: finché la connessione è diretta (o quasi, come nel caso di una linea commutata), il problema è minimo; quando la connessione attraversa più nodi, il problema diventa delicato.

Oltre a questo, occorre considerare che le informazioni delicate come le parole d'ordine non possono apparire in una riga di comando, perché sarebbero leggibili semplicemente analizzando l'elenco dei processi attivi. Per questo, quando si vuole automatizzare il processo di recupero della posta remota senza dover ogni volta inserire la parola d'ordine, questa può essere annotata soltanto in un file di configurazione, protetto opportunamente contro ogni accesso da parte di altri utenti.

### 39.11.1 IMAP toolkit: ipop3d, ipop2d, imapd

IMAP toolkit è una raccolta di demoni per i servizi di trasferimento della posta locale verso i clienti che lo richiedono, mostrando le credenziali necessarie. Si tratta precisamente dei programmi

‘**ipop3d**’, ‘**ipop2d**’ e ‘**imapd**’. Permettono rispettivamente di utilizzare i protocolli POP3, POP2 e IMAP. Sono gestiti normalmente dal supervisore dei servizi di rete.<sup>7</sup>

Nell’esempio seguente, vengono mostrate le righe di ‘`/etc/inetd.conf`’ in cui si dichiara il loro possibile utilizzo per quanto riguarda il caso particolare di Inetd:

```
...
pop-2    stream  tcp    nowait  root    /usr/sbin/tcpd  ipop2d
pop-3    stream  tcp    nowait  root    /usr/sbin/tcpd  ipop3d
imap     stream  tcp    nowait  root    /usr/sbin/tcpd  imapd
...
```

In alcune distribuzioni GNU questi tre demoni potrebbero fare parte di un pacchetto unico, mentre in altri casi i pacchetti potrebbero essere distinti in base al servizio particolare che viene offerto.

### 39.11.2 Popclient

«

Popclient<sup>8</sup> è un programma molto semplice che permette di scaricare la posta da un recapito remoto utilizzando il protocollo POP2 o POP3, inserendola in un file che corrisponda alla cartella della posta in ingresso dell’utente nel nodo locale, oppure passandola a un MDA (*Mail delivery agent*) che faccia sostanzialmente la stessa cosa. In questo modo, una volta scaricata, la posta può essere letta con un programma tradizionale come Mailx. È importante sottolineare che per questo scopo, non è necessario che sia attivo un server SMTP locale.<sup>9</sup>

L’eseguibile ‘**popclient**’ va usato secondo la sintassi rappresentata dal modello successivo, considerando che è generalmente opportuno predisporre anche un file di configurazione:

```
popclient [opzioni] [nodo_remoto]
```

Nelle opzioni della riga di comando, si può osservare che non è stata indicata la possibilità di inserire la parola d'ordine: infatti, ciò non è possibile. Per non dover inserire la parola d'ordine ogni volta che si scarica la posta, è necessario predisporre un file di configurazione.

Opzione	Descrizione
-2	Viene utilizzato il protocollo POP2.
-3	Viene utilizzato il protocollo POP3.
-k --keep	Copia i messaggi dal server remoto senza cancellarli da lì.
-s --silent	Non mostra i messaggi di progressione dell'operazione.
-v --verbose	Visualizza attraverso lo standard error tutti i messaggi che intercorrono tra il programma e il server remoto.
-u <i>utente</i> --username <i>utente</i>	Permette di specificare il nome dell'utente così come è registrato nel sistema remoto. Il valore predefinito è il nome dell'utente così come è conosciuto nel sistema locale.

Opzione	Descrizione
-r <i>cartella_remota</i> --remote <i>cartella_remota</i>	Permette di specificare una cartella della posta nel server remoto, diversa da quella predefinita. Dipende dal server remoto se questa cartella alternativa esiste. Questa opzione può essere utilizzata solo con il protocollo POP2.
-o <i>cartella_locale</i> --local <i>cartella_locale</i>	Permette di specificare una cartella della posta locale alternativa. Quando non viene specificata una cartella per la posta ricevuta, si intende quella predefinita dal sistema locale.
-c --stdout	Permette di emettere attraverso lo standard output la posta, invece di utilizzare la cartella della posta.

Popclient può essere configurato in modo personale attraverso il file ‘~/ .poprc’. In tal modo, l’utente può predisporre tutti i dati necessari ad automatizzare la connessione, inclusa la parola d’ordine necessaria per l’identificazione presso il server remoto.

L’esempio seguente riguarda il caso in cui si voglia prelevare la posta dal nodo *weizen.mehl.dg*, utilizzando il protocollo POP3, con un nominativo-utente «tizio» e la parola d’ordine «tazza», depositando i messaggi nel file ‘/home/tizio/mail/inbox’:

```
# .poprc

server  weizen.mehl.dg      \
        proto pop3         \
        user tizio         \
        pass tazza         \
        localfolder /home/tizio/mail/inbox
```

Si può leggere eventualmente la pagina di manuale *popclient(1)*.

### 39.11.3 Fetchmail

Fetchmail<sup>10</sup> è un sistema di recupero della posta remota molto complesso. Permette di inserire i messaggi ottenuti nel sistema di consegna locale attraverso un MDA come Sendmail o equivalente; oppure può utilizzare direttamente il protocollo SMTP per ottenere lo stesso risultato, o per inserire i messaggi in un sistema di trasporto più vicino (quale quello di una rete locale).

Può funzionare anche come demone personale (di un utente) in modo da provvedere regolarmente allo scarico dei messaggi.

Fetchmail ha il vantaggio di poter utilizzare una grande varietà di protocolli fatti per questo scopo. In linea di massima ci si può concentrare sui soliti POP2, POP3 e IMAP, ma è bene tenere presente che le possibilità sono maggiori, nel caso si presentasse l'occasione.

L'eseguibile '**fetchmail**' può essere gestito molto bene attraverso la riga di comando, ma è consigliabile anche la sua configurazione attraverso il file '`~/fetchmailrc`', il quale permette di agevolare le operazioni di routine.

```
fetchmail [opzioni] nodo_remoto
```

Se si pone un conflitto tra quanto specificato tramite le opzioni della riga di comando e le direttive del file di configurazione, le prime prevalgono.

Opzione	Descrizione
-a --all	Scarica tutti i messaggi, compresi quelli che risultano già visti.
-k --keep	Non cancella i messaggi che vengono scaricati.
-u <i>utente_remoto</i> --username <i>utente_remoto</i>	Specifica precisamente il nome da utilizzare per accedere al server remoto. Se non viene indicata questa informazione (attraverso la riga di comando, oppure attraverso la configurazione), si intende lo stesso nome utilizzato nel sistema locale.
-t <i>n_secondi</i> --timeout <i>n_secondi</i>	Permette di stabilire un tempo massimo per la connessione, oltre il quale Fetchmail deve abbandonare il tentativo.

Opzione	Descrizione
<p><code>-d <i>n_secondi</i></code></p> <p><code>--daemon <i>n_secondi</i></code></p>	<p>Avvia Fetchmail in modalità demone, cioè sullo sfondo, allo scopo di eseguire la scansione dei server in modo regolare. L'argomento esprime la durata dell'intervallo tra una scansione e l'altra, espresso in secondi.</p> <p>Ogni utente può avviare una sola copia dell'eseguibile '<b>fetchmail</b>' in modalità demone; tuttavia, se si tenta di avviare una nuova copia di '<b>fetchmail</b>', quando è già attivo il demone, ciò fa sì che venga eseguita immediatamente una nuova scansione.</p>

Il file di configurazione di Fetchmail è molto importante. È interessante notare che non esiste un file di configurazione generale, ma solo quelli dei singoli utenti; infatti, il recupero della posta elettronica è un'operazione personale.

Per motivi di sicurezza, dal momento che può contenere informazioni delicate, è necessario che il file di configurazione abbia esclusivamente i permessi di lettura e scrittura per l'utente proprietario (0600<sub>8</sub>). Se il file ha permessi maggiori, Fetchmail avverte e si rifiuta di proseguire.

Prima di analizzare la sintassi che può essere utilizzata al suo interno, si può notare che i commenti vengono espressi nel modo consueto, attraverso il simbolo '#' che li introduce, dove poi tutto quello

che segue, fino alla fine della riga, viene ignorato. Così anche le righe bianche e quelle vuote vengono ignorate.

Ogni direttiva del file `~/ .fetchmailrc` contiene tutte le specifiche riferite al recupero della posta elettronica da un server determinato. Queste direttive possono impiegare più righe, senza la necessità di indicare simboli di continuazione, distinguendosi perché iniziano con la parola chiave `'poll'`, oppure `'skip'`.

Una direttiva `'poll'` rappresenta un server da interpellare, mentre una direttiva `'skip'`, uno da saltare. Di fatto non serve una direttiva `'skip'`, ma può essere utile per evitare di cancellarla, riservando per il futuro la possibilità di riutilizzarla rimettendo la parola chiave `'poll'`.

Le direttive sono composte da una serie di parole chiave che rappresentano delle opzioni, a volte accompagnate da un argomento. Alcune parole chiave sono speciali, in quanto, pur non avendo alcun significato, sono utili per facilitare la lettura delle direttive. Tali parole sono: `'and'`, `'with'`, `'has'`, `'wants'` e `'options'`. Nello stesso modo, possono essere usati la virgola, il punto e virgola, i due punti, i quali vengono ignorati ugualmente.

All'interno di ogni direttiva, deve essere rispettato un certo ordine nell'indicazione delle opzioni. Se ne distinguono due tipi: opzioni del server e opzioni dell'utente. Le opzioni del server devono apparire prima di quelle dell'utente.

Per comprendere il senso di queste direttive, è bene fare mente locale al formato generale che queste possono avere:

```
poll servente [protocol protocollo] [username utente_remoto] ↵
↵ [password parola_d'ordine]
```

Gli argomenti delle opzioni che rappresentano delle stringhe, possono essere racchiusi tra apici doppi, in modo da poter contenere simboli particolari, come gli spazi (specialmente quando si tratta di indicare le parole d'ordine).

### Opzioni del servente

Opzione	Descrizione
poll <i>servente</i> skip <i>servente</i>	Specifica l'accesso a un servente. Se si usa la parola chiave ' <b>skip</b> ', tutta la direttiva viene ignorata.
proto <i>protocollo</i> protocol <i>protocollo</i>	Il tipo di protocollo da utilizzare, viene determinato normalmente in modo automatico. Con questa opzione può essere specificato espressamente, indicando una parola chiave determinata: ' <b>POP2</b> ', ' <b>POP3</b> ', ' <b>IMAP</b> ', ' <b>IMAP-K4</b> ', ' <b>IMAP-GSS</b> ', ' <b>APOP</b> ', ' <b>KPOP</b> '. Si noti che queste parole chiave possono essere espresse anche utilizzando solo lettere minuscole.

Opzione	Descrizione
<code>port <i>n_porta</i></code>	Permette di specificare il numero della porta da utilizzare, nel caso il server ne utilizzi una non standard.
<code>timeout <i>n_secondi</i></code>	Specifica il tempo massimo di inattività, dopo il quale si conclude la connessione, o il suo tentativo.
<code>interface <i>interfaccia / numero_ip / maschera</i></code>	Permette di specificare un'interfaccia di rete, assieme al gruppo di indirizzi che deve avere, prima di tentare la connessione con il server remoto.

### Opzioni dell'utente

Opzione	Descrizione
<code>user <i>utente_remoto</i></code> <code>username <i>utente_remoto</i></code>	Specifica il nome da utilizzare per accedere al sistema remoto.
<code>is <i>utente_locale</i> here</code>	Rappresenta il nome dell'utente locale che deve ricevere il messaggio. Di solito non si specifica, essendo quello che effettua l'operazione di recupero.

Opzione	Descrizione
pass <i>parola_d'ordine</i> password <i>parola_d'ordine</i>	La parola d'ordine per accedere al sistema remoto.
fetchall	Richiede espressamente il recupero di tutti i messaggi, compresi quelli già prelevati, ma mantenuti nel server per qualche motivo.
limit <i>n_byte</i>	Fissa la dimensione massima dei messaggi che possono essere prelevati. Quelli che eccedono tale limite vengono lasciati nel server e risultano «non letti».
syslog	Utilizza il registro di sistema per annotare gli errori.

Segue la descrizione di alcuni esempi.

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"
```

Rappresenta la scansione del server *roggen.brot.dg* con il protocollo POP3, utilizzando il nominativo-utente '**tizio**' che richiede la parola d'ordine '**frase segreta**' (indicato opportunamente tra virgolette).

```
poll roggen.brot.dg protocol pop3 username tizio password "frase segreta"  
poll schwarz.brot.dg username tizio1 password "ciao ciao"
```

Qui si prevede la scansione di due server, dove nel secondo caso non viene specificato il protocollo e anche il nominativo utilizzato risulta differente dal primo.

```
poll roggen.brot.dg
  protocol pop3
  username tizio
  password "frase segreta"

poll schwarz.brot.dg
  username tizio1
  password "ciao ciao"
```

Come nell'esempio precedente, ma più strutturato e più facile da leggere.

```
poll roggen.brot.dg protocol pop3
  username tizio password "frase segreta" is tizio here
  username caio password "ciao caio" is caio2 here
  username pippo password "marameo maramao" is pippo here
```

In questo caso, per uno stesso servente sono stati indicati diversi utenti remoti e locali. Per intendere il senso, si osservi che l'utente remoto '**caio**' corrisponde all'utente locale '**caio2**'.

Evidentemente, per ottenere un tale risultato, è necessario che l'utente che avvia Fetchmail conosca tutte le parole d'ordine di questi utenti.

#### 39.11.4 Gestione remota della posta elettronica



Trattando l'argomento del trasferimento della posta remota, non bisogna dimenticare la possibilità offerta da certi programmi MUA (*Mail user agent*) di gestirsi la posta elettronica senza doverla scaricare.

Va comunque osservato che la tendenza è quella di utilizzare la posta elettronica lì dove si trova, attraverso applicativi MUA offerti

con la mediazione di pagine HTML dinamiche (programmi CGI). In generale questo approccio è più «semplice» per l'utente comune, comportando però dei rischi maggiori per chi ha a cuore la riservatezza e la durata dei propri dati.

## 39.12 Messaggi, allegati ed estensioni MIME

Il messaggio di posta elettronica tradizionale è composto utilizzando soltanto la codifica ASCII a 7 bit e ha un aspetto simile all'esempio seguente: <<

```
Date: Tue, 17 Jul 2001 11:27:59 +0200
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Messaggio tradizionale
Message-Id: <E15MR95-0001Wb-00@dinkel.brot.dg>
```

Questo rappresenta un esempio di messaggio di posta elettronica tradizionale, dove si utilizzano solo i primi 7 bit.

In pratica, per quanto riguarda la lingua italiana, non si possono usare le lettere accentate.

Per garantire che un messaggio di posta elettronica viaggi attraverso qualsiasi server SMTP, può essere necessario che si rimanga nell'ambito dei soli 7 bit, oltre al fatto di mettere un limite alla lunghezza delle righe.

La necessità di scrivere in lingue differenti dall'inglese e di poter trasmettere informazioni diverse dal solito testo puro e semplice, ha fatto nascere lo standard multimediale MIME (*Multipurpose internet mail extensions*).

Con le estensioni multimediali MIME è possibile definire come deve

essere interpretato il contenuto di un messaggio di posta elettronica, il quale così può essere codificato in modo particolare, per trasportare anche informazioni diverse dal solo testo ASCII puro, rispettando i limiti tradizionali dei sistemi di trasporto dei messaggi.

Negli esempi che si mostrano in questo capitolo, viene omessa la riga di intestazione iniziale del tipo seguente, la quale è comunque essenziale per completare il messaggio, ma che qui non serve per comprendere quanto spiegato e rischia solo di creare confusione con il campo **'From:':**

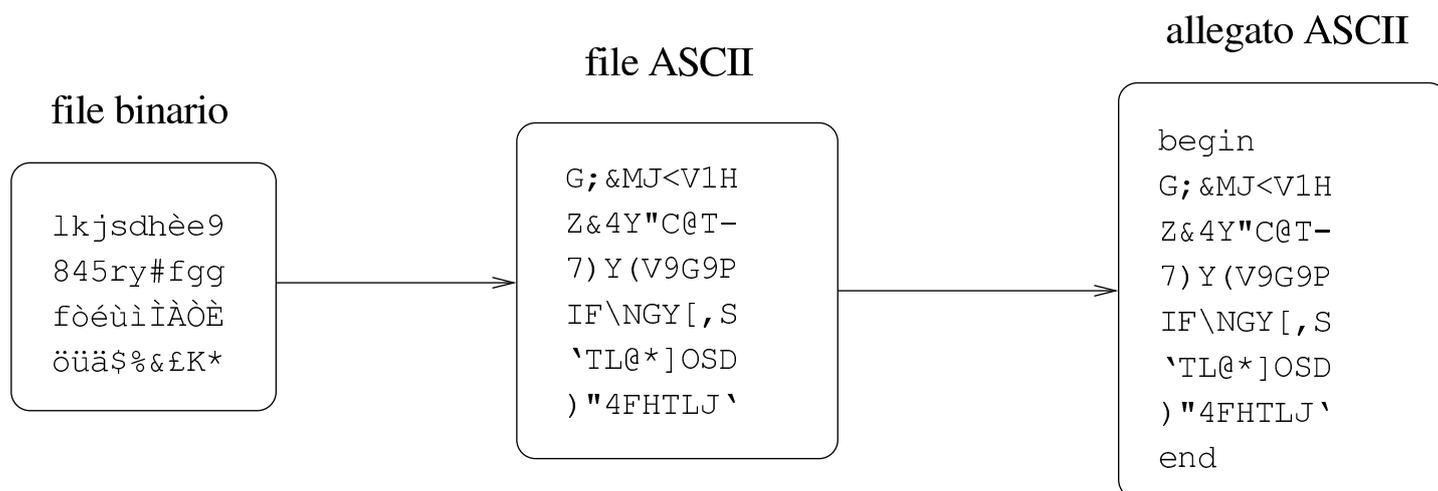
```
From danielle@swlibero.org Tue Jul 17 12:28:15 2001 +0200
```

### 39.12.1 Allegati

«

L'invio di un file allegato a un messaggio di posta elettronica richiede un modo per inserire e circoscrivere questo file, oltre alla sua trasformazione in modo tale che possa essere gestito come un file di testo normale. In pratica, è come allegare un file a un file di testo, dal quale deve poter essere estrapolato correttamente in un momento successivo.

Figura 39.44. Procedimento necessario a produrre un allegato.



Dal momento che in un messaggio di posta elettronica alcuni caratteri hanno un significato speciale (senza contare l'importanza di alcune parole chiave quando collocate a partire dalla prima colonna), sono da escludere anche questi nelle trasformazioni necessarie a creare gli allegati.

La figura 39.44 mostra in modo semplificato il problema che si tenta di descrivere: un file viene prima trasformato, in base a un certo algoritmo, in un file di testo puro che possa essere trasmesso attraverso il sistema della posta elettronica; questa trasformazione genera necessariamente un file più grande di quello di partenza; quindi, per diventare un allegato, occorre un modo per circoscriverlo, aggiungendo anche le informazioni necessarie a riprodurre il file originale (che nell'esempio della figura sono state omesse per semplicità).

### 39.12.2 Uuencode

Uuencode<sup>11</sup> è il sistema storico per la conversione di file di qualunque tipo in un allegato in forma di file ASCII, utilizzato senza gestire le estensioni MIME. Si compone di due eseguibili: **'uuencode'** per la codifica e **'uudecode'** per la decodifica.

Il programma **'uuencode'** si comporta in maniera differente a seconda che riceva il file da codificare dallo standard input, oppure che questo gli sia indicato come argomento della riga di comando:

```
uuencode [-m] file_da_codificare nome_da_usare
```

```
cat file_da_codificare | uuencode [-m] nome_da_usare
```

In entrambi i casi, il risultato della codifica viene emesso attraverso lo standard output, con la differenza che nel primo caso il file da codificare viene indicato come primo argomento, mentre nel secondo viene fornito attraverso lo standard input. L'ultimo argomento è sempre obbligatorio e rappresenta il nome che si vuole attribuire a questo file, ovvero il nome che viene usato nel momento dell'estrazione.

L'unica opzione disponibile, **'-m'**, consente di richiedere espressamente l'utilizzo della codifica Base64.

Disponendo del file già visto nella figura 39.44, ovvero il testo

```
lkjsdhèe9  
845ry#fgg  
fòéùìÌÀÒÈ  
öüä$%&£K*
```

supponendo che si tratti del file **'prova.xxx'**, si potrebbe codificare con **'uuencode'** nel modo seguente:

```
$ uuencode prova.xxx prova.xxx > allegato.txt [Invio]
```

Si può osservare che il nome **'prova.xxx'** appare due volte nella riga di comando: la prima volta indica il file da leggere per la codifica; la seconda indica il nome da indicare nell'allegato, in modo

che al momento della decodifica si riottienga lo stesso file. Il file ‘allegato.txt’ che si ottiene ha l’aspetto seguente:

```
begin 664 prova.xxx
G; &MJ<V1HZ&4Y"C@T-7) Y (V9G9PIF\NGY [, S `TL@*] OSD) "4FHTLJ
\
end
```

In alternativa, usando la codifica Base64,

```
$ uuencode -m prova.xxx prova.xxx > allegato.txt [Invio]
```

si ottiene invece:

```
begin-base64 664 prova.xxx
bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
=====
```

Evidentemente il principio è lo stesso, cambiando il modo di delimitare il file e di indicare le sue caratteristiche.

Il numero che appare dopo la parola chiave ‘**begin**’, o dopo ‘**begin-base64**’, rappresenta i permessi da attribuire al file, indicato subito dopo, in ottale. Nel caso dell’esempio, trattandosi di  $664_8$ , si intendono attribuire i permessi di lettura e scrittura al proprietario e al gruppo, lasciando solo i permessi di lettura agli altri utenti.

Naturalmente, si possono creare anche situazioni più complesse, come nel caso in cui il file di origine sia prima compresso, poi codificato e quindi trasmesso attraverso la posta elettronica:

```
$ cat prova.xxx | gzip | uuencode prova.xxx.gz ↵
↵| mail tizio@dinkel.brot.dg [Invio]
```

In questo caso, il messaggio che deve ricevere *tizio@dinkel.brot.dg* è, più o meno, quello seguente:

```
To: tizio@dinkel.brot.dg
Message-Id: <E15L3u4-00009I-00@dinkel.brot.dg>
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 16:26:48 +0200

begin 664 prova.xxx.gz
M'XL(`"<%3SL``\O)SBI.R7B1:LEE86):5*F<EI[.E?;IY<\W9PY<.L'U[<\3
/%56UQ=Y:`#NWZ88G````
`
end
```

Il programma **‘uudecode’** funziona in modo simmetrico rispetto a **‘uuencode’**. In questo caso, dal momento che il nome del file da rigenerare fa già parte delle informazioni necessarie dell'allegato, è sufficiente fornire a **‘uudecode’** il file di testo contenente l'allegato. Il file in questione può anche essere un messaggio di posta elettronica, completo di intestazione, come nell'ultimo esempio mostrato per la codifica.

```
uudecode [-o file_da_generare] file_con_allegato...
```

```
cat file_con_allegato | uudecode [-o file_da_generare]
```

In generale non si usa l'opzione **‘-o’**, a meno che ci sia la necessità di generare un file con un nome differente da quanto previsto da chi ha predisposto l'allegato.

```
$ uudecode allegato.txt [Invio]
```

L'esempio soprastante è elementare, ma rappresenta l'uso normale

di **'uudecode'**. In questo caso, il file `'allegato.txt'` è ciò che contiene l'allegato, dal quale viene estratto probabilmente un file, il cui nome è già stato deciso in precedenza.

### 39.12.3 Involucro MIME

Un messaggio realizzato secondo le estensioni MIME contiene informazioni aggiuntive specifiche nell'intestazione, come si vede nell'esempio seguente:

```
Date: Tue, 17 Jul 2001 12:28:23 +0200 (CEST)
From: caio@dinkel.brot.dg
To: danielle@dinkel.brot.dg
Subject: Messaggio MIME semplice
Message-ID: <Pine.LNX.4.04.10107171139070.5873@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=iso-8859-1
Content-Transfer-Encoding: QUOTED-PRINTABLE
```

Questo è un messaggio un po' più complesso, perché consente l'uso di un insieme di caratteri più ampio.

In generale appare il campo **'MIME-Version:'**, il quale dichiara l'utilizzo delle estensioni, secondo la versione indicata, anticipando così la presenza di altri campi specifici. L'elenco seguente descrive quelli essenziali.

- `Content-type: tipo/sottotipo [; opzione] ...`

Il campo **'Content-type:'** serve a specificare il tipo e il sottotipo MIME del messaggio. Esiste un tipo MIME particolare che serve a dichiarare la presenza di più componenti; si trat-

ta di **'multipart'** e viene chiarito meglio nel seguito il suo significato.

Il campo **'Content-type:'**, oltre al tipo e al sottotipo MIME, consente l'indicazione aggiuntiva di informazioni opzionali, precedute da un punto e virgola (';'), che chiariscono ulteriormente le caratteristiche dell'informazione contenuta. Per esempio, quando si tratta di **'text/plain'**, può essere specificato l'insieme di caratteri con l'opzione **'charset=insieme\_di\_caratteri'**. In mancanza di indicazioni, l'insieme di caratteri corrisponde a **'us-ascii'**, mentre nell'esempio si vede l'uso dell'insieme **'iso-8859-1'**, corrispondente a ISO 8859-1. Segue la descrizione delle opzioni più frequenti.

– `charset=insieme_di_caratteri`

Definisce l'insieme di caratteri nel caso si tratti di un testo. Il valore predefinito è **'us-ascii'**, mentre **'iso-8859-*n*'** rappresenta una codifica secondo lo standard ISO 8859-*n*.

– `name=file`

Definisce il nome del file nel caso il contenuto venga salvato.

– `boundary="stringa"`

Definisce la stringa di delimitazione del confine delle componenti MIME multiple.

• `Content-Transfer-Encoding: codifica_per_il_trasferimento`

Il campo '**Content-Transfer-Encoding:**' serve a specificare in che modo avviene la trasformazione delle informazioni stabilite nel campo '**Content-type:**', per le esigenze legate al trasferimento del messaggio. In pratica si tratta di indicare una parola chiave che chiarisca come interpretare il contenuto del messaggio al momento della ricezione. L'esempio mostra l'uso del tipo '**quoted-printable**' (non fa differenza l'uso delle maiuscole o delle minuscole).

— `Content-Transfer-Encoding: 7bit`

Si tratta della codifica predefinita, ovvero della situazione in cui non è necessario apportare alcuna trasformazione, perché si utilizzano solo i primi 7 bit e le righe di testo non sono troppo lunghe.

— `Content-Transfer-Encoding: 8bit`

In questo caso si tratta di un testo in cui vengono usati 8 bit, senza trasformazioni, con righe non troppo lunghe. Tuttavia, questa sarebbe una codifica non conveniente, perché non si può essere certi che tutti i server SMTP siano in grado di mantenere invariate tali informazioni.

— `Content-Transfer-Encoding: binary`

Le informazioni sono inserite così come sono, senza alcuna trasformazione. In generale è impossibile trasmettere messaggi di questo tipo.

– `Content-Transfer-Encoding: quoted-printable`

I caratteri che richiedono l'uso di 8 bit, si rappresentano nella forma `'=hh'`, dove la coppia *hh* rappresenta un numero esadecimale, corrispondente al codice del carattere. In pratica, la lettera «è» si rappresenta come `'=E8'` (come si può vedere dall'esempio); inoltre, per evitare di avere righe troppo lunghe, queste vengono spezzate ponendo il simbolo `'='` alla fine della riga; infine, il carattere «=`'=3D'`» viene rappresentato necessariamente come `'=3D'`.

– `Content-Transfer-Encoding: base64`

Si tratta di una trasformazione in cui ogni gruppo di 24 bit (3 byte) viene trasformato in quattro caratteri (4 byte), su righe non troppo lunghe. Il nome della codifica deriva dal fatto che per ogni byte si possono rappresentare solo 64 simboli, essendo necessario escludere tutto ciò che può creare problemi alla trasmissione del messaggio. Pertanto:  $2^{24} = 64^3$ .

Questo tipo di codifica rende completamente illeggibile, a livello umano, il suo contenuto. In questo senso, si presta alla trasmissione di immagini o di altri tipi di file che non sarebbero comunque leggibili in questo modo.

#### 39.12.4 Messaggi contenenti più parti MIME

«

Il tipo MIME `'multipart'` prevede la presenza di più componenti separate, con altrettante intestazioni specifiche. In questo caso si indica comunemente il confine tra una componente e l'altra attraverso



```
---1463811839-324931406-994342670=:16889
Content-Type: IMAGE/JPEG; NAME="caio-2.jpg"
Content-Transfer-Encoding: BASE64

/9j/4AAQSkZJRgABAQEAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsL
DBkSEw8UHRofHh0aHBwgJC4nICIsIxwckDcpLDAxNDQ0Hyc5PTgyPC4zNDL/
2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy
...
...
AgkiBwEifAQArSQpJAD/APah2keikDwgBeEj0iOUKXAFXKIZXKQ5PSJ54tdA
VIH7Innrwm9nlABJ8JpKcRQQDfKAGpD5TiEPhACHISI5TttBCigAcoHt08IA
ikACv1JHj5SXAP/Z
---1463811839-324931406-994342670=:16889--
```

In questo caso, la stringa **'-1463811839-324931406-994342670=:1'** viene usata per delimitare i vari componenti del messaggio. Si può osservare che quanto contenuto tra la fine dell'intestazione del messaggio e il primo componente MIME viene ignorato dai programmi utilizzati per leggerlo. Questa zona può essere usata per annotare informazioni tecniche destinate alla lettura umana, nel caso di un accesso diretto al file.

Si noti che ogni componente MIME è preceduto dalla stringa di delimitazione, a cui si aggiungono inizialmente due trattini ('--'). Alla fine, dopo l'ultimo componente la stringa di delimitazione ha altri due trattini finali. Volendo schematizzare la cosa:

```
Date: data
From: mittente
To: destinatario
Subject: oggetto
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="delimitatore"
```

```
[commento]
```

```
...
```

```
--delimitatore
Content-Type: tipo / sottotipo [; opzione] ...
Content-Transfer-Encoding: codifica_per_il_trasferimento
```

```
contenuto_codificato
```

```
...
```

```
...
```

```
[--delimitatore
Content-Type: tipo / sottotipo [; opzione] ...
Content-Transfer-Encoding: codifica_per_il_trasferimento
```

```
contenuto_codificato
```

```
...
```

```
...] ...
```

```
--delimitatore--
```

Teoricamente, un elemento MIME potrebbe scomporsi in altri sottoelementi, dichiarando nuovamente un tipo **'multipart'**, ma

questo modo di intervenire è sconsigliabile.

Un caso particolare di messaggi **'multipart'** è quello che consente di trasmettere il contenuto in forme alternative, come quando si affianca un messaggio in forma testuale a una copia più appariscente in formato HTML. In tal caso si aggiunge il sottotipo **'alternative'**:

```
Content-Type: multipart/alternative; boundary="xxxx"
```

La composizione del messaggio è analoga a quanto già visto, con la differenza che il programma che consente la lettura del messaggio ricevuto, sceglie in che modo visualizzare il contenuto.

### 39.12.5 Sistemazione manuale di un allegato MIME



I programmi usati generalmente per scrivere e inviare la posta elettronica sono in grado normalmente di gestire gli allegati, sia per inviarli, sia per estrarli. Ogni programma aggiunge a modo suo dei campi particolari per qualche scopo, anche se non si tratta di informazioni essenziali. Seguono due esempi, realizzati con programmi differenti.

```
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Message-ID: <Pine.LNX.4.04.10107131839040.579@dinkel.brot.dg>
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; ↵
↳BOUNDARY="-1463811839-1689890199-995042379=:579"
Content-ID: <Pine.LNX.4.04.10107131839530.579@dinkel.brot.dg>

---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
Content-ID: <Pine.LNX.4.04.10107131839531.579@dinkel.brot.dg>
```

Esempio di trasmissione con Pine.

```

---1463811839-1689890199-995042379=:579
Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1; NAME="prova.xxx"
Content-Transfer-Encoding: BASE64
Content-ID: <Pine.LNX.4.04.10107131839390.579@dinkel.brot.dg>
Content-Description:
Content-Disposition: ATTACHMENT; FILENAME="prova.xxx"

bGtqc2Ro6GU5DQo4NDVyeSNmZ2cNCmby6fnszMDSyA0K9vzkJCUmo0sq
---1463811839-1689890199-995042379=:579--

```

```

From: caio@dinkel.brot.dg
User-Agent: Mozilla/5.0 ↵
↳(X11; U; Linux 2.4.2 i586; en-US; m18) Gecko/20001103
MIME-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed;
  boundary="-----050408090202040304080207"

This is a multi-part message in MIME format.
-----050408090202040304080207
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit

Ecco un esempio di allegato con Mozilla.

-----050408090202040304080207
Content-Type: application/octet-stream;
  name="prova.xxx"
Content-Transfer-Encoding: base64

```

```
Content-Disposition: attachment;
  filename="prova.xxx"

bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
-----050408090202040304080207--
```

Purtroppo, alcune volte può capitare di ricevere messaggi in cui gli allegati sono stati inseriti in modo non standard, oppure utilizzando standard troppo recenti. In questi casi capita di non riuscire a estrarre il contenuto in alcun modo, a meno di mettere mano direttamente al messaggio, per correggere gli errori.

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Message-ID: <B761F178.202%caio@dinkel.brot.dg>
Mime-version: 1.0
Content-type: multipart/mixed;
  boundary="MS_Mac_OE_3076649336_173889_MIME_Part "

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: multipart/alternative;
  boundary="MS_Mac_OE_3076649336_173889_MIME_Part "

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/plain; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

Ecco, ti allego il file che tanto aspettavi.

--MS_Mac_OE_3076649336_173889_MIME_Part
Content-type: text/html; charset="ISO-8859-1"
Content-transfer-encoding: quoted-printable

<HTML>
<HEAD>
```

```
<TITLE>Esempio di allegato non corretto</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<P ALIGN=3DCENTER>
```

```
Ecco, ti allego il file che tanto aspettavi.
```

```
</BODY>
```

```
</HTML>
```

```
--MS_Mac_OE_3076649336_173889_MIME_Part--
```

```
--MS_Mac_OE_3076649336_173889_MIME_Part
```

```
Content-type: multipart/appledouble;
```

```
boundary="MS_Mac_OE_3076649333_192109_MIME_Part"
```

```
--MS_Mac_OE_3076649333_192109_MIME_Part
```

```
Content-type: application/applefile; name="prova.jpg"
```

```
Content-transfer-encoding: base64
```

```
Content-disposition: attachment
```

```
AAUWBwACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJAAAAPgAAACAAAAADAAAAXgAAABIAAAAC  
AAAACAAA02xKUEVHOEJJTQUA//8CAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEZSQU5DT18yIHntYWxs  
LmpwZwAAAQAAADrqaAA56gAAAIIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

```
...
```

```
...
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAAA66gAAOeoAAACCU09SVAN2AIAAHACCAARJQ04j  
AAAAK1BJQ1QAAAA2U1RSIAAAAEJpY2w4AAAAATnBub3QAAABav7n//wAAOOYM1aTQVHD//wAA  
ABoAAAAAv/T//wAAAAAM1afMv7n//wAANOIM1afcAAD//wAANNAM1aTY
```

```
--MS_Mac_OE_3076649333_192109_MIME_Part
```

```
Content-type: image/jpeg; name="prova.jpg";
```

```
x-mac-creator="3842494D";
```

```
x-mac-type="4A504547"
```

```
Content-disposition: attachment
```

```
Content-transfer-encoding: base64
```

```
/9j/4AAQSkZJRgABAQgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m  
bwAAAAB4ACgAAABIAEgAAAAAAxgcQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQFs
```

```

ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxvg6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==

--MS_Mac_OE_3076649333_192109_MIME_Part--

--MS_Mac_OE_3076649336_173889_MIME_Part--

```

L'esempio che si vede sopra è ovviamente abbreviato. L'intenzione di Caio era quella di inviare un'immagine a Tizio. Si tratta precisamente del file 'prova.jpg', ma per qualche motivo, non si riesce a estrarla.<sup>12</sup>

Il messaggio inizia con una breve descrizione, seguita dalla stessa cosa in HTML. Quindi appare un primo allegato, che in realtà non serve, quindi l'ultimo allegato corrispondente all'immagine cercata. Per rimediare, occorre salvare il messaggio in un file separato per poi metterci mano direttamente. Il messaggio trasformato per estrarre esclusivamente l'immagine cercata, può avere l'aspetto seguente, tenendo conto che probabilmente è necessario lasciare la prima riga di intestazione contenente il campo '**From** ...', che però qui è stata omessa:

```
Date: Fri, 13 Jun 2001 17:30:00 +0200
Subject: Esempio di allegato non corretto
From: caio@dinkel.brot.dg
To: tizio@dinkel.brot.dg
Mime-version: 1.0
Content-type: image/jpeg; name="prova.jpg";
Content-transfer-encoding: base64
```

```
/9j/4AAQSkZJRgABAQgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQF's
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxv6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
```

Si può osservare che il messaggio non è più di tipo MIME multiplo, così non è necessario indicare i confini con la stringa dell'opzione **'boundary'**.

Volendo, dal momento che l'immagine è stata codificata con la codifica Base64, si può usare anche Uuencode senza preoccuparsi di rispettare le specifiche MIME. Il file si riduce all'estratto seguente, dove il codice della figura è delimitato come si vede:

```
begin-base64 664 prova.jpg
/9j/4AAQSkZJRgABAQgEBLAEsAAD/7Ro4UGhvdG9zaG9wIDMuMAA4QklNA+kKUHJpbnQgSW5m
bwAAAAB4ACgAAABIAEgAAAAAAxgCQf/3//cDQAJKIAIFewPgAAAAAAFoAWgAAAAAD3gLRQF's
ADILRUcYAFAAAQEBAAAAAScPAAEAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAZAAAAAAAAAAA
...
...
QI08T/rKM3l/9Q/kKl+d8gj10WjbW2szHqF/2lrR6pr9PfEGJ3+n7v5bdyk9rgdwcQ6AIjSe
SpO/nP7P/fgp92f69mo9fBGjnnHtyOoMy72AOxQ5mKxv6Bde/8Al7Wtrr/cYrXpt+ltPMz8
uFK3+cPy/iif4H5JaXp9U+rr9H//2Q==
====
```

Per l'estrazione basta usare il programma **'uudecode'**, come è già

stato descritto in precedenza.

### 39.12.6 Mpack

«

Mpack<sup>13</sup> consente di generare allegati MIME, ovvero allegati con più informazioni e per questo più facili da estrarre. Anche in questo caso si distinguono due eseguibili: ‘**mpack**’ per la codifica e ‘**munpack**’ per la decodifica. Il primo, tra le altre cose, è anche in grado di inviare direttamente il risultato della codifica a un recapito di posta elettronica.

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↔
↔ [-c sottotipo_mime] ↔
↔ file_da_codificare indirizzo_posta_elettronica...
```

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↔
↔ [-c sottotipo_mime] ↔
↔ -o file_da_generare file_da_codificare
```

```
mpack [-s oggetto] [-d file_introduttivo] [-m n_caratteri] ↔
↔ [-c sottotipo_mime] ↔
↔ -n indirizzo_usenet [, indirizzo_usenet] ... file_da_codificare
```

I tre modelli sintattici mostrano tutte le opzioni disponibili e i tre contesti di utilizzo di ‘**mpack**’. Nel primo caso, il file codificato viene inviato direttamente attraverso la posta elettronica, agli indirizzi specificati; nel secondo caso si crea un file; nell’ultimo caso si invia il file codificato a uno o più gruppi di discussione di Usenet.

È importante chiarire il significato di alcune opzioni. ‘**-d**’ permette di indicare un file, il cui contenuto viene poi usato come introduzione all’allegato che si crea. In altri termini, permette di spiegare di cosa si tratta, senza interferire con il file da codificare. ‘**-m**’ consente di indicare la dimensione massima, espressa in caratteri, ovvero in byte, dei messaggi. Ciò permette di creare automaticamente diversi file, oppure di inviare diversi messaggi, ognuno non eccedente la dimensione richiesta.<sup>14</sup> Infine, l’opzione ‘**-c**’ consente di indicare un sottotipo MIME, dei tipi ‘**application**’, ‘**audio**’, ‘**image**’ e ‘**video**’. Se non si indica questa informazione, è ‘**mpack**’ a determinarla in modo automatico. È il caso di osservare che l’oggetto viene richiesto in modo interattivo, se non si usa l’opzione ‘**-s**’ esplicitamente.

A titolo di esempio si può vedere cosa succede se l’utente ‘**caio**’ invia a *tizio@dinkel.brot.dg* il file già visto in precedenza, denominato ‘*prova.xxx*’:

```
$ mpack -s "Prova di trasmissione" prova.xxx ↵  
↵      tizio@dinkel.brot.dg [Invio]
```

Ciò che viene ricevuto può assomigliare al messaggio seguente, dove si può notare che la stringa di delimitazione è ridotta a un solo trattino:

```
Message-ID: <846.995041413@dinkel.brot.dg>
Mime-Version: 1.0
To: tizio@dinkel.brot.dg
Subject: Prova di trasmissione
Content-Type: multipart/mixed; boundary="--"
From: caio@dinkel.brot.dg
Date: Fri, 13 Jul 2001 18:23:32 +0200
```

```
This is a MIME encoded message. Decode it with "munpack"
or any other MIME reading software. Mpack/munpack is available
via anonymous FTP in ftp.andrew.cmu.edu:pub/mpack/
```

```
---
```

```
Content-Type: application/octet-stream; name="prova.xxx"
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="prova.xxx"
Content-MD5: JSc+xPLb3o3I5NlBYvyVJA==
```

```
bGtqc2Ro6GU5Cjg0NXJ5I2ZnZwpm8un57MzA0sgK9vzkJCUmo0sq
```

```
-----
```

L'uso di **'munpack'** è più semplice, dal momento che nella maggior parte dei casi è sufficiente fornire il file contenente l'allegato, come argomento oppure attraverso lo standard input:

```
munpack [opzioni] file_con_allegato...
```

```
cat file_con_allegato | munpack [opzioni]
```

Il file che contiene l'allegato può anche essere un messaggio di posta elettronica, in cui appare ancora l'intestazione. Tuttavia, è da tenere in considerazione che viene estratto solo il primo messaggio che contiene un allegato, salvo il caso di allegati suddivisi in più

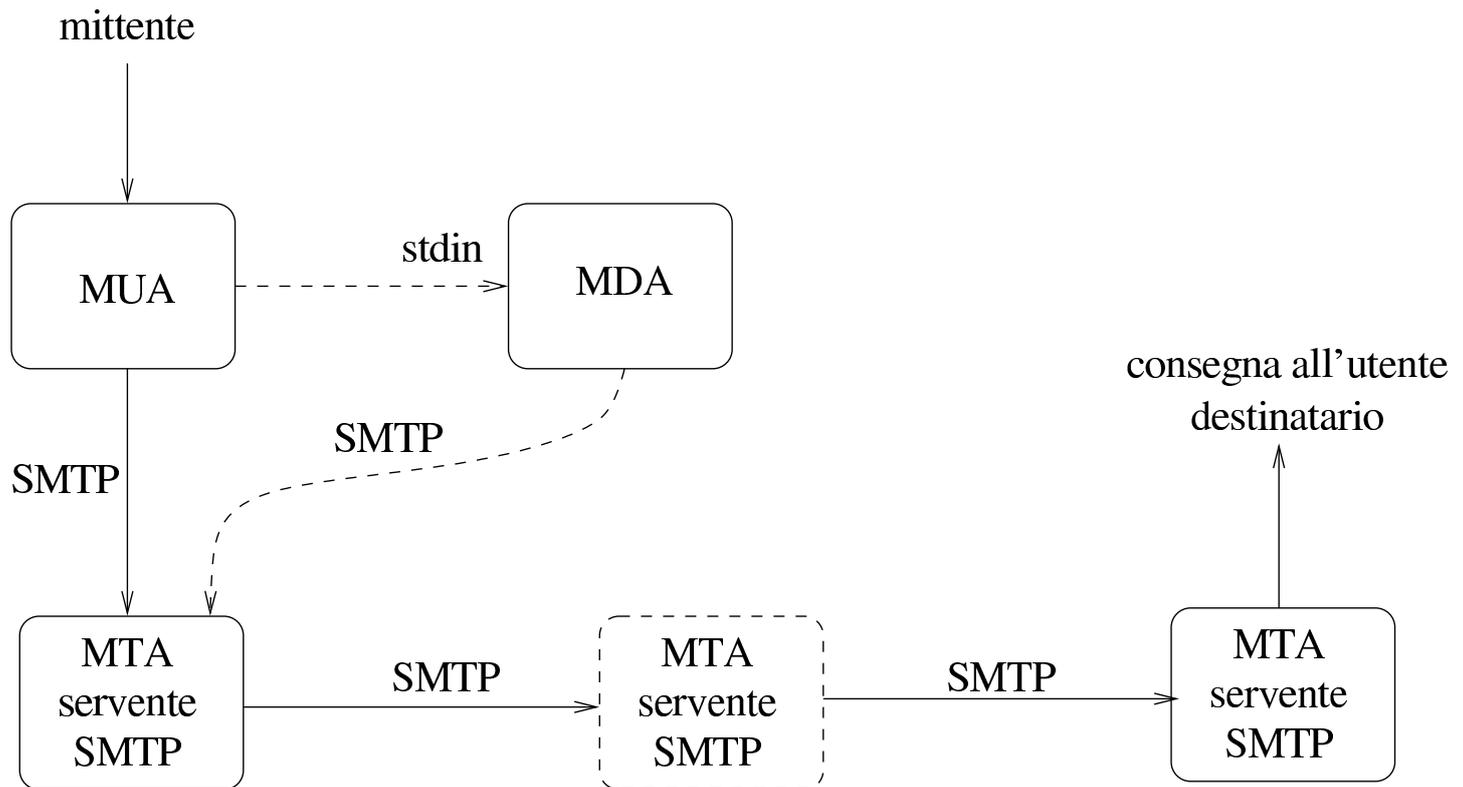
messaggi.

In condizioni normali, se il file o il messaggio contenente l'allegato è preceduto da una descrizione (un commento), questa informazione viene salvata in un file con estensione `' .desc '`.

### 39.13 Gestione della posta elettronica in generale

« I problemi di sicurezza che si presentano quando si amministra un MTA, impongono una conoscenza maggiore rispetto alla semplice messa in funzione del servizio. Generalmente, lo schema essenziale di funzionamento del sistema di trasferimento dei messaggi di posta elettronica è basato sul protocollo SMTP (*Simple mail transfer protocol*) e utilizza fondamentalmente due componenti: MTA (*Mail transport agent*), che include anche l'MDA (*Mail delivery agent*), e MUA (*Mail user agent*). Il primo dei due è il sistema che si occupa del trasferimento e della consegna dei messaggi, mentre il secondo è il programma che viene utilizzato per comporre i messaggi e passarli all'MTA.

Figura 39.57. Schema semplificato del meccanismo di trasmissione della posta elettronica tra MTA (MDA) e MUA.



Di solito l'MDA è un componente dello stesso MTA, il quale permette di provvedere alla consegna di un messaggio localmente, oppure alla trasmissione attraverso il protocollo SMTP, dopo averlo ricevuto dallo standard input. I programmi MUA più semplici dipendono dall'MDA, non essendo in grado di provvedere da soli a instaurare una connessione SMTP con un server di posta elettronica.

La sequenza di MTA, o meglio, di server SMTP utilizzati per trasmettere il messaggio a destinazione, dipende dall'organizzazione di ognuno di questi. La situazione più comune è quella in cui ne sono coinvolti solo due: quello utilizzato per iniziare la trasmissione e quello di destinazione che si occupa anche della consegna. In realtà, si possono porre delle esigenze diverse, a causa della struttura della

rete nel punto di partenza e nel punto di destinazione. Per rendere l'idea, si possono indicare i casi seguenti.

- L'MTA utilizzato nell'origine si avvale di uno *smarthost*, ovvero un altro MTA, collocato in una posizione conveniente della rete, che si occupa di smistare i messaggi. Ciò è utile quando l'MTA di origine è collocato in una posizione della rete per cui esiste un solo percorso per raggiungere la rete esterna: quando un messaggio è inviato a più di un destinatario è conveniente trasmetterlo una volta sola attraverso questo tratto di rete, lasciando che sia l'MTA esterno a provvedere alla duplicazione dei messaggi per i vari destinatari. Lo *smarthost* svolge quindi l'attività di relè, o di scambio.
- L'MTA di destinazione è il punto di ingresso a una rete privata, nella quale vengono poi usati altri MTA per la consegna effettiva dei messaggi.
- L'MTA di destinazione è solo il punto di arrivo di un alias (da quel punto riprende l'invio del messaggio all'indirizzo vero dell'utente).

### 39.13.1 Composizione di un messaggio

Un messaggio di posta elettronica è composto da due parti fondamentali: l'intestazione e il corpo. Il corpo è quella parte che contiene il testo del messaggio, mentre l'intestazione contiene informazioni amministrative di vario genere, compreso l'oggetto (*subject*). All'interno dell'intestazione, si distingue in particolare la **busta** o *envelope*, cioè quelle informazioni amministrative necessarie al trasporto



del messaggio; queste appaiono nella parte superiore e si espandono mano a mano che il messaggio attraversa i vari MTA necessari a raggiungere la destinazione.

L'esempio seguente mostra un breve messaggio trasmesso da *pippo@router.brot.dg* a *daniele@dinkel.brot.dg*.

```
From pippo@router.brot.dg Mon Jun 8 21:53:16 1998
Return-Path: <pippo@router.brot.dg>
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
From: pippo@router.brot.dg
Received: (from pippo@localhost)
        by router.brot.dg (8.8.7/8.8.7) id AAA00384
        for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
Date: Tue, 9 Jun 1998 00:00:09 +0200
Message-Id: <199806082200.AAA00384@router.brot.dg>
To: daniele@dinkel.brot.dg
Subject: Una vita che non ci si sente :-)

Ciao Daniele!
Quanto tempo che non ci si sente.
Fai un cenno se possibile :-)

Pippo
```

Per distinguere la conclusione dell'intestazione dall'inizio del corpo, si utilizza una riga vuota. Nell'esempio, L'oggetto è l'ultimo elemento dell'intestazione, quindi appare una riga vuota di separazione e finalmente inizia il testo del messaggio.

L'intestazione è composta da record separati dal codice di interruzione di riga. Ognuno di questi record, definisce l'informazione contenuta in un campo nominato all'inizio del record stesso, precisamente nella prima colonna del testo. Questi campi (*field*) termina-

no necessariamente con il carattere due punti (‘:’), seguito da uno spazio; il resto del record descrive il loro contenuto. Un record può continuare su più righe; la continuazione viene segnalata da un carattere di tabulazione orizzontale, `<HT>`, all’inizio della riga che continua il record interrotto in quella precedente (si osservino a questo proposito i campi ‘**Received:**’ dell’esempio).

Il programma usato come MUA genera l’intestazione necessaria a iniziare la trasmissione del messaggio. In particolare, sono fondamentali i campi seguenti.

Campo	Descrizione
Date:	Contiene la data di invio del messaggio.
Message-Id:	Contiene una stringa generata automaticamente, in modo da essere unica per il messaggio. In un certo senso, serve a dare un’impronta al messaggio che permette di distinguerlo e di farvi riferimento.
From:	Contiene le informazioni sul mittente del messaggio; generalmente si tratta dell’indirizzo di posta elettronica e probabilmente anche il suo nome reale.
To:	Contiene l’indirizzo di posta elettronica del destinatario.
Subject:	L’oggetto del messaggio.

Oltre ai campi già visti, ne possono essere aggiunti altri, a seconda delle esigenze o dell’impostazione del programma utilizzato come MUA.

Campo	Descrizione
Reply-To:	Permette di indicare un indirizzo al quale si desidera siano inviate le risposte.
Organization:	Permette di definire l'organizzazione proprietaria della macchina da cui ha origine il messaggio di posta elettronica.
X-...:	I campi che iniziano per 'X-' sono ammessi, senza essere definiti. In pratica, vengono utilizzati per scopi vari, accordati tra le parti.

Per una convenzione ormai consolidata, il primo record dell'intestazione di un messaggio di posta elettronica inizia con la parola chiave **From** seguita immediatamente da uno spazio. Questo record è diverso da quello che definisce il campo **From:** (cioè quello che termina con i due punti), tanto che per distinguerlo viene spesso indicato come **From\_**, per sottolineare il fatto che non appaiono i due punti prima dello spazio.

La presenza di questo campo un po' anomalo, fa sì che quando si scrive un messaggio, nel corpo non possa apparire la parola **From** scritta in questo modo e a partire dalla prima colonna. Convenzionalmente, se ne esiste la necessità, viene aggiunto il carattere '>' davanti a questa (**>From**). Il problema si pone essenzialmente quando si vuole incorporare un messaggio di posta elettronica all'interno di un nuovo messaggio; il programma che si usa per comporre il testo dovrebbe provvedere da solo a correggere la riga in cui appare il record **From\_**.

I vari MTA che si occupano di trasferire e consegnare il messaggio a destinazione sono responsabili dell'aggiunta dei campi

**‘Received:’**. Questi vengono aggiunti a ogni passaggio, dal basso verso l’alto, allo scopo di tenere traccia degli spostamenti che il messaggio ha dovuto subire. Nell’esempio mostrato in precedenza, sono stati interessati solo due MTA.

1. Il primo campo **‘Received:’** partendo dal basso rappresenta il primo MTA che è stato interpellato.

```
Received: (from pippo@localhost)
      by router.brot.dg (8.8.7/8.8.7) id AAA00384
      for daniele@dinkel.brot.dg; Tue, 9 Jun 1998 00:00:09 +0200
```

Trattandosi dello stesso nodo da cui è stato inviato il messaggio, appare solo l’informazione dell’MTA, **‘by router.brot.dg’**, e la destinazione, **‘for daniele@dinkel.brot.dg’**.

2. Il secondo campo **‘Received:’** viene aggiunto dal secondo MTA interpellato, che in questo caso è anche l’ultimo.

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
      by dinkel.brot.dg (8.8.7/8.8.7) with ESMTTP id VAA00615
      for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

L’MTA provvede prima a identificare l’origine, ovvero l’MTA che gli ha trasmesso il messaggio, attraverso l’indicazione **‘from router.brot.dg’**; quindi identifica se stesso attraverso l’indicazione **‘by dinkel.brot.dg’**.

I vari record **‘Received:’** possono essere più o meno ricchi di informazioni e questo dipende dall’MTA che li genera. In particolare, l’indicazione della data permette eventualmente di comprendere in che punto la trasmissione del messaggio è stata ritardata; inoltre, la presenza dell’identificativo **‘id’** può permettere di ricercare informazioni su una trasmissione particolare all’interno di registrazioni eventuali.

Alcuni MTA, per motivi di sicurezza, verificano l'origine della trasmissione attraverso il sistema DNS e includono il nome e l'indirizzo IP così ottenuto tra parentesi. Nell'esempio mostrato, il secondo MTA ha indicato `'from router.brot.dg (pippo@router.brot.dg [192.168.1.254])'`.

### 39.13.2 Messaggi contraffatti e punto di iniezione

«

La posta elettronica è stato il primo problema della comunicazione nella rete; così, gli standard che si sono ottenuti e i programmi a disposizione sono potentissimi dal punto di vista delle possibilità che vengono offerte. Tutto questo, assieme al fatto che la trasmissione dei messaggi di posta elettronica è un'operazione gratuita per il mittente, ha favorito chi usa la posta elettronica per «offendere»: sia attraverso la propaganda indesiderata, sia attraverso altre forme più maliziose. Pertanto, la conoscenza dei punti deboli di un MTA è importante per comprendere con quanta serietà vada presa la sua amministrazione e anche con quanta prudenza vadano mosse delle accuse verso il presunto mittente di un messaggio indesiderato.

Chi utilizza la posta elettronica per attaccare qualcuno, cerca di farlo in modo da non essere identificato. Per questo si avvale normalmente di un MTA di partenza diverso da quello normalmente competente per la sua rete di origine (il proprio ISP). Oltre a tutto, di solito l'attacco consiste nell'invio di un messaggio a una grande quantità di destinatari, per cui, la scelta di un MTA estraneo (e innocente) serve per scaricare su di lui tutto il lavoro di distribuzione. Il «lavoro» di ogni ipotetico aggressore sta quindi nella ricerca di un MTA che si lasci manovrare e nella composizione di un messaggio con un'intestazione fasulla che lasci intendere che il messaggio è già transitato

da un'altra origine (che può esistere effettivamente o meno).

A parte il problema derivato dal fatto che la configurazione degli MTA è difficile, per cui capita spesso che qualcosa sfugga cosicché l'MTA si trova a permettere accessi indesiderabili, lo standard SMTP è tale per cui l'MTA che riceve un messaggio deve accettare le informazioni che gli vengono fornite riguardo ai punti di transito precedenti (i vari campi '**Received:**' già esistenti). Quando i campi '**Received:**' sono stati contraffatti l'MTA dal quale ha origine effettivamente la trasmissione è il cosiddetto *punto di iniezione*.

L'esempio seguente mostra un messaggio di questo tipo, in cui l'origine, *hotmail.com*, si è dimostrata fasulla. Probabilmente, il punto di iniezione è stato '**cnn.Princeton.EDU**', ma questo non può essere stabilito in modo sicuro.

```
X-POP3-Rcpt: daniele@tv
Return-Path: <seeingclearly40@hotmail.com>
Received: from outbound.Princeton.EDU (outbound.Princeton.EDU [128.112.128.88])
        by tv.calion.com (8.8.4/8.8.4) with ESMTP
        id HAA02209 for <daniele@tv.shineline.it>;
        Tue, 9 Jun 1998 07:12:59 +0200
Received: from IDENT-NOT-QUERIED@Princeton.EDU (port 4578 [128.112.128.81])
        by outbound.Princeton.EDU with SMTP
        id <542087-18714>;
        Tue, 9 Jun 1998 00:48:58 -0400
Received: from cnn.Princeton.EDU by Princeton.EDU (5.65b/2.139/princeton)
        id AA09882; Tue, 9 Jun 98 00:17:18 -0400
Received: from hotmail.com by cnn.Princeton.EDU (SMI-8.6/SMI-SVR4)
        id AAA12040; Tue, 9 Jun 1998 00:17:13 -0400
Message-Id: <199806090417.AAA12040@cnn.Princeton.EDU>
Date: Mon, 08 Jun 98 11:09:01 EST
From: "Dreambuilders" <seeingclearly40@hotmail.com>
To: Friend@public.com
Subject: Real Business
```

HOW WOULD YOU LIKE TO BE PAID LIKE THIS?

\*How about if you received compensation on 12 months Business Volume for

```
every transaction in your entire organization and this made it possible for
you to earn over $14000.00 US in your first month ?
```

```
* How about if you were paid daily, weekly, and monthly ?...
```

```
* How about if you could do business everywhere in the world and be paid in
US dollars ?
```

```
* What if your only out of pocket expense was a $10 processing fee to get
started...
```

```
* Would you want to evaluate a business like that ?
```

```
If so reply with "real business" in subject box to foureal25@hotmail.com
```

### 39.13.3 Identificazione della destinazione



In precedenza, si è accennato al meccanismo di trasferimento dei messaggi tra diversi MTA. L'MTA di origine, o comunque quello utilizzato come distributore di origine (relè), deve identificare l'MTA più adatto a ricevere il messaggio per ottenere la consegna di questo all'utente destinatario. Intuitivamente, il problema potrebbe ridursi alla trasformazione del nome a dominio dell'indirizzo di posta elettronica del destinatario in un numero IP, per poi tentare di contattare tale nodo con la speranza di trovare un MTA pronto a rispondere. Ma la realtà è più complessa e può darsi benissimo che l'MTA competente per ricevere la posta elettronica di un certo utente sia un nodo diverso da quello che appare nell'indirizzo di posta elettronica.

Per pubblicizzare gli MTA competenti per la gestione di un certo dominio di posta elettronica, si utilizzano i record **'MX'** nella configurazione dei DNS. L'esempio seguente mostra un caso descritto meglio nel capitolo [33](#) in cui si stabilisce che, per consegnare messaggi

di posta elettronica nel dominio *brot.dg*, è competente il server *dinkel.brot.dg*.

```
...
brot.dg.          IN          MX          10 dinkel.brot.dg.
...
```

### 39.13.4 Misure di sicurezza

Le misure di sicurezza fondamentali attraverso cui si cerca di evitare l'uso improprio di un MTA sono essenzialmente di due tipi: l'identificazione del sistema da cui proviene la richiesta di inoltro di un messaggio (attraverso il DNS) e il rifiuto dei messaggi che sono originati da un dominio estraneo e sono diretti anche a un dominio estraneo.

La prima delle due misure si concretizza nell'indicazione tra parentesi del nome a dominio e del numero IP del nodo chiamante nel campo '**Received:**'. Nell'esempio visto in precedenza, l'MTA del nodo *dinkel.brot.dg* ha verificato l'indirizzo di chi lo ha contattato (*router.brot.dg*).

```
Received: from router.brot.dg (pippo@router.brot.dg [192.168.1.254])
        by dinkel.brot.dg (8.8.7/8.8.7) with ESMTTP id VAA00615
        for <daniele@dinkel.brot.dg>; Mon, 8 Jun 1998 21:53:15 +0200
```

La seconda misura si avvale generalmente del servizio di risoluzione dei nomi (record '**MX**'), attraverso il quale si può determinare quale sia il dominio di competenza per il recapito dei messaggi, stabilendo così che i messaggi provenienti dall'esterno che non siano diretti al proprio dominio di competenza, non possono essere accettati.

Nella maggior parte dei casi, gli MTA sono (o dovrebbero essere) configurati in questo modo. Ciò dovrebbe spiegare il motivo per cui

spesso è impossibile inviare messaggi di posta elettronica in una rete locale se prima non si attiva un servizio DNS.

### 39.13.5 Referente per l'amministrazione del servizio

«

L'amministratore di un servizio di distribuzione di posta elettronica deve essere raggiungibile attraverso dei nominativi convenzionali. Fondamentalmente si tratta di *postmaster@dominio*. Ultimamente, a causa della crescente invadenza di chi utilizza la posta elettronica in modo fraudolento, è diventato comune l'utilizzo dell'indirizzo *abuse@dominio* per identificare la persona competente nei confronti di possibili abusi originati dal servizio di sua competenza.

Naturalmente, tali indirizzi sono generalmente degli alias attraverso cui i messaggi possono essere rinivati al recapito dell'utente che incorpora effettivamente tali competenze.

## 39.14 Pratica manuale con i protocolli

«

È importante avere un minimo di dimestichezza con i protocolli utilizzati per la gestione della posta elettronica. Oltre all'aspetto puramente didattico, il loro utilizzo manuale attraverso un cliente TELNET, può aiutare a verificare la configurazione di un server SMTP, oppure di manovrare all'interno di una propria casella postale remota.

In queste sezioni vengono mostrati solo i comandi elementari che si possono utilizzare con il protocollo SMTP e POP3.

## 39.14.1 SMTP attraverso un cliente TELNET



È già stato mostrato in precedenza un esempio di connessione con un servizio SMTP allo scopo di inviare manualmente un messaggio. Lo stesso esempio viene mostrato nuovamente a vantaggio del lettore.

```
$ telnet roggen.brot.dg smtp [Invio]
```

```
Trying 192.168.1.2...
```

```
Connected to roggen.brot.dg.
```

```
Escape character is '^]'.  
↵
```

```
220 roggen.brot.dg ESMTP Sendmail 8.8.5/8.8.5; ↵
```

```
↵Thu, 11 Sep 1997 19:58:15 +0200
```

```
HELO brot.dg [Invio]
```

```
250 roggen.brot.dg Hello dinkel.brot.dg [192.168.1.1], ↵
```

```
↵pleased to meet you
```

```
MAIL From: <daniele@dinkel.brot.dg> [Invio]
```

```
250 <daniele@dinkel.brot.dg>... Sender ok
```

```
RCPT To: <toni@dinkel.brot.dg> [Invio]
```

```
250 <toni@dinkel.brot.dg>... Recipient ok
```

```
DATA [Invio]
```

```
354 Enter mail, end with "." on a line by itself
```

```
Subject: Saluti. [Invio]
```

```
Ciao Antonio, [Invio]
```

```
come stai? [Invio]
```

```
Io sto bene e mi piacerebbe risentirti. [Invio]
```

**Saluti**, [*Invio*]

**Daniele** [*Invio*]

. [*Invio*]

250 TAA02951 Message accepted for delivery

**QUIT** [*Invio*]

221 dinkel.brot.dg closing connection  
Connection closed by foreign host.

L'esempio mostra tutto quello che serve fare per inviare un messaggio. I comandi **'HELO'**, **'MAIL'**, **'RCPT'** e **'DATA'**, vanno inseriti rispettando questa sequenza e la loro sintassi dovrebbe essere evidente dall'esempio.

Un problema importante che si incontra quando si configura il proprio servizio SMTP è quello del filtro rispetto al relè, cioè all'attività di ritrasmissione dei messaggi. Solitamente si consente di fare il relè senza alcuna limitazione per i messaggi provenienti dai nodi della propria rete locale, mentre lo si impedisce quando il messaggio è di origine esterna a tale rete e in più la stessa destinazione è esterna alla rete locale. Il concetto si esprime facilmente a parole, ma la configurazione del servizio SMTP potrebbe essere complessa e si può rischiare di tagliare fuori dal servizio proprio alcuni nodi che invece dovrebbero poterlo utilizzare. L'esempio seguente mostra un caso di cattiva configurazione e da ciò si intende quanto sia utile l'utilizzo manuale del protocollo SMTP per controllare tali situazioni.

```
$ telnet dinkel.brot.dg smtp [Invio]
```

Dal nodo *roggen.brot.dg* si vuole inviare un messaggio al nodo

*weizen.brot.dg*, utilizzando per questo il server *dinkel.brot.dg*, il quale dovrebbe fare da relè, almeno per la rete locale *brot.dg*.

```
Trying 192.168.1.1...
Connected to dinkel.brot.dg.
Escape character is '^]'.
220 roggen.brot.dg ESMTP Exim 1.90 #1 Wed,
    4 Nov 1998 09:47:05 +0100
```

**HELO brot.dg** [*Invio*]

```
250 dinkel.brot.dg Hello daniele at roggen.brot.dg
    [192.168.1.2]
```

**MAIL From: daniele@roggen.brot.dg** [*Invio*]

```
250 <daniele@roggen.brot.dg> is syntactically correct
```

**RCPT To: tizio@weizen.brot.dg** [*Invio*]

```
550 relaying to <tizio@weizen.brot.dg> prohibited by
    administrator
```

Come si può vedere, qualcosa non va: il server ha accettato l'origine, ma da quell'origine non accetta la destinazione.

**QUIT** [*Invio*]

```
221 roggen.brot.dg closing connection
```

## 39.14.2 POP3 attraverso un cliente TELNET

Anche l'utilizzo manuale del protocollo POP3 può essere utile. Il problema si pone normalmente quando la propria casella postale remota è stata riempita in maniera abnorme da un aggressore. Se si dispone di un collegamento troppo lento, è meglio evitare di scari- <<

care tutta la posta, mentre sarebbe opportuno eliminare direttamente i messaggi che sembrano essere inutili.

L'esempio seguente serve a capire in che modo è possibile visionare la situazione della propria casella postale remota e come è possibile intervenire per eliminare i messaggi indesiderati.

```
$ telnet dinkel.brot.dg pop-3 [Invio]
```

```
Trying 192.168.1.1...
Connected to dinkel.brot.dg.
Escape character is '^]'.
+OK POP3 dinkel.brot.dg v4.47 server ready
```

La prima cosa richiesta è l'inserimento del nominativo-utente e subito dopo la parola d'ordine.

```
USER tizio [Invio]
```

```
+OK User name accepted, password please
```

```
PASS tazza [Invio]
```

Dopo l'indicazione della parola d'ordine, il servizio POP3 indica quanti messaggi sono presenti. In questo caso solo due.

```
+OK Mailbox open, 2 messages
```

Il comando '**LIST**' consente di avere un elenco dei messaggi con a fianco la loro dimensione in byte. Ciò può essere utile per individuare messaggi «bomba», dove l'indizio potrebbe essere dato dalla dimensione esageratamente grande di un messaggio o dal ripetersi di messaggi con la stessa identica dimensione.

```
LIST [Invio]
```

```
+OK Mailbox scan listing follows
```

```
1 520
```

```
2 498
```

```
.
```

In questo caso, i messaggi sembrano proprio innocui. Eventualmente, se si vede il ripetersi di un messaggio breve, si può controllarne il contenuto, con il comando **RETR**.

```
RETR 2 [Invio]
```

Viene letto il secondo messaggio.

```
+OK 498 octets
```

```
Return-path: <daniele@dinkel.brot.dg>
```

```
Envelope-to: daniele@dinkel.brot.dg
```

```
Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100
```

```
Received: from daniele by dinkel.brot.dg with local  
for daniele@dinkel.brot.dg
```

```
id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100
```

```
To: daniele@dinkel.brot.dg
```

```
Subject: SPAM
```

```
Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>
```

```
From: daniele@dinkel.brot.dg
```

```
Date: Wed, 4 Nov 1998 10:06:30 +0100
```

```
Status:
```

```
questo e` un messaggio SPAM.
```

```
.
```

La dimensione del messaggio comprende tutto ciò che lo compone, compresa la riga iniziale in cui si informa che questa è di 498 ottetti (gruppi di 8 bit), ovvero byte.

Per cancellare un messaggio, si può utilizzare il comando **DELE**,

seguito dal numero corrispondente.

```
DELE 2 [Invio]
```

```
+OK Message deleted
```

Per concludere si utilizza il comando **QUIT**.

```
QUIT [Invio]
```

```
+OK Sayonara
```

### 39.14.3 POP3s attraverso un cliente TELNET-SSL

«

Il protocollo POP3s si distingue in quanto si inserisce a sua volta in un protocollo SSL. Si può intervenire manualmente anche in questo caso, se Telnet consente di gestire anche SSL.

```
$ telnet [-8] -z ssl mail.brot.dg 995 [Invio]
```

```
Trying 192.168.1.99...
```

```
Connected to mail.brot.dg.
```

```
Escape character is '^]'.
```

```
+OK POP3 mail.brot.dg v2003.83 server ready
```

```
USER tizio [Invio]
```

```
+OK User name accepted, password please
```

```
PASS tazza [Invio]
```

```
+OK Mailbox open, 2 messages
```

```
LIST [Invio]
```

+OK Mailbox scan listing follows

1 520

2 482

.

**RETR 2** [*Invio*]

+OK 482 octets

Return-path: <daniele@dinkel.brot.dg>

Envelope-to: tizio@mail.brot.dg

Delivery-date: Wed, 4 Nov 1998 10:06:30 +0100

Received: from daniele by dinkel.brot.dg with local (Exim 1.90 #1  
for tizio@mail.brot.dg

id 0zayta-00009R-00; Wed, 4 Nov 1998 10:06:30 +0100

To: tizio@mail.brot.dg

Subject: SPAM

Message-Id: <E0zayta-00009R-00@dinkel.brot.dg>

From: daniele@dinkel.brot.dg

Date: Wed, 4 Nov 1998 10:06:30 +0100

Status:

questo e` un messaggio SPAM.

.

**DELE 2** [*Invio*]

+OK Message deleted

**QUIT** [*Invio*]

+OK Sayonara

## 39.14.4 Script per l'invio di un messaggio attraverso Telnet



Come è stato mostrato nelle sezioni precedenti, se lo scopo è quello di scrivere un messaggio semplice (ASCII puro), privo di allegati, è possibile usare direttamente il protocollo SMTP attraverso Telnet. In questa sezione viene mostrato uno script che permette di inviare un messaggio preparato in un file di testo separato, a un indirizzo pre-stabilito, inviandone una copia anche a se stessi, per memoria. Supponendo che lo script si chiami `'mail-tizio@dinkel.brot.dg'`, lo si potrebbe usare così:

```
mail-tizio@dinkel.brot.dg file_messaggio [oggetto]
```

Ecco il contenuto dello script `'mail-tizio@dinkel.brot.dg'`:

```
#!/bin/sh

SENDER="caio@roggen.brot.dg"
SMTP_SERVER="mail.brot.dg"
RECIPIENT="tizio@dinkel.brot.dg"
MESSAGE_ID=`makepasswd --chars 11`
DATE=`date -R`
MESSAGE_BODY=`cat $1`
SUBJECT=$2
MAIL_FILE=$1~

echo "HELO $SENDER" > $MAIL_FILE
echo "MAIL From: <$SENDER>" >> $MAIL_FILE
echo "RCPT To: <$RECIPIENT>" >> $MAIL_FILE
echo "RCPT To: <$SENDER>" >> $MAIL_FILE
echo "DATA" >> $MAIL_FILE
echo "Message-ID: $MESSAGE_ID" >> $MAIL_FILE
echo "Date: $DATE" >> $MAIL_FILE
echo "Sender: $SENDER" >> $MAIL_FILE
```

```
echo "From: $SENDER" >> $MAIL_FILE
echo "To: $RECIPIENT" >> $MAIL_FILE
echo "Subject: $SUBJECT" >> $MAIL_FILE
echo "" >> $MAIL_FILE
echo "$MESSAGE_BODY" >> $MAIL_FILE
echo "" >> $MAIL_FILE
echo "." >> $MAIL_FILE

cat $MAIL_FILE | telnet $SMTP_SERVER 25
rm -f $MAIL_FILE
```

L'intestazione del messaggio che si ottiene è abbastanza completa, in modo da non dover costringere il server SMTP a completarla. Si può osservare in particolare che viene generata una stringa casuale attraverso il programma `makepasswd` per il campo `Message-ID:`. Da come è fatto lo script è evidente che il mittente e il destinatario sono fissi, così come suggerisce il nome stesso dello script.

Si suppone di avere preparato il messaggio seguente nel file `messaggio`:

```
Ciao Tizio,
come va?

E' da tanto che non ci si sente...
Raccontami qualcosa!

Caio
```

Come si può osservare, per prudenza si evita di indicare lettere accentate. Per inviare il messaggio si può procedere in questo modo, specificando l'oggetto `«Ciao!»`:

```
$ mail-tizio@dinkel.brot.dg messaggio "Ciao!" [Invio]
```

Prima dell'invio, lo script genera il file 'messaggio~' con il contenuto seguente:

```
HELO caio@roggen.brot.dg
MAIL From: <caio@roggen.brot.dg>
RCPT To: <tizio@dinkel.brot.dg>
RCPT To: <caio@roggen.brot.dg>
DATA
Message-ID: LPhJyaTLUvE
Date: Mon, 16 Jun 2003 11:36:51 +0200
Sender: caio@roggen.brot.dg
From: caio@roggen.brot.dg
To: tizio@dinkel.brot.dg
Subject: Ciao!

Ciao Tizio,
come va?

E' da tanto che non ci si sente...
Raccontami qualcosa!

Caio

.
```

Come si vede dallo script, questo file viene inviato a Telnet attraverso lo standard input e ciò è sufficiente per ottenere l'invio. Alla fine, il file temporaneo viene rimosso.

Eventualmente, si può sostituire Telnet con Netcat (sezione [43.8.9](#)).

## 39.15 Procmail



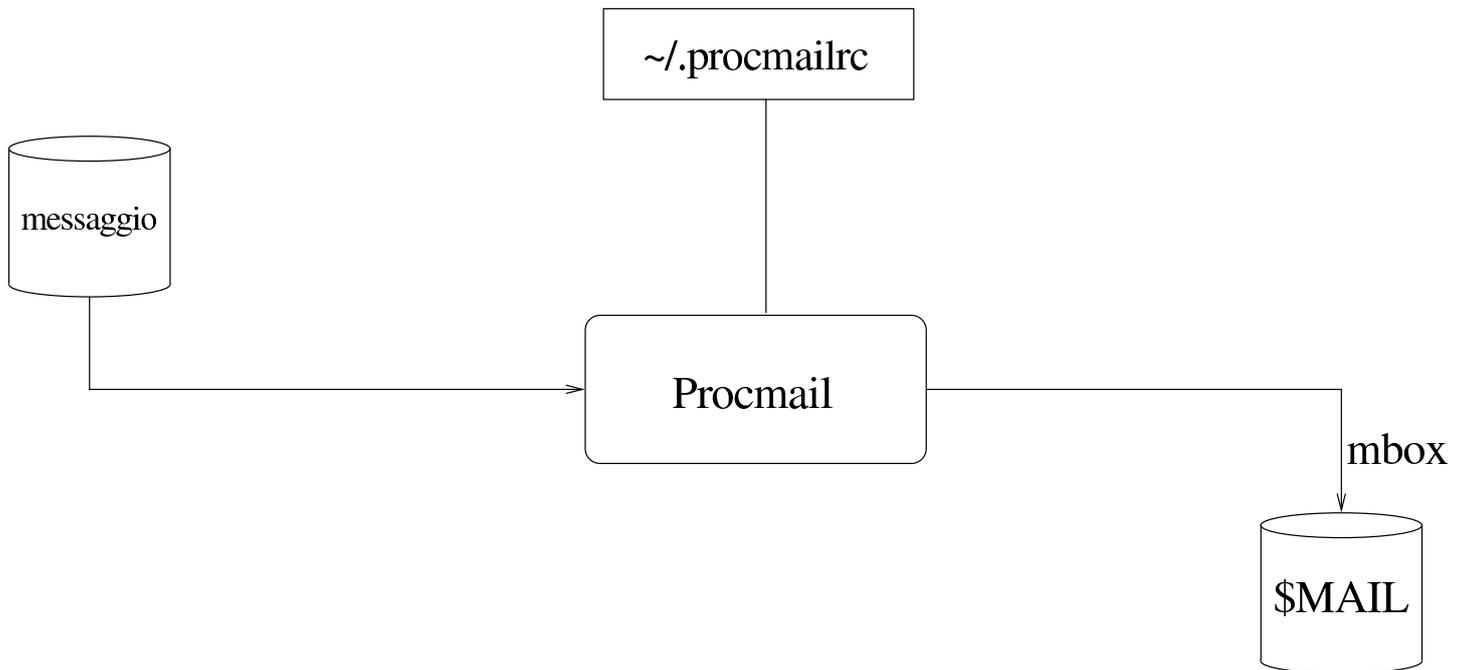
Procmail<sup>15</sup> è un sistema di analisi e selezione dei messaggi di posta elettronica, che si inserisce subito dopo un MDA (*Mail delivery agent*). Viene usato praticamente per ogni tipo di controllo che riguardi la posta elettronica, a livello di singolo utente, ma ha un grande difetto: la sintassi per la sua configurazione.

Procmail, nel suo utilizzo normale, viene avviato con i privilegi di un certo utente e serve per ricevere un messaggio di posta elettronica attraverso lo standard input, da depositare nel file appropriato che rappresenta la casella di posta in entrata di quello stesso utente. Per la precisione, qualsiasi sia la forma dei dati che vengono ricevuti in ingresso, questi vengono depositati tali e quali nella casella di posta.

```
cat messaggio | procmail
```

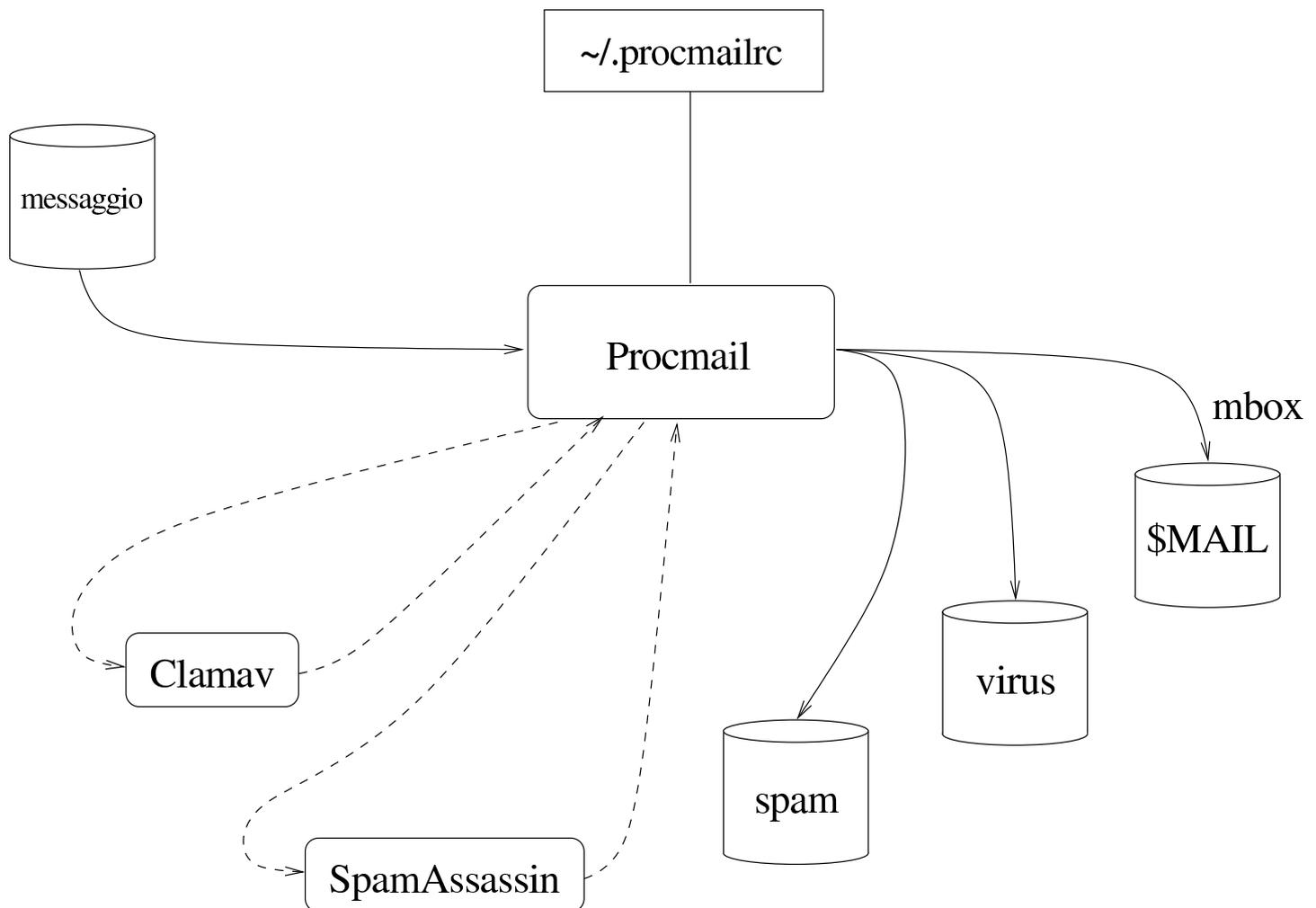
Il funzionamento di Procmail dipende dalla presenza e dal contenuto di un file di configurazione. Generalmente si considera solo il file ‘~/`.procmailrc`’, di competenza dell’utente, proprio perché Procmail lo si intende uno strumento che deve gestire l’utente singolo.

Figura 39.95. Schema del funzionamento più semplice di Procmail.



Attraverso la configurazione, si può istruire Procmail in modo da selezionare i messaggi per depositarli in file differenti, in base a qualche criterio, così come è possibile utilizzare altri programmi per il controllo della presenza di virus o per l'individuazione di «spam», i quali aggiungono delle voci nell'intestazione dei messaggi, così che lo stesso Procmail possa poi separarli dai messaggi normali.

Figura 39.96. Una situazione tipica in cui Procmail si avvale di altri programmi per individuare i contenuti e sapere poi come separare i messaggi, recapitandoli in file differenti.



### 39.15.1 Configurazione di partenza e verifica del funzionamento

Per cominciare a comprendere l'uso di Procmail, occorre predisporre un file di configurazione iniziale ('~/procmailrc'), molto simile a quello seguente:



```
PATH=/usr/local/bin:/usr/bin:/bin
MAILDIR=$HOME/mail
DEFAULT=$MAILDIR/mbox
LOGFILE=$MAILDIR/procmail.log
```

Come si può intuire, vengono definite delle variabili di ambiente per il funzionamento di Procmail stesso. In particolare, la variabile **MAILDIR** rappresenta la directory in cui vengono depositati i file per il recapito dei messaggi, mentre **DEFAULT** rappresenta il file che deve ricevere i messaggi in modo predefinito. Eventualmente, la variabile **DEFAULT** potrebbe anche corrispondere a `‘/var/mail/$LOGNAME’`.

Per la precisione, la directory rappresentata dalla variabile **MAILDIR** è la directory corrente durante il funzionamento di Procmail; pertanto, i file che vengono indicati con percorsi relativi, fanno riferimento a questa directory di partenza.

Avendo fatto questo, si può utilizzare un file contenente il testo seguente, per verificare il funzionamento di Procmail:

```
From tizio@brot.dg Wed Jul 5 12:13:59 2012 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: <E1Fy4ON-0005yF-00@127.0.0.1>
From: tizio@brot.dg
Date: Wed, 05 Jul 2012 12:13:59 +0200

ciao
```

Supponendo che questo file si chiami ‘messaggio’, si può vedere se Procmail lo può recapitare regolarmente:

```
$ cat messaggio | procmail [Invio]
```

Indipendentemente dal fatto che l'utente sia effettivamente 'caio', dovrebbe trovare il messaggio nel file '~/.mail/mbox', da come si vede nella configurazione stabilita. Ma più importante di questo, nel file '~/.mail/procmail.log' si deve vedere cosa ha fatto Procmail:

```
From tizio@brot.dg Wed Jul 5 12:13:59 2012 +0200
Subject: ciao
Folder: /home/tizio/mail/mbox 387
```

### 39.15.2 Attivazione di Procmail

Per svolgere il suo compito, Procmail deve essere avviato ogni volta che c'è un messaggio da recapitare a un certo utente. <<

Si parte dal presupposto che il sistema, senza Procmail, sia già in grado di recapitare i messaggi agli utenti, pur senza compiere analisi dei contenuti di questi. Quando si vuole inserire Procmail, quello che prima svolgeva il compito di MDA, dopo deve avvalersi a sua volta di Procmail per completare il recapito.

A seconda dei casi, può darsi che Procmail venga preso in considerazione in modo automatico dal sistema di recapito dei messaggi di posta elettronica, oppure che si debba intervenire all'interno di file '~/.forward'.

A titolo di esempio viene mostrato un estratto della configurazione di Postfix, dove viene richiesto l'uso di Procmail:

```
...
mailbox_command = procmail -a "$EXTENSION"
...
```

Quando invece il programma che gestisce la consegna dei messaggi ignora l'esistenza di Procmail, occorre utilizzare il file `'~/ .forward'`. Potrebbe essere necessario utilizzare una delle due forme seguenti, ma si deve verificare con la documentazione del sistema MDA:

```
"|exec /usr/bin/procmail"
```

```
! /usr/bin/procmail
```

### 39.15.3 Esempi semplici di configurazione

«

Il file di configurazione di Procmail contiene, oltre alle direttive per assegnare un valore a delle variabili di ambiente, delle «ricette» (*recipe*) con cui si dice cosa fare dei messaggi elaborati. Si osservi l'esempio seguente:

```
PATH=/usr/local/bin:/usr/bin:/bin
MAILDIR=$HOME/mail
DEFAULT=$MAILDIR/mbox
LOGFILE=$MAILDIR/procmail.log
#
# Lista "scuola"
#
:0 c
* ^To:.*scuola@lists\.linux\.it
didattica
```

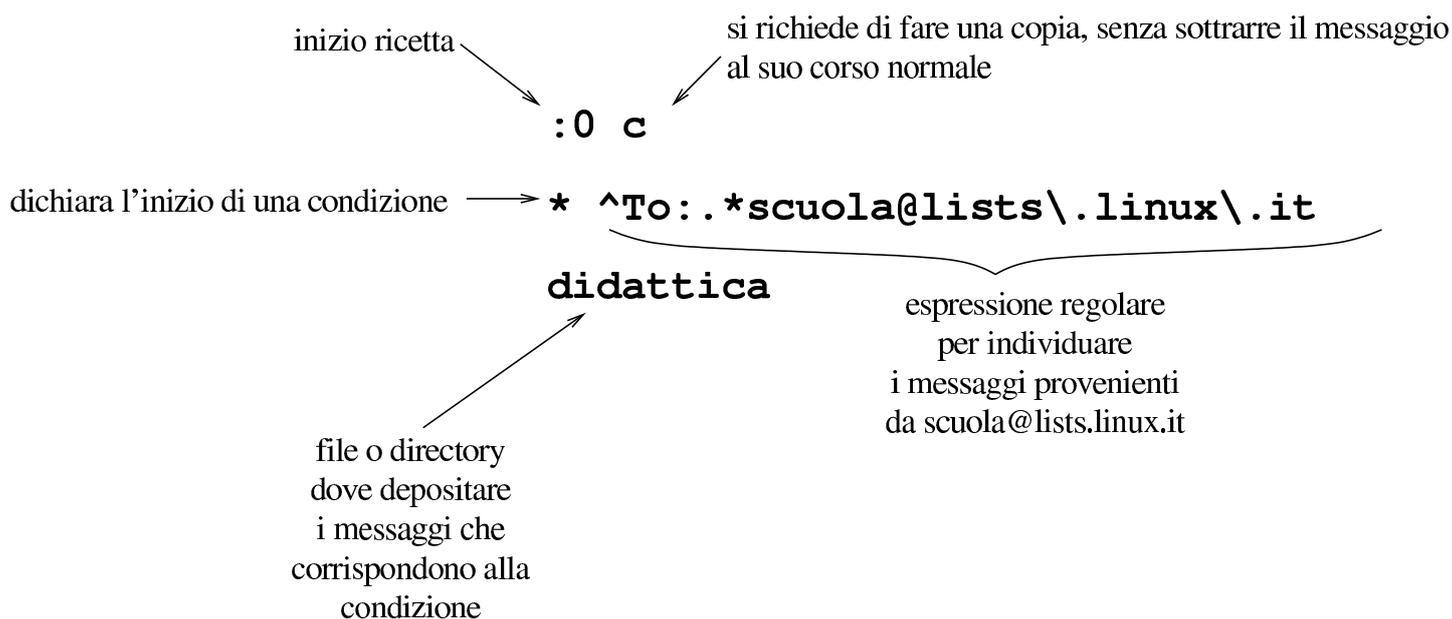
In questo caso si mostra un file completo, il quale, dopo l'assegnamento delle variabili di ambiente e dopo un'annotazione (commento) contiene una ricetta:

```
:0 c
* ^To:.*scuola@lists\.linux\.it
didattica
```

Le ricette si distinguono perché iniziano sempre con la sigla ‘:0’. In questo caso, la ricetta indica che si vuole mettere una copia dei messaggi che risultano diretti all’indirizzo *scuola@lists.linux.it* nel file ‘didattica’ (precisamente il file ‘\$MAILDIR/didattica’).

Si osservi che ‘didattica’ potrebbe anche essere una directory, ma in tal caso ci sarebbe da specificare se salvare i messaggi in formato «MH» o *maildir*.

Figura 39.105. Spiegazione dettagliata della ricetta.



Nell’esempio seguente, invece, si vede la stessa ricetta, con la differenza che manca la «c», per fare in modo che i messaggi individuati dalla condizione vengano messi solo nel file o nella directory ‘didattica’:

```
:0
* ^To:.*scuola@lists\.linux\.it
didattica
```

L'esempio successivo riguarda l'uso di un programma antivirus e si avvale di due ricette in sequenza:

```
#
# Scan for viruses
#
:0
VIRUS=|clamscan --no-summary --stdout -

:0
* VIRUS ?? ^.*FOUND
virus
```

La prima ricetta richiede di avviare il programma '**clamscan**' (con le opzioni che si vedono), inviandogli il messaggio attraverso lo standard input. Il risultato della scansione è un testo descrittivo che viene emesso dal programma attraverso lo standard output, che così viene assegnato alla variabile **VIRUS**. La seconda ricetta prende lo stesso messaggio e verifica che la variabile **VIRUS** contenga la stringa '**FOUND**' alla fine: se c'è la corrispondenza, il messaggio viene messo nel file o nella directory '`virus`'.



attraverso il condotto che si vede dopo la condizione, venga modificato e passato alle ricette successive con tale modifica. La lettera **'w'** richiede di attendere l'esecuzione del programma e di verificare il valore di uscita dello stesso: se si ottiene un errore, occorre rifiutare le modifiche al messaggio, il quale così passa intatto alle ricette successive.

Dopo le opzioni, appare il segno di due punti (':'), perché poi viene indicato il nome di un file, `'spamassassin.lock'`, che viene creato nel momento dell'utilizzo della ricetta e cancellato subito dopo. La presenza di questo file serve a evitare che il programma **'spamassassin'** venga avviato quando ne esiste già un altro che non ha ancora completato il suo compito; pertanto, all'esecuzione della ricetta, se il file esiste già, si attende che il file scompaia prima di procedere (in inglese si definisce: *lock file*).

La condizione della prima ricetta si avvera se il messaggio ha una dimensione inferiore a 256000 byte (ovvero 250 Kibyte). Ciò serve a evitare di scandire messaggi di dimensioni troppo grandi. Tali messaggi più grandi non vengono così controllati dal programma **'spamassassin'**.

Nella prima ricetta, il programma **'spamassassin'** aggiunge delle intestazioni ai messaggi, in particolare una denominata **'X-Spam-Level:'**, contenente una fila di asterischi: più sono gli asterischi, più è probabile che si tratti di messaggi indesiderabili. Nella seconda ricetta, infatti, si verifica la presenza di un'intestazione di questo tipo: se appaiono almeno 15 asterischi, il messaggio viene messo nel file o nella directory `'spam'`.

Per la descrizione dettagliata della sintassi da usare per la co-

struzione delle ricette di Procmail, occorre leggere i documenti: *procmailrc(5)* e *procmailex(5)*.

## 39.16 SpamAssassin

SpamAssassin è un sistema sofisticato per l'analisi dei messaggi di posta elettronica, allo scopo di individuare quelli che sono da ritenere *spam*, ovvero messaggi privi di alcun valore. «

Il sistema di SpamAssassin può articolarsi in modi differenti; qui viene preso in considerazione semplicemente l'uso attraverso il programma '**spamassassin**', tralasciando la possibilità di usare la coppia '**spamd**'/'**spamc**'.

SpamAssassin ha la capacità di «imparare» a selezionare i messaggi in base a esempi reali di messaggi *spam* e di messaggi «buoni» (*ham*). Per accumulare queste informazioni può avvalersi di un DBMS esistente, oppure può semplicemente salvare dei file nella directory '`~/ .spamassassin/`'. In questo capitolo si considera solo tale ultima possibilità.

### 39.16.1 Configurazione di SpamAssassin

SpamAssassin, per distinguere ciò che è da intendersi come *spam*, utilizza dei file di configurazione che possono trovarsi in '`/usr/share/spamassassin/`' e hanno estensione '`.cf`'. Questi file possono essere aggiornati, attraverso il programma '**sa-update**', il quale però li scarica all'interno di '`/var/lib/spamassassin/...`'; pertanto, se esistono i file all'interno di '`/var/lib/spamassassin/`', vengono ignorati quelli all'interno di '`/usr/share/spamassassin/`'.

Per modificare questa configurazione non si deve intervenire nelle directory appena descritte, mentre è possibile agire all'interno del file `/etc/spamassassin/local.cf` ed eventualmente nei file personali `~/ .spamassassin/user_prefs`.

Per le situazioni comuni, non è necessario intervenire nella configurazione e l'uso di `sa-update` è più che sufficiente.

```
sa-update [opzioni]
```

Naturalmente, `sa-update` va usato con i privilegi dell'utente `root` e non servono opzioni se è stato compilato con i valori predefiniti corretti.

```
# sa-update [Invio]
```

## 39.16.2 Cosa fa SpamAssassin

«

Qui si intende mostrare l'uso del programma `spamassassin`, il quale riceve dallo standard input un messaggio di posta elettronica e, dopo la verifica, gli aggiunge delle intestazioni con cui è possibile qualificarlo:

```
spamassassin [opzioni] < file_messaggio > file_modificato
```

Nella documentazione originale si fa riferimento al programma `spamassassin-run`, ma in generale si usa solo il nome `spamassassin`.

A titolo di esempio si veda cosa succede con un file che ha il contenuto seguente:

```

From tizio@brot.dg Thu Jul 6 19:17:20 2012 +0200
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
Date: Thu, 06 Jul 2012 19:17:19 +0200

ciao

```

Supponendo che si tratti del file ‘messaggio’, si può usare ‘**spamassassin**’ per controllarlo e ottenere il file ‘messaggio\_controllato’:

```
$ spamassassin < messaggio > messaggio_controllato [Invio]
```

In questo caso, SpamAssassin dovrebbe considerarlo un messaggio normale e in tal caso, si limita a segnalarlo con delle intestazioni aggiuntive. Ecco come dovrebbe apparire il file ‘messaggio\_controllato’:

```

From tizio@brot.dg Thu Jul 6 19:17:20 2012 +0200
X-Spam-Checker-Version: SpamAssassin 3.1.1 ↵
↵(2006-03-10) on nanohost
X-Spam-Level:
X-Spam-Status: No, score=0.5 required=5.0 ↵
↵tests=NO_REAL_NAME,NO_RECEIVED,
      NO_RELAYS autolearn=no version=3.1.1
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: ciao
Message-Id: >E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg

```

```
Date: Thu, 06 Jul 2012 19:17:19 +0200
```

```
ciao
```

In presenza di un messaggio che SpamAssassin considera essere *spam*, le cose vanno diversamente: le intestazioni aggiuntive sono più corpose, inoltre il messaggio originale viene allegato in un rapporto esplicativo. Ecco il file di partenza, contenente un elenco ripetuto di nomi di prodotti farmaceutici che spesso qualcuno tenta di spacciare attraverso la rete:

```
From tizio@brot.dg Thu Jul 6 19:17:20 2012 +0200
Envelope-to: caio@brot.dg
Delivery-date: Thu, 06 Jul 2012 19:17:20 +0200
To: caio@brot.dg
Subject: need cialis, levitra, soma, valium, vicodin?
Message-Id: <E1FyXTb-000093-00@127.0.0.1>
From: caio@brot.dg
Date: Thu, 06 Jul 2012 19:17:19 +0200
```

```
cialias, levitra, soma, valium, vicodin
...
```

Ecco il risultato dopo l'elaborazione con **'spamassassin'**:

```
From tizio@brot.dg Thu Jul 6 19:17:20 2012 +0200
Received: from localhost by nanohost
```

with SpamAssassin (version 3.1.1);

Thu, 06 Jul 2012 19:27:07 +0200

From: caio@brot.dg

To: caio@brot.dg

Subject: need cialis, levitra, soma, valium, vicodin?

Date: Thu, 06 Jul 2012 19:17:19 +0200

Message-Id: <E1FyXTb-000093-00@127.0.0.1>

**X-Spam-Flag: YES**

X-Spam-Checker-Version: SpamAssassin 3.1.1 (2006-03-10) on nanohost

**X-Spam-Level: \*\*\*\*\***

**X-Spam-Status: Yes**, score=6.3 required=5.0 tests=AWL,DRUGS\_ANXIETY, DRUGS\_ANXIETY\_ERE, DRUGS\_ERECTILE, DRUGS\_MANYKINDS, DRUGS\_MUSCLE, DRUGS\_PAIN, NO\_REAL\_NAME, NO\_RECEIVED, NO\_RELAYS, SUBJECT\_DRUG\_GAP\_C, SUBJECT\_DRUG\_GAP\_L, SUBJECT\_DRUG\_GAP\_S, SUBJECT\_DRUG\_GAP\_VA, SUBJECT\_DRUG\_GAP\_VIC autolearn=no version=3.1.1

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="-----=\_44AD47EB.2E5AF19D"

This is a multi-part message in MIME format.

-----=\_44AD47EB.2E5AF19D

Content-Type: text/plain

Content-Disposition: inline

Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "nanohost", has identified this incoming email as possible spam. The original message has been attached to this so you can view it (if it isn't spam) or label similar future email. If you have any questions, see the administrator of that system for details.

Content preview: cialis, levitra, soma, valium, vicodin ... [...]

Content analysis details: (6.3 points, 5.0 required)

pts	rule name	description
0.6	NO_REAL_NAME	From: does not include a real name
2.4	SUBJECT_DRUG_GAP_VA	Subject contains a gappy version of 'valium'



```
-----=_44AD47EB.2E5AF19D--
```

La prima volta che un certo utente usa SpamAssassin, il programma crea automaticamente la directory ‘~/ .spamassassin/’ con la configurazione predefinita e altri file, che servono per le proprie annotazioni interne.

### 39.16.3 Filtrare i messaggi automaticamente

Per filtrare i messaggi di posta elettronica con SpamAssassin, si può intervenire nella configurazione del MDA (*Mail delivery agent*), oppure, in modo più semplice e generalizzato, si può fare affidamento su Procmal (sezione 39.15). Quelle che seguono sono le direttive che si possono inserire nel file ‘~/ .procmalrc’ per questo scopo, tratte dalla documentazione di SpamAssassin stesso, con qualche piccola semplificazione: <<

```
# SpamAssassin sample procmalrc
#
# Pipe the mail through spamassassin (replace 'spamassassin' with 'spamc'
# if you use the spamc/spamd combination)
#
# The condition line ensures that only messages smaller than 250 kB
# (250 * 1024 = 256000 bytes) are processed by SpamAssassin. Most spam
# isn't bigger than a few k and working with big messages can bring
# SpamAssassin to its knees.
#
# The lock file ensures that only 1 spamassassin invocation happens
# at 1 time, to keep the load down.
#
:0fw: spamassassin.lock
* < 256000
| spamassassin

# Mails with a score of 15 or higher are almost certainly spam (with 0.05%
# false positives according to rules/STATISTICS.txt). Let's put them in a
# different mbox. (This one is optional.)
```



## 39.17 Liste di posta elettronica

Una lista di posta elettronica, o *mailing-list*, o più semplicemente *lista*, è un servizio attraverso cui un gruppo di persone può inviare dei messaggi di posta elettronica a tutti i partecipanti, creando in pratica un mezzo per discutere di un certo argomento. Sotto questo aspetto, la *mailing-list* compie lo stesso servizio di un *newsgroup*, con la differenza che ci si deve iscrivere presso il servente (o il «robot») che offre il servizio e che i messaggi vengono inviati a tutti i partecipanti iscritti.

Dal momento che la lista di posta elettronica richiede questa forma di iscrizione, tende a escludere i visitatori occasionali (o casuali), ma permette ugualmente l'accesso a un numero di utenti più vasto: tutti quelli che hanno la possibilità di usare la posta elettronica. Infatti, per quanto riguarda i *newsgroup*, sono rari gli utenti di Internet che possono accedere a tutti i gruppi di discussione.

Il servizio di una lista di posta elettronica viene svolto normalmente da un programma che si occupa di ricevere la posta da un certo indirizzo e conseguentemente di rispedire i messaggi a tutti gli iscritti. Per iscriversi occorre inviare un messaggio speciale al programma che lo gestisce, contenente il nome della lista e l'indirizzo di posta elettronica di colui che si iscrive; in modo analogo si interviene per cancellare l'iscrizione.

Dal punto di vista amministrativo, si distinguono due tipi di liste: moderate e non moderate. Una lista moderata è quella in cui tutti i messaggi, prima di essere ritrasmessi agli iscritti, vengono controllati da uno o più moderatori; l'altro tipo di lista non viene controllata da alcuno.

### 39.17.1 Lista elementare

«

Prima di vedere il funzionamento di un applicativo organizzato per la gestione di una lista, conviene apprenderne i rudimenti realizzandone una elementare attraverso la gestione degli alias.

Se l'obiettivo che ci si prefigge è solo quello di definire un indirizzo di posta elettronica che serva come punto di riferimento per il proseguimento (*forward*) dei messaggi a un elenco di persone, si può agire in due modi differenti: modificando il file `/etc/aliases`, oppure creando un utente fittizio che possieda nella sua directory personale il file `~/ .forward`.

Il secondo caso, quello dell'utente fittizio, è il più semplice da comprendere. Se si suppone di voler creare la lista **prova**, basta registrare un utente con lo stesso nome nel sistema operativo, facendo opportunamente in modo che questo non abbia una parola d'ordine valida e nemmeno una shell funzionante. Nella sua directory personale si crea e si gestisce il file `~/ .forward` nel quale vanno inseriti gli indirizzi degli utenti iscritti alla lista **prova**. È tutto qui; spetta all'amministratore del servizio l'aggiornamento manuale di questo file. Eventualmente, questo amministratore potrebbe essere un utente diverso dall'utente **root**, per cui si potrebbe anche fare in modo che l'utenza **prova** possa funzionare regolarmente (con parola d'ordine e shell), lasciandola usare a tale persona.

Il metodo della creazione dell'alias è più efficace. Generalmente si crea un file contenente l'elenco degli indirizzi degli iscritti alla lista e si fa in modo che un alias faccia riferimento a tutti questi indirizzi. Per esempio, se nel file `/etc/aliases` viene inserita la riga seguente,

```
prova:                :include:/var/liste/prova/iscritti
```

si fa in modo che tutti i messaggi diretti all'indirizzo **'prova'** siano poi rinviati a tutti gli indirizzi indicati nel file `'/var/liste/prova/iscritti'`. Dal momento che con questo sistema si hanno maggiori possibilità nella definizione dei nomi, si può aggiungere convenientemente un alias per l'amministratore del servizio, come nell'esempio seguente:

```
prova:                :include:/var/liste/prova/iscritti
prova-admin          daniele
```

A seconda delle caratteristiche specifiche del MTA utilizzato, può darsi che sia necessario usare il comando **'newaliases'** dopo una modifica del file `'/etc/aliases'`. Nel caso fosse così, è importante ricordarsene.

In entrambi i casi visti è possibile mantenere un archivio dei messaggi ricevuti dalla lista, con la semplice aggiunta di un indirizzo che faccia riferimento a un file su disco. Per esempio, il file `'~prova/.forward'` potrebbe iniziare nel modo seguente:

```
"/home/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
...
```

Nello stesso modo, il file `'/var/liste/prova/iscritti'` potrebbe iniziare come segue:

```
"/var/liste/prova/archivio"
Tizio Tizi <tizio@dinkel.brot.dg>
Caio Cai <caio@dinkel.brot.dg>
...
```

Bisogna fare attenzione ai permessi. È molto probabile che il file venga creato con i privilegi dell'utente `'mail'`. La prima volta conviene fare in modo che la directory che deve accogliere tale file abbia tutti i permessi necessari alla scrittura da parte di chiunque, in modo da vedere cosa viene creato effettivamente. Successivamente si possono regolare i permessi in modo più preciso.

## 39.17.2 Mailman

«

Mailman<sup>16</sup> è un sistema per la gestione di una lista di posta elettronica, gestito attraverso programmi CGI (capitolo 40). Questo tipo di lista di posta elettronica dipende pertanto, oltre che da un MTA adatto, anche da un server HTTP (capitolo 40) in grado di consentire il funzionamento di programmi CGI; inoltre richiede di configurare Cron per la gestione delle operazioni periodiche.

Nella descrizione che qui viene fatta di Mailman, si trascura completamente, o quasi, ciò che riguarda la configurazione di Cron, dell'MTA e del server HTTP, perché è molto probabile che la propria distribuzione GNU sia in grado di predisporre tutto questo in modo automatico, nel momento dell'installazione del pacchetto che corrisponde a questo applicativo. Eventualmente si può leggere la documentazione originale di Mailman che dovrebbe essere accessibile a partire da <http://www.gnu.org/software/mailman/mailman.html>.

### 39.17.2.1 Privilegi durante il funzionamento

«

Quando un programma di Mailman viene messo in funzione, dovrebbe acquisire privilegi limitati. Per questo, di solito gli si associa un utente e un gruppo particolari, che potrebbero corrispondere a un nome del tipo `'mailman'`, oppure `'list'`. In condizioni normali, se

si installa Mailman da un pacchetto predisposto per la propria distribuzione GNU, tutto dovrebbe essere sistemato in modo automatico, compreso l'aggiornamento del file `/etc/aliases`, con la ridirezione della posta elettronica destinata a questo utente fittizio, verso l'utente `root`.

### 39.17.2.2 Configurazione

La configurazione particolare di Mailman è contenuta in un file denominato `mm_cfg.py`, che potrebbe trovarsi nella directory `/etc/mailman/`. Come suggerisce l'estensione, si tratta di uno script di Python. «

La parte più significativa di questo file riguarda la dichiarazione di alcune variabili, come si vede dall'estratto seguente:

```
#####  
# Put YOUR site-specific configuration below, #  
# in mm_cfg.py . #  
# See Defaults.py for explanations of the values. #  
  
DEFAULT_HOST_NAME = 'dinkel.brot.dg'  
DEFAULT_URL       = 'http://dinkel.brot.dg/cgi-bin/mailman'  
DELIVERED_BY_URL  = '/doc/mailman/images/mailman.jpg'  
  
MAILMAN_OWNER     = 'mailman-owner@%s' % DEFAULT_HOST_NAME  
  
PUBLIC_ARCHIVE_URL = '/pipermail'  
PRIVATE_ARCHIVE_URL = '/mailman/private'  
  
USE_ENVELOPE_SENDER = 0
```

Per prima cosa, si può osservare che i programmi CGI di Mailman dovrebbero essere accessibili a partire da `http://dinkel.brot.dg/cgi-`

*bin/mailman/*; pertanto, il servente HTTP deve risultare configurato per consentire l'accesso in questo modo a tali file. In base all'esempio, si può verificare che ciò sia così provando a interrogare l'indirizzo *http://dinkel.brot.dg/cgi-bin/mailman/admin*, dal quale si deve ottenere una pagina di informazioni sull'amministrazione delle liste.

Come si può intuire dalla configurazione, si definisce che l'amministratore del sistema Mailman si chiama *mailman-owner@...*, pertanto è necessario definire a chi deve corrispondere effettivamente questo indirizzo, intervenendo nel file `/etc/aliases` e avviando successivamente `newaliases` (se necessario). Supponendo che si tratti effettivamente dell'utente `tizio`, potrebbe essere una riga come quella seguente:

```
mailman-owner: tizio
```

Infine, è necessario definire una parola d'ordine per l'amministrazione complessiva. Per questo si usa il programma `mmsitepass`:

```
# mmsitepass [Invio]
```

```
New password: digitazione_all'oscuro [Invio]
```

```
Again to confirm password: digitazione_all'oscuro [Invio]
```

In questo modo, la parola d'ordine viene annotata in modo cifrato per evitare che possa essere individuata facilmente.

### 39.17.2.3 Creazione e cancellazione di una lista



La creazione di una lista di Mailman è guidata dal programma `newlist`, che si usa in pratica come nell'esempio seguente, in cui si crea la lista *prova@...*:

```
# newlist [Invio]
```

Enter the name of the new list: **prova** [*Invio*]

Enter the email of the person running the list: ↵  
↵ **caio@dinkel.brot.dg** [*Invio*]

Initial prova password: **digitazione\_all'oscuro** [*Invio*]

Entry for aliases file:

```
## prova mailing list
## created: 29-Aug-2002 root
prova:          "|/var/lib/mailman/mail/wrapper post prova"
prova-admin:    "|/var/lib/mailman/mail/wrapper mailowner prova"
prova-request: "|/var/lib/mailman/mail/wrapper mailcmd prova"
prova-owner:    prova-admin
```

Hit enter to continue with prova owner notification...

Come si vede dal messaggio che si ottiene, è necessario intervenire poi manualmente nel file `/etc/aliases`, per aggiungere alcune righe. In questo modo, gli indirizzi `prova@...`, `prova-admin@...`, `prova-request@...` e `prova-owner@...` possono poi funzionare regolarmente per la gestione e l'accesso alla lista.

Per eliminare una lista, si procede in modo analogo, con l'aiuto del programma `rmlist`, che se usato con l'opzione `-a`, cancella anche l'archivio dei messaggi:

```
# rmlist -a prova [Invio]
```

Infine, è possibile consultare rapidamente l'elenco degli iscritti a una lista con il comando `list_members`:<sup>17</sup>

```
# list_members prova [Invio]
```

## 39.17.2.4 Amministrazione della lista



Mailman è fatto per essere utilizzato prevalentemente attraverso un navigatore, con il protocollo HTTP. Per verificare l'esistenza della lista appena creata, basta consultare il programma CGI '**admin**' che, secondo la configurazione già vista in precedenza, dovrebbe essere accessibile all'indirizzo *http://dinkel.brot.dg/cgi-bin/mailman/admin*. Ciò che si dovrebbe vedere è rappresentato dal listato seguente:

```
dinkel.brot.dg mailing lists - Admin Links
```

```
Welcome!
```

```
Below is the collection of publicly-advertised mailman mailing lists on dinkel.brot.dg. Click on a list name to visit the configuration pages for that list. To visit the administrators configuration page for an unadvertised list, open a URL similar to this one, but with a '/' and the list name appended.
```

```
General list information can be found at the mailing list overview page.
```

```
(Send questions and comments to mailman-owner@dinkel.brot.dg.)
```

```
List Description
```

```
Prova [no description available]
```

Per configurare meglio la lista *prova@...* è sufficiente seguire il riferimento ipertestuale che si trova in corrispondenza del nome che appare sulla pagina, il quale porta in pratica all'indirizzo *http://dinkel.*

*brot.dg/cgi-bin/mailman/admin/prova*. Come spiega la stessa pagina, se esistono delle liste non pubblicizzate, la loro configurazione si raggiunge in questo modo, mettendo il loro nome dopo quello del programma CGI **'admin'**.

Prova Administrative Authentication

List Administrative Password: \_\_\_\_\_

Let me in...

Important: From this point on, you must have cookies enabled in your browser, otherwise no administrative changes will take effect.

Session cookies are used in Mailman's administrative interface so that you don't need to re-authenticate with every administrative operation.

This cookie will expire automatically when you exit your browser, or you can explicitly expire the cookie by hitting the Logout link under Other Administrative Activities (which you'll see once you successfully log in).

La pagina che si ottiene serve a richiedere l'identificazione dell'amministratore della lista in base alla parola d'ordine, come inserito quando è stato utilizzato il programma **'newlist'**. Superata questa fase si raggiunge la pagina di configurazione vera e propria, che corrisponde però allo stesso indirizzo precedente.<sup>18</sup>

Prova mailing list administration

General Options Section

Configuration Categories	Other Administrative Activities
* General Options	* Tend to pending administrative requests
* Membership Management	
* Privacy Options	* Go to the general list

- \* Regular-member (non-digest) information page
- Options \* Edit the HTML for the public
- \* Digest-member Options list pages
- \* Bounce Options \* Go to list archives
- \* Archival Options \* Logout
- \* Mail-News and News-Mail gateways
- \* Auto-responder

-----

Make your changes below, and then submit them using the button at the bottom. (You can change your password there, too.)

### General Options

Fundamental list characteristics, including descriptive info and basic behaviors.

Description	Value
The public name of this list (make case-changes only). (Details)	Prova_____
The list admin's email address - having multiple admins/addresses (on separate lines) is ok. (Details)	caio@dinkel.brot.dg_____ _____ _____
A terse phrase identifying this list. (Details)	_____
An introductory description - a few paragraphs - about the list. It will be included, as html, at the top of the listinfo page.	_____ _____ _____ _____
Carriage returns will end a paragraph - see the details for more info. (Details)	_____
Prefix for subject line of list postings. (Details)	[Prova]_____
List-specific text prepended to new-subscriber welcome	_____ _____

message (Details) \_\_\_\_\_

Text sent to people leaving the list. If empty, no special text will be added to the unsubscribe message.

(Details)

Where are replies to list messages directed? Poster is strongly recommended for most mailing lists.

Poster  This list  Explicit address

(Details)

Explicit Reply-To: header.

(Details)

(Administrivia filter) Check postings and intercept ones that seem to be administrative requests?

No  Yes

(Details)

Send password reminders to, eg, "-owner" address instead of directly to user.

No  Yes

(Details)

Suffix for use when this list is an umbrella for other lists, according to setting of previous "umbrella\_list" setting.

-owner\_\_\_\_\_

(Details)

Send monthly password reminders or no? Overrides the previous option.

No  Yes

(Details)

Send welcome message when people subscribe? (Details)

No  Yes

Should administrator get immediate notice of new requests, as well as daily

No  Yes

```

    notices about collected
        ones? (Details)
Should administrator get
    notices of (* ) No ( ) Yes
subscribes/unsubscribes?
    (Details)
Send mail to poster when
their posting is held for (* ) Yes ( ) No
    approval? (Details)
Maximum length in Kb of a
message body. Use 0 for no 40_____
    limit. (Details)
Host name this list prefers. dinkel.brot.dg_____
    (Details)
Base URL for Mailman web
interface. The URL must end
in a single "/". See also http://dinkel.brot.dg/cgi-bin/mailman/
the details for an important
warning when changing this
value. (Details)

```

#### To Change The Administrator Password

```

+-----+ +-----+
| Enter current|_____|| Enter new|_____||
| password:| | password:| |
+-----+ |-----+
| | | Confirm new|_____||
| | | password:| |
+-----+ +-----+

```

[ Submit Your Changes ]

Quello che si vede sopra riguarda solo la configurazione generale, mentre sono disponibili altre voci per altre caratteristiche da configurare.

Al termine del lavoro, è bene indicare a Mailman la conclusione dell'attività selezionando la voce **'logout'**.

### 39.17.2.5 Accesso alla lista da parte degli utilizzatori normali

Gli utenti che possono avere interesse a iscriversi a una lista di quelle amministrate devono raggiungere l'indirizzo <http://dinkel.brot.dg/cgi-bin/mailman/listinfo>: «

```
dinkel.brot.dg Mailing Lists
```

```
Welcome!
```

```
Below is a listing of all the public mailing lists on
dinkel.brot.dg.
```

```
Click on a list name to get more information about the list,
or to subscribe, unsubscribe, and change the preferences on
your subscription.
```

```
To visit the info page for an unadvertised list, open a URL
similar to this one, but with a '/' and the list name
appended.
```

```
List administrators, you can visit the list admin overview
page to find the management interface for your list.
```

```
(Send questions or comments to
mailman-owner@dinkel.brot.dg.)
```

List	Description
Prova	[no description available]

Seguendo il riferimento ipertestuale corrispondente al nome della

lista a cui si è interessati, si arriva alla pagina dalla quale ci si può iscrivere:

About Prova

To see the collection of prior postings to the list, visit the Prova Archives.

Using Prova

To post a message to all the list members, send email to prova@dinkel.brot.dg.

You can subscribe to the list, or change your existing subscription, in the sections below.

Subscribing to Prova

Subscribe to Prova by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a public list, which means that the members list is openly available (but we obscure the addresses so they are not easily recognizable by spammers).

Your email \_\_\_\_\_  
address:

You must enter a privacy password. This provides only mild security, but should prevent others from messing with your subscription. Do not use a valuable password as it will occasionally be emailed back to you in cleartext. Once a month, your password will be emailed to you as a reminder.

Pick a \_\_\_\_\_  
password:

Reenter  
password to \_\_\_\_\_  
confirm:

Would you  
like to

```
receive list  (*) No ( ) Yes
mail batched
in a daily
digest?
```

[ [Subscribe](#) ]

Prova Subscribers

Click here for the list of Prova subscribers:

[ [Visit Subscriber list](#) ]

To change your subscription (set options like digest and delivery modes, get a reminder of your password, or unsubscribe from Prova), either enter your subscription email address:

\_\_\_\_\_ [ [Edit Options](#) ]

... or select your entry from the subscribers list  
(see above).

**Chi si iscrive, indicando l'indirizzo di posta elettronica e la parola d'ordine per poter gestire la propria configurazione personale, viene richiesta successivamente una conferma via posta elettronica, simile a questa:**

```
From: prova-request@dinkel.brot.dg
To: daniele@dinkel.brot.dg
Reply-To: prova-request@dinkel.brot.dg
Subject: Prova -- confirmation of subscription -- request 779881
```

Prova -- confirmation of subscription -- request 779881

We have received a request from 192.168.1.1 for subscription of your email address, <daniele@dinkel.brot.dg>, to the prova@dinkel.brot.dg mailing list. To confirm the request, please send a message to prova-request@dinkel.brot.dg, and

either:

- maintain the subject line as is (the reply's additional "Re:" is ok),
- or include the following line - and only the following line - in the message body:

```
confirm 779881
```

(Simply sending a 'reply' to this message should work from most email interfaces, since that usually leaves the subject line in the right form.)

If you do not wish to subscribe to this list, please simply disregard this message. Send questions to [prova-admin@dinkel.brot.dg](mailto:prova-admin@dinkel.brot.dg).

Di solito è sufficiente rispondere a questo messaggio, senza includere il testo precedente per ottenere l'iscrizione. A iscrizione avvenuta si riceve un messaggio di conferma, in cui è annotata la parola d'ordine che è stata definita per la personalizzazione dell'iscrizione alla lista; in seguito si riceve mensilmente un promemoria del genere.

Per accedere alla gestione della configurazione personalizzata, si parte dalla stessa pagina già vista in precedenza, mettendo soltanto il proprio indirizzo di posta elettronica nella parte inferiore:

About Prova

...

...

Prova Subscribers

Click here for the list of Prova subscribers:

[ Visit Subscriber list ]

To change your subscription (set options like digest and delivery modes, get a reminder of your password, or unsubscribe from Prova), either enter your subscription email address:

\_\_\_\_\_ [ Edit Options ]

... or select your entry from the subscribers list (see above).

Da lì si accede a una pagina in cui è possibile richiedere la cancellazione dalla lista o la modifica delle caratteristiche configurabili, con l'inserimento della parola d'ordine personale.

## 39.18 Riferimenti

- Olaf Kirch, *NAG, The Linux Network Administrators' Guide*
- Doug Muth, *The SPAM-L FAQ*, <http://www.dmuth.org/spam-l>
- *Rlytest: test mail host for third-party relay*, <http://www.unicom.com/sw/rlytest>

<sup>1</sup> **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro

<sup>2</sup> La tradizione richiede che l'eseguibile '**sendmail**' sia collocato nella directory '/usr/lib/', ma dal momento che questo fatto va in contrasto con la logica di una gerarchia ordinata del file system, in pratica si tratta solitamente di un collegamento simbolico a un eseguibile che si trova in una posizione più appropriata.

<sup>3</sup> **Mailx** UCB BSD

<sup>4</sup> **Nail** UCB BSD e altre

<sup>5</sup> **Mutt** GNU GPL

<sup>6</sup> **Grepmail** GNU GPL

<sup>7</sup> **IMAP toolkit** software libero con licenza speciale

<sup>8</sup> **Popclient** GNU GPL

<sup>9</sup> È questo punto che può rendere vantaggioso l'utilizzo di Popclient al posto di Fetchmail.

<sup>10</sup> **Fetchmail** GNU GPL

<sup>11</sup> **GNU Sharutils** GNU GPL

<sup>12</sup> L'esempio proviene da un caso accaduto realmente, senza che sia stato possibile chiarire il motivo della composizione errata. Viene proposto questo esempio perché reale, anche se incompleto, considerato il fatto che il mittente e il destinatario sono stati sostituiti, inoltre alcune informazioni sono state eliminate dal messaggio.

<sup>13</sup> **Mpack** software libero con licenza speciale

<sup>14</sup> In realtà la dimensione indicata con questa opzione è solo un riferimento approssimato, dal momento che i messaggi di posta elettronica e di Usenet tendono a espandersi, mano a mano che si aggiungono informazioni sul loro percorso.

<sup>15</sup> **Procmail** GNU GPL o Artistic

<sup>16</sup> **Mailman** GNU GPL

<sup>17</sup> Sono disponibili anche altri comandi, ma in generale è più semplice il controllo attraverso l'interfaccia dei programmi CGI.

<sup>18</sup> Come spiega Mailman stesso, è necessario che il navigatore sia in grado di accettare i *cookie*.