Posta Off-Line HOW-TO

Mirko Caserta, ik0zsn@amsat.org Aprile 1997

Questo documento si prefigge di descrivere come configurare la propria macchina Linux per utilizzare al meglio i servizi di posta elettronica nel caso in cui non si abbia a disposizione una connessione permanente alla rete Internet ma piuttosto ci si debba collegare ad un provider (generalmente via modem in PPP o SLIP tramite rete telefonica commutata) che fornisce i servizi SMTP e POP3. In particolare, lo scopo di questo documento consiste nell'incoraggiare l'utente di Eudora a passare in modo definitivo ad un sistema di messaggistica basato su un software (o una serie di software) che gira sotto Linux (ma non solo visto che quasi tutti i pacchetti qui trattati lavorano anche sotto altri UNIX).

Indice

1 Introduzione

Provate ad indovinare quale è il servizio Internet più utilizzato in tutto il mondo? Si, è proprio la posta elettronica. Effettivamente l'email è di un'utilità spaventosa: permette di scambiare messaggi con persone in tutto il mondo (chiaramente devono avere anche loro accesso ai servizi di email) includendo file e quindi documenti, filmati, immagini, suoni, ecc.

Uno dei motivi per cui ho scritto questo documento risiede nel fatto che quando ho iniziato ad usare Linux, ciò che ancora mi legava a Windows 95 e Eudora (ebbene si, devo confessarlo) era la posta elettronica. Se avessi avuto questa documentazione all'epoca, probabilmente avrei lasciato definitivamente Windows 95 molto prima!

Inoltre ho una memoria pessima per cui tendo a scrivere tutto ciò che andrebbe altrimenti perso, anche le cose più banali e semplici. Quindi perchè non farne una documentazione utile a tutti?

1.1 Prerequisiti

Se vi siete posti il problema di utilizzare la posta elettronica allora probabilmente sapete già di cosa si tratta. Magari siete anche in grado di prelevare le versioni più recenti di tutti i software che fanno al nostro caso da un sito ftp come sunsite.unc.edu o uno dei suoi mirror. O almeno avete un cdrom con una distribuzione di Linux che contiene tali software. O ancora avete su cdrom il sunsite o un altro sito che contiene software a iosa per Linux. Come? Sei sicuro di si? Aspetta... hai installato il supporto per il TCP/IP nel kernel? E quello per il PPP? E la seriale è configurata correttamente? E il modem? E la versione di `pppd` che hai è compatibile con la versione del kernel che usi attualmente? E riesci a collegarti in PPP o SLIP al tuo provider in modo corretto? Come? Davvero? Davvero hai risposto si a tutte le domande? Bene, allora vai pure avanti. In caso contrario consiglio vivamente una visita alla *home page del PLUTO* <http://www.dei.unipd.it/it/linux/pluto> ed alla documentazione disponibile in linea grazie ai collaboratori del PLUTO.

Nel mondo reale le cose non sono così drastiche come le ho descritte qui sopra poichè la distribuzione di Linux che hai installato probabilmente ha già provveduto a risolvere gran parte dei problemi che potresti incontrare. Per chi cerca un'interfaccia user friendly alla configurazione di *pppd* (e non solo) consiglio il pacchetto *linuxconf* che ho avuto modo di provare una volta e mi è sembrato molto ben fatto. Per chi usa la Red Hat invece c'è `netcfg` che permette di configurare una interfaccia PPP in modo simile a *linuxconf* (o viceversa, dipende da chi ha copiato l'altro).

Non pensare che ti voglia scoraggiare con quanto ho detto sopra. Semplicemente non vorrei che considerassi questo documento troppo globale. Qui si parla solo di **posta off-line**. Inoltre ho un pessimo vizio: non scrivo mai riguardo cose che non conosco :) per cui se c'è qualcuno che si vuole fare avanti e spiegarci come usare IMAP, UUCP, ecc si faccia avanti.

Quasi dimenticavo... condizione essenziale perchè tu possa usare la posta elettronica è la conoscenza degli indirizzi dei server POP3 e SMTP del tuo provider.

1.2 Disclaimer

Inutile dire che potrei aver scritto cose inesatte in questo documento (spero di non aver preso delle grosse cantonate! Mica mi avete preso per un wizard?!?) per cui vi prego di non assalirmi ma, piuttosto, scrivetemi cosa correggere, aggiungere, modificare, eliminare, disintegrare, e così via...

1.3 Copyleft

1.4 Altre risorse in rete

Lo scopo di questo documento non è spiegare cosa sia la posta elettronica in generale e come configurare il proprio sistema operativo per usarla al meglio. Ci sono altri documenti che trattano quest'argomento in modo esplicito ed esaustivo. In particolare consiglio:

- *"The Linux Electronic Mail HOWTO"* di Vince Skahan
- *"Queue-R-Mail-HOWTO: Queue Remote Mail + Deliver Local Mail"* di Leif Erlingsson
- *"Linux PPP HOWTO"* di Al Longyear

Il sito ufficiale di questi HOWTO è *sunsite.unc.edu* `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` o uno dei suoi mirror.

Per quanto riguarda la connessione con i provider Italiani c'è il *"Provider Italiani HOWTO"* di Alberto Menegazzi, prelevabile dal *sito web di ILDP* `<http://www.psico.unipd.it/ildp/altri/index.html>`. Presso questo stesso URL troverai questo HOWTO. L'ultima versione del *Posta Off-Line HOWTO* è sempre disponibile all'indirizzo http://www.eurolink.it/˜mirko/po/

Marco d'Itri (md@linux.it), come i migliori sostenitori di Linux, si è immoltato sull'altare sacrificale (che animali sacrificheranno mai 'sti guru di Linux? GNU forse? ...si, lo so che potrei anche risparmiarmele queste battute...) e mi ha autorizzato a pubblicare il suo indirizzo email in modo da poter essere contattato per le problematiche relative ad *uso e configurazione di UUCP*.

1.5 Ringraziamenti

Vorrei ringraziare per la collaborazione, in rigoroso ordine alfabetico:

- Arrigo Benedetti (benedett@dsi.unimo.it)
- Daniele Carbonetti (ftplinux@rtmol.it)
- Ciro Cattuto (ciro@stud.unipg.it)
- Marco D'Itri (md@linux.it)
- Alberto Menegazzi (flash.egon@iol.it)
- Daniele Sanna (dsanna@mbox.vol.it)

Un particolare ringraziamento va a Linus Torvalds, il LDP, l'ILDP, il gruppo PLUTO e l'intera comunità di Linux (ho dimenticato di citare il pianeta Terra e gli abitanti di Marte forse...).

1.6 Glossario

Capisco che molte cose in questo glossario e nel resto del documento possono sembrare ovvie ma, credetemi, non lo sono per l'utente alle prime armi.

Email: Electronic Mail, ovvero *"Posta Elettronica"*.

ISP: Internet Service Provider. Letteralmente *"Fornitore di Servizi Internet"*. In poche parole è quell'organizzazione alla quale dobbiamo sborsare una certa quantità di soldi per farci mantenere la casella postale quando siamo lontani da Internet. Un buon ISP ci da anche la possibilità di spedire la posta con il protocollo SMTP.

Mailbox: la casella postale dove risiedono i messaggi destinati a noi, in attesa di essere prelevati e letti (qualcuno usa anche rispondere di tanto in tanto). La casella postale è in pratica costituita da un file contenente messaggi archiviati e suddivisi in un particolare formato costituito da *header* e *corpo* del messaggio. Gli header forniscono le informazioni tecniche riguardo il messaggio (da chi proviene, a chi è destinato, data e ora, ecc) mentre il corpo del messaggio contiene in genere il testo vero e proprio.

MDA: Mail Delivery Agent, ovvero *"Agente di smistamento della posta"*. Si tratta del software che si occupa di far arrivare al giusto utente i messaggi una volta che sono arrivati sulla macchina di destinazione. Per esempio `procmail` e `deliver` sono dei MDA.

MTA: Mail Transport Agent, ovvero *"Agente di trasporto della posta"*. Si tratta del software che ci permette di trasportare la posta attraverso la rete. Per esempio `sendmail`, `smail`, `qmail`, `exim` sono dei MTA.

MUA: Mail User Agent, ovvero *"Agente di posta dell'utente"*. Niente a che vedere con il SISMI o il SISDE... si tratta semplicemente del software che usiamo per gestire la posta dal lato utente (es: `pine`, `elm`, `XFMail`, `TkRat`, ecc).

POP: Post Office Protocol. Letteralmente *"Protocollo dell'Ufficio Postale"*. Lo si usa per ricevere la posta da un server remoto. I dettagli sono negli RFC 1725, 1225, 1081. Una versione precedente al POP3, il protocollo POP2, è discussa nello RFC 937.

Provider: vedi ISP.

RFC: Request For Comments. Letteralmente *"Richieste di commenti"*. Quando bisogna definire un nuovo standard riguardo i protocolli usati su Internet, qualcuno si prende la briga di scrivere un RFC contenente le specifiche da usare. Se l' RFC viene approvato diventa uno standard.

SMTP: Simple Mail Transfer Protocol. Letteralmente *"Semplice Protocollo per il Trasferimento della Posta"* (andate a farglielo capire a quelli di `sendmail`... mi riferisco alla parola *semplice* naturalmente). Lo usiamo per spedire posta al mondo esterno. I dettagli sono negli RFC 819, 821 e 822.

2 Ricevere la posta

Molti ISP oggi forniscono accesso alla mailbox tramite il protocollo POP3. In questo documento tratteremo unicamente tale protocollo dato che gli altri sono usati meno spesso. Inoltre non ci addentreremo nei dettagli di tale protocollo in quanto non è compito di questo documento farlo.

Volendo descrivere in breve il protocollo POP3 potremmo dire che il server è in costante ascolto sulla porta TCP/IP 110. Nel momento in cui c'è una connessione da parte di un client su tale porta, il server richiede un'autenticazione (generalmente del tipo *nome utente + password* in chiaro), quindi da accesso alla mailbox tramite una serie di comandi che servono principalmente a

- controllare quanti e quali messaggi si trovano nella mailbox
- prelevare i messaggi
- cancellare i messaggi

In genere queste operazioni vengono svolte tutte in modo automatico da un apposito software. Vediamo quali tra questi sono i più usati e come configurarli.

2.1 Popclient

L'ultima versione di popclient è prelevabile via ftp da *sunsite.unc.edu* `<ftp://sunsite.unc.edu/pub/Linux/system/Mail/pop>`. Nel momento in cui scrivo la versione corrente è la 3.0 ed il file relativo si chiama `popclient-3.0.tar.gz`.

Per un'uso base di popclient basta chiamarlo con le seguenti opzioni:

```
popclient -3 -u utente -p password -o /path/per/la/mia/mailbox nomehost
```

Dunque, vediamo un po' di analizzare questa linea di comando:

- `-3` sta ad indicare che usiamo il protocollo POP3;
- `-u` deve essere seguito dal nostro nome-utente (il login);
- `-p` deve essere seguito dalla nostra password sull'host remoto;
- `-o` indica il percorso completo per la nostra mailbox (il file dove si trova la posta sul nostro computer, non quello remoto). Di solito il percorso è `/var/spool/mail/nome-utente` dove `nome-utente` è il nostro login sulla macchina Linux locale;
- `nomehost` deve essere sostituito dal nome del server POP3.

Facciamo un esempio:

Il mio nome-utente è `pippo`, la password `baudo`, la mia mailbox sulla macchina locale si trova sotto `/var/spool/mail/pippo` ed il nome dell'host cui devo collegarmi per prelevare la posta è `katia.rai.it` ... dunque:

```
popclient -3 -u pippo -p baudo -o /var/spool/mail/pippo katia.rai.it
```

`popclient` per default cancella la posta sull'host remoto dopo averla scaricata. Per disabilitare questa funzione aggiungi `-k` tra le opzioni.

Come mi ha fatto giustamente notare Alberto Menegazzi (flash.egon@iol.it), `fetchmail` sostituisce `popclient`, dal quale deriva. E' anche bene fare presente che `fetchmail`, a differenza di `popclient`, ha bisogno di un MDA locale per cui, se abbiamo modificato la configurazione di `sendmail` per l'uso della coda (come spiegato successivamente), ci troveremmo in difficoltà e l'unica soluzione consiste nell'interfacciare `fetchmail` a `procmail` inceve che a `sendmail`. Scusa il giro di parole :)

Per interfacciare direttamente `popclient` a `procmail` dobbiamo usare l'opzione `-c` invece di `-o` per cui un esempio di sintassi corretta potrebbe essere:

```
popclient -3 -u pippo -p baudo -c katia.rai.it | procmail
```

Per ulteriori spiegazioni:

- lanciare `popclient` senza opzioni per avere un elenco delle opzioni passabili da linea di comando;
- `man popclient` per avere una descrizione dettagliata di tutte le funzioni.

2.2 Fetchpop

L'ultima versione di fetchpop è prelevabile via ftp da *sunsite.unc.edu* `<ftp://sunsite.unc.edu/pub/Linux/system/Mail/pop>`. Nel momento in cui scrivo la versione corrente è la 1.9 (ci sono già in giro delle patch per risolvere alcuni bug di questa versione) ed il file relativo si chiama `fetchpop1.9.tar.gz`.

Uno dei vantaggi di fetchpop consiste nella possibilità di essere interfacciato direttamente a procmail (maggiori dettagli su procmail successivamente) usando l'opzione `-p` sulla linea di comando.

Per poter usare `fetchpop` dovremo farlo partire una prima volta senza parametri sulla linea di comando. Ci verranno chiesti nell'ordine: indirizzo del server POP3, login e password sul server POP3, tempo di inattività espresso in secondi che passa prima di controllare nuovamente l'arrivo di nuova posta quando `fetchpop` viene lanciato come daemon (opzione `-d`). All'ultima domanda rispondere con 300 che è il valore minimo specificabile, tanto dal momento che usiamo la posta off-line non useremo mai l'opzione `-d`. Non fate caso all'errore che si verifica dopo aver inserito quest'ultimo parametro: è normale in quanto fetchpop cerca di collegarsi al server POP3 e non ci riesce (a meno che non siamo collegati in quel momento).

A questo punto `fetchpop` si è creato un file nella nostra home directory chiamato `.fetchhost` dove risiedono le informazioni che gli abbiamo dato in questa prima fase. D'ora in poi sarà sufficiente essere collegati alla rete e chiamare `fetchpop` con le opzioni `-r` e `-a` per poter ricevere la nostra posta, inclusi i messaggi eventualmente già letti (opzione `-a`) e, allo stesso tempo, rimuovere i messaggi sul server dopo averli prelevati (opzione `-r`).

Quindi la sintassi corretta sarà:

```
fetchpop -a -r
```

e, nel caso vogliamo usare procmail:

```
fetchpop -a -r -p
```

Per ulteriori spiegazioni:

- lanciare `fetchpop -h` per avere un elenco delle opzioni su linea di comando;
- `man fetchpop` per avere una descrizione dettagliata di tutte le funzioni.

2.3 Fetchmail

Probabilmente in questo momento `fetchmail` è il migliore client in circolazione. Ritengo giusto tradurre e riportare le parti salienti del file README:

```
----------------------------------------------------------------------------
fetchmail è un programma di utilità per il forwarding/prelievo
della posta con POP2, POP3, APOP e IMAP, completo, robusto e ben
documentato, inteso per essere usato su collegamenti TCP/IP su-richiesta
(come connessioni SLIP o PPP). Esso preleva la posta da server remoti e la
invia al sistema di smistamento locale della tua macchina, in modo da
permettere a MUA come elm o Mail di leggerla.

Questa sono le caratteristiche principali di fetchmail. Quelle uniche di
fetchmail sono marcate con **.

*  Supporto per i protocolli **POP2, POP3, **APOP, **IMAP.

** Supporto Kerberos per l'autenticazione dell'utente.

** La macchina viene auto-scandagliata in modo da trovare un server
funzionante se non è stato specificato alcun server per la
connessione. Così non hai bisogno di sapere in anticipo quali tipi di
server stanno girando sulla macchina; l'opzione verbose ti può
mostrare quale sta funzionando.

** Smistamento tramite SMTP alla porta 25 della macchina client. Questo
significa che la posta viene automaticamente inviata all'MDA locale come se
fosse normalmente arrivata dall'esterno via SMTP.

** Timeout se la connessione con il server viene a mancare.
```

```
** Supporto per prelevare e forwardare da caselle postali multiple che
garantisce di non causare loop con la posta.

*  Semplice controllo tramite linea di comando o file di controllo di
esecuzione libero-dal-formato.

*  Modo Daemon -- fetchmail può essere lanciato in background per
richiedere la posta da uno o più server ad un intervallo specificato.

*  Gli header From:, To:, Cc:, e Reply-To: sono riscritti in modo che nomi
utenti relativi alla macchina di fetchmail diventino indirizzi Internet
completamente qualificati (l'originale Inglese rende meglio: fully-qualified
Internet addresses, ndt). Questo fa in modo che le risposte funzionino
correttamente. (Sarebbe stata una funzione unica di fetchmail se non
l'avessi aggiunta a fetchpop).

*  Stretto rispetto degli RFC rilevanti e buone opzioni di debugging.
Potresti usare fetchmail per fare un debug sulle implementazioni di un
server.

*  Pagina di manuale scritta con cura, comprensiva ed aggiornata, la quale
descrive non solo i modi di operazione ma anche (**) come diagnosticare i
problemi più comuni e cosa fare riguardo server deficienti.

*  Codice sorgente a prova di bomba, semplice e ben sperimentato -- l'autore
ne fa uso tutti i giorni e non ha mai perso un messaggio, anche nelle
versioni sperimentali.

*  Larga comunità di utenti -- fetchmail ha ereditato una
significativa base di utenti dalla comunità di popclient, scritto da
Carl Harris. Questo significa che il feedback è rapido e i bachi sono
scovati e corretti rapidamente.

Puoi facilmente prelevare l'ultima versione di fetchmail via FTP da:

        ftp://ftp.ccil.org/pub/esr/fetchmail-1.9.tar.gz

Oppure puoi prelevarla dalla home page dell'autore:

        http://www.ccil.org/~esr
----------------------------------------------------------------------------
```

Bene, spero di averti convinto che `fetchmail` vale la pena di essere usato nei casi in cui un semplice client POP non è sufficiente.

In ogni caso tieni presente il problema già accennato alla fine della sezione relativa a `popclient`: `fetchmail` ha bisogno di un MDA locale per consegnare la posta quindi, se hai configurato `sendmail` per l'uso della coda (opzione defer), dovrai usare `procmail` come MDA. Inoltre alcuni server POP recenti non implementano più il comando LAST (gli utenti di Italia On Line lo sanno bene) per cui per loro si impone l'uso di `fetchmail`.

Se hai scaricato il pacchetto con i sorgenti dall'URL indicato qui sopra, allora probabilmente vorrai installarlo. Niente di più semplice! L'autore di fetchmail è un hacker riconosciuto [1] e, come tale, sa come rendere più facile la vita di quelli che non lo sono :)

[1] attenzione a non confondere la parola hacker con cracker!

Facciamo un `cd /usr/src` e un `tar vxzf /percorso/per/fetchmail-1.9.tar.gz` in modo da ritrovarci il pacchetto originale scompattato sotto `/usr/src/fetchmail-1.9`

Ora entriamo nella directory di `fetchmail` e digitiamo `configure`

Lo script farà un po' di ricerche sulle caratteristiche del nostro sistema e, alla fine, ci riporterà al prompt. Dopo esserci assicurati che il nostro sistema abbia flex versione 2.5.3 o maggiore (è necessario per la compilazione) scrivamo semplicemente `make`

La compilazione dura molto poco (sul mio P60 con 16MB di RAM meno di un minuto). A questo punto basta diventare root e digitare `make install` per installare il programma in `/usr/local/bin` e la pagina di manuale in `/usr/local/man/man1`. Per cambiare queste directory bisogna modificare il Makefile dopo aver lanciato `configure` e prima di aver fatto partire la compilazione con `make`.

Per finire dobbiamo andarci a creare il file `~/.fetchmailrc` dandogli i giusti permessi con:

```
chmod go-rwx,u=rw ~/.fetchmailrc
```

Un esempio di `~/.fetchmailrc` è:

```
poll host_remoto with protocol POP3:
    user tizio there with password secret1 is caio here;
```

In questo modo stiamo dicendo a `fetchmail` di usare tutti i default (leggi la pagina di manuale al riguardo) e collegarsi al server *host_remoto* per prelevare tramite il protocollo POP3 la posta di *tizio*, il quale ha come password *secret1* e sulla macchina locale ha come login *caio*.

Per ulteriori spiegazioni:

- `fetchmail --help` per avere un elenco delle opzioni su linea di comando;
- `man fetchmail` per avere una descrizione dettagliata di tutte le funzioni.

2.4 Smistare la posta in arrivo, ovvero procmail

Nell'ambito della gestione della posta off-line, procmail può rivelarsi di estremo aiuto nel caso in cui il volume di email quotidiano superi il normale (non è raro che si verifichi un caso simile: basta iscriversi ad un paio di mailing list dal traffico intenso).

Procmail può smistare automaticamente la posta nei folder appropriati filtrando in base a qualsiasi parte di un messaggio (intestazione, uno dei campi dell'intestazione, corpo del messaggio, ecc) oppure può inviarla in pasto (ovvero tramite un pipe) ad un altro programma che potrebbe occuparsi ad esempio di archiviare i messaggi secondo un certo criterio e magari inserendo dei campi opportuni per determinati scopi... insomma le possibilità sono davvero infinite, l'unico limite è la fantasia.

Per dire a procmail come smistare la posta andiamo a crearci il file `~/.procmailrc`

Il file `~/.procmailrc` è composto da una serie di regole. Per semplificare le cose, diciamo che ogni regola inizia con una riga contenente `:0` seguita da una o più righe che descrivono una condizione (tali righe iniziano con un asterisco seguito da una espressione regolare estesa compatibile con `egrep`), quindi da un'altra riga che descrive l'azione da compiere se le condizioni sono verificate. Vediamo un esempio pratico:

```
:0
* ^From.*tizio
* ^Subject:.*patagarro
patagarro
```

Se il messaggio viene da *tizio* ed ha come soggetto *patagarro*, allora mettilo nella mailbox *patagarro*. Altro esempio:

```
:0:
* Pluto-meeting
`date +%m-%y`/Pluto-meeting
```

Se gli header del messaggio contengono la parola magica *Pluto-meeting* allora mettili in una folder che ha come nome la data corrente nel formato mese-anno più `/Pluto-meeting`

```
:0
* From.*print-server
| lpr
```

In questo caso i messaggi provenienti da *print-server* vengono inviati direttamente in pasto alla stampante.

Per ulteriori spiegazioni:

- `man procmail` per avere una descrizione dettagliata di tutte le funzioni
- `man procmailrc` per avere una descrizione del formato del file `~/.procmailrc`
- `man procmailex` per avere una serie di esempi da usare nel file `~/.procmailrc`
- `man grep` per avere una descrizione delle espressioni regolari estese compatibili con `egrep`

2.5 Un cenno su IMAP

Come avevo già accennato nell'introduzione, è mia intenzione parlare per il momento solo di POP3. Ma voglio fare una piccola eccezione e farvi leggere questo articolo molto interessante scritto da Luca Polo e da me pescato su `it.comp.linux`:

```
----------------------------------------------------------------------------
> Mi avete incuriosito :-) Cos'è l'IMAP 4?

Diamine, ma è soltanto il buon vecchio Internet Message Access
Protocol versione 4!! :-)

Praticamente un super-superset di POP, ma più orientato verso un
effettivo client-server (con POP si ha un download, punto), supporto
multi-client (io leggo la posta da più macchine in posti diversi),
multi-server, ecc.

Un altro punto di forza sta nel fatto che è il server IMAP a gestire
MIME & Co. (client molto più semplici, e inoltre c'è solo
una macchina da tenere aggiornata); inoltre, mediante il protocollo
ACAP in fase di sviluppo alla Carnegie Mellon, anche i file di
configurazione (personalizzazioni, bookmark, alias) possono risiedere
sul server, così anche loro risultano indipendenti dalla singola
macchina.

La pacchia degli amministratori di sistema, insomma... :-P

Poi... beh, guardatevi http://www.imap.org/

Lì c'è tutto, compresi tutti i client conosciuti o in fase di
```

```
sviluppo; poi ditemi se non vale la pena tenerlo d'occhio... BTW, i server
IMAP di mia conoscenza sono anche POP server (alcuni dicono che siano
perfino molto meglio del "classico" qpopper).

Saluti,
Luca Polo
--
 / Luca Polo   : jake@gest.unipd.it    || System administrator          \
| (http://www.gest.unipd.it/~jake for  || Ist. di Ingegneria Gestionale  |
 \ address and phone numbers)          || Universita' di Padova, Italy  /
-----------------------------------------------------------------------------
```

3 Leggere la posta (ovvero i MUA)

3.1 Pine

Uno dei problemi più frequenti con `pine` nell'uso off-line è il fatto che esso mostra nel campo `From:` dei messaggi che spediamo il nostro nome e cognome (andandoli a prendere da `/etc/passwd` credo) seguito da un indirizzo email costituito dal nostro login sulla macchina locale più il nome dell'host locale. Per esempio se il mio login sulla mia macchina Linux a casa è `mirko` ed il nome della macchina è `blackhole`, `pine` mostrerà nel campo `From:` una cosa del genere:

```
From:  Mirko Caserta <mirko@blackhole>
```

Ovviamente un tale indirizzo su Internet non ha un granchè come significato perchè `blackhole` non è un nome di dominio registrato. A questo punto ci sono due soluzioni: una legata a `sendmail` e al relativo uso di un database di utenti locali (a tal proposito vedi la sezione *Un database di utenti locali: perchè e come*), ed un'altra di uso piu' generale e slegata da `sendmail`. L'ultima soluzione, anche se non molto igienica in termini di sicurezza (ma che tuttavia potrebbe risultare indispensabile in alcuni casi), consiste nel ricompilare `pine` con l'opzione per poter modificare l'header `From:`

Niente di impossibile anche per l'utente appena giunto nel mondo di Linux. Cominciamo andando a prendere l'ultima versione di Pine da uno di questi URL:

- `ftp://ftp.cac.washington.edu/pine`
- `http://www.cac.washington.edu/pine`

Al momento in cui scrivo, la versione corrente è la 3.95 ed il nome del relativo file è `pine395.tgz`

Portiamoci nella directory `/usr/src` e scompattiamo il file appena prelevato con:

```
tar vxzf pine395.tgz
```

Quindi andiamo nella directory `/usr/src/pine3.95/pine/osdep` e modifichiamo il file `os-lnx.h`

Alla riga 71 dovremmo trovare:

```
/* #define ALLOW_CHANGING_FROM /* comment out to not allow changing From */
```

Modifichiamola in modo che diventi:

```
#define ALLOW_CHANGING_FROM /* comment out to not allow changing From */
```

`#define` deve trovarsi incollato al margine sinistro dello schermo, altrimenti non funzionerà. Ora facciamo un `cd ../..` in modo da tornare in `/usr/src/pine3.95` e diamo il comando

```
build lnx
```

Dopo un bel po' di messaggi strani circa la compilazione, il prompt dovrebbe fare capolino ed informarci che tutto è andato a buon fine (beh, almeno sulla mia macchina Linux 2.0.12 con librerie aggiornate, ecc, è andato tutto ok senza fare nemmeno una piega).

Sotto `/usr/src/pine3.95/bin` dovremmo trovare la nostra versione di `pine`. Copiamola al posto della vecchia versione (di solito `/usr/bin/pine`) senza dimenticare di aver prima fatto una copia di backup (non si sa mai... hai presente una certa legge di Murphy?).

Volendo possiamo anche installarci la versione appena compilata di `pico` (l'editor di testi interno di `pine`) sempre sotto `/usr/bin/pico`

Ora facciamo partire `pine` e, dal menù principale, scegliamo **S** (Setup), quindi **C** (Config). Scorriamo l'elenco delle opzioni in giù fino a trovare `Customized-hdrs`. Il valore corrente dovrebbe essere il default, quindi `No Value Set`.

Usiamo **A** (Add Value) per aggiungere un valore, quindi specifichiamo:

```
From: Nome Cognome <nome.utente@mio.provider.it>
```

ovviamente sostituendo il nostro nome, cognome ed indirizzo di posta elettronica a quelli dell'esempio. Fine!! Ce l'abbiamo fatta!

Per fare le cose in modo più puntiglioso potremmo anche aggiungere sempre con il comando **A** (Add Value) al campo `Customized-hdrs` i valori:

```
Reply-to: Nome Cognome <nome.utente@mio.provider.it>
Return-Path: Nome Cognome <nome.utente@mio.provider.it>
```

anche se personalmente sconsiglio vivamente di specificare i campi `Reply-to:` e `Return-Path:` a meno che non ci sia una **reale necessità**.

3.2 XFMail

XFMail è probabilmente il MUA più interessante in questo momento per chi lavora sotto X in quanto ha un'interfaccia utente molto user-friendly, supporta i protocolli POP3 ed IMAP e ha un sistema di filtraggio della posta tipo procmail incorporato. Fino a qualche tempo fa la grave lacuna di questo MUA era la mancanza di un editor decente, mancanza che è stata colmata nella versione 1.0.

L'URL di XFMail è `http://burka.netvision.net.il/xfmail/xfmail.html`

Dall'URL sopraindicato puoi prelevare sia i sorgenti di XFMail (avrai bisogno dell'ultima versione della libreria XForms per compilare), sia alcune versioni precompilate per Linux (sia per librerie statiche che dinamiche). L'ultima versione al momento in cui scrivo è la 1.0 ma gli autori ci tengono a far presente che il numero della versione non significa che si tratta di qualcosa di stabile.

Da notare l'opzione *Send offline* presente nella configurazione (menù *Misc* -> *Config* -> *Send*) che permette di mantenere i messaggi da spedire nella cartella *outbox*. Quando saremo collegati in rete basterà scegliere *Send all* dal menù *Send*. In questo modo possiamo anche evitarci di configurare `sendmail` dal momento che XFMail ci permette di inviare la posta direttamente al server SMTP del provider.

3.3 TkRat

Recentemente mi sono imbattuto in un altro MUA per X molto interessante: `TkRat`. Attualmente è alla versione 1.0 ma dispone già di numerose funzioni molto interessanti. Non mancano le opzioni per poter lavorare comodamente off-line senza la necessità di dover configurare appositamente `sendmail` (dal menù *Amministrazione* scegli *Preferenze*, quindi seleziona la sezione *Spedizione* e alla voce *Modo consegna* imposta *Differita*). Non dimenticare di impostare nella configurazione di TkRat il tuo indirizzo email reale per fare in modo che le risposte vengano spedite al giusto indirizzo (dal menù *Amministrazione* scegli *Preferenze*, quindi seleziona la sezione *Componi messaggio* e alla voce *Rispondi-A Standard* imposta il tuo indirizzo email reale).

Altro punto a vantaggio di `TkRat` è l'interfaccia utente completamente in Italiano (c'è anche l'help in linea in italiano!). Compilare i sorgenti ed installare l'intero pacchetto è un gioco da ragazzi grazie ad `autoconf`.

L'URL consigliato dall'autore del software per prelevare l'ultima versione di `TkRat` è http://www.dtek.chalmers.se/ maf/ratatosk/

Presso questo stesso URL troverai l'elenco delle caratteristiche più interessanti di `TkRat`.

4 Spedire la posta, ovvero sendmail

Beh, a dire il vero ci sono diversi MTA, tra cui `smail`, `qmail`, ed altri ma, non avendo mai avuto modo di provarli, mi soffermerò solo su `sendmail`.

`sendmail` è forse uno dei software più complicati da configurare nella nostra galassia; in compenso ci permette di avere un sistema di gestione della posta all'altezza di qualsiasi situazione, tanto che è spesso sprecato in molti casi per i quali basterebbe un ben più semplice MTA. In ogni caso, per una configurazione base di `sendmail` ci viene in aiuto `m4` che, con le sue macro, ci permette di creare in maniera estremamente semplice un file di configurazione adatto al nostro caso specifico.

Nota bene: `sendmail` è un software che avrà sempre dei buchi (bug o bacarozzi che dir si voglia) riguardanti la sicurezza, per cui consiglio vivamente di prelevare l'ultima versione disponibile (anche perchè tutte le prove sono state fatte sulla versione 8.8.3 e non posso assicurare che quanto è qui descritto funzioni anche con le versioni precedenti) da uno di questi URL:

- `www.sendmail.org`
- `ftp.sendmail.org` sotto la directory `/pub/sendmail/`

4.1 Creazione del file sendmail.cf con m4

Per far capire a `sendmail` che operiamo off-line, dobbiamo andare a modificare il file di configurazione che di solito si chiama `/etc/sendmail.cf` Dal momento che non avrebbe molto senso andare a modificare a mano tale file, vediamo come possiamo piuttosto generarne uno nuovo usando `m4` (ovviamente devi avere `m4` installato e funzionante).

Come prima cosa andiamo a prendere via `ftp` l'ultima versione di `sendmail` da `ftp://ftp.sendmail.org/pub/sendmail/` quindi andiamo a scompattare il file appena prelevato sotto `/usr/src` con il comando

```
tar vxzf /percorso/per/sendmail.x.x.x.tar.gz
```

A questo punto, per semplificarci la vita, creiamo un link simbolico in modo da far risultare la directory di `sendmail` come `/usr/src/sendmail` in questo modo:

```
ln -s /usr/src/sendmail-x.x.x /usr/src/sendmail
```

Ovviamente le `x` usate nel path di `sendmail` stanno ad indicare il numero della versione e dovranno essere sostituite!!!

Ora andiamo nella directory `/usr/src/sendmail/cf/cf` e creiamoci il file con le indicazioni per `m4` chiamandolo `linux-offline.mc` ed avente come contenuto quanto segue:

```
include(`../m4/cf.m4')
VERSIONID(`linux per uso off-line')dnl
OSTYPE(linux)
FEATURE(nouucp)dnl
FEATURE(always_add_domain)dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(confDELIVERY_MODE, defer)
define(`SMART_HOST', mio_smtp_host)
define(confUSERDB_SPEC, /etc/userdb.db)
FEATURE(notsticky)
```

Le due linee che contengono `define(confUSERDB_SPEC, /etc/userdb.db)` e `FEATURE(notsticky)` devono essere inserite solo se vogliamo usare il database di utenti locali (vedi la sezione *Un database di utenti locali: perchè e come*). Inoltre `mio_smtp_host` deve essere sostituito con il nome del server SMTP del provider. Ti faccio anche notare che quell'apostrofo al contrario che si trova ad esempio prima di `linux per uso off-line')dnl` è molto importante: è diverso dal semplice apostrofo e corrisponde in ASCII al codice 96 (decimale).

Ora facciamo una copia di riserva del file `/etc/sendmail.cf` e generiamone uno nuovo con il comando:

```
m4 linux-offline.mc > /etc/sendmail.cf
```

4.2 Creazione del file sendmail.cf e /etc/aliases con make

Per i più esperti, ecco in regalo un `Makefile` da mettere in `/etc/mail` per compilare automaticamente `sendmail.cf` e gli alias. Prima però bisogna togliere da sendmail.mc la riga con l'include(), quindi si deve creare un link simbolico con il comando:

```
ln -s /etc/sendmail.cf /etc/mail/sendmail.cf
```

Segue il contenuto del file `/etc/mail/Makefile`:

```
------------------------ taglia qui ------------------------
M4LIB=/usr/lib/sendmail.cf HOSTNAME=wonderland

all: cfg

sendmail.cf: $(HOSTNAME).mc m4 $(M4LIB)/m4/cf.m4 $(HOSTNAME).mc >
        sendmail.cf
# Queste righe sono un esempio di come applicare automaticamente delle patch
#       patch --silent < $(M4LIB)/smartdom.diff patch --silent <
#       $(M4LIB)/selective-masq.diff rm -rf sendmail.cf.orig

/etc/aliases: /etc/aliases.db

/etc/aliases.db: /etc/aliases sendmail -bi
```

```
cfg: sendmail.cf /etc/aliases

test: cfg sendmail -bt -C./sendmail.cf #-oQ/tmp/mqueue

clean: rm sendmail.cf /etc/aliases.db
----------------------- taglia qui -----------------------
```

4.3 Ultimi ritocchi a sendmail

Una volta generato il file `sendmail.cf` facciamo ripartire `sendmail` ed il gioco è fatto... o quasi :)

In genere la distribuzione di Linux che abbiamo installato fa in modo che al boot della macchina, da uno degli script nella directory `/etc/rc.d`, parta `sendmail`. Ora, dal momento che quelli che creano le distribuzioni di Linux danno per assunto che ognuno di noi sia collegato in rete con una T1 da casa, fanno partire `sendmail` per default con l'opzione `-q`, la quale dice a `sendmail` di processare immediatamente la coda dei messaggi in uscita oltre che ad certo intervallo di tempo.

Per evitare che ciò succeda, individuiamo in quale file viene fatto partire `sendmail` ed eliminiamo l'opzione -q (che generalmente è seguita anche dall'intervallo di tempo per processare la coda, per esempio 15m sta per 15 minuti). Per individuare il file, portiamoci nella directory `/etc/rc.d` e facciamo una

```
grep sendmail *
```

Nella Red Hat, la directory in questione è `/etc/rc.d/init.d` ed il file si chiama `sendmail.init`

A questo punto ogni messaggio che inviamo a `sendmail`, sia direttamente, sia via SMTP sulla porta locale, viene messo in una coda nella directory `/var/spool/mqueue` e la coda verrà processata (i messaggi verranno inviati) solo con il comando

```
sendmail -q -v
```

Per vedere il contenuto della coda digita `mailq`

L'opzione `-v` serve a dire a `sendmail` di visualizzare cosa combina durante l'invio dei messaggi, mentre `-q` serve proprio ad indicargli di processare la coda. Se vogliamo avere un log di cosa combina `sendmail` possiamo farlo partire in questo modo:

```
sendmail -q -v >> /var/log/sendmail
```

Un'ultima cosa: per compilare `sendmail` ed installarlo al posto della versione attuale:

```
cd /usr/src/sendmail/src
makesendmail
makesendmail install
```

4.4 Un database di utenti locali: perchè e come

Come abbiamo già visto con `pine`, uno dei problemi più ricorrenti nell'uso della posta off-line consiste nel fatto che il campo `From:` automaticamente generato dal nostro MUA non corrisponde al nostro indirizzo Internet reale. Per ovviare a questo problema basterebbe inserire un header del tipo `Reply-to:` con il nostro indirizzo effettivo, ma chi riceverà la nostra posta continuerà a vedere nel campo `From:` un indirizzo sbagliato che, se un nostro amico memorizza in un elenco credendolo corretto, sarebbe semplicemente inutile e ci farebbe anche perdere un sacco di tempo e messaggi.

La soluzione consiste nel creare un database di utenti locali in cui ad una chiave consistente in un certo login corrisponde un indirizzo email reale. Per fare ciò dobbiamo avere precedentemente specificato nel file per `m4` (vedi la sezione *Creazione del file `/etc/sendmail.cf` con m4*) le righe:

```
define(confUSERDB_SPEC, /etc/userdb.db)
FEATURE(notsticky)
```

Inoltre dobbiamo avere installato il pacchetto *db* di Berkeley dal momento che questo sistema non funziona con DBM. Puoi prelevare i sorgenti da ftp://ftp.cs.berkeley.edu/pub/4bsd/db.tar.gz ma non ne avrai bisogno se stai utilizzando una distribuzione abbastanza recente come ad esempio la Red Hat 4.0.

Ora andiamo ad editare il file `/etc/userdb` in questo modo:

```
login:mailname  nome.utente@mio.provider.it
nome.utente@mio.provider.it:maildrop    login
```

Si tratta di sostituire *login* con il nostro login sulla macchina locale e *nome.utente@mio.provider.it* con il nostro indirizzo Internet reale. Quindi un esempio concreto potrebbe essere:

```
mirko:mailname  ik0zsn@amsat.org
ik0zsn@amsat.org:maildrop       mirko
```

Usa TAB per separare i campi. Ora generiamo il database con:

```
makemap btree /etc/userdb.db < /etc/userdb
```

Facciamo ripartire `sendmail` ed il gioco è fatto. Ora, ad esempio con Berkeley's Mail, i messaggi in uscita avranno l'indirizzo corretto nell'header `From:` e `Return-Path:`

Come avrai già notato, con `pine` invece non è cambiato un bel niente e si ostina a indicare un indirizzo scorretto. La soluzione viene dalla documentazione di `sendmail` (dalle FAQ per essere più esatti) e la riporto esattamente così com'è, limitandomi a tradurla.

```
======================================================================
Data: 19 Luglio 1996
Soggetto:  Q3.6 -- Come posso far funzionare il database di utenti
                   locali con Pine o con FEATURE(always_add_domain)?

    L'incompatibilita` di base tra Pine e il database di utenti
risiede in come Pine scrive il tuo indirizzo nella intestazione dei
messaggi. Molti MUA scrivono il tuo indirizzo come "From: user",
mentre Pine, per ragioni date nella sua documentazione, scrive
l'indirizzo come "From: user@FQDN" (FQDN=fully qualified domain name,
ovvero il nome di dominio completo di una macchina su
Internet). Usando la macro di m4 "always_add_domain" si ha lo stesso
effetto. Data questa differenza, il database di utenti locali non
riscrive queste intestazioni.

    Una soluzione a questo problema consiste nell'apportare la
seguente modifica nel file sendmail.mc compilato da m4 nel tuo
/etc/sendmail.cf (oppure ovunque il tuo file sendmail.cf risiede) dopo
che hai installato il database di utenti locali e l'hai fatto
funzionare con altri MUA:

    All'inizio della sezione dove imposti le variabili di
```

```
configurazione, aggiungi quanto segue:

        # Define our userdb file for FQDN rewrites
        Kuserdb btree -o /etc/userdb.db

    Ed un po' piu` in seguito, prima delle righe "MAILER()", ma dopo
che altre opzioni di configurazione siano state specificate:

   LOCAL_RULE_1
   ########################################################
   ### Local Ruleset 1, rewrite sender header & envelope ##
   ########################################################
   #Thanks to Bjart Kvarme <bjart.kvarme@usit.uio.no>
   S1
   R$-                   $1 < @ $j . >              user => user@localhost
   R$- < @ $=w . > $*    $: $1 < @ $2 . > $3 ?? $1  user@localhost ?
   R$+ ?? $+             $: $1 ?? $(userdb $2 : mailname $: @ $)
   R$+ ?? @              $@ $1                      Not found
   R$+ ?? $+             $>3 $2                     Found, rewrite

   # NOTA BENE   ^^^^^^^^^^^^^^^      ^^^^^^^^^^^^^^^^^^^^^^^^^
   # Usa il tasto Tab in queste regioni in modo da avere tre
   # colonne (la linea con "mailname" ha solo 2 colonne).

    Ora il database di utenti dovrebbe riscrivere i messaggi spediti
con Pine o qualsiasi altro MUA che vuole avere un indirizzo
completamente qualificato (FQDN). Se con il metodo appena descritto
non hai ancora risolto il problema, prova ad aggiungere quanto segue
sia al file di configurazione di sistema pine.conf, pine.conf.fixed, o
al tuo file di configurazione personale .pinerc:

        user-domain=localhost

    Sappiamo che questo ha risolto il problema a molte persone.

    Ad ogni modo, una soluzione piu` elegante (leggi: basata su m4)
per la versione 8 di sendmail deve essere ancora creata.
======================================================================
```

4.5 Makemap non supporta il tipo btree. E adesso?

Potrebbe succedere che `makemap` ti dice di non supportare il tipo *btree* rifiutandosi di generare il database. Cosa fare in questo caso? Soluzione: ricompilare `makemap` con il supporto per i database di tipo *btree*.

I sorgenti di `makemap` sono nella distribuzione di `sendmail`, quindi se abbiamo seguito le istruzioni precedenti dovremmo ritrovarci il tutto sotto `/usr/src/sendmail/makemap`

Per poter compilare `makemap` con il supporto per il tipo *btree* dobbiamo avere installato il pacchetto `db` di Berkeley (vedi la sezione *Un database di utenti locali: perchè e come*).

L'unica difficoltà nel compilare `makemap` consiste nel far funzionare il makefile. Dal momento che su ogni sistema le cose potrebbero cambiare (librerie diverse, ecc) ti riporto il makefile che ho usato con successo (per dovere di cronaca al momento delle prove avevo la Red Hat 4.0). Chiamalo semplicemente `Makefile` dopo avere rinominato quello già presente nella directory e lancia `make` quindi, se la compilazione è andata a buon fine, fai qualche prova con il tuo nuovo `makemap` ed infine installalo con `make install`

```
------------------------ taglia qui ------------------------
```

```
O=      -O
SRCDIR= ../src
DBMDEF= -DNDBM -DNEWDB
ENVDEF=
INCDIRS=-I${SRCDIR} -I/usr/include
LDOPTS=
LIBDIRS=-L/usr/lib
LIBS=   -ldb -lgdbm
BINDIR= /usr/sbin
OBJADD=

############  Non modificare al di sotto di questa linea  ##############

CFLAGS= -I. $O ${INCDIRS} ${DBMDEF} ${ENVDEF}
OBJS=   makemap.o ${OBJADD}
LINKS=  ${DESTDIR}/usr/ucb/newaliases ${DESTDIR}/usr/ucb/mailq
BINOWN= bin
BINGRP= bin
BINMODE=555

ALL=    makemap makemap.0

all: ${ALL}

makemap: ${BEFORE} ${OBJS}
        ${CC} -o makemap ${LDOPTS} ${OBJS} ${LIBDIRS} ${LIBS}

NROFF=  groff -Tascii
MANDOC= -mandoc

makemap.0: makemap.8
        ${NROFF} ${MANDOC} makemap.8 > makemap.0

install: install-makemap install-docs

install-makemap: makemap
        install -o ${BINOWN} -g ${BINGRP} -m ${BINMODE} makemap ${BINDIR}

install-docs: makemap.0

clean:
        rm -f ${OBJS} makemap makemap.0

${OBJS}: ${SRCDIR}/conf.h
------------------------ taglia qui ------------------------
```