

Sendmail: introduzione

Destinatari e formati degli indirizzi	279
Alias, inclusione e forward	280
Configurazione di Sendmail con il pacchetto di Berkeley	281
Introduzione al sistema	281
Struttura e contenuto del file di configurazione	282
Macro «VERSIONID»	282
Macro «OSTYPE»	283
Macro «DOMAINS»	283
Macro «MAILERS»	283
Macro «FEATURE»	284
Macro «HACK»	284
Esempio di una distribuzione GNU/Linux	284
File «/etc/mail/ip_allow»	285
File «/etc/mail/name_allow»	285
File «/etc/mail/relay_allow»	285
File «/etc/mail/deny»	286



software non libero: non è consentita la commercializzazione a scopo di lucro

Sendmail ¹ è divenuto lo standard per quanto riguarda i programmi di gestione della posta elettronica in qualità di MTA. La sua adattabilità e la conseguente difficoltà nella definizione della sua configurazione, sono estreme.

Nel capitolo ??capitolo email?? si è già accennato al funzionamento di Sendmail. Questo capitolo espande un po' i concetti, ma si tratta sempre di informazioni limitate; il documento di riferimento per questo resta: *Sendmail* edito da O'Reilly.

Destinatari e formati degli indirizzi

Sendmail, per quanto riguarda la composizione degli indirizzi di posta elettronica, utilizza le convenzioni seguenti.

- Ciò che appare tra parentesi viene eliminato, perché considerato un commento.
- Ciò che appare tra parentesi angolari (<>) viene preferito rispetto a ogni altra indicazione. In pratica, ciò permette di comporre gli indirizzi inserendo anche il nome effettivo del mittente o del destinatario, evidenziando l'indirizzo di posta elettronica vero e proprio all'interno delle parentesi angolari. Per esempio, **'Tizio Tizi <tizio@dinkel.brot.dg>'** è un modo formalmente corretto per abbinare all'indirizzo *tizio@dinkel.brot.dg* il nome e cognome dell'utente: Tizio Tizi.
- Gli apici doppi permettono di delimitare una stringa. In questo modo, alle volte si delimita il nominativo dell'utente, come nell'esempio seguente:
'"Tizio Tizi" <tizio@dinkel.brot.dg>'
Nello stesso modo, la barra obliqua inversa ('\') può essere usata per proteggere il carattere successivo.

Per Sendmail, il destinatario di un messaggio di posta elettronica può essere anche un file o un programma. In pratica, se l'indirizzo utilizzato inizia con una barra verticale ('|'), si intende trattarsi di un condotto, all'interno del quale deve essere inviato il messaggio; se invece l'indirizzo inizia con una barra obliqua normale ('/'), si intende trattarsi di un file, che viene creato appositamente oppure gli viene aggiunto il testo del messaggio se esiste già.

L'utilizzo di questi indirizzi speciali, riferiti a file o a condotti, può essere fatto ovunque; per esempio nel file '~/.forward' o come

destinatario di un alias nel file `/etc/aliases`. Queste possibilità, tra le altre cose, sono alla base del funzionamento delle liste di posta elettronica (*mailing-list*).

Alias, inclusione e forward

All'interno di un sistema è possibile definire dei recapiti fittizi, definiti alias. La predisposizione di questi viene fatta nel file `/etc/aliases`, ma prima che questi abbinamenti siano recepiti da Sendmail, occorre rigenerare il file `/etc/aliases.db` con il comando `newaliases`. Attraverso gli alias è possibile:

- definire dei nominativi utenti lunghi e articolati, che non sarebbero ammissibili nella gestione normale di un sistema Unix (il quale pone generalmente il limite degli otto caratteri di lunghezza per i nomi degli utenti),
- definire dei nominativi di utenti standard riferiti a determinate competenze amministrative tipiche, girando i messaggi loro rivolti alle persone che ricoprono effettivamente gli incarichi corrispondenti;
- definire un elenco di destinatari differenti a cui deve essere inviata una copia dei messaggi riferiti a un certo alias;
- definire un condotto contenente un comando che deve occuparsi di elaborare i messaggi riferiti a un certo alias;
- definire un file per l'archiviazione dei messaggi indirizzati a un certo alias;
- definire un elenco di destinatari differenti in base a un elenco di indirizzi contenuto in un file esterno.

L'esempio seguente fa in modo che i messaggi inviati all'utente fittizio `Tizio.Tizi` siano girati al nome dell'utente gestito effettivamente nel sistema.

```
Tizio.Tizi: tizio
```

L'esempio seguente riguarda la situazione tipica in cui i messaggi indirizzati a un utente fittizio riferito a una competenza amministrativa vengono girati all'utente reale che svolge quel compito particolare.

```
postmaster: danielle
```

L'esempio seguente mostra un alias per il quale i messaggi vengono rinviati (vengono fatti proseguire, o meglio, secondo la tradizione postale, vengono proseguiti) e duplicati per una serie di utenti che devono essere informati contemporaneamente.

```
abuse: danielle, tizio, caio@roggen.brot.dg
```

L'esempio seguente mostra un alias per il quale tutti i messaggi vengono elaborati da un comando, che li riceve attraverso lo standard input. Questo è il modo tipico attraverso cui si inviano i messaggi a un programma di gestione di una lista di posta elettronica (*mailing-list*).

```
lista-pippo: *|/home/liste/bin/ricezione-messaggi lista-pippo
```

L'inclusione è un modo di definire un alias dinamico, riferito a un elenco di indirizzi contenuti in un file di testo normale. La forma

```
:include:percorso_assoluto
```

equivale a includere tutti gli indirizzi definiti nel file specificato, che deve comprendere necessariamente il percorso assoluto per raggiungerlo. Utilizzando questa forma di definizione degli elenchi di destinatari, si evita di dover modificare ogni volta il file `/etc/aliases`, ma soprattutto si evita di dover rieseguire il comando `newaliases`.

L'esempio seguente invia i messaggi destinati all'utente fittizio `lista-pippo-inv` a tutto l'elenco contenuto nel file `/home/liste/pippo/iscritti`.

```
lista-pippo-inv: :include:/home/liste/pippo/iscritti
```

Il *forward* è la gestione di un alias personale (allo scopo di fare proseguire i messaggi verso altre destinazioni), che ogni utente può definire senza dover chiedere la modifica del file `/etc/aliases`. Si

possono fare proseguire i messaggi generando il file di testo `~/forward` che può contenere uno o più indirizzi differenti, compresi i condotti, i file e le inclusioni. Il risultato che si ottiene è che i messaggi destinati all'utente che ha predisposto questo file nella propria directory personale, vengono rinviati a tutti gli indirizzi contenuti nel file stesso. Generalmente, per la sua natura, il file `~/forward` viene usato dagli utenti che hanno diversi recapiti e vogliono concentrare la posta elettronica in un solo punto di destinazione. Per questo motivo, nel file `~/forward` viene indicato quasi sempre un solo indirizzo di posta elettronica.

Configurazione di Sendmail con il pacchetto di Berkeley

Si è già accennato al fatto che la configurazione di Sendmail, attraverso la modifica diretta del file `/etc/sendmail.cf`, sia un'impresa estrema. Fortunatamente, per alleviare queste difficoltà, si sono sviluppati nel tempo diversi programmi in grado di generare automaticamente il file `/etc/sendmail.cf` utilizzando dei segmenti di codice già pronto da combinare opportunamente assieme.

Attualmente, il tipo di configurazione più diffuso è quello predisposto dall'università di Berkeley. Si tratta di una serie di file macro per M4, un macro-compilatore concettualmente analogo al precompilatore del linguaggio C.

Il pacchetto viene installato da qualche parte, a seconda dell'organizzazione predisposta dalla propria distribuzione GNU, ma probabilmente si tratta della directory `/usr/lib/sendmail-cf/`. Da quella directory se ne diramano altre contenenti i diversi pezzi di configurazione che possono essere combinati assieme.

Introduzione al sistema

A partire dalla directory di origine del pacchetto di configurazione di Sendmail, si trovano in particolare i file *readme* che rappresentano tutta la documentazione disponibile, oltre a una serie di directory contenenti a loro volta i file componenti del sistema di macro.

Directory	Descrizione
<code>'m4/'</code>	Contiene alcune macro di partenza, di cui, la più importante è <code>'cf.m4'</code> che viene usata per iniziare il procedimento.
<code>'cf/'</code>	Contiene i file di configurazione utilizzati dalla macro <code>'m4/cf.m4'</code> ; questi file hanno l'estensione <code>'mc'</code> . Di questi file ne viene usato solo uno: quello predisposto per il proprio sistema. È molto probabile che la propria distribuzione GNU inserisca il file utilizzato effettivamente per ottenere la configurazione di Sendmail che questa utilizza; potrebbe trattarsi di <code>'linux.mc'</code> oppure di un altro nome che ricorda quello della distribuzione (per esempio <code>'redhat.mc'</code>).
<code>'sh/'</code>	Contiene degli script di shell utilizzati automaticamente da M4, in base alle istruzioni contenute nelle macro utilizzate. Le altre directory che discendono dall'origine del pacchetto di configurazione, sono utilizzate per classificare i vari file macro incorporabili in quello che si scrive all'interno della directory <code>'cf/'</code> . Per esempio, un'istruzione del tipo <code>'MAILER(local)'</code> , fa riferimento al file <code>'mailer/local.m4'</code> .
altre directory	

Quando si predispose un file di configurazione nella directory `'cf/'`, la sua compilazione avviene nel modo seguente:

```
m4 ../m4/cf.m4 file_di_configurazione > file-risultato
```

Per esempio, supponendo di avere realizzato il file di configurazione `'cf/prova.mc'` e di voler generare il file `'cf/prova.cf'`, si procede come segue:

```
# cd /usr/lib/sendmail-cf [Invio]
```

In questo modo ci si posiziona nella directory principale del pacchetto di configurazione.

```
# cd cf [Invio]
```

Prima di iniziare la compilazione occorre posizionarsi nella directory contenente il file di configurazione.

```
# m4 ../m4/cf.m4 prova.mc > prova.cf [Invio]
```

A questo punto il file `cf/prova.cf` è stato generato, quindi è sufficiente cambiargli nome e sostituirlo al posto del vecchio `/etc/sendmail.cf`.

Naturalmente, perché Sendmail prenda atto della nuova configurazione, deve essere riavviato (dovrebbe bastare l'invio di un segnale di aggancio, `SIGHUP`).

Struttura e contenuto del file di configurazione

Il file di configurazione inizia generalmente con delle annotazioni, che possono riguardare il copyright o lo scopo del file. Osservando i file già esistenti si potrebbe pensare che il simbolo `#` rappresenti l'inizio di un commento; in realtà si tratta di un commento per il file `.cf` che si vuole generare, perché all'interno del sistema di macro di M4 è stato ridefinito opportunamente il simbolo di commento in modo che `#` venga trattato come un carattere qualunque senza significati particolari. Questo significa che le espansioni hanno luogo anche all'interno dei commenti per il file `/etc/sendmail.cf`.

Il modo adottato comunemente per eliminare le intestazioni contenenti le informazioni sul copyright e le riserve all'uso dei vari file, è quello di dirigere l'output in modo da perderlo, attraverso la macro `divert(-1)`.

In teorica, l'aspetto normale di un file di configurazione per questo pacchetto dovrebbe essere il seguente:

```
divert(-1)
#
# Copyright (c) 1983 Eric P. Allman
# Copyright (c) 1988, 1993
#   The Regents of the University of California. All rights reserved.
#
# Redistribution and use in source and binary forms, with or without...
# ...
divert(0)dnl
include('../m4/cf.m4')
VERSIONID('@(#)generic-linux.mc 8.3 (Berkeley) 3/23/96')
OSTYPE(linux)dnl
DOMAIN(generic)dnl
MAILER(local)dnl
MAILER(smtp)dnl
```

In pratica, questo potrebbe generare un file `.cf` insufficiente al funzionamento corretto di Sendmail.

Si può osservare all'inizio l'inclusione del file `m4/cf.m4` che è il responsabile dell'impostazione di questo sistema di macro.

Quasi tutte le macro specifiche che si utilizzano in questo file (quelle che appaiono in lettere maiuscole), rappresentano in realtà l'inclusione di un file, quello che appare come parametro, proveniente dalla directory corrispondente al nome della macro stessa. Per esempio, `OSTYPE(linux)` rappresenta in pratica l'inclusione del file `ostype/linux.m4`. Nelle sezioni seguenti vengono descritte brevemente alcune di queste macro specifiche.

Macro «VERSIONID»

```
VERSIONID(descrizione della versione)
```

La macro `VERSIONID` permette semplicemente di includere un'annotazione sulla versione della configurazione, nei commenti del file `.cf` generato. È utile per documentare diversi tipi di configurazione, tenuto conto che la forma per definire la versione non è prestabilita.

Macro «OSTYPE»

```
OSTYPE(macro_da_includere)
```

Attraverso la macro `OSTYPE` si può definire il nome del sistema operativo utilizzato. In pratica, si tratta di indicare il nome (senza estensione) di un file macro contenuto nella directory `ostype/`, da includere in quel punto.

Attraverso l'inclusione di questo file, si ottiene la definizione di alcune informazioni importanti riguardo all'installazione di Sendmail nel proprio sistema operativo; per esempio si può definire la collocazione del file contenente gli alias, il programma da usare per la consegna dei messaggi, le opzioni e gli argomenti che questo programma deve avere. Tutte queste informazioni vengono specificate attraverso la definizione di macro specifiche, come se si trattasse della definizione di variabili. Se tali macro non sono definite in questa occasione, vengono definite in un altro momento, ricevendo un valore predefinito, come documentato regolarmente nei file che accompagnano il pacchetto di configurazione.

L'esempio più semplice possibile del file `ostype/linux.m4` è il seguente,

```
divert(-1)
#
# ...
divert(0)
define('LOCAL_MAILER_PATH', /bin/mail)dnl
```

dove si definisce soltanto che il programma di consegna dei messaggi è `/bin/mail`. In pratica però, normalmente, questo file viene modificato opportunamente da chi allestisce il pacchetto di configurazione per una distribuzione GNU particolare.

La possibilità che questo file non sia conforme alla distribuzione standard del pacchetto di configurazione di Sendmail, deve essere tenuto in considerazione quando si vuole provare a generare un file `.cf` differente dal `/etc/sendmail.cf` già predisposto dalla propria distribuzione. Infatti, le modifiche che potrebbero essere state apportate possono pregiudicare l'effetto prevedibile delle altre macro.

Macro «DOMAINS»

```
DOMAINS(macro_da_includere)
```

Attraverso la macro `DOMAINS` si può definire il nome di una configurazione riferita a un dominio particolare. Si ottiene in pratica l'inclusione di un file contenuto nella directory `domains/`.

Il pacchetto di configurazione fornisce il file `domains/generic.m4`, che dovrebbe adattarsi a tutte le situazioni normali. Spesso, questo non viene utilizzato, inserendo direttamente quello che serve nel file di configurazione normale.

Quello che segue è un estratto dal file `domains/generic.m4`.

```
divert(-1)
#
# ...
divert(0)
VERSIONID('@(#)generic.m4 8.3 (Berkeley) 3/24/96')
define('confORWARD_PATH', '$z/.forward.$w:$z/.forward')dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
```

Macro «MAILERS»

```
MAILERS(macro_da_includere)
```

Attraverso la macro `MAILERS` si può definire il nome di una configurazione riferita a un tipo particolare di sistema di invio dei messaggi. Si ottiene in pratica l'inclusione di un file contenuto nella directory `mailers/`.

Normalmente, questa macro viene utilizzata più volte all'interno del file di configurazione, per definire diverse possibilità. Tipicamente si tratta di:

```
MAILER(local)
```

che si occupa della gestione dei messaggi all'interno del sistema e viene utilizzato in modo predefinito;

```
MAILER(smtp)
```

che si occupa di configurare la gestione dei messaggi attraverso il protocollo SMTP, cioè riguarda la configurazione necessaria all'invio dei messaggi al di fuori del sistema.

Nel primo caso si ha l'inclusione del file 'mailers/local.m4', nel secondo di 'mailers/smtp.m4'

Dalla macro 'MAILER(smtp)' dipende la base del sistema di sicurezza contro gli utilizzi indesiderati del proprio server SMTP. Infatti, è qui che vengono definite le istruzioni necessarie nel file '.cf' per impedire l'utilizzo da parte di nodi che non facciano parte della zona DNS di competenza. Cioè, quello che si vuole evitare è che un nodo diverso da quelli definiti nella zona per cui è stato previsto un record 'MX', possa utilizzare il server SMTP per raggiungere indirizzi al di fuori del sistema locale (si veda eventualmente quanto discusso nel capitolo precedente).

Macro «FEATURE»

```
FEATURE(macro_da_includere)
```

Attraverso la macro 'FEATURE' si può definire il nome di una configurazione riferita a una particolarità che si vuole includere. Si ottiene in pratica l'inclusione di un file contenuto nella directory 'feature/'.

Normalmente, questa macro viene utilizzata più volte all'interno del file di configurazione, ma questo preferibilmente prima di 'MAILER'.

Macro «HACK»

```
HACK(macro_da_includere)
```

Attraverso la macro 'HACK' si può definire il nome di una configurazione riferita a una particolarità sperimentale che si vuole includere. Si ottiene in pratica l'inclusione di un file contenuto nella directory 'hack/'.

Teoricamente, questa macro non dovrebbe essere utilizzata; in pratica succede spesso il contrario a causa delle esigenze di definire dei filtri aggiuntivi contro gli accessi indesiderati.

Esempio di una distribuzione GNU/Linux

A titolo di esempio, viene presentata la configurazione utilizzata dalla distribuzione GNU/Linux Red Hat 5.0, trattandosi precisamente del file 'cf/redhat.mc'.

```
divert(-1)
include('..m4/cf.m4')
define('CONFDEF_USER_ID','8:12')
OSTYPE('linux')
undefine('UUCP_RELAY')
undefine('BITNET_RELAY')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
MAILER(procmail)
MAILER(smtp)
HACK(check_mail3,'hash -a@JUNK /etc/mail/deny')
HACK(use_ip,'/etc/mail/ip_allow')
HACK(use_names,'/etc/mail/name_allow')
HACK(use_relayto,'/etc/mail/relay_allow')
HACK(check_rcpt4)
HACK(check_relay3)
```

La prima cosa che si osserva è che il file inizia con la macro 'divert(-1)', senza commenti da eliminare e senza il consueto 'divert(0)' successivo. In questo modo, dal momento che nessuna delle macro utilizzate dopo deve restituire qualcosa, si evita di terminare le varie macro con il solito 'dn1'.

Per sicurezza, nel caso servisse, vengono cancellate le macro 'UUCP_RELAY' e 'BITNET_RELAY'.

Invece di utilizzare una macro 'DOMAIN' vengono incluse direttamente le particolarità attraverso l'uso della macro 'FEATURE'. In particolare viene definito quanto segue.

• 'FEATURE(redirect)'

Si tratta di una particolarità poco importante, con la quale si ottiene di emettere un avviso nel caso sia utilizzato un indirizzo di posta elettronica nella forma 'indirizzo_normale.REDIRECT'. Si ottiene una segnalazione di errore in cui si invita a utilizzare la parte di indirizzo precedente a '.REDIRECT'.

• 'FEATURE(always_add_domain)'

Inserendo questa configurazione, si ottiene di aggiungere il nome a dominio all'utente destinatario quando questo non viene specificato esplicitamente.

• 'FEATURE(use_cw_file)'

In questo modo si ottiene di fare accettare a Sendmail l'identificazione del proprio nodo attraverso uno dei nomi elencati nel file '/etc/sendmail.cw'.

• 'FEATURE(local_procmail)'

Fa in modo di utilizzare 'procmail' come sistema di consegna dei messaggi in ambito locale.

Le macro 'HACK' inserite alla fine, sono state aggiunte per permettere una migliore gestione dei filtri di accesso al servizio di invio dei messaggi, comprendendo in questo anche la definizione di nodi per i quali il proprio server SMTP può agire come relè.

Per la precisione, è consentito l'uso dei file descritti nelle sezioni seguenti.

File «/etc/mail/ip_allow»

Si tratta di un file di testo contenente un elenco di indirizzi IP (uno per riga) riferiti a nodi particolari o a intere reti. A questi elaboratori viene consentito di utilizzare il server SMTP come relè. Per esempio,

```
192.168.1.2
192.168.2
```

permette l'accesso al nodo 192.168.1.2 e a tutta la rete 192.168.2.

Questo file, al di fuori della configurazione particolare della distribuzione Red Hat, potrebbe chiamarsi '/etc/mail/LocalIP'.

File «/etc/mail/name_allow»

Si tratta di un file di testo contenente un elenco di nomi a dominio (uno per riga) riferiti a nodi particolari o a tutti i nodi di un dominio particolare. A questi elaboratori viene consentito di utilizzare il server SMTP come relè. Per esempio,

```
roggen.brot.dg
mehl.dg
```

permette l'accesso al nodo *roggen.brot.dg* e a tutto il dominio *mehl.dg*.

Questo file, al di fuori della configurazione particolare della distribuzione Red Hat, potrebbe chiamarsi '/etc/mail/LocalNames'.

File «/etc/mail/relay_allow»

Si tratta di un file di testo contenente un elenco di indirizzi IP o di nomi a dominio (uno per riga) riferiti a nodi particolari o a tutti i nodi di una rete particolare o di un dominio. Questi indirizzi sono ammessi come destinatari di messaggi quando il server SMTP viene utilizzato come relè. Per esempio,

```
192.168
roggen.brot.dg
mehl.dg
```

permette di inviare messaggi alla rete 192.168.*.*, al nodo *roggen.brot.dg* e a tutto il dominio *mehl.dg*, quando il servente SMTP funziona come relè.

Questo file, al di fuori della configurazione particolare della distribuzione GNU/Linux Red Hat, potrebbe chiamarsi `/etc/mail/RelayTo`.

File `«/etc/mail/deny»`

«

Il file di testo `/etc/mail/deny` viene utilizzato per annotare un elenco di indirizzi di posta elettronica, nomi a dominio e indirizzi IP di mittenti indesiderati. A fianco di ogni indirizzo, separato da un carattere di tabulazione (`<HT>`), si indica il messaggio di errore che si vuole sia restituito all'MTA che ha contattato il servente per l'inoltro del messaggio.

Segue un esempio molto semplice di questo file.

```
spam@marameo.dg "Spiacente sig. Spam, non accettiamo messaggi da Lei."
spam.brot.dg    "Dal Vostro host non accettiamo email."
spammer.dg     "Non vogliamo spam, grazie."
192.168.13.13  "Dal Vostro host non accettiamo email."
192.168.17     "Non vogliamo spam, grazie."
```

Questo file non può essere usato così com'è; occorre generare un file adatto a Sendmail. Si utilizza in pratica il comando seguente:

```
# makemap -v hash /etc/mail/deny < /etc/mail/deny [Invio]
```

Quello che si ottiene è il file `/etc/mail/deny.db`.

¹ **Sendmail** software non libero: non è consentita la commercializzazione a scopo di lucro