Parte xviii

Posta elettronica

Sendmail: introduzione

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:     daniele
```

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:          daniele, 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
'`m4/`'	Contiene alcune macro di partenza, di cui, la più importante è '`cf.m4`' che viene usata per iniziare il procedimento.
'`cf/`'	Contiene i file di configurazione utilizzati dalla macro '`m4/cf.m4`'; questi file hanno l'estensione '`.mc`'. 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 '`linux.mc`' oppure di un altro nome che ricorda quello della distribuzione (per esempio '`redhat.mc`').
'`sh/`'	Contiene degli script di shell utilizzati automaticamente da M4, in base alle istruzioni contenute nelle macro utilizzate.
altre directory	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 '`cf/`'. Per esempio, un'istruzione del tipo '**`MAILER(local)`**', fa riferimento al file '`mailer/local.m4`'.

Quando si predispone 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('confFORWARD_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 servente 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 servente 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 '**`dnl`**'.

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 servente 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 servente 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 servente 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 servente 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

Exim 3: introduzione

In questo capitolo si introduce l'utilizzo di Exim, [1] un MTA che offre qualche piccolo vantaggio rispetto all'uso di Sendmail: è abbastanza compatibile con le consuetudini di questo ultimo; ha un sistema di configurazione meno criptico; è predisposto per IPv6; quando possibile utilizza processi senza i privilegi dell'utente '**`root`**', in modo da lasciare meno occasioni alle aggressioni.

Questo capitolo si riferisce precisamente a Exim nelle versioni 3.∗. Le versioni successive prevedono una configurazione differente.

Compatibilità con Sendmail e differenze importanti

Exim è compatibile con Sendmail per tutti quegli aspetti che coinvolgono gli utenti comuni e anche per ciò che riguarda gli amministratori che non hanno o non desiderano avere conoscenze troppo approfondite sulla gestione della posta elettronica. Questa compatibilità riguarda tre punti fondamentali: il file '`/etc/aliases`', i file '`~/.forward`' e un collegamento che simula la presenza dell'eseguibile '**`sendmail`**'. Per di più, l'eseguibile '**`exim`**' accetta buona parte delle opzioni standard di '**`sendmail`**', in modo da permettere il funzionamento di programmi come Mailx, o il funzionamento di script che si affidano alla presenza dell'eseguibile '**`sendmail`**'.

I file ‘`/etc/aliases`’ e ‘`~/.forward`’ si comportano in modo quasi identico rispetto a quando è in funzione Sendmail. In particolare, con ‘`~/.forward`’ si possono usare anche delle estensioni.

Un’eccezione, rispetto alla compatibilità di questi file, riguarda l’indicazione di condotti. Con Sendmail, si presume che il comando sia elaborato da una shell; con Exim, no. Di conseguenza, i comandi interni di questa non sono accessibili. Si osservino gli esempi seguenti, riferiti al contenuto del file ‘`/etc/aliases`’: si tratta dello stesso condotto a cui vengono ridiretti i messaggi giunti per l’utente ipotetico, denominato ‘**`lista-pippo`**’.

```
# con Sendmail
lista-pippo:  "| exec /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Sendmail senza exec
lista-pippo:  "| /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Exim non si può usare exec che è un comando interno di shell
lista-pippo:  "| /home/liste/bin/ricezione-messaggi lista-pippo"
```

```
# con Exim, avviando prima una shell e quindi il comando
lista-pippo:  "| /bin/sh -c '/home/liste/bin/ricezione-messaggi lista-pippo'"
```

Se si deve inserire un condotto in un file ‘`~/.forward`’, vale lo stesso ragionamento, con la differenza che qui non si mette più l’indicazione del destinatario perché è implicita (ma questo vale anche per Sendmail).

Il primo vantaggio che si osserva rispetto a Sendmail è che il file ‘`/etc/aliases`’ non deve essere «ricompilato» attraverso ‘**`newaliases`**’: basta modificarlo e non occorre nemmeno riavviare il servizio perché viene riletto ogni volta dal sistema di consegna locale.

Se nel file ‘`~/.forward`’ si inserisce un indirizzo che crea un circolo senza fine, come per esempio quando si indica lo stesso indirizzo dell’utente per il quale è stato creato il file, i messaggi vengono consegnati presso quello stesso recapito, invece di ignorarli semplicemente.

I permessi del file ‘`~/.forward`’ non possono concedere la scrittura al gruppo e al resto degli utenti. Questo particolare va considerato quando si utilizza una maschera dei permessi pari a 002_8 (è così, solitamente, quando si usano i gruppi privati), che tende a concedere la scrittura al gruppo in modo predefinito.

Come già accennato, Exim fornisce un collegamento denominato ‘**`sendmail`**’, per favorire il funzionamento dei programmi che dipendono dalla presenza di questo; inoltre, offre il collegamento ‘**`mailq`**’, come fa Sendmail, per permettere la verifica dei messaggi in coda.

Installazione

L’installazione di Exim può costituire un problema se non si parte da un pacchetto già predisposto per la propria distribuzione GNU; quindi è decisamente preferibile cercare un tale pacchetto già pronto. Purtroppo, in certi casi, anche questo non basta: occorre preparare qualcosa prima, forse è necessario definire la configurazione; infine occorre fare delle sistemazioni finali dopo alcune prove di verifica.

Utente specifico

Quando possibile, se la configurazione lo consente, Exim cerca di avviare processi con privilegi inferiori a quelli dell’utente ‘**`root`**’. Per esempio, la consegna locale della posta avviene normalmente con un processo che utilizza i privilegi dell’utente destinatario. In tutte le altre circostanze, si può stabilire un utente e un gruppo che Exim deve utilizzare: nei sistemi che utilizzano i gruppi privati (un gruppo per ogni utente), si potrebbe creare l’utente e il gruppo ‘**`exim`**’; negli altri sistemi, può essere conveniente creare solo l’utente ‘**`exim`**’, a cui abbinare il gruppo ‘**`mail`**’.

Pertanto, la prima cosa da fare è la creazione di questo utente ed eventualmente del gruppo corrispondente (se non è già previsto, o

se si utilizzano i gruppi privati). Nel file ‘`/etc/passwd`’ potrebbe apparire una riga come quella seguente, dove il numero GID 12 è inteso corrispondere a ‘**`mail`**’.

```
exim:*:501:12:Exim mailer:/:
```

Se si usano i gruppi privati, si potrebbero avere i record seguenti, rispettivamente nei file ‘`/etc/passwd`’ e ‘`/etc/group`’.

```
exim:*:501:501:Exim mailer:/:
```

```
exim:x:501:
```

In ogni caso, come si è visto, è importante che l’accesso sia impossibile, cosa che si ottiene con l’asterisco nel campo della parola d’ordine.

File «/etc/aliases»

Dopo l’installazione di Exim, si può fare in modo di recuperare il vecchio ‘`/etc/aliases`’, se esiste, oppure se ne deve creare uno nuovo. Per il momento, fino a che non è stata vista la configurazione di Exim, è meglio lasciare stare gli alias che si traducono in file o in condotti.

Exim può essere configurato per utilizzare un file diverso da ‘`/etc/aliases`’ con questo stesso scopo, ma in generale dovrebbe essere conveniente mantenere la vecchia convenzione. In ogni caso, Exim ha bisogno della definizione di alcuni alias indispensabili: in generale, Exim non permette la consegna di messaggi direttamente all’utente ‘**`root`**’, quindi è necessario definire chi sia l’utente corrispondente che deve ricevere la posta diretta a ‘**`root`**’. Di seguito viene mostrato un esempio che dovrebbe andare bene in molte piattaforme GNU: l’alias di ‘**`root`**’ deve essere modificato opportunamente.

```
# Obbligatori
MAILER-DAEMON:  postmaster
abuse:          postmaster
postmaster:     root

# Ridirezione per evitare trucchi con gli utenti speciali di sistema.
bin:            root
daemon:         root
adm:            root
lp:             root
sync:           root
shutdown:       root
halt:           root
mail:           root
news:           root
uucp:           root
operator:       root
games:          root
gopher:         root
ftp:            root
nobody:         root
postgres:       root
exim:           root

# Chi è root (modificare in base alla realtà del proprio sistema)
#root:          daniele
```

File «~/.forward»

Anche la gestione dei file ‘`~/.forward`’ può essere controllata (e stravolta) attraverso la configurazione di Exim; tuttavia, se si desidera mantenere le consuetudini, si possono recuperare questi file che gli utenti potrebbero avere già utilizzato con Sendmail. Valgono naturalmente le stesse riserve già espresse in riferimento a destinatari costituiti da file o da condotti.

In ogni caso, perché tali file ‘`~/.forward`’ possano essere accettati da Exim, occorre che siano assenti i permessi di scrittura per il gruppo e per gli altri utenti. Utilizzando la shell Bash, si potrebbe usare un comando come quello seguente:

```
# find /home -name .forward -exec chmod go-w \{\} \;
```
[*Invio*]

Directory di destinazione dei messaggi locali

Sendmail utilizza una directory comune a tutti gli utenti per inserirvi i file contenenti i messaggi di questi, quando giungono a destinazione. In passato si è trattato di ‘`/var/spool/mail/`’ e attualmente dovrebbe essere ‘`/var/mail/`’, per conformità con lo standard

FHS. Exim può funzionare nello stesso modo, oppure può consegnare i messaggi direttamente nelle directory personali degli utenti. In generale, qualunque sia la scelta, è necessario che la variabile di ambiente '**MAIL**' contenga il percorso assoluto per raggiungere il file di destinazione, in modo che i programmi di lettura della posta vi si possano adeguare. Questo lo si fa normalmente nella definizione del profilo della shell personale.

Per esempio, se si utilizza la shell Bash, il file '`/etc/profile`' potrebbe contenere le righe seguenti per indicare che i file dei messaggi si trovano nella directory '`/var/mail/`'.

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

Nel caso la posta venisse consegnata nel file '`~/Messaggi`' della directory personale di ogni utente, l'istruzione per definire la variabile '**MAIL**' potrebbe essere la seguente:

```
MAIL="$HOME/Messaggi"
export MAIL
```

Infine, si deve tenere presente che Exim utilizza i privilegi dell'utente destinatario per aprire i file di destinazione, per cui i premessi della directory devono essere regolati convenientemente. Il problema si pone quando si usa la directory '`/var/mail/`', o un'altra simile, per tutti i file di destinazione: è necessario che sia attribuita a questa directory la modalità 1777_8. Infatti, l'attivazione del bit Sticky, permette il blocco dei file (*lock*). Se non si ha l'accortezza di sistemare questo particolare, la posta elettronica non può essere consegnata.

```
# chmod 1777 /var/mail [Invio]
```

Organizzazione della configurazione

La configurazione di Exim è più semplice di Sendmail, ma resta comunque una cosa piuttosto delicata, dati i problemi che sono coinvolti nella gestione della posta elettronica. Alla fine di questo gruppo di sezioni viene mostrato un esempio completo di configurazione che dovrebbe funzionare correttamente nella maggior parte delle situazioni.

La documentazione di Exim è voluminosa e abbastanza dettagliata. Questo è un fatto positivo; purtroppo occorre dedicarvi un po' di tempo per la sua lettura. Se si desidera utilizzare Exim a livello professionale, ciò diventa necessario.

Il file di configurazione di Exim, volendo seguire lo standard dei sistemi GNU (e non solo di quelli), dovrebbe trovarsi nella directory '`/etc/`'. In pratica però, potrebbe non essere così. Una volta installato il pacchetto di Exim, occorre cercare il file di configurazione.

Il file di configurazione deve appartenere all'utente '**root**', oppure all'utente specificato in fase di compilazione dei sorgenti di Exim, attraverso l'opzione '**EXIM_UID**'; inoltre non può essere accessibile in scrittura dal gruppo né dagli altri utenti.

Quando si avvia Exim, se il file di configurazione contiene errori sintattici, viene emesso un messaggio di errore attraverso lo standard error, specificando anche la riga in cui questo si trova. Dopo tale segnalazione, Exim termina di funzionare, per cui il servizio non viene avviato.

Struttura

Il file di configurazione si divide in sei parti che devono apparire nell'ordine previsto. Ognuna di queste termina con la parola chiave '**end**', posta da sola in una riga. Le varie parti sono elencate di seguito.

1. Configurazione principale. Si tratta di direttive in cui si assegnano dei valori a delle opzioni di funzionamento.
2. Configurazione dei driver di trasporto. Si tratta della definizione dei meccanismi attraverso cui i messaggi vengono recapitati alla destinazione, copiandoli all'interno dei file, o inserendoli nei condotti.
3. Configurazione dei driver di direzione (*director*). Si tratta dei processi di consegna all'interno dei domini locali, cosa che include la gestione degli alias e del proseguimento dei messaggi (*forward*).
4. Configurazione dei driver di instradamento. Si tratta dei processi di consegna a destinazioni remote, ovvero, quelle destinazioni che non sono classificate come appartenenti ai domini locali.
5. Configurazione delle regole per i tentativi ripetuti (*retry*).
6. Configurazione delle regole di riscrittura. Si tratta della definizione di modifiche sistematiche a elementi dell'intestazione dei messaggi.

In ogni parte della configurazione possono apparire dei commenti; questi sono introdotti dal simbolo '**#**' all'inizio della riga e conclusi dalla fine della riga stessa. Non sono ammessi commenti alla fine delle direttive; in pratica, i commenti possono apparire solo su righe apposite. Inoltre, le righe bianche e quelle vuote vengono ignorate come di consueto.

Elementi comuni della configurazione

Prima di affrontare la descrizione di alcune direttive importanti che possono essere usate nel file di configurazione, conviene conoscere alcune convenzioni comuni, o direttive particolari che coinvolgono tutto l'insieme della configurazione.

Macro

All'interno della prima parte del file di configurazione, quella che riguarda le definizioni generali, è possibile inserire delle direttive che dichiarano delle macro. Queste si distinguono perché devono avere l'iniziale maiuscola. In generale, per convenzione comune derivante da altri linguaggi di programmazione, le macro si dichiarano con nomi composti esclusivamente da lettere maiuscole.

```
nome = valore_da_sostituire
```

Il nome può essere composto da lettere, numeri e dal trattino basso ('_'); inoltre, come accennato, la prima lettera deve essere maiuscola. Il valore che si abbina a questo nome, è tutto ciò che appare dopo il simbolo '**=**', escludendo eventuali spazi iniziali, fino alla fine della riga.

Assegnamento

In molti punti del file di configurazione si usano delle direttive che rappresentano in pratica l'assegnamento di un valore a una sorta di variabile. La sintassi è semplice e intuitiva.

```
nome = valore
```

La differenza rispetto alla dichiarazione di macro sta nel fatto che i nomi utilizzati in questo caso sono prestabiliti e non iniziano mai con una lettera maiuscola.

Valori booleani

Quando una variabile è fatta per definire l'attivazione o la disattivazione di qualcosa, può ricevere solo i valori *Vero* o *Falso*, espressi attraverso le solite parole chiave: '**true**' o '**yes**' rappresenta il valore *Vero*; '**false**' o '**no**' rappresenta il valore *Falso*.

In particolare, in presenza di variabili di questo tipo, è possibile fare a meno di indicare espressamente l'assegnamento, lasciando intuire il valore predefinito derivante dal nome della variabile. In generale, comunque, sarebbe bene esplicitare l'intenzione, se si vogliono utilizzare tali variabili.

Interi

I numeri interi possono essere annotati utilizzando diverse basi di numerazione:

- un numero che inizia con il prefisso 0x... viene inteso essere espresso in base esadecimale;
- un numero che inizia con uno zero viene inteso essere espresso in base ottale;
- un numero che inizia con una cifra diversa da zero viene inteso essere espresso in base decimale.

Tali numeri interi possono essere seguiti dalla lettera '**K**' (maiuscola) o dalla lettera '**M**', intendendo esprimere un multiplo di 1024, o di 1024*1024 rispettivamente (kibi e mebi delle convenzioni che si usano in informatica).

Numeri con parte decimale

Un numero contenente una parte decimale può essere espresso solo utilizzando la numerazione a base 10 (base decimale), indicando le cifre della parte frazionaria dopo un punto. Sono consentite un massimo di tre cifre decimali dopo la parte intera.

Intervalli orari

Le indicazioni di valori riferiti a intervalli orari (periodi di tempo), fanno uso di una serie di suffissi che descrivono il significato del numero che li precede.

- '**s**' secondi;
- '**m**' minuti;
- '**h**' ore;
- '**d**' giorni;
- '**w**' settimane.

Per esempio, il valore '**3h45m**' rappresenta 3 ore e 45 minuti. Questo formato di rappresentazione viene usato anche nell'output.

Stringhe

Le stringhe possono essere rappresentate con o senza apici doppi di delimitazione. L'utilizzo o meno di tale delimitazione ha delle conseguenze diverse.

Le stringhe non delimitate sono rappresentate da tutto ciò che appare dopo il simbolo '**=**' utilizzato nell'assegnamento, letteralmente, escludendo eventuali spazi iniziali, continuando fino alla fine della riga. Ciò significa in pratica che una stringa di questo tipo non può continuare nella riga successiva.

Si utilizzano le stringhe delimitate tutte le volte in cui occorre rappresentare qualcosa di particolare, come dei caratteri speciali, attraverso il prefisso '\' che assume il ruolo di carattere di escape, oppure quando è necessario continuare la stringa nella riga successiva.

La tabella u52.11 mostra l'uso di queste sequenze di escape ottenute con la barra obliqua inversa.

Tabella u52.11. Elenco delle sequenze di escape utilizzabili all'interno delle stringhe delimitate da apici doppi.

Simbolo	Significato
`\\`	Rappresenta una barra obliqua inversa singola.
`\n`	Il carattere <*NL*>.
`\r`	Il carattere <*CR*>.
`\t`	Una tabulazione orizzontale (<*HT*>).
`\`***n_ottale***	Identifica un carattere attraverso il suo numero ottale.
`\x`***n_esadecimale***	Identifica un carattere attraverso il suo numero esadecimale.

Inoltre, una barra obliqua inversa posta alla fine della riga, subito prima del codice di interruzione di riga, rappresenta la continuazione della stringa nella riga successiva, eliminando gli spazi iniziali aggiunti nella riga successiva.

Se si utilizza una barra obliqua inversa davanti a un carattere con il quale non forma alcuna sequenza di escape prevista, si conferma semplicemente il carattere successivo alla barra. Dal momento che all'interno delle stringhe possono essere usati altri simboli con significati speciali, si può usare la barra obliqua inversa per dare loro un significato puramente letterale.

Elenchi di stringhe

In situazioni determinate si può indicare un elenco di stringhe. La rappresentazione di tali elenchi avviene di fatto in una sola stringa, i cui elementi sono separati attraverso due punti verticali ('**:**'). Per esempio, '**uno:due:tre**' è un elenco composto dalle sottostringhe '**uno**', '**due**' e '**tre**'. È importante sapere subito che attorno ai due punti verticali, possono essere inseriti degli spazi, che poi vengono eliminati dalle sottostringhe; quindi, tornando all'esempio già presentato, sarebbe stata esattamente la stessa cosa scrivere '**uno: due :tre**'

A seconda delle esigenze, tali elenchi possono essere racchiusi globalmente attraverso gli apici doppi delle stringhe normali, oppure possono farne senza, con le stesse considerazioni già fatte su questo argomento.

Da quanto descritto, si intende che i due punti verticali abbiano un significato speciale, per cui non possono essere usati per altri scopi, a meno che questi appaiano in coppia ('**::**'), perché in tal caso rappresentano esattamente due punti verticali testuali.

Gli elenchi di stringhe vengono usati per rappresentare vari tipi di informazioni, spesso molto utili per una configurazione efficace di Exim. Qui viene trascurata la descrizione di queste indicazioni, che possono essere approfondite leggendo la documentazione originale.

Espansione delle stringhe

All'interno delle stringhe possono essere inseriti degli elementi che vengono sostituiti in qualche modo, in base a ciò che questi rappresentano. Per identificare tali elementi si utilizza il simbolo dollaro ('**$**') seguito da un nome, che eventualmente può anche essere racchiuso tra parentesi graffe, in caso si temano delle ambiguità.

$***nome_di_variabile***	${***nome_di_variabile***}

Esistono poi una serie di operazioni che possono essere compiute attraverso l'operatore di sostituzione (il dollaro), che qui non vengono descritte.

Nel caso si debba inserire il simbolo '**$**' in una stringa con un significato letterale, occorre indicare '**\$**' se si tratta di una stringa non delimitata, oppure '**\\$**' se si tratta di una stringa delimitata (incoerente, ma è così).

Espressioni regolari

In alcune situazioni, le stringhe possono servire a esprimere delle espressioni regolari. Tali espressioni regolari si distinguono per il fatto che iniziano con l'accento circonflesso ('**^**') e possono terminare o meno con il simbolo dollaro, che in tal caso rappresenta la fine della stringa con cui avviene il confronto.

Le regole per la realizzazione di tali espressioni regolari sono simili a quelle di Perl 5, facendo attenzione però alle barre oblique inverse, che se si trovano racchiuse tra apici doppi, devono essere raddoppiate.

Configurazione in pratica

Nelle sezioni successive vengono descritte, a gruppi di competenza, le direttive principali per la configurazione di Exim.

Configurazione principale

La prima parte del file di configurazione, fino al primo '**end**', riguarda la definizione delle opzioni principali. Come è già stato accennato, è in questa parte che possono essere create delle macro; la loro definizione si distingue in quanto i nomi di queste devono iniziare con una lettera maiuscola.

Questa parte della configurazione è la più semplice, perché richiede solo l'assegnamento di qualche valore a delle variabili prestabilite. L'elenco di tali variabili è molto lungo, ma in ogni caso, è sufficiente definire gli assegnamenti riferiti alle opzioni che si vogliono modificare rispetto a quanto risulta predefinito.

Directory	Descrizione
`exim_path =` ***percorso_assoluto_di_exim***	Permette di specificare il percorso assoluto dell'eseguibile '**exim**'. Questa informazione serve quando la collocazione del programma non corrisponde all'informazione indicata in fase di compilazione. La conoscenza di tale percorso serve a Exim quando deve avviare una copia di se stesso.
`primary_hostname =` ***nome_canonico***	Permette di definire esplicitamente il nome canonico primario del nodo locale. Se non viene specificata questa opzione, tale nome viene ottenuto dal sistema operativo, attraverso la funzione '**uname()**' (praticamente ciò che si otterrebbe con il comando '**hostname -f**').
`qualify_domain =` ***nome_di_dominio***	Permette di specificare il nome a dominio da aggiungere agli indirizzi che non lo possiedono. In mancanza di questa indicazione si usa il valore fissato con la direttiva '**primary_hostname**'.
`host_lookup =` ***indirizzo_ip/n_bit_maschera***	Permette di definire un gruppo di indirizzi per i quali verificare sempre il nome attraverso il DNS. Generalmente, viene assegnato '*****' che rappresenta ogni indirizzo possibile.

Directory	Descrizione
`local_domains =` ***nome_locale*** `[:` ***nome_locale*** `]…`	Permette di definire i nomi a dominio completi che fanno capo al sistema locale, per i quali la posta elettronica viene consegnata localmente, senza attivare una connessione SMTP. In pratica, consente di definire anche i domini virtuali che fanno capo allo stesso nodo locale. Come si vede dalla sintassi, si tratta di un elenco di stringhe, separato attraverso due punti verticali. Dovrebbe essere conveniente indicare sempre almeno i domini *localhost* e *localhost.localdomain*, nell'ipotesi che qualcuno usi indirizzi del tipo *tizio@localhost*, per quanto ciò possa sembrare strano assurdo.
`local_domains_include_host = {true|false}`	Attivando questa opzione ('**true**'), si fa in modo che il nome del nodo locale ottenuto dal sistema operativo, venga incluso automaticamente nell'elenco di domini locali ('**local_domains**').
`local_domains_include_host_literals = {true|false}`	Attivando questa opzione ('**true**'), si fa in modo di accettare messaggi, destinati al nodo locale, che fanno uso dell'indirizzo IP in forma letterale, al posto di un nome a dominio (secondo la forma ***nome*** *@[* ***indirizzo_ipv4*** *]*).
`forbid_domain_literals`	La presenza di questa opzione impedisce in generale l'uso di indirizzi numerici al posto del nome a dominio.
`log_level =` ***n_livello***	Permette di definire il livello di dettaglio per le informazioni memorizzate nei file delle registrazioni. Il valore zero corrisponde al minimo; valori superiori aumentano le informazioni (sei dovrebbe essere il valore che genera la massima quantità di notizie). Se questa opzione non viene dichiarata, il livello predefinito è cinque.
`message_size_limit =` ***dimensione***	Permette di fissare un tetto massimo alla dimensione dei messaggi in transito. Qui si usano normalmente i moltiplicatori '**K**' o '**M**'.

Directory	Descrizione
`never_users = `*utente*`[ : `*utente*`]`...	Permette di escludere il recapito di messaggi a utenti determinati, a meno che sia stato specificato un alias adatto nel file '`/etc/aliases`'. In pratica, serve per indicare l'elenco degli utenti di sistema, a cominciare da '**`root`**', che non possono o non dovrebbero ricevere posta.
`host_accept_relay = `*nodo*`[ : `*nodo*`]`... `host_accept_relay = ↩` `↪`*indirizzo_ipv4*`[ / `*n_bit_maschera*`][ : ... ]`...	Permette di definire a quali nodi è consentito usare il servente locale in qualità di relè. Questi nodi possono essere specificati per nome, per indirizzo IP o per gruppi di indirizzi IP.
`relay_domains = `*nome_di_dominio*`↩` `↪[ : `*nome_di_dominio*`]`...	Permette di elencare i domini per i quali si consente al servente di funzionare come relè.
`relay_domains_include_local_mx = ↩` `↪{true|false}`	Attivando l'opzione, si fa in modo di consentire la funzionalità di relè verso i domini che, secondo il DNS, dovrebbero essere serviti dal nodo locale. In pratica, si fa in modo di seguire la configurazione definita attraverso il DNS, con i record '**`MX`**', con cui si stabilisce che tale servente SMTP si deve occupare della consegna presso quei domini, accettando messaggi provenienti da qualunque dominio esterno.
`spool_directory = `*directory*	Definisce il percorso della directory usata come coda dei messaggi da Exim. Generalmente potrebbe trattarsi di '`/var/spool/exim/`', che poi si articola ulteriormente.
`trusted_users = ↩` `↪`*utente_fidato*`[ : `*utente_fidato*`]`... `trusted_groups = ↩` `↪`*gruppo_fidato*`[ : `*gruppo_fidato*`]`...	Definisce quali processi possono passare messaggi a Exim, specificando il mittente attraverso l'opzione '**`-f`**' della riga di comando. Tali processi sono accettati in quanto avviati con i privilegi dell'utente o del gruppo indicati. Se non viene specificata alcuna di queste due opzioni, ciò può essere fatto solo da un processo con i privilegi dell'utente '**`root`**'. Solitamente si definisce in questo modo l'utente '**`exim`**' o l'utente '**`mail`**' (senza indicare alcun gruppo); potrebbe essere conveniente aggiungere altri utenti nel caso si vogliano gestire delle liste (*mailing-list*) con caratteristiche particolari.

Configurazione dei driver

La seconda, la terza e la quarta parte del file di configurazione sono dedicate alla definizione delle istanze dei driver di trasporto, di direzione (*director*) e di instradamento.

In queste parti, le direttive del file di configurazione sono suddivise a gruppetti, ognuno riferito alla definizione di un'istanza particolare. In pratica, appare la dichiarazione del nome dell'istanza che termina con due punti verticali, seguita da una riga contenente la dichiarazione del driver di riferimento, oltre a una serie di altre righe opzionali contenenti le impostazioni che gli si devono applicare (quando quelle predefinite non vanno bene).

nome_di_istanza_del_driver :
    `driver = `*nome_del_driver*
    [*direttiva_di_opzione*]
    [*direttiva_di_opzione*]
    ...

Le opzioni che possono essere indicate, si distinguono in generiche e private. Le opzioni generiche possono essere utilizzate con tutti i driver di uno stesso tipo (trasporto, direzione, instradamento), mentre quelle private si riferiscono solo a driver particolari. La direttiva che definisce il driver è un'opzione generica, che deve essere posta all'inizio, come mostra lo schema sintattico.

Nelle prime versioni di Exim è necessario separare le opzioni con una virgola, mettendo prima le opzioni generiche e dopo quelle specifiche; inoltre, il passaggio da opzioni generiche a opzioni specifiche dove essere segnalato con un punto e virgola. Nelle versioni più recenti queste restrizioni non esistono e non è richiesta l'indicazione di virgole o punti e virgola. Tuttavia, l'informazione viene riportata a spiegazione del motivo per il quale diversi esempi di configurazione in circolazione hanno queste virgole e questi punti e virgola, qua e là, senza un motivo apparente.

A titolo di esempio vengono mostrate e descritte un paio di dichiarazioni significative, che appaiono anche nell'esempio completo mostrato più avanti.

```
local_delivery:
  driver = appendfile
  file = /var/mail/${local_part}
```

Nella configurazione del trasporto, definisce l'istanza '**`local_delivery`**' del driver '**`appendfile`**'. Dal nome si intende che si tratta del trasporto che si deve occupare di consegnare localmente la posta elettronica.

Attraverso il driver '**`appendfile`**' si ottiene di aggiungere i messaggi a un file già esistente, specificato attraverso l'opzione '**`file`**': in questo caso si tratta di '**`/var/mail/${local_part}`**', che in pratica si espande in un file denominato come l'utente che deve riceverlo, collocato nella directory '`/var/mail/`'.[2]

```
localuser:
  driver = localuser
  transport = local_delivery
```

Questo esempio fa riferimento alla configurazione del sistema di direzione; il nome dell'istanza è lo stesso di quello del driver, ma si tratta di cose differenti. Si può osservare la dichiarazione del trasporto utilizzato: '**`local_delivery`**', cioè il tipo di trasporto (l'istanza) già vista nell'esempio precedente.

Configurazione dei tentativi ripetuti

La penultima parte del file di configurazione, serve a definire il modo in cui scandire la ripetizione dei tentativi di invio (o di consegna) della posta. Ciò permette di distinguere il comportamento in base al dominio di destinazione e al tipo di errore che ha impedito la consegna del messaggio. Generalmente si trova già un esempio generico sufficiente.

Gli intervalli con cui vengono ripetuti i tentativi, devono tenere conto della frequenza con cui viene riavviato il processo di scansione della coda. Per esempio, se viene avviato '**exim**' con l'opzione '**-q30m**', che, come viene descritto in seguito, richiede il controllo della coda ogni 30 minuti, è poco sensato specificare nella configurazione intervalli inferiori, perché non potrebbero essere rispettati.

Configurazione della riscrittura degli indirizzi

L'ultima parte della configurazione è generalmente assente, o senza direttive. Serve a definire delle regole di alterazione sistematica degli indirizzi.

Per comprendere il problema viene descritto un caso pratico che potrebbe interessare. Quando si passa da Sendmail a Exim, potrebbe sentirsi la necessità di fare in modo che gli indirizzi ***nome*+*qualcosa*@*dominio***, vengano consegnati a ***nome*@*dominio***. Alcuni utenti potrebbero utilizzare questo trucco (comune per Sendmail) per distinguere la fonte da cui lo scrivente può avere tratto il loro indirizzo e avere implicitamente un'idea del contesto per il quale viene inviato ogni messaggio.

Exim ha dei meccanismi più potenti, ma quando si passa da Sendmail a Exim, gli utenti potrebbero desiderare di mantenere le vecchie convenzioni. La direttiva seguente dovrebbe risolvere il problema.

```
^(..*?)\+(.*)@(..*)$ $1@$3 T
```

Come si vede, attraverso un'espressione regolare vengono estratti gli elementi che contano dall'indirizzo, che poi viene ricostruito senza la parte superflua che ne impedirebbe il recapito.

Un esempio più semplice di questo problema è quello di una rete locale che utilizza nomi a dominio inesistenti nella rete esterna, pertanto si vuole sostituire l'indicazione dei domini locali con un dominio che esiste realmente:

```
*@brot.dg                $local_part@linuxdidattica.org E
*@localhost              $local_part@linuxdidattica.org E
*@localhost.localdomain  $local_part@linuxdidattica.org E
```

In questo modo, tutti i campi della busta del messaggio che corrispondono ai modelli indicati, vengono rimpiazzati con un indirizzo del dominio *linuxdidattica.org*.

Configurazione generica

Le versioni più recenti di Exim vengono distribuite normalmente con una configurazione generica, ben commentata, sufficiente al recapito locale dei messaggi, senza la possibilità di ricevere messaggi dall'esterno o di inviarne dall'interno.

L'attenzione maggiore va posta naturalmente a tutte quelle direttive che incorporano la parola chiave '**relay**', come '**relay_domains**' per esempio; inoltre diventa indispensabile verificare che la posta locale sia recapitata esattamente dove previsto, con la direttiva '**local_delivery**'.

Interazione con Procmail

Procmail è un programma per l'elaborazione dei messaggi di posta elettronica, che può essere usato per vari fini. La configurazione predefinita di Exim prevede spesso la presenza di direttive per passare a Procmail il compito di recapitare i messaggi locali, come nell'esempio seguente:

```
#
# Transports configuration.
#
…
#
# This transport is used for procmail.
#
procmail_pipe:
  driver = pipe
  command = "/usr/bin/procmail"
  return_path_add
  delivery_date_add
  envelope_to_add
  suffix = ""
```



```
…
#
# Directors configuration
#
…
#
# This director runs procmail for users who have a .procmailrc file
#
procmail:
  driver = localuser
  transport = procmail_pipe
  require_files = ${local_part}:+${home}:+${home}/.procmailrc:+/usr/bin/procmail
  no_verify
```

Senza entrare nel dettaglio del significato delle direttive, basta osservare che la presenza del file '`~/.procmailrc`', assieme al programma ('`/usr/bin/procmail`'), fa sì che questo tipo di recapito venga attivato. Di conseguenza, con tale situazione, **la configurazione errata del file '`~/.procmailrc`' potrebbe causare un funzionamento errato nel recapito dei messaggi locali**.

Avvio di Exim

L'avvio di Exim, allo scopo di attivare il servizio SMTP, avviene di solito attraverso la procedura di inizializzazione del sistema, come processo indipendente dal supervisore dei servizi di rete, anche se questa ultima possibilità è comunque consentita. Ma Exim può essere avviato anche per altri motivi, in particolare per ricevere un messaggio dallo standard input, da recapitare in qualche modo, oppure per ripassare i messaggi rimasti in coda, per ritentare il loro invio.

A seconda dello scopo per il quale viene avviato l'eseguibile '**exim**', possono essere richiesti dei privilegi particolari. Per la precisione, si distingue tra utenti comuni e amministratori. L'amministratore è l'utente '**root**', l'utente abbinato a Exim (normalmente '**exim**') e gli utenti definiti attraverso l'opzione '**trusted_users**'.

Uno dei motivi per cui può essere più conveniente avviare il servizio SMTP di Exim, in modo indipendente dal supervisore dei servizi di rete, è il fatto di poter affidare al demone Exim, così avviato, anche il compito di provvedere alla gestione dei messaggi in coda in modo automatico. Se si utilizza il controllo del supervisore dei servizi di rete, occorre affidare il lavoro di gestione della coda a un altro processo.

Quando si usa Exim come demone, cioè in modo autonomo dal supervisore dei servizi di rete, si usa l'opzione '**-bd**', seguita quasi sempre da '**-q*tempo***', che serve a specificare l'intervallo di scansione della coda di messaggi in attesa.

Se si vuole gestire il servizio SMTP attraverso il controllo del supervisore dei servizi di rete, occorre specificare l'opzione '**-bs**' e si deve dichiarare una riga simile a quella seguente nel file '`/etc/inetd.conf`'.

```
smtp    stream  tcp     nowait  root    /usr/sbin/exim  exim -bs
```

La riga di comando dell'eseguibile '**exim**' si può rappresentare sinteticamente così:

```
exim [opzioni]
```

Di seguito vengono elencate solo alcune opzioni assolutamente indispensabili, che servono a rendere l'idea delle funzioni di questo eseguibile.

Directory	Descrizione
`-bd`	Avvia '**exim**' come demone in attesa di connessioni SMTP. '**exim**' può essere avviato in questo modo solo da un amministratore; comunque ciò avviene di solito per mezzo della procedura di inizializzazione del sistema. Quando '**exim**' viene avviato con questa opzione, ma in modo manuale, può essere conveniente aggiungere l'uso delle opzioni diagnostiche '**-d**' o '**-dm**'.

Directory	Descrizione
`-bs`	Avvia '**`exim`**' in modo che questo si metta in attesa di ricevere messaggi dallo standard input, rispondendo poi a questi attraverso lo standard output. Questa opzione viene usata normalmente per gestire l'avvio di '**`exim`**' attraverso il supervisore dei servizi di rete.
`-bp`	Questa opzione permette di visualizzare l'elenco dei messaggi rimasti in coda per qualche motivo. Il funzionamento non è perfettamente identico a Sendmail, in quanto solo un utente con i privilegi necessari può ottenere queste informazioni.
`-bt` [*indirizzo*]	Avvia '**`exim`**' in una modalità di verifica degli indirizzi. Se viene indicato un indirizzo come argomento dell'opzione, si ottengono le informazioni essenziali sulla consegna verso tale destinazione. Ciò permette di verificare la correttezza della configurazione, dal momento che si ottiene anche l'indicazione del tipo di direzione e di trasporto utilizzati. Se non viene specificato l'indirizzo nella riga di comando, '**`exim`**' funziona in modo interattivo, proponendo un invito (il simbolo '**`>`**') per l'inserimento di ogni indirizzo da verificare (per terminare si può utilizzare la combinazione [*Ctrl c*]).
`-d`[*n_livello*]	Permette di avviare '**`exim`**' in modo diagnostico, allo scopo di visualizzare informazioni attraverso lo standard error. Dopo la lettera dell'opzione, può essere aggiunto un numero che serve a rappresentare l'entità di informazioni desiderate: uno rappresenta un livello minimo, che viene utilizzato se non si specifica alcun numero, mentre un valore più grande rappresenta più informazioni.
`-dm`	Permette di ottenere informazioni diagnostiche riferite all'allocazione e alla deallocazione di memoria.
`-q`	L'opzione '**`-q`**' può avere un argomento, ma se usata da sola, fa in modo che '**`exim`**' esegua una scansione (una soltanto) dei messaggi in coda, tentando di consegnare ogni messaggio trovato al suo interno. La scansione non segue un ordine preciso e alla sua conclusione '**`exim`**' termina di funzionare. Questa opzione può essere usata solo da un amministratore.
`-q`*tempo*	L'opzione '**`-q`**', seguita dall'indicazione di una durata temporale, fa in modo che '**`exim`**' esegua una scansione della coda in modo ripetitivo, a intervalli della durata specificata dall'argomento. In questo modo, '**`exim`**' deve continuare a funzionare a tempo indeterminato. Questa opzione, con argomento, viene usata preferibilmente per l'avvio di '**`exim`**' come demone, in modo tale che possa prendersi cura sia del servizio SMTP che della verifica della coda. L'intervallo specificato in questo modo, determina in pratica il tempo minimo che può essere indicato nella configurazione dei tentativi ripetuti.
`-v`	È sinonimo di '**`-d1`**' (diagnosi di livello minimo).

Come accade spesso nei sistemi Unix, l'eseguibile '**`exim`**' può essere avviato utilizzando nomi diversi che definiscono implicitamente l'uso di opzioni determinate, che potrebbero essere difficili da ricordare. Non sempre i pacchetti di Exim includono tutti i collegamenti

che potrebbero essere utili. Vale quindi la pena di riassumere quelli più comuni, che potrebbero essere realizzati utilmente se mancano nel proprio pacchetto.

Directory	Descrizione
'`/usr/lib/sendmail`', '`/usr/sbin/sendmail`'	Questi due collegamenti sono praticamente indispensabili se si vogliono utilizzare gli MUA comuni, cioè i programmi come Mailx ('**`mail`**'), che per l'invio dei messaggi si avvalgono proprio dell'eseguibile '**`sendmail`**'. Exim riconosce molte delle opzioni di Sendmail e in tal modo è possibile usare tali collegamenti. Secondo la logica dei sistemi GNU, l'eseguibile '**`sendmail`**' dovrebbe trovarsi esclusivamente nella directory '`/usr/sbin/`', ma per rispettare la tradizione è meglio aggiungere anche l'altro ('`/usr/lib/sendmail`') perché si vedono ancora script che fanno affidamento su questo.
'`/usr/bin/mailq`'	Quando '**`exim`**' viene avviato con il nome '**`mailq`**', permette di conoscere lo stato della coda, come se fosse stato avviato con l'opzione '**`-bp`**'.
'`/usr/bin/runq`'	Quando '**`exim`**' viene avviato con il nome '**`runq`**', fa in modo che '**`exim`**' esegua una singola scansione della coda, cercando di inviare i messaggi rimasti in attesa. Ciò, in pratica, come se fosse stato avviato con l'opzione '**`-q`**'.

Code e registri

Molto probabilmente (dipende da come è stato configurato in fase di compilazione), la directory '`/var/spool/exim/`' si articola in varie sottodirectory destinate a contenere informazioni variabili di vario tipo, tra cui le code dei messaggi e i file delle registrazioni.

La directory '`input/`' contiene precisamente i file delle code. Per ogni singolo messaggio che venga messo in attesa, si formano almeno due file: uno che termina con la sigla '`-D`' (*data*), che contiene il corpo del messaggio, e uno che termina con la sigla '`-H`' (*head*), che contiene le altre informazioni. Se un messaggio è diretto a diversi destinatari, la sua consegna può richiedere molto tempo e l'annotazione delle destinazioni presso cui è stato recapitato con successo. In tal caso viene creato un terzo file, che termina con la sigla '`-J`' (*journal*), all'interno del quale si annotano gli indirizzi già raggiunti.

Se un messaggio finisce in coda, ci deve essere un motivo. Nella directory '`msglog/`' vengono annotati file con gli stessi nomi utilizzati per i dati in coda, senza sigle finali, contenenti l'elenco degli insuccessi accumulati durante i vari tentativi ripetuti.

Se si decide di intervenire in modo brutale nei file delle code, cancellandoli, ci si deve ricordare di eliminare anche i file corrispondenti della directory '`msglog/`'.

Naturalmente sono disponibili anche dei file di registrazioni veri e propri, che potrebbero trovarsi in '`/var/spool/exim/log/`', oppure, più convenientemente, in '`/var/log/exim/`'. Si tratta di tre file: '`mainlog`', '`rejectlog`', '`processlog`' e '`paniclog`'. Il significato di questi nomi dovrebbe essere intuitivo: '`mainlog`' è l'archivio principale delle operazioni compiute, in cui si segnalano l'arrivo e la consegna di ogni messaggio; '`rejectlog`' registra le informazioni sui messaggi il cui transito è rifiutato in funzione della configurazione; '`processlog`' serve a segnalare l'effetto della ricezione di alcuni segnali (come '**`SIGHUP`**' e '**`SIGUSR1`**'); infine, '`paniclog`' permette di annotare le situazioni di errore che Exim non riesce a gestire.

Attraverso l'opzione '**`log_level`**' del file di configurazione, è possibile definire il livello di dettaglio delle informazioni che appaiono nel file delle registrazioni. Il valore predefinito corrisponde comunque a un buon livello di dettaglio.

Con l'opzione '**`preserve_message_logs`**', attivandola, è possibile evitare la cancellazione dei file delle registrazioni collocati nella directory '`msglog/`'. Ciò può essere utile solo nel caso in cui si volesse fare un controllo approfondito degli errori che si verificano durante i vari tentativi di consegna.

Archiviazione dei file delle registrazioni

Le distribuzioni GNU dovrebbero essere organizzate per gestire in modo elegante l'archiviazione dei file delle registrazioni, spezzando i file in parti che contengono periodi relativamente brevi, solitamente distinte attraverso un'estensione numerica progressiva che indica l'età relativa del file.

Exim fornisce un proprio script per svolgere questo compito, '**`exicyclog`**', che può essere usato quando la propria distribuzione GNU non dovesse già provvedere per conto proprio.

Per avviarlo, si potrebbe mettere un'istruzione come quella seguente nel file '`/etc/crontab`' (ammesso che lo script si trovi nella directory '`/usr/sbin/`').

```
01 0 * * * root /usr/sbin/exicyclog
```

Riferimenti

- *Exim*
  http://www.exim.org/

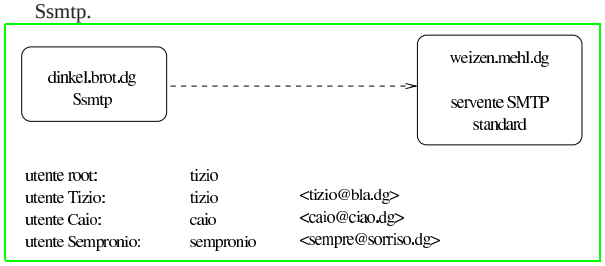
[1] **Exim** GNU GPL

[2] La directory in questione deve avere i permessi 1777_8, altrimenti non può funzionare il sistema di blocco dei file (*lock*) e in pratica i messaggi non vengono recapitati.

Ssmtp

Ssmtp [1] è un sistema molto semplice per l'invio della posta elettronica a un servente SMTP. Lo scopo di questo programma è quello di sostituire l'MTA tipico, quando non è necessaria la gestione della consegna locale (MDA) e si può fare affidamento continuamente su un MTA esterno, contattato attraverso il protocollo SMTP.

Figura u53.1. Esempio di una situazione in cui viene utilizzato Ssmtp.



Per comprendere il funzionamento di Ssmtp è necessario partire da un esempio, come si vede nella figura u53.1. Nell'elaboratore *dinkel.brot.dg* è stato installato Ssmtp; l'elaboratore *weizen.mehl.dg* offre un servente SMTP a cui *dinkel.brot.dg* può accedere. Nell'elaboratore *dinkel.brot.dg* sono registrati tre utenti: Tizio, Caio e Sempronio, ognuno dei quali riceve la propria posta elettronica presso una casella remota (come indicato nella figura stessa); inoltre, è Tizio che svolge anche il ruolo di amministratore.

Ssmtp non è un demone e viene avviato solo quando serve. In quel momento, contatta il servente SMTP indicato nella sua configurazione e gli affida il messaggio. Pertanto, il servente in questione deve essere sempre disponibile, ovvero deve essere sempre disponibile la comunicazione verso quell'elaboratore remoto. Da questo si comprende che Ssmtp non può essere usato in un elaboratore che si collega alla rete esterna solo saltuariamente.

In generale, i messaggi destinati all'elaboratore locale non si possono consegnare, ma anche se non si intende usare un programma come Mailx ('**`mail`**') non bisogna dimenticare che in un sistema Unix standard vengono generati automaticamente dei messaggi che spesso sono diretti all'amministratore, per aggiornarlo sullo svolgimento di operazioni periodiche o per avvisarlo di qualunque tipo di problema. In altri termini, un sistema Unix non può fare a meno di un programma che consenta almeno l'invio dei messaggi di posta elettronica. Ssmtp è in grado di ridirigere questi messaggi a un indirizzo di posta elettronica, che però deve esterno all'elaboratore locale.

Quando un utente invia un messaggio, è necessario che Ssmtp sia in grado di modificare il campo '**`From:`**', in modo che appaia un indirizzo di posta elettronica valido; diversamente, se apparisse l'indirizzo dell'elaboratore locale (*dinkel.brot.dg* nell'esempio), non potrebbe ricevere alcuna risposta dal suo interlocutore.

Configurazione

Ssmtp utilizza solo due file per la configurazione. Si tratta di '`/etc/ssmtp/ssmtp.conf`' e di '`/etc/ssmtp/revaliases`'. È sufficiente mostrare degli esempi compatibili con quanto visto in figura u53.1 per comprendere l'uso delle direttive di questi file. Si comincia con '`/etc/ssmtp/ssmtp.conf`', dove i commenti relativi alle direttive non utilizzate sono rimasti in inglese:

```
# /etc/ssmtp/ssmtp.conf

# La persona che riceve i messaggi diretti agli utenti con UID inferiore a 10
root=tizio@bla.dg

# Il servente SMTP per l'invio dei messaggi.
mailhub=weizen.mehl.dg

# Where will the mail seem to come from?
#rewriteDomain=dinkel.brot.dg

# Il nome completo del nodo locale
hostname=dinkel.brot.dg

# Set this to never rewrite the "From:" line (unless not given) and to
# use that address in the "from line" of the envelope.
#FromLineOverride=YES
```

Come si vede, i messaggi diretti a *root@localhost* vengono modificati e inviati a *tizio@bla.dg*; l'invio dei messaggi avviene facendo uso del servizio offerto da *weizen.mehl.dg*; il nome dell'elaboratore locale è *dinkel.brot.dg*.

Quanto visto fino a questo punto basta per inviare i messaggi, ma non è sufficiente a modificare il campo '**`From:`**', perché si tratta di un compito affidato alla configurazione con il file '`/etc/ssmtp/revaliases`':

```
# /etc/ssmtp/revaliases

root:           tizio@bla.dg
tizio:          tizio@bla.dg
caio:           caio@ciao.dg
sempronio:      sempre@sorriso.dg
```

Si può osservare che al nominativo-utente si abbina l'indirizzo di posta elettronica appropriato, compreso il caso di '**`root`**'.

Ricezione della posta elettronica

«

La ricezione della posta elettronica è un'attività al di fuori della competenza di Ssmtp, pertanto la si deve prelevare attraverso il protocollo POP3 o un altro simile, ma questo direttamente attraverso il programma utilizzato per la lettura della stessa. In altri termini, non si può usare Fetchmail (39.11.3) e nemmeno altri programmi che rinviano i messaggi prelevati nel sistema di recapito locale.

Considerazioni sulla sicurezza

«

L'utilizzo di Ssmtp si giustifica solo quando si vuole evitare di dovere gestire un servente SMTP standard, con tutti i problemi di sicurezza che ciò comporta. Inoltre, si evita di avere in funzione continuamente il programma, riducendo così le risorse elaborative richieste.

[1] **Ssmtp** GNU GPL