

TUTORIAL DE SENDMAIL

Este documento fue realizado en la UJI (Universidad Jaume I de Castellon) durante el transcurso del curso universitario 97-98, en la asignatura Interconexion de Sistemas Abiertos. Se permite la libre distribucion o copia de este documento. Si tienes cualquier duda, critica o sugerencia, puedes contactar con el autor, que soy yo:

David Rubert Viana (al004260@alumail.uji.es)

1. Introducción

1.1 Función y características de sendmail

1.2 Situación y descripción de los programas y ficheros de soporte utilizados por sendmail

2. Funciones de sendmail

2.1 Sendmail como agente de usuario

2.2 Sendmail como agente de transporte

2.3 La cola de correo

2.4 Los Alias en sendmail

2.4.1 El fichero /etc/aliases

2.4.2 El fichero .forward

2.4.3 Las listas de correo

2.5 Sendmail en modo traza

2.6 Parámetros más comunes de la línea de comandos en sendmail

3. El fichero de configuración de sendmail

3.1 Función y organización del fichero

3.2 Macros

3.2.1 Los nombres largos

3.2.2 Los condicionales

3.2.3 Las macros internas de sendmail

3.3 Clases

3.3.1 Clases internas más comunes

3.3.2 Ejemplo del uso conjunto de macros y clases

3.4 Opciones

3.4.1 Opciones más comunes

3.4.2 Modos de reparto

3.4.3 Timeouts

3.5 Precedencia de mail

3.6 Trusted Users

3.7 Formato de las cabeceras

3.8 Las reglas de reescritura

3.8.1 Los conjuntos de reglas de reescritura (Rewriting rule sets)

3.8.2 La parte izquierda y derecha de una regla (Left-hand-side & Right-hand-side)

3.8.3 Los separadores de tokens

3.8.4 Como probar nuestras propias reglas de reescritura

3.8.5 Ejemplos de reglas de reescritura extraídas del fichero de configuración de LINUX

3.9 Definición de mailers

3.9.1 Campos más comunes en la definición de mailers

3.9.2 Parámetros más comunes en la definición de mailers

3.9.3 Ejemplos de definición de mailers

3.9.4 La definición del mailers en el fichero de configuración de LINUX

4. Links interesantes

1. Introducción

Este tutorial intenta dar una visión global de sendmail v8.8.7 (Berkeley), programa de distribución gratuita creado (y actualmente revisado) por Eric Allman. La versión actual es la v8.8.8, aunque nosotros trataremos la anterior versión. Se intentará explicar claramente la mayoría de sus funciones sin entrar muy a fondo en cada una de ellas, pero dando ejemplos comprensivos. También veremos como podemos configurar sendmail para su funcionamiento en un sistema específico.

1.1 Función y características de sendmail

Existe una gran variedad de programas de correo electrónico que proveen al usuario de una aplicación para la creación y envío de mail. Estos programas son los llamados Agentes de Usuario o MUA (Mail User Agent), y su propósito es el aislar al usuario de los Agentes de Transporte o MTA (Mail Transport Agent), que son los encargados de transferir los mails a su correcto destino.

Sendmail es el agente de transporte de correo más común de Internet (en los sistemas UNIX). Aunque actúa principalmente como MTA, también puede ser utilizado como MUA (aunque no posee interfaz de usuario). Las misiones básicas de sendmail son las siguientes:

- Recogida de mails provenientes de un Mail User Agent (MUA) como pueden ser elm, Eudora o pine; o provenientes de un Mail Transport Agent (MTA) como puede ser el propio sendmail.
- Elección de la estrategia de reparto de los mails, basándose en la información de la dirección del destinatario contenida en la cabecera:
 - Si el mail es local en nuestro sistema, enviará el mail al programa de reparto local de mails.
 - Si el mail no es local, sendmail utilizará el DNS de nuestro sistema para determinar el host al que debe ser enviado el mail. Para transferir el mensaje,

- iniciará una sesión SMTP con el MTA de dicho host.
- Si no es posible mandar el mail a su destino (porque la maquina receptora esta desconectada, o va muy lenta), sendmail almacenará los mails en una cola de correo, y volverá a intentar el envío del mail un tiempo despues. Si el mail no puede ser enviado tras un tiempo razonable, el mail será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada mensaje llegue correctamente a su destino, o si hay error este debe ser notificado (ningún mail debe perderse completamente).
- Reformatear el mail antes de pasarlo a la siguiente máquina, según unas reglas de reescritura. Según el tipo de conexión que poseamos con una determinada máquina, o según el agente de transporte al que vaya dirigido el mail, necesitaremos cambiar los formatos de las direcciones del remitente y del destinatario, algunas lineas de la cabecera del mail, o incluso puede que necesitemos añadir alguna línea a la cabecera. Sendmail debe realizar todas estas tareas para conseguir la máxima compatibilidad entre usuarios distintos.
- Otra funcion muy importante de sendmail es permitir el uso de "alias" entre los usuarios del sistema; lo que nos permitirá (entre otras funciones) crear y mantener listas de correo entre grupos.
- Ejecución como agente de usuario (MUA). Aunque no posee interfaz de usuario, sendmail tambien permite el envío directo de mails a través de su ejecutable.

Todas estas características y muchas otras que posee el sendmail deben ser configuradas y variarán de unos sistemas a otros. Para configurarlas hacemos uso del fichero de configuración de sendmail. La revisión y modificación de este fichero es bastante complicada y necesita de una serie de conocimientos previos.

1.2 Situación y descripción de los programas y ficheros de soporte utilizados por sendmail

1. **/usr/sbin/sendmail** (**/usr/lib/sendmail** en algunos sistemas)

Ejecutable del sendmail.

2. **/etc/sendmail.cf** (**/usr/lib/sendmail.cf**)

Fichero de configuración de sendmail.

3. **/etc/aliases**

Fichero donde se almacenan los alias del sistema.

4. **/usr/bin/newaliases**

Reconstruye la base de datos de alias usada por sendmail, a partir del fichero aliases. Es un enlace simbólico a sendmail en el modo - bi.

5. /var/spool/mqueue

Es el directorio donde se almacena la cola de correo.

6. /usr/bin/mailq

Saca por pantalla el contenido de la cola de correo. Es un enlace simbólico a sendmail en el modo -bp.

2. Funciones de sendmail

2.1 Sendmail como agente de usuario

Como agente de usuario, sendmail lee por defecto de su entrada estándar hasta encontrar un end-of-file (EOF) o una línea que contenga únicamente un punto ("."). En ese momento manda una copia de ese mensaje a cada una de las direcciones destino (que se le habrán pasado previamente en la línea de comandos). Él mismo determinará la ruta a seguir (consultando el DNS si fuera necesario), basándose en el contenido de la dirección destinataria.

Sendmail determina cómo enrutar un mensaje (cuyo destino sea la dirección o direcciones que le habremos establecido), de acuerdo a la información que posee en su fichero de configuración (por defecto /usr/lib/sendmail.cf).

Ejemplo 1:

Este comando manda un mail a "alonso@xyz.com", sin título y cuyo cuerpo contiene únicamente "hola que tal".

```
$ sendmail alonso@xyz.com
hola que tal
..
$
```

Ejemplo 2:

Podemos también pasarle el cuerpo del mail mediante un fichero:

```
$ cat fichero | sendmail alonso@xyz
```

Ejemplo 3:

Podemos incluso pasarle los datos de la cabecera manualmente, mediante el parámetro -t:

```
$ sendmail -t alonso@xyz
From: al001010@alumail.uji.es
Date: ....
.....
```

2.2 Sendmail como agente de transporte

La utilización de sendmail como demonio en el sistema nos permite enviar y recibir correo SMTP. Para ello, sendmail se queda como proceso residente escuchando el puerto 25, admitiendo y realizando conexiones SMTP cuando sea necesario (es decir, actúa como un Agente de Transporte de Correo (MTA)). Cuando reciba o una petición de conexión, creará un proceso hijo que se encargará de ello, mientras el proceso padre seguirá escuchando el puerto 25.

Solo puede ser ejecutado en este modo por el superusuario del sistema.

Para utilizar sendmail como demonio, debemos iniciarlo en el arranque del sistema en uno de los scripts de inicio con los siguiente parámetros:

```
sendmail -bd -q15m
```

Con esta orden, sendmail actuará en modo background admitiendo conexiones por el puerto 25. La opción -q15m le indica que actualice la cola de correo cada 15 minutos (este parámetro puede variar).

El parámetro -bD es igual al -bg, pero actuando en foreground.

2.3 La cola de correo

Cuando el envío de un mail no puede alcanzar su destino porque nos rechazan la conexión (normalmente debido a que la máquina destinataria esté apagada o sobrecargada), éste debe ser almacenado en una cola de la máquina que manda el mail, para intentar de nuevo el envío más tarde.

La cola de correo se encuentra en /var/spool/mqueue. En este directorio se crean unos ficheros temporales para cada mail que se almacena. El formato de los ficheros en este directorio es el siguiente:

- d f - - - - - (ficheros donde se guardan los cuerpos de los mensajes, sin las cabeceras).
- q f - - - - - (ficheros donde se guardan la información necesaria para procesar los trabajos).
- t f - - - - - (ficheros temporales imagen de los ficheros qf cuando estos están siendo reconstruidos).
- x f - - - - - (fichero donde se almacena toda la información transmitida durante la apertura y cierre de una sesión).

Para visualizar el contenido de la cola usamos el comando "mailq", que es un link simbólico a el comando "sendmail -bp". Este producirá una lista con los identificadores de los mensajes, su tamaño, la fecha en la que el mensaje entró en la cola, el remitente y el destinatario.

Para procesar la cola de correo (automática o manualmente), se utiliza el comando:

sendmail -q<tiempo>

que procesa la cola de correo cada <tiempo> (solo puede ser ejecutado por el superusuario, y normalmente se carga junto con el sendmail en modo demonio en un script de inicio).

<tiempo> es un número seguido de caracteres. El carácter "s" significa segundos, "m" significa minutos, "h" significa horas, "d" significa días y "w" significa semanas. Si se omite <tiempo>, sendmail procesará la cola en ese momento

Ejemplo:

sendmail -q1h30m

Actualizará la cola de correo cada hora y media. Y es equivalente a sendmail -q90m.

Si lo que queremos es procesar la cola en un instante, utilizaremos el comando: sendmail -q

Pueden utilizarse otros parámetros especiales para procesar la cola:

- q|Substr Procesará únicamente aquellos trabajos que tengan <Substr> como subcadena de los identificativos de mail.
- qRSubstr Procesará únicamente aquellos trabajos que tengan <Substr> como subcadena de los destinatarios.
- qSSubstr Procesará únicamente aquellos trabajos que tengan <Substr> como subcadena de los remitentes.

2.4 Los Alias en sendmail

El uso de alias en correo electrónico, y gracias a sendmail, nos permiten:

- Tener nombres alternativos (nicknames) para usuarios individuales.
- Envío de correo a otras máquinas, aunque la dirección sea local.
- Listas de correo.

2.4.1 El fichero /etc/aliases

Existe un fichero de texto (/etc/aliases) donde es posible ver o modificar la base de datos actual de alias del sistema. Pero no será éste el fichero usado directamente por sendmail, sino que primero debe ser convertido y procesado con el comando "newaliases", que creará la base de datos de alias que sí podrá utilizar sendmail.

El formato del fichero (/etc/aliases) es el siguiente:

alias: recipient [recipient,.....]

alias	es el nombre de la persona a la que mandamos el mail (Debe ser de un usuario local en la máquina).
recipient	es el nombre al cual mandamos el mail. Puede ser un nombre de usuario, el nombre de otro alias, o una dirección e-mail completa conteniendo tanto

	el nombre de usuario como el nombre de la máquina (y su dominio). (Incluyendo el nombre de la máquina el mail puede ser mandado a una máquina remota)
[recipient,...]	Podemos añadir varios receptores para un mismo alias. Un mail mandado a ese alias sera recibido por todos los recipients. (Es una lista de correo (Mailing list)). Una particularidad es que aunque incluyamos el nombre de origen a la lista de correo, éste no recibirá el mail (a menos que establezcamos la opción de MeToo activada).

Ejemplo del fichero /etc/aliases:

```
# nombres especiales para los administradores del sistema
```

```
postmaster: rob
```

```
root: mike
```

```
# Reenvío del correo a una dirección remota
```

```
luisa: luisa@xyz.es
```

```
# lista de correo
```

```
clientes: pepe, juan@xyz, al001010@alumail.uji.es
```

```
# (Si se manda un mail a "clientes", una copia del mail se mandará para cada uno de los destinatarios)
```

2.4.2 El fichero .forward

A nivel de usuario, sendmail tambien permite a cualquier usuario normal crearse su propia tabla de alias. Ésta viene definida en el fichero .forward del \$HOME de cada usuario. Y tiene precedencia sobre el fichero /etc/aliases creado por el root.

2.4.3 Las listas de correo

Una función especial de sendmail para las listas de correo es la siguiente:

```
owner-<lista>: <nombre>
```

donde <lista> es el nombre de una lista de correo. La persona especificada en <nombre> es la responsable de la lista de correo identificada en <lista>. Esta función es util si, por ejemplo, sendmail tiene problemas mandando correo a alguno de los destinatarios de la lista de correo. Gracias a esta identificación del responsable de la lista, se le mandará un mensaje de error a juan indicándole que han habido problemas en el envio del mail.

Ejemplo:

```
clientes: pepe,juan@xyz,al001010@alumail.uji.es
```

```
owner-clientes: marcos
```

Una forma más fácil de mantener listas de correo sería la siguiente línea en /etc/aliases (o en el .forward, si no somos superusuarios):

```
pepe: "| sendmail 'cat <lista>'"
```

donde <lista> contiene todos los suscriptores actuales de la lista de correo. Con la tubería conseguimos que, al recibir un mail el usuario "pepe" se ejecute la línea de comandos que sigue a la tubería.

Las prioridades del programa que se ejecute, serán las mismas que las de el usuario que recibe el mail.

2.5 Sendmail en modo traza

El parámetro "-dx.l" establece sendmail en modo traza. La x indica la parte de sendmail de la que queremos hacer la traza. La l indica el nivel de detalle de la traza. Al ejecutar sendmail con este parámetro, al mandar correo a una dirección va explicando cada uno de los pasos que va realizando para conseguir mandar el mensaje.

Ejemplo: `sendmail -d21.20 al010100@alumail.uji.es`

Los tipos de traza más útiles son los siguientes:

- 21 REESCRITURA DE DIRECCIONES
 - d21.4 Muestra las direcciones reescritas.
 - d21.20 Muestra la parte izquierda y derecha de las reglas de reescritura.
- 37.1 Muestra las opciones utilizadas de sendmail.
- 35.9 Muestra todas las definiciones de macros.
- 32.1 Muestra las líneas de cabecera utilizadas.
- 0.15 Muestra las reglas de reescritura de direcciones y las definiciones de mailers utilizados.
- 8.8 Muestra la información del DNS accedida.

2.6 Parámetros más comunes de la línea de comandos en sendmail

-B<tipo>	Establece el tipo del cuerpo del mensaje a <tipo> (7BIT o 8BITMIME).
-ba	Modo Arpanet. Es el mismo modo que el standard (-bm), excepto que todas las líneas de entrada deben terminar con CR-LF, todas las líneas de salida terminarán con CR-LF, y que extrae el remitente leyendo el campo de la cabecera "From:".
-bd	Ejecutar el demonio de sendmail. Ver sendmail como agente de transporte

-bm	Envía el mail introducido por la entrada estandar hasta la dirección de destino. Puede ir seguido por cualquier otro parámetro o modo en la línea de comandos. Es el modo de trabajo estandar.
-bp	Visualizar el contenido de la cola de correo. Ver La cola de correo
-bi	Actualizar la base de datos de alias. Ver Los alias en sendmail
-bt	Modo test de direcciones. Se usa para testear el correcto funcionamiento de las reglas de reescritura de direcciones.
-bv	Modo verificación. Simplemente verifica la existencia de los nombres, no recoge ni manda los mensajes. Se usa normalmente para validar usuarios o listas de correo.
-C<fichero>	Uso de <fichero> como fichero de configuración de sendmail alternativo.
-dx.l	Ver Sendmail en modo traza
-f [address]	Establece la dirección fuente, es decir, quién envía el mail. Será válido solo si la dirección [address] se corresponde con el usuario que está ejecutando sendmail, o si éste es un "trusted user" definido en el fichero de configuración.
-h [cont]	Establece el contador de "hops" a [cont]. El contador de "hops" se incrementa cada vez que un mismo mail es procesado. Cuando se supera el límite, se abandona el envío de ese mail, y se le devuelve al emisor con un mensaje de error "too many hops". Por defecto, sendmail determina la cuenta de hops de un mensaje contando las líneas "Received:" del mensaje.
-F name	Establece el nombre completo del usuario a name.
-Mx valor	Establece la macro "x" al valor especificado.
-n	No utilizar los alias.
-ox valor -O option=val	Establece la opción "x" al valor especificado.
-R <tipo>	Establece el tipo de devolución que hará sendmail si el mail no puede ser mandado a su destino. El parámetro "full" indicará que se devuelva todo el mail, y "hdrs" indicará que se devuelva solo la cabecera.
-t	Extrae el o los destinatarios de el mismo mensaje. Leerá las direcciones de "To:", "Cc:" y "Bcc:". Cualquier dirección de la línea de comandos será suprimida (no recibirá el mail).

-q[intervalo]	Tiempo de procesamiento automático de la cola de correo. Ver La cola de correo
---------------	--

3. El fichero de configuración de sendmail

3.1 Función y organización del fichero

En el fichero de configuración de sendmail se encuentra toda la información relativa a nuestro sistema que sendmail utilizará cada vez que sea ejecutado. Este fichero no puede ser compilado con el ejecutable, ya que debe permitir la portabilidad de la configuración entre sistemas. El lenguaje utilizado en este fichero de configuración está pensado para que dicho fichero sea procesado lo más rápidamente posible. En las antiguas máquinas, esta velocidad de procesamiento era absolutamente necesaria, ya que dicho fichero debe ser procesado cada vez que ejecutamos sendmail. Debido a esto, el lenguaje utilizado es muy críptico y poco comprensible. Las nuevas versiones de sendmail, y gracias a la potencia de las nuevas tecnologías, se consigue hacer más comprensible el lenguaje utilizado para configurar sendmail, utilizando el procesamiento m4 (aunque no vamos a tratarlo en este tutorial).

Este fichero de configuración podemos encontrarlo en /etc/sendmail.cf. Con respecto a su contenido, podemos organizarlo en las siguientes secciones:

Macros Generales	Información específica sobre la red local (Nombre del dominio, nombre oficial de la máquina, ...).
Clases	Definición de grupos de nombres de máquinas o nombres de dominios.
Opciones	Definición de las opciones de sendmail.
Precedencia de mensajes	Definición de los valores de la precedencia de mensajes usada por sendmail.
Usuarios de confianza (Trusted users)	Definición de usuarios a los que le proporcionamos la opción de que puedan omitir su dirección cuando mandan mails (Suele causar problemas de seguridad).
Formato de las cabeceras	Definición del formato de las cabeceras que sendmail inserta en los mails (no debe ser modificada).
Reglas de reescritura de direcciones	Definición de las reglas usadas para reescribir las direcciones de mail.
Definición de mailers	Definición de las instrucciones que usa sendmail para invocar los programas de reparto del correo.

Para implementar estas secciones, cada una de las líneas del fichero de configuración comienzan por una de estas letras, según la función que realizan:

S	Definición de un conjunto de reglas de reescritura
R	Definición de una regla de reescritura
D	Definición de una macro
C	Definición de una clase
F	Extraer los tokens de una clase de un fichero
O	Definición de opciones
H	Formato del texto de la cabecera
T	Usuarios de confianza (Trusted Users)
P	Precedencia y prioridad para los mensajes
M	Definición de un agente de transporte

3.2 Macros

Las macros nos sirven para definir una información específica sobre la red local o nuestra máquina. Las clases nos permiten definir grupos de nombres de máquinas o nombres de dominios. La utilidad de estas dos funciones es que nos permiten exportar las configuraciones de sendmail a otros sistemas similares, sin tener que cambiar todo el fichero de configuración, sino simplemente cambiando los valores de algunas macros o clases.

Una macro puede definirse de la siguiente forma: `Dxval`

La primera "D" indica que estamos usando el comando "definir macro". La "x" es el nombre de la macro que estamos definiendo. Y "val" es el valor asignada a esa macro.

Podemos utilizar las macros, por ejemplo en una regla de reescritura. Por ejemplo, con esta definición de macro:

```
DDxyz.com      (Definición de un dominio)
```

poniendo `$D` en la lhs o la rhs de una regla de reescritura, estaremos haciendo referencia a el dominio "xyz.com".

3.2.1 Los nombres largos

Un nombre de macro, de clase o de opción de sendmail puede ser un simple carácter ASCII o un nombre largo. Las antiguas versiones de sendmail solo soportan nombres cortos, pero con las nuevas versiones, y gracias al aumento de velocidad de procesamiento de las máquinas se ha incluido la opción de definir nombres largos. Para definir un nombre largo de macro, clase u opción utilizamos la siguiente sintaxis:

```
D{LongName}   Definición de una macro de nombre largo.
${LongName}   Referencia a una macro de nombre largo.
```

Las letras minúsculas están reservadas para las macros internas de sendmail, por tanto, para definir una macro de usuario utilizaremos una letra mayúscula o un nombre largo que comience por una letra mayúscula.

3.2.2 Los condicionales

Otra utilidad de utilizar las macros en las reglas de reescritura es el uso de condicionales, con el siguiente formato:

```
$?x text1 $| text2 $
```

indicamos que usará text1 si la macro \$x está establecida, y si no usará text2 (opcional).

3.2.3 Macros internas

Las macros internas más importantes de sendmail son las siguientes:

\$a	Fecha de origen del mail en formato RFC 822 (Extraída de la línea "Date:").
\$b	Fecha actual en formato RFC 822
\$c	Contador del número de hops. (Cuenta el número de líneas "Received:").
\$d	Fecha actual en formato UNIX.
\$f	Dirección del remitente extraída de la cubierta (envelope).
\$g	Dirección del remitente relativa al destinatario. (Es decir, si \$f es "luis", \$g será "luis@host.domain").
\$h	Máquina destinataria.
\$i	Identificador de la cola.
\$j	Nombre oficial del dominio.
\$m	Nombre del dominio, menos la máquina. (Normalmente \$j es equivalente a \$w.\$m).
\$n	Nombre del demonio encargado de los mensajes erróneos (Por defecto: MAILER-DAEMON).
\$o	Separadores.
\$p	Identificador del proceso de sendmail.
\$q	Formato por defecto de la dirección del destinatario. Ejemplos: Dq\$g\$?x (\$x)\$ user@host (Nombre completo del usuario) Dq\$?x \$x \$. <\$g> Nombre del usuario <user@host>
\$s	Nombre de la máquina del remitente.
\$w	Nombre de la máquina local.

\$u	Nombre de usuario del destinatario.
\$x	Nombre completo del remitente.
\$_	Dirección validada del remitente.

3.3 Clases

Las clases son grupos de tokens (por ejemplo de nombres de máquinas o nombres de dominios) que nos sirven para hacer comparaciones en las LHS de una regla de reescritura. Para definir una clase, utilizamos:

CXvalor1 valor2

esta expresión añade el valor1 y el valor 2 a la clase \$=X (Siempre se van añadiendo los nuevos tokens metidos en una clase a los que ya estaban).

FX /dir/fichero

extrae los valores de la clase \$=X del fichero /dir/fichero.

El nombre de una clase puede ser un carácter ASCII o una nombre largo entre {corchetes}. Las letras minúsculas y caracteres especiales están reservadas uso del sistema. Para definir una clase de nombre largo:

C{Longname} Añadir tokens a una clase de nombre largo.
 \$={Longname} Comparación de un token con una clase de nombre largo.

Ejemplo 1:

CHmarret llevant gregal

esta expresión es equivalente a:

CHmarret
 CHllevant
 CHgregal

Lo que hacen es añadir a la clase H los nombres de máquinas "marret", "llevant" y "gregal".

3.3.1 Clases internas más comunes

\$=m	Conjunto de dominios por los que se conoce la máquina actual.
\$=t	Conjunto de usuarios de confianza (trusted users).
\$=w	Conjunto de nombre por los que se conoce la máquina actual.

3.3.2 Ejemplo del uso conjunto de macros y clases

Cambio de todos los nombres por los que es conocido el dominio de una máquina, por el nombre oficial del dominio:

Regla LHS	Regla RHS
R\$+<@\$=m>	\$1<@\$j>

La regla LHS busca tokens que sean miembros de la clase D (definida por el usuario)

La regla RHS cambia el token anterior por la macro j (con el nombre oficial del dominio)

3.4 Opciones

Existe un gran número de opciones globales que pueden definirse en el fichero de configuración directamente. Nos permiten definir por ejemplo la localización de los ficheros utilizados por sendmail, los modos de operación por defecto, o la asignación de valores a determinadas características. La sintaxis es la siguiente:

O opt=val

Establece la opción "opt" al valor "val". La sintaxis de las antiguas versiones es la siguiente:

Ooval

Establece la opción "o" al valor "val".

Los valores que podemos asignar a una opción pueden ser enteros, una cadena, un booleano, o un intervalo de tiempo.

3.4.1 Opciones más comunes

Las opciones más comunes son las siguientes (entre corchetes viene indicado su nombre corto si lo posee):

AliasFile=fichero [A]	Usa "fichero" como el fichero de alias del sistema. Por defecto es el fichero "/etc/aliases".
AutoRebuildAliases [D]	Si está actitava, reconstruye automáticamente la base de datos de los alias cuando sea necesario (si fuera posible).
ConnectionRateThrottle=N []	No permite recibir más de N conexiones por segundo al demonio sendmail. Por defecto es 0 (sin límite).
DeliveryMode=x [d]	Modos de reparto. Ver apartado 3.4.2.

ForwardPath=path [J]	Path donde sendmail buscará los ficheros ".forward" de cada uuario. Por defecto es "\$z/.forward".
HostsFile=path []	Path de la base de datos de hosts. Por defecto "/etc/hosts".
IgnoreDots [i]	Ignorar los puntos en los mensajes entrantes. Por defecto esta desactivado.
MaxDaemonChildren=N []	Máximo números de demonios sendmail procesando correo entrante al mismo tiempo. Si se supera se rechazarán las conexiones de correo entrante.
MaxHopCount=N [h]	Máximo número de veces que un mensaje es procesado.
MaxQueueRunSize=N	Máximo número de trabajos procesados en la ejecución de la cola. Si no se establece, no hay límite.
MeToo [m]	Mandar el mail al usuario incluso si está en una expansión de alias (en una lista de correo).
QueueDirectory=dir [Q]	Uso de "dir" como el directorio de la cola de correo.
Timeout.type=timeout	Establece los valores de timeout para cada tipo. Ver apartado 3.4.4.

3.4.2 Modos de reparto

Existen cuatro modos de reparto en sendmail (entre paréntesis el valor de la opción):

- **Modo inmediato** (i). Sendmail realiza las conexiones de modo interactivo, es decir, no devuelve el control del sistema hasta que ha recibido un resultado de la operación ejecutada. Es un método que ralentiza mucho el sistema, poco recomendado.
- **Modo background** (b). Cada vez que sendmail reciba una petición de conexión, generará un proceso hijo que se encargará del mensaje. Es el modo más común.
- **Modo cola** (q). Mete el mail directamente en la cola de correo, para ser procesado más tarde. Antes de meterlo en la cola, consulta el servidor de nombres para verificar las direcciones de remitente y destinatario.
- **Modo postergado** (d). Parecido al anterior, con la única diferencia que sendmail no consulta el servidor de nombres. Útil en sistemas con servidores de correo cuya conexión a INTERNET no es permanente, o aquellos en que la consulta al DNS tiene un coste elevado. La cola será procesada en el momento en que sendmail se conecte a INTERNET.

3.4.3 Timeouts

Son los tiempos máximos permitidos para establecer una conexión con otra máquina. Si alguno de ellos es rebasado, se rechazará la conexión y se almacenará el mail en la cola de correo para ser procesado más tarde. Se divide en varios subapartados:

- connect. Tiempo de espera máximo para el establecimiento de la conexión SMTP.
- initial. Tiempo de respuesta máximo para el mensaje de bienvenida inicial.
- helo. Tiempo de respuesta máximo para el comando HELO o EHLO.
- mail. Tiempo de respuesta máximo para un comando MAIL.
- rcpt. Tiempo de respuesta máximo a un comando RCPT.
- datainit. Tiempo de respuesta máximo a un comando DATA.
- datablock. Tiempo de espera máximo para la lectura del cuerpo del mensaje.
- datafinal. Tiempo de espera máximo a el punto que termina un mensaje.
- rset. Tiempo de respuesta máximo a un comando RSET.
- quit. Tiempo de respuesta máximo a un comando QUIT.
- misc. Tiempo de respuesta máximo a un comando misceláneo.
- command. Actuando como servidor SMTP, tiempo de espera de otro comando.
- ident. Tiempo de espera para la respuesta a una consulta IDENT.

Para el establecimiento del tiempo a cada un de estos apartados, usamos el mismo método que usamos en la cola de correo. Por ejemplo, para asignar un tiempo de establecimiento de conexión máximo de 3 minutos y medio utilizamos:

```
O Timeout.connect=3m30s
```

3.5 Precedencia de mails

Mediante esta propiedad asignamos prioridad a los mensajes según entren en la cola. Cuanto más alto sea el número de precedencia de un mensaje, antes será procesado. Para definir la precedencia de los mails utilizamos la siguiente sintaxis:

```
Pnombre=num
```

Cuando el campo "Precedence:" incluido en la cabecera de un mensaje posea el valor "nombre", este mensaje poseerá una precedencia de valor "num". La precedencia asignada por defecto es 0.

Un mail puede poseer una precedencia negativa, lo que nos indicará que el mensaje no deberá ser devuelto cuando se produzca un error en el envío. Esto es útil para las listas de correo, o para los mensajes de propaganda.

Ejemplo:

```
Pfirst-class=0  
Pspecial-delivery=100  
Plist=-30  
Pbulk=-60
```


Pjunk=-100

3.6 Trusted Users

La definición de usuarios de confianza o "Trusted Users" en nuestro fichero de configuración implica que dichos usuarios tendrán la capacidad de mandar mails con nombres falsos, sin que sendmail avise al receptor del mail.

Para definirla:

```
Tnom1 [nom2 nom3 ...]
```

Donde nom1 es el nombre del usuario de confianza.

Ejemplo:

Los usuarios de confianza más comunes de un sistema:

Trout

Tdaemon

Tuucp

No es recomendable tener muchas (mejor ninguna) personas en esta lista, ya que da lugar a abusos en el envío de mails falsos. La forma que tiene un usuario de cambiar la identidad de un mail es usando el parámetro -f en la línea de comandos, por ejemplo:

```
cat fichero | sendmail -f fulano@xyz usuario
```

Este comando manda un mail a "usuario", con el nombre de remitente falso "fulano@xyz". Si el usuario que manda el mail no lo tenemos como Trusted User en el fichero de configuración de sendmail, al leer el mail el remitente, tendrá una cabecera como ésta:

```
Return-Path: <fulano@xyz>
```

```
Date: lo-que-sea
```

```
X-Authentication-Warning: <Usuario real que manda el mail> set sender to fulano@xyz  
using -f
```

Esta última línea será la que nos indica quién manda realmente el mail, y es la línea que no nos aparecerá si ese mismo usuario está como Trusted User en el sendmail.cf.

3.7 Formato de las cabeceras

Formato de las líneas de la cabecera de un mail insertadas por sendmail. Definen qué líneas de cabecera deben ser incluidas basándose en los parámetros especificados por el mailer y las líneas ya existentes en el mensaje. La sintaxis es la siguiente:

```
H[?mflags?]{hname:htemplate
```

hname indica el valor de la línea de cabecera.

htemplate hace referencia a una macro interna de sendmail.

El parámetro mflags puede estar o no incluido en la línea. Si está incluido, indica que este parámetro debe estar incluido en la definición del mailer para que la línea se incluya en el mail. Para mayor información ver los parámetros usados en la definición de mailers.

Ejemplos:

```
H?P?Return-Path: <$g>
```

La línea se incluirá en el mail si el mailer posee el parámetro P. La línea que se incluirá será la siguiente:

```
Return-Path: <Direccion@del.remitente>
```

3.8 Las reglas de reescritura

La principal utilidad de las reglas de reescritura es la modificación de direcciones de correo según convenga a nuestro sistema. Las direcciones deben poder ser modificadas, porque pueden ser especificadas de muchas formas, según el tipo particular de envío que se utilice.

Por ejemplo, si queremos mandar un mail a `juan@xyz`, y si únicamente disponemos de acceso vía modem a la máquina `xyz`, el mensaje deberá ser mandado con software UUCP. Este software necesita que las direcciones sean expresadas de la forma: `xyz!juan`, por lo que las reglas de reescritura de sendmail serán las encargadas de transformar la dirección.

Otra utilidad de las reglas de reescritura es la detección de errores en las direcciones introducidas. Esta detección se encarga de buscar y rechazar los errores en los mails locales, consiguiendo así que no se propague el error por toda la red. Por ejemplo, mandar un mail sin nombre de usuario: `@xyz` será detectado localmente, en vez de mandarlo y dejar que lo detecte la máquina `xyz`.

3.8.1 Los conjuntos de reglas de reescritura (Rewriting rule sets)

Para utilizar estas reglas, existen los llamados conjuntos de reglas de reescritura, que son una agrupación de reglas. Cada uno de estos conjuntos es similar a una subrutina de un lenguaje de programación. Para definir un conjunto de reglas utilizaremos la letra "S" al comienzo de una línea del fichero de configuración.

Por ejemplo, `S0` marca el comienzo de la declaración de las reglas que conforman el conjunto de reglas de reescritura 0.

La declaración de estos conjuntos de reglas pueden aparecer en cualquier orden dentro del fichero de configuración, es decir, no importa si primero declaramos la `S1`, la `S2`, la `S0` o cualquier otra.

Todas las líneas por debajo de `S0`, serán reglas pertenecientes a `S0`. Cuando aparezca una nueva `S`, comenzará un nuevo conjunto de reglas de reescritura.

Cada conjunto de reglas de reescritura tiene una función específica dentro del sendmail. Los conjuntos de reescritura S0 a S9 están reservados para uso interno de sendmail. Los demás conjuntos de reglas de reescritura serán subrutinas llamadas por estas 6 primeras, o por un mailer que necesite reescribir una determinada dirección en un determinado formato.

Todos los mails pasan primero por el ruleset S3, que se encarga del pre-procesamiento. Pone las direcciones de manera estandar para el correcto manejo de las direcciones por S0.

S0 se encarga de decidir el mailer correspondiente para un determinado mail

Según el mailer especificado por S0, a cada mail se le aplicaran el conjunto de reglas de reescritura definidos para ese mailer.

S4 se encarga de deshacer los cambios realizados por S3. Las direcciones se quedan de la manera correcta para que el mail pueda ser enviado.

En la siguiente tabla vemos un resumen de las funciones de cada conjunto de reglas de reescritura:

S0	Es el conjunto de reglas de reescritura encargada del reparto de mensajes. Manipula la dirección del destinatario en la cubierta del mail. Necesita una dirección en formato interno: usuario<@máquina.dominio> Elimina de las direcciones la máquina local (si es una dirección local). Decide cómo enviar el mail basándose en el formato de la dirección.
S1	Conjunto de reglas de reescritura del remitente usado por todos los mailers (normalmente vacío).
S2	Conjunto de reglas de reescritura del destinatario usado por todos los mailers (normalmente vacío).
S3	Se encarga del pre-procesamiento. Pone el foco <> en el correcto lugar de la dirección, que sera usado posteriormente por S0. Es decir: usuario<@máquina.dominio> Estandariza las direcciones para que sean del estilo correcto (con @).
S4	Se encarga del post-procesamiento. Su misión es: Quitar el foco <>. Se deshacen las transformaciones de formato realizadas por S3. Convierte la dirección a la forma exterior si ha de salir de la red local
S5	Reescribe direcciones despues del aliasing local, pero antes del reparto local. Todas las direcciones de usuario para un mailer local que no empiezan con una @, pasan por S5. Permite la selección de un mailer, host, user diferente despues del aliasing.
S96	Es la segunda mitad del conjunto de reglas S3 para algunos ficheros de configuración.

3.8.2 La parte izquierda y derecha de una regla (Left-hand-side & Right-hand-side)

Dentro de estos conjuntos de reglas de reescritura, podemos encontrarnos con muchas reglas individuales, o con ninguna. Estas reglas interiores comienzan con la letra "R" y siguen el siguiente patrón:

S0

Rlhs <TAB (1 o más)> **rhs** <TAB (1 o más)> **comentario(opcional)**

lhs es una abreviatura de left-hand-side, o parte izquierda. Este es el modelo o patrón en el que buscaremos la coincidencia de las direcciones que se introduzcan.

rhs es una abreviatura de right-hand-side, o parte derecha. Si la entrada ha coincidido con la parte izquierda, ésta será reescrita al formato que posea esta parte derecha.

Ejemplo:

S0

Rhello **bye**

En este ejemplo, al aplicar el conjunto de reglas de reescritura S0, siempre que se encuentre con la palabra "hello" se reescribirá con "bye".

Como podemos ver en el ejemplo, las letras normales tienen una correspondencia directa tanto en la lhs como en la rhs.

Para conjuntos de letras o de tokens, podemos utilizar los siguientes operadores especiales en la lhs (llamados Metasímbolos):

\$*	Correspondencia entre 0 o más tokens.
\$+	Correspondencia entre 1 o más tokens.
\$-	Correspondencia de exactamente 1 token.
\$X	Correspondencia con la macro X
\$=x	Correspondencia con cualquier palabra en la Clase x.
\$~x	Correspondencia con ninguna palabra en la Clase x.

Y los siguientes Metasímbolos (para hacer la sustitución) en la rhs:

\$n	Correspondiente valor del Metasímbolo encontrado en la posición "n" del lhs.
\$_[name\$]	Nombre canonicalizado.

\$>n	Lo que queda de línea se pasa al conjunto de reglas "n" para que lo procese.
\$X	Valor de la macro X
\$:	Aplica la acción una única vez, sin iterar. Va directamente a la siguiente regla después de aplicar ésta.
\$@	Devuelve el resultado de esta acción. Sólo puede ir al principio de la línea.

Estos metasímbolos sólo pueden ser utilizadas por la regla de reescritura S0:

\$#mailer	Especificación del agente de reparto de mails.
\$@host	Especificación de la máquina.
\$:user	Especificación de el usuario.

Ejemplo 1:

La siguiente LHS: R\$-@\$*
 ante la entrada "al001010@alumail.uji.es", en la RHS tendremos:
 \$1 = "al001010"
 \$2 = "alumail.uji.es"

Ejemplo 2:

La siguiente regla: R\$-@\$* juan@\$2
 ante la entrada "al001010@alumail.uji.es", producirá la salida "juan@alumail.uji.es"

La lhs y la rhs forman una estructura de tipo bucle "do-while". Es decir, mientras la lhs evalúe direcciones y la evaluación sea cierta, se procesará la rhs. Sólo si lhs se evalúa como falsa, sendmail pasa a la siguiente regla de ese conjunto de reglas.

Cuando un conjunto de reglas contiene varias reglas individuales, el flujo es de la primera a la última regla, en el orden que son declaradas en el fichero de configuración. Por ejemplo:

```
S0
Rx    y
Ry    z
Rz    a
```

En este ejemplo, siempre que metas como dirección una "x", "y", o una "z", el resultado

será siempre una "a", debido al flujo.

3.8.3 Los separadores de tokens

Para conseguir separar unos tokens de otros, necesitamos definir unos separadores que nos marquen el límite entre un token y otro. Sendmail posee 10 separadores que son internos, y ya están definidos:

```
( ) < > , ; \ " \r \n
```

Podemos definir nosotros mismos los separadores que queremos que utilice sendmail mediante la macro "o". Por ejemplo:

```
Do.:%@!^=/[[]
```

Por ejemplo, si "@" esta en la macro \$o, ante una entrada "a@b", sendmail la separará en 3 tokens: "a", "@" y "b"

3.8.4 Como probar nuestras propias reglas de reescritura

Existe un método sencillo con el que podemos hacer pruebas con reglas de escritura creadas por nosotros, sin necesidad de modificar el fichero de configuración. Para utilizarlo, primero creamos un fichero de configuración mínimo, que no hará nada:

```
Mlocal,    A=x,    P=X
Mprog      A=x,    P=X
Do.:%@!^=/[[]
```

debajo de estas líneas del fichero, ya podemos incluir nuestras reglas, por ejemplo:

```
Mlocal,    A=x,    P=X
Mprog      A=x,    P=X
Do.:%@!^=/[[]
S0
R$+@$+    $2!$1
```

Si hemos llamado a este fichero, por ejemplo, test.cf, podemos ejecutar sendmail en modo test, introduciendo el número de regla a aplicar y una dirección para que sendmail nos devuelva el resultado de la aplicación de dicha regla. La salida sería la siguiente:

```
#sendmail -oQ/tmp -Ctest.cf -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 0 juan@xyz
rewrite: ruleset 0 input: "juan" "@" "xyz"
```

```
rewrite: ruleset 0 returns: "xyz" "!" "juan"
```

3.8.5 Ejemplos de reglas de reescritura extraídas del fichero de configuración de LINUX

Vamos a comentar algunos conjuntos de reglas de reescritura usados por los mailers del fichero de configuración del Linux.

Esta es un conjunto de reglas de reescritura utilizado por el repartidor local de Linux:

```
#
# Reescritura de las direcciones de la envoltura del remitente
#
S10
R<@>    $n                Si hay error, lo mandará al MAILER-DEMON (macro $n)
R$+     $: $>50 $1        Llama a la S50 que añadira el dominio local si lo
necesita
R$*     $: $>94 $1        Llama a la S94 para hacer masquerading

#
# Reescritura de las direcciones de la envoltura del destinatario
#
S20
R$+ < @ $* >           $: $1      Extrae unicamente el nombre de la maquina

#
# Reescritura de las direcciones de la cabecera del remitente
#
S30
R<@>    $n                Manda los errores al demonio (macro $n)
R$+     $: $>50 $1        Llama a la S50 que añadira el dominio local si lo necesita
R$*     $: $>93 $1        Llama a la S93 para hacer masquerading

#
# Reescritura de las direcciones de la cabecera del destinatario
#
S40
R$+     $: $>50 $1        Llama a la S50 que añadira el dominio local si lo
necesita
#R$*    $: $>93 $1        Llama a la S93 para hacer all-masquerading

#
# Subrutina que añade el nombre del dominio local a una dirección local (si lo necesita)
#
S50
R$* < @ $* > $*         $@ $1 < @ $2 > $3      (ya poseía el formato
correcto)
R$+     $: $@ $1 < @ *LOCAL* >      (añade el dominio local)
```

3.9 Definición de mailers

Otro de los puntos a configurar en el fichero sendmail.cf son los agentes de transporte (mailers) y la forma en que sendmail se comunicara con cada uno de ellos. El formato para definirlos es el siguiente:

Mnombre, {campo=valor}

donde nombre es un nombre arbitrario usado internamente por sendmail para referirse a ese "mailer"; y la pareja "campo=valor" definen los atributos de ese "mailer".

3.9.1 Campos más comunes en la definición de mailers

Los campos más importantes para definir los atributos de un "mailer" son los siguientes:

P	Path	Directorio donde se encuentra el agente de transporte.
F	Flags	Parámetros especiales de el agente de transporte.
S	Sender	Número de conjunto de reglas que se le aplica a las direcciones del remitente
R	Recipient	Número de conjunto de reglas que se le aplica a las direcciones del destinatario
A	Argv	Argumentos para el mailer.
E	Eol	Fin de línea usado por el mailer.
M	Maxsize	Longitud máxima del mensaje.

3.9.2 Parámetros más comunes en la definición de mailers

El campo F nos permite definir los parámetros que usará sendmail con ese mailer específico. Los parámetros más utilizados son los siguientes:

C	Añade @dominio a aquellas direcciones que son introducidas sin @.
D	El agente de transporte necesita una línea en la cabecera del mail de fecha ("Date:").
E	Añade ">" a las líneas del mensaje que comienzan por "From:" (Para no confundirlas con la cabecera).
e	El mailer realiza una conexión cara, por lo que intenta no conectar normalmente, sino que intentará el envío cuando se procese la cola de correo.
F	El mailer necesita una línea de cabecera con "From:".
I	El mailer establecerá una conexión SMTP con otro sendmail para transmitir los mails.
l	El mailer es local. Para reparto de correo local.

m	El mailer puede mandar mensajes a varios usuarios de una misma máquina en una sola transacción.
M	El mailer necesita una línea de cabecera "Message-Id:".
n	No insertar una línea "From:" (estilo UNIX) en el mensaje).
P	El mailer necesita una línea de cabecera "Return-Path:".
R	Abrir las conexiones SMTP desde un puerto seguro.
s	quita las comillas de la dirección antes de llamar al mailer.
S	No resetear el identificador de usuario antes de llamar al mailer. Esto se usa en entornos seguros donde sendmail se ejecuta con permisos de root.
x	El mailer necesita una línea de cabecera "Full-Name:".
0	No buscar registros MX para mails mandados a máquinas via SMTP..

3.9.3 Ejemplos de definición de mailers

El mailer definido con el nombre "local" es obligatorio en todo fichero de configuración. Es el repartidor de correo local:

```
Mlocal, P=/bin/mail, F=lsDFMmn, S=10, R=20, A=mail -d $u
```

Otro mailer que es obligado definir es un programa (shell), siempre llamado "prog":

```
Mprog, P=/bin/sh, F=lsDFMe, S=10, R=20, A=sh -c $u
```

Algunos mailers son definidos dentro del propio sendmail:

- Un mailer definido con el nombre "error", que se usará para generar un error de usuario.
- Un mailer con el nombre "file" para mandar mails a ficheros.
- Un mailer con el nombre ":include:", para mandar mails a listas :include:...

Todos los restantes mailers son definidos por el usuario, incluidos los agentes de transporte SMTP.

3.9.4 La definición del mailers en el fichero de configuración de LINUX

Estos son los mailers especificados en el fichero de configuración de Linux. Podemos observar tanto el repartidor local, el shell y los agentes de transporte SMTP. Está comentado el primero, los restantes son muy parecidos.

```
# El procmail es el repartidor de mails locales.
# Las reglas de reescritura aplicadas al remitente son 10 y 30
# Las reglas de reescritura aplicadas al destinatario son 20 y 40
# Los argumentos pasados al programa en la línea de comandos son:
# procmail -a <maquina destinataria> -d <usuario destinataria>
```

Mlocal, P=/usr/bin/procmail, F=lsDFMAw5:/|@qShP, S=10/30, R=20/40,
 T=DNS/RFC822/X-Unix,
 A=procmail -a \$h -d \$u

Mprog, P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=\$z:/,
 T=X-Unix,
 A=sh -c \$u

Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
 T=DNS/RFC822/SMTP,
 A=IPC \$h

Mesmtpl, P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
 T=DNS/RFC822/SMTP,
 A=IPC \$h

Msmtp8, P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
 T=DNS/RFC822/SMTP,
 A=IPC \$h

Mrelay, P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
 T=DNS/RFC822/SMTP,
 A=IPC \$h

1. Links interesantes

Página oficial de sendmail	http://www.sendmail.org
Grupo de discusión de sendmail	comp.mail.sendmail
Lista de correo moderada de sendmail	Enviar un mail a: majordomo@lists.sendmail.ORG con el mensaje: "suscribe sendmail-announce"
Libro oficial de sendmail escrito por: Bryan Costales Eric Allman Neil Richert	http://www.ora.com/catalog/sendmail2
Manual de sendmail: "Sendmail Installation and Operation Guide"	http://www.informatik.uni-kiel.de/%7Eca/email/english.html

Artículo sobre sendmail: "Unix World Online: Tutorial 008: Sendmail V8: A"	http://www.wcmh.com/uworld/archives/95/tutorial/008/008.txt.html
FAQ de sendmail	http://www.informatik.uni-kiel.de/%7Eca/faqs/sendmailv8.html
Introducción a sendmail y tutorial de sendmail en modo traza, por Christian Hallquist	http://www.ethz.ch/ETH/ID/KS.html.docs/sendmail.html
Código fuente de sendmail	ftp://ftp.sendmail.org/pub/sendmail
Todo tipo de enlaces relacionados sobre sendmail: "Harker's sendmail References Page"	http://www.harker.com/sendmail/sendmail-ref.html