# PROCMAIL.- filtro y agente de entrega de correo local

**Autor: Jorge Izquierdo Alonso**

# 1.- Objetivos y descripción del servicio.

El programa *procmail* ofrece un servicio de procesamiento de correo compatible con varios agentes de correo como *sendmail, Zmailer, smail, MMDF, mailsurr* y otros. El programa pretende como objetivo cubrir varias funcionalidades de procesamiento del correo de los usuarios facilitando la organización de los mails recibidos.

Así pues, *procmail*, permite crear servidores de correo, crear listas de correo, ordenar el correo entrante en diferentes carpetas según las preferencias de cada usuario, preprocesar el correo (función de filtro de los mails recibidos), arrancar programas a la llegada de algún mail o reenviar el correo entrante a algun otro lugar de forma selectiva. Además, si se utiliza conjuntamente con el programa *formail*, permite generar respuestas automáticas a los mails recibidos, extraer cabeceras (sencillas) , forzar a adoptar el formato adecuado a los mails para poder ser procesados por programas de correo estándar, extraer mails del *mailbox* o reconocer mails duplicados.

*Procmail*, es un servicio dirigido por eventos, es decir, el programa se invoca automáticamente cuando ocurre en el sistema un suceso determinado, en el caso de *procmail*, se invoca cuando llega un mail. Todas las funcionalidades descritas en el apartado anterior, requieren únicamente de dos ficheros (tres en el caso de las funcionalidades conjuntas con *formail*): uno es el ejecutable (*procmail*) y el otro un fichero de configuración (*.procmailrc*) (además del ejecutable *formail* para las últimas funcionalidades) que va a permitir a cada usuario definir el procesamiento que se le va a dar a cada mail recibido según ciertas características del mismo (se verá con más detalle en el apartado de configuración).

El modo de proceder del programa es el siguiente (en su uso más característico de filtrado): una vez invocado, primero inicializa algunas variables de entorno a los valores por defecto (pueden especificarse cualquier variable junto con su valor desde la línea de comandos), lee de la entrada estándar el mensaje hasta llegar a la marca de fin de fichero, separa las cabeceras del cuerpo del mensaje y (a falta de otros argumentos desde la línea de comandos) va a leer los ficheros de configuración para saber las operaciones de procesamiento del mensaje que debe realizar (procmailrc) . Según sean las indicaciones en este fichero, se distribuirá el mail hacia la carpeta deseada (u otros procesamientos). Si no existe fichero de configuración (puede ser global o local) *procmail* almacena el mail en el buzón de correo por defecto.

Además de los ficheros necesarios comentados, para permitir el bloqueo (lock) seguro del mailbox usando el servicio de ficheros NFS (cosa que pocos agentes de correo ofrecen), es necesario un fichero adicional llamado *lockfile*. Junto con estos ficheros necesarios para el funcionamiento del servicio en cada una de las situaciones, el servicio genera un fichero de registro (*log file*) que guarda información de todas las llegadas de mails mostrando un resumen en el que se indica cuando llegó, de qué trata el mensaje, a que carpeta se envía y su longitud en bytes. Existe una herramienta incluida en el paquete que traduce la información almacenada en este fichero de registro mostrándonos un resumen de su contenido: *mailstat procmail.log* . Esto nos permite obtener un amplio rango de mensajes de error y diagnóstico.

Además de todas las funcionalidades anteriores que son los usos típicos de este servicio, *procmail* puede ser usado como agente de entrega local con soporte *comsat/biff* completamente compatible con */bin/mail*. Para ello el programa debe ejecutarse con privilegios de root.

Otras características del servicio son las siguientes:

- Se trata de un servicio muy sencillo de mantener y configurar pues como hemos dicho sólo consta de un ejecutable y un fichero de configuración (veremos en los apartados siguientes las posibles opciones de configuración del servicio así como los ficheros involucrados en las mismas).

- Se trata de un programa cuyo código se encuentra completamente escrito en C, lo cual producirá un impacto mínimo sobre los recursos del sistema y lo hace más rápido en tiempo de CPU que el /bin/mail medio. El programa está diseñado para trabajar bajo las peores condiciones de manera que intenta enviar los mails a pesar de cualquier limitación del sistema (sistema de ficheros lleno, sin espacio swap, tabla de procesos llena, etc.). Si no se consigue tras todos los esfuerzos, se puede devolver el mensaje al que lo envió si así lo deseamos.

- No impone ninguna limitación de número de líneas, longitud del mail (tanta como permita la memoria del equipo) o uso de cualquier carácter en el mismo. Los mail son tratados de forma sencilla en las operaciones de filtrado y entrega basado en decisiones de sí o no que pueden tener en cuenta campos como el tamaño del mail. Todo ello se indica en el fichero de configuración mediante expresiones regulares estándar de egrep.

- Maneja cuatro tipos de carpetas de correo estándar hacia donde dirigir los mails: carpetas de fichero único (de formato UNIX estándar y no-estándar), carpeta de directorio con un fichero por cada mail y carpetas de directorio MH (con ficheros numerados).

## 2.- Requisitos.

Dada la sencillez del servicio expresada en el apartado anterior, los requisitos necesarios para su instalación y correcto funcionamiento son mínimos. Así, el programa ha sido desarrollado para funcionar en cualquier sistema tipo UNIX (o que obedezca la norma POSIX).

Necesita además de la instalación de un MTA (Agente de Transferencia de Correo) de los que hemos indicado como compatibles en el apartado anterior (sendmail, Zmailer, smail, MMDF y mailsurr) en nuestro sistema UNIX. En realidad será válido cualquier MTA que pueda procesar mails que obedezcan la norma RFC-822.

Finalmente, aparte de los programas que incluye el paquete *procmail*, será necesaria un entorno de desarrollo C con las bibliotecas necesarias por el programa, dado que, como ya hemos comentado, este paquete ha sido completamente desarrollado en código C. Para saber las bibliotecas a las que referencia el programa podemos consultar el fichero *src/includes.h* en el paquete de instalación del servicio (*procmail.tar.gz*).

Existen algunos ficheros opcionales que permiten mejorar el funcionamiento del servicio añadiendo funcionalidad adicional pero que no son estrictamente necesarios:

*lockfile:* sólo necesita ser instalado si vamos a leer varios buzones de correo con uno de los agentes de correo estándar que no soporte bloqueo de fichero.
*formail:* sólo necesita ser instalado si vamos a recibir correo fuera del formato estándar de los buzones de correo (o para añadir las funcionalidades que ya se han comentado en el primer apartado).
*mailstat:* no es en absoluto necesario pero puede ser de gran utilidad para diagnosticar errores y otros aspectos ya comentados. Se trata de un sencillo shell script.
Estos tres programas son de fichero único, no necesitan de la existencia de ningún fichero compilado para su funcionamiento a excepción de ellos mismos que además son independientes entre sí.

# 3.- Arquitectura del servicio.

Dada la sencillez del servicio, su arquitectura se basa en un único fichero ejecutable al que pueden acceder los usuarios del sistema UNIX para distribuir de forma eficiente su correo según este va llegando. Sin embargo, existen varios modos de uso del programa de forma que según sea la forma de acceder al programa necesitamos una u otra configuración de los distintos ficheros. Los modos de uso del servicio son:

- Acceso al servicio (e incluso instalación) de un usuario sin privilegios. Cualquier usuario de un sistema Linux puede hacer uso de este servicio de forma individual. Aunque el programa no haya sido instalado para todos los usuarios, este puede ser instalado de forma local y configurarse para que procese el correo de la forma deseada por el usuario.

- Acceso al sistema como filtro de correo genérico para todos los grupos de mensajes, es decir, como filtro de propósito general de forma que pueda ser invocado por cualquier usuario del sistema. Esta forma de instalar el servicio permitirá indicar sentencias de procesamiento del correo para cualquier usuario (mediante un fichero de configuración común: /etc/procmailrc ).

- Como agente de entrega local, funcionando con los privilegios de root y reemplazando el agente /bin/mail (con soporte biff/comsat).

La forma de configurar el sistema varía según el modo de uso que vayamos a hacer del servicio de entre los anteriores. Así pues, en el primer caso, es necesario invocar la llamada al programa desde un fichero *.forward* del HOME (para agentes de entrega tipo *sendmail*. Para MMDF se llama *.maildelivery*) de cada usuario mientras que en los otros casos no es necesario este fichero sino que la invocación será directa a la llegada de un mail, sin embargo será necesaria la configuración de otros ficheros (los de configuración del agente de transferencia empleado).

Así pues, estos son los posibles usos que se puede obtener de nuestro servicio, cuya arquitectura, la misma para cualquiera de los casos, se puede explicar de forma parecida a como explicamos su funcionamiento en el primer apartado dada la sencillez de la misma. Se trata única y exclusivamente de un fichero que leerá de la entrada estándar los mails que le proporcionará el agente de transferencia del sistema a su llegada y, siguiendo las directrices de otro fichero de configuración del que va a hacer uso, filtrará su contenido procesándolo de forma que lo enviará a uno u otro sitio (tanto carpetas del receptor como al agente para que lo devuelva al emisor) según ese fichero de configuración.

De todo lo explicado hasta ahora, parece que lo más importante del servicio es la configuración del fichero *.procmailrc* en el que cada usuario dará las directrices de lo que hacer con su correo. Sin embargo, hay otros pasos para configurar el servicio en cada una de las situaciones anteriores. Tanto estos pasos como, con más detenimiento, el aspecto del fichero de configuración, será comentado en el siguiente apartado.

# 4.- Configuración.

### *Instalación y configuración según el modo de uso*

La forma de configurar nuestro servicio depende, como hemos dicho, del modo de uso que vayamos a efectuar del mismo de entre los expuestos en el apartado anterior. Sin embargo, algunas partes de la configuración, como la del fichero *.procmailrc* son necesarias hagamos lo que hagamos. Así pues, iremos en este apartado indicando en cada paso a qué modo de uso se refiere o sí es común para los tres.

Lo primero a realizar antes de configurar ningún fichero es instalar correctamente el programa (tanto para usuario privilegiado como para no privilegiado). Para ello, el paquete nos proporciona un fichero *Makefile*, con el que introduciendo la orden *make install* nos instalará el paquete. Existen varias opciones de configuración de este archivo así como del *config.h* según como queramos hacer nuestra instalación. Así pues, el fichero *Makefile* nos permite indicar donde queremos instalar nuestro programa así como el directorio destino para pruebas de bloqueo (LOCKINGTEST), páginas del manual, etc . Mientras, en el fichero *config.h,* podemos indicar otras cosas como que nuestro programa va a tener soporte MMDF (como agente de transferencia) o prefijar valores para las variables de entorno (incluyendo el PATH). En esté fichero se definen una gran cantidad de variables que nos van a permitir el tratamiento de los mensajes (y que no hay que modificar a no ser de estar muy seguros de lo que hacemos).

Las posibles opciones que nos ofrece *make* son:

```
# help (or targets)   Muestra esta lista de opciones
# init (or makefiles) Realiza un chequeo preliminar del sistema y genera
#                     Makefiles según el resultado.
# bins                Preinstala sólo los ficheros binarios en./new
# mans                Preinstala sólo las páginas de manual ./new
# all                 Realiza las dos anteriores
# install.bin         Instala los ficheros binarios desde ./new a $(BINDIR)
# install.man         Instala las páginas de manual desde ./new a $(MAN[15]DIR)
# install             Realiza las dos anteriores
# recommend           Muestra los modos recomendados de suid/sgid para nuestro sistema.
# install-suid        Impone los modos recomendados.
# clean               Intenta volver al estado anterior a realizar el make.
# deinstall           Elimina cualquier archivo binario del programa y páginas
#                     de manual previamente instaladas en el sistema.
# autoconf.h          Muestra un listado de las anomalías del sistema
# procmail            Preinstala sólo lo relacionado con procmail en ./new
# formail             Preinstala sólo lo relacionado con formail en ./new
# lockfile            Preinstala sólo lo relacionado con lockfile en ./new
```

Al invocar la orden *make install* se llamará a algunas de las anteriores para ser realizadas de forma conjunta. Así: se invocará implícitamente *make init* para chequear el sistema, se ejecutará un script llamado *autoconf* que será el encargado de generar el fichero *autoconf.h* y creará los ficheros binarios a utilizar (procmail, formail y lockfile si no se le indica lo contrario) junto con las páginas de manual de los mismos y almacenará cada uno en los directorios indicados en el mismo *Makefile* (BINDIR, MAN1DIR, MAN5DIR).

Como tanto el fichero *Makefile* como el *config.h* son autocontenidos, no nos detendremos más en su análisis. Para profundizar más en las posibles opciones de instalación no habrá más que consultar estos ficheros antes de hacer el *make install*.

Una vez tenemos instalado el paquete, según el modo de uso que vayamos a hacer del mismo deberemos proceder de una u otra manera. Empezaremos comentando el caso más sencillo que es el de instalación local por un usuario no privilegiado:

En este caso, para que se ejecute el programa *procmail* sobre cada mail que nos llega, debemos hacerlo a través del fichero *.forward* (es el utilizado en la mayoría de sistemas). En él, debemos indicar que la llegada de cualquier mensaje se va a enviar a *procmail* para que este procese el mensaje. La orden a poner en este fichero será:

```
“| exec /usr/bin/procmail”    ( si está en ese directorio )
```

o en el caso de estar usando exim:

```
| /usr/bin/procmail
```

con ello ya se procesará el correo con nuestro filtro *procmail* según las opciones que le indiquemos en el programa de configuración que se detalla más adelante.

En el segundo caso (uso como filtro de correo genérico para el sistema) no será necesaria la creación del fichero *.forward* sino que el programa se invocará automáticamente a la llegada de nuevo correo. Para ello, habrá que indicar de alguna manera a nuestro MTA que se va a realizar este procesado del correo. La forma de realizar esto va a depender del MTA utilizado, nos limitaremos a explicarlo con *sendmail,* pues es el más utilizado. Así pues, en Debian, sólo necesitamos añadir unas líneas al fichero de configuración del agente de transferencia (*sendmail.cf*). Para mayor sencillez, estas serán generadas automáticamente por el fichero *sendmail.mc* sin más que añadir en este la siguiente línea:

```
FEATURE (local_procamail)dnl
```

Esto es para versiones de sendmail 8.7 o más nuevas. Además, en estas versiones, podemos editar algún parámetro más en el fichero *sendmail.cf* de forma que nos permita obtener mayor funcionalidad de nuestro programa *procmail* a través del paso de meta argumentos al mismo. Para saber como realizar esto, así como la forma de configurar el agente de correo para versiones anteriores de *sendmail* o utilizando otros agentes distintos, se puede consultar el fichero *advanced* incluido en el directorio de ejemplos proporcionados con el paquete *procmail.tar.gz.*

Finalmente, para que nuestro programa opere en el último coso, es decir, como agente de entrega local integrado, además de los pasos indicados para el uso anterior (edición del fichero de configuración del agente de transferencia), será necesaria una última operación. Lo que hay que hacer, es eliminar nuestro agente de entrega actual (por ejemplo */bin/mail* o */bin/lmail*), crear un enlace (simbólico o duro) desde este hacia *procmail*:

```
ln /usr/bin/procmail /bin/mail  ( o /bin/lmail ).
```

Usando este método, nuestro agente de entrega sólo va a poder utilizado para entregar correo si bien en muchos sistemas */bin/mail* tenía varios usos (por ejemplo leer o chequear mail), por lo que será mejor no utilizarlo de esta manera sino, si podemos editar la invocación de */bin/mail* desde el MTA, invocar procmail en su lugar con lo que no se perderán el resto de funcionalidades.

Explicado la configuración particular a realizar según sea el caso de uso que hagamos del servicio, pasamos a detallar la confección del fichero de configuración que será el que imponga las operaciones a realizar sobre los mails entrantes para cualquiera de las situaciones anteriormente descritas.

### *Configuración del fichero .procamilrc*

El principal elemento de configuración de configuración del servicio, como se ha indicado a lo largo del documento, es el fichero rc que se va a encontrar en el directorio HOME de cada usuario del servicio. Este fichero es una mezcla de asignaciones a variables de entorno (algunas de ellas con significado especial para *procmail*), y *recetas* sobre cuales son las decisiones que debe tomar nuestro servicio a la llegada de un mail.

El aspecto de esas recetas, es el de expresiones regulares de una línea que van a ser cotejadas con la cabecera del mail entrante para decidir si sobre el se va a llevar a cabo una determinada acción o no. Si ninguna de las recetas casa con el mail recibido al llegar al final del fichero rc, este será enviado al lugar indicado por la variable de entorno DEFAULT.

Existen dos tipos de recetas: recetas de envío y de no envío. Las de envío son aquellas que provocan que el mail sea enviado a una carpeta o procesado por un programa. Tras la ejecución de una de estas recetas, el procesamiento del fichero de configuración termina aunque no se haya llegado al final. Las de no envío son las que hacen que procmail capture la salida de un programa o las que comienzan un bloque. Tras ellas se sigue con el procesamiento normal del fichero. Puede desearse que tras una receta de envío se siga con el procesamiento del correo para, por ejemplo, mandar copia del mail a varios sitios. Esto se consigue especificando un flag (letra c) en la receta de envío.

Como ya se ha dicho, el fichero de configuración pueden haber tanto recetas como asignaciones a variables de entorno. Estas últimas pueden ir mezcladas entre las recetas de forma que se apliquen en el momento de lectura de las mismas pudiendo provocar que varias recetas sean aplicadas dentro de entornos distintos. Las asignaciones a las variables de entorno se harán exactamente igual que en la shell *sh*, eliminándose los espacios en blanco y desapareciendo aquellas que no muestren el signo "=". Algunas de las variables más importantes para especificar son:

| | |
|---|---|
| MAILDIR | directorio donde se ejecuta *procmail.* |
| DEFAULT | buzón de correo por defecto. Si no se especifica nada El servicio tomará el estándar del sistema. |
| SENDMAIL | agente de transferencia de correo utilizado. |
| LOGFILE | fichero donde se almacenarán las trazas de error y diagnóstico generadas por el servicio. |
| LOCKEXT | Extensión que van a tener los ficheros de bloqueo que nos van a servir para permitir el acceso en exclusión mutua a las carpetas de correo. |

Existen muchas variables más cuyo uso se puede consultar en las páginas de manual del fichero de configuración (*procmailrc(5)*).

En cuanto a las recetas, que son lo que realmente se va a ocupar del procesamiento de nuestros mensajes dentro del entorno dado por las variables, van a tener el siguiente aspecto:

```
:0 [flags] [: [ficherodebloqueolocal] ]
<condiciones (cero o más, una por línea)>
<una línea de acción>
```

Toda receta comienza por el carácter ":" que marcará el comienzo de la misma. Seguidamente se pueden adjuntar al a misma tantos flags como se deseen que van a aportar distintas funcionalidades al procesamiento de nuestro correo. Así por ejemplo:

| | |
|---|---|
| A | Indica que esa receta depende de la inmediatamente anterior. Permite encadenar acciones que dependen de la misma condición. |
| c | Genera copia del mensaje como se ha explicado para recetas de envío. |

| | |
|---|---|
| E | Indica que la receta sólo se ejecutará si la anterior no fue ejecutada. |
| e | Igual que la anterior pero si la acción falló. |

Existen muchos otros flags cuyo uso se puede consultar en las mismas páginas de manual que las variables de entorno.

Tras los flags (optativos) se puede especificar un fichero local de bloqueo para la carpeta que se indicará posteriormente en la línea de acción (también de forma opcional). Esto se hará con la inclusión de otro nuevo carácter de ":" tras los flags y un nombre de fichero (sin extensión pues ya está especificada en una variable de entorno) si se desea. Si no se especifica el nombre del fichero pero se ponen los dos puntos, el servicio tomará el fichero por defecto para estos menesteres indicado también en una variable de entorno.

La siguiente (o siguientes) líneas en la receta sirven para especificar las condiciones que deben cumplir nuestros mails para aplicar sobre ellos la acción que vendrá a continuación. Estos es lo que llamamos *condiciones* y vendrán en cero o más líneas precedidas del carácter "*". Lo que vendrá después será transformado al tratamiento interno de expresiones regulares *egrep* eliminando los espacios en blanco de comienzo y fin. Dichas expresiones son completamente compatibles con las expresiones regulares egrep, de tal forma que "^" indica el comienzo de una línea, "$" indica el final de línea, "." indica cualquier carácter excepto una nueva línea, etc. Pero, además de esto, existen condiciones especiales que no son exactamente expresiones regulares, para seleccionarlas, la condición debe empezar por:

| | |
|---|---|
| ! | Invierte la condición |
| $ | Evalúa el resto de acuerdo con las reglas de sustitución de *sh.* |
| ? | Usa códigos de salida del programa especificado. |
| < ; > | Mira si la longitud del mail es menor (mayor) del número de bytes especificado (en decimal). |

Las condiciones son evaluadas y, si todas ellas se cumplen (conjunción AND implícita), se ejecutará la línea de acción que seguidamente pasaremos a describir. Si no existen condiciones el resultado por defecto será *true*.

La línea de acción puede comenzar con alguno de los siguientes caracteres:

| | |
|---|---|
| ! | Indica envío a todas las direcciones especificadas tras él. |
| \| | Comienza la ejecución del programa indicado (sobre cada mail que cumpla las condiciones) . Si no se especifica ningún programa, el mail se envía a la salida estándar. |
| { | Seguido de al menos un espacio, indica el principio de un bloque, de forma que todo lo especificado hasta el cierre de llave depende de la misma condición. |

Cualquier otra cosa dentro de la línea de acción será entendida como el nombre de un buzón de correo. Si es un fichero, el mail será añadido al mismo. Si es un directorio, se añadirá al mismo como un nuevo fichero con garantías de unicidad por la variable MSGPREFIX*. Si el nombre del directorio termina por en "/", se supone que va a ser una carpeta de tipo MH y por tanto se usará para el mensaje el siguiente número de fichero disponible.

Con todo esto ya podemos configurar nuestro servicio de *procmail* para que procese nuestro correo de la forma deseado. A continuación mostramos unos ejemplos sencillos de configuración del fichero para procesamiento del correo en carpetas. Para ver más ejemplos de otras configuraciones más avanzadas o con otras funcionalidades, se puede consultar las páginas de manual *procmailex(5)* o acceder a los ejemplos facilitados con el paquete en el directorio *./examples*.

| | |
|---|---|
| :0 c<br>* ^From.*juan<br>^Subject:.*laboratorio<br>! lsc@dit.upm.es | Reenvía todos los mails de juan<br>que hablen sobre el laboratorio<br>a la dirección indicada |
| :0 A<br>juanlab | Y además los almacena en una<br>carpeta apropiada. |
| :0 hwc:<br>* !^FROM_MAILER<br>\| gzip >> headc.gz | Añade las cabeceras de los mensajes<br>que no vienen del postmaster a nuestro<br>fichero de cabeceras (flag h)tras comprimirlas<br>mediante el programa gzip. Además,<br>hace uso de un fichero de bloqueo (headc.lock)<br>que no se borrará hasta que el pipe<br>haya finalizado (flag w) |

## 5.- Pruebas.

Para comprobar la correcta instalación y funcionamiento del servicio podemos realizar pruebas sobre el mismo. Para ello debemos provocar la llegada de correo y comprobar el correcto procesamiento del mismo por nuestro servicio (en aquellos usuarios que hayan decidido hacer uso del mismo editando el fichero *.procmailrc* y *.forward* en el caso de no estar instalado globalmente).

La manera de comprobar la corrección del servicio es examinar el fichero de registro (se comentará en el apartado siguiente con más detalle) o hacer que el programa saque mensajes por pantalla de las operaciones que está realizando así como de los posibles errores. Para que el programa nos muestre los resultados por pantalla es necesario indicarlo en la invocación del servicio mediante la orden *procmail VERBOSE=ON*. Si esto no funciona, o si deseamos obtener la máxima información sobre el procesamiento realizado por nuestro servicio, podemos acceder al fichero *config.h* de nuestro paquete donde debemos introducir (está ya pero como comentario) la línea: *define console*. Con esto será más sencillo encontrar cualquier problema debido a la amplia información que nos ofrecerá en pantalla el programa, sin embargo, una vez resuelto cualquier fallo, sería aconsejable volver a comentarla para no sobrecargar la salida por consola.

Con esto probaríamos nuestro servicio si bien podría ocurrir que el error no estuviera en el mismo sino en el agente de transferencia utilizado para hacer uso de *procmail* con lo que serían necesarias otras pruebas referentes al MTA usado y que no procede comentar en este documento.

## 6.- Gestión diaria.

Dada la sencillez del servicio que se ha venido mostrando a lo largo del documento, el mantenimiento del mismo no va a ser ninguna carga para el administrador del sistema. Es más, en el caso de usarse en el modo local por usuarios no privilegiados, el administrador no debe hacer nada siendo cada usuario el único obligado a mantener actualizada la versión del programa que más se adecue a sus necesidades y de realizar los cambios que crea oportunos para la gestión de su correo en el fichero de configuración descrito en el apartado 4 del presente documento.

Tanto para el uso local como para el uso como filtro genérico, a la hora de actualizar el paquete con nuevas versiones, al existir ya en el sistema una versión antigua, antes de proceder con la instalación es necesario revisar el fichero HISTORY que estará incluido en el paquete. En el se nos indican las mejoras introducidas en cada nueva versión así como los cambios necesarios de configuración a realizar para el correcto funcionamiento. Así pues, será necesario comprobar las entradas que aparecen en el fichero entre la versión antigua, y la que pretendemos instalar, para resolver cualquier conflicto que pudiera existir entre ambas y para realizar los cambios que se estimen oportunos para permitir la correcta operación del servicio en su nueva versión.

Una vez instalada la nueva versión o si cambiamos el modo de proceder del programa a través de su fichero de configuración, podemos hacer que cualquier carpeta de correo antigua pase por nuestro nuevo filtro de nuevo con las reglas establecidas para nuestra nueva configuración. Haremos esto con la siguiente orden:

```
cat mbox | formail –s procmail
```

Otra ventaja para el mantenimiento es la que nos permite el fichero de registro que, como indicamos en el primer apartado, nos va a crear nuestro programa según va recibiendo el correo. En él se irán almacenando tanto información sobre los mails recibidos por cada usuario a lo largo del tiempo de ejecución de nuestro servicio en las última fechas, como posibles errores que se hayan producido en su procesamiento así como diagnósticos sobre ese procesamiento. Es pues obligación de cualquier buen administrador del servicio (ya sea usuario privilegiado o no), comprobar periódicamente los informes de dicho fichero (*procmail.log*) para analizar el comportamiento de nuestro servicio detectando posibles anomalías que deberíamos resolver. Para facilitar esta labor, el paquete de nuestro servicio nos ofrece, como ya se indicó, una herramienta que nos permite mostrar la información del fichero de registro del servicio de forma ordenada y fácil de comprender. Así pues, invocando el programa *mailstat procmail.log*, desde la línea de comandos vamos a obtener la información necesaria para el mantenimiento de nuestro servicio, indicándonos si el procesamiento no es correcto y las operaciones necesarias para solucionar el problema (cambios en el fichero de configuración, instalación de bibliotecas necesarias que no hubieran sido correctamente instaladas, incompatibilidades con es agente de transferencia u otros, etc.).

Dada la sencillez del servicio, con la realización de estas dos funciones (actualización de versión y examinar el fichero de registro del servicio) de forma periódica (no es necesario diariamente), tendríamos el servicio funcionando correctamente en nuestro sistema sin problema alguno. Sin embargo, el mantenimiento de nuestro servicio, al estar ligado de manera estrecha con el agente de transferencia de correo, implicaría la correcta gestión diaria del MTA que estemos utilizando (probablemente *sendmail*), lo cual es algo más complejo y que no explicaremos dado que se sale de los objetivos del presente documento.