

Correo desde la línea de comandos

CARTA LIGERA

Hao Wang, Fotolia

El cliente de correo de la línea de comandos mailx es sencillo y eficiente. Se ocupa del correo interactivamente o mediante opciones de la línea de comandos. Aunque carece de una vistosa GUI, mailx lo compensa sobradamente con su capacidad de scripting. **POR HEIKE JURZIK**

Thunderbird, Evolution o Kmail no son las únicas opciones cuando se trata de clientes de correo. Aunque los clientes de correo gráficos ofrecen a sus usuarios una gran cantidad de comodidades, a veces una solución de la línea de comandos es una ventaja. Tanto Mutt [1] como el ligero mailx [2] soportan opciones de procesamiento por lotes desde la shell.

La mayoría de las distribuciones incluyen mailx de manera predeterminada. OpenSUSE y Debian incluyen el paquete en sus medios de instalación. En OpenSUSE, `/bin/mail` es un enlace simbólico que apunta a `/usr/bin/mailx`, mientras que en sistemas Debian las cosas se hacen de otra manera: `/usr/bin/mailx` apunta a `/usr/bin/mail`. Los comandos que siguen y las opciones de comandos se comprobaron todas en Debian 4.0 "Etch".

Arrancar el cliente de correo con `mailx` en el prompt de la shell nos lleva a un entorno interactivo. El cliente muestra el contenido de nues-

tro buzón de entrada; la configuración por defecto para esto en la mayoría de los sistemas es `/var/mail/username`. Seguidamente podremos ver una presentación de los mensajes, incluyendo al remitente, fecha, tamaño y línea de asunto.

Escribiendo `?` y pulsando `Enter` nos presenta un fichero de ayuda rápida sobre el uso del mailer. La manera

más fácil de presentar un mensaje se consigue introduciendo un número y pulsando `Enter`. La tecla `Enter` además desplaza el texto, aunque podemos usar las teclas de flecha arriba y abajo también. Con `Q` abandonamos la presentación y nos vuelve a la pantalla principal. Si los mensajes se han desplazado fuera de la ventana, podemos pulsar `H` y `Enter` para presentarlos de nuevo.

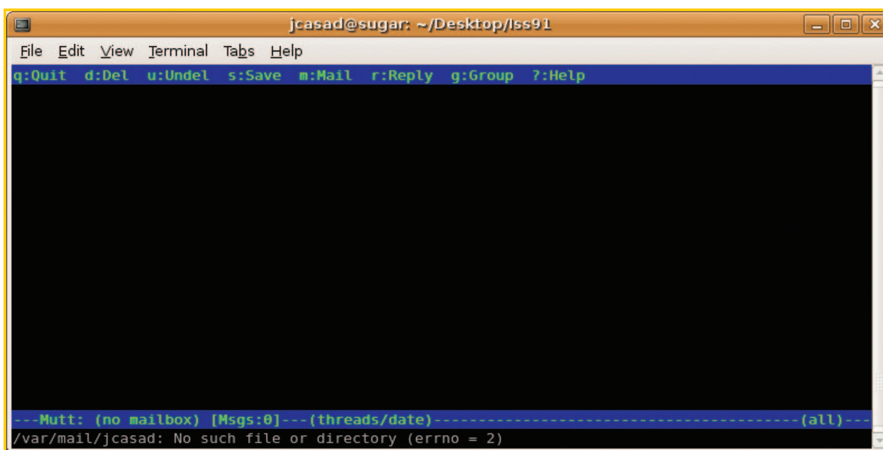


Figura 1: Mutt también viene con una GUI rudimentaria, aunque el sistema de menú por teclado no es muy diferente del trabajo en la línea de comandos.

Mailx posee dos métodos para cerrar el programa: pulsando *X* y Enter se abandona el cliente de correo sin eliminar los mensajes que hemos leído del buzón de entrada, mientras que *Q* más Enter hace lo siguiente:

```
Saved 2 messages in
/home/petros
ilie/mbox
Held 3 messages in
/var/mail/pet
rosilie
```

Introduciendo *D* y pulsando Enter borraremos los mensajes. De nuevo, podemos seleccionar mensajes para borrarlos especificando sus números. Por ejemplo, *d 1 3* elimina los mensajes primero y tercero. También podemos seleccionar un rango, como *d 1-10*, con lo que se borrarán los primeros diez mensajes o *d **, que borra todos. Si por error eliminamos un mensaje, *u* lo recuperará. El comando usa la misma combinación de dígitos que la opción *d*. Si escribimos *R* podremos responder a un mensaje, mientras que con *M* comenzaremos otro nuevo. Para especificar la dirección de destino usamos *m*:

```
m petronella@huhnix.org
```

Mailx nos pide entonces que introduzcamos la línea de asunto antes de que podamos seguir adelante con el mensaje en sí.

Una vez lo hayamos hecho, escribimos un punto seguido de Enter y añadimos más receptores después de Cc: (copia):

```
Cc:ole@huhnix.org,max@huhnix.org
```

Si preferimos arrancar nuestro editor favorito para componer mensajes, escribimos *~v* directamente después de la línea de asunto. Mailx evalúa la

Uso de Opciones Mailx

Envío de una copia

```
$ mailx -s 'Hello! petronella@huhnix.org -c petrosilie@huhnix.org [...]
```

Mensaje con Echo

```
$ echo 'Hello, this is a test.' |mailx -s 'Hello!' petronella@huhnix.org
-c petrosilie@huhnix.org
```

Envío de un mensaje de texto

```
$ mailx -s 'Hello!'
petronella@huhnix.org
< mail.txt
```

Resolución de problemas por informe de estado

```
$ echo 'Hello, this is a
test.' | mailx -v -s 'Hello!'
petronella@huhnix.org
Mail delivery Status Report will
be mailed to <huhn>
```

variable de entorno *\$VISUAL* para identificar nuestro editor favorito. Si nos gusta trabajar con, por ejemplo, *joe*, podemos configurarlo para mailx introduciendo *export VISUAL=joe*.

Mailx muestra su fuerza, flexibilidad y velocidad cuando se trabaja desde la shell. La línea de comandos genérica es:

```
mailx [Opcion] user@target.tld
```

Los usuarios pueden definir opciones, como la línea de asunto (*-s*), la dirección de correo para el campo *bbc* (copia oculta) (*-b*), más receptores a *cc* (*-c*), etc. Es importante no olvidar usar comillas para espacios en blanco y caracteres no estándares.

Aún falta el cuerpo del correo, es decir, el texto del mensaje. Para un mensaje corto, *echo* es probablemente una buena elección. Para el caso de textos largos se puede coger el cuerpo del texto desde un fichero externo. Mailx puede leer este fichero desde la entrada estándar.

Si necesitamos resolver los problemas de nuestra configuración, *-v* es

una gran ayuda. En sistemas Debian, mailx le dice al usuario que ha enviado un informe de estado, que aparece un poco de tiempo después en el buzón de entrada. El cuadro “Uso de Opciones Mailx” muestra un ejemplo del uso de las opciones de mailx.

La única manera de añadir un adjunto a un mensaje es mediante el uso de *uuencode*. Esta herramienta convierte ficheros binarios, como imágenes o ejecutables, a caracteres ASCII imprimibles. Tras hacerlo, puede enviarse el fichero de texto (véase el cuadro “Envío de Adjuntos”). El receptor ve entonces el fichero como un bloque de texto dentro de las palabras clave *begin* y *end*. *Uudecode* vuelve a reconvertir el fichero. El comando *uudecode attach_image.txt* tiene como consecuencia el fichero *hej.jpg*.

Las cosas son más fáciles si ejecutamos Mutt en modo procesamiento por lotes. El cliente de correo Mutt no sólo ofrece *-s* para el asunto, *-c* para copiar y *-b* para opciones de copia oculta a las que hemos hecho referencia en este artículo, sino también el parámetro *-a* para el envío de adjuntos simple. Véase el cuadro “Envío de Adjuntos”.

Envío de Adjuntos

Adjuntos con uuencode y mailx

```
$ uuencode hej.jpg hej.jpg > attach_image.txt
$ mail -s 'photo' petronella@huhnix.org < attach_image.txt
```

Adjuntos con Mutt

```
$ mutt -s 'photo' -a hej.jpg petronella@huhnix.org < mail.txt
```

RECURSOS

[1] Mutt: <http://www.mutt.org>

[2] Mailx: <http://www.scit.wlv.ac.uk/cgi-bin/mansec?1+mailx>