



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Colecta de trucos para Mutt (14252 lectures)

Per Adeodato Simó, [dato](http://chistera.yi.org/~adeodato/blog/) (<http://chistera.yi.org/~adeodato/blog/>)

Creado el 22/11/2003 13:30 modificado el 22/11/2003 13:30

En este mini-artículo os dejo algunos trucos para Mutt que he encontrado realmente útiles. Ya sé que son pocos, pero lo mejor será la segunda parte, cuando todos empecéis a enviar (espero) los vuestros como comentarios...

El antiguo formato PGP *inline*

Como la mayoría sabréis, el formato *inline* para firmar o cifrar correos mediante PGP está obsoleto. Sin embargo, aún siguen mandándose muchos mensajes firmados así, ya sea por mantener la compatibilidad con clientes de correo que no soportan el nuevo formato PGP/MIME o porque utilizarlo requiere algunos [sencillos pasos](#)⁽¹⁾ que hay gente que no parece querer realizar.

Bueno, a lo que iba; la mayoría de usuarios de Mutt ya conocerán la función `check-traditional-pgp`, asociada por defecto a la combinación de teclas `Esc+P` y que permite verificar la firma de un mensaje en el formato antiguo. Ahora bien, a mí personalmente me parece un rollo tener que estar pendiente de darle a la tecla cada vez que me llega un correo de éstos. Así que lo que yo quería era automatizarlo.

Primero probé con una receta de procmail que se puede encontrar fácilmente por la red que realiza la conversión del antiguo formato al nuevo. Pero no me convenció porque fallaba cuando había más de una parte en el correo, se perdía el charset del mismo, etc.

Por fin ayer se me ocurrió combinar la función `check-traditional-pgp` con un `message-hook`, tal que así:

```
message-hook '!(~g|~G) ~b"^-----BEGIN\ PGP\ (SIGNED\ )?MESSAGE"' \
    "exec check-traditional-pgp"
```

Así se consigue que antes de entrar en un correo que tenga la pinta de estar firmado mediante el formato antiguo, Mutt verifique la firma y tal. (Ojo, no caigáis en la tentación de poner algo más sencillo tipo

```
message-hook . "exec check-traditional-pgp"
```

porque Mutt se quedará colgado, ya que internamente la función `check-traditional-pgp` llama a `display-message` y se entretienen llamándose todo el rato la una a la otra.)

La movida de `reply` y `list-reply`

Una de las cosas que me reventaba cuando empecé a usar Mutt es el hecho de que existan las funciones `reply` y `list-reply` pero no `list-reply-o-reply-a-secas`. Me explico, y así los que no usan Mutt se enteran:

Mutt tiene varias funciones para responder a un correo, `reply` a secas para responder al autor, `group-reply` para responder a todos y `list-reply` para responder a una lista (previamente le habremos dicho a Mutt qué direcciones pertenecen a listas de correo y si estamos suscritos a ellas o no).

La función `list-reply` viene muy bien usarla, porque se encarga de no enviar copias a quien no debe, incluimos o no en el `Mail-Follow-Up` según estemos o no suscritos, y varias cosas más. La cosa es que, al ser una función diferente, tiene que estar en otra tecla diferente y es un rollo acordarse de si estás leyendo una lista o no para darle a L o



a `r` según corresponda. Normalmente lo que uno quisiera es darle siempre a `r` de responder, y que si fuera una lista respondiera a la lista y si es una persona, a la persona.

Esto se puede conseguir con los siguientes *hooks*:

```
folder-hook      .      bind index r reply
folder-hook      .      bind index L list-reply
folder-hook      '=l/'  bind index r list-reply
folder-hook      '=l/'  bind index L reply
```

Esto lo que hace es que la tecla `r` sea siempre responder, y según sea una lista o no usa `list-reply` o `reply` a secas. Si quiero responder a un mensaje de una lista en privado puedo usar `L`. (Deberías sustituir `"=l/"` por alguna expresión que incluya todas las carpetas en las que hay correo de listas. En mi caso, todas las listas están en `~/ .mail/l.`)

Los mensajes en HTML

Por mucho que se diga, siempre habrá gente que siga enviando los correos en HTML. Yo normalmente usaba el commando `auto_view text/html` con una entrada tipo

```
text/html; lynx -dump '%s'; nametemplate=%s.html; copiousoutput
```

en mi `~/ .mailcap`. No obstante, he observado que bastante gente tiene la (mediana) decencia de enviar el mensaje tanto en HTML como en texto plano mediante un `mime/alternative`. Podemos aprovecharnos de esto diciéndole a Mutt: `alternative_order text/plain text/html`, que nos permitirá ahorrarnos el llamar a `lynx` siempre que el remitente haya incluido una versión en texto plano de su mensaje.

~/ .muttrc on asteroids

Ya para terminar, recordar que además de ficheros normales mediante el comando `source`, nuestro fichero `muttrc` puede incluir también la salida de algún comando si añadimos al final del nombre del fichero (que en este caso será un ejecutable) el carácter pipe `|`. Tal que así, por ejemplo:

```
source "~/ .mutt/mutt.py debian|"
```

Esto nos permite ampliar en gran medida la inicialización de Mutt, pues permite hacerla mucho más dinámica. Por ejemplo, uno puede escribirse un pequeño script que genere de manera automática todos los comandos relacionados con las suscripciones a listas de correo, de tal manera que si nos suscribimos a nuevas listas éstas se configuren automáticamente.

Etc., etc., etc.

Lista de enlaces de este artículo:

1. <http://bulma.net/body.phtml?nIdNoticia=1692>

E-mail del autor: asp16_ARROBA_alu.ua.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1916>