“Infobia”- Como.

Francisco José Montilla, pacopepe@insflug.org v0.98, 26 de Junio de 1998.

Este documento pretende ser una guía rápida de configuración y puesta en funcionamiento de procedimientos para conectarse a Internet a través de Infovía mediante enlaces ppp. También puede aplicarse a casos de acceso ”directo” sin mediar Infovía, o a través de retenet. Así mismo, describiré un método tal vez no muy ortodoxo pero sí sencillo y eficiente para recoger y mandar el correo a través de este tipo de conexiones.

Índice General

1 Introducción

En este documento intentaré explicar un par de métodos para establecer conexiones ppp a servidores de acceso a Internet, así como un sistema para recoger y enviar el correo a la cuenta del Servidor con el que se establece la conexión.

No soy ni pretendo ser un *gurú* en LiNUX. Lo que aquí se describe lo he extraído de correo que generosamente me enviaron en su momento usuarios de `Fidonet` e Internet, y he probado, experimentado, y buscado la forma más sencilla y fácil de hacerlo, ya que me consta que una de las cosas que más ilusión hace a los recién llegados es precisamente conectarse a Internet a través de LiNUX — cosa que por cierto, se realiza mucho más eficientemente en este sistema operativo que en otros, al obtenerse soporte directo del núcleo o *kernel* del sistema, sin tener que recurrir a "chapuzas" para ello.—

La motivación por tanto, que me ha llevado a hacer este documento ha sido el ponerlo a disposición de los demás (y por que ya estaba cansado de *forwardear* una y otra vez los mismos mensajes que previamente había recibido ;-) y como agradecimiento y *granito de arena* a toda la comunidad LiNUX.

Si encuentra cualquier error de concepto, u opina que el método se puede mejorar, o simplemente quiere hacer alguna aportación a este documento, no dude en ponerse en contacto conmigo. Estaré encantado de saberlo.

2 Requisitos.

2.1 *Hardware*

2.1.1 Módems

Está claro: `:)` además del ordenador, un módem. En cuanto al tipo de módem, siempre he recomendado lo mismo: Externo. Un módem interno sólo tiene razón de ser en la poco probable situación de no poseer UARTs rápidas (16550A). Si este no es su caso, la mejor apuesta será siempre (hablando de módems por RTC[1]) un módem externo, de cuanta mejor reputación mejor; no me gusta entrar en marcas y modelos, pero sé que esta es un pregunta frecuente en aquellos que se disponen a actualizar o adquirir uno, por lo que haré una excepción.

Las marcas aconsejables son las de siempre: *USR*, en sus modelos *Sportster* o *Courier*, siempre que no sean *winmodems* , *Supra* (actualmente *Diamond*) en su modelo *Fax*, *Zyxel*, etc. Siempre y cuando no sean *winmodems*. Recientemente ha pasado uno por mis manos de fabricación nacional, cuyo nombre era *Vayris* (o algo así), que no estaba nada mal. En cuanto a velocidades, comprar menos de 33.6 Kbps hoy en día es un desperdicio.

Una cosa sí que está muy clara en todo caso: rehuir como de la peste de los denominados *winmodems*; éstos no poseen chip inteligente, y realizan sus funciones lógicas a través de software, que normalmente no está disponible (siendo poco probable que alguna vez lo esté, dada la escasa calidad de dicho "hardware") en LiNUX y otros SOs.

Modelos y Marcas conocidos de éstos son:

- *USR Sportster Winmodem*
- *IBM Aptiva MWAVE*
- *Sitre Super PC336*
- *Zoltrix VoiceMail 33600 Win HSP*
- Módems con chip *Rockwell RPI*
- También he recibido últimamente frecuentes preguntas de propietarios de módems de chip *PCTEL HSP*, que desafortunadamente, no podrán beneficiarse de las capacidades de conexión de Linux, debido a que pertenece a la funesta categoría de *winmodems*.

Resumiendo: **NADA de winmodems**, a ser posible NO internos, y nada de PnP.

[1] Red Telefónica Conmutada, en oposición a la reciente RDSI

2.1.2 Configuración del módem

Un problema frecuente es el hecho de que "*el módem no marca*". En el 90% de los casos, y asumiendo que no son *winmodems*, se debe a estar intentando que LiNUX comparta IRQ, bien por estar usando un módem interno, en la típica configuración DOS COM4, irq 3, bien por tener la IRQ asignada a ese puerto ocupada con otro dispositivo.

Linux NO puede compartir IRQs, y esto no es un fallo, es una necesidad. Así pues, la estrategia es:

1. Configurar el módem para que su puerto interno pase a ser el COM2 (`/dev/ttyS1` en Linux); la configuración en Linux por defecto para este puerto es irq 3, dirección base 0x02f8. Así pues, si el módem admite ser configurado **por jumpers** de tal modo, nos habremos ahorrado trabajo. No olvidar desactivar el COM2 de la Placa madre.

2. Si lo anterior no puede hacerse, pero el módem admite (**por jumpers, nada de PnP!**) al menos cambiar la IRQ que usará el puerto interno del módem, asignarle una IRQ distinta de la 3 o 4. Si se tiene tarjeta de sonido, posiblemente ésta ocupe la IRQ 5, y la 7 es del LPT1 aunque no se emplee si utilizamos el driver de *polling* del núcleo. La 9 está en cascada con la 2, así que una apuesta segura son las IRQs de la 10 a la 12.

3. Si esto tampoco puede hacerse, la estrategia a seguir es desactivar el COM2 en la placa base, mediante los jumpers o como suele ser posible con las placas Pentium, mediante la BIOS, a fin de dejar la IRQ 3 libre, que será usada por defecto por el puerto interno del módem (COM4); o bien cambiar la IRQ utilizada por el COM2 de la placa, a fin de que pueda ser usada por el puerto interno del módem.

4. Una vez nos hemos asegurado de que el *hardware* está empleando los recursos que debe, hemos de decírselo al *software*. Si hemos conseguido poner el puerto interno del módem como COM2 (¡y hemos desactivado el de la placa!), no hay más que hacer, todo lo que sigue está pensado para ese caso. Una respuesta típica del comando `setserial` sería:

   ```
    ~]# setserial /dev/ttyS1
   /dev/ttyS1, UART: 16550A, Port: 0x02f8, IRQ: 3
   ```

5. En caso de haber cambiado la IRQ a utilizar por el puerto interno del módem, (COM4) deberemos decírselo a Linux cada vez que lo arranquemos (incluyendo el comando en el *script* de arranque `/etc/rc.serial`, si nuestra distribución es una *Slackware*, o en `/etc/rc.d/rc.serial` si es *RedHat*). Si le hemos puesto en la IRQ 10, y tenemos un módem superior a 22.8 Kbps, el comando (o la línea a poner en dicho *script*) sería:

   ```
   setserial -v /dev/ttyS3 irq 10 spd_vhi
   ```

   Con el cual le indicamos que el COM4 (`/dev/ttyS3`) usará la IRQ 10, y que bloquee el puerto a alta velocidad (*SPeeD Very HIgh*).

   El parámetro `-v` hará que el comando devuelva la información de configuración final del puerto.

Si se continúa com problemas, e incluso si no los tiene, es recomendable leer el *Serie-Como*, disponible en `http://www.insflug.org` o en el directorio de traducciones (`pub/Linux/docs/HOWTO/translations/es`) disponible en cualquier *mirror* de *sunsite*.

2.2 *Software*

Básicamente, lo único necesario es tener soporte ppp ya por parte del núcleo o *kernel* o por módulos, en cuyo caso son necesarios tener cargados `shlc.o` y `ppp.o`, mediante por ejemplo la orden:

```
modprobe ppp
```

existen métodos más modernos como `kerneld` y otros en los que la carga se automatiza al llamar al otro requisito, el "demonio" o *daemon* `pppd`, que suele instalarse en un paquete aparte. Téngase en cuenta que si se emplea un kernel posterior al 1.3.95 ha de utilizarse una versión de `pppd` posterior a la 2.2.0e. Para el kernel 1.2.13 vale a partir de la 2.1.2d[2].

Los fuentes de distribución del kernel contienen un módulo de compresión ppp, `bsd_comp.o`, que por problemas de copyright no puede ser compilado sino es como módulo, ni cargado automáticamente por `kerneld`. El uso de este módulo mejora el rendimiento de la conexión, especialmente en lo referente a transferencia de ficheros. Para evitarnos el tener que cargarlo "a mano" tras `shlc.o` y `ppp.o`, podemos crear un alias para `pppd`:

```
alias pppd="pppd; modprobe bsd_comp"
```

que incluiremos en el fichero de personalización correspondiente, p. ej. `/etc/bashrc` si queremos que afecte a todos los usuarios globalmente, o `~/.bash_profile` para cada uno de los usuarios de RedHat o `~/.bashrc` para Slackware.

Ciertamente, no es una solución muy elegante, pero funciona :-).

Para conectarse a un *ISP* (*Internet Service Provider*, o Proveedor de Acceso a Internet) a través de nuestra queridísima Infovía, pueden utilizarse los métodos que a continuación describo:

3 Conexiones a través de Infovía.

3.1 Método "A"

1. Fichero `/etc/resolv.conf`.
   En el caso cada vez más corriente, en que obtengamos nuestra dirección por asignación dinámica, se ha de conocer la dirección en notación decimal del servidor de nombres o *nameserver* del ISP que nos proporciona acceso. Esta información se la ha de proporcionar su ISP, y *generalmente* será de la forma `194.xxx.yyy.zzz`. (En la posición `zzz` generalmente suele emplearse el 2)

   El dato restante es el nombre de dominio de su servidor, que será el mismo que aparezca en su dirección de correo *email*, es decir, todo lo que se encuentra tras la arroba. En mi caso, `pacopepe@insflug.org` sería por tanto `insflug.org`.

   Una vez conocemos estos datos, editamos (con `vi`, por ejemplo) el fichero `/etc/resolv.conf`, de modo que añadimos:
   `/etc/resolv.conf`

   ```
   domain insflug.org
   nameserver 194.xxx.yyy.zzz
   ```

2. Elaboramos el fichero `/etc/ppp/options` (Sólo con versiones de `pppd` iguales o inferiores a la `2.2.x`)

   ```
   connect /etc/ppp/infovia
   crtscts
   modem
   passive
   +ua /etc/ppp/infoviappp   # Ojo en pppd version 2.3.x, opcion no valida
   noipdefault
   debug
   defaultroute
   asyncmap a0000
   /dev/modem # (Este fichero es un enlace a /dev/ttySX)
   38400 # (Siempre que su modem soporte esa velocidad)
   ```

[2] Esto obviamente está ºbsoleto".

Nota: Si nuestro `pppd` es de version `2.2.3` o superior, deberemos modificar el fichero `/etc/ppp/options`, suprimiendo la línea `+ua /etc/ppp/infoviappp` y añadiendo:

```
...
+pap
user id@dominio
...
```

Así, utilizará en su lugar el fichero `/etc/ppp/pap-secrets` para la autentificación:

```
infovia       *       infovia
id@dominio    dominio clave
```

Para más información sobre `pap-secrets` ver apartado correspondiente de la sección 3.2.

`/dev/ttySX` es el fichero de dispositivo correspondiente al puerto donde tengamos el módem, generalmente, el COM2 si lo vemos desde msdos, o `/dev/ttyS1` en LiNUX. En caso de que en su sistema no exista `/dev/modem`, puede crear un enlace o *symlink* al puerto donde se encuentra el módem, con la orden:

```
ln -s /dev/ttyS1 /dev/modem
```

Siempre que el COM2 sea el que esté usando el módem. Puede por supuesto incluir directamente `/dev/ttyS1` en lugar de `/dev/modem` en el anterior script si lo prefiere.

Para los usuarios de *Intercom* o *bankinter*, los ficheros `options` serían:

- Intercom

```
connect /etc/ppp/infovia
crtscts
modem
passive
noipdefault
ipcp-accept-local
ipcp-accept-remote
debug
defaultroute
lock
asyncmap a0000
/dev/modem
38400
```

- Bankinter

```
connect /etc/ppp/infovia
crtscts
modem
defaultroute
lock
/dev/modem
38400
```

Los permisos del anterior script pueden ser 640 en forma octal o

```
-rw-r-----   1 root     root
```

lo cual podemos conseguir con la orden:

```
chmod 640 /etc/ppp/options
```

3. Fichero `/etc/ppp/infovia`

```
#!/bin/sh
/usr/sbin/chat -v  "" atdt055 CONNECT ""
```

Este fichero debe de hacerse ejecutable, con la orden por ejemplo:

```
chmod 744 /etc/ppp/infovia
```

4. Fichero `/etc/ppp/infoviappp`

```
su_login
su_password
```

`su_login` quiere decir `nombre@proveedor`, en mi caso `pacopepe@insflug`. Este fichero es especialmente delicado, ya que contiene la contraseña o password de acceso al ISP, por lo que conviene tener cuidado con sus permisos; yo no soy un gurú en eso, si alguien con más experiencia me recomienda otro tipo de permisos, se lo agradeceré, yo por ahora lo tengo como 640, por lo que con la orden

```
chmod 640 /etc/ppp/infoviappp
```

quedarían establecidos los permisos.

5. Ejecutar, como `root`, `pppd`. Al momento se escuchará marcar al módem, y una vez establecida la conexión, se escuchará actividad por parte del disco duro; también, en el caso de poseer un módem externo, se observará las luces de cd, sd y tr encendidas o parpadeando; en caso de ser interno, podemos constatar que la conexión está establecida correctamente, y que por tanto, el dispositivo ppp0 ha sido creado, con una orden como "`top`" o "`ps`" en la que se observará como proceso activo.

También podemos observar el proceso de conexión conmutando a otra VC, y tecleando la orden

```
tail -f /var/log/messages
```

lo cual nos mostrará, en caso de problemas, los fallos que están ocurriendo. Un proceso de conexión normal aparecería como:

```
May 23 01:51:00 beastie pppd[4485]: pppd 2.1.2 started by root, uid 0
May 23 01:51:00 beastie pppd[4488]: Connecting with /etc/ppp/infovia
May 23 01:51:02 beastie chat[4490]: send (atdt055^M)
May 23 01:51:02 beastie chat[4490]: expect (CONNECT)
May 23 01:51:23 beastie chat[4490]: atdt055^M^M
May 23 01:51:23 beastie chat[4490]: CONNECT -- got it
May 23 01:51:23 beastie chat[4490]: send (^M)
May 23 01:51:23 beastie pppd[4488]: Connected...
May 23 01:51:24 beastie kernel: ppp: channel ppp0 mtu = 1500, mru = 1500
May 23 01:51:24 beastie kernel: ppp: channel ppp0 open
May 23 01:51:24 beastie pppd[4488]: set kernel debugging level to 0
May 23 01:51:24 beastie pppd[4488]: Using interface ppp0
May 23 01:51:24 beastie pppd[4488]: Connect: ppp0 <--> /dev/modem
[...]
May 23 01:51:25 beastie pppd[4488]: ipcp: received ADDR
May 23 01:51:25 beastie pppd[4488]: (172.16.1.1)
May 23 01:51:25 beastie pppd[4488]:  (ACK)
May 23 01:51:25 beastie pppd[4488]: ipcp: returning Configure-ACK
May 23 01:51:25 beastie pppd[4488]: fsm_sdata(IPCP): Sent code 2, id 1.
May 23 01:51:25 beastie pppd[4488]: fsm_rconfnakrej(IPCP): Rcvd id 1.
May 23 01:51:25 beastie pppd[4488]: local IP address 194.179.123.229
May 23 01:51:25 beastie pppd[4488]: fsm_sdata(IPCP): Sent code 1, id 2.
May 23 01:51:25 beastie pppd[4488]: IPCP: sending Configure-Request, id 2
May 23 01:51:25 beastie pppd[4488]: fsm_rconfack(IPCP): Rcvd id 2.
```

```
May 23 01:51:25 beastie pppd[4488]: ipcp: up
May 23 01:51:25 beastie pppd[4488]: local  IP address 194.179.123.229
May 23 01:51:25 beastie pppd[4488]: remote IP address 172.16.1.1
```

6. Para finalizar la conexión podemos emplear el script que suele acompañar al paquete pppd, `ppp-off`, o bien "matar" directamente el proceso una vez identificado su PID con `ps`; para ello, si una vez ejecutado `ps` observamos la respuesta:

   ```
   PID   TTY   STAT  TIME  COMMAND
   58    v01   S     0:01  -bash
   [...]
   353   v03  R      1:12  pppd
   [...]
   ```

   la orden

   ```
   kill -9 353
   ```

   matará el proceso. No obstante, algunas personas han experimentado "cuelgues" de sus servidores si no finalizan la conexión con métodos "civilizados" como el script `ppp-off`.

   Uno puede hacerse un `ppp-off` rudimentario mediante el comando:

   ```
   killall pppd
   ```

   Si se quiere saber más sobre los comandos de este script, consulte el comando `chat` y la documentación sobre `pppd`.

3.2 Método "B"

El mismo que el empleado para conectar sin mediar Infovía, descrito en la sección 4 a excepción de:

1. Fichero `/etc/ppp/pap-secrets`, que quedaría así:

   ```
   infovia             *                  infovia
   id@dominio          *                  su_password
   ```

   donde `id@dominio` sería, en mi caso, `pacopepe@insflug`, es decir, su dirección *email* sin el `.es` del dominio perteneciente a España.

   Este fichero es especialmente sensible por contener el password, por lo que se aplica lo dicho anteriormente para el fichero `/etc/ppp/infoviappp` en la sección 3.2, punto número 4.

   Como se puede observar, lo único que varía es que se añade la línea referente a Infovía.

2. Cambiar la variable `NUMERO` del script `/usr/local/bin/infovia` por `055`, como corresponde a Infovía.

4 Conexiones sin mediar Infovía.

En el caso de que tengamos acceso directo a un servidor, los scripts y ficheros necesarios serían los siguientes:

1. Script `/usr/local/bin/infovia`

```
#!/bin/sh

LOCKDIR=/var/spool/uucp
DEVICE=modem
NUMERO=numero_del_Proveedor

if [ -f $LOCKDIR/LCK..$DEVICE ]
then
  echo /dev/$DEVICE "El modem esta ocupado."
  exit 1
fi

/usr/lib/ppp/fix-cua $DEVICE
(
    stty 38400 -tostop crtscts

    if /usr/lib/ppp/chat ABORT "NO CARRIER" ABORT BUSY "" ATZ0 OK ATDT$NUMERO CONNECT ""
    then
      pppd /dev/$DEVICE 38400 crtscts modem lock mtu 1500 defaultroute noipdefault user id@dominio
      sleep 10
      route add default ppp0
      exit 0
    else
        echo "La llamada PPP ha fallado." 1>&2
        exit 1
    fi
) < /dev/$DEVICE > /dev/$DEVICE
```

en donde:

- En la variable `NUMERO=` deberá reflejar el número de su Proveedor.
- En `id@dominio` tendrá que poner su dirección *email* sin el `.es` del dominio perteneciente a España.

Este script ha de ser ejecutable, por lo que tenemos que otorgarle permisos de ejecución, con una orden como por ejemplo:

```
chmod 750 /usr/local/bin/infovia
```

A decir verdad, este script lo puede colocar donde quiera, si bien `/usr/local/bin/infovia` sería la situación más "estándar".

2. Fichero `/etc/ppp/pap-secrets`

```
id@dominio            *                   su_password
```

nuevamente, se aplica lo dicho en la sección 3.2.

3. Fichero `/etc/resolv.conf`
aquí se aplica lo mismo que en la sección 3.1 punto número 1.

4. A partir de aquí, se aplica lo mismo que en la sección 3.1, punto 5, a excepción de que por supuesto, no ha de ejecutarse pppd, ya que lo hacemos ejecutando el script `/usr/local/bin/infovia`.

5. **ATENCIÓN usuarios de RedHat**
Si el sistema LiNUX que tiene instalado pertenece a una distribución RedHat, deberá tener en cuenta lo siguiente:

   - En el script `/usr/local/bin/infovia` deberá modificarse las líneas 13 y 17, por:

```
[...]
/usr/lib/ppp/fix-cua $DEVICE  -->  /usr/sbin/fix-cua $DEVICE
[...]
if /usr/lib/ppp/chat...  --> if /usr/sbin/chat...
[...]
```

ya que la localización de dichos ficheros en RedHat está en esos directorios.

- Para ciertos programas que hacen uso del módem, como el `binkley`, y otros, resulta inocuo y muy conveniente crear el enlace o *symlink* siguiente:

```
ln -s /var/spool /usr
```

Para obtener una visión más completa y detallada en lo que a `ppp` se refiere, recomiendo hacerse con la traducción del **PPP-Como**, realizada por Rafael Agundo, `ragundo@bitmailer.net`. En la sección 8 se detallan los servidores donde obtenerlo.

5 Gestión de Correo de Internet.

A continuación describiré dos métodos para gestionar el correo en el caso que nos ocupa, una máquina aislada, con conexiones esporádicas a su Servidor de Acceso a Internet. El método B es desde luego, poco ortodoxo y se puede mejorar mucho, por lo que una colaboración en lo que a configuraciones "ideales" de red de este tipo de máquinas será harto agradecida.

5.1 Método "A" o fácil y *güindosero* ;-).

Instalar, usar y configurar Netscape, Mosaic u otro navegador con capacidad de gestionar correo, news, etc.

Como me consta que la inmensa mayoría de los que empiezan a usar Linux o bien no poseen una cantidad desmesurada de RAM, ni les sobra disco duro como para sacrificar más de 6 megas en el Netscape, y además desean aprender a usar métodos más *nixeros y eficaces de gestión de correo, propongo el siguiente (más fácil de configurar incluso que el netscape) método:

5.2 Método "B".

5.2.1 Requisitos

1. **Popclient.** Se precisa instalar el paquete Popclient. En caso de que la versión de éste use `perl`, se deberá instalar este último también.
2. popclient se ha quedado desfasado últimamente, siendo `fetchmail` el que más se emplea ahora por ser más seguro.
3. **Sendmail+IDA**. No, no os asustéis ;-) El `sendmail+IDA`, que viene en la inmensa mayoría de las distribuciones, lo tendremos configurado con editar dos líneas.

5.2.2 Configuración del sistema.

1. Crear una cuenta en la máquina con el mismo identificativo que se tenga en el Proveedor. Por ejemplo, mi identificativo o *login* en mi ISP es `pacopepe`, cosa fácilmente deducible debido a mi dirección de correo *email*; por tanto, creo una cuenta en el sistema con login `pacopepe`, con el comando `adduser`: (por supuesto, hay que hacerlo como `root`).

   Supongamos el login "`probancio`":

```
 /home/linuxdoc-sgml-1.5/working]# adduser probancio

Looking for first available UID... 502
Looking for first available GID... 502

Adding login: probancio...done.
Creating home directory: /home/probancio...done.
Creating mailbox: /var/spool/mail/probancio...done.

Don't forget to set the password.
```

ahora, le asignamos un password:

```
 /home/linuxdoc-sgml-1.5/working]# passwd probancio
Changing password for probancio
Enter an empty password to quit.
New password (? for help):
New password (again):
Password changed for probancio
```

y tenemos creada su cuenta.

popclient

1. Ahora creamos el siguiente script, que será el que ejecutemos para recoger el correo, al que llamamos, por ejemplo,
   `/usr/local/bin/getmail`

   ```
   #!/bin/sh
   #
   # getmail, para bajarnos el correo...
   #
   PATH=/bin:/usr/bin:/usr/local/bin
   echo Bajando el correo.....
   popclient -3 -u <nombre_usuario> -p <password_del_ISP> -o /var/spool/mail/login <servidor_POP>
   ```

   Dado que este fichero contiene datos delicados como las passwords del ISP, lo protegeremos dándole los permisos adecuados (`700` es lo recomendable).

   Donde en:

   <nombre_usuario>
   pondremos nuestro identificativo, en mi caso, `pacopepe`.

   <password_del_ISP>
   Pues exactamente eso, la clave con la que accede a su servidor.

   <...login>
   Como se observará tras crear la cuenta que describimos anteriormente, en `/var/spool/mail/` se creará un fichero de igual nombre que el *login* de dicho usuario; en el caso supuesto anterior, `probancio`, este fichero sería `/var/spool/mail/probancio`.

   <servidor_POP>
   Aquí ha de ponerse la dirección de vuestro servidor POP; en mi caso (y suele ser común) `pop03.insflug.org`.

Nota: Al elaborar el script prescindiremos de los signos "<" y ">"; en el ejemplo están simplemente para resaltar los parámetros a completar.

Juan Manuel Villar Navarro `juanma@gaps.ssr.upm.es` apunta que en las versiones `3.xx` del `popclient` no se puede dar por línea de comandos la contraseña del ISP, (con `-p`) para ello ha de usarse el fichero `~/.poprc`, en el que podemos definir otros parámetros de comportamiento, como el que mantenga los mensajes en el servidor (`keep`) en caso de que estemos de pruebas, o por cualquier otra razón.

Iñigo González `nexus@adv.es` recomienda usar versiones del `popclient` superiores a la `3.0b6`, además de sugerir el uso de un programa filtrador de correo como `procmail`, para lo que deberemos añadir al comando `getmail` el parámetro `-m procmail`.

`fetchmail` En caso de usar fetchmail, un cliente muy potente y cuya documentación es bastante clara y precisa, la configuración personal se almacena en el fichero del directorio personal del usuario, `~/.fetchmailrc`.

Un ejemplo del mismo:

```
poll host-servidor-pop proto pop3 user usuario password pass_usuario is usuario here;
```

donde

host-servidor-pop
: sería el nombre del la máquina servidora de correo vía pop del proveedor que utilicemos;

pop3
: sería el protocolo a emplear, ya que `fetchmail` soporta otros también, como *pop2* (obsoleto) *imap2bis* *imap4* y *apop* y *kpop*.

seguidamente, le otorgaremos permisos de lectura/escritura únicamente para el propietario, hecho muy importante, ya que de lo contrario podrían ser accesibles las contraseñas, e incluso el programa se negaría a funcionar:

```
chmod 600 .fetchmailrc
```

`fetchmail` ofrece una serie de prestaciones adicionales, como temporización, reenvío, funcionamiento en modo *daemon* etc... Es un cliente muy potente y recomendable en cuanto a seguridad se refiere.

En caso de emplearlo, no haría falta el *script* `getmail`, bastaría con invocar a `fetchmail` a secas.

`sendmail`

1. Modificación de la llamada al demonio `sendmail`, hecha normalmente en el arranque desde el script `/etc/rc.d/init.d/sendmail.init`, (RedHat) o `/etc/rc.d/rc.M` (Slackware) buscar la línea que dice algo así como `daemon sendmail ....` en RedHat, o `/usr/sbin/sendmail -bd -q 15m` en Slackware, y modificarla, editándola para que quede:

   ```
   [...]
           .... sendmail -bd -q2d
   [...]
   ```

Esto lo que hace es que `sendmail` no intente continuamente mandar el correo que haya *en la cola* para salir, o en *spool*, ya que lo haremos nosotros manualmente.

Si no hacemos esto veremos que al enviar un email estando desconectados, el programa donde estemos (el `pine`, por ejemplo) se quedará "congelado" un buen rato, debido a que sendmail intentará enviar inmediatamente el email, y no encontrará el destino, hasta que finalmente se produzca un *timeout*.

2. Modificación de `/etc/sendmail.cf`. Aquí buscaremos una línea que comienza por `DS`:

```
# "Smart" relay host (may be null)
# DS
```

y la modificaremos para que quede reflejado nuestro servidor SMTP o de correo saliente: (en mi caso, `smtp.insflug.org`):

```
# "Smart" relay host (may be null)
DSsmtp.insflug.org
```

ahora buscaremos otra que comienza por `DM`:

```
# who I masquerade as (null for no masquerading)
# DM
```

y la modificamos para que refleje el dominio de nuestra dirección de correo, en mi caso `insflug.org`:

```
# who I masquerade as (null for no masquerading)
DMinsflug.org
```

Con esto, lo que hemos hecho es básicamente, "enmascarar" nuestra dirección en la máquina propia; supongamos que nuestra máquina se llama `beastie.insflug.org` y enviamos un mensaje sin la modificación anterior; el mensaje llegará correctamente a su destino, pero no podrá ser respondido, ya que la dirección de retorno no existirá, al figurar la de nuestra propia máquina, que en nuestro caso ficticio sería `probancio@beastie.insflug.org`, en lugar de la de la cuenta de nuestro ISP, que es `probancio@insflug.org`.

Realmente, lo único que enmascaramos es el dominio, de ahí la necesidad de crear una cuenta en nuestra máquina con el mismo login que en nuestro ISP (`probancio` en este caso); la línea `DS...` hace que sendmail rute todos los mensajes salientes hacia internet a través de nuestro servidor SMTP, que hace de servidor de *relevo* hacia internet. Podríamos no decirle nada, y dejar que se encargara de contactar y enviar directamente con el servidor de correo entrante de cada dirección, pero eso haría más lento el envío de los correo, además de que es mucho más rápida la transferencia con nuestro ISP, al no tener que salir a internet siquiera.

`DM...` cambia los `from` de nuestros mensajes por nuestra verdadera dirección en el ISP.

5.2.3 Cómo escribir

Para responder o escribir nuestro correo podremos usar cualquier programa escritor de correo, los simples como `mail` o `mailx`, un poco más completos como el facilísimo `elm`, o `pine`, el modo de correo del versátil `emacs`, etc... recordando **siempre** hacer uso de estos programas desde la cuenta que creamos para tal fin (la de `probancio` en nuestro caso ficticio).

5.2.4 Procedimiento.

1. Establecer la conexión `PPP` con nuestro servidor, con cualquiera de los métodos descritos en las secciones 4 o 3. Esto se hará normalmente como `root`.

2. Ejecutar el script `getmail` en caso de que queramos recoger el correo; en caso de querer mandar el correo pendiente por salir, teclear la orden:

   ```
   sendmail -q
   ```

   que ordenará a sendmail a enviar el correo. (el parámetro `-q` viene de *queue* o la "cola" de correo pendiente por salir).

Por supuesto, los procedimientos para establecer la conexión y recoger/mandar correo se pueden automatizar escribiendo scripts sencillos, pero eso lo dejo ya al gusto y según las circunstancias de cada uno. Estaré encantado de recibirlos, a fin de incluirlos en la próxima versión de este *COMO*.

5.3 Método "C".

Empleando clientes de correo capaces de enmascarar al usuario/dominio, podemos prescindir de la fase de configuración del enmascaramiento de dominio del `sendmail`. El cliente de correo (*MUA*[3] `mutt`, puede hacer esto, a nivel tanto de dominio como de usuario, entre otras muchas prestaciones que harán las delicias de los amantes del modo texto: gestión `pgp` integrada, *threads*, color... Un cliente muy recomendable. Su servidor de ftp primario es:

`ftp://ftp.cs.hmc.edu/pub/me/mutt`

Es posible también prescindir de la "*chapucilla*" de tener que emplear el mismo usuario que en el proveedor empleando un *MTA*[4] de configuración más flexible y cómoda que `sendmail`, como el prometedor `qmail`, fácilmente obtenible en Internet, que además ofrece muchas otras prestaciones, sin la fragilidad en cuanto a seguridad de `sendmail`, y menos exigente en cuanto a recursos, lo que le hace ideal para listas de correo..

6 Agradecimientos

Mi más sincero agradecimiento a todos los contertulios de R34.LINUX, y a los de la Lista de correo de RedHat, así como a José L. Navarro Simón, `2:345/102.36` y Miguel Cruz por los scripts a través de infovía, a Urko Lusa, `2:344/25.8` por los de acceso directo, y a Eric S. Pulley, `pulley@dabus.com` por las explicaciones con lo relacionado con el correo.

A Carlos Terrón por sus correcciones, a Juan Manuel Villar Navarro `juanma@gaps.ssr.upm.es` por sus apuntes sobre `popmail`, y a Iñigo González `nexus@adv.es` por las sugerencias sobre `procmail`, y a Jesús Fuentes Saavedra `jesus.fuentes@etsi.uva.tel.es` por el detalle del `bsd_comp`, y tantísimas otras cosas...

A Antonio Verdejo García aka *wait_man*, `wait_man@hotmail.com` por su lista de winmodems y tantas otras críticas y sugerencias.

7 Copyright

[3]*Mail User Application*, aplicación de gestión de correo a nivel usuario)

[4]*Mail Tranfer Agent*, Agente de Gestión de transferencia de Correo

3. Si se distribuye el Trabajo parcialmente, deben de incluirse instrucciones para poder obtener la versión completa (en forma impresa o electrónica), así como los medios para conseguirla.

4. Pueden ser reproducidas pequeñas porciones como ilustraciones para revistas o citas para otros trabajos sin esta notificación de permiso si se cita apropiadamente su procedencia.

8 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del INSFLUG para más información al respecto.

En la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

Francisco José Montilla, `pacopepe@insflug.org`.