

Mini-COMO para inn2+suck+varios servidores de news.

Han Solo <hsolo@escomposlinux.org>

v. 0.1 24 de Marzo de 2000

Este documento describe cómo montar de manera rápida un servidor inn2 local e intercambiar noticias de varios servidores con suck.

Índice General

1	Descargo y licencia.	2
2	Dedicatoria.	2
3	Consideraciones previas.	2
4	Cómo funciona todo esto.	2
5	Elementos necesarios.	3
6	Configuración de inn.	3
6.1	El sistema de almacenamiento.	3
6.2	Ficheros de configuración.	4
6.3	inn.conf	4
6.4	newsfeeds	4
6.5	incoming.conf.	6
6.6	expirectl.	7
6.7	nntp.access.	7
6.8	Comprobando los ficheros.	8
6.9	Creación de los ficheros db.	8
6.9.1	Configurando los ficheros db por primera vez.	8
6.9.2	Configuración a partir de una versión anterior de inn.	8
6.10	Si la cosa no funciona...	9
6.11	...y si funciona	9
6.12	Añadiendo grupos.	10
7	Configuración de suck.	10
7.1	get-news.conf.	11
7.2	sucknewsrsrc.	11
7.3	Filtros.	12
7.4	Configuración del segundo servidor (o los siguientes).	12

7.5	baja-news, el meollo de todo esto.	13
7.6	Bugs, apaños y otras trampas...	13
8	Feedback.	14

1 Descarga y licencia.

Lo habitual en estos casos, yo no garantizo nada, no me responsabilizo de lo que pueda estropearse si haces caso de este documento, no tiene porqué funcionar en tu sistema y lo único que puedo decir es que a mi me ha funcionado.

Este documento se distribuye bajo licencia GPL y todas esas cosas que ya sabéis; es decir, lo podéis copiar, manipular, traducir... pero sería un detalle que me mandarais una copia.

2 Dedicatoria.

Este documento está dedicado a Benjamín Albiñana, por dos razones: La primera es que se saque la espinita de `inn2`; no se puede ir por ahí, presumiendo de ser un tipo de pelo en pecho, y funcionar con `leafnode`. La segunda es para que luego no diga que los emacseros no tenemos nuestro corazoncito...

3 Consideraciones previas.

Hay varias maneras de configurar `inn2` frente al tradicional `inn`. Ahora, se pueden almacenar las news en un `spool`/ local más compacto, en lugar de la forma tradicional de una fichero por cada mensaje. Yo he escogido hacerlo a la manera tradicional por dos razones: le primera es que estoy actualizando desde un sistema `inn` 1.7, y ya tenía artículos en el `spool` tradicional. La segunda es que (no lo he comprobado), `suck` no es capaz de intercambiar mensajes si están almacenados en el nuevo formato. Yo tengo el sistema funcionando a partir de los script que vienen con `suck` modificados. Si usas el nuevo sistema de almacenamiento de almacenamiento de `inn2`, tendrás que usar `newsx` para el intercambio de noticias. El usar varios servidores, no sé cómo se podrá hacer entonces y te lo tendrás que currar tú ;-).

Yo intercambio artículos con dos servidores: `news.cis.dfn.de` y `news.antakira.conf`. El primero es un servidor alemán, que requiere registrarse previamente en `http://www.cis.dfn.de`

Por otra parte, he escrito este documento en `sgml` utilizando el DTD `linuxdoc`. Últimamente, parece que es estándar de moda es `DocBook`, pero no tengo tiempo ahora de ponerme a explorar ese DTD, así que si a alguien le apetece, puede pasarlo a `DocBook`. Por otra parte, a mí me gusta más la salida del `linuxdoc` clásico.

4 Cómo funciona todo esto.

La idea de todo esto es tener un sistema de news local, para leer las news offline. Para ello, instalamos un servidor de news local, `inn`, que es al que se conectarán nuestros clientes (Gnus, Netscape, e incluso, por qué no, `slrn`). cuando escribamos, lo haremos a nuestro servidor. Estos mensajes que escribamos, lo tendremos que mandar a otros servidores de news, generalmente los de nuestro ISP. Por otro lado, para necesitaremos bajar las news del servidor de nuestro ISP a nuestro servidor local. De todo este intercambio de mensajes se encarga `suck`.

Suck es el que “llama” a los distintos servidores, se identifica si es necesario, baja los artículos nuevos y envía los que hayamos escrito. `suck` se comporta como un cliente ante ambos servidores, el local y el remoto.

Esto es así, porque existen dos protocolos para el intercambio de noticias: el `nntp`, usado para el intercambio entre servidores, y el `nntp`, para el intercambio con clientes. Para bien o para mal, nuestro servidor no puede intercambiar artículos mediante `nntp`, ya que los servidores “grandes” sólo admiten este tipo de intercambio con un puñado de servidores “escogidos”.

Así pues, tendremos que configurar por separado `inn` y `suck`, y a eso dedicaremos los siguientes capítulos.

5 Elementos necesarios.

Pues no son demasiados, `inn2` y `suck` ;-) Yo estoy usando una distribución *Debian Potato*, con los paquetes siguientes:

```
$ dpkg -l suck inn2
Desired=Unknown/Install/Remove/Purge/Hold
| Estado=No/Instalado/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: mayúsc.=malo)
||/ Nombre          Versión          Descripción
+++-----
ii suck              4.2.2-4         Small newsfeed from an NNTP server with stan
ii inn2              2.2.2.2000.01. News transport system 'InterNetNews' by the
```

Si usas otra distribución, las cosas podrían cambiar un poco, pero no creo que demasiado.

6 Configuración de inn.

Para empezar, supondré que ya tienes `inn` instalado. No entraré pues en los detalles de la compilación ni en la instalación desde paquetes `.deb`, `.rpm` o similares. Para eso está la documentación de la instalación de `inn` y los manuales de los sistemas de paquetes. Sólo hablaré de los ficheros de configuración.

6.1 El sistema de almacenamiento.

Como dijimos más arriba, hay varias formas de almacenar los artículo que bajemos en nuestro servidor local: la primera es el sistema *tradicional*, en el que cada mensaje se almacena en un fichero, y estos ficheros se ordenan en directorio siguiendo el nombre de la jerarquía a la que pertenecen; esto es, un artículo de `es.comp.os.linux.misc`, se guardará en `/var/spool/news/es/comp/os/misc/`. Según la documentación, el problema de este sistema es que el acceso a los artículos es un cuello de botella en sistemas con mucha carga.

La segunda forma de almacenar los mensajes es el tipo *timehash*, que es parecido al tradicional, pero la clasificación en directorios se hace en base a la fecha de llegada. El cuello de botella se reduce porque hay menos archivos en cada directorio.

La tercera forma es el método *cnfs*. Los artículos se guardan en un *buffer* preconfigurado. Cuando los artículos van llegando, se van borrando automáticamente los más antiguos.

Como dijimos antes, nos quedaremos con el sistema tradicional, que es el compatible con `suck`.

6.2 Ficheros de configuración.

En principio, al menos en Debian, los ficheros de configuración de `inn` están en el directorio `/etc/news/`. En adelante, nos referiremos a ellos sólo por su nombre. Otro detalle importante, es que todas las modificaciones a los ficheros de configuración hay que hacerlas como usuario `news`; de lo contrario, habrá cosas que no funcionen aunque aparentemente estén bien. De hecho, lo más probable es que `inn` ni siguiera arranque, y os aviso que los mensajes de error de `inn` son bastante “opacos”.

6.3 `inn.conf`

Este es uno de los ficheros más importantes de la configuración de `inn`. Tiene multitud de opciones que se describen en su página `man (man inn.conf)`. Conviene leerla con calma. Os pongo las que yo he cambiado:

```
organization:      The Rebel Alliance
server:            alderaan.maptel.es
pathhost:         alderaan.maptel.es
domain:           maptel.es
fromhost:         alderaan.maptel.es
```

Aparte, cambié alguna más, en lo referente a los directorios donde se almacenan los artículos, pero no recuerdo exactamente los cambios que hice, así que os pego el final del fichero para que veáis cómo lo tengo:

```
# -----
# Paths to various aspects of the news system
#
pathnews:         /usr/lib/news
pathbin:          /usr/lib/news/bin
pathfilter:       /usr/lib/news/bin/filter
pathcontrol:      /usr/lib/news/bin/control
pathdb:           /var/lib/news
pathetc:          /etc/news
pathrun:          /var/run/news
pathlog:          /var/log/news
pathhttp:         /var/log/news
pathtmp:          /tmp
pathspool:        /var/spool/news
patharticles:     /var/spool/news
pathoverview:    /var/spool/news/overview
pathoutgoing:    /var/spool/news/outgoing
pathincoming:    /var/spool/news/incoming
patharchive:     /var/spool/news/archive
pathuniover:     /var/spool/news/uniover
overviewname:    .overview
```

6.4 `newsfeeds`

Este es el fichero más oscuro y difícil de toda la configuración de `inn`. Es el que controla cómo y hacia dónde se envían los artículos, hace los enlaces en los mensajes *crossposteados*...¹. Este fichero tiene muchas opciones, y no es fácil comprender lo que hace cada una de ellas. Os sugiero una lectura cuidadosa. Como siempre, `man newsfeeds`.

¹Prometo flagelarme un par de horas después de haber escrito semejante “palabra”.

Este fichero se organiza en *feeds*². Cada *feed* se divide en cuatro secciones, separadas por “:”. Cada sección, puede ocupar más de una línea, separándolas por “\”.

El esquema, sacado de la página [man](#), e el siguiente:

```
sitename[/exclude,exclude,...]\
:pattern,pattern...[/distrib,distrib...]\
:flag,flag...\
:param
```

La primera línea indica el nombre del *feed*; aunque en el manual le llame `sitename`, el concepto es más amplio. De hecho, puede ser cualquier nombre, aunque si se coge el nombre del servidor remoto, simplifica algo las cosas, si se sabe lo que se está haciendo. Lo describiremos más adelante. Opcionalmente, puede llevar una lista de exclusión, que explicaremos más tarde con un ejemplo.

La segunda sección indica cual serán los grupos que se distribuirán en este *feed*. Lo de las `distributions` todavía no he llegado a saber para que sirve, pero tampoco lo he necesitado. No deben ser imprescindibles para un uso local. Vienen descritas en `newsfeeds` (5), así que si tienes curiosidad, ya sabes. ¡Ah!, y si me cuentas para que sirven, pues mejor.

El siguiente campo es una lista de marcas que determina el tipo de *feed* y establece ciertos parámetros. Para una descripción completa, [man newsfeeds](#).

El último campo depende de lo que hayamos puesto en el tercero. Como antes, mira la página [man](#) para más detalles

Ahora, vamos a ir viendo las distintas entradas que tengo en mi `/etc/news/newsfeeds`:

```
ME\
:*,@alt.binaries.warez.*,!junk,!control*,!local*,!foo.*\
/world,usa,na,gnu,bionet,pubnet,u3b,eunet,vmsnet,inet,ddn,k12\
::
```

La entrada `ME` es una entrada especial, y tiene que ser necesariamente la primera en `newsfeeds`. La explico. Si la entrada `ME` tuviera una lista de exclusión (en este caso no la tiene), cualquier mensaje que tuviera en el campo (cabecera) `path`: un servidor de la lista sería descartado inmediatamente. La segunda línea indica que se aceptan todos los mensajes (“*”) excepto lo que vengan de los grupos `junk`, `control*`, `local*` y `foo.*`. Además, se rechazan los que vayan a los grupos `alt.binaries.warez.*` aunque vengan por *crosspost* de otros grupos admitidos. Este matiz lo define la arroba (“@”). Los asteriscos actúan como comodines: de esta forma, `local*` incluye `local.pruebas`, `local.experimentos`...

Además, de limitar la entrada de mensajes, `ME` limita la salida de dichos grupos. Es muy útil para pruebas locales, o para tener grupos corporativos, sin que intenten distribuirse a otros servidores.

La línea siguiente (dentro del mismo campo, o sección), indica las distribuciones que se aceptan, caso de que el mensaje lleve indicada una distribución concreta.

El siguiente campo, está vacío en este caso. La opción por defecto es `Tf` (una vez más, [man newsfeeds](#) si tienes curiosidad...).

La última sección, también está vacía, lógico teniendo en cuenta que la anterior también estaba en blanco.

Lo siguiente en mi `newsfeeds` es:

```
crosspost:*:Tc,Ap,WR:/usr/lib/news/bin/crosspost
```

²He sido incapaz de encontrar una traducción satisfactoria para esta palabra. Lo más aproximado, en el contexto de este fichero, sería *fichero de alimentación*, pero me parece lamentable.

Si usas crosspost (altamente recomendado en la doc de inn), hay que añadir la opción `-L` a la variable `innflags` en el fichero `inn.conf`.

```
innfeed!:\
    !*,\
    :Tc,Wnm*,S30000:/usr/lib/news/bin/startinnfeed

controlchan!\
    :!*,control,control.*,!control.cancel\
    :Tc,Wnsm:/usr/lib/news/bin/controlchan
```

Y ahora, la parte emocionante, la que se refiere a los servidores que estamos usando:

```
# Servidor news.cis.dfn.de

cis-dfn/news.cis.dfn.de,news.dfncis.de,fu-berlin.de,uni-berlin.de,\
    news.antakira.com,kenobi.antakira.com\
    :*,!alt.sex.*,!es.pruebas,!linux.debian.spanish\
    ::

# Servidor antakira

antakira/news.cis.dfn.de,news.dfncis.de,fu-berlin.de,uni-berlin.de,\
    news.antakira.com,kenobi.antakira.com\
    :!*,alt.sex.*,es.pruebas,linux.debian.spanish\
    ::
```

Como se ve, se puede usar cualquier nombre para referirse al *sitename* o *feed*. La lista de exclusión, contiene los nombres de los servidores de los que estemos bajando artículos y sus alias. Son iguales en las dos entradas para que no se crucen mensajes entre servidores. Los nombres de los servidores que aparecen se comparan con la cabecera `"path:"` de cada mensaje. Así, un mensaje que venga de *news.cis.dfn.de* no será reenviado por ninguno de los dos servidores.

El siguiente campo indica qué grupos se enviarán por cada servidor. Para la entrada `cis-dfn`:

```
:*,!alt.sex.*,es.pruebas,!linux.debian.spanish\
```

Indica que por aquí se enviarán todos los mensajes ("`*`"), excepto ("`!`") los del grupo³ `alt.sex*`, `es.pruebas` y `linux.debian.spanish`. Por el contrario, la línea equivalente de la entrada de `antakira`:

```
:!*,alt.sex.*,es.pruebas,linux.debian.spanish\
```

Indica que por este servidor no se envíe ningún mensaje ("`!`*"), salvo `alt.sex*`, `es.pruebas` y `linux.debian.spanish`.

6.5 incoming.conf.

Este fichero indica qué máquinas se pueden conectar a nuestro servidor, como servidores. Para un uso casero, basta con nuestra máquina y las de nuestra red. Aquí se pueden definir claves para según que máquinas o grupos, el número de conexiones simultáneas... Yo tengo:

³No seáis mal pensados, no son grupos porno aunque pudiera parecerlo. Se trata de `alt.sex.jargon.ecol` y de `alt.sex.kick.me.I.am.using.emacs`

```
peer ME { hostname:
                "localhost, 127.0.0.1, alderaan.maptel.es,
                192.168.0.3" }
```

Que es totalmente redundante, pero me daba un problema cuando lo estaba configurando y creía que esa eso. Al final era otra cosa, pero me olvide de cambiarlo y ahora no tengo tiempo ;-)

Si queréis saber todas las posibilidades de este fichero, `man incoming.conf`, aunque los comentarios que vienen en él son bastante claros.

6.6 expirectl.

Transcribo la documentación de `inn` sobre este fichero:

```
The expirectl file is the configuration for the expiration policy. See
the expirectl man page for more details.
```

Vamos, que RTFM. Tampoco tiene mayor secreto. Yo puse:

```
/remember/:14
*:A:1:10:never
news.announce.*:M:1:35:90
```

6.7 nnrp.access.

Este fichero indica qué clientes se pueden conectar y si pueden leer, escribir...

Cada línea tiene el siguiente formato:

```
servidor:acceso:usuario:clave:grupos
```

`servidor` puede ser una IP, `192.168.0.3`, un rango de direcciones, `192.168.0.0/24`, un nombre de máquina, `alderaan.maptel.es`, o un grupo definido por comodines, `*.maptel.es`

`acceso` puede ser "R" (*read*, lectura), "P" (*posting*, escritura), u otras cosas más esotéricas que vienen en el manual (`man nnrp.access`).

De los campos `usuario` y `clave`, la documentación de `inn` dice:

```
See the man page for details of how to use the 'user' and
'password' fields. Leave them blank to use solely host-based
authentication.
```

Otro RTFM, pes lo dicho `man nnrp.access`. Por si os sirve de consuelo:

```
*:::*
stdIn:Read Post::*
localhost:Read Post::*
127.0.0.1:Read Post::*
alderaan.*:Read Post::*
```

Lo de la redundancia es por la misma razón que antes.

Por último, el campo `grupos` es simplemente un lista de grupos, separada por comas, en la que se pueden usar comodines ("`*`"). Como véis en lo que yo tengo, lo más sencillo es poner un asterisco para todos los grupos.

6.8 Comprobando los ficheros.

Para comprobar que todos estos ficheros están en orden, hay una utilidad: `inncheck`, que (al menos en Debian) está en `/usr/lib/news/bin/`. Si ejecutamos este fichero, comprobará los permisos y la sintaxis de todos los ficheros de configuración de los que hemos hablado. Es una herramienta muy útil, ya que protesta por la mínima pijada. Cuando deje de dar errores, tendremos casi listo el sistema.

En versiones anteriores de `inn`, una vez que `inncheck` daba todo como correcto, el sistema ya estaba listo para funcionar; lamentablemente, ahora las cosas han cambiado, y todavía hay que seguir haciendo ajustes.

6.9 Creación de los ficheros db.

Ahora hay que configurar la base de datos del servidor de news, es decir, los ficheros que controlan los artículos que se encuentran en el servidor, los grupos que se alojan...

Es un paso delicado, así que te aconsejo que leas este apartado hasta el final antes de tocar nada. Yo me puse a hacer cambios, y casi jodo todo. De hecho, la configuración de esta parte fue lo que más me costó con diferencia.

Insistir, como insiste la documentación de `inn` en este punto, que debemos trabajar como usuario `news`, para que los permisos queden correctamente establecidos.

6.9.1 Configurando los ficheros db por primera vez.

Si estás configurando `inn2` desde cero, es decir, si no estás actualizando desde una versión anterior de `inn`, necesitarás los ficheros `active`, y `newsgroups`, que se pueden bajar de <ftp://ftp.isc.org/pub/usenet/CONFIG>. También puedes bajar los que utilizo yo, de mi página, que ocupan bastante menos: <http://www.escomposlinux.org/hsolo/archivos/inn+suck/active>, <http://www.escomposlinux.org/hsolo/archivos/inn+suck/newsgroups>

Descarga los ficheros `active`, y `newsgroups` y cópialos en el directorio de los archivos `db`, que viene definido en `inn.conf` por la variable `pathdb`; por defecto `/var/lib/news`.

Ahora, hay que generar el fichero `history`; esto se hace con el comando⁴

```
makehistory -i
```

Cuando acabe, hay que renombrar los ficheros `history.n*` a `history*`. Para acabar con la configuración de `inn`, hay que cambiar los permisos de los ficheros que se acaban de crear:

```
chmod 0664 *
```

Si todo ha ido bien, ya tenemos acabada la configuración de `inn`.

6.9.2 Configuración a partir de una versión anterior de inn.

Si partimos de una versión anterior de `inn`, ya tendremos un fichero `history` y los archivos `active` y `newsgroups`. Estos ficheros valen para la nueva configuración, pero hay que incluirles ciertas modificaciones. De esta forma, nos evitamos tener que descargarlos, y mantenemos nuestra configuración antigua. Para poder usar los archivos existentes, hay que añadir la línea

⁴el comando `makehistory` se encuentra en `/usr/lib/news/bin/`. Si añades el directorio al `PATH` del usuario `news` el trabajo es más cómodo

```
control.cancel 0000000000 0000000000 n
```

en `active`, y la línea

```
control.cancel          Pseudo-group where news system directives are filed.
```

en `newsgroups`. Para regenerar el fichero `history`

```
makehistory -h history -t H
```

Esto debería regenerar el `history` con el nuevo formato⁵

Al igual que antes, hay que renombrar los ficheros `history.n*` a `history*` y cambiar los permisos:

```
chmod 0664 *
```

De nuevo, si todo está correcto, podemos lanzar el demonio `inn2` con⁶:

```
/etc/init.d/inn2 start
```

6.10 Si la cosa no funciona...

Pues si la cosa no funciona, mal asunto. Pueden pasar dos cosas; que yo me haya equivocado (posible, pero improbable, ya que yo tengo el sistema funcionando en casa), o que tu te hayas equivocado, así que empieza a leer desde el principio, y ten más suerte la próxima vez.

Bromas aparte, `inn` es completamente opaco en lo que a mensajes de error se refiere. Todas las salidas de `inn` se almacenan en `/var/log/news/`, pero cuando uno está haciendo pruebas, es más cómodo seguir los mensajes de error en el `syslog`:

```
tail -f /var/log/syslog
```

Lo poco que escape `inn` se puede ver de esta manera.

6.11 ...y si funciona

Pues si funciona enhorabuena, ya tienes la parte de `inn` configurada correctamente, lo más difícil ya está hecho. Para comprobar que la cosa funciona, puedes hacer un `telnet` a nuestra máquina:

```
$ telnet localhost 119
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
200 alderaan.maptel.es InterNetNews server INN 2.2.2 13-Dec-1999 ready
quit
205 .
Connection closed by foreign host.
```

⁵Lamentablemente, yo lo hice mal la primera vez, y borré accidentalmente el fichero `history`, así que esto que escribo es un acto de fe en la página `man` de `history`. Para evitar lo que me pasó a mí, os aconsejo que hagáis copias de todos los ficheros antes de regenerar el `history`, especialmente este último. Si todo falla, lo peor que puede pasar es que tengas que regenerar desde cero, como se describe en 6.9.1 (Configurando los ficheros `<tt>db</tt>` por primera vez)

⁶En Debian, en otras distribuciones esto pude cambiar un poco.

6.12 Añadiendo grupos.

Si todo funciona correctamente, ya podremos añadir grupos a nuestro servidor. Esto se hace con el programa `ctlinnd`. Por ejemplo, para añadir un grupo local:

```
$ ctlinnd newgroup local.pruebas
Ok
```

O uno de nuestros preferidos:

```
$ ctlinnd newgroup es.comp.os.linux.misc
Ok
```

Y para borrar los grupos:

```
$ ctlinnd rmgroup local.pruebas
Ok
```

Os recomiendo la lectura detenida de la página `man` de `ctlinnd` (8), en la que explica el mantenimiento de grupos.

7 Configuración de suck.

Como dijimos antes, `suck` es el encargado de intercambiar artículos entre nuestro servidor local y los servidores remotos.

El sistema normal de configuración de `suck`, coloca los ficheros de configuración en `/etc/suck`. Para llamar a `suck`, hay un script (como siempre, hablo de mi Debian), `/usr/sbin/get-news`. Para bajar artículos de varios servidores, lo que yo he hecho es modificar este script, de forma que acepte un parámetro extra, `-dirconf`, que especifica los distintos ficheros de configuración para cada servidor. La nueva versión del script se llama `/usr/sbin/baja-news`. Evidentemente, los ficheros ya no estarán directamente en `/etc/suck`. Yo los he colocado en subdirectorios bajo `/etc/suck`:

```
$ ls -l /etc/suck/
total 10
drwxrwsr-x  2 news  news      1024 mar 29 16:02 antakira
drwxrwsr-x  2 news  news      1024 mar 29 16:01 cis-dfn
lrwxrwxrwx  1 news  news        21 dic 31 03:10 get-news.conf -> cis-dfn/get-news.conf
-rw-r--r--  1 news  news       818 feb 18  2000 get-news.conf.dpkg-dist
-rw-r--r--  1 hsolo  hsolo       618 mar 28 00:43 personal.killfile
lrwxrwxrwx  1 news  news        30 sep  6  2000 suckkillfile -> /home/hsolo/web/suckkillfile
-rw-r--r--  1 news  news       405 dic 31 16:01 sucknewsrsc
```

Como podéis ver, yo tengo varios subdirectorios, que corresponden a los distintos servidores: `cis-dfn` para `news.cis.dfn.de`, y `antakira` para `news.antakira.com`. Los enlaces, que apuntan a `cis-dfn` los tengo para cuando se llama a `get-news`, para que `news.cis.dfn.de` sea el servidor por defecto.

Y en cada directorio de configuración:

```
$ ls -l /etc/suck/cis-dfn/
total 5
-rw-r--r--  1 news  news      862 dic 15 12:38 get-news.conf
lrwxrwxrwx  1 root  news      20 ene  6 23:14 personal.killfile -> ..
/personal.killfile
lrwxrwxrwx  1 root  news      15 ene  6 23:13 suckkillfile -> ../suck
killfile
-rw-r--r--  1 root  root      404 mar 29 15:59 sucknewsrsc
-rw-r--r--  1 root  root      404 mar 29 04:05 sucknewsrsc.old
```

7.1 get-news.conf.

El fichero de configuración principal de suck es `get-news.conf`. Describo las opciones:

```
server: alderaan
```

El nombre del servidor local (el nuestro, vaya)

```
remoteserver: news.cis.dfn.de
```

El nombre del servidor remoto, evidentemente.

```
outgoingfile: cis-dfn
```

El nombre, es el que hayamos decidido para el *feed* en `/etc/news/inn.conf`. Es la única precaución que hay que tener. Esto es válido teniendo en cuenta que no hemos cambiado el fichero de salida en las opciones del *feed*, añadiendo un *flag* `-Tf` e indicando otro nombre alternativo.

```
sedcmd:/^NNTP-Posting-Host:|^Xref:|^X-Trace:|^X-Complaints-To:|^NNTP-Posting-Date:/d
```

Esto es un punto delicado, haz un acto de fe de momento. Lo explicaré más adelante en 7.6 (Bugs, apaños y otras trampas...)

```
userid: tu_usuario
```

Pues eso, el nombre del usuario remoto.

```
password: tu_clave
```

La clave en el servidor, evidentemente.

7.2 sucknewsrsc.

El fichero `sucknewsrsc` describe qué grupos nos bajaremos de cada servidor. Si estamos bajando distintos grupos de distintos servidores, lo suyo es que sean complementarios, para no bajar los mismos artículos dos (o más) veces. Sin embargo, tampoco hay nada que nos impida hacerlo. Si bajamos noticias de servidores que pierden muchos mensajes, puede ser incluso una buena política, aunque yo creo que mejor política es cambiar de servidor. En cualquier caso, es un gasto de tiempo, ya que primero `suck` baja todas las noticias pendientes de cada servidor, y luego es `inn` quien las rechaza por repetidas. Os pego el `sucknewsrsc` para `news.cis.dfn.de`, y os lo explico:

```

control 901240
junk 0
test -100
to 1000708
#es.comp.os.linux 1078346
es.comp.lenguajes.java 17802
es.comp.seguridad.pgp 2957
#es.comp.hackers 1134047
es.comp.os.linux.misc 34503
es.comp.os.linux.instalacion 24050
es.comp.os.linux.programacion 5747
es.comp.os.linux.redes 13307
es.ciencia.matematicas 14540
es.pruebas 90123
#es.news.admin 1005265

```

Cada línea indica el nombre de un grupo que hay que bajarse. Las cuatro primeras, hay que ponerlas para todos los servidores, esto es, en cada `sucknewsrsc` que tengamos. El número que va detrás de cada grupo, es actualizado por `suck` después de cada descarga, y es el número del último artículo bajado de cada servidor para cada grupo, para que la próxima vez que se conecte baje desde ese número. Para bajar artículos por primera vez, puedes ponerlos a “0”, con lo que se bajará todos los artículos que haya en el *spool* del servidor remoto. Si pones un número negativo, bajará los *n* últimos artículos; por ejemplo, si pones -100, bajará los últimos 100 artículos.

Las líneas que comienzan por “#” se consideran comentarios, y no son tenidas en cuenta.

7.3 Filtros.

El fichero de filtros por defecto es `/etc/suck/suckkillfile`. Como estamos cambiando los directorios de configuración, la ubicación de los filtros también cambia. Así, si usamos como directorio de configuración `/etc/suck/cis-dfn/`, `suck` buscará los ficheros de filtros allí. Yo lo que hago es tener los filtros en `/etc/suck/killfile`, y mantengo enlaces a él desde los distintos directorios de configuración:

```

$ ls -l /etc/suck/cis-dfn/*killfile
lrwxrwxrwx  1 root  news          20 ene  6 23:14 /etc/suck/cis-dfn/personal.killfile -> ../personal.killfile
lrwxrwxrwx  1 root  news          15 ene  6 23:13 /etc/suck/cis-dfn/suckkillfile -> ../suckkillfile
$ ls -l /etc/suck/antakira/*killfile
lrwxrwxrwx  1 root  news          20 dic 30 17:10 /etc/suck/antakira/personal.killfile -> ../personal.killfile
lrwxrwxrwx  1 root  news          15 dic 30 17:11 /etc/suck/antakira/suckkillfile -> ../suckkillfile

```

Como podéis ver, los ficheros de filtros en los dos directorios apuntan al mismo fichero del directorio superior.

7.4 Configuración del segundo servidor (o los siguientes).

Pues no tienen ningún misterio, basta con seguir los pasos anteriores, escribiendo los ficheros de configuración en un directorio aparte, bajo `/etc/suck/`, y ya está.

7.5 baja-news, el meollo de todo esto.

El script `/usr/sbin/baja-news/` es el que se encarga de buscar en los distintos directorios de configuración, y se conecta a los distintos servidores. Es una versión modificada de `/usr/sbin/get-news`, que viene con la Debian, y que es parte del paquete `suck`. La modificación fundamental, es que ahora acepta el parámetro `-dirconf directorio`, que indica a `suck` el directorio en el que debe buscar los ficheros de configuración, los filtros... Te puedes bajar este script de <http://www.escomposlinux.org/hsolo/archivos/baja-news>

El uso es sencillo:

```
$ /usr/sbin/bajanews -dirconf cis-dfn
```

Esto hará que `suck` busque la configuración en `/etc/suck/cis-dfn`

7.6 Bugs, apaños y otras trampas...

Aparte de las modificaciones mencionadas en `baja-news`, he introducido otras, para corregir un pequeño *bug* de ese fichero⁷. Este bug consiste en lo siguiente: muchos servidores que requieren identificación (*news.cis.dfn.de* es uno de ellos, y *news.antakira.com* es otro), necesitan que el cliente se identifique nada más conectarse, en vez de esperar a la petición de autorización (sacado de `man 1 suck`). Esto se consigue pasando a `suck` y a `rpost` (el programa que se usa para enviar artículos a un servidor remoto, y que es llamado automáticamente por `get-news` o `baja-news`), el parámetro `-u`. Por defecto, al menos como viene `get-news` con la Debian, no se pasa ese parámetro ni en las llamadas a `suck` ni a `rpost`. Yo lo he metido en los sitios que llaman a estos programas. Esto no debería tener efecto en los servidores que no necesitan autenticación, o los que no son tan estrictos en ese sentido. Desgraciadamente, no lo he probado, así que si tienes problemas en ese sentido, escíbeme para que adapte el script.

En 7.1 (`get-news.conf`), nos apareció la siguiente línea de configuración:

```
sedcmd: /^NNTP-Posting-Host:|^Xref:|^X-Trace:|^X-Complaints-To:|^NNTP-Posting-Date:/d
```

Tanto *news.cis.dfn.de* como *news.antakira.com*, no aceptan todas la cabeceras que coloca nuestro `inn` local, y rechazan los mensajes que contienen esas cabeceras. Lo que hace la variable de arriba, es eliminar precisamente esas cabeceras. Del manual de `get-news.conf`:

```
sedcmd Indica el comando que empleará put-news para filtrar los mensajes antes de ser enviados al servidor remoto. Todo el artículo pasará a través del editor de flujo "sed" con este argumento. Por ejemplo, si se desea eliminar los campos NNTP-Posting-Host y Xref de la cabecera del artículo debería usarse el valor sedcmd: /^NNTP-Posting-Host:|^Xref:/d
```

Estos valores pueden cambiar para cada servidor de news, por lo que es posible que los tengas que ajustar a mano. Yo he puesto los que son correctos para los servidores que yo utilizo. Puedes probar con el valor de `sedcmd` que viene por defecto e ir ajustando hasta que funcione:

```
sedcmd: /^NNTP-Posting-Host:|^Xref:/d
```

⁷Tengo que agradecer desde aquí la colaboración que me prestó Andrés Herrera aherrerm@escomposlinux.org para descubrir y solucionar este *bug*, que me aguantó pacientemente durante casi dos horas una noche en el IRC

Esto elimina las cabeceras `NNTP-Posting-Host:` y `Xref:`. De hecho, esta configuración era suficiente en versiones anteriores de `inn`. Os pego un mensaje de error cuando el servidor remoto rechaza un mensaje por una cabecera:

```

Connected to news.cis.dfn.de
200 news.fu-berlin.de welcomes 212-7-50-134.wolbiz.worldonline.es (212.7.50.134)
. Authorization required for reading and posting.
340 Ok, recommended ID <99ct6g$oc5j$1@ID-63837.news.dfncis.de>
441 Can't set system "X-Trace" header
      ~~~~~
340 Ok, recommended ID <99ct6h$oc5j$2@ID-63837.news.dfncis.de>
240 Article posted <87wv9ij6mp.fsf@alderaan.maptel.es>
Closing connection to news.cis.dfn.de
Error remote posting
You can hang up the modem now

```

Por último, contar un pequeño problema que tuve. En algún momento de la configuración hice cambios sobre el lugar del *spool* de artículos, cambiando entre `/var/spool/news/articles/`, y `/var/spool/news`. A resultas de esto, los artículos se guardan en `/var/spool/news`, pero `suck`, a la hora de buscarlos para enviarlos al servidor remoto, trata de encontrarlos en `/var/spool/news/articles/`. Supongo que se podría resolver revisando la configuración. De hecho, parece que en la nueva versión de `inn` los artículos se guardan en `/var/spool/news/articles/` por defecto. Puedes dejarlo así, con losa configuraciones por defecto, que deberían funcionar correctamente, o hacer un pequeño apaño, que es convertir el directorio `articles` en un enlace simbólico:

```

$ cd /var/spool/news/
$ln -s . articles

```

De esta forma, el directorio `/var/spool/news/articles/` “se convierte” en `/var/spool/news`. Esto es un apaño como otro cualquiera. No es de lo más elegante, pero funciona (y no es con mucho de lo peor que tengo en mi sistema). Si consigues un sistema más depurado, escíbeme y lo añado para futuras versiones de este documento.

8 Feedback.

Pues lo de siempre. He escrito esto basándome en mi configuración y en mi experiencia, y aunque os aseguro que a mí me funciona, puedo haber cometido errores, tanto de procedimiento como de redacción. Por eso, cualquier comentario, sugerencia, crítica, etc será bienvenida.

Un área importante que estaría bien incluir para futuras versiones, es la configuración de todo esto con `newsx` en lugar de hacerlo con `suck`. Así, que si alguien se anima y escribe algo, podemos incluirlo más adelante.

La última versión de este documento, la puedes bajar de mi página personal: <http://www.escomposlinux.org/hsolo>, y todos los ficheros de configuración de <http://www.escomposlinux.org/hsolo/archivos/inn+suck>