



RDSI con Linux

Historia Personal:

- [Vida y milagros](#)
- [Curriculum Vitae](#)

Temas de Interés:

- [Linux](#)
- [Radioafición](#)
- [Emacs](#)

[<<- Volver](#)

Últimas modificaciones

- 2000/2/27 Santiago González nos facilita el fichero de opciones de pppd y la cadena de inicialización que el usa con RedHat 6.0 para [poder usar el Novacom Mix](#).
- 2000/2/15 Un nuevo paquete para configurar RDSI facilmente desde KDE. Se llama KISDN y se puede encontrar en <http://www.millenniumx.de/kisdn.html>. Parece ser que hay algunas prestaciones adicionales si les da \$20 pero aún así funciona bien sin ellas.
- 2000/1/23 El famoso [teléfono Novacom Mix](#) ya se puede usar con Linux sin problemas.
- 2000/1/16 Añadido un pequeño (pero imprescindible) [comentario](#) sobre la necesidad de eliminar una línea de los ficheros de configuración generados automáticamente por `isdnconfig` para que la instalación sea operativa.
- 1999/12/24 Los ficheros de configuración necesarios para usar RDSI con [ReHat 6.1](#) por fin disponibles.
- 1999/11/10 Las tarjetas con el chipset [Winbond W6692](#) (como la Asuscom ISDNLink 128 PCI) funcionan perfectamente.

Tabla de Contenidos:

- [Introducción.](#)
- [Copyright y esas cosas...](#)
- [Y ahora... ¡en marcha!](#)
- [Nuevo hardware soportado](#)
- [Configuración del hardware](#)
- [El método de emulación de módem analógico \(PPP asíncrono\)](#)
- [Infovia+, o como marear a los usuarios sin razon aparente \(PPP síncrono\)](#)
- [Comentarios de algunos usuarios de RDSI e Infovia+.](#)
- [Referencias y agradecimientos \("The Hall of Fame"\)](#)
- [Despedida y cierre.](#)

Introducción.

Esta página pretende dar algunos detalles de como utilizar ciertas tarjetas RDSI con el sistema operativo Linux. En concreto se centrará en aquellas tarjetas que estén soportadas con el driver HiSax, del paquete [isdn4linux](#). Los ejemplos concretos usados a lo largo del documento harán referencia a una tarjeta Elsa QuickStep 1000 y a un sistema Debian GNU/Linux 2.2, ya que ésta es la configuración que posee el autor. Sin embargo el uso de cualquier otra tarjeta de las gestionadas con el driver HiSax o de otra distribución debería plantear cambios mínimos en los ejemplos.

Se ha añadido una nueva sección al documento que indica como conectarse a Infovia+ con Linux y RDSI, ya que a Telefónica le ha dado por cambiar el protocolo de conexión de bajo nivel, sin consultar a nadie. Bueno, en realidad simplemente ha eliminado el antiguo sistema de conexión, ya que en la Infovia tradicional funcionaban tanto el método antiguo como el nuevo, necesario ahora por imperativo legal.

No obstante me gustaría poner en claro ciertos detalles legales para evitarme posibles quebraderos de cabeza.

Copyright y esas cosas

Se ha cambiado la licencia de uso del contenido de esta pagina. La nueva licencia está mucho mejor adaptada a los contenidos de tipo "literario" que la GPL, que es una licencia diseñada para ser aplicada principalmente al software. Por lo tanto, lea con atención los próximos párrafos y la licencia (que podrá encontrar en la URL que se indica enseguida) para saber cuáles son sus derechos (casi todos) y sus obligaciones (muy pocas ;-).

Este documento es Copyright ©1998-2000 Ignacio Arenaza Nuño y es OpenContent (Contenido Abierto). Usted puede redistribuirlo y/o modificarlo bajo los términos de la Licencia OpenContent (OPL) versión 1.0, tal y como fue publicada por la OpenContent Organization. Este documento se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTIA; sin ni siquiera la garantía implícita de COMERCIALIZABILIDAD o CONVENIENCIA PARA UN PROPOSITO PARTICULAR. Vea la Licencia OpenContent para más detalles. Existe una versión disponible en <http://www.opencontent.org/opl.shtml>.

El copyright no es para restringir los derechos a nadie, es para garantizar que todo el mundo pueda usarlo y que de paso no me intenten colgar algún muerto si a alguien le falla algo al intentar hacer lo que aquí indico. Como se suele decir en estos casos, a mi me funciona y su caso puede variar.

Y ahora... ¡en marcha!

Una vez que los abogados están contentos, pasemos a detalles más jugosos. Los pasos que se indican aquí sirven para usar la tarjeta RDSI de dos formas diferentes:

1. Emulando un módem analógico tradicional, de forma que podamos usar el demonio pppd habitual y los scripts de conexión tradicionales.
2. Emulando una interfaz de red, para lo cual hay que usar las `isdnutils` y el demonio `ippdd` (observe la diferencia en el nombre) que permite usar PPP **síncrono**. Este modo de funcionamiento es necesario para usar Infovia+ con Linux, y los scripts de conexión son bastante diferentes.

Aun cuando la segunda forma de trabajar está descrita en el RDSI-COMO del SLUG, hay

algunos usuarios que están teniendo problemas para ponerla en marcha. Voy a tratar de plasmar en esta página como he realizado yo la configuración en mi instalación de Debian 2.2 con un núcleo 2.2.19.

Nuevo hardware soportado.

Varias personas me han preguntado si algunos modelos concretos de tarjetas RDSI de última generación funcionan con Linux (concretamente algunos modelos PCI). Como sólo puedo probar aquellas tarjetas que caen en mis manos, por el momento he podido probar y confirmar los siguientes modelos:

- **Asuscom ISDNLink 128 PCI:** Es una de las tarjetas que incorporarán el nuevo chip RDSI de Winbond, el W6692. He probado con un núcleo 2.2.13 y el soporte RDSI que viene de serie, y la detecta y maneja sin ningún problema. Basta con indicar que es una tarjeta tipo 36 (ver más abajo para saber que significa este número) y listo. Eso sí, el driver está marcado como experimental (al menos en la versión 2.2.13) así que recuerda indicar soporte para características experimentales al compilar el núcleo.

Ojo, que hay otro modelo en el mercado con el mismo nombre pero diferente chipset, concretamente el HFC PCI 2BDS0, con lo cual en caso de duda se puede compilar el soporte para ambos chips a la vez (perfectamente posible) y después probar con cada uno de los dos hasta acertar con el correcto (basta con cambiar el número de chipset en los comandos de abajo y usar 36 o 35 respectivamente en cada caso). Gracias a José A. Barrera por indicarme este detalle.

- **¡Por fin!** Tras mucho dar la tabarra, y gracias a los esfuerzos de Santiago Gonzales Herrero, el fabricante del famoso teléfono Novacom Mix, que tan pobres resultados ofrecía bajo Linux, ya funciona de forma adecuada. He aquí el anuncio del propio Santiago sobre el asunto, realizado en es.comp.os.linux.redes :

Hola a todos.

Por fin se ha solucionado el problema del teléfono RDSI Novacom Mix y Linux. Los que tengan este teléfono y estén interesados en usarlo con Linux, pueden solicitar una actualización del firmware a Antonio Cervantes en la dirección:

tel-rdsi@interisa.es.

Es una versión Beta, pero ya la he probado y el adaptador va correctamente con Linux ahora. Al parecer no la distribuirán con el teléfono hasta que Telefónica no la homologue, pero nos permite "rescatar" ese teléfono.

Ya era hora. Saludos

***** Mensaje original *****
On Fri, 21 Jan 2000 09:50:48 +0100, Antonio Cervantes wrote:

Hola a todos:

Se acaba de desarrollar una versión Beta que resuelve el problema que impedía utilizar el Novacom Mix con Linux. Como es normal en cualquier desarrollo de software, el plan es hacer una prueba de campo para detectar posibles errores. Aquellos usuarios del Novacom Mix y de Linux que estén interesados en participar pueden solicitarme la versión y se la enviaré por e-mail. Por supuesto, siempre pueden volver atrás y telecargar la versión homologada cuando lo deseen. Saludos

Antonio Cervantes
antcerv@iies.es
***** Mensaje original *****

Si tienes alguna tarjeta que no te funciona (o que el núcleo no detecta) y no te importa enviar unos cuantos mensajes de correo, ponte en contacto conmigo y veremos que se puede hacer.

Configuración del hardware.

Como paso previo a cualquiera de los dos modos de funcionamiento, es requisito indispensable que el núcleo reconozca nuestra tarjeta RDSI y nos deje usarla. De esto se encargan varios componentes del núcleo que será necesario compilar. Para ello, recomiendo usar un núcleo de linux igual o posterior al 2.2.18. De esta forma el driver de HiSax más reciente ya viene incorporado en el núcleo y no hay que conseguirlo aparte y parchear el mismo.

Pasos a seguir para la compilación del soporte de la tarjeta en el núcleo

Una vez que dispongamos de las fuentes del núcleo tenemos que realizar las siguientes operaciones:

1. Compilar el núcleo de Linux que hemos obtenido en el punto anterior (puede ver las instrucciones necesarias para compilarlo en `/usr/src/linux/README` una vez que lo haya descomprimido en `/usr/src`). En el núcleo, para referirse a RDSI se utilizan las siglas ISDN (Integrated Services Digital Network). Ya está avisado.
2. A la hora de especificar las características que queremos incluir en el núcleo, debemos seleccionar **por lo menos** las siguientes (los nombres que aparecen en la lista corresponden a las etiquetas mostradas al ejecutar `make config` con un núcleo de la serie 2.2.x, los comentarios del autor aparecen entre paréntesis).
 - `CONFIG_EXPERIMENTAL=y` (Necesario para que nos pregunte todas las opciones)
 - `CONFIG_MODULES=y` (El driver HiSax es obligatorio usarlo como módulo, así que...)

- CONFIG_NET=y (Para que podamos usar servicios de red)
 - CONFIG_SYSVIPIC=y (Necesario para usar kerneld con los módulos)
 - CONFIG_KERNELD=y (Kerneld se encarga de cargar los módulos automáticamente...)
 - CONFIG_INET=y (Soporte para TCP/IP)
 - CONFIG_NETDEVICES=y (Dispositivos de red)
 - CONFIG_PPP=m (Soporte para PPP, yo prefiero cargarlo como módulo, pero puede ser compilado en estático respondiendo "Y" en vez de "M").
 - CONFIG_ISDN=m (Soporte para RDSI en módulo)
 - CONFIG_ISDN_PPP=y (OPCIONAL: compilar si vamos a usar PPP sincrónico, p.ej. con Infovia+)
 - CONFIG_ISDN_DRV_HISAX=m (Soporte para el driver HiSax)
 - CONFIG_HISAX_EURO=y (Importante, usamos protocolo Euro-DSS1 o Euro-ISDN)
 - CONFIG_HISAX_ELSA=y (Soporte para la Elsa QuickStep 1000, elija la opción que corresponda a su tarjeta RDSI)
 - CONFIG_PROC_FS=y (Para diagnósticos e información del sistema en general)
3. Una vez se han indicado las opciones imprescindibles (más el resto de ellas necesarias para el sistema de cada uno), compilamos el núcleo y hacemos los pasos necesarios para poder arrancar con el núcleo recién compilado.
 4. Ahora podemos arrancar con el nuevo núcleo. Hay que reinicializar la máquina para que los cambios surjan efecto. Todo debería ir como antes (si hemos compilado todo lo que teníamos antes **más** el soporte para PPP y RDSI).
 5. Podemos usar el comando `dmesg` para ver los mensajes de arranque del núcleo. No debería haber nada raro (ya que el soporte para RDSI se carga como módulo cuando es necesario y todavía no hemos intentado usarlo).

Pasos necesarios para la configuración de la tarjeta

1. Si la tarjeta no es Plug&Play, será necesario editar el fichero `/etc/lilo.conf` (si usted usa lilo) para indicarle al núcleo los valores de `irq`, `e/s` y otros detalles. Vea los ficheros de `/usr/src/linux/Documentation/isdn/` para los detalles concretos de cada tarjeta.
2. Si la tarjeta RDSI es Plug&Play (caso de la Elsa QuickStep1000) es necesario configurar el hardware antes de poder usarla. Para ello necesitamos la ayuda de dos herramientas: `pnpdump` e `isapnp`. Los pasos a seguir son:

- Ejecutar:

```
pnpdump > /tmp/isapnp.conf
```

Esto creará un fichero con un contenido similar al siguiente (suponiendo que ésta sea la única tarjeta Plug&Play del sistema; si hubiera otras el fichero contendría más líneas). Hay que editar el fichero obtenido eliminando los comentarios de las líneas con la configuración deseada, para obtener algo similar a lo que aparece aquí. Para más detalles, consulte la documentación de `isapnp` y `pnpdump`.

```
# Trying port address 0203
# Board 1 has serial identifier 2a 00 00 00 00 33 01 93
15
#(DEBUG)
(READPORT 0x0203)
(ISOLATE)
(IDENTIFY *)

# Card 1: (serial identifier 2a 00 00 00 00 33 01 93 15)
# ELS0133 Serial No 0 [checksum 2a]
# Version 1.0, Vendor version 0.0
# ANSI string -->ELSA QuickStep 1000<--
#
(CONFIGURE ELS0133/0 (LD 0
(IO 0 (BASE 0x0160)) <--- Indicar aquí la dirección
de E/S.
(INT 0 (IRQ 15 (MODE +E))) <--- Indicar aquí la IRQ a
usar
(ACT Y)
))
# End tag... Checksum 0x00 (OK)
```

- Una vez haya editado el fichero anterior, debe ejecutar:

```
isapnp /tmp/isapnp.conf
```

Esto configurará la tarjeta Plug&Play.

3. Si no ha habido ningún error, ahora puede copiar el fichero `/tmp/isapnp.conf` a `/etc/isapnp.conf`. Si ha instalado correctamente las `pnpools` las próximas veces que arranque el sistema, la configuración de la(s) tarjeta(s) Plug&Play será automática.
4. A la hora de elegir las direcciones de E/S e IRQ a usar, puede ver cuáles están ya ocupadas haciendo:

```
cat /proc/ioports
cat /proc/interrupts
```

5. Ya tenemos la mitad del camino hecho. Tenemos la tarjeta lista para ser usada y

tenemos compilado el soporte para RDSI en el núcleo. Ahora sólo falta unir las dos cosas. Para ello vamos a indicar al núcleo que queremos usar el driver HiSax. Si al cargar el módulo éste detecta la tarjeta ya tenemos la batalla casi ganada. Sino, habrá que repasar los puntos anteriores para ver si hemos compilado el soporte adecuado o si las direcciones de E/S, IRQ u otras están siendo ya usadas por otros dispositivos.

6. Para cargar el módulo de la tarjeta RDSI en el núcleo hacemos:

```
modprobe hisax type=7 io=0x160 irq=15 protocol=2 id=qs1000
```

Para ver cuáles son los números mágicos a usar en el comando anterior tiene que leer el fichero `/usr/src/linux/Documentation/isdn/README.HiSax`. En concreto, en la línea anterior le indicamos que queremos usar una tarjeta Elsa QuickStep 1000 (`type=7`), cuya dirección de E/S es la `0x160` (la misma que en el fichero `/etc/isapnp.conf`), que la interrupción a usar es la 15 (como se indica en `/etc/isapnp.conf`) y que el tipo de protocolo a usar es EURO-DSS1, que es el usado en España. El último valor es sólo un identificativo asignado a esta tarjeta (ya que existe la posibilidad de usar varias simultáneamente).

7. Si nos dice que el driver ha sido cargado y la tarjeta detectada, ¡ya está casi todo listo! (si no apareciera nada en pantalla, puede usar el comando `dmesg` para ver los mensajes del núcleo).
8. Si le parece demasiado pesado tener que teclear todo esto cada vez que quiera usar la tarjeta RDSI, tiene razón. Lo que puede hacer es añadir las siguientes líneas al fichero `/etc/conf.modules`:

```
# 01.02.1999 Ignacio Arenaza. Añada las siguientes líneas
para que
# el módulo de la tarjeta RDSI se cargue automáticamente
bajo demanda.
alias char-major-43 hisax
alias char-major-45 isdn
options hisax type=7 io=0x160 irq=15 protocol=2 id=qs1000
```

Por supuesto debe cambiar los parámetros `type`, `io`, `irq`, `protocol` e `id` para que se ajusten a su configuración.

Además, si quiere que el módulo se cargue en el núcleo de forma automática, deberá compilar el soporte de `kernel` en el núcleo y editar el fichero `/etc/modules` e incluir la palabra `auto` justo después de las líneas de comentarios (en caso de que esté usando Debian GNU/Linux 2.1 o posterior).

El método de emulación de módem analógico (PPP asíncrono)

Para usar la tarjeta RDSI emulando un módem analógico convencional necesitaremos un demonio `pppd` reciente (el que venga con su distribución valdrá seguramente). También será de gran ayuda el documento "eng-i41-faq", disponible en <http://www.isdn4linux.de/>. Hay sendas versiones en alemán e inglés. Yo personalmente prefiero la inglesa ;-)

En este momento ya está disponible la tarjeta RDSI para su uso. Para ello, en los scripts de configuración de `pppd` tenemos que realizar un par de cambios o tres. Estos son:

1. En lugar de usar el dispositivo `/dev/ttyS0`, `/dev/ttyS1`, ... que representan los puertos serie donde suelen estar conectados los módems convencionales, usaremos el dispositivo `/dev/ttyI0`, `/dev/ttyI1`, ...
2. En el script de conexión de chat, es necesario añadir un par de comandos AT para inicializar el valor del MSN/EAZ (vea `eng-i41-faq` para detalles) así como el protocolo de nivel 2 usado para la conexión (cuyo valor por defecto es incorrecto).
3. También es conveniente indicar el tamaño máximo del bloque de transmisión, para no desbordar los buffers de la tarjeta.

A continuación adjunto una copia de mis ficheros `/etc/ppp/peers/provider` (fichero de configuración de `pppd` en Debian GNU/Linux 2.2) y `/etc/chatscripts/provider` (fichero de conexión de chat en Debian GNU/Linux 2.2). En ambos aparecen en negrita aquellas líneas que son diferentes con respecto a una conexión con módem convencional (me he basado en el Infobia-COMO como base para ambos scripts).

`/etc/ppp/peers/provider` (permisos 640, propietario: root, grupo dip)

```
connect "/usr/sbin/chat -v -f
/etc/chatscripts/provider"
passive
user inaki.arenaza@jet
noipdefault
debug
defaultroute
asynmap 00000
mtu 576
mru 576
/dev/ttyI0
115200
:172.16.1.96
ipcp-accept-local
ipcp-accept-remote
```

`/etc/chatscripts/provider` (permisos 640, propietario: root, grupo dip)

Lo que aparece entre paréntesis son comentarios que no deben incluirse en el fichero real.

```
" " "ATS14=3" (usar HDLC como protocolo de nivel 2)
" " "AT&B2000" (Tamaño máximo del bloque de
transmisión)
" " "AT&E900123456" (MSN/EAZ = número de teléfono
propio incluyendo prefijo).
ABORT BUSY
ABORT "NO CARRIER"
ABORT VOICE
ABORT "NO DIALTONE"
" " ATD055
CONNECT " "
```

/etc/ppp/pap-secrets (permisos 600, propietario root, grupo root)

```
inaki.arenaza@jet * mi_contraseña
```

Entre el nombre del usuario, el asterisco y la contraseña debe haber un tabulador (ojo, no usar espacios en blanco).

En este momento debe asegurarse de que tiene creados los ficheros de dispositivo `/dev/ttyI0`, `/dev/ttyI1`, etc. Si no es así, puede crearlos con el comando `MAKEDEV` que se haya en el directorio `/dev`, o a mano con el comando `mknod`. Si usa `MAKEDEV` (al menos la versión de Debian GNU/Linux), basta con que ejecute lo siguiente en el directorio `/dev`:

```
./MAKEDEV isdn-tty
./MAKEDEV isdn-io
```

Si ya tiene creados los dispositivos `/dev/ttyI*`, y tiene instalado `pppd` y `chat`, ya puede probar su conexión a Internet. En concreto, con la configuración de ejemplo de este documento y usando un sistema Debian GNU/Linux, basta con teclear `pon`, y se lanzará la ejecución del demonio `pppd` que establecerá la conexión con su proveedor a través de Infovia. Puede ver un registro de la conexión en `/var/log/ppp.log` (en los sistema Debian GNU/Linux). Si la conexión falla por alguna razón, en ese fichero aparecerá el porqué.

Ahora ya puede usar su navegador favorito o el lector de correo. Para cortar la conexión, basta con ejecutar el comando `poft` (en los sistemas Debian GNU/Linux).

PPP Síncrono.

Todo lo anterior está muy bien, pero ¡¡ya no sirve para Infovia+!!!. Por alguna razón que al común de los mortales se nos escapa, Telefónica ha decidido eliminar de un plumazo el soporte de PPP asíncrono en la nueva Infovia+. Así que la mayoría de los usuarios de RDSI con Linux nos hemos visto forzados a cambiar toda la configuración del subsistema RDSI para poder usar el protocolo estrella de Infovia+: **el PPP síncrono**.

Qué nos hace falta

Para poder usar el PPP síncrono con Linux es necesario:

- Haber compilado el soporte para PPP síncrono en el núcleo como se indica al comienzo de este documento.
- Disponer del paquete `isdnutils`, presente en la mayoría de las distribuciones de Linux. No obstante, si usted ha actualizado su núcleo a una versión posterior a la que venía con su distribución, mi recomendación (obligación en el caso de usar 2.0.36 o posterior) es que se pase por [ftp://www.isdn4linux.de/](http://www.isdn4linux.de/) y que obtenga la última versión disponible del paquete `isdnutils` para el tipo de núcleo que esté usando actualmente. Yo estoy usando la versión `isdn4k-utils-3.1pre1b` con un núcleo 2.4.5 actualmente.

Compilación del paquete `isdn4k-utils`

1. Obtener una copia del fichero `isdn4k-utils-3.XXXX.tar.gz` de la URL citada arriba. Descomprimirlo en `/usr/src` y leer el fichero `README`.
2. Ejecutar `make config`.
3. Aparece un sistema de configuración similar al que aparece cuando configuramos el núcleo con la opción `make menuconfig`. Las **únicas** modificaciones que yo realizo a los valores por defecto son:
 - En la opción "General Configurations", dar el valor `"/usr/sbin"` al campo "Directory to install admin binaries".
4. Guardar la configuración y salir del programa de configuración. Tras unos minutos en los que la utilidad `autoconf` de GNU hace de las suyas, volvemos al prompt del sistema.
5. Ahora ejecutar `make`. Comienza la compilación de todas las utilidades que hayamos configurado. Paciencia, se toma su tiempo.

Instalación y configuración de `isdn4k-utils`

Las operaciones anteriores podían hacerse como un usuario normal. Sin embargo, los pasos a realiza a partir de ahora deben hacerse como usuario `root`.

1. Instalar el paquete `isdnutils` **que viene con mi distribución** (Debian GNU/Linux) para tener todos los ficheros de configuración en su sitio y con el formato que le gusta mi distribución.
2. Ejecutar `make install` para instalar las `isdnutils` que acaba de compilar, desde el directorio donde las he compilado.
3. Ejecutar la utilidad `isdnconfig`. Esta utilidad es específica de Debian GNU/Linux y

sirve para configurar más fácilmente todo el subsistema de RDSI con PPP síncrono. Para ver si existe una utilidad similar en su distribución, consulte la documentación que acompaña al paquete isdnutils que viene con su distribución. Sino, siempre puede cambiarse a Debian ;-)

4. Los pasos a realiza con isdnconfig son los siguientes:

1. Aparecerá la siguiente pantalla, que es la pantalla principal.

```
Isdnutils configuration
=====

1      network devices
2      synchronous ppp daemon
3      modem emulation
4      isdnlog configuration
5      voice box configuration

Q      Quit

Your choice :
```

2. Elegir la opción "1.- network devices". Aparecerá la pantalla:

```
Isdnutils configuration network interfaces
===== -----

Searching for existing configuration
files...
Found : NONE

Devices should be named :
- rawip interface           isdn0
isdnl ...
- synchronous ppp interface ipp0
ipp1 ...

Name for new interface :
```

3. Teclear como nombre de la nueva interfaz "ipp0" y pulsar Enter. Pulsar de nuevo Enter para retornar a la pantalla principal.

4. Ahora elegir la opción "2.- synchronous ppp daemon". Aparecerá la pantalla:

```
Isdnutils configuration synchronous ppp
daemon
===== -----

Searching for existing configuration
files...
Found : NONE

Normal setup: one ipppd for all devices:
/dev/ipp0 /dev/ipp1

                All ipp* network
interfaces are handled by a
                single ipppd process
using a single config
                file.

Special setup: one ipppd for each device:
/dev/ipp0 /dev/ipp1

                Each ipp* network
interface is handled by its
                own ipppd process, using
a specific config file
                for each.
                You need to use pppbind
on each network
                interface to select which
/dev/ipp* device
                will be used.

Usually only a single ipp interface is
used (for connecting to an ISP), so the
above isn't really relevant to most users.

Name for new configuration (ipp0 ipp1
...):
```

5. Teclear como nombre de la nueva configuración "ipp0" (mismo nombre que el de la interfaz del paso 3) y pulsar Enter. Pulsar de nuevo Enter para retornar a la pantalla principal.

6. Si en algún momento nos interesa usar la tarjeta RDSI como un módem analógico convencional, podemos elegir la opción "3.- modem emulation". Se nos informará de que ha instalado el fichero de datos necesario para usar iprofd y deberemos pulsar Enter para retornar a la pantalla principal.

7. Ahora elegir la opción "4.- isdnlog configuration". Aparecerá la pantalla:

```
Isdnutils configuration isdnlog
```

```

===== -----
Installing example configuration files ...
/etc/isdn/isdn.conf already
exists,preserving original version.
callerid.conf installed.

Searching for existing configuration
files...
Found : NONE

You need one options file per isdn card.
This file is named like the device it uses
to get the data. isdnctrl0 isdnctrl2 ... An
exception is if you want to run isdnlog in
"dual mode"; this means thesecond isdn card
is connected "in reverse" so that it can
monitor _outgoing_ traffic. In that case,
only specify one options file

Name of the device:

```

8. Teclear como nombre de la nueva configuración "isdnctrl0" y pulsar Enter. Pulsar de nuevo Enter para retornar a la pantalla principal.

Con esto ya hemos creado el **esqueleto** de los ficheros de configuración. Ahora **tenemos que editarlos** para adaptarlos a nuestras necesidades. Todos estos ficheros están situados en el directorio `/etc/isdn` en el caso de Debian GNU/Linux. Los ficheros que hay que editar son:

- `/etc/isdn/callerid.conf`
- `/etc/isdn/device.ippp0`
- `/etc/isdn/ippdpd.ipp0`
- `/etc/isdn/isdn.conf`
- `/etc/isdn/isdnlog.isdnctrl0`

Le recomiendo que lea con atención los comentarios de los ficheros para saber para que sirve cada uno de ellos. Un detalle **muy importante** a la hora de editar estos ficheros es que hay que **eliminar** (no vale con que sean comentarios) las líneas que dicen:

```

# REMOVE the next line once configuration is complete
#####
echo "Warning! $0 not configured yet! Aborting..."; exit
1
# REMOVE the above line once configuration is complete
#####

```

de todos los ficheros en los que aparezcan, ya que de lo contrario las funciones de configuración del sistema RDSI suponen que sus ficheros siguen sin configurar y no activan ni la interfaz RDSI ni el resto de los servicios RDSI.

Para facilitar su correcta configuración (de forma básica), a continuación tiene una copia de los ficheros tal y como están ahora mismo en mi sistema:

- [callerid.conf](#)
- [device.ippp0](#)
- [ippdpd.ipp0](#)
- [isdn.conf](#)
- [isdnlog.isdnctrl0](#)

No use el fichero `device.ippp0` tal cual. Debe editarlo y cambiar los valores de LOCALMSN por su número de teléfono propio y REMOTEMSN por el número de teléfono del nodo de Infovia+ que use.

Si ha observado el fichero `/etc/isdn/device.ippp0` que yo proporciono, se han comentado las líneas que hacen referencia al establecimiento de las rutas por defecto en la interfaz `ipp0`. En mi caso esto es así porque tengo otras necesidades de enrutado. Si usted sólo tiene la interfaz de red que corresponde a la tarjeta RDSI, entonces descomente las líneas que comienzan por "route add ..." y "route del ...", situadas hacia el final del fichero `/etc/isdn/device.ippp0`.

Por último, necesita el fichero de inicio de todas las utilidades para que todo quede correctamente configurado en cada arranque del sistema. El fichero que se encarga de todo esto en Debian GNU/Linux es `/etc/init.d/isdnutils`. Aquí tiene una copia del fichero tal y como se encuentra en mi sistema ahora mismo: [/etc/isdn/isdnutils](#). Lo único que yo he modificado en este fichero es añadir una línea para cargar el módulo del driver HiSax en el núcleo antes de arrancar todo el proceso de configuración, para evitar la aparición de algunos mensajes de error.

Puesta en marcha (por fin ;-)

Ya tenemos todo listo. Ahora llega el momento de comprobar que todo este montaje funciona.

Para activar la conexión debemos ejecutar el comando:

```

/usr/sbin/isdnctrl dial ipp0

```

Esto hará que la tarjeta marque el número indicado en REMOTEMSN (en el fichero `/etc/isdn/device.ippp0`) y que comience todo el proceso de conexión y negociación del protocolo.

Yo personalmente no tecleo este comando cada vez que quiero conectar. Me he creado un pequeño script al que llamo `ipon` (puesto que Debian GNU/Linux incluye un script llamado `pon` usado para conectar con módems analógicos con `pppd`, yo llamo al mío `ipon` porque usa `ippd`). El script está situado en `/usr/bin` y sus permisos son:

```
-rwxr-xr-x 1 root root 41 Jan 24 10:58 /usr/bin/ipon
```

Su contenido es:

```
#!/bin/sh
/usr/sbin/isdnctrl dial ipp0
/usr/bin/tail -f /var/log/syslog
```

La segunda línea es opcional, y sirve para visualizar el registro de eventos del sistema, para ver si todo va correctamente.

Ahora puede ejecutar el script y ver si puede hacer unos cuantos pings a algunas maquinas de Internet. Si todo ha ido correctamente, ya puede usted navegar (o lo que más le guste) por Internet. ¡Enhorabuena!

Sino, una atenta revisión de los ficheros de configuración junto con los mensajes de error que aparezcan en el registro del sistema deberían dar con el error en poco tiempo.

Por ultimo, para indicarle al sistema que deseamos cortar la conexión, basta con teclear:

```
/usr/sbin/isdnctrl hangup ipp0
```

De nuevo, yo he creado un script similar al anterior, al que llamo `ipoff`, situado en el mismo directorio y con los mismos permisos. Su contenido es:

```
#!/bin/sh
/usr/sbin/isdnctrl hangup ipp0
```

Comentarios de algunos usuarios de RDSI e Infovia+

He aquí los comentarios enviados por algunos usuarios de RDSI con Infovia+. Si cree que falta algo en esta página, o si desea comentar su experiencia personal con el tema, no dude en enviarme un mensaje con sus comentarios, y por favor, concédame el permiso para publicarlos aquí también, de forma que puedan beneficiar al resto de la comunidad. Muchas gracias.

- Alberto Gonzalez Iniesta escribía lo siguiente en un mensaje de correo:

```
Hola Iñaki, la verdad es que al final conseguí ponerlo a
andar. Encontre esta web: http://sites.inka.de/picard/rh52\_isdn.html

Que como puedes ver es sobre RedHat y la RDSI. Conseguí echarlo a
andar con sus scripts y ahora estoy a ver si les echo una mano con la
versión española. :-)
```

Y es que no veas lo que he sudado, la de scripts que he probado, y la de folios que he leído hasta que la maldita I+ ha querido hablar con mi Linux.

- Josep Lladonosa i Capell escribía lo siguiente en un mensaje de correo:

```
Hola,
```

he leído tu página de configuración de RDSI y linux debian, y ha sido una lástima leerla tan tarde, ya que me tuve que espabilar para conseguir la conexión. El resultado final, el mismo.

Quería comentarte algunas cuestiones interesantes que podrías añadir a tu página.

A mí, el fichero `/etc/isdn/ippd.ipp0` que mejor resultado me está dando es el siguiente:

```
-----
# /etc/ppp/options.ippd for ippd
# comment out the following three lines if you don't need
# dynamic IP negotiation.
ipcp-accept-local
ipcp-accept-remote
noipdefault
#defaultroute      # la ruta por defecto se establece en
                   # /etc/ppp/ip-up.d/isdnutils

lcp-restart 1
pap-timeout 9
pap-restart 1
pap-max-authreq 3
-ipx
-ccp
-vjccomp
-ac
-pc
-detach
-bsdcomp
-vj
lock
user minombre@miproveedor
-----fin de fichero
```

Los tiempos de espera para pap los añado porque gracias a una leve modificación del ippdp de las isdnutils hago que, en caso de fallar la autenticación, se corte la conexión.

Con esto evito sobrepasar los 15 segundos a partir de los cuales telefónica empieza a cobrar (ahorrador que es uno ;-)..

Completo el procedimiento auth_withpeer_fail() de ippdp/auth.c con las siguientes líneas:

```
void auth_withpeer_fail(int unit,int protocol)
{
    /*
     * We've failed to authenticate ourselves to our peer.
     * He'll probably take the link down, and there's not
     * much we can do except wait for that.
     */
    lcp_close(lns[unit].lcp_unit,"auth failed to peer"); /* by JLC */
    lns[unit].phase = PHASE_TERMINATE; /* hangup, otherwise we pay.
                                         by JLC
                                         */
}
```

Después compilo y ya está, mi ippdp cuelga si infobia+ no me autentifica. Espero que esto sirva a la comunidad.

Supongo que esto se puede preparar en formato diff, pero no lo he hecho nunca. Además, envié la propuesta a los del isdn4linux y ni caso (supongo que es porque lo que hago 'viola' el ppp).

Ah, yo uso una Teles 16.3, a través de centralita y 'no problem'. Además el hisax permite utilizar el PC como contestador automático.

Referencias y agradecimientos ("The Hall of Fame")

Este documento no habría sido posible sin la colaboración, directa o indirecta, de un montón de personas. Es muy posible, como ocurre en este tipo de cosas, que me deje a mucha gente en el tintero. Si alguno cree que falta, ya sabe donde encontrarme.

Los "culpables" por tanto son (el orden de aparición no indica ningún tipo de importancia en el asunto ;-):

- [Javier Cantero](#). El infatigable investigador de Linux.
- Alberto Gonzalez Iniesta, por sus comentarios sobre la página y su granito de arena aportado a esta página.
- Josep Lladonosa i Capell, por su mensaje de correo electrónico lleno de consejos útiles para mejorar el rendimiento del enlace con ippdp y el parche para ahorrar en la factura telefónica, que siempre viene bien ;-)
- La gente del grupo [es.comp.os.linux.redes](#). Cada uno colaborando con su granito de arena, hacéis que merezca la pena el esfuerzo.
- La gente del Infobia-COMO, por darme las pistas iniciales sobre como funcionaba todo esto.
- La gente del RDSI-COMO, por hacer el trabajo sucio de leerse toneladas de manuales en vaya usted a saber que idiomas y redactar un documento que nos sirva a todos. Y por mencionarme en él, por supuesto ;-)
- La gente del isdn4linux, porque sin ellos todo esto no tendría sentido.
- Y por supuesto a Telefónica, por proporcionarnos estos retos intelectuales tan estimulantes. Gracias a ellos he aprendido que es LCP, IPCP, y otros muchos términos similares. Sin su inestimable ayuda yo sería un tipo gris que usaría Infovia/Internet sin enterarme de lo que pasa realmente. Así que, gracias muchachos ;-)

Despedida y cierre

¡Eso es todo! Disfrute de Infovia+ (si se puede hacer tal cosa) con su estupendo sistema operativo... Linux, por supuesto.