



EL KONSULTORIO DE KLAUS




Klaus Knopper es el creador de Knoppix y co-fundador de la LinuxTag Expo. En la actualidad trabaja como profesor, programador y consultor. Si tiene algún problema de configuración, o simplemente quiere conocer mejor cómo funciona Linux, no dude en escribir sus preguntas a: preguntas@linux-magazine.es

WiFi

? Gracias por su respuesta a mi correo electrónico en el que pedía ayuda con el WiFi de mi portátil. Estoy usando Ubuntu 8.04.1. `/etc/wpa_supplicant` no contiene `wpa_supplicant.conf`, sin embargo `/etc/network/interfaces` contiene líneas que parecen correctas. Y `sudo wpa_cli status` también parece correcta. No tengo ni idea de cómo hacer la prueba sería de verdad, que sería verificar si la petición DHCP llega demasiado

pronto o es que ni siquiera llega. Utilizo `System | Administration | Network` para introducir los caracteres correctos y la contraseña, pero parece que algo los resetea a los datos por defecto.

En los ordenadores modernos a veces se cambian cosas sin que uno siquiera lo sepa. La contraseña WPA es la única cosa que me ha causado serios problemas, sin contar que he reinstalado Ubuntu 8.04.1 esta semana y sólo puedo conectarme a Internet por cable, no por WiFi. ¡Sólo espero estar aprendiendo cosas casi tan rápido como van cambiando!

 De hecho, muchas cosas diseñadas para facilitar la configuración al usuario tienden a hacer el sistema más complejo e introducir más posibilidades de error.

No estoy seguro de qué es lo que hace exactamente la interfaz gráfica de configuración de red de su escritorio en `System | Administration | Network`, pero generalmente en sistemas tipo Debian, simplemente administra las entradas de `/etc/network/interfaces`, en lugar de mantener sus propios archivos de configuración. Su configuración WiFi podría estar correcta, incluso sin un archivo `wpa_supplicant.conf`. `wpa_cli` podría cambiar los parámetros del WiFi/WPA de forma dinámica.

Las versiones de Debian y Ubuntu más recientes soportan una sintaxis extendida en `/etc/network/interfaces` que tiene un aspecto como este:

```
auto wifi0
iface wifi0 inet dhcp
wpa-driver wext
wpa-ssid <your_essid>
wpa-ap-scan 1
wpa-key-mgmt WPA-PSK
wpa-psk <your_wpa_key>
```

El comando

```
ifup wifi0
```

realiza la llamada a `wpa_cli` con los parámetros de `/etc/network/interfaces` para cambiar la configuración en uso de `wpa_supplicant` para que coincida con nuestra configuración WPA. Nótese que `wpa_supplicant` puede estar en ejecución. Una línea como

```
pre-up pidof wpa_supplicant &
>/dev/null 2>&1 || &
wpa_supplicant -u -B
```

directamente bajo `iface wifi0...` en `/etc/network/interfaces` iniciará `wpa_supplicant` si no se está ejecutando ya.

Respecto a las cosas que podrían ir mal: `wpa-driver wext` se usa en todas las tarjetas WiFi que soportan las extensiones wireless de Linux (la inmensa mayoría de los chipsets soportados por el kernel lo hacen). Sin embargo, podría tener que usar un driver diferente en algún raro caso. Consulte la página man de `wpa_supplicant` y los archivos de ayuda. Sin embargo, no todos los “viejos” drivers están compilados con versiones actuales de `wpa_supplicant`. Igualmente, `wpa_supplicant` podría no ser capaz de arrancar la tarjeta y escanear en busca de señal antes de que el cliente DHCP se inicie. Podemos verificarlo con el comando

```
iwconfig wifi0
```

(use el nombre de su tarjeta en lugar de `wifi0`) para ver si la tarjeta WiFi intenta asociarse a un punto de acceso. Ver la dirección MAC de un punto de acceso en la información mostrada y una potencia de señal del 70% o más es una buena indicación.

El cliente DHCP (*dhclient*) podría fallar al intentar conseguir una dirección IP del servidor DHCP de los puntos de acceso. Algunas tarjetas WiFi tienen que llevarse al estado “up” antes de que puedan aceptar nada de la red local. El comando para hacer esto es

```
ifconfig wifi0 up
```

sin una dirección IP. Esto puede llevarse a */etc/network/interfaces* como directiva “pre-up”, como se mostró previamente con *wpa_supplicant*.

Ahora la parte más molesta: conseguir una dirección IP mediante DHCP. Aunque todo lo demás parece estar configurado correctamente, otro servicio podría estar acaparando la tarjeta WiFi, probablemente el NetworkManager. Debido a su sencilla auto-configuración y administración con un applet de escritorio (nm-applet), se ha vuelto muy popular en muchas distribuciones. NetworkManager ejecuta su propio cliente DHCP, que es

```
ps auxwww | grep dhclient
```

para la actual, o

```
ps auxwww | grep dhcdd
```

para las versiones antiguas (0.6.4). Debido a que el cliente DHCP propio de NetworkManager mantiene la interfaz en estado “scanning”, todo lo que configuremos manualmente (direcciones IP estáticas) o dinámicamente vía *dhclient* desaparecerá tras unos segundos.


Para configurar manualmente el WiFi (más bien, configuración “semi-automática” con *ifup*), debemos asegurarnos de que NetworkManager no esté ejecutándose

```
/etc/init.d/network-manager stop
/etc/init.d/dhcdd stop
```

y que ningún otro cliente DHCP mantenga ocupada a la tarjeta WiFi.

Si *iwconfig wifi0* no muestra señal, puede ser que nuestra tarjeta WiFi tenga un interruptor que esté apagado. Algunos módulos WiFi tienen una opción *rkill = 0* para ignorar la posición del interruptor, pero para la mayoría de las tarjetas, tenemos que ejecutar un kernel actual que pueda habilitar el dispositivo al pulsar el botón “WiFi”.

Módem UMTS

 He convertido recientemente a un amigo mío a Linux (Ubuntu), y le ha gustado, pero necesita conectarse a la Web

mediante un módem UMTS BandRich C100. He intentado todas las sugerencias de foros y no me ha funcionado nada: no se conecta. ¿En qué me estoy equivocando? Debido a este pequeño problema, otras dos personas que iban a pasarse a Linux están dudando.



Los módems UMTS funcionan como los módems normales. Recuerde que Linux no necesita “drivers” para cada uno de los dispositivos o marcas. En lugar de esto, los módulos del kernel sólo tienen que soportar el chipset del dispositivo serie asociado que use el módem.

Los kernels recientes tienen soporte para este módem. Cuando se conecta, un archivo llamado *ttyUSB0* debería aparecer dentro del directorio */dev*, el cual permite a los programas acceder al módem. Si no aparece, necesitaremos un kernel de Linux más reciente.

Ahora tenemos que suministrar la información de acceso al módem, del mismo modo que cuando lo instalamos bajo Windows. Como estamos cambiando una configuración del sistema que es global en este equipo, necesitaremos permisos de administrador. Usamos *sudo* o *su* para cambiar a la cuenta de root. Nótese que esta es una tarea de configuración que se hace una sola vez. Tras esto, establecer la conexión es muy sencillo.

Para una conexión de marcado UMTS se necesitan crear dos archivos de configuración de manera independiente al programa de marcado. Yo utilizo aquí el método manual con shell para crear la conexión, ya que funciona en cualquier caso, pero podemos transferir la misma información a un programa de marcado gráfico como *kppp* si preferimos una herramienta gráfica a la línea de comandos.

Como pre-requisitos, deben instalarse *pppd* y *chat*. El archivo */etc/chatscripts/umts* se comunicará con el módem y debería contener las líneas mostradas en el Listado 1. Debemos tener mucho cuidado con los caracteres. Una vez sea llamado por *pppd/chat*,

- envíe el número PIN para desbloquear el módem. Debemos cambiar 1234 por nuestro PIN particular.
- envíe la configuración específica del proveedor *AT+CGDCONT=1,"IP","internet"*. En función del proveedor de servicio móvil que estemos usando, esto puede variar, por lo que tendremos que preguntar a nuestro proveedor acerca de la configuración correcta si no se especifica la información de conexión proporcionada cuando compramos el módem.
- marque **99 1#*. Este número de teléfono no es real, pero inicia el marcado Internet para UMTS. El número **99#*, más breve, podría funcionar igualmente.

El segundo archivo que deberíamos ubicar en */etc/ppp/peers/umts* tiene como finalidad configurar los parámetros del protocolo de Internet (véase el Listado 2). Entre todo el resto de parámetros descritos con detalle en *man pppd*, la línea

```
user ""
```

indica que esta conexión funciona sin autenticación login/password, lo que es cierto para todos los proveedores UMTS, hasta donde tengo conocimiento (la autenticación se hace mediante el número de teléfono de nuestro módem). Nuevamente, */dev/ttyUSB0* indica el archivo de dispositivo coincidente con el

Listado 1: /etc/chatscripts/umts

```
01 TIMEOUT 120                                AT+CGDCONT=1,"IP","internet"
02 ABORT BUSY                                  08 TIMEOUT 120
03 ABORT "NO CARRIER"                        09 ABORT ERROR
04 "" ATE1                                     10 REPORT CONNECT
05 OK AT+CPIN=1234                             11 TIMEOUT 10
06 TIMEOUT 4                                   12 OK ATD*99***1#
07 OK-AT+CPIN?-OK                             13 CONNECT \c
```

Listado 2: Configuración Parámetros Protocolo Internet

```
01 noauth                                       08 debug
02 connect "/usr/sbin/chat -v -f              09 passive
    /etc/chatscripts/umts"                    10 noipdefault
03 /dev/ttyUSB0                                11 ipcp-accept-local
04 115200                                       12 ipcp-accept-remote
05 modem                                         13 usepeerdns
06 persist                                       14 user ""
07 crtscts
```

módem. El resto simplemente se asegura de que el módem configure todo adecuadamente para el acceso a Internet (rutado por defecto, nombres de servidores, etc.).

Tras instalar estos dos archivos, el comando

```
pppd nodetach call umts
```

iniciará la conexión. Debido al *debug* del segundo archivo de configuración, seremos capaces de seguir el progreso en el establecimiento de la conexión. En cuanto veamos líneas con direcciones, podremos acceder a los servicios de Internet, y podremos finalizar la conexión con Ctrl + C en la ventana en la cual iniciamos pppd.

Particiones

? A pesar de tener un equipo Linux desde hace un par de años, soy esencialmente un “novato”. Tengo varios libros sobre la materia, pero no he encontrado respuestas a mis preguntas acerca de formatear y particionar discos duros.

He adquirido un disco duro externo USB con una capacidad declarada de 1 TB. Quiero formatearlo para hacer un uso más eficiente de esta gran capacidad.

Hace muchos años (yo usaba Windows 3.1 por aquel entonces), recuerdo que el tamaño de partición más eficiente era 128 MB. En una partición como ésta, un sencillo archivo de texto que ocupaba unos 2 KB, ocuparía unos 64 KB en un disco de 1GB con una sola partición. Por lo que recuerdo, este problema tenía que ver con el número de “slots” en el VTOC.

Sospecho que ya no existen razones para seguir las mismas reglas hoy día, pero ¿cuáles son las consideraciones de hardware y de software implicadas a la hora de formatear y particionar un disco duro de gran capacidad para un uso eficiente de los recursos? Igualmente, ¿cuáles son los valores (por ejemplo, 128 MB) a tener como guía o límite al formatear y particionar? No he sido capaz de encontrar esta información.

Una cosa más: este disco duro en particular tiene algunas utilidades de copia de seguridad que sólo funcionan con máquinas Windows. Están ubicadas en una única partición formateada con NTFS, lo cual no es muy útil con mi entorno openSUSE 10.2. ¿Qué debería hacer para 1) librarme de la basura pensada para Windows, y 2) hacer este disco duro tan útil como sea posible para mi entorno Linux?

? Debería ser capaz de encontrar información acerca de los sistemas de archivo [(V) FAT, ext2, ext3, JFS, ReiserFS,

XFS, y muchos más] en las páginas man de las correspondientes herramientas de formateado *mk < sistemadearchivo >* y en las páginas de los desarrolladores. Igualmente, los artículos de la Wikipedia acerca de los sistemas de archivo específicos son excelentes.

Para grandes particiones, los sistemas de archivo con una tabla índice “estática” para archivos y directorios (tabla de asignación de archivos) probablemente no son buenas elecciones. FAT32 y ext2 usan estos índices de archivos, y pueden agotarse rápidamente cuando la partición se llena con millones de pequeños archivos. Dependiendo del tamaño del bloque y del tamaño de porción, los archivos pueden tener físicamente un tamaño asignado mayor del necesario. Con los sistemas de archivos estáticos es posible tener el 99% del espacio libre, y aún así no poder crear un nuevo archivo debido a que el índice de archivos del sistema de archivo está lleno. Si usamos por ejemplo ext2, deberíamos planificar cuidadosamente si vamos a usar el disco para muchos pequeños archivos o para pocos y grandes, y usar los parámetros *mke2fs -N y -I* en consecuencia.

Con los nuevos sistemas de archivos, ya no tenemos índices fijos para las entradas de archivo. La (antigua) tabla de asignación de archivos se reparte ahora por toda la partición como una lista enlazada. La lista se extiende tanto como sea necesario y se optimiza para encontrar los archivos más rápidamente con árboles balanceados. De igual modo, el tamaño máximo de archivo ya no está limitado a unos pocos gigabytes como ocurría con FAT32.

Si vamos a usar solamente Linux para acceder al disco, yo recomendaría ReiserFS, JFS o XFS para un uso más eficiente. Sólo consideraría ext3 si no tenemos previsto crear más de 32.000 subdirectorios dentro de un solo directorio, ya que tiene (casi) las mismas limitaciones de inodos que ext2.

NTFS, en teoría, usa métodos similares, pero tendremos que instalar *ntfs-3g* para poder montar particiones NTFS en lectura y escritura bajo Linux. Sin embargo, muchos tipos de archivo específicos de Unix, como los enlaces simbólicos, archivos de dispositivo y sockets, no están soportados en NTFS. Si tenemos previsto usar el disco para intercambiar información con sistemas Windows, para procesado de vídeo, u otros archivos multimedia que no necesiten atributos de archivo o permisos especiales, NTFS podría no ser una mala elección.

Como disco de información, tener una única gran partición nos dará probablemente

la mayor flexibilidad. Podemos liberarnos de los programas no útiles del disco creando un nuevo sistema de archivos con *mkreiserfs*, *mke3fs*, *ntfsformat* o de la herramienta de formateado del sistema de archivos que elijamos. Para borrar realmente toda la información con seguridad, podemos sobrescribir en primer lugar todo el disco con ceros o números aleatorios con

```
dd if=/dev/zero \
of=/dev/diskname bs=1024k
```

o

```
dd if=/dev/urandom \
of=/dev/diskname bs=1024k
```

o con la herramienta *wipe*, que también se supone que machaca el contenido de la partición de manera que no se pueda recuperar.

Tras esto, como acabamos de invalidar el sector de arranque en la tabla de particiones, podemos usar *fdisk* o *cdisk* para crear nuevas particiones en el disco antes de formatearlo con el sistema de archivos de nuestra elección.

Drivers de Impresora

? Me encanta su Knoppix 5.1.1. Funciona en la mayoría de mis equipos. Knoppix 5.3.1 es más complicado y no me funciona en tantas máquinas como con Knoppix 5.1.1.

Mi pregunta: Knoppix 5.1.1 no tiene todos los drivers de impresora instalados que necesito. ¿Sería difícil añadir los drivers de impresora a Knoppix 5.1.1 (por ejemplo para la HP Deskjet F2180)?

? Necesitará un paquete *hplip* actual para tener soporte completo de esta impresora (también usará el escáner integrado con XSane, pues es un dispositivo impresora/escáner todo en uno, ¿no es así?).

Los comandos para instalar *hplip* son

```
sudo apt-get update
sudo apt-get install hplip
```

pero si aprecia que esto requiere muchos paquetes como dependencias, por ejemplo si su sistema de impresión basado en cups es muy viejo, puede intentar instalar una versión más antigua

```
sudo apt-get \
install -t stable hplip
```

para conseguir que su impresora esté lista para funcionar. ■