



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: [klaus@linux-magazine.com](mailto:klaus@linux-magazine.com)

### Conexión en Caliente en Dapper Drake

**?** Acabo de leer el artículo acerca de la conexión en caliente del número 24 de Linux Magazine y me gustaría realizar algunos experimentos. El archivo `/proc/sys/kernel/hotplug` lo tengo vacío tanto en mi sistema Drake como en FC5. ¿Tengo que compilar el kernel para conseguir ambas funcionalidades? En ese caso, ¿cómo consigo Ubuntu montar automáticamente las memorias USB?

# EL KONSULTORIO DE KLAUS



**?** Si usamos una distro reciente, `/proc/sys/kernel/hotplug` está vacío debido a que ya no se usa. La mayoría de las distros usan ahora `udev(d)` para autodetectar cambios en la configuración del hardware, cargar los módulos e iniciar servicios tan pronto como se conecta algo en caliente. En el Número 24 de la revista Linux Magazine edición en castellano se habla de este tema.

`udev`, junto con HAL (hardware abstraction layer), notifica al escritorio cualquier conexión de dispositivo a través del bus del sistema `dbus`, y el escritorio (si está configurado para ello) nos presentará varias opciones sobre qué hacer con el dispositivo: montarlo como disco duro o acceder a él con algún programa adecuado (para discos de vídeo o cámaras digitales).

En cualquier caso, si tenemos `/proc/sys/kernel/hotplug`, el kernel ya está compilado con el soporte para conexión en caliente, soporte que `udev` también necesita, no sólo a través del sencillo script ubicado en `/proc/sys/kernel/hotplug`.

Las configuraciones específicas de la conexión en caliente y los script se guardan en `/etc/udev/*`. El archivo de configuración responsable de mostrar o retirar los iconos del disco en el escritorio, o de iniciar los programas asociados es `/etc/udev/hal.rules`, que realiza las llamadas para las acciones específicas a través de un socket al que escucha nuestro escritorio. Knoppix también hace uso de esta funcionalidad para añadir entradas en `/etc/fstab`, y para crear sus propios iconos para el escritorio KDE, pero este tipo de adaptación es específica de cada distribución.

### Fijar el Tamaño de Página Correcto

**?** Soy novato en Linux y estoy usando PCLinuxOS en mi equipo.

¿Cómo puedo fijar el tamaño de página de mi navegador Web? Cuando cambio a otro escritorio y pulso sobre “konqueror”, la página Web me vuelve a aparecer en el tamaño original.

**?** Existe una configuración en el perfil del navegador al arrancar. Tras redimensionar Konqueror a la geometría deseada, nos dirigimos al menú *Settings* y elegimos *Configure View Profiles...* En el cuadro (véase la Figura 1), marcamos *Save window size in profile* y guardamos este perfil como *webbrowsing*, que es el perfil por defecto si Konqueror arranca en modo navegador Web.

Nótese que también existe un bug en Konqueror (al menos yo pienso que es un bug). Al arrancar con un tamaño definido, por ejemplo con:

```
konqueror -geometry 850x600+85+70
http://www.debian.org/
```

como se realiza con algunos scripts, Konqueror parece “recordar” esta geometría inicial e intenta fijarla automáticamente en cada página que se carga, lo cual puede resultar bastante molesto.

### Pendrivel USB de Arranque

**?** Soy nuevo en Linux y quiero crear una unidad de arranque USB. ¿Alguna página que me pueda guiar paso a paso en este propósito?

Crear una unidad de arranque USB es (técnicamente) sencillo. Yo empezaría con una breve introducción al tema de arrancar desde USB.

**?** No todas las placas base son capaces de arrancar desde USB. Más aún, algu-

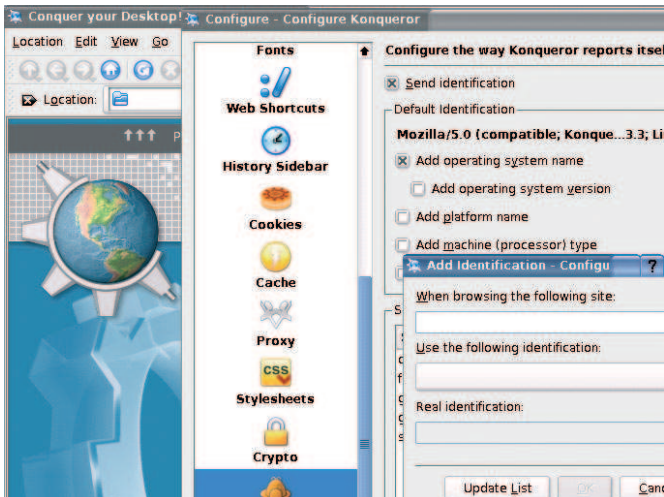


Figura 1: Guardamos el tamaño de ventana en el perfil de Konqueror.

nas combinaciones de pendrives USB y placas no arrancan juntas, aunque lo hagan por separado. Además de esto, algunas combinaciones de teclado sólo arrancan de ciertas maneras, en función de cómo se preparó la llave USB y cómo se configuró la BIOS.

Para terminar de empeorar las cosas, algunas llaves intentan empezar a cargar el sistema operativo pero yerran en los procedimientos del checksum de la BIOS interna tras transferir unos pocos kilobytes de información. He visto incluso casos en los que el éxito depende de qué puerto USB usemos y si usamos o no un cable entre el pendrive y el ordenador.

Esto puede que no suene muy prometedor. Sin embargo, la mayoría de las combinaciones de placa (capaz de arrancar desde USB) y llave, funciona. Un arranque desde un pendrive USB es una buena manera de ejecutar Knoppix, ya que podemos guardar nuestra imagen persistente (o disco virtual escribible) directamente en el mismo pendrive.

Tenga en cuenta que hay algunos obstáculos que tenemos que superar en el proceso de arranque. En el momento en el que el bootloader debería leer el kernel de Linux y el initrd desde el pendrive, no hay sistema operativo que se esté ejecutando, y la BIOS del ordenador debe hacer todo el trabajo de detectar las particiones y cargar la información de la unidad sin tener ningún otro “driver” disponible más que los del firmware de la BIOS en la placa. Por otro lado, tan pronto como se carga el kernel de Linux, la llave USB “desaparece”. En otras palabras, ya no es visible y tiene que re-detectarse por el kernel usando los módulos del disco RAM inicial que contiene los drivers USB y el sistema de archivos usado por las parti-

tes ejemplos vamos a usar `/dev/sda` como nombre del dispositivo del pendrive USB. Esto puede ser incorrecto en muchos casos, especialmente si nuestro equipo alberga discos SATA. Es aconsejable volver a verificar el nombre de dispositivo asignado al dispositivo USB cada vez que lo conectemos.

Para la mayoría de los comandos no vamos a necesitar permisos de root si los archivos del dispositivo que están en `/dev` se pueden escribir por un usuario de cuenta normal y son montables mediante las correspondientes entradas de `/etc/fstab`. En caso contrario, usaremos `sudo` o un shell de root, con lo que debemos ser especialmente cuidadosos.

**Paso 1: Creamos el sistema de archivos** Algunos pendrives USB sólo arrancan si se particionan de cierta manera, y algunas BIOS aparentemente buscan un cierto tipo de cabecera de partición y sistema de archivo, o renunciarán a arrancar. Hemos tenido casos en los que una placa sólo quería arrancar desde un pendrive USB sin particionar, y otros en los que se requerían tanto el flag de un sector de arranque DOS como el de la partición de arranque.

En primer lugar podemos probar con una tabla de particiones por defecto y un sistema de archivos más o menos sin modificar. La mayoría de los pendrives USB recientes tienen una (especie de) tabla de particiones válida en la que la primera partición primaria está formateada en FAT16 o FAT32. En este caso debemos asegurarnos de que la partición es arrancable.

Para una partición de un pendrive USB que aparezca como `dev/sda1`, el comando para configurar el flag como “arrancable” sería:

```
sfdisk -A1 /dev/ sda
```

Para reparticionar la unidad USB podemos usar las mismas herramientas que usamos para un disco duro normal (es decir, `qparted` o, mi favorito: `fdisk` o `sfdisk`).

El tipo de partición para un sistema de archivo DOS/FAT (el menos problemático para el arranque desde una unidad USB) sería “6” para FAT16 y “b” para FAT32. Usamos FAT16 para particiones menores de 2GB y FAT32 si son mayores. Véase arriba el comando de `sfdisk` para fijar el flag de la partición como “arrancable” (o “activo”), lo cual es independiente del sistema de archivo de las particiones o del contenido.

El comando de `mkdosfs` para dar formato a una partición FAT16 como `/dev/sda1` es:

```
mkdosfs -F16 /dev/sda1
```

Mientras que para FAT32, sería:

```
mkdosfs -F32 /dev/sda1
```

No hace falta decir que, llegados a este punto, perderemos toda la información que resida en la partición. Por lo que si reparticionamos y reformateamos un pendrive USB, deberíamos hacer copia de seguridad de toda la información de la partición (y posiblemente de toda la información de toda partición del disco que podamos confundir con la unidad USB en un momento dado).

Muchos pendrives USB no muestran la nueva tabla de particiones inmediatamente, por lo que es buena idea sacar y reconectar el pendrive USB o disco duro cada vez que sigamos los pasos anteriormente mencionados. Volvemos a verificar la nueva partición con:

```
cat /proc/partitions
```

tras haberlo conectado de nuevo y después de algunos (bastantes) segundos. (Algunas unidades son bastante lentas en la detección de la tabla de particiones). Si la unidad ya aparece como un nombre de nuevo dispositivo (es decir, `sdb` en lugar de `sda`), probablemente hemos olvidado desmontar el sistema de archivos antes de desconectarla. Podemos hacerlo ahora. No obstante, seguramente no exista ya, tras la repartición de la unidad.

Para los pendrives USB sin partición (si el método de arranque particionado

falla), nos saltamos el paso de particionado y tecleamos el comando `mkdosfs` sobre el pendrive USB entero.

Esa ha sido la parte más difícil, dado que aún no se ha comenzado la instalación. El resto es realmente fácil.

**Paso 2: Instalar syslinux como cargador del kernel**

Lo que `isolinux` hace para un CD o DVD, `syslinux` lo hace para particiones o discos formateados en DOS.

Para unidades particionadas:

```
syslinux /dev/sda1
```

Para unidades no particionadas:

```
syslinux /dev/sda
```

(Debemos asegurarnos de que estamos usando un nombre correcto de dispositivo para nuestro sistema).

Si usamos un pendrive USB particionado, podemos también configurar (o resetear) el sector de arranque adecuadamente. No obstante, no siempre es necesario hacer esto. Puede que primero queramos ver si el arranque funciona sin cambiar el MBR.

Instalamos un sector de arranque `syslinux` para unidades particionadas:

```
ms-sys -s devicename
```

Usamos `/dev/sda` si `/dev/sda1` es la partición FAT16/FAT32 del pendrive USB. Debemos vigilar que el nombre del dispositivo sea correcto.

**Paso 3: Copiamos los archivos de arranque y la información**

En este paso, tenemos que montar el pendrive USB (o partición, si usamos el modo particionado) en modo sensible a mayúsculas:

```
mkdir -p /media/usbkey
mount -t vfat -o rw,umask=000,shortname=winnt /dev/sda1 /media/usbkey
```

A continuación, deberíamos cambiar `/dev/sda1` a la partición en cuestión, y entonces saltarnos el número de la partición en caso de usar todo el pendrive.

Copiamos ahora los archivos de arranque desde el CD o DVD de Knoppix. Vamos a suponer que el CD se monta en una sesión live de Knoppix, aunque en realidad podemos realizar estos pasos perfectamente con cualquier distribución de Linux.

```
cp /cdrom/boot/isolinux/* /media/usbkey
```

Mientras que `isolinux` busca todos los archivos de arranque en `/boot/isolinux`, `syslinux` los busca en el directorio `/` del sistema de archivos FAT.

Pero...un momento... ¡`syslinux` da un nombre diferente a su archivo de configuración!

```
mv /media/usbkey/isolinux.cfg /media/usbkey/syslinux.cfg
```

Podemos borrar con seguridad los archivos específicos de `iso/eltorito`, ya que no los vamos a necesitar.

```
rm -f U /media/usbkey/isolinux.bin /media/usbkey/boot.cat
```

En este punto, el pendrive ya es arrancable, y contiene tanto el kernel como el `initrd`, y podemos comenzar a copiar los archivos de `cloop`, que residen en el directorio de Knoppix del CD.

```
rsync -Hav --progress /cdrom/KNOPPIX U /media/usbkey/KNOPPIX
```

Una vez hecho esto, tenemos que desmontar el dispositivo para poder sincronizar el búfer de datos pendientes a disco y cerrar la sesión. No se sorprenda si este comando se toma un buen tiempo en ejecutarse. Puede haber muchos datos sin escribir en memoria debido la fabulosa caché del búfer dinámico de bloques de Linux, que hace parecer que las escrituras en disco son más rápidas de lo que realmente son.

```
umount /media/usbkey
```

**Paso 4: Configurar la BIOS del ordenador**

Una placa capaz de arrancar desde USB normalmente tiene al menos dos modos de arranque por USB:

- Arranque desde disco duro USB
  - Arranque desde unidad ZIP USB
- lo que debería en realidad llamarse “arranque desde una unidad USB particionada” y “arranque desde una unidad USB sin particionar”.

Para nuestro nuevo pendrive USB “por defecto”, la opción *disco duro USB* sería

la correcta, pero en realidad depende de lo que hayamos hecho en el paso 1.

**Solución de Problemas**

Si nuestro pendrive USB no funciona a la primera, el problema podría ser:

- Que el pendrive USB no se reconozca como arrancable en la secuencia de arranque. Debemos comprobar que la configuración de la BIOS se ajusta a la manera en la que hemos preparado la unidad, podemos intentar conectarla a otro puerto USB y, como último recurso, probar a preparar el pendrive con otro método de los descritos anteriormente.
- Que el pendrive USB arranque, pero que la carga del kernel falle tras cierto tiempo, o tras un “error de checksum”. En este caso, probablemente el pendrive USB no funcione con esa placa base. Tenemos un buen índice de éxito usando pendrives “Trekstor”. Si el presupuesto nos lo permite, podemos probar varias marcas en busca de alguno compatible. Existe la posibilidad de que el pendrive funcione en el otro modo (particionado o sin particionar).
- Que el kernel y el `initrd` arranquen, pero que Knoppix se queje de que “no encuentra el sistema de archivos de KNOPPIX” y nos derive a un shell de rescate. El pendrive no se detectó en el arranque. Esto puede ser debido a una respuesta lenta (las particiones del pendrive no disponibles en el momento en el que `linuxrc` trata de montarlas) o el controlador no está soportado. Aparentemente la BIOS tuvo conocimiento del pendrive USB en un primer momento, aunque Linux no lo encuentra más tarde. Ha de recordarse que la BIOS usa un conjunto de drivers completamente diferentes para el pendrive de los que incluye Linux. Este inconveniente puede ser un problema de interrupciones con el controlador. En este caso, los cheatcodes candidatos de Knoppix pueden ser de utilidad:

```
knoppix acpi=off
noapic pnpbios=off
pci=bios irqpoll...
```

Knoppix 5.1.0 incluye un script llamado `mkbootdev`, escrito por Martin Öhler y yo mismo, que realiza todos los pasos comentados con unos pocos golpes de ratón. Si ejecutamos `mkbootdev` desde un terminal en un sistema Knoppix CD o DVD mientras está conectado nuestro pendrive USB sin particiones montadas, nos guiará a través de los pasos 1 al 3. ■