

Conexión en caliente con Udev, HAL y D-Bus

ENCHUFADOS

Lo que los usuarios quieren es que el hardware funcione.

Las distribuciones

Linux actuales están progre-

sando mucho para conseguirlo. En

este artículo, vamos a investigar cómo funciona el sistema de

conexión en caliente. **POR OLIVER FROMMEL, MARCEL HILZINGER Y RENÉ REBE**

Es mucho pedir? Sólo quiero que Linux ejecute el programa correcto cuando conecto una cámara digital nueva, pero el sistema operativo ha decidido no hacerlo. Esta situación es bastante común, aunque ya ha empezado a solucionarse. Linux debería manejar cualquier clase de hardware adecuadamente, pero la capacidad de manejar dispositivos que se conectan o desconectan a los puertos USB y Firewire mientras el ordenador está encendido (conocido como "conexión en caliente" [1]) se ha llegado a convertir en algo importante. Este artículo explica qué hace un sistema Linux moderno para manejar dispositivos que se le conectan y por qué, a veces, no funcionan tal y como estaba previsto.

Agentes Secretos

Cuando se conecta un dispositivo que soporta conexión en caliente, el kernel manda una señal de evento al sistema de conexión en caliente. Dicho sistema usa Udev para crear un fichero de dis-

positivo para el dispositivo y luego llama al agente apropiado. Un agente es un script, normalmente almacenado en el directorio `/etc/hotplug`, que se encarga de manejar las acciones asociadas a estos eventos. En este caso, el agente realiza la tarea de añadir y registrar el nuevo dispositivo.

Los pasos que el agente realiza pueden variar dependiendo de la distribución y del tipo de hardware que se esté instalando. El agente USB, para el hardware que se conecta a este puerto en caliente, puede ser un buen ejemplo. El agente USB primero comprueba si el controlador para el nuevo dispositivo está disponible (por ejemplo, `isdn`) y luego llama a `modprobe` para que cargue el módulo. Si el agente encuentra un script con el mismo nombre que el controlador en el directorio

`/etc/hotplug/usb/`, el agente ejecuta el script.

El acto de cargar un módulo típicamente dispara unos cuantos eventos más

de la conexión en caliente, que de nuevo invoca a otros agentes. Es común que varios agentes de conexión en caliente trabajen juntos. Por ejemplo, cuando se conecta un disco duro externo, primero se carga el agente USB y luego se carga el agente de SCSI para montar las particiones individuales como dispositivos SCSI con la ayuda del módulo `usb-storage`. Si se conecta un conector Bluetooth, el agente USB se ejecuta primero, seguido por un agente Bluetooth llamado `bluetooth.agent`.

La Lista Negra

El fichero `/etc/hotplug/blacklist` contiene una lista de módulos que ningún agente

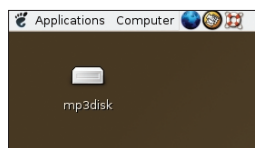


Figura 1: Tras modificar el fichero de configuración de Udev, el icono del escritorio Gnome tiene una etiqueta más intuitiva.

encontrar los identificadores USB (IDs). Si existe un fichero de dispositivo, *udevinfo -q ruta -n /dev/fichero* muestra la ruta en el SysFS (véase el Cuadro 2: "Ficheros de Dispositivos con Udev"), pero no proporciona el punto de montaje */sys*. Se necesita la ruta como parámetro (*-p*), con este comando para obtener la información SysFS:

```
udevinfo -a -p /block/hda/hda1
...
SYSFS{idVendor}="10d6"
...
```

Este comando ayuda a descubrir los valores específicos de los dispositivos que son necesarios para las configuraciones especiales. Existe un "how-to" detallado para

crear ficheros Udev en [2]. El sitio web de Fedora tiene un breve resumen del sistema Udev [3].

Asignaciones Propias

Un fichero de asignaciones contiene una o varias IDs que identifican de forma unívoca un componente hardware. Cuando un dispositivo que coincida con

Cuadro 2: Ficheros de Dispositivos con Udev

En Linux las aplicaciones usan ficheros de dispositivos para acceder al hardware. Estos ficheros especiales situados en el directorio */dev* están definidos por referencias a sus tipos, así como por sus números mayores y menores, que los enlaza con el kernel.

En el pasado, este directorio era un repositorio para todo tipo de ficheros de dispositivos, unidades de disco IDE y SCSI, USB, IEEE1394 y dispositivos virtuales. Esto significa que el directorio */dev* tiene miles de entradas.

Este sistema tiene algunas desventajas. Por ejemplo, no se indica que dispositivos existen realmente o han sido identificados correctamente por los controladores. Además, los ficheros de dispositivos pueden cambiar a veces, dicho de otro modo: el orden aleatorio en el que los dispositivos se conectan al sistema decide que dispositivo SCSI se ligará a */dev/sg2*.

Nuevos Mundos

Udev [1] es el sucesor de los ficheros de dispositivos estáticos y la mayoría de las distribuciones actuales lo usan. Udev se basa en el mecanismo de conexión en caliente que genera los ficheros de dispositivos según se requieran. Cuando un dispositivo cambia, el kernel llama al programa especificado en */proc/sys/kernel/hotplug*, típicamente */sbin/hotplug*.

Dependiendo del tipo de dispositivo, el programa cargará módulos, modificará privilegios de acceso, configurará dispositivos de red o en el caso de Udev, manejará nodos de dispositivos.

El subsistema Udev necesita algunos detalles para ser capaz de crear un fichero: el tipo de dispositivo (carácter o bloque) y los números mayores y menores. En el kernel 2.6 o posteriores Udev referencia al sistema de ficheros *sys* (SysFS, que normalmente se encuentra en */sys*) para descubrir esta información.

Los dispositivos de bloques están localizados en */sys/block* y los de carácter en

/sys/class. Los números mayores y menores se almacenan en un fichero *dev*. Por ejemplo, el siguiente comando indica los números para el primer disco duro IDE, *hda*:

```
cat /sys/block/hda/dev
3:0
```

Udev puede analizar cualquier información SysFS, como la clase del dispositivo, nombre, número, etc., para crear los dispositivos apropiados. Si los nombres son estables. Udev puede incluso ejecutar programas complejos para comprobar si debería configurar una impresora que haya sido conectada en */dev/usb/lp0* o */dev/usb/lp1*, basando la decisión en el número de serie de la impresora. Es incluso posible usar nombres arbitrarios y llamar nodos, por ejemplo, */dev/lp-epson* y */dev/lp-kyocera*.

Configuración de Udev

Udev tiene dos opciones de configuración. Los ficheros en */etc/udev/rules.d* especifican los nombres de ficheros de dispositivos, otros en */dev/udev/permissions.d* especifican los privilegios. Las reglas por defecto crean ficheros de dispositivos que usan los nombres familiares de Linux.

Al principio de cada regla, hay una o más condiciones que tienen que ser detalladas si Udev tiene que crear los ficheros de dispositivos. A continuación le sigue el nombre. Lo siguiente es una entrada típica de las impresoras USB:

```
BUS="usb", KERNEL="lp[0-9]*",
NAME="usb/%k"
```

Si el dispositivo está conectado al bus *usb* y el nombre interno del kernel es *lp* con un número arbitrario, Udev crea un fichero con el nombre del kernel (*tal* y como se indica con *%k*) en el directorio */dev/usb*.

Además de las reglas estáticas de este tipo, también es posible llamar a programas externos. La página de ayuda de *man* tiene un ejemplo para los CD-ROMs

IDE que comprueba si un directorio llamado */proc* existe para identificar el dispositivo como CD-ROM:

```
KERNEL="hd[a-z]",
PROGRAM="/bin/cat /
proc/ide/%k/media",
RESULT="cdrom",
NAME="%k", SYMLINK="cdrom%e"
```

En este ejemplo, Udev llama a */bin/cat* para el fichero */proc* para todos los dispositivos cuyo nombre empiece por *hd*. Si el fichero especifica el CD-ROM como el medio, Udev recordará el nombre pero creará adicionalmente un enlace simbólico para *cdrom*. El parámetro *%e* le indica a Udev que seleccione el próximo número libre si un fichero con el mismo nombre ya existe.

Una buena solución es el uso de los números de serie de los dispositivos para realizar la asignación:

```
BUS="usb", SYSFS{serial}="
HXOLL0012202323480",
NAME="lp-epson"
```

Esta regla le dice a Udev que cree un fichero de dispositivo, */dev/lp-epson*, si encuentra un dispositivo con el número mencionado en su fichero *serial* bajo el árbol de directorios de SysFS.

Udev y los Privilegios Las reglas para los privilegios de acceso comprenden tan solo una línea con valores separados por el carácter dos puntos con el nombre, el propietario, el grupo y los privilegios.

```
usb/lp*:root:lp:0660
```

Todos los ficheros de dispositivos llamados *usb/lp** pertenecen al usuario *root* y al grupo *lp*. Los privilegios de acceso se especifican en formato octal.

El nuevo modelo de conexión en caliente está teniendo tal éxito que se está usando incluso para arrancar el sistema. Se llama a *udev* con las variables apropiadas para cualquier dispositivo conocido en */sys/class* y */sys/block*.

una de estas entradas se conecta al sistema, el subsistema automáticamente ejecuta el programa especificado, que puede ser un script.

Esto permite habilitar un adaptador WLAN USB que la distribución no configure de forma adecuada. En nuestro laboratorio, el sistema de conexión en caliente detectó el adaptador con el chipset Prism2, pero falló a la hora de ejecutar el script que lanzaba las funciones WLAN requeridas.

El comando `lsusb` nos proporcionó el ID del dispositivo USB conectado. El ID del vendedor en nuestro ejemplo es 0x0846 y del dispositivo es 0x4110.

Ahora tan sólo hay que añadir los siguientes valores hexadecimales a un nuevo fichero llamado `/etc/hotplug/usb/prism2.usermap`:

```
prism2 0x0003 0x0846 0x4110
0x0000 0x0000 0x00 0x00 0x00
0x00 0x00 0x00 0x00
```

La mayoría de las asignaciones se parecen a ésta y tan sólo usan los cuatro primeros valores. El primer valor especifica que programa debe ejecutar el sistema de conexión en caliente si los valores que siguen en la línea coinciden. El primer valor numérico es un campo de bits que especifica el número requerido de valores a coincidir. Si se quiere que la conexión en caliente compruebe los dos primeros valores, se necesita 0x0003. El primer bit indica un valor de 1, el segundo es 2 y los dos juntos es 3. El sistema ignora el resto de las columnas del registro y por ello se rellenan con el valor 0x00.

El script que se pretende ejecutar, `prism2`, tiene que estar en el mismo directorio y tiene que ser ejecutable, `chmod +x`. En nuestro ejemplo, el script de comienzo `rc.wlan` del paquete Prism2, configura el interfaz `wlan0 network` y luego solicita al servidor DHCP una dirección IP:

```
#!/bin/sh
/etc/rc.wlan start
/sbin/ifconfig wlan0 up
/sbin/dhclient wlan0
```

Tras estos cambios, el adaptador USB WLAN funcionó inmediatamente tras conectarse. Desafortunadamente, nuestros intentos para configurar una cámara

de vídeo DV de la misma forma fallaron debido al lamentable estado del subsistema Firewire. El controlador IEEE1394 para el kernel actual no proporciona la información SysFS que se necesita, así que no hay otra alternativa que ejecutar

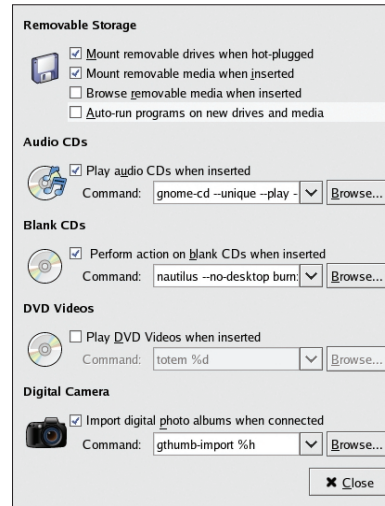


Figura 3: En Gnome 2.8, `gnome-volume-properties` especifica que aplicación se ejecuta por cada evento de conexión en caliente.

`mknod` para crear el fichero de dispositivo.

Del Hardware a la Aplicación

Otra capa del sistema de conexión en caliente proporciona una interfaz entre el hardware y las aplicaciones. La capa de abstracción de hardware (HAL, [4]) tiene información detallada del hardware, que se almacena en ficheros de información de dispositivos (*.fdi*).

Se puede usar la capa HAL para realizar cambios para dispositivos especiales. Por ejemplo, un usuario resolvió los problemas de su iPod que no se desregistraba correctamente [5].

Los ficheros FDI son ficheros con formato XML que proporcionan descripciones detalladas de los dispositivos. El comando `lshal` muestra los detalles. El `hal-device-manager` posee los mismos datos pero con una interfaz gráfica agradable (Figura 2). Los usuarios de Suse tendrán que arreglárselas sin los componentes HAL, ya que Suse tiene una forma diferente de manejar los detalles del hardware (véase el Cuadro 1: "Conexión en Caliente con Suse Linux").

En el futuro, las aplicaciones serán capaces de preguntar los detalles del hardware por D-Bus [6]. D-Bus es un sistema de comunicación software en el que las aplicaciones pueden conectarse y registrarse ellas mismas para ciertos eventos. Por ejemplo, un programa de edición de vídeo podría necesitar saber cuándo una cámara nueva se conecta a un PC. El programa `gnome-volume-properties`, cuya última versión ha presentado Gnome, usa D-Bus y HAL para asociar las aplicaciones con los eventos de conexión en caliente (Figura 3).

Se espera que D-Bus juegue un papel importante en las comunicaciones entre aplicaciones en Gnome, aunque no hay muchas aplicaciones que hagan uso de esta característica en la actualidad.

Un Mundo Feliz - Algún Día

A pesar de todo el progreso hecho con respecto a la detección de dispositivos, las cosas aun no son perfectas. Los agentes del sistema de conexión en caliente necesitan información detallada del hardware y esta información será obsoleta en un sistema que tenga unos meses de antigüedad. Una base de datos online para los componentes hardware podría ayudar y los usuarios podrían contribuir con los detalles que hayan elaborado en formato FDI.

El proyecto HAL se está moviendo hacia esta dirección, dando al sistema de conexión en caliente la información que el kernel no puede proporcionar. Ya hay unas cuantas distribuciones que han empezado a utilizarlo. Esperemos que Suse siga esta tendencia. Cuanto más consistente sea la gestión del hardware en Linux, mejor. ■

RECURSOS

- [1] Conexión en caliente en Linux: <http://linux-hotplug.sourceforge.net>
- [2] Como escribir nuestras propias reglas Udev: <http://www.reactivated.net/udevrules.php>
- [3] Documentación Udev de Fedora: <http://fedora.redhat.com/docs/udev>
- [4] HAL: <http://www.freedesktop.org/Software/hal>
- [5] iPod con Udev: <http://www.kgarner.com/blog/archives/2005/01/11/fc3-hal-ipod/>
- [6] D-Bus: <http://www.freedesktop.org/Software/dbus>