

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: [Alejandro Perez García 2003-2004.](#)

Si te gusta lo que ves, **puedes contratarme** para impartir **cursos presenciales** en tu empresa o para ayudarte en proyectos (Madrid).

Contacta: alejandropg@autentia.com.



Una vez escuche esta frase:

Cuando todo esta bajo control, es que no vamos suficientemente deprisa

La mayoría de las veces construimos estos tutoriales al mismo tiempo que recordamos como hacer cosas o probamos nuevas tecnologías, poniendo más celo en el contenido que en la forma (cosa que veo que perturba a muchos de los lectores por lo que os pido disculpas). Es cuestión de criterios aunque trataremos de mejorar.

Espero que seáis comprensivos y generosos ayudando a encontrar las erratas, faltas de ortografía, enlaces rotos u otros posibles errores en estas páginas. Escribidme siempre que encontréis algun error y pronto ya no habrá y os estaremos francamente agradecidos. Este es tu Web colabora libremente ...

[Roberto Canales](#)

Descargar este documento en formato PDF [usbdebian.pdf](#)

Unidades Almacenamiento

Tu CD-ROM, DVD-ROM, CD-RW, DVD-RW, Disco Duro, ZIP o DDS
¡en Dell!

Solución de Backup Online

Copias de seguridad remotas.
¡Prueba gratis para particulares!

Reparaciones Backup

Unidades de cinta,
autocargadores, librerías -
reparación profesional.

Disco Duro Portátil

Capacidad 40 Gb. Económico y
compatible con 9 tipos de tarjetas .

Anuncios Goooooogle

Como acceder por nombre a los dispositivos USB en GNU / Linux (Debian)

Índice de contenidos

1. Introducción
2. Entorno
3. Instalación
4. Configuración de Ficheros
 - 4.1. Configuración de udev
 - 4.2. Configuración de las reglas
5. Conclusiones
6. Sobre el autor

1. Introducción

A menudo cuando conectamos hardware externo (como un pendrive, un iPod, un disco duro, una iPaq) a través de los puertos USB nos encontramos con el problema de que no sabemos que nombre le ha dado el kernel al dispositivo (sda1, sdb1, sdc1, ...). Especialmente si conectamos más de un dispositivo a la vez.

Este problema resulta bastante incómodo a la hora de montar el dispositivo para acceder a él, sobre todo si queremos escribir reglas en el fichero /etc/fstab.

En este tutorial vamos a ver como podemos asignar nombres a los dispositivos, independientemente del nombre que le de el kernel, de forma que siempre podamos montar el dispositivo por el mismo nombre, y podamos escribir reglas en el fichero `/etc/fstab` basadas en estos nombres.

2. Entorno

- Hardware: Portátil Ahtex Signal X-9500M (Centrino 1.6 GHz, 1024 MB RAM, 60 GB HD).
- Sistema Operativo: GNU / Linux, Debian Sid (unstable), Kernel 2.6.9, KDE 3.3

3. Instalación

Debemos instalar el paquete `udev`. Para ello basta con:

```
# apt-get install udev
```

Este paquete es el que va a permitir, según una serie de reglas que podemos definir nosotros mismos, asignar nombres dependiendo del dispositivo.

Una vez instalado es recomendable reiniciar la máquina (aunque no obligatorio, como nos dice la propia instalación del paquete).

4. Configuración de Ficheros

4.1. Configuración de udev

Una vez instalado podemos ver que tenemos un directorio nuevo:

```
/etc/udev
```

En este directorio nos fijaremos especialmente en el fichero `udev.conf`. En este fichero podemos encontrar la línea:

```
udev_root="/dev/"
```

Esto indica donde `udev` va a crear dinámicamente los nodos de los dispositivos. Para separarla de la estructura habitual de `/dev` yo he preferido cambiarla por:

```
udev_root="/dev/udev/"
```

De forma que todos los nodos que cree `udev` estén en el directorio `/dev/udev`.

4.2. Configuración de las reglas

También cabe destacar el directorio `/etc/udev/rules.d/`. En este directorio es donde podemos situar todos los ficheros de reglas que queremos que se procesen por `udev`.

`udev` procesa todos los ficheros terminados en `".rules"` por orden alfabético. Es aconsejable que se procesen nuestras reglas particulares antes que las reglas que trae por defecto `udev`, así que es recomendable que tengamos en este directorio los ficheros con los siguientes nombres:

```
10-udev.misReglas.rules
50-cd-alias.rules
50-udev.rules
```

Observad que hemos creado un fichero `10-udev.misReglas.rules`, este se procesará el primero y será donde escribamos nuestras propias reglas. Así no interferimos en los ficheros de configuración propios de `udev`.

En mi caso el fichero `10-udev.misReglas.rules` es muy sencillo, simplemente contiene:

```
BUS="scsi", SYSFS{vendor}="Apple*", SYSFS{model}="iPod*", NAME="ipod%n"
BUS="scsi", KERNEL="sd?[1-5]", NAME="pen%n"
```

Con la primera línea estoy indicando que cuando se conecte un dispositivo `scsi` de fabricante `Apple...` y modelo `iPod...` lo llame `ipod%n`, es decir, lo llamaré `ipod1` (esto es bueno si conectamos más de un `ipod`, siendo los nombres `ipod1`, `ipod2`, ...)

Con la segunda línea estoy indicando que cualquier dispositivo `scsi` que el kernel le de un nombre `sd?[1-5]`, es decir nombre del estilo: `sda1`, `sda2`, `sdb1`, `sdd3`, ... `udev` le va a asignar un nombre `pen1`, `pen2`, ... Con esto consigo poder pinchar cualquier `pendrive` o dispositivo de almacenamiento a través del USB y montarlo como `pen1`, `pen2`, ...

Ahora sólo me falta añadir un par de líneas en el `/etc/fstab` para simplificar aun más las cosas:

```
/dev/udev/pen1 /mnt/pen vfat defaults,user,noauto 0 0
/dev/udev/ipod2 /mnt/ipod vfat defaults,user,noauto 0 0
```

Ahora si conecto mi pendrive o mi ipod (o los dos a la vez) puedo montarlos tranquilamente con

```
mount /mnt/pen
mount /mnt/ipod
```

O haciendo accesos directos en mi escritorio con el KDE, o como me resulte más cómodo. Pero ya no me tendré que preocupar sobre que nombre le dio el kernel al dispositivo (/dev/sda1, /dev/sdd2, ...).

Por supuesto que hay muchas maneras de configurar las reglas. Si queréis más información os remito a:

<http://www.reactivated.net/udevrules.php>

<http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>

5. Conclusiones

Este tutorial no es más que una pequeña introducción al uso de udev, pero creo que algo tan sencillo como lo que se muestra aquí puede simplificarlos en gran medida el día a día con nuestros dispositivos USB. Así que espero que os sea de utilidad.

Saludos a todos, y hasta el próximo tutorial.

6. Sobre el autor

Alejandro Pérez García

Dir. Implantación y Rendimiento

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L.

<http://www.autentia.com>

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con



Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Acceso seguro a CVS a través de SSH](#)

[Administrar UNIX mediante Webmin](#)

[Como crear un Cron en Unix](#)

Descripción

Os mostramos como securizar los accesos a CVS a través de SSH, utilizando herramientas gratuitas

En este artículo os mostramos como administrar una máquina UNIX, mediante un interfaz web, con Webmin

Cuando desarrollamos aplicaciones, numerosas veces defemos programar tareas que se ejecuten periódicamente. Os enseñamos a hacerlo en Unix

[Upload de ficheros en Java](#)

Os mostramos como enviar ficheros a un servidor Web y manipularlos en un servlet en el servidor, gracias a APIs de apache

[MySQL en Windows](#)

MySQL es una de las principales bases de datos "gratuitas" que podemos encontrar en Internet. En este tutorial aprendereis a instalarlo en Windows

[Framework desarrollo eclipse](#)

Aquí os mostramos algunas de las características de Eclipse

[Evitar doble-click en JSPs y Struts](#)

Os mostramos como construir unas librerías de TAGs para evitar el problema de doble-click en aplicaciones JSP y como se soluciona (qué teneis que hacer) con el framework Struts

[Integración de IPlanet y Weblogic](#)

Cesar Crespo nos muestra como instalar e integrar IPlanet y Weblogic

[Activar SSL en IIS](#)

Os mostramos como activar el soporte de https en IIS, creando vuestros propios certificados autofirmados, usando OpenSSL

[Compartir impresoras y ficheros con Linux](#)

Cesar Crespo Martín y Alejandro Perez García nos enseñan como realizar la compartición de impresoras y ficheros con Linux, CUPS y SAMBA con clientes Windows.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600