# 背景

Linux用户常常会很难鉴别同一类型的设备名，比如eth0，eth1，sda，sdb等等。通过观察这些设备的内核设备名称，用户通常能知道这些是什么类型的设备，但是不知道哪一个设备是他们想要的。

例如，在一个充斥着本地磁盘和光纤磁盘的设备名清单 (/dev/sd\*) 中，用户无法找到一个序列号为“35000c50000a7ef67”的磁盘。在这种情况下，**udev就能动态地在/dev目录里产生自己想要的、标识性强的设备文件或者设备链接，以此帮助用户方便快捷地找到所需的设备文件**。

# 概述

## 定义

udev（u是指user space，dev是指device）是Linux2.6内核里的一个功能，它替代了原来的devfs，成为当前Linux默认的设备管理工具。

udev以守护进程的形式运行，**通过侦听内核发出来的uevent来管理/dev目录下的设备文件**。不像之前的设备管理工具，**udev在用户空间 (user space) 运行**，而不在内核空间 (kernel space) 运行。

## 优点

我们都知道，所有的设备在Linux里都是以设备文件的形式存在。在早期的Linux版本中，/dev目录包含了所有可能出现的设备的设备文件。很难想象Linux用户如何在这些大量的设备文件中找到匹配条件的设备文件。现在udev只为那些连接到Linux操作系统的设备产生设备文件。并且udev能通过定义一个udev规则 (rule) 来产生匹配设备属性的设备文件，这些设备属性可以是内核设备名称、总线路径、厂商名称、型号、序列号或者磁盘大小等等。

动态管理：当设备添加/删除时，udev 的守护进程侦听来自内核的uevent，以此添加或者删除/dev下的设备文件，所以udev只为已经连接的设备产生设备文件，而不会在/dev下产生大量虚无的设备文件。

自定义命名规则：通过Linux默认的规则文件，udev在/dev/里为所有的设备定义了内核设备名称，比如/dev/sda、/dev/hda、/dev/fd等等。由于udev是在用户空间 (user space) 运行，Linux用户可以通过自定义的规则文件，灵活地产生标识性强的设备文件名，比如/dev/boot\_disk、/dev/root\_disk、/dev/color\_printer等等。

设定设备的权限和所有者/组：udev可以按一定的条件来设置设备文件的权限和设备文件所有者/组。在不同的udev版本中，实现的方法不同，在“如何配置和使用udev”中会详解。

# 原理

## 命名规则

udev的命名规则保存在/etc/udev/rules.d中，脚本按照文件的编号顺序，由小到大执行。当规则匹配时，将不再继续向下执行。

下面以Turbolinux 10为例，编写一个U盘命名规则。

# vi /etc/udev/rules.d/05-users.rules

根据硬件的序列号(serial)编写规则，在文件中添加下面内容：

BUS="usb", SYSFS{serial}="033446747503", KERNEL="sd?1", NAME="bruce-%k", SYMLINK="bruce3-%n"

BUS 表示设备使用的总线。

SYSFS 表示指定硬件信息中，我们需要使用的哪一项值，这里我们使用U盘的序列号作为识别。

KERNEL 表示内核对这个设备进行的命名，内核对U盘的命名为sda1，sdb1…，所以这里我们使用”sd?1”。

NAME 设备名,即我们要改变的设备名称.我们使用bruce-%k.%k代表内核对这个设备的命名.

SYMLINK 表示建立一个连接,连接到这个设备上.这个连接的名称是”bruce3-%n”.%n代表kernel号，即设备名后面的数字。

现在将U盘拔出,然后再重新连接到计算机上：

# ls /dev/sd\* -hl

brw-rw---- 1 root disk 8, 0 Aug 17 20:11 /dev/sda

# ls /dev/bruce\* -hl

lrwxrwxrwx 1 root root 10 Aug 17 20:11 /dev/bruce3-1 -> bruce-sda1

brw------- 1 root root 8, 1 Aug 17 20:11 /dev/bruce-sda1

可以看到，U盘设备名变成了bruce-sda1，并且有一个bruce3-1的连接指向它。

## udev工作原理

当系统添加设备时，/sys目录下会生成设备的相应信息，udev就是通过读取/sys下的信息来识别硬件设备的。

udev有3部分组成：namedev，ibsysfs，udev

namedev负责解析/etc/udev/rules.d中的命名规则，遇到匹配规则后，会返回一个设备名。

libsysfs负责读取/sys下的硬件信息。

udev程序负责协调namedev和libsysfs，来完成指定的命名策略。当有设备添加或移出时，/sbin/hotplug由内核调用。提供相应设备的特征信息，如:设备类型，sysfs中的哪个设备等。udev获得这些信息，然后调用namedev，为设备指定一个名称。如果这是一个已经增加的新设备，则udev使用libsysfs来获得设备的主从设备号，然后在/dev下建立设备文件。如果这个设备已经移出，则将从/dev下删除它的设备文件。

# 使用