# 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

udev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。

使用udev的好处：

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

# 使用