“Modern” Linux Device Management

Bonnie King

Fermi Linux User Group

October 31 2012

udev?

- Device manager for the Linux kernel
- Replaces **/dev, devfs, hotplug**...
- Depends on **sysfs**
- Uses **netlink** to send messages

.../dev?

- Provided a file-like interface (nodes) to devices
- Names created at install
- Software interacts with driver with standard I/O system calls
- Special files:
  - Block files (seekable)
  - Character files (sequential)
  - Pseudo-devices (`/dev/null`, `/dev/urandom`)

problems

- It is a dir created by `makedev (mknod)`: full of files and directories, (usually) signifying nothing.
- Populated with everything that may or may not be present in the system
- No persistent naming
- Ran out of major/minor number pairs

devfs (2.4.10)

- Virtual filesystem
- Files created by calls by device driver
- Problems with persistent naming

udev (2.6)

Takes information exported by **sysfs**, and does stuff based on rules provided by user

udev system

- libudev
- udevd
- udevadm

sysfs (2.6)

VFS that is like (parts of) procfs, but not a complete mess.

- `/devices`
- `/bus/`

  `/bus/devices`

  `/bus/drivers`
- `/class`
- `/block`

udevd

- Runs as a daemon and listens for uevents from the kernel (**netlink** socket, asynchronous [can be queued])
- Dynamically populates the `/dev` directory
- Can create device nodes and load drivers according to **udev** rules

uevent

- message sent from kernel when device is detected (over **netlink**)
- contains MODALIAS and other info
- **udev** triggers module loading/arbitrary stuff defined in rules

rules

- Read from `/etc/udev/rules.d/*` by **udevd**, stored in memory
- Will update when rules are edited (**inotify**)
- `udevd` uses data from kernel via **netlink socket** and matches it to rules

udevadm

Get info from udevadm:

```
$ udevadm info -a -p /sys/class/net/eth0/
```

Show messages:

```
$ udevadm monitor –env
```

Show all kinds of info :

```
$ udevadm info -a -p {device path}
```

rules

```
# console

KERNEL=="console", MODE="0600", OPTIONS="last_rule"


# serial devices

SUBSYSTEM=="usb*", ATTRS{product}=="[Tt][Rr]*560",
  OWNER=”bonniek”, GROUP=”engineering”, MODE=”660”,
  SYMLINK+="tricorder",
  RUN=”/usr/bin/sync_tricorder.sh”


# serial port

  KERNEL=="ttyS*", GROUP=”uucp”, MODE=”600”
```

HAL/dbus

HAL (deprecated, but being merged into dbus) provides the abstraction layer/interface between udev and dbus (in **SL5/6**):

```
[root@gerda rules.d]# cat 90-hal.rules
#pass all events to the HAL daemon
RUN+="socket:@/org/freedesktop/hal/udev_event"
```

udisks/upower

- No need for HAL since disk/power event notifications are handled by udisks/upower (but not in SL yet).
- Have their own dbus interfaces

Resources

http://www.linux.com/news/hardware/peripherals/180950-udev

http://www.freedesktop.org/software/systemd/man/udev.html

http://www.reactivated.net/writing_udev_rules.html