2025/12/02 19:24 1/6 Informationsbeschaffung

Udev ist ein System zur überwachung von Hotplug-fähigen geräten. Bekommt udev Informationen über ein neues Gerät, wertet es diese anhand frei konfigurierbarer Regeln aus und gibt dem Gerät einen Namen

Ab Ubuntu 10.04 werden wohl auch Eingabegeräte (Maus, Tastatur, Grafiktablet) über udev erkannt (nicht mehr über HAL).

Die Regeln für die Erkennung der Geräte wird in /etc/udev/rules.d/ abgelegt. In diesem Verzeichnis sind für die verschiedensten Geräte (USB, PCMCIA, eSATA, ...) verschieden Konfigurationsfiles hinterlegt (Bei der einen Distribution mehr, bei der anderen weniger).

Sollte man mal in den Genuss kommen, selbst udev-Regeln zu erstellen, gibt es hier ein kleines Tutorial.

Informationsbeschaffung

Zuerst sollte man so viele Informationen wie möglich über das Gerät sammeln. Unter anderem sind Vender- und DevicelD, sowie Seriennummern sehr hilfreich. Je mehr Informationen voliegen, desto eindeutiger kann das Gerät erkannt werden. Bestes Beispiel: Man hat zwei USB-Sticks des selben Herstellers, der selben Baureihe. Hier ist es nicht möglich, nur über Vendor- und DevicelD eindeutig den einen oder anderen Stick zu identifizieren.

USB-Geräte

Methode 1

Bei USB-Geräten ist es noch relativ einfach, Informationen zu sammeln. ein simples

lsusb

bringt hier schon mal die Vendor- und DevicelD.

Bus 002 Device 011: ID 13fe:1d00 Kingston Technology Company Inc. DataTraveler 2.0 1GB/4GB Flash Drive / Patriot Xporter 4GB Flash Drive

Mehr Informationen bringt da schon

```
lsusb -v > ~/full lsusb
```

Da dies sehr viele Informationen bringt, wird es in eine Datei umgeleitet. Sucht man jetzt nach der Device-ID, kommt man zu weiteren Informationen

Bus 002 Device 009: ID 13fe:1d00 Kingston Technology Company Inc. DataTraveler 2.0 1GB/4GB Flash Drive / Patriot Xporter 4GB Flash Drive Device Descriptor:

bLength 18 bDescriptorType 1

```
bcdUSB
                        2.00
  bDeviceClass
                           0 (Defined at Interface level)
  bDeviceSubClass
                           0
  bDeviceProtocol
                           0
  bMaxPacketSize0
                          64
  idVendor
                      0x13fe Kingston Technology Company Inc.
                      0x1d00 DataTraveler 2.0 1GB/4GB Flash Drive / Patriot
  idProduct
Xporter 4GB Flash Drive
  bcdDevice
                        1.10
  iManufacturer
                           1 Kingston
  iProduct
                           2 DataTraveler 2.0
  iSerial
                           3 5B7415A30A98
  bNumConfigurations
                           1
  Configuration Descriptor:
    bLength
                             9
                             2
    bDescriptorType
                            32
    wTotalLength
    bNumInterfaces
                             1
    bConfigurationValue
                             1
    iConfiguration
                             0
    bmAttributes
                          0x80
      (Bus Powered)
    MaxPower
                           200mA
    Interface Descriptor:
                               9
      bLength
      bDescriptorType
                                4
      bInterfaceNumber
                               0
      bAlternateSetting
                               0
      bNumEndpoints
                               2
      bInterfaceClass
                               8 Mass Storage
      bInterfaceSubClass
                               6 SCSI
                              80 Bulk (Zip)
      bInterfaceProtocol
      iInterface
                               0
      Endpoint Descriptor:
                                  7
        bLength
        bDescriptorType
                                  5
        bEndpointAddress
                              0x81
                                     EP 1 IN
        bmAttributes
                                  2
          Transfer Type
                                     Bulk
          Synch Type
                                     None
          Usage Type
                                     Data
        wMaxPacketSize
                            0x0200
                                     1x 512 bytes
        bInterval
                                  0
      Endpoint Descriptor:
                                  7
        bLength
        bDescriptorType
                                  5
        bEndpointAddress
                                     EP 2 OUT
                              0x02
        bmAttributes
                                  2
          Transfer Type
                                     Bulk
          Synch Type
                                     None
          Usage Type
                                     Data
```

https://wiki.da-checka.de/ Printed on 2025/12/02 19:24

2025/12/02 19:24 3/6 Informationsbeschaffung

| wMaxPacketSize | 0x0200 | 1x 512 bytes |
|----------------|--------|--------------|
| bInterval | 0 | |

Hier kann man dann auch die Seriennummer, Produktname und Hersteller auslesen.

Aus diesen Informationen kann man dann schon eine udev-Regel bauen.

Methode 2

Noch einfacher ist es, sich die Daten per udevadm zu besorgen. Großer Vorteil ist, dass die Einträge ATTRS{...} schon dabei sind.

Hier sollte man nach dem anstecken mal dmesg aufruften und schauen, unter welchem Device der USB-Stick ansprechbar ist. In meinem Fall ist das /dev/sdc1.

Jetzt per udevadm die Informationen anzeigen lassen

```
udevadm info --query=all --attribute-walk --name=/dev/sdc1
```

Jetzt noch nach der VenderlD suchen und schon hat man fertige Attribute mit Bezeichner. Diese kann man einfach in die regeldatei kopieren

```
looking at parent device
'/devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.1':
   KERNELS=="2-1.1"
   SUBSYSTEMS=="usb"
   DRIVERS=="usb"
   ATTRS{configuration}==""
   ATTRS{bNumInterfaces}==" 1"
   ATTRS{bConfigurationValue}=="1"
   ATTRS{bmAttributes}=="80"
   ATTRS{bMaxPower}=="200mA"
   ATTRS{urbnum}=="904"
   ATTRS{idVendor}=="13fe"
   ATTRS{idProduct}=="1d00"
   ATTRS{bcdDevice}=="0110"
   ATTRS{bDeviceClass}=="00"
   ATTRS{bDeviceSubClass}=="00"
   ATTRS{bDeviceProtocol}=="00"
   ATTRS{bNumConfigurations}=="1"
   ATTRS{bMaxPacketSize0}=="64"
   ATTRS{speed}=="480"
   ATTRS{busnum}=="2"
   ATTRS{devnum}=="15"
   ATTRS{version}==" 2.00"
   ATTRS{maxchild}=="0"
   ATTRS{quirks}=="0x0"
   ATTRS{authorized}=="1"
   ATTRS{manufacturer}=="Kingston"
   ATTRS{product}=="DataTraveler 2.0"
```

ATTRS{serial}=="5B7415A30A98"

udev-Regel bauen

Eine udev-Regel ist einfach eine filterung von verschieden Eigenschaften eines Geräts.

Unser Beispiel-USB-Stick würde schon mit folgenden Informationen erkannt werden

98-usbmount.rules

```
ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00"
```

da es aber tausende USB-Sticks diesen Typs gibt, sollte man hier noch die Seriennummer mit einbeziehen.

98-usbmount.rules

```
ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00", ATTRS{serial}=="5B7415A30A98"
```

Zur Übersicht ist hier eine Tabelle mit häufig genutzen Identifizierungsvariablen

| Name | Bedeutung | |
|--------------|----------------------------------------------------------------------------------------------------------------------------------|--|
| SUBSYSTEM | Gerätetyp, zum Beispiel usb_device, block, und so weiter | |
| BUS | Bussystem des Geräts, zum Beispiel ieee1394 oder usb | |
| ID | Geräte-ID (bezogen auf den Bus) | |
| NAME | Name des Netzwerkgeräts (eth0, eth1 etc.) oder der Gerätedatei in /dev | |
| KERNEL | Gerätename laut Kernel | |
| SYSFS{Datei} | Verwendet Informationen bestimmter Dateien aus /sys (enthalten Angaben zur Hardware); pro Regel maximal fünf SYSFS-Konstanten | |

Jetzt hat man das Gerät eindeutig identifiziert. doch <u>was</u> soll mit dem Gerät <u>wann</u> geschehen. Dazu muss die Regel erweitert werden.

Wann

Hier greift die Variable ACTION

Soll die Regel beim Anstecken ausgeführt werden, muss es folgendermaßen erweitern werden

98-usbmount.rules

```
ACTION="add", ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00", ATTRS{serial}=="5B7415A30A98"
```

https://wiki.da-checka.de/ Printed on 2025/12/02 19:24

2025/12/02 19:24 5/6 Informationsbeschaffung

Soll die Regel beim **Ab**stecken ausgeführt werden, muss es folgendermaßen erweitern werden

98-usbmount.rules

```
ACTION="remove", ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00", ATTRS{serial}=="5B7415A30A98"
```

was

Hier greifen die Variablen RUN und SYMLINK.

SYMLINK bedeutet einfach, dass auf das erkannte Gerät (z. B. als /dev/sdc) ein symbolischer Link angelegt wird. Zum Aufruf wird die Regel folgendermaßen erweitert.

98-usbmount.rules

```
ACTION="add", ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00",
ATTRS{serial}=="5B7415A30A98", SYMLINK+="datenstick"
```

Jetzt kann man den Stick immer per /dev/datenstick mounten und damit arbeiten

RUN bedeutet, dass man ein Programm, dass angegeben wird, ausführen lassen kann. Das beste Beispiel ist hier eine BackupFestplatte. Sollte diese BackupPlatte angeschlossen werden soll ein Backup-Skript ausgeführt werden. Zum Aufruf muss die Regel folgendermaßen erweitert werden.

98-usbmount.rules

```
ACTION="add", ATTRS{idVendor}=="13fe", ATTRS{idProduct}=="1d00", ATTRS{serial}=="5B7415A30A98", SYMLINK+="datenstick", RUN="/usr/local/bin/backup.sh"
```

Zur Information ist hier eine Tabelle mit den meistgenutzten Aktionsvariablen

| Name | Bedeutung | |
|---------|----------------------------------------------------------------------------------|--|
| ACTION | Ereignis, entweder add oder remove | |
| RUN | führt das angegebene Programm aus. | |
| SYMLINK | legt einen symbolischen Link an, der auf den tatsächlichen Devicenamen verweist. | |
| OWNER | Ändert den Besitzer des Devices (Benutzername oder UID) | |
| GROUP | Ändert die Gruppe des Devices (Name oder GID) | |
| MODE | Ändert die Zugriffsrechte des Devices (Oktalzahlen) | |

Last update: 2012/02/08 12:34

From:

https://wiki.da-checka.de/ - PSwiki

Permanent link:

https://wiki.da-checka.de/doku.php/wiki/dienste/udev?rev=1328700859

Last update: 2012/02/08 12:34



https://wiki.da-checka.de/ Printed on 2025/12/02 19:24