

● Apparaten (devices) in Linux ●

Hans Lunsing

Aan Linux-gebruikers die wat meer van de werking van dit besturingssysteem willen weten, laat ik in dit artikel zien hoe in Linux de toegang tot apparaten is geregeld.

Daarbij gaat het natuurlijk in de eerste plaats om fysieke apparaten (hardware) zoals een harde schijf of een muis, maar ook om virtuele apparaten (software), zoals een terminal die je gewoon op je scherm kunt zien en gebruiken, of een toevalsgenerator.

Alles is een bestand

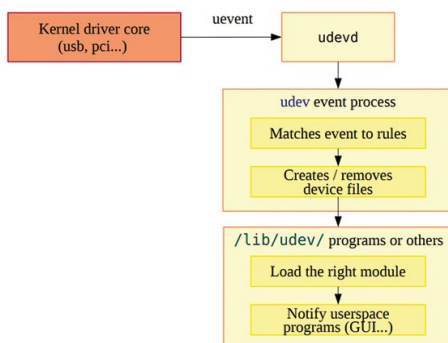
In een Linux-systeem wordt alles wat deel uitmaakt van het systeem aangeduid als een bestand. Het gaat daarbij niet alleen om de programma's, documenten, afbeeldingen, enz. die op de harde schijf staan, maar ook om door de Linux-kernel¹ gebruikte gegevensstructuren, gegevens over alle in het systeem draaiende processen en - en daar gaat het hier om - de aan de computer gekoppelde fysieke en virtuele apparaten. Dit alles wordt bij de start van het systeem ondergebracht in virtuele bestandssystemen², die aan het reguliere bestandssysteem op de harde schijf worden gekoppeld via enkele daarvoor bedoelde speciale mappen.

De gegevensstructuren van de kernel komen in de map `/sys`, de draaiende processen in `/proc`, en de apparaten (devices) in `/dev`. Terwijl die mappen deel uitmaken van het reguliere bestandssysteem op de harde schijf, krijgen ze pas een relevante inhoud wanneer het Linux-systeem wordt gestart³.

Alles in een Linux-systeem is toegankelijk als een bestand in het bestandssysteem, en dat houdt in dat voor heel veel verschillende dingen dezelfde tools kunnen worden gebruikt. Dat maakt dit concept - *alles is een bestand* - uiterst krachtig en flexibel.

Je vraagt je natuurlijk af hoe het dan in Windows gaat. Wel, ook Windows biedt vanzelfsprekend toegang tot apparaten, maar daarin gaat het via apparaatobjecten in een objectmap met de naam `\device`⁴ die niet is gekoppeld aan het reguliere bestandssysteem: de toegang tot die apparaatobjecten verschilt van de toegang tot normale bestanden. Hierna gaan we in op het beheer en de inhoud van de apparatenmap `/dev`.

udev summary - typical operation



22

Free Electrons. Kernel, drivers and embedded Linux development, consulting, training and support. <http://free-electrons.com>

Beheer van de apparatenmap `/dev`

De map `/dev` wordt tijdens de start van het Linux-systeem door de apparaatbeheerder (device manager) `udev` gevuld met zgn. apparaatbestanden voor alle feitelijk aanwezige apparaten in een dynamisch virtueel bestandssysteem met de naam `devtmpfs` (dit staat voor devices temporary file system). `Udev` handelt op de achtergrond verder alles af wat er moet gebeuren bij het toevoegen of verwijderen van apparaten, met inbegrip van het bijhouden van `devtmpfs` en van het laden en ontladen van drivers en mogelijk benodigde firmware.



De configuratie van `udev` bestaat vnl. uit een hardware-database en regels voor het aan- en afkoppelen van allerlei verschillende apparaten. De standaardconfiguratie staat in de map `/usr/lib/udev`, terwijl de plek voor een aangepaste

configuratie `/etc/udev` is. In beide gevallen zijn de regels te vinden in de deelmap `rules.d`. De aangepaste configuratie heeft voorrang op de standaardconfiguratie, zodat het mogelijk is de afhandeling van bepaalde apparaten te veranderen door de daartoe gewijzigde regels te plaatsen in `/etc/udev/rules.d`. Welke apparaatbestanden bij een apparaat horen wordt ook door zo'n regel bepaald. Bij `udev` behoort een beheersapplicatie genaamd `udevadm`, met behulp waarvan je onder meer informatie over door `udev` beheerde apparaten kunt krijgen.

Dat gaat met de opdracht:

`udevadm info <pad naar het apparaatbestand>`



Om alle mogelijke informatie over bijvoorbeeld de harde schijf *sda* te krijgen geef je de opdracht:

```
udevadm info /dev/sda
```

De opdracht ‘*udevadm info*’ heeft ook nog een aantal andere opties, waarvan je een lijstje kunt krijgen met de opdracht:

```
udevadm info -help
```

Naam en nummer van apparaatbestanden

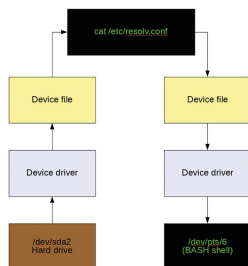
De apparaatbestanden hebben natuurlijk allemaal een passende naam, maar daarnaast ook nog een identificatienummer, bestaande uit een primair en een secundair deel, gescheiden door een komma. Het primaire deel geeft het **soort** apparaat aan, terwijl het secundaire deel het **specifieke** apparaat binnen de soort aanduidt.

Zo heeft de eerste (SATA) harde schijf, *sda*, het identificatienummer (8,0), terwijl de partities van die schijf, genummerd *sda1*, *sda2*, enz. de nummers (8,1), (8,2), enz. hebben. De limiet voor partitienummering is 15. Van de tweede harde schijf, *sdb*, is het identificatienummer (8,16), terwijl de partities *sdb1*, *sdb2*, enz. genummerd zijn vanaf (8,17) enz. Een complete lijst van identificatienummers en namen staat op de website van de Linux-kernel www.kernel.org⁵. In sommige niet-standaardgevallen kan een Linux-distributie hier wel eens van afwijken.

Types apparaten

Er worden twee types apparaten onderscheiden: **blok-** en **teken-** (block- en character-) apparaten.

Een blokapparaat geeft gebufferde toegang tot het apparaat. Zo'n apparaat wordt bloksgewijs gelezen of beschreven. Een blok kan elke willekeurige grootte hebben, met zowel eenvoudige tekens (characters of bytes) als meerdere tekens tegelijkertijd. Vaak gaat het om 256 bytes. De belangrijkste voorbeelden zijn harde schijven en cd's en dvd's.



Een tekenapparaat geeft ongebufferde, directe toegang tot het apparaat. Zo'n apparaat wordt per afzonderlijk teken (character of byte) gelezen of beschreven. Voorbeelden van tekenapparaten zijn de muis, de terminals en pseudo-terminals.

Inhoud van de apparatenmap

De apparatenmap */dev* bevat apparaatbestanden voor alle aangesloten apparaten. Je vindt er bijvoorbeeld (tussen haakjes) het primaire identificatienummer.

- *kmem* (1): toegang tot het virtuele kernelgeheugen.
- *mem* (1): toegang tot het fysieke geheugen.
- *port* (1): toegang tot I/O (input/output) poorten.
- *loop* (7): een zgn. loop device, genummerd vanaf 0, voor het koppelen van bestanden met een intern bestandssysteem, zoals iso-bestanden⁶, aan het bestandssysteem.
- *sd* (8): harde schijven met SATA-aansluiting *sda*, *sdb*... enz., en de genummerde partities op die schijven *sda1*, *sda2*..., *sdb1*, *sdb2*..., enz.

- *md* (9): metadiskgroep (RAID7), genummerd vanaf 0
- *psaux* (10): PS/2-muis
- *sr* (11): SATA-cd/dvd, genummerd vanaf 0 (verouderd; wordt vervangen door *scd*)
- *tty* (4): de virtuele terminals *tty1*, *tty2*..., enz.

Bijzondere virtuele apparaten zijn:

- *null* (1), de ‘bittenbak’, die alles wat ernaar toe gestuurd wordt in het niets laat verdwijnen,
- *random* (1), dat een stroom willekeurige bytes levert,
- *urandom* (1), dat eveneens een stroom willekeurige bytes levert, sneller, maar minder veilig dan */dev/random*
- *zero* (1), dat een stroom nulbytes produceert.

Naast apparaten bevat */dev* ook symbolische links (symlinks) naar apparaatbestanden. Zo'n symlink biedt een alternatieve representatie voor een apparaat.

Een voorbeeld van een symlink is */dev/cdrom*, die verwijst naar */dev/sr0*, het cd- of dvd-station als blokapparaat. Ook zijn er enkele symlinks naar andere virtuele delen van het bestandssysteem in de mappen */proc* en */run*. Het gaat dan om bijvoorbeeld de Linux-kernel (*/dev/core*) en standaard input, output en error (*/dev/stdin*, */dev/stdout* en */dev/stderr*)

Een aantal apparaatbestanden en symlinks is ondergebracht in deelmappen van */dev*. Voorbeelden van deelmappen met apparaten zijn:

- *cpu* met de centrale processoreenheden,
- *input* met invoerapparaten zoals muis en toetsenbord,
- *pts* met genummerde pseudo-terminals⁹,
- *snd* met geluidsapparaten,
- *usb* met usb-interfaces.

Een belangrijk voorbeeld van een deelmap met symlinks is:

- *disk* met symlinks van schijven en hun partities in mappen die elk een mogelijke identificatie van die schijven en partities weergeven. Het gaat om de mappen:
 - *disk/by-id* met unieke namen voor schijven en hun partities, gebaseerd op hun seriële nummer,
 - *disk/by-path* met het kortste fysieke pad naar de hardware,
 - *disk/by-partuuid*: met UUID's¹⁰ voor partities (alleen op GPT-schijven),
 - *disk/by-partlabel*: met de aan partities toegekende labels.
 - *disk/by-uuid*: met UUID's voor het bestandssysteem op partities,
 - *disk/by-label*: met de aan het bestandssysteem op partities toegekende labels.

Er zijn twee deelmappen met symlinks naar alle blokapparaten en (bijna) alle tekenapparaten:

- *block*: bevat symlinks naar alle blokapparaten
- *char*: bevat symlinks naar bijna alle tekenapparaten (met als belangrijkste uitzondering van de pseudo-terminals onder *pts*).

De naam van deze symlinks is het identificatienummer van het apparaat, waarbij het primaire en secundaire deel worden gescheiden door een dubbele punt.

Zo is */dev/block/8:1* een symlink naar *sda1*, de eerste partitie van de eerste (sata) harde schijf, en */dev/char/11:1* een symlink naar */dev/psaux*, de PS/2-muis.

Enkele deelmappen hebben een eigen bestandssysteem. De belangrijkste daarvan is */dev/pts* met het bestandssysteem *devpts* voor pseudo-terminals.

Lijst van de inhoud van de apparatenmap

Een uitgebreide lijst van de inhoud van de map `/dev` krijg je met de opdracht `'ls -l /dev'`. Deze lijst toont van elk item (apparaat, map of symbolische link):

- **Type:** Met een letter die aangeeft of het een blokapparaat (b: block device), tekenapparaat (c: character device), map (d: directory) of symbolische link (l: symbolic link) is. Als het geen van deze is, staat er een streepje.
- **Permissions:** Net zoals voor gewone mappen en bestanden het triplet lees- (r: read), schrijf- (w: write) en uitvoer- (x: execute) permissie voor achtereenvolgens de eigenaar (u: user), groep (g: group) en anderen (o: others).

`rw-rw-r--` bijvoorbeeld betekent dat de eigenaar alle drie de permissies heeft, dat de groep lees- en schrijfpermissie heeft, en dat anderen alleen leespermissie hebben.

In plaats van de `x` voor execute kan er ook een `s` staan (gebruik voor uitvoering het ID van gebruiker dan wel groep) of `t` (vnl. bij mappen: een vlag die aangeeft dat verwijdering van bestanden in die map gerestricteerd is).

Bij symbolische links zijn de aangegeven permissies `'rwxrwx-rwx'` niet relevant omdat voor hen de permissies van het doel van de link van kracht zijn.

- **Eigendom:** Net zoals voor gewone mappen en bestanden de eigenaar-gebruiker en de groep. Van de inhoud van `/dev` is de eigenaar standaard root, maar van pseudo-terminals (in map `pts`) is de eigenaar degene die de pseudo-terminal heeft geopend.

Ook de groep is standaard root, maar is onder meer disk voor harde schijven en loop devices, `cdrom` voor `cd's` en `dvd's`, `tty` voor terminals, en `dialout` voor seriële poorten.

- **Identificatie:** Voor apparaten de primaire en secundaire identificatienummers, gescheiden door een komma. Voor mappen en symbolische links de grootte van hun metadata, net als voor gewone mappen en symbolische links.
- **Datum en tijd:** Net zoals voor gewone mappen en bestanden de datum en de tijd van laatste modificatie. Dat zal voor apparaten normaliter de datum en de tijd zijn waarop het systeem is opgestart (boot).
- **Naam:** Net zoals voor gewone mappen en bestanden de naam van het apparaat, de map of de symbolische link. Bij wijze van voorbeeld volgen hier een aantal regels uit zo'n listing (`ls -l /dev`) zijn:

```
drwxr-xr-x 2 root root      960 aug 7 11:26 block
lrwxrwxrwx 1 root root      3 jul 27 10:04 cdrom -> sr0
drwxr-xr-x 2 root root     4740 aug 7 11:26 char
drwxr-xr-x 8 root root     160 jul 26 07:03 disk
crw-rw-rw- 1 root root      1, 3 jul 26 07:03 null
drwxr-xr-x 2 root root      0 jul 26 07:03 pts
crw-rw-rw- 1 root root      1, 8 jul 26 07:03 random
brw-rw---- 1 root disk      8, 0 jul 26 07:03 sda
brw-rw---- 1 root disk      8, 1 jul 26 07:03 sda1
brw-rw----+ 1 root cdrom    11, 0 jul 27 10:04 sr0
crw-rw-rw- 1 root tty        5, 0 aug 7 16:36 tty
crw--w---- 1 root tty        4, 0 jul 26 07:03 tty0
crw-rw-rw- 1 root root      1, 5 jul 26 07:03 zero
```

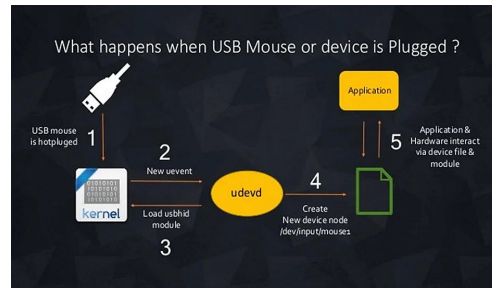
Dat de grootte van de map `pts` als 0 wordt gegeven, is omdat aan die map een eigen bestandssysteem is gekoppeld. Zonder die koppeling is de map leeg, hetgeen de 0 verklaart.

Lezen en schrijven

Apparaatbestanden voor fysieke apparaten kunnen worden

gezien als portalen naar de bijbehorende devicedrivers. Data die door een applicatie of het systeem aan een apparaatbestand worden aangeboden, worden doorgesluist naar de devicedriver, die ze vervolgens naar het fysieke apparaat stuurt. Omgekeerd wordt dezelfde weg afgelegd, van fysiek apparaat via devicedriver en apparaatbestand naar applicatie óf systeem.

Printer



Stel dat een printer aan het systeem is gekoppeld die de naam `/dev/usb/lp0` krijgt (lp van line printer). Dan kan een simpel tekst-

bestand worden afgedrukt met de opdracht:

```
cat [txt] > /dev/usb/lp0
```

waarin `[txt]` staat voor de naam van het tekstbestand. Als de devicedriver in staat is om rechtstreeks pdf-bestanden door de printer af te laten drukken, kan een pdf met de volgende opdracht worden afgedrukt:

```
cat [pdf] > /dev/usb/lp0
```

waarin `[pdf]` de naam van het af te drukken pdf-bestand is.

Virtuele terminals

Je kunt ook teksten naar virtuele terminals (`tty`) sturen. Druk op `Ctrl-Alt-F2` om naar terminal 2 (`tty2`) te gaan en log in als gewone gebruiker. Dan kun je zien in welke terminal je bent door de opdracht `'tty'` te geven. Die zal in dit geval als antwoord `'tty2'` geven. Ga nu naar terminal 3 door op `Ctrl-Alt-F3` te drukken en log daar ook in als dezelfde gebruiker. Geef nu de volgende opdracht:

```
echo "Hallo wereld!" > /dev/tty2
```

Ga nu terug naar terminal 2. Je zult zien dat `'Hallo wereld!'` (zonder de aanhalingstekens) op deze terminal wordt getoond. Druk na het lezen van zo'n bericht altijd even op de Enter-toets om de opdrachtprompt terug te krijgen. Hetzelfde kun je doen met een pseudo-terminal in de grafische omgeving. Stel dat de grafische omgeving wordt uitgevoerd in terminal 1. Ga er dan met `Ctrl-Alt-F1` naar terug, open zo'n pseudo-terminal (bijvoorbeeld Gnome terminal in Linux Mint) en kijk met de opdracht `tty` welke naam hij heeft. Stel dat het `/dev/pts/3` is. Je kunt nu vanuit deze terminal een bericht sturen naar terminal 2:

```
echo Hallo > /dev/tty2
```

In terminal 2 (`Ctrl-Alt-F2`) zul je zien dat het bericht is aangekomen. Stuur nu een bericht terug naar pseudo-terminal 3:

```
echo Leuk om van je te horen > /dev/pts/3
```

en je zult zien dat het bericht in de pseudo-terminal is aangekomen.

Je kunt natuurlijk ook een compleet tekstbestand naar de andere terminal sturen, bijvoorbeeld ¹¹:

```
cat [txt] > /dev/pts/3
```

Schijven en partities

Direct lezen van en schrijven naar schijven en hun partities kan ook. Zo is het bijvoorbeeld mogelijk om van een in MS-

DOS gepartitioneerde schijf (met een MBR ofwel Master Boot Record) de MBR te lezen. Het gaat daarbij om de eerste 512 bytes van de schijf¹²:

```
sudo dd if=/dev/sda of=sda_mbr bs=512 count=1
```

Hier wordt *sudo* gebruikt omdat alleen *root* en leden van de groep *disk* van en naar harde schijven en partities mogen lezen en schrijven. Het bestand *sda_mbr* bevat nu de MBR van schijf *sda*. De 446 bytes 0 t/m 445 bevatten de opstartcode, de 64 bytes 446 t/m 509 de partitietabel en de bytes 510 en 511 het bootrecord-kenteken (*boot signature*) 0x55AA. Er kan natuurlijk ook van moderne GUID gepartitioneerde schijven worden gelezen. De eerste 512 bytes bevatten de 'Protective MBR' voor gebruik door BIOS in plaats van UEFI. Daarna volgt de primaire partitietabel (de 512+128*128 bytes 512 t/m 17407), bestaande uit een kop (de 512 bytes 512 t/m 1023) en 128 partitie-ingangen van elk 128 bytes (128*128 bytes 1024 t/m 17407). Daaropvolgend de partities. Aan het eind van de schijf bevindt zich de secundaire partitietabel, die dient als backup van de primaire. De primaire partitietabel kun je lezen met het commando:

```
sudo dd if=/dev/sda of=sda_ppt ibs=128 skip=4 count=132
```

De output van dit commando is:

```
132+0 records gelezen
33+0 records geschreven
16896 bytes (17 kB, 16 KiB) gekopieerd,
0,000569334 s, 29,7 MB/s
```

Na afloop bevat het bestand *sda_gpt* de partitietabel, vanzelfsprekend in een voor mensen niet-leesbare vorm. Je ziet dat er, zoals gevraagd, 132 records zijn gelezen, maar dat er maar 33 zijn geschreven.

Dat is zo omdat de overige records alleen met *nulbytes* gevuld zijn. Het totaal aantal naar *sda_gpt* geschreven bytes klopt wel. Je kunt de inhoud in hexadecimaal en ASCII-formaat bekijken met het programma *hexdump*¹³:

```
hexdump -C sda_gpt
```

Opeenvolgende nulbytes worden verkort aangegeven met een sterretje (*).

Schrijven raad ik beslist af, omdat dat gebeurt zonder rekening te houden met de gegevensstructuren op de schijf, dan wel de partitie. De kans is bijna 100% dat de schijf, danwel de partitie, niet of niet volledig meer leesbaar is.

Dat kan natuurlijk in bijzondere gevallen wel de bedoeling zijn, bijvoorbeeld wanneer een schijf of partitie volledig leeg gemaakt moet worden. Om bijvoorbeeld schijf *sdC* te legen geef je de opdracht:

```
sudo dd if=/dev/zero of=/dev/sdC bs=2048
```

LET OP: de hele schijf wordt dan met *nulbytes* overschreven. Voor nieuw gebruik moet de schijf eerst weer worden geformatteerd. Het is natuurlijk ook mogelijk om alleen één partitie leeg te maken. De gegevensstructuren op de schijf buiten die partitie worden dan niet overschreven.



Geheugen

Lezen van en schrijven naar het geheugen */dev/mem* kan ook, maar alleen van en naar die delen van het geheugen waar je als gebruiker toegang toe hebt. En ook in dit geval raad ik schrijven absoluut af. Als *root* kun je bijvoorbeeld de eerste 100 blokken van elk 2048 bytes lezen en naar het bestand '*geheugen*' schrijven met de opdracht:

```
sudo dd if=/dev/mem of=geheugen bs=2048 count=100
```

Na uitvoering wordt de volgende output getoond:

```
100+0 records gelezen
100+0 records geschreven
204800 bytes (205 kB, 200 KiB) gekopieerd,
0,00131184 s, 156 MB/s
en het bestand geheugen bevat nu de eerste 204800 bytes van de inhoud van het geheugen.
```

Uitleiding

Tot zover het verhaal over apparaten (devices) in Linux. Ik hoop dat het je wat inzicht heeft gegeven in een heel belangrijk onderdeel van het bestandssysteem, dat evenwel bij gewoon gebruik van Linux onopgemerkt blijft. Experimenteer maar eens met opdrachten zoals gegeven onder 'Lezen en schrijven', maar wees uiterst voorzichtig met schrijven, zeker naar harde schijven. Gelukkig zijn die voor gewone gebruikers niet toegankelijk, tenzij *sudo* wordt gebruikt. Veel genoegen ermee!

Links

- 1 De kernel is het centrale deel, de kern, van het besturingssysteem.
- 2 Een virtueel bestandssysteem is een bestandssysteem dat niet op de harde schijf staat.
- 3 In Ubuntu en daarop gebaseerde Linux-systemen is de map */dev* in een niet draaiend systeem al wel gevuld met apparaatbestanden voor de meest voorkomende apparaten. Wanneer het systeem start wordt daar een virtueel bestandssysteem overheen gelegd met bestanden voor de actuele apparaten.
- 4 Zie <https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/introduction-to-device-objects>
- 5 Zie <https://www.kernel.org/doc/html/latest/admin-guide/devices.html>
- 6 Een iso-bestand bevat de inhoud van een cd of dvd.
- 7 Redundant array of independent disks: hierbij worden meerdere fysieke harde schijven gecombineerd tot één of meer logische opslageenheden met als doel de veiligheid, snelheid en/of capaciteit te vergroten. Zie https://nl.wikipedia.org/wiki/Redundant_array_of_independent_disks
- 8 Een virtuele terminal is een applicatie die een fysieke terminal simuleert in software. Zo'n terminal draait buiten de grafische omgeving en wordt bereikt met de toetscombinatie Ctrl-Alt-Fx, waarin x het nummer is. Meestal zijn er 6 of 7 terminals geopend. De grafische omgeving draait zelf ook in een virtuele terminal, en wel nummer 7 als die er is, en anders is het 1.
- 9 Een pseudo-terminal emuleert een virtuele terminal, bijvoorbeeld in de grafische omgeving.
- 10 Een UUID is een Universally Unique Identifier. Zie https://nl.wikipedia.org/wiki/Universally_unique_identifier
Voor Windows-partities en de ESP (EFI System Partition) worden afwijkende formaten gebruikt.
- 11 Met het programma *cat* kunnen bestanden sequentieel worden gelezen en naar standard output worden geschreven.
Zie [https://en.wikipedia.org/wiki/Cat_\(Unix\)](https://en.wikipedia.org/wiki/Cat_(Unix))
- 12 Met het programma *dd* kunnen bestanden van of naar elkaar worden gekopieerd.
Zie https://en.wikipedia.org/wiki/Dd_%28Unix%29
- 13 Hexdump is meestal standaard geïnstalleerd. In Ubuntu en afgeleiden (zoals Linux Mint) maakt het deel uit van het pakket *bsdextrautils*.