

Alles is een bestand

Posted 21 Sep 2015 [David BothFeed](#)
Adapted Feb 2017 Henry van Aarssen

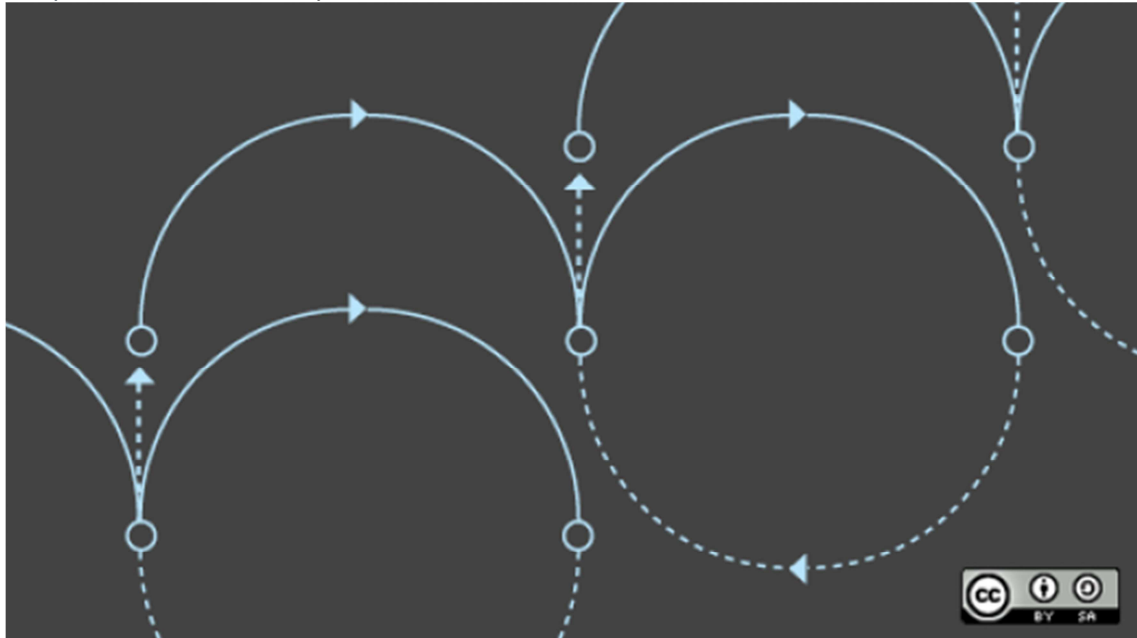


Image by : [opensource.com](#)

We beginnen met een strikvraag: Welk van de volgende zijn bestanden?

- Directories
- Shell scripts
- LibreOffice documents
- Serial ports
- Kernel data structures
- Kernel tuning parameters
- Hard drives
- Partitions
- Logical Volumes (LVM)
- Printers
- Sockets

Misschien geloof je dit niet, maar voor Unix en Linux zijn dit allemaal bestanden. Dat is een van de krachtigste uitgangspunten en maakt het mogelijk om sommige (erg) simpele, maar krachtige manieren te bedenken voor de vele administratieve taken, die anders misschien erg moeilijk of onmogelijk zouden zijn.

Back up the Master Boot Record

Neem bijvoorbeeld de eenvoudige taak om een backup van het Master Boot Record (MBR) van je harde schijf te maken. Voor als ik per ongeluk de MBR of specifiek de partitietabel moet herstellen of hercreëren. Hercreëren is een ramp, maar terugzetten vanaf een kopie is simpel. Linux heeft daarvoor een krachtige GNU tool, dd, die dat en veel meer kan.

De dd staat voor "disk dump," maar voor hen die al wat langer meelopen kennen het als "disk destroyer" omdat, als je niet erg goed oplet, het precies doet wat jij verteld dat het moet doen, fout of niet!!! Dus ook alles vernietigen!!

Het volgende commando maakt een backup van MBR. Het moet gebeuren met root-privilege, omdat niet-root gebruikers geen toegang hebben tot [device files](#) in de /dev map. BS is Block Size en count is het aantal blokken om te lezen van de bron. Dit commando maakt een bestand, myMBR.bak in de /tmp map. Het bestand is 512 bytes groot en bevat de inhoud van je MBR inclusief de bootstrap code en partitie tabel.

```
dd if=/dev/sda of=/tmp/myMBR.bak bs=512 count=1
```

In geval de MBR beschadigd/onbruikbaar zou zijn, dankun je starten van een rescue-disk en het volgende commando gebruiken, die het omgekeerde van wat we net gedaan hebben. Merk op dat het niet nodig is om hier een block size en block count op te geven, omdat het dd commando simpelweg het backup-bestand kopieerd naar de eerste sector van de hard schijf en stopt als het einde van het backupbestand daar is.

```
dd if=/tmp/myMBR.bak of=/dev/sda
```

Het is allemaal onderdeel van het filesystem (bestandssysteem).

Alles op een Linux computer is toegankelijk als een bestand in het filesystem. Belangrijk is de mogelijkheid om [het gebruik van algemene tools voor verschillende dingen mogelijk te maken](#).

Het dd commando kan gebruikt worden om een hele partitie of harde schijf te copieren naar een bestand of een andere harde schijf. In het volgende voorbeeld kopieerd het dd commando data tot het einde van het input device en stopt dan. Let wel op dat de bron (source) kleiner is dan de kopie (destination)!!

```
dd if=/dev/sdf2 of=/dev/sdg3  
dd if=/dev/sda of=/dev/sdg
```

Andere filesystem gereedschappen werken ook. Het `cat` commando, kan bijvoorbeeld gebruikt worden om de inhoud van elk bestand te sturen naar Standard Output. Dat is dus inclusief partitions en complete harde schijven. De output kan doorgestuurd worden naar een bestand.

```
cat /dev/sda1 > partition1.backup
```

Het `cat` commando, is wat minder krachtig dan `dd`. We kunnen bijvoorbeeld niet aangegeven hoeveel dat er gelezen moet worden van de bron.

Here is een interessant experiment ter illustratie van het feit dat alles een bestand is. De meeste Linux distributies beschikken over meerdere virtuele consoles, 1 t/m 7, die je kunt gebruiken om in te loggen op een lokale console-sessie met een shell-interface. Gebruik hiervoor de toestencombinaties `Ctrl-Alt-F1` voor console 1, `Ctrl-Alt-F2` voor console 2, en zo verder.

Druk op `Ctrl-Alt-F2` om over te schakelen naar console 2. In sommige distributies bevat de login-informatie het `tty` (Teletype) device wat gekoppeld is aan die console, maar vele distributies laten dat niet zien! Het zou in elk geval `tty2` moeten zijn omdat je op console2 zit.

Login als een niet-root gebruiker. Gebruik daarna het `who am i` commando—ja exact zo, inclusief de spaties—om te controleren welk `tty` device gekoppeld is aan deze console.

Voordat we ons experiment gaan beginnen, kijken we eerst even naar een listing van de `tty2` en `tty3` devices in `/dev`.

```
ls -l /dev tty[23]
```

Er zijn een groot aantal `tty` devices gedefinieerd, maar we richten ons nu even op de `tty2` en `tty3` devices. Als device files, is er niks apart aan; het zijn normale character type devices. We gaan deze devices gebruiken voor dit. Het `tty2` device is gekoppeld aan virtual console 2 en het `tty3` device is geloppeld aan virtual console 3.

Druk op `Ctrl-Alt-F3` om naar console 3 te schakelen. Log opnieuw in als dezelfde non-root user.

Geef nu het volgende commando op console 3.

```
echo "Hello world" > /dev/tty2
```

Druk Ctrl-Alt-F2 om terug te gaan naar console 2. De woorden "Hello world" (zonder quotes) zijn te zien op console 2.

Dit experiment kan ook uitgevoerd worden met terminal emulators op de GUI desktop. Terminal sessions op de desktop gebruiken zgn. pseudo terminal devices in de /dev boom, zoals /dev/pts/1. Open tweeterminal sessies met Konsole or Xterm. Bepaal welke pseudo-terminals dat ze gebruiken en gebruik de ene om een boodschap naar de andere te sturen.

Ga nu verder met het experiment door het cat commando te gebruiken om de /etc/fstab file op een andere terminal te laten zien.

Een ander interessant experiment is om een bestand rechtstreeks naar de printer te sturen middels het cat command. Er van uitgaand dat de printer verbonden is via /dev/usb/lp0, en dat deze PDF bestanden rechtstreeks kan printen, print het volgende commando een PDF bestand op je printer.

```
cat test.pdf > /dev/usb/lp0
```

Het dd commando kan ook gebruikt worden om een geschikt bestand direct af te drukken op je printer, maar daarvoor moet je printer wat meer kunnen, dus cat geniet de voorkeur.

Implicaties van "alles is een bestand"

De implicaties van "alles is een bestand" reiken veel verder dan we in zo'n kort artikel kunnen weergeven. Je hebt al een paar experimenten kunnen zien. Hieronder volgen nog een paar opties ter illustratie.

1. Clonen van (harde) schijven.
2. Back up partities.
3. Back up van het master boot record (MBR).
4. Instaleren van ISO images naar USB sticks.
5. Communicatie met gebruikers op andere terminals.
6. Print files naar een printer.
7. Het aanpassen van bepaalde bestanden in het /proc pseudo filesystem om parameters te wijzigen in de draaiende kernel.

8. Overschrijf bestanden, partities of zelfs complete harde schijven met willekeurige data of nullen.
9. Doorsturen van ongewenste commando-output naar het /dev/null device waarmee deze output echt weggaat.
10. etc., etc., etc.

Er zijn zoveel mogelijkheden dat de lijst ervan ellenlang zou worden. Het motto is dus “uitproberen”. Een belangrijk hulpmiddel hierbij is altijd het man commando.

Extra informatie (engelstalig)

Voor meer informatie over de /dev/ directory en de devices die je er tegen kunt komen, check dit [article](#) van The Linux Journal. Voor meer specifieke informatie over individuele devices, check [hier](#) en/of [hier](#) bij het [Linux Documentation Project](#) .