

● Waar Linux z'n bestanden laat ●

Hans Lunsing

Als u in Linux een verkenner als Dolphin of Nautilus of een dialoogvenster voor het lezen of bewaren van een bestand opent ziet u mappen met ondoorgrondelijke namen als /dev, /etc en /usr/bin. Niet wat u gewend bent in Windows met z'n 'Program Files', 'Windows' en 'Users'! Dit artikel maakt u wegwijs in de mapstructuur van Linux, zodat /dev, /etc en /usr/bin voor u geen geheimen meer hebben.)

Waar Windows z'n bestanden laat, zullen de meesten van u wel in meer of mindere mate weten. De Windows spullen vindt u onder 'C:\Windows', geïnstalleerde programma's onder 'C:\Program Files', en uw eigen instellingen, documenten, foto's, enzovoort, vindt u onder 'C:\Users\w gebruikersnaam'. Andere harde schijven of hardeschijfpartities hebben eigen driveletters D:, E:, enz., net zoals aangekoppelde USB-sticks en harde schijven, DVD-spelers en wat dies meer zij.

Hoe zit dat nu in Linux? Veel mensen die Linux proberen of erop overstappen zijn, zeker in het begin, de weg kwijt. Waar vind ik in de vredesnaam dat programma? Bijna niets van wat in Windows vertrouwd was, vind je in de mapstructuur van Linux gemakkelijk terug. In dit artikel zal ik u wegwijs maken in het labrynt van de door Linux gebruikte mapstructuur.

Algemene uitgangspunten

Laten we beginnen met een paar algemene uitgangspunten waarin Linux verschilt van Windows:

Mapstructuur vastgelegd in standaard

De mapstructuur van Linux is ontleend aan die van het besturingssysteem UNIX en vastgelegd in de FHS (File Hierarchy Standard). Alle Linux-distributies houden zich daaraan. De jongste nog in ontwikkeling zijnde versie 3.0 is hier te vinden: <http://tinyurl.com/fhs30>. In tegenstelling tot Linux heeft Windows geen standaard nodig, omdat er maar één Windows is met één eigenaar, Microsoft, die bepaalt hoe het eruit ziet.

Alles is een bestand

In Linux worden ook apparaten en dynamische proces- en systeemdata weergegeven als een bestand, in resp. de mappen /dev, /proc en /sys. Dit zijn bestanden die op geen enkele schijf voorkomen.

Geen driveletters maar koppelpunten
Linux kent geen driveletters. In plaats

van een aparte mapstructuur voor elk opslagmedium, zoals in Windows, kent Linux maar één basisstructuur, namelijk die van de systeemdrive (wat in Windows C: zou zijn). De mapstructuren van andere opslagmedia worden aan een map binen die structuur gekoppeld. Standaard zijn dat mappen onder /media voor verwisselbare opslagmedia zoals USB-drives en DVD-roms, en /mnt, of mappen daarvoor voor andere opslagmedia. Het is misschien leuk om te weten dat dat ook in Windows kan. Dat kan handig zijn voor vaste media, zoals harde schijven. Driveletters anders dan C: worden dan gereserveerd voor verwisselbare media.

Mappenscheider in paden: niet \, maar /.

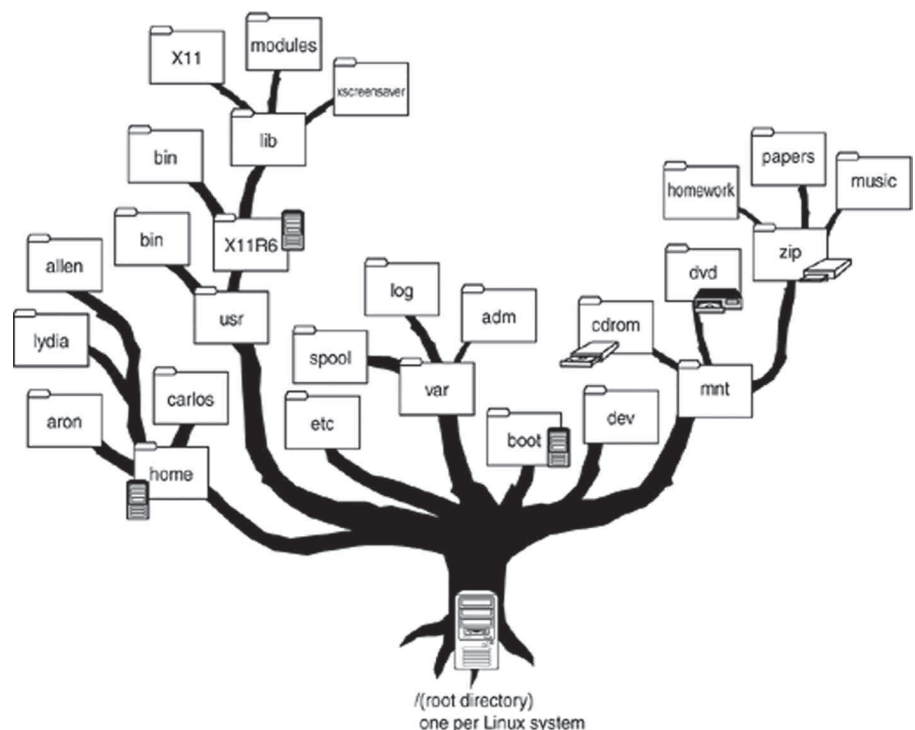
In paden door de mapstructuur naar een bestand worden de namen van de achtereenvolgende mappen gescheiden door een voorwaartse schuine streep (/) in plaats van een achterwaartse (\) zoals in Windows. In Windows zeggen we \\map1\map2\bestand, maar in Linux

/map1/map2/bestand.

Hoofd- en kleine letters verschillend
In Linux hebben hoofd- en kleine letters in namen van mappen en bestanden een verschillende betekenis. Zo zijn Bestand, bestand en BESTAND verschillende bestanden, terwijl die namen in Windows naar hetzelfde bestand verwijzen.

De File Hierarchy Standard

De FHS wordt onderhouden door de Linux Foundation en is nu onderdeel van de LSB (Linux Standard Base, <http://tinyurl.com/linuxsb>), waarvan de jongste versie 5.0 in ontwikkeling is. In de LSB is de inwendige structuur van het Linux-besturingssysteem vastgelegd om de compatibiliteit tussen de verschillende Linux-distributies te bevorderen. De huidige versie van de FHS, 2.3 van januari 2004, is te raadplegen op: <http://tinyurl.com/fhs230>.



De Linux mappenboom

Omdat deze inmiddels wat verouderd is, wordt gewerkt aan de nieuwe versie 3.0, waarin rekening wordt gehouden met de gevolgen van allerhande nieuwe ontwikkelingen. Het is hier niet de plaats om daar verder op in te gaan, maar u kunt er op webpagina <http://tinyurl.com/fhs30look> (Engels-talig) meer over lezen. In de door de FHS voorgeschreven namen van mappen, en ook bestanden, is de lange geschiedenis van UNIX, als het ware de vader van Linux, af te lezen. UNIX dateert van rond 1970. In die tijd waren computers nog niet zo krachtig, en moest er zuinig met ruimte op schijf en in het werkgeheugen worden omgegaan. Daarom kregen mappen en bestanden zo kort mogelijke namen die nogal cryptisch over kunnen komen. Het is wel even wennen! Zo heb je 'bin', 'mnt' en 'usr' in plaats van 'binaries' (uitvoerbare programma's), 'mount' (om andere drives aan te koppelen) en 'user programs', (programma's voor gebruikers), om maar wat voorbeelden te noemen. Windows heeft een veel recentere oorsprong en kon dan ook iets verkwestender met ruimte omgaan. Zo zie je daar veel duidelijker namen als 'Windows', 'Program Files' en 'Users'.

Omwille van de compatibiliteit met oudere software heeft de UNIX naamgeving alle ontwikkelingen op het gebied van computers overleefd. Toch zijn er ook andere structuren en modernere naamgevingen ontwikkeld. Het bekendste voorbeeld daarvan is Mac OS X van Apple. Dat systeem is gebaseerd op UNIX, en heeft dan ook alle structuren en naamgeving daarvan overgenomen. Daarbovenop heeft Apple echter een andere structuur met een duidelijker naamgeving gelegd, waarbij voor gewone gebruikers de UNIX-structuur onzichtbaar is gemaakt. De ontwikkelaars van GoboLinux, een bijzondere Linux distributie (<http://www.gobolinux.org>), vonden bij Mac OS X de inspiratie voor het ontwerp van een eigen mapstructuur. Daarin hebben de mappen geen verkorte cryptische namen meer maar duidelijke namen uit het woordenboek. Zo zijn er op het hoogste niveau de mappen 'Data', 'Mount', 'Programs', 'System' en 'Users'. Programma's worden in hun geheel onder een eigen map 'Programs' geïnstalleerd, net zoals in Windows. Net als Apple werkt GoboLinux in het verborgene toch ook met de FHS, omdat programma's daar nu eenmaal op rekenen. Een groot deel van zijn FHS-structuur bestaat uit onzichtbare koppelingen naar de voor gebruikers zichtbare Gobo-mappenstructuur.

Een belangrijk deel van de FHS weerspiegelt de functie van de bestanden die erin zijn opgenomen. Uitvoerbare programma's hebben vaste mappen (/bin, /usr/bin, etc.), net als de bijbehorende libraries (/lib, /usr/lib, etc.),

configuratie (/etc), en overige bestanden, waaronder documentatie (/usr/share). Dat heeft tot gevolg dat programmapakketten bij installatie helemaal uit elkaar worden getrokken, zodat op allerlei plekken wat staat. Daarom maakt elke Linux-distributie gebruik van de diensten van een pakketbeheerprogramma, dat in een database precies bijhoudt welke pakketten geïnstalleerd zijn, en waar hun bestanden zijn geplaatst. In Windows hoef je maar in 'Program Files' te kijken om een indruk te krijgen van wat er allemaal is geïnstalleerd. Er is in Linux één map waar programma's op de Windows-manier kunnen worden geïnstalleerd: zijn naam, /opt (van 'optional'), zegt al dat dat niet standaard is. De indeling naar functie is wel erg gemakkelijk voor het eenvoudig starten van programma's buiten het menu om. Het systeem hoeft alleen te kijken in het programmapad, een lijstje van de standaardmappen met programma's, om het programma te kunnen vinden. Dat is in Windows veel lastiger.

De mapstructuur

We beginnen onze verkenning van de mapstructuur van Linux bij de rootmap (in het Engels root directory), dat is de map op het hoogste niveau. Hij heeft geen eigen naam, maar wordt eenvoudig aangeduid met het teken dat in paden als mappenscheider wordt gebruikt: de voorwaartse schuine streep /. Een / aan het begin van een pad betekent dat het pad bij de root begint. Verwarrend is wel dat er in de rootmap ook een map met de naam 'root' is. Die heeft niets met de rootmap van doen, maar is de thuismap voor de systeembeheerder die in Linux 'root' heet.

De aanduiding van de rootmap als / is vergelijkbaar met Windows, waar de rootmap van een drive ook wordt aangeduid met de mappenscheider: een achterwaartse schuine streep \.

In de rootmap komen we heel wat mappen tegen. Hoe ze kunnen worden onderverdeeld en waar ze voor dienen staat in het volgende overzicht:

Systeemmappen, te vergelijken met wat je op een Windows systeem onder C:\Windows vindt:
 /boot voor systeemstart (bootmanager)
 /bin, /sbin en /lib voor systeemprogramma's en -libraries
 /dev, /proc en /sys voor virtuele bestanden (apparaten, processen, e.d.)
 /run, /srv, /tmp en /var voor sessiegerelateerde, tijdelijke en veranderlijke bestanden (databases, websites, log- en spoolbestanden, caches, etc.)

/etc voor configuratiebestanden, te vergelijken met de systeemregistry in Windows, maar hier is het allemaal leesbaar

Koppelmappen, te vergelijken met extra driveletters in Windows:
 /media voor verwisselbare opslag-media, zoals DVD-ROMs en USB-sticks

/mnt voor tijdelijk aankoppelen van andere bestandssystemen

Gebruikerstoepassingen, te vergelijken met 'C:\Program Files' en C:\ProgramData in Windows:

/usr voor centraal (door pakketbeheer) geïnstalleerde toepassingen, waarin /usr/share te vergelijken is met C:\ProgramData in Windows

/usr/local voor lokaal (zelf) geïnstalleerde toepassingen, met eenzelfde mapstructuur als /usr.

/opt voor optionele softwarepakketten en add-ons, met per pakket een eigen map, zoals in 'C:\Program Files' in Windows.

Gebruikersdata, te vergelijken met C:\Users (voorheen 'C:\Documents and Settings') in Windows:

/root voor de systeembeheerder ('root', die onder Windows Administrator is)

/home voor alle andere gebruikers, met per gebruiker een eigen map.

We gaan nu wat dieper in op al deze mappen en hun gebruik.

Systeemstart

/boot bevat bestanden voor de bootmanager en meestal ook één of meer Linux-kernels (genaamd vmlinuz, vaak gevolgd door een versienummer), ook al mogen de kernels ook in de hoofdmap staan. De meest gebruikte bootmanager is Grub. De gegevens die door Grub bij het opstarten worden gebruikt, zoals het bootmenu, staan in /boot/grub.

Systeemprogramma's en -libraries

/bin en /sbin bevatten voor het systeem essentiële programma's. /bin (staat voor 'binaries', dat zijn uitvoerbare programma's zoals exe- en bat-bestanden in Windows) bevat programma's die door iedereen kunnen worden gebruikt, zoals de shell bash (te vergelijken met de Windows shell cmd.exe), ls en cp (dir en copy in Windows). /sbin (staat voor 'system binaries') bevat programma's voor beheerdoeleinden die normaliter alleen door root worden gebruikt, zoals programma's voor het formatteren van een schijf.

/lib (staat voor 'libraries') bevat voor het systeem essentiële libraries (zoals de dll-bestanden in Windows, die

hier so (shared object) heten) en bijbehorende gegevens. Deze kunnen in eigen submappen zijn geplaatst. Onder `/lib/modules` vinden we alle kernel-modules, waaronder drivers voor allerlei apparaten, bestandssystemen en netwerkinterfaces. `/lib32` en `/lib64` komen voor op 64-bit systemen. De eerste bevat 32-bit libraries voor programma's die dat nodig hebben, terwijl de tweede, net als `/lib` 64-bit, libraries of koppelingen ernaar kan bevatten.

Virtuele bestanden

`/dev` (afkorting van 'devices' ofwel apparaten) bevat representaties van alle aangesloten apparaten en interfaces als bestanden, deels onderverdeeld in submappen voor de verschillende typen apparaten. Ook bij een niet-draaiend systeem is deze map gevuld met een aantal standaard apparaten, maar bij een draaiend systeem wordt de feitelijk aangetroffen situatie weergegeven. Zo vinden we daar `/dev/sda` voor de eerste (de 'a' van sda) harde schijf en `/dev/sda1` voor de eerste primaire partitie op die schijf. Het is mogelijk om rechtstreeks van apparaten te lezen en er naar te schrijven, maar dat gebeurt dan wel 'rauw', zonder tussenkomst van drivers die van de geëigende structuren gebruik maken. Zo is het bijvoorbeeld mogelijk om naar de schijf `/dev/sda` te schrijven. Ik raad dat echter ten zeerste af omdat de kans heel groot is dat je essentiële data en structuren op de schijf overschrijft. Zeg dan maar dag met je handje tegen je data. De luidspreker van de computer heeft eveneens een eigen ingang in `/dev`: `/dev/dsp`. Ook daar kunt u naar schrijven. Probeer het maar eens door er de inhoud van het shell programma `/bin/bash` heen te schrijven met `'cat /bin/bash > /dev/dsp'`. Wat u hoort is het geluid van bash! Als u het te lang vindt dueren kunt u het afbreken met Ctrl-C. Naast fysieke apparaten treft u in `/dev` ook softwarematige apparaten aan. Zo is `/dev/null` 'de bittentak' die alles wat ernaartoe gestuurd wordt in het niets laat verdwijnen, levert `/dev/random` een stroom aan willekeurige bytes, en `/dev/zero` een stroom nulbytes.

Je leest wel eens op het web dat `/dev/sda1` te vergelijken is met C: in Windows. Niets is minder waar. C: is de virtuele drive-aanduiding in de route door de mappenstructuur naar de bestanden op de drive, terwijl `/dev/sda1` een fysiek apparaat weergeeft, in dit geval de eerste partitie op een harde schijf. Het verschil is goed te zien in een RAID1-configuratie, waarbij exact dezelfde data op twee schijven worden geplaatst ('mirroring' ofwel 'spiegeling'). Er zijn dan twee

schijfpartities, zeg `/dev/sda1` en `/dev/sdb1`, gespiegeld, waarbij er één logische structuur overheen is gelegd die in Windows C: wordt genoemd.

`/proc` (van 'process data') bevat in de vorm van mappen met daarin bestanden gegevens over alle processen die op het systeem draaien. Het systeem geeft aan dat deze bestanden een grootte nul hebben, maar ze bevatten toch gegevens! Het zijn virtuele bestanden die hun inhoud ontlene aan de draaiende Linux kernel. `/proc` kan ook worden gebruikt voor uitwisseling van gegevens tussen processen en de kernel. In een aantal gevallen is het zelfs mogelijk om data te wijzigen en zo direct in te grijpen in de werking van het systeem. Wees daar wel heel voorzichtig mee! Op een niet draaiend systeem is deze map leeg. Programma's die informatie tonen over het systeem, zoals KinfoCenter van KDE, ontlene hun informatie vrijwel allemaal aan de virtuele bestanden in `/proc`. Het voordeel van dat soort programma's is, dat ze de informatie goed leesbaar en geordend tonen, maar u kunt al die virtuele bestanden ook met een simpele viewer lezen om de gewenste informatie te krijgen. Kijk maar eens naar `/proc/cpuinfo`, `/proc/meminfo` en `/proc/mount`.

`/sys` ('system data') is vergelijkbaar met `/proc`. Het toont de informatie over apparaten en hun drivers zoals die bij de Linux kernel bekend is, en over bepaalde kenmerken van de Linux-kernel. Ook deze map is op een niet-draaiend systeem leeg.

Sessiegerelateerde, tijdelijke en veranderlijke bestanden

`/run` ('runtime data') bevat gegevens over het lopende systeem sinds het is opgestart, zoals pid-(process identification) bestanden, die voor elk draaiend programma een procesnummer geven. In `/proc` zijn in de map met dat nummer gegevens over dat proces te vinden. Op een niet-draaiend systeem bevat de map `/run` alleen enkele standaardbestanden en -mappen. In oudere distributies is deze map te vinden onder `/var`, dus `/var/run`.

`/srv` ('service data') bevat site-specifieke data voor servers (diensten) die op het systeem draaien. Daarbij moet u denken aan onder meer ftp (file transfer protocol), www (websites) en rsync (synchronisatie van mappen en bestanden tussen computers). Ook al komt deze map wel voor in alle voornaamste distributies, hij wordt niet in alle gevallen gebruikt. OpenSUSE gebruikt hem, maar Ubuntu niet. In Ubuntu staan deze gegevens onder `/var`.



`/tmp` ('temporary data') dient voor het opslaan van tijdelijke bestanden door programma's die dat nodig hebben. Zo gauw een programma niet meer draait kunnen door dat programma aangemaakte tijdelijke bestanden worden verwijderd voor zover het programma dat niet zelf al gedaan heeft. Afgezien van het feit dat ze ruimte innemen, kan het overigens geen kwaad dat ze blijven staan.

`/var` ('variable data') dient voor de opslag van variabele gegevensbestanden, zoals databases, spoolbestanden van de printer, logs, administratieve bestanden, caches en in tijdelijke bestanden die tussen sessies bewaard moeten blijven (anders dan `/tmp` dus). Onder `/var` is voor al die verschillende data een secundaire mapstructuur gedefinieerd met onder meer `/var/lib`, `/var/spool`, `/var/log`, `/var/cache` en `/var/tmp`. Een distributie als Ubuntu slaat hier onder `/var/www` ook websites voor het World Wide Web op, terwijl de FHS hier eigenlijk de map `/srv` voor heeft bedoeld.

Heel belangrijk om in de gaten te houden zijn de logbestanden in `/var/log`. Daarin is het hele reilen en zeilen van het systeem of te lezen. Zo bevat `/var/log/dmmsg` ('display message') berichten die tijdens het booten van de computer door het systeem worden gegenereerd, terwijl `/var/log/messages` alle berichten laat zien die daarna tijdens het draaien door het systeem worden verstuurd.

Configuratiebestanden

`/etc` bevat alle configuratiebestanden die op het hele systeem betrekking hebben. De map vervult dus ongeveer dezelfde functie als het systeemregister in Windows, maar is veel gemakkelijker in het gebruik. In plaats van een speciale structuur in een apart bestand maakt het gewoon deel uit van de normale mapstructuur, terwijl de configuratiebestanden zelf bijna allemaal 'platte' tekstbestanden zijn, die gemakkelijk met een simpele teksteditor (zoals Kladblok in Windows) kunnen worden gelezen en bewerkt. Om het systeemregister van Windows te kunnen bekijken en bewerken is een speciaal programma, `regedit.exe`, nodig.

Over de oorsprong van de naam 'etc' zijn de deskundigen het niet eens. De meest houtsnijdende uitleg vind ik wel dat 'etc' staat voor 'editable text configuration', configuratie als bewerkbare tekst dus.

Koppelmappen

`/media` is de map voor koppelpunten van verwisselbare opslagmedia, zoals DVD's, USB-sticks en USB-harde schijven die door het systeem worden aangekoppeld. In de praktijk zie je dat ook niet-verwisselbare opslagmedia, zoals een Windows-partitie, hier worden aangekoppeld. Voor elk opslagmedium waarvoor u dat vraagt maakt het systeem een koppelpunt onder `/media` met een naam die aan het opslagmedium zelf wordt ontleend. Sinds kort wordt dat zelfs per gebruiker gedaan: onder `/media` wordt een map met de naam van de gebruiker aangemaakt en daaronder verschijnt dan de map met het opslagmedium. Als voorbeeld diene mijn Philips USB-stick, die aan `'/media/hans/PHILIPS UFD'` wordt gekoppeld, met lees- en schrijfpermissies voor mijzelf. Duidelijker dan een willekeurige driveletter die door Windows wordt toegekend, nietwaar? Als ik het systeem vraag de stick weer te ontkoppelen verdwijnt de map `'PHILIPS UFD'`, en ook de map `'hans'` als ik geen andere opslagmedia aangekoppeld heb.

`/mnt` ('mount') is het koppelpunt van handmatig aangekoppelde opslagmedia. Binnen `/mnt` kunnen eventueel mappen worden gemaakt, waarvan elk als koppelpunt voor bijvoorbeeld een bepaald type medium fungeert.

Gebruikerstoepassingen

`/opt` ('optional') kan worden gebruikt voor optionele softwarepakketten en add-ons. Elk pakket heeft z'n eigen map, net als in `'C:\Program Files'` in Windows. Softwarepakketten die niet via het pakketbeheersysteem kunnen worden geïnstalleerd doordat alleen een tar.gz of zip wordt aangeboden kunnen gemakkelijk handmatig in `/opt` worden geplaatst. Er zijn maar weinig pakketten die zich via het pakketbeheer in `/opt` laten installeren. Een voorbeeld is het bekende TeamViewer voor besturing van een computer op afstand.

`/usr` ('user programs') is de plek voor centraal (door het pakketbeheer) geïnstalleerde toepassingen. Daartoe is binnen `/usr` een secundaire mapstructuur gedefinieerd. Daarin bevatten `/usr/bin`, `/usr/sbin` en `/usr/lib` voor het systeem niet essentiële programma's voor gebruikers (bin) en systeembeheerders (sbin) en de bijbehorende libraries (lib). Verder zijn er `/usr/include` met headerbestanden voor C-programma's en,

heel belangrijk, `/usr/share`. Deze laatste map bevat officieel van de architectuur onafhankelijke data, hetgeen wil zeggen dat ze niet afhankelijk zijn van wat voor processor er in de computer zit en dus door verschillende architecturen (computers) kunnen worden gedeeld. Vandaar de naam 'share'. Het gaat om documentatie (`/usr/share/doc`), handleidingen (`/usr/share/man` en `/usr/share/info`), printerdefinities (`/usr/share/ppd`), vertalingen (`/usr/share/locale`) en nog veel meer. Elk programma is vrij om er eigen mappen aan toe te voegen. Onder `/usr` vinden we ook `/usr/local` voor lokaal (zelf) geïnstalleerde toepassingen. Ook daarbinnen is een (nu tertiaire) mapstructuur gedefinieerd, die ongeveer gelijk is aan die onder `/usr` zelf. Zo zijn er `/usr/local/bin`, `/usr/local/sbin` en `/usr/local/lib` voor programma's en libraries, `/usr/local/include` voor headerbestanden, `/usr/local/share` voor gedeelde data. Handleidingen horen hieronder `/usr/local/man`. Daarnaast is er ook nog een vaste map `/usr/local/etc` voor configuratiebestanden. Pakketten die als broncode worden gedownload en met de bijgeleverde scripts worden gecompileerd en geïnstalleerd, worden normaliter in `/usr/local` geïnstalleerd.

Gebruikersdata

`/home` is de plek waar gebruikers van het systeem hun bestanden opslaan. Elke gebruiker heeft onder `/home` zijn eigen thuismap die normaliter genoemd is naar zijn inlognaam. Naast de documenten, foto's, video's en geluidsbestanden van de gebruiker wordt ook de persoonlijke configuratie van de door de gebruiker toegepaste programma's in die map opgeslagen. Helaas niet in de vorm van mappen en bestanden in een map `/home/<gebruiker>/etc`, wat ik zelf duidelijk en consequent zou vinden, maar als verborgen mappen en bestanden direct onder de thuismap. Verborgen mappen en bestanden onderscheiden zich van zichtbare door een punt voor de naam. Vrijwel alle programma's maakten hun eigen configuratiebestand of `-map` direct onder de thuismap, maar inmiddels is er wel wat orde in de chaos gebracht door de introductie van een aantal

standaardmappen - `.cache`, `.config` en `.local/share` - waarin programma's hun gegevens kunnen plaatsen. Jammer genoeg zijn er nog steeds heel wat programma's die daar geen gebruik van maken, zoals nota bene de browser Firefox en het fotobewerkingsprogramma GIMP. Het kantoorpakket LibreOffice doet het wél goed!

Het is goed gebruik om de inhoud van `/home` op een eigen partitie van een harde schijf te zetten en deze bij het opstarten van het systeem automatisch aan `/home` te laten koppelen (via het bestand `/etc/fstab`, waarin `fstab` staat voor 'file system table'). Dat heeft het voordeel dat bij een herinstallatie of nieuwe installatie van het systeem alle gebruikersinstellingen en -gegevens bewaard blijven.

`/root` is de thuismap voor de systeembeheerder, in Linux 'root' genoemd. Hij is niet in `/home` ondergebracht omdat `/home` vaak op een aparte partitie wordt geplaatst, terwijl hij ook beschikbaar moet zijn als `/home` niet aangekoppeld is. Z'n indeling is precies gelijk aan die van een normale gebruikersmap.

Bijzondere mappen

`/lost+found` In de rootmap van een partitie zult u meestal ook een map met deze naam aantreffen. Deze hoort bij het huidige standaard bestandssysteem van Linux, ext4, en zijn voorgangers ext3 en ext2. Hierin worden bestanden opgeslagen met data die bij herstel van het systeem na een storing niet meer bij een map of bestand hoorden, maar nog wel konden worden teruggevonden.

Tot slot

Hiermee eindigt onze globale verkenning van de mapstructuur van Linux en is een mooie basis gelegd voor een verdergaande verkenning. Daarbij kan het Nederlandstalige Wikipedia-artikel over de File Hierarchy Standard mogelijk een nuttige rol vervullen: kijk op <http://tinyurl.com/wikifhs>.

