

Korte uitleg: File systems

Wat is een filesystem en wat kan er belangrijk zijn bij het beheren van filesystemen. Hieronder een aantal onderwerpen.

Wat is het?

Zoals in *Korte uitleg: Directories, meta-data van bestanden, inodes* behandeld, zijn directories, inodes en adressering onderdelen van een filesystem. Een filesystem is de implementatie om een structuur van deze onderdelen de lucht te houden en ook een mate van stabiliteit te geven. Vooral in het laatste verschillen veel types filesystemen van elkaar. Hoe maak je addresseringsketens redundant zoda bij leesproblemen toch het redden van de gegevens mogelijk is? Wat doe je bij plotseling uitvallen van het systeem? In de loop van de tijd zijn hiervoor veel filesystemen omtworpen.

De implementatie van een filesystem bevat Kernel modulen en hulpprogramma's.

Een filesystem type is veelal afhankelijk van het systeem waarvoor het is ontworpen. Voor Unix/Linux moet het bijvoorbeeld voorzien in eigenaar, groep, toestemmingsbitjes en tijdstempels. Voor Linux noem ik Reiserfs, de ext2/3/4 serie, Btrfs. Ook voor MS-DOS/Windows zijn er filesystemen, maar deze voldoen niet aan de eisen van een Unix/Linux systeem.

Waar is het?

Een filesystem moet ergens "op staan". Anders gezegd er is een stuk mass-storage/schijf nodig om het op te zetten. Dat kan een disk-partition zijn, maar er zijn veel meer mogelijkheden. Zo kan ook een niet gepartitioneerde (hele) disk gebruikt worden. Of een Logical Volume van de Logical Volume Manager. Of een MD device van Linux Software RAID. Als algemene term gebruik ik daar van het woord "container" voor.

Hoe maak ik het?

Een filesystem moet worden aangemaakt. Er worden dan ketens van inodes, over de container verspreid, geschreven. Ook andere zaken zoals algemene gegevens komen op de container te staan. Daarvoor zijn programma's, die uiteraard per filesystem type verschillen. In het algemeen is hun naam een variatie van `mkfs` (make file system). In feite is `mkfs` een omvattend gereedschap waar je met het aangeven van het filesystem type allerlei soorten filesystemen kan aanmaken. `mkfs` roept dan zelf bijv. `mke2fs` aan. Overigens, `mke2fs` kun je ook aanroepen als `mkfs.ext2` en er zijn varianten als `mkfs.ext3`, `mkfs.ext4`.

Ieder type filesystem heeft ook zijn eigen aanpasbare parameters. Die kun je bij aanmaken opgeven. Sommige parameters kun je later wijzigen. Voor

ext2/3/4 filesystemen gaat dat met `tune2fs` (daarmee kun je bijvoorbeeld een Volume Label aan je filesystem hangen). Je kunt hiermee zelfs (met beperkingen) van het ene ext filesystem naar het andere omschakelen.

Ter verhoging van de spraakverwarring wordt het aanmaken van een filesystem door ex MS-DOS gebruikers vaak "formatteren" genoemd.

Meestal gebruik je voor het aanmaken van een filesystem een GUI tool zoals YaST > Systeem > Partitionering. Er zijn ook andere zgn. partitionerings programma's. Er zijn ook vanaf een CD/DVD/USB direct te booten partitionerings programma's, zodat je ook kunt werken als je geen systeem hebt of dit kapot is. Overigens kan dat ook met booten van je installatie medium en dan Rescue kiezen en vandaar met de commando's of met YaST werken.

Merk op dat deze tools als Partitionering tools/programs bekend staan. In feite kunnen ze vaak veel meer dan alleen partitioneren, zoals filesystem aanmaken (reden waarom ik ze hier vermeld), vergroten/-kleinen, mountpoints maken en fstab configureren.

Hoe groot?

Omdat binnen een filesystem adressen van plekken binnen het filesystem worden gebruikt is het van belang hoe groot die adressen kunnen worden. Dat bepaalt hoe groot een filesystem maximaal kan zijn. Aangezien de schijven steeds groter worden zijn er nog wel eens filesystem ontwerpen aan hun maximum gekomen. Nieuwere (versies van) filesystemen zijn daar vaak aan aangepast. Van belang is te weten dat er grenzen zijn.

Andere beperkingen zijn die van de maximale grootte van de individuele bestanden en van het aantal bestanden op een filesystem. Meestal niet iets om je echt zorgen over te maken.

Hoe vol?

Zoals we zagen bevat een filesystem administratie (inodes e.d.). Dus niet alle ruimte is bruikbaar voor bestanden.

Uiteraard kan een filesystem vol raken. Het is echter niet handig om alles te gebruiken. Als het filesystem bijna vol is, zal het moeite hebben om gunstige plekken voor nieuwe bestanden te vinden en dat beïnvloedt de performance nadelig. Ook kunnen er grotere problemen ontstaan als het systeem geen schijfruimte meer kan krijgen voor log bestanden en helemaal als het bij booten niets meer kan aanmaken. Daarom stopt een filesystem meestal met het uitgeven van nieuwe schijfblokken als nog slechts vijf procent (5%) van de ruimte vrij is. Als gebruiker merk je dat bijv. doordat je niet meer kunt inloggen. Let op!! Die 5% kan wel door `root` processen verkregen worden. Dit heeft tot gevolg dat je in ieder geval als `root` kunt inloggen (niet in de GUI!) om de zaak te repareren.

Nu is 5% ongebruikte ruimte bij een 20 GiB filesystem geen groot verlies, maar als je een filesystem van een TiB hebt is dat veel te veel. Je kunt dat op

ext2/3/4 filesystemen met `tune2fs` aanpassen of zelfs helemaal uitzetten. Aanbeveling: houdt het op een root filesystem (20 - 100 Gib) op 5%. Zet het op andere filesystemen lager of uit. Nog een aanbeveling: controleer af en toe eens hoe vol het is:

```
df -h
```

In dit verband is het goed te weten dat er zgn. "sparse files" bestaan. Dat zijn bestanden waarvan niet alle schijfblokken in gebruik zijn. Het is bijv, bij databases zeer gebruikelijk om de benodigde ruimte van de Kernel te claimen door bijvoorbeeld het laatste blok van een bestand te schrijven terwijl er tussenin (nog) niets staat. Dat kan verrassingen opleveren bij het beoordelen van de vrije ruimte.

Een laatste opmerking over Btrfs filesystemen. Btrfs kan zogenaamde "snapshots" maken. Bij verstek staat dit aan in openSUSE. Deze snapshots nemen natuurlijk ruimte in, maar worden niet "gezien" door df. Er zijn meerdere problemen in de forums met volle btrfs filesystemen die volgens de eigenaars met df gezien, niet vol waren.

Beperkingen bij het booten.

Om te booten moet een bootloader (bijv. GRUB) enkele zaken (zoals de GRUB configuratie en de Kernel) op een filesystem kunnen vinden. Deze zaken zijn allemaal geplaatst in `/boot`. Dat betekent dat GRUB eenvoudige handelingen zoals het vinden van een bestand en het lezen daarvan op zo'n filesystem moet kunnen. GRUB ondersteunt niet alle types filesystem, dat zou er een waterhoofd van maken. Dat betekent dat het soms nodig is om van `/boot` een apart filesystem te maken zodat de rest van het systeem wel op het gewenste type kan staan.