Bestanden

In dit hoofdstuk worden verschillende soorten **bestanden** besproken en wordt naar het **permissiesysteem** gekeken. Uiteraard maken we weer nader kennis met de **commando's** die hierbij een rol spelen.

Doelstelling

Aan het eind van dit hoofdstuk is de cursist bekend met:

- verschillende soorten bestanden en hun rol in het filesysteem
- het feit dat ”alles” in Linux een bestand is
- naamgeving
- bestanden inzien
- bestanden vergelijken
- users en groepen
- het Linux permissiesysteem
- inodes
- de bij al bovenstaande acties behorende commando's

Soorten bestanden

Eerste karakter in `ls -l` output geeft type bestand aan:

`-` normaal data file: tekst, binary, jpg, gezipped

`d` directory. Bevat **alleen** filenamen gekoppeld aan verwijzingen (pointers) naar *inodes*

`l` symbolic link: verwijzing naar een ander file

`p` named pipe (data transport tussen programma's)

`s` socket (named pipe maar voor netwerk connecties)

`b` block device (data transfer in blocken: disks, cdrom's, ..)

`c` character device (data transfer per byte: terminal, serie ële en parallele poorten)

Alles in Linux is een file!

Naamgeving

- maximaal 255 karakters
- forward slash (/) niet toegestaan
- **vermijd speciale tekens**. Gebruik vooral (lowercase) alfabetische tekens met evt. hyphen (-), underscore (_), cijfers en de punt.
- Linux is case sensitive!

---

links

Van een file is een ”duplicaat” te maken met het commando `ln`

Bijvoorbeeld:

```
ln file linknaam
```

- `linknaam` krijgt dezelfde *inode* als `file`. Dit wordt ook wel een **hard link** genoemnd. De *link count* (2e veld in de `ls -l` output) wordt met 1 opgehoogd.

Met optie `-s` maak je een **symbolische link**:

```
ln -s file symlink
```

- `symlink` heeft eigen inode en is een **verwijzing** naar `file`

hard linken van directories kan niet

hard links zijn beperkt tot het filesysteem (vanwege de inode)

symbolic links kunnen over mountpoints heen wijzen (target hoeft niet eens te bestaan)

Bestanden inzien

Bepaal eerst het type file (voordat je bv. een binary file opent): `$ file filenaam`

Commando's om bestanden in te zien:

`cat`	concatenate
`more`	vang output op
`less`	vang output op, ook terug te scrollen
`pr`	format output (eenvoudig) voor printen

Bestanden vergelijken (1/2)

`cmp`

- byte voor byte vergelijken (ook binary files)
- geeft alleen regelnummer en plaats in de regel waar de bestanden voor het eerst verschillen

```
comm [optie]... file1 file2
```

- alleen (gesorteerde) ascii files
- toont de verschillen en overeenkomsten in 3 kolommen
  - 1e kolom: toont regels uniek in file1
  - 2e kolom: toont regels uniek in file 2
  - 3e kolom: toont regels die voorkomen in beide files

  Met opties kunnen kolommen worden weggelaten

Bestanden vergelijken (2/2)

`diff`

- ook alleen voor ascii files
- geeft regel voor regel de verschillen
- de output van diff kan bewaard worden in een file en dat file (patchfile) kan met het commando patch gebruikt worden om het origineel te patchen. Voordeel: alleen het patchfile hoeft gecommuniceerd te worden: minder data.

Patchen doe je dan met:

```
patch [optie]... orig pathfile
```

Users en groups

Onderdeel van Linux 2 maar in het kort:

- er zijn users (met een naam / uid)
  - administratie in `/etc/passwd`
- er zijn groepen (groups)
  - administratie in `/etc/group`
- een user kan in meerdere groepen zitten
- een file heeft een owner (user) en behoort in een group. Met `ls -l` worden deze getoond (resp. de derde en vierde kolom in de output)

/etc/passwd

Ook wel ”het password file” genoemd (hoewel er geen passwords in staan)

7 velden gescheiden door de dubbele punt (:)

- veld 1: usernaam (loginnaam)
- veld 2: verwijzing naar het password file (/etc/shadow)
- veld 3: user-id (uid)
- veld 4: group-id (gid). Dit is de standaard group voor de user
- veld 5: GECOS veld, weinig gebruikt, commentaar
- veld 6: home-directory van de user
- veld 7: de shell die de user krijgt

1 regel per user

```
oscar:x:1000:100:Oscar Buse,,,:/home/oscar:/bin/bash
```

/etc/group

4 velden gescheiden door de dubbele punt (:)

- veld 1: groupnaam
- veld 2: wachtwoord veld
- veld 3: group-id (gid)
- veld 4: group leden (*members*)

1 regel per group

```
admin:x:119:oscar
other:x:100:
```

---

het permissiesysteem

Bepalend:

- karakters 2 t/m 10 (de *permissie bits*) van de `ls -l` output.

  drie triolen (*triplets*):

  - owner (**u**), group (**g**) en others (**o**)

    met elk:

    - 1e karakter: lees (read) permissie
    - 2e karakter: schrijf (write) permissie
    - 3e karakter: execute permissie

Voorbeeld: (`ls -l /etc/passwd`)

---

enkele mogelijke permissies

permissie string	code (octal)	uitleg
rw-r--r--	644	lees rechten voor iederen, schrijf rechten voor de user
rwxr-xr-x	755	lees/execute voor iedereen, schrijf rechten voor de user
rw-rw-rw	666	lees/schrijf rechten voor iedereen
rwx------	700	less/schrijf/execute rechten voor de user

code = octale waarde van de permissie bits:

permissie = 1, geen permissie = 0

Bijvoorbeeld:

```
rw- = 110 = 6
r-- = 100 = 4
```

Dus voor de string rw-r--r--:

```
rw-r--r-- = 110 100 100 = 6 4 4 = 644
```

Octale code legt de permissie strings eenduidig vast

---

verschil files en directories

permissie	op file	op directory
read	mogelijk om file te **lezen** (cat, vi, more, less, ...)	mogelijk om **inhoud** van de directory te zien (ls)
write	mogelijk om file te **wijzigen** (vi)	mogelijk om **inhoud** te wijzigen (touch, cp, mv, rm, mkdir, rmdir)
execute	mogelijk om file **uit te voeren** (./file). Voor binaries, shell scripts)	mogelijk om naar directory te **verplaatsen** (cd)
default (met umask 022)	rw-r--r--	rwxr-xr-x

De default permissie wordt bepaald door de **umask** setting:

- typisch waarde `022` (geen write permissie voor de group en others).
- Aan te passen met commando `umask` of permanent in profile bestand
- Files krijgen standaard geen execute permissie, directories wel.

Permissies op symbolic links zijn altijd rwxrwxrwx. De permissie van het targetfile bepalen uiteindelijk wat mag.

Permissies wijzigen

Commando:

- `chmod [optie]...  ”mode” file...`

Twee manieren van wijzigen:

- **relatief**: zet die bits aan/uit die je wilt, doe niets met de andere
  - `chmod ugo+w file`
  - `chmod g-w,o+r file`
- **absoluut**: geef met een octale code precies aan hoe het moet worden:
  - `chmod 755 file -> rwxr-xr-x`
  - `chmod 400 file -> r--------`
- `chmod a=r <=> chmod 444`

---

speciale permissies (1/2)

SUID Set user ID

- op executable files
- runt het file (commando) met de privileges van de **eigenaar** van het commado i.p.v. (normaal) met de privileges van degene dat het commando start
- suid-bit (”s-bit”, **geen** sticky bit!) in user-deel:

```
-rwsr-xr-x 1 root root 41284 2011-06-24 11:36 /usr/bin/passwd
```

SGID Set groups ID

zelfde als suid-bit maar dan een s in het group-deel: programma runt dan ook met de privileges van de group

MAAR: als op een **directory** dan een andere betekenis: nieuwe files (of directories) in die directory krijgen de group eigenaar van de directory met het s-bit en niet, zoals normaal, de group eigenaar van de user's standaard group.

speciale permissies (2/2)

Het sticky bit

Vroeger: op een commando (als `vi`) om aan te geven dat een programma na afsluiten nog even in memory kon blijven (sticky).

Bij weer opnieuw starten is het dan al reeds in memory (en start dus veel sneller op).

Maar tegenwoordig op bv. tmp (`/tmp`, `/var/tmp`) directories:

```
drwxrwxrwt 17 root root 12288 2012-01-31 22:39
/tmp
```

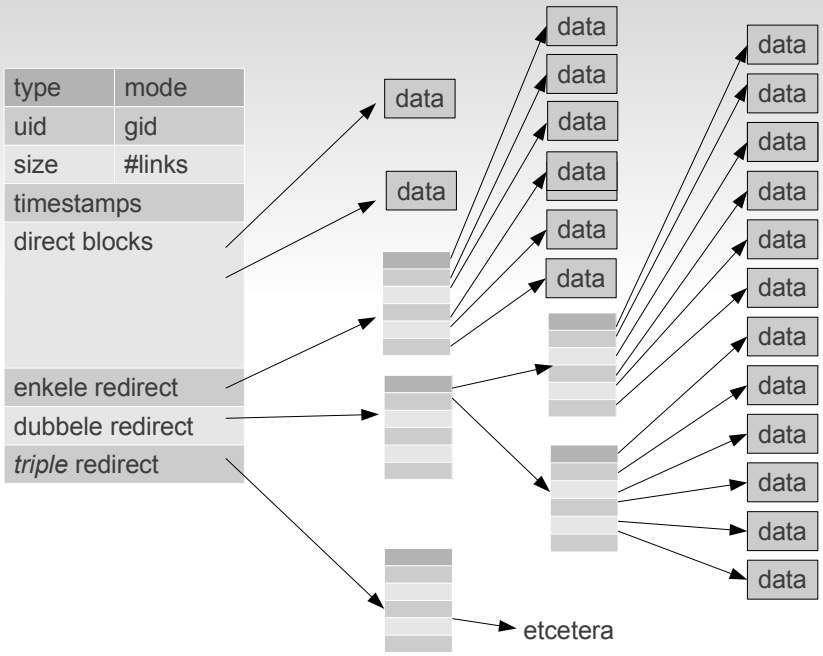
- **`t`** op *other* executie bit (laatste bit in de permissie string)
- ondanks w-bit voor iedereen **voorkomt** het sticky bit dat users files van **anderen** kunnen verwijderen

---

de inode

- Inhoud van een directory: namen gekoppeld aan verwijzingen (pointers) naar inodes.
- Verwijzing naar inode te zien met `ls -i`
- Vrije inodes: `df -i`
- De inode bevat de data en eigenschappen (permissies, owner, group, size, timestamps, ...).
- Wat u ziet met `ls -l` komt niet uit de directory maar uit de inode
- Bij grote bestanden wijzen de datablokken in de inode weer naar andere data blocken (enkele, dubbele en driedubbele redirects, zie schema volgende slide):

Inode schematisch



Oefeningen

Tijd voor oefening!