

# Bestanden en permissies in linux

## Over dit document

- Auteur: Merijn Vogel
- Datum: 2023-02-05

## Inleiding werken met bestanden in linux

Linux is een systeem dat kan worden gebruikt door meerdere mensen tegelijk. Dit willen we uiteraard goed begrijpen! Daarom gaan we onderzoeken hoe het zit met bestanden, mappen en bijhorende rechten.

De vraag is: hoe werk je met meerdere gebruikers aan bestanden in linux?

Dit document beantwoordt de volgende vragen:

- wat zijn bestandspermissies?
- hoe maak je mappen en bestanden?
- hoe scherm je mappen en bestanden af?
- hoe worden gebruikers gebruikers in groepen ondergebracht om bepaalde bestanden te delen?

## Bestandspermissies

# Drie soorten permissies

Linux onderscheidt drie soorten permissies:

- `read` : leesrechten op een bestand
- `write` : schrijfrechten op een bestand,
- `execute` : het bestand kan worden uitgevoerd.

`read` betekent dat het bestand gelezen kan worden. `write` betekent dat naar het bestand kan worden geschreven.

Het intrekken van de `write` permissie betekent dat een bestand alleen leesbaar is (read-only).

`execute` betekent dat een bestand direct uitvoerbaar is. Een python-bestand is normaal gesproken niet uitvoerbaar; dat moet je uitvoeren met `python bestand.py`. `python` zelf is wel een uitvoerbaar bestand.

## Drie permissiegroepen

Een bestand kent drie groepen waarvoor de soorten permissies kunnen worden gezet:

- `owner` : de eigenaar van het bestand,
- `group` : de groep van het bestand (zie de sectie gebruikers en groepen)
- `other` : rechten voor gebruikers die niet de eigenaar van het bestand zijn, noch in de groep van het bestand zitten.

De owner, ook wel `user` genoemd, is degene van wie het bestand is; vaak degene die het bestand heeft aangemaakt.

Wat group is bespreken we in de sectie over gebruikers en groepen, elk bestand behoort aan precies 1 groep toe.

Others is "iedereen die niet owner of group is."

## Inzien van bestandspermissies

Met het linux-commando `ls` zie je welke bestanden er in een map staan. Met het linux-commando `ls -l` zie je "lange" informatie over bestanden.

Bijvoorbeeld:

```
[alice@contabo ~]$ ls -l
total 4
-rw-r--r-- 1 alice hackers 959 Jan 25 2022
readme.md
```

Het bestand `readme.md` heeft als permissies `-rw-r--r--`.

- Het eerste karakter `-` betekent dat het een "gewoon" bestand is (dus geen map of iets speciaals)
- `rw-` de eerste drie karakters zijn voor de owner of `user` van wie het bestand is: het bestand kan door alice worden gelezen en geschreven.
- `r--` de tweede drie karakters gaan over de groep-permissies, rechten voor iedereen in de groep `hackers`
- `r--` de laatste drie karakters gaan over de permissies van alle andere gebruikers die dus niet `alice` zijn of in de groep `hackers` zitten.

`alice` hierboven is de eigenaar van het bestand, en het bestand heeft `hackers` als groep.

## Wijzigen van bestandspermissies

De gebruiker `bob` zit ook in de groep `hackers` en mag van Alice ook dit bestand bewerken.

Alice kan met het programma `chmod` de permissies aanpassen; zie `man chmod` voor meer uitleg:

```
[alice@contabo ~]$ chmod g+w readme.md
[alice@contabo ~]$ chmod o-r readme.md
[alice@contabo ~]$ ls -l
total 4
-rw-rw---- 1 alice hackers 959 Jan 25 2022
readme.md
```

`chmod g+w` geeft schrijfrechten `w` aan de groep `hackers`.  
`chmod o-r` trekt de leesrechten van anderen dan `alice` zelf en de groep `hackers` in.

## Bestanden en mappen maken

Natuurlijk kun je in linux, net als in windows in een grafische omgeving bestanden en mappen maken, maar we richten ons hier op de command line.

Een nieuwe map aanmaken kan met `mkdir`:

```
[alice@contabo ~]$ mkdir hackers-geheimen
[alice@contabo ~]$ ls -l
total 8
```

```
drwx----- 2 alice alice 4096 Feb 5 23:30
hackers-geheimen
-rw-rw---- 1 alice hackers 959 Jan 25 2022
readme.md
```

De map binnengaan met `cd`:

```
[alice@contabo ~]$ cd hackers-geheimen/
[alice@contabo hackers-geheimen]$
```

Een nieuw bestand maken dat verder leeg is kan met `touch`:

```
[alice@contabo hackers-geheimen]$ touch deel-je-
geheim-hier.txt
[alice@contabo hackers-geheimen]$ ls -l
total 0
-rw----- 1 alice alice 0 Feb 5 23:32 deel-je-
geheim-hier.txt
```

De standaard-instelling voor permissies van bestanden die `alice` aanmaakt is erg beperkt: alleen `alice` zelf kan dit bestand lezen of schrijven.

Alice en Bob willen echter graag samenwerken aan het bestand dat ze net heeft aangemaakt. Daarvoor moeten we iets dieper duiken in de wereld van "groepen".

## Gebruikers en groepen

home-directory: de "thuishaven" in linux. Elke gebruiker heeft een home-directory waar die standaard in start na het

inloggen. Deze directory is eigendom van die gebruiker; die kan er bestanden en mappen in maken.

Het commando `cd` zonder argumenten brengt je terug naar die home-directory.

Alice en Bob willen graag samenwerken aan het `deel-je-geheimen.txt` document.

Dat bestand staat in de map `hackers-geheimen` in de home-directory van Alice.

Wat we willen bereiken:

1. De map `hackers-geheimen` heeft als groep `hackers`, zodat we voor alle gebruikers in die groep permissies kunnen uitdelen.
2. De map `hackers-geheimen` geeft Bob (en andere `hackers`) de permissie "er in te kunnen" (met `cd`).
3. Het bestand `deel-je-geheim-hier.txt` kan worden bewerkt door alle `hackers`
4. Niemand anders kan dat bestand lezen of bewerken.

## 1. Groep wijzigen van de map

De groep wijzigen van iets kan met het commando `chgrp`:

```
[alice@contabo ~]$ chgrp hackers hackers-geheimen/
[alice@contabo ~]$ ls -l
total 8
```

```
drwx----- 2 alice hackers 4096 Feb  5 23:32
hackers-geheimen
```

De map `hackers-geheimen` heeft nu `hackers` als groep.

## 2. Directory (map) toegankelijk maken

Om toegankelijk te zijn moet een map zowel leesbaar (`read` permissie) als uitvoerbaar (`execute` permissie) zijn.

Dit kan worden aangepast met `chmod`:

```
[alice@contabo ~]$ chmod g+rx hackers-geheimen/
[alice@contabo ~]$ ls -l
total 8
drwxr-x--- 2 alice hackers 4096 Feb  5 23:32
hackers-geheimen
```

TIP: Chmod kan ook werken met getalletjes; maar de letters vind ik (Merijn) duidelijker. Je past hier de groep-permissie aan (g) en met de + en - geef je aan of het wel/niet permissie is, dus `g+rx` betekent group krijgt read en execute recht. De letters laten ook alle andere "permissie-bits" staan.

## 3. Bestand bewerkbaar maken voor hackers

Dit doen we met alle commando's die we geleerd hebben:

```
[alice@contabo ~]$ cd hackers-geheimen/
[alice@contabo hackers-geheimen]$ ls -l
```

```
total 0
-rw----- 1 alice alice 0 Feb  5 23:32 deel-je-geheim-hier.txt
[alice@contabo hackers-geheimen]$ chgrp hackers
deel-je-geheim-hier.txt
[alice@contabo hackers-geheimen]$ chmod g+rw
deel-je-geheim-hier.txt
[alice@contabo hackers-geheimen]$ ls -l
total 0
-rw-rw---- 1 alice hackers 0 Feb  5 23:32 deel-je-geheim-hier.txt
```

Dit gaan we uittesten; we loggen in als Bob en proberen in de map van Alice te komen:

```
[bob@contabo ~]$ cd ~alice/hackers-geheimen
-bash: cd: /home/alice/hackers-geheimen:
Permission denied
```

Een probleem: we kunnen nog niet in die map, wat is er aan de hand?

- Bob zit in de groep hacker, dus dat is het probleem niet:

```
[bob@contabo ~]$ id
uid=1012(bob) gid=1012(bob)
groups=1012(bob),1013(hackers)
```

Het blijkt dat de hele keten van mappen vanaf de root ( / ) van het filesystem bereikbaar moet zijn, dus alice moet haar homedirectory ook permissie geven voor Bob:

```
[alice@contabo ~]$ chgrp hackers .
[alice@contabo ~]$ chmod g+rx .
[alice@contabo ~]$ ls -ld .
drwxr-x--- 3 alice hackers 4096 Feb  5 23:30 .
```

Nogmaals testen:

```
[bob@contabo ~]$ cd ~alice/hackers-geheimen/
[bob@contabo hackers-geheimen]$ ls -l
total 0
-rw-rw---- 1 alice hackers 0 Feb  5 23:32 deel-
je-geheim-hier.txt
[bob@contabo hackers-geheimen]$ echo "Bob vindt
Alice lief!" > deel-je-geheim-hier.txt
[bob@contabo hackers-geheimen]$ cat deel-je-
geheim-hier.txt
Bob vindt Alice lief!
[bob@contabo hackers-geheimen]$
```

Nu werkt het wel! En Bob kan het bestand lezen en bewerken zoals je hierboven kunt zien.

Eve die dit geheim graag wil weten, kan er niet bij omdat ze niet in de groep `hackers` zit:

Eve kan niet in de `hackers-geheimen` directory:

```
[eve@contabo ~]$ cd ~alice/hackers-geheimen
-bash: cd: /home/alice/hackers-geheimen:
Permission denied
[eve@contabo ~]$ cat ~alice/hackers-
geheimen/deel-je-geheim-hier.txt
```

En ook het bestand niet zien, zelfs als ze de naam weet:

```
cat: /home/alice/hackers-geheimen/deel-je-geheim-
hier.txt: Permission denied
[eve@contabo ~]$ id eve
uid=1013(eve) gid=1014(eve) groups=1014(eve)
```

## Samenvatting en take-aways

- Linux kent drie soorten permissies: `read`, `write` en `execute`.
- Bestanden hebben precies 1 eigenaar en 1 groep.
- In linux bestaan gebruikers die kunnen worden ondergebracht in groepen.
- Een map heet `directory` en moet `execute`-permissie hebben om erin te kunnen navigeren met `cd`.
- Met de commando's `chmod`, `chown` en `chgrp` kunnen we permissies en eigendom van documenten en mappen veranderen
- Daarmee heb ik een groep `hackers` gemaakt zodat `alice` en `bob` kunnen samenwerken, terwijl `eve`, die niet tot de groep `hackers` behoort niet kan meekijken.

Met dank aan `alice`, `bob` en `eve` voor de demonstratie!

## Bronnen

1. <https://www.linuxfoundation.org/blog/blog/classic-sysadmin-understanding-linux-file-permissions>

Geraadpleegd op 2023-02-05.

2. <https://www.linode.com/docs/guides/linux-users-and-groups/> Geraadpleegd op 2023-02-05.
3. `man chmod` de manpage van chmod, zoals op de linux machine van Merijn
4. `man chown` de manpage van chown, zoals op de linux machine van Merijn
5. `man chgrp` de manpage van chgrp, zoals op de linux machine van Merijn