

# Algemeen

- [Het find commando gebruiken om directories te laten zien](#)
- [Rechten toekennen met chmod](#)
- [TAR archief maken](#)
- [Tmux terminal multiplexer](#)

# Het find commando gebruiken om directories te laten zien

Onderstaand commando kan handig zijn om even snel de directories te laten zien in Linux:

```
find directory_path -maxdepth 1 -type d
```

Het aantal levels van de directories wat je wilt zien bepaal je met de parameter "-maxdepth".

# Rechten toekennen met chmod

Met het commando chmod kan je rechten toekennen en wijzigen op bestanden en directories. De syntax voor chmod is:

```
$ chmod options mode file
```

Meer informatie hierover kan je opvragen in de terminal met het commando:

```
man chmod
```

## Gebuikersklassen en chmod

In Linux heeft elk bestand op een server individuele toegangsrechten. Hetzelfde geldt voor directories. De toegangsrechten worden geregeld volgens **drie klassen van gebruikers**:

- **Eigenaar (user)**: een gebruiker die een bestand aanmaakt in Unix wordt gewoonlijk automatisch gedefinieerd als de 'user' van het bestand. Het eigendom kan later worden gewijzigd door het commando 'chown'. In de symbolische notatie krijgt de gebruikersklasse 'user' de letter 'u'.
- **Groep (group)**: de gebruikersklasse 'group' omvat verschillende gebruikersaccounts op de server. In Unix-bestandssystemen wordt elk gebruikersaccount automatisch toegewezen aan een hoofdgroep. Lidmaatschap in andere groepen is ook mogelijk. Zowel de eigenaar als de rootgebruiker kan bestanden groeperen met behulp van het 'chgrp' commando. De gebruikersklasse 'groep' wordt in de symbolische notatie met de letter 'g' weergegeven.
- **Andere (others)**: deze gebruikersklasse omvat alle gebruikers die geen user zijn van het bestand en ook niet bij een groep horen. Deze gebruikersklasse heeft de symbolische notatie 'o'.

## Toegangsrechten

Het Linux bestandssysteem heeft **drie basisrechten**. Elk van de bovengenoemde gebruikersaccounts kan verschillende rechten hebben die door de eigenaar van het bestand zijn toegewezen:

- **Lezen (read)**: het toegangsrecht 'read' geeft een gebruiker leestoegang tot een bestand. In het geval van een directory kan hij de hele inhoud van die map lezen. Maar met het read-toegangsrecht kun je geen bestandsrechten bekijken. In de symbolische notatie wordt dit recht gecodeerd met de letter 'r', ook wel r-bit genoemd.
- **Schrijven (write)**: gebruikers met het toegangsrecht 'write' mogen de inhoud van het bestand wijzigen. Als 'write' is toegewezen aan een directory, mag de gebruiker in die map submappen en bestanden aanmaken. Het schrijfrecht wordt in de symbolische

notatie weergegeven met een 'w'. Het heet daarom ook wel w-bit.

- **Uitvoeren (execute)**: een gebruiker met dit toegangsrecht mag bestanden als programma uitvoeren en naar een andere directory gaan om daar te kijken welke submappen of bestanden er staan. De symbolische notatie kent aan dit recht de letter 'x' toe, ofwel het x-bit.

Denk er bij het toekennen van toegangsrechten aan dat **machtigingen in Linux niet worden geërfd**. Ofwel, als je een bestand in een directory aanmaakt, krijgt dat bestand niet automatisch de toegangsrechten die voor de hele directory gelden. Individuele bestandsrechten vloeien voort uit de machtiging van het programma dat het bestand heeft aangemaakt.

## Weergave van toegangsrechten in de terminal

Met het volgende commando kan je de bestanden en directories weergeven en de bijbehorende rechten zien:

```
ls -l
```

Hier onder zie je een voorbeeld:

```
alex@alex-Lenovo-G500:~/test-chmod$ ls -l
totaal 12
drwxr-xr-x 2 alex alex 4096 mei 4 13:47 dir01
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir02
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir03
-rw-r--r-- 1 alex alex 0 mei 4 13:47 file01.txt
```

De volgende illustratie laat duidelijk zien hoe de rechten in elkaar steken:

[chmod01.jpg](#)

De rwx (read, write, execute) parameters kan je aanpassen voor de **eigenaar (user)**, **groep (group)** en **anderen (others)**.

Dit is het zogenaamde rechtenmasker en bestaat uit een **octale notatie** van drie cijfers. Onderstaande illustratie maakt dit duidelijk:

[chmod02.png](#)

Ieder cijfer bestaat dus uit drie bits en kan dus lopen van 0 t/m 7. Op deze manier kan je de rechten toekennen.

## Voorbeelden:

Stel dat we in ons voorbeeld de rechten willen wijzigen van de directory "dir01", zodanig dat iedereen alles mag, dan kan dat als volgt:

```
chmod 777 dir01
```

Stel dat we in ons voorbeeld de rechten willen wijzigen van het bestand “file01.txt”, zodanig dat alleen de gebruiker kan lezen en schrijven, dan kan dat als volgt:

```
chmod 600 file01.txt
```

Stel dat we alleen de rechten van **alle onderliggende directories willen wijzigen** en niet van de bestanden:

```
find /home/alex/test-chmod/ -type d -exec chmod 755 {} \;
```

Stel dat we alleen de rechten van **alle onderliggende bestanden** willen wijzigen en niet van de directories:

```
find /home/alex/test-chmod/ -type f -exec chmod 644 {} \;
```

## Symbolische notatie

In plaats van de octale notatie kan men ook de rechten wijzigen door een symbolische notatie. Zie ook onderstaande illustratie:

[chmod03.png](#)

### Voorbeelden:

Stel dat we in ons voorbeeld de rechten willen wijzigen van de directory “dir01”, zodanig dat iedereen alles mag, dan kan dat als volgt:

```
chmod ugo+rwx dir01
```

Onderstaand commando heeft hetzelfde effect:

```
chmod a+rwx dir01
```

Op deze manier kan men ook bepaalde rechten weer verwijderen:

```
chmod o-rwx dir01
```

## Sticky bit:

Met sticky bit kan je de rechten op mappen en bestanden beperken. Als je een map gedeeld hebt met andere gebruikers, maar je wilt niet dat die gebruikers bestanden kunnen verwijderen of

hernoemen, dan kan je de sticky bit aanzetten om dit te voorkomen. De rechten van de gebruikers worden daarmee ingeperkt. Alleen de eigenaar en de root gebruiker hebben dan nog wel alle rechten. Gebruikers kunnen dan nog wel bestanden toevoegen. Sticky bit wordt enkel toegepast op directories en niet op bestanden. Wanneer de sticky bit aan staat voor een directory, zijn de bestanden in die directory ook beschermd.

Het toepassen van de sticky bit is eenvoudig, hiervoor wordt de **t** vlag gebruikt met **chmod**:

## Voorbeelden:

In onderstaand voorbeeld wordt de map "dir01" beschermd door het sticky bit aan te zetten:

```
chmod +t dir01
```

Uitzetten kan dan dus weer met:

```
chmod -t dir01
```

Je kunt zien wanneer er een sticky bit is gezet op een directory. Dit wordt aangegeven met de letter "t" in het laatste octet:

```
alex@alex-Lenovo-G500:~/test-chmod$ ls -l
totaal 12
drwxrwxrwt 2 alex alex 4096 mei 4 13:47 dir01
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir02
drwxr-xr-x 2 alex alex 4096 mei 4 13:48 dir03
-rw----- 1 alex alex 0 mei 4 13:47 file01.txt
```

Stel dat je nu de sticky bit wilt aan zetten voor alle onderliggende directories, dan kan dat als volgt:

```
find /home/alex/test-chmod/ -type d -exec chmod +t {} \;
```

Uitzetten kan dan weer met:

```
find /home/alex/test-chmod/ -type d -exec chmod -t {} \;
```

# TAR archief maken

Met behulp van onderstaand commando kan men een archiefbestand maken:

```
tar --exclude='db.*' -zcvf vaultwarden-$(date +%d-%m-%Y_%H-%M-%S).tar.gz  
/opt/dockerdata/vaultwarden
```

In bovenstaand voorbeeld wordt er van mijn Vaultwarden directory (/opt/dockerdata/vaultwarden) een archiefbestand gemaakt waarin ook de datum en tijd verwerkt zijn in de bestandsnaam. De optie "--exclude=db.\*" zorgt ervoor dat de bestanden die beginnen met "db." uitgesloten worden in het archief (in dit geval de Sqlite3 database bestanden).

Wanneer je veel bestanden hebt die je wilt uitsluiten kan je ook verwijzen naar een exclude file. In deze file zet je dan per regel het bestand of directory in die je wilt uitsluiten. Het tar commando ziet er dan als volgt uit:

```
tar -zcvf vaultwarden-$(date +%d-%m-%Y_%H-%M-%S).tar.gz -X exclude_file.txt  
/opt/dockerdata/vaultwarden
```

De parameter -X verwijst naar de exclude file, in ons voorbeeld exclude\_file.txt. Je mag ook de volledige parameter opgeven, dit is --exclude-from=exclude\_file.txt

# Tmux terminal multiplexer

## Installing tmux in Debian

```
sudo apt-get install tmux
```

To start tmux, just type it and hit Enter:

```
tmux
```

## A Ctrl+B Cheat Sheet

We've included a cheat sheet of the different commands you can use in `tmux` below.

### Session Commands

- **S** : List sessions.
- **\$** : Rename current session.
- **D** : Detach current session.
- **Ctrl+B, and then ?** : Display Help page in `tmux`.

### Window Commands

- **C** : Create a new window.
- **,** : Rename the current window.
- **W** : List the windows.
- **N** : Move to the next window.
- **P** : Move to the previous window.
- **0 to 9** : Move to the window number specified.

### Pane Commands

- **%** : Create a horizontal split.
- **"** : Create a vertical split.
- **H or Left Arrow**: Move to the pane on the left.
- **I or Right Arrow**: Move to the pane on the right.
- **J or Down Arrow**: Move to the pane below.
- **K or Up Arrow**: Move to the pane above.
- **Q** : Briefly show pane numbers.
- **O** : Move through panes in order. Each press takes you to the next, until you loop through all of them.
- **}** : Swap the position of the current pane with the next.
- **{** : Swap the position of the current pane with the previous.
- **X** : Close the current pane.

**BRON:** [How to Use tmux on Linux \(and Why It's Better Than Screen\)](#)