

3.2 File and Directory permissions

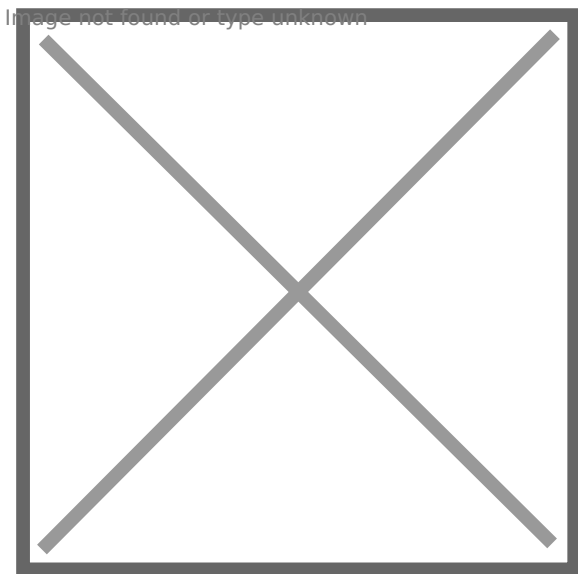
Voor deze les moet je weten hoe je door directory's kunt navigeren en hoe je een file kunt aanmaken en kan editen. Als je niet weet hoe dat werkt dan moet je eerst terug naar de vorige les.

In deze les gaan we een wat lastigere onderwerp behandelen, rechten binnen Linux.

In de opgaven gaan we zelf oefenen. Veel opgaven worden in de filmpjes voorgedaan.

Inleiding

Het Unix file systeem is een hiërarchisch systeem, net als bij Windows. Dat betekent dat er ergens een begin is; onder windows is dat (bijna altijd) c:\ en onder Linux is dat /



Rechten worden gegeven op directory's of files. Elke file of directory heeft een eigenaar/user en een groep/group. Er zijn drie soorten rechten Read, Write en eXecute. Deze drie rechten kunnen aan de user, group of aan iedereen (other) worden gegeven.

<https://www.youtube.com/embed/xrOfPVevCGo>

Owner en Group

Files en directories hebben een owner (ook wel user) en een group. De owner is meestal degene die de file heeft aangemaakt. De rechten van de owner en de group kunnen apart worden ingesteld.

RWX - Read, Write en Execute

We kennen drie soorten rechten: Read, Write en eXecute (RWX). Read is voor lezen, write voor aanpassen/schrijven en execute betekent dat je een file mag uitvoeren. Voor een directory betekent de execute dat je naar de directory toe mag gaan.

ls -l

Als je een `ls -l` doet dan zie je bijvoorbeeld

```
drwxrwxr-x  17 max    apache 4096 Nov 27 22:03 www
```

De drwxrwxrwx geven de file permissions (rechten) aan, daarover later meer). De 17 geeft het aantal links aan, dit is voor nu niet belangrijk. 'max' is de owner, ook wel user van de file. apache is de group waartoe de file behoort. 4096 is het aantal bytes dat de file in beslag neemt (dat is niet helemaal hetzelfde als de grootte van de file), dan volgt de 'last modification date' en als laatste zie je de file name.

De owner kan bijvoorbeeld Read and Write-rechten hebben en leden van de group kunnen bijvoorbeeld alleen Read-rechten hebben. Je kunt tenslotte ook nog de rechten van 'iedereen' (other) instellen, dat zijn alle gebruikers die niet de owner zijn en niet in de group zitten.

Image result for linux owner and group

Image not found or type unknown

Voor elke file en directory staan 10 letters, we kijken naar de laatste 9. Dit zijn 3 groepjes van 3; rwx, rwx en rwx. De eerste rwx staat voor de owner/user van de file, de tweede voor de group en de laatste voor others.

Stel je ziet dit als je een `ls -l` uitvoert:

```
-rw-r--r--  1 root root          52 Jan 11 18:18 subuid
```

Dat betekent dat de file *subuid* van root is (owner/user) en dat die ook aan de group root toebehoort. De owner root heeft Read en Write rechten en de groep heeft alleen read rechten. Ten slotte hebben others ook read rechten.

File permissies instellen

Met het commando `chmod` (CHange file MODe bits) kunnen de rechten worden ingesteld van de User, Group en Others. Stel je wilt de rechten instellen van de User en de Group dan gebruik je `UG`. Stel je wilt de rechten instellen van de User, Group en Others dan gebruik je `UGO`.

Dan kun je rechten toevoegen of afnemen, dat doe je met een `+` of een `-`.

Je kunt ook rechten instellen dat doe je met een `=`

Vervolgens kun je read, write en execute rechten toekennen of afnemen. Read en write toevoegen doe je met `+rx`. Afnemen van alle rechten doe je met `-rwx`.

Het zetten van bijvoorbeeld alleen lees rechten doe je met `=r`

Voorbeelden chmod

Afnemen van read rechten van others voor file_a	<code>chmod o-r file_a</code>
Toevoegen van read en write rechten aan user en group voor alle files in deze directory	<code>chmod ug+rw *</code>
Toevoegen van alle rechten aan iedereen voor alle files in deze directory en alles 'hieronder'	<code>chmod -R ugo+rwx</code>
Instellen van alleen leesrechten voor de group op alle files in deze directory	<code>chmod g=r *</code>

Recursief

In het laatste commando is de flag `-R` toegevoegd. Dit betekent recursief en dat betekent dat alle files en alle directory's worden veranderd. Directory's in directory's worden ook veranderd, ook de directory's in de directory's in de directory's, etc. worden veranderd.

File permissies instellen - octaal

In plaats van rechten toekennen in meerdere stappen kan je dat ook in één doen. De `r` is 4, de `w` is 2 en de `x` is 1. Dus als je `rw-` wilt instellen dan gebruik je het `4+2=6`. Als je `r-x` wilt instellen gebruik je `4+1=5`. Dat doe je dan drie keer voor de user, de group en others. `rwx` wordt `4+2+1=7`. Als de user, group en others allemaal `+rwx` willen geven dan kan dat met een `chmod 777`.

Read	4
Write	2
Execute	1

Het getal dat je gebruikt is een getal in het 8-tallig stelsel, we noemen dat het octale stelsel.

<https://i.redd.it/vkxuqbatopk21.png>

Image not found or type unknown

$r=4$, $w=2$ en $x=1$. Dus $rwX=4+2+1=7$

Nog meer voorbeelden

geef alleen alle rechten op file01 aan owner	chmod u+rwX file01 chmod go-rwX file01	chmod 700 file01
geef owner, groep en others alle rechten	chmod ugo+rwX file01	chmod 777 file01
geef alleen de owner en de groep leesrechten op file01	chmod ug+r file01 chmod ug-wX file01 chmod o-rwX file01	chmod 440 file01

User en groups aanpassen

Met chown kun je de user/owner aanpassen en met chgrp kun je de group aanpassen.

chown	CHange Owner
chgrp	CHange GRoup

chown voorbeelden

maak root owner van file file01	chown root ./file01	
maak apache owner van alle file in /var/www	chown -R apache /var/www	-R is recursief

van de file01 maar root owner en voeg de group root ook toe	chown root:root ./file01	
---	--------------------------	--

In het laatste voorbeeld zie je dat je de user en group ook in één keer kunt aanpassen.

chgrp voorbeeld

Geeft de group root rechten op file02	chgrp root ./file02	
---------------------------------------	---------------------	--

Opdracht 1

- Ga naar jouw home directory met `cd ~`
- Maak een directory en noem die test
- Ga naar die directory en maak drie files: file01, file02 and file03.
- Zorg dat de alleen de owner (jij) de files kan lezen en schrijven en de groep alleen kan lezen. Others kunnen niets:

```
-rw-r----- 1 max max 0 Jan 11 19:45 file01
-rw-r----- 1 max max 0 Jan 23 08:11 file02
-rw-r----- 1 max max 0 Jan 23 08:11 file03
```

Schrijf de commando's die je hebt gebruikt op.

- Zet de groep van de drie files gelijk aan 'root' (zie afbeelding).
Het is soms mogelijk dat je geen root group hebt in dat geval kan je ook adm gebruiken als group. Met het groups commando kun je controleren welke groepen je hebt.

```
-rw-r----- 1 max root 0 Jan 11 19:45 file01
-rw-r----- 1 max root 0 Jan 23 08:11 file02
-rw-r----- 1 max root 0 Jan 23 08:11 file03
```

Schrijf de commando's die je hebt gebruikt op.

- Ga terug naar jouw home directory, dus daar waar de directory test in staat.
- Geef de directory test de naam test01.
Schrijf het commando dat je hebt gebruikt op.
- Geef iedereen (owner, group en others) alle rechten (rwx) op de directory test01

Schrijf het commando dat je hebt gebruikt op.

- Zoek op hoe je de directory `test01` moet verwijderen en verwijder deze.

Opdracht 2

Voor deze opdracht moet je de les hebben gelezen en ik raad ook aan om het volgende filmpje te bekijken. Veel antwoorden zijn te vinden in de uitleg in het filmpje.

<https://www.youtube.com/embed/5z-kviCTjGE>

Ga naar jouw home directory en maak de file `struct.txt` en zet daar het volgende in.

(Tip: je kunt *nano* gebruiken om de inhoud van de file `struct.txt` te wijzigen)

```
shared
shared/projects/01
shared/projects/02
shared/projects/03
shared/series
shared/movies
shared/movies/action
```

Voer nu het volgende commando uit:

```
sed '/^$/d;s/ /\//g' struct.txt | xargs mkdir -p
```

Magic! Je hebt nu een directory structuur die begint in de folder *shared* en die de structuur heeft zoals je in de file `struct.txt` hebt opgegeven. Hoe dit werkt precies werkt, voert nu te ver. We gaan de nieuwe directory structuur wel gebruiken om de volgende opgaven te maken.

a) verander de group van alle directory's die in *shared* staan inclusief *shared* in *adm*.
Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

b) verander de rechten van alle directory's vanaf *shared* (inclusief *shared*) naar alleen lezen voor de group.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

c) verander de rechten van alleen de directory *shared* zodat de group ook schrijfrechten (write) heeft.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

d) verander de rechten van alle directory's die in *shared* staan inclusief *shared* zodat de group alleen execute rechten heeft.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

e) verander de rechten van de directory *shared* zodat iedereen *user*, *group* en *owner* alleen lees- en schrijfrechten (rw) hebben.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

f) ga met `cd shared` naar *shared*, wat gebeurt er? Waarom heb je geen toegang?

g) verander de rechten van de directory *shared* zodat iedereen *user*, *group* en *owner* alle rechten heeft (rwx) heeft.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

h) met het commando `rm` kan je alle directory's weer verwijderen. Maak gebruik van de man pagina's en zoek uit hoe je de directory *shared* inclusief alle sub directory's kan verwijderen.

Schrijf het commando dat je hebt gebruikt op je antwoordenblad.

Opdracht 3

Maak een file `execute.sh` en zet daar het volgende in.

```
echo "Dit is een test"
```

Voer dit commando uit.

Wat heb je moeten doen om deze file te kunnen uitvoeren?

Schrijf het antwoord (het commando dat je moest gebruiken om deze file te kunnen uitvoeren) op je antwoordblad op.

--