

UNIX File Permissions

Computersystemen 1 – Linux

Inleiding

Elk bestand en directory op een unix systeem heeft een eigenaar (owner), groep (group) en een set van rechten (permissions). Je kan deze opvragen met het commando `ls -l`, eventueel gevolgd door de bestandsnaam.

```
user@host:~$ ls -l /etc/services
-rw-r--r-- 1 root root 19605 okt 25 2014 /etc/services
```

Permissions **Owner** **Group**

Permissions

De permissions (of rechten/permissions) van een bestand vertellen ons wie het bestand kan lezen, aanpassen of uitvoeren.

Op Unix systemen wordt er een onderscheid gemaakt tussen 3 scopes of klassen van permissies: de permissies voor de *owner* (eigenaar), de permissies voor de *group* (groep) en de permissies voor *others* (alle anderen).

Ook directories hebben permissies waar de betekenis iets anders is tov bestanden:

Permissie	Effect op bestand	Effect op directory
read	Kan het bestand openen, de inhoud tonen	Kan de inhoudslijst van een directory weergeven
write	Kan de inhoud van het bestand aanpassen	Kan in deze directory bestanden aanmaken, hernoemen of verwijderen
execute	Kan het bestand uitvoeren	Kan de directory browsen en details over de inhoud bekijken (permissions, owners, times, ...)

Permissies worden net zoals de owner, group, access/creation/modification time, ... bijgehouden in de bijhorende inode van het bestand. We kunnen deze permissies op verschillende manieren weergeven of noteren.

Symbolische notatie

De symbolische notatie zien we in de output van `ls -l` en bestaat uit 10 karakters met een vaste positie en betekenis. Het eerste karakter bepaalt het type van het bestand, de andere karakters zijn *flags* die aan of uitstaan.

-rw-r--r--

We kunnen deze symbolische notatie opsplitsen in 4 stukken of groepjes waarbij het eerste karakter/groepje het type van het bestand aangeeft en de 3 andere groepjes telkens al dan niet (-) read, write of execute rechten :

- rw- r-- r--

Type

Bestand (-)
Directory (d)
Link (l)
Block device (b)
Character special device (c)

Owner Permissions

Bepalen de rechten (read, write, execute) van de *owner* van deze file / directory

Group Permissions

Bepalen de rechten van de *group* van deze file / directory

Others Permissions

Bepalen de rechten van *others* – iedereen die niet tot *owner* of *group* behoort – van deze file / directory

Deze voorbeeldpermissies betekenen dus:

- **Dit is een bestand**
- **De owner heeft read & write rechten (geen execute)**
- **De group heeft read rechten (geen write of execute)**
- **Al de rest heeft ook read rechten (geen write of execute)**

Andere voorbeelden

Symbolisch	Verklaring
-----	Een bestand waar niemand rechten op heeft
-rwxrwxrwx	Een bestand waar iedereen alle rechten op heeft
brw-rw----	Block device met lees- en schrijfrechten voor owner & group
drwxr-xr-x	Directory waarvan inhoud voor iedereen zichtbaar is, maar enkel owner kan er bestanden in aanmaken, verwijderen, ...
crw-rw-rw-	Character device: iedereen kan lezen en schrijven

Octale Notatie

Een alternatieve notatie voor permissies is de octale notatie waarbij we de 3 klassen of groepjes (*owner*, *group* & *others*) een cijfer geven van 0 tot 7 (vandaar octaal) om aan te duiden welke permissies ze krijgen.

Om dat cijfer per groepje te bepalen, tellen we gewoon de waarden van hun permissies op volgens de tabel hieronder:

Permissie	Waarde
read	4 (100 binair)
write	2 (010 binair)
execute	1 (001 binair)

Op die manier kunnen we permissies ondubbelzinnig noteren als een getal van 3 cijfers. Bijvoorbeeld `drwx-r-xr-x` kunnen we octaal schrijven als 755.

- owner heeft read, write & execute: $4+2+1 = 7$ (of 111 binair)
- group heeft read & execute: $4+1 = 5$ (of 101 binair)
- others hebben read & execute: $4+1 = 5$ (of 101 binair)

De octale notatie is minder visueel maar korter (en dus sneller) om permissies te beschrijven.

Voorbeelden

Symbolisch	Octaal
-----	000
-rwxrwxrwx	777
brw-rw----	660
drwxr-xr-x	755
crw-rw-rw-	666
-rw-r--r--	644

Owner, group & permissions aanpassen

Owner en group aanpassen

Om de owner of group van een bestand of directory aan te passen gebruiken we de commando's `chown` (change owner) en `chgrp` (change group)

```
chown root funnycat.avi
```

Zal de owner van het bestand `funnycat.avi` veranderen naar `root`

```
chown pippa:users text3
```

Zal voor het bestand `text3` de owner aanpassen naar de gebruiker `pippa`, en de group aanpassen naar de group `users`.

```
chgrp admins sshd.log
```

Zal de group van het bestand `sshd.log` aanpassen naar `admins`, een alternatief hiervoor is `chown :admins sshd.log`

Een vaak gebruikte optie met `chown` of `chgrp` is `-R`, dit zorgt ervoor dat het commando de nieuwe owner of group recursief toepast (op de gegeven directory en alle onderliggende bestanden en directories)

```
chown -R kvothe:wind /home/kvothe
```

Zorgt ervoor dat `/home/kvothe` en alle onderliggende bestanden en directories de owner `kvothe` en group `wind` krijgen.

Permissies aanpassen

Om de permissies van een bestand of directory aan te passen gebruiken we het commando `chmod` (change file mode bits)

```
chmod o+rx bestand1
```

Geeft read & execute permissies op `bestand1` aan *others* (*o*)

```
chmod g-w bestand1
```

Neemt write permissies op `bestand1` weg van de *group* (*g*)

```
chmod a+rwX bestand2
```

Geeft read, write & execute permissies op `bestand2` aan iedereen/all (*owner*, *group* & *others*)

```
chmod u+rwx,g-w,o-w bestand3
```

Geeft read, write & execute permissies op bestand3 aan de *owner* (*o*), neemt write permissies weg van *group* (*g*) en *others* (*o*)

```
chmod u=rw,g=r,o=r bestand4
```

Zet de permissies op een absolute manier, de oude permissies worden volledig overschreven. Dit is analoog met:

```
chmod 644 bestand4
```

Je merkt dat we zowel symbolische als octale notatie kunnen gebruiken om permissies te wijzigen. Ook kunnen we permissies op een absolute manier of een relatieve manier wijzigen.

Als we de huidige permissies als basis nemen en daar permissies aan toevoegen of verwijderen (chmod in combinatie met een + of -) dan noemen we dat **relatief**.

Als we de huidige permissies niet als basis nemen, maar een volledig nieuwe set permissies geven, dan noemen we dat **absoluut**. (chmod in combinatie met = of octale notatie)

Speciale permissies

Bovenop de gewone (rwx) permissies zijn er ook 3 speciale permissies:

Speciale Permissie	Symbool	Effect op bestand	Effect op directory
SUID (4) <i>set user id</i>	s	Bestand wordt uitgevoerd met effectieve UID van de owner van dat bestand	-
SGID (2) <i>set group id</i>	s	Bestand wordt uitgevoerd met effectieve GID van de group van dat bestand	Wanneer een bestand wordt aangemaakt in deze directory, zal de groep van het bestand gezet worden op de group van deze directory
Sticky Mode (1)	t	<i>Niet meer van toepassing</i>	In een directory met sticky bit of mode kunnen bestanden enkel hernoemd, verwijderd of verplaatst worden door de owner van dat bestand, ongeacht de andere permissies

Symbolische notatie speciale permissies

Bij SUID wordt een s geplaatst op de x positie van de *owner*

Bij SGID wordt een s geplaatst op de x positie van de *group*

Bij sticky mode wordt een t geplaatst op de x positie van *others*

Voorbeelden:

-rwsr-xr-x	SUID Bestand
drwxrwsr-x	SGID Directory
drwxrwxrwt	Sticky directory
drwxrwsrwt	SGID & Sticky directory

Doordat die s'en of t's op de positie van de x komen te staan, kunnen we niet meer zien of het bestand uitvoerbaar is (voor die klasse) of niet. Daarom wordt er gebruik gemaakt van een kleine s of t als het bestand uitvoerbaar is, en een hoofdletter S of T als het bestand niet uitvoerbaar is.

Octale notatie speciale permissies

Voor de octale notatie gebruiken we een 4de cijfer dat we voor de andere cijfers zetten, bv:

Octaal	Speciale permissies
0644	Geen
4755	SUID
2775	SGID
1777	Sticky
3777	SGID & Sticky

Het tweede, derde en vierde cijfer blijft gewoon z'n octale betekenis houden.

Als bij octale notatie het eerste cijfer een 0 is mogen we het ook weglaten, 0644 of 644 zijn dus gelijk aan elkaar.

Voorbeelden voor het aanpassen van speciale permissies:

```
chmod u+s script.sh  
Maakt script.sh SUID
```

```
chmod 1777 /opt  
Geeft alle permissies aan iedereen op /opt en maakt /opt sticky
```