

HoGent

BEDRIJF
EN
ORGANISATIE

Besturingssystemen

Labo 5

Wie heeft welke rechten?

Elke file en elke directory heeft een owner en een group

Permissions

Group

Date & Time
Last Modified

Name

File type

Owner

-rwxrwxrwx
-r-----

1 linfo linfo
1 linfo linfo

26 Dec 9 14:36 .htaccess
27 Dec 9 14:35 test.php

Owner

Group

Others

first character	file type
-	normal file
d	directory
l	symbolic link
p	named pipe
b	block device
c	character device
s	socket

Wat betekenen die rechten?

permission	on a file	on a directory
r (read)	read file contents (cat)	read directory contents (ls)
w (write)	change file contents (vi)	create files in (touch)
x (execute)	execute the file	enter the directory (cd)

Hoe kan je die rechten instellen?

Permissions can be changed with **chmod**. The first example gives the user owner execute permissions.

chmod **wie** krijgt **wat**
chmod **u+x**

```
paul@laika:~/perms$ ls -l permissions.txt
-rw-r--r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
paul@laika:~/perms$ chmod u+x permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwxr--r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

This example removes the group owners read permission.

chmod **g-r**

```
paul@laika:~/perms$ chmod g-r permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwx---r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

This example removes the others read permission.

chmod **o-r**

```
paul@laika:~/perms$ chmod o-r permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwx----- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

This example gives all of them the write permission.

chmod **a+w**

```
paul@laika:~/perms$ chmod a+w permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwx-w--w- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Hoe kan je die rechten instellen?

You don't even have to type the a.

```
paul@laika:~/perms$ chmod +x permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwx-wx-wx 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

chmod **wie** krijgt **wat**
chmod **+x**

You can also set explicit permissions.

```
paul@laika:~/perms$ chmod u=rw permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rw--wx-wx 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

chmod **u=rw**

Feel free to make any kind of combination.

```
paul@laika:~/perms$ chmod u=rw,g=rw,o=r permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rw-rw-r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

chmod **u=rw,g=rw,o=r**

Octale notatie (old school notation)

- ▶ In plaats van rwx kunnen we de octale notatie gebruiken

Binair	Octaal	Permissie	Representatie
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

- ▶ Je kan chmod ook met octale notatie gebruiken om de permissies te wijzigen

```
paul@laika:~/perms$ chmod 777 permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwxrwxrwx 1 paul paul 0 2007-02-07 22:34 permissions.txt
paul@laika:~/perms$ chmod 664 permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rw-rw-r-- 1 paul paul 0 2007-02-07 22:34 permissions.txt
paul@laika:~/perms$ chmod 750 permissions.txt
paul@laika:~/perms$ ls -l permissions.txt
-rwxr-x--- 1 paul paul 0 2007-02-07 22:34 permissions.txt
```

Wat zijn de default permissies?

- ▶ De default permissies worden bepaald door umask. Dit specificeert de permissies die je by default **niet** wil instellen

Een bestand is by default niet executable. Je moet gebruik maken van `chmod +x` om een file executable te maken

```
[Harry@RHEL4b ~]$ umask
0002
[Harry@RHEL4b ~]$ touch test
[Harry@RHEL4b ~]$ ls -l test
-rw-rw-r-- 1 Harry Harry 0 Jul 24 06:03 test
[Harry@RHEL4b ~]$
```

- ▶ Je kan gebruik maken van `mkdir -m` om de permissies van een directory in te stellen bij het creëren

```
paul@debian5~$ mkdir -m 700 MyDir
paul@debian5~$ mkdir -m 777 Public
paul@debian5~$ ls -dl MyDir/ Public/
drwx----- 2 paul paul 4096 2011-10-16 19:16 MyDir/
drwxrwxrwx 2 paul paul 4096 2011-10-16 19:16 Public/
```


Hoe kan je owner en group wijzigen?

- ▶ **chgrp**: de group van een file wijzigen

```
root@laika:/home/paul# touch FileForPaul
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root root 0 2008-08-06 14:11 FileForPaul
root@laika:/home/paul# chgrp paul FileForPaul
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root paul 0 2008-08-06 14:11 FileForPaul
```

- ▶ **chown**: de owner van een file wijzigen

```
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root paul 0 2008-08-06 14:11 FileForPaul
root@laika:/home/paul# chown paul FileForPaul
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 paul paul 0 2008-08-06 14:11 FileForPaul
```

- ▶ **chown**: de owner en de group van een file tegelijk wijzigen

```
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 paul paul 0 2008-08-06 14:11 FileForPaul
root@laika:/home/paul# chown root:project42 FileForPaul
root@laika:/home/paul# ls -l FileForPaul
-rw-r--r-- 1 root project42 0 2008-08-06 14:11 FileForPaul
```

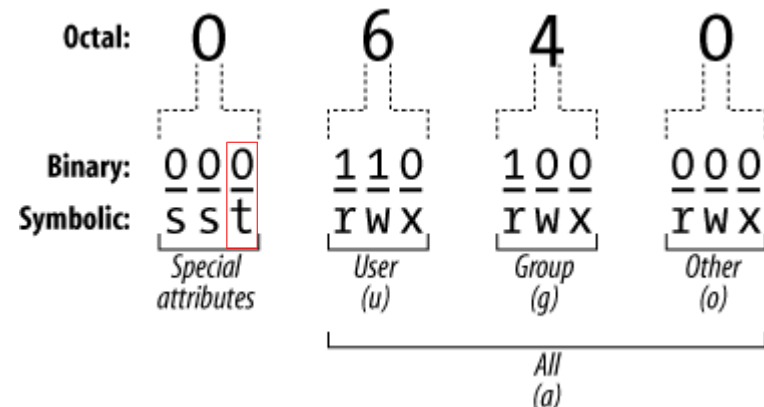
Sticky bit on directory

- ▶ Sticky bit on directory: om ervoor te zorgen dat een bestand enkel kan verwijderd worden door de eigenaar
- ▶ De sticky bit wordt getoond op dezelfde plaats als de x – permissie voor others.
 - t => x is ook aanwezig
 - T => x is niet aanwezig

```
root@RHELv4u4:~# mkdir /project55
root@RHELv4u4:~# ls -ld /project55
drwxr-xr-x  2 root root 4096 Feb  7 17:38 /project55
root@RHELv4u4:~# chmod +t /project55/
root@RHELv4u4:~# ls -ld /project55
drwxr-xr-t  2 root root 4096 Feb  7 17:38 /project55
root@RHELv4u4:~#
```

- ▶ De sticky bit kan ook ingesteld worden met octale permissies

```
root@RHELv4u4:~# chmod 1775 /project55/
root@RHELv4u4:~# ls -ld /project55
drwxrwxr-t  2 root root 4096 Feb  7 17:38 /project55
root@RHELv4u4:~#
```



setgid bit on directory

- ▶ setgid: zorgt ervoor dat de group owner van alle files in de directory = de group owner van de directory
- ▶ De setgid bit wordt getoond op dezelfde plaats als de x – permissie voor group.
 - s => x is ook aanwezig
 - S => x is niet aanwezig

```
root@RHELv4u4:~# groupadd proj55
root@RHELv4u4:~# chown root:proj55 /project55/
root@RHELv4u4:~# chmod 2775 /project55/
root@RHELv4u4:~# touch /project55/fromroot.txt
root@RHELv4u4:~# ls -ld /project55/
drwxrwsr-x  2 root proj55 4096 Feb  7 17:45 /project55/
root@RHELv4u4:~# ls -l /project55/
total 4
-rw-r--r--  1 root proj55 0 Feb  7 17:45 fromroot.txt
root@RHELv4u4:~#
```

setuid op regular files

- ▶ Het instellen van setuid zorgt ervoor dat een executable file wordt uitgevoerd met de permissies van de eigenaar van het bestand in plaats van met de permissies van degene die het bestand uitvoert
- ▶ Bijvoorbeeld als setuid is ingesteld voor een executable file van de root, dan kan een gewone gebruiker het bestand ook uitvoeren (als was hij root)
- ▶ Bijvoorbeeld
 - Paswoorden worden opgeslagen in /etc/shadow. Dit bestand kan enkel gelezen worden door de root

```
root@RHELv4u4:~# ls -l /etc/shadow
-r----- 1 root root 1260 Jan 21 07:49 /etc/shadow
```

- Als een gewone gebruiker zijn paswoord aanpast, moet dit bestand aangepast worden. Hoe kan een non-root gebruiker dit doen? Je gebruikt de credentials van de root

```
root@RHELv4u4:~# ls -l /usr/bin/passwd
-r-s--x--x 1 root root 21200 Jun 17 2005 /usr/bin/passwd
```

setgid op regular files

- ▶ Het instellen van de setgid bit op een executable file zorgt ervoor dat de executable file wordt uitgevoerd met de credentials van de group owner