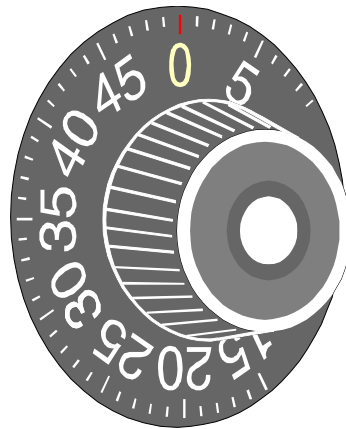


13 Filrettigheder

Filrettigheder



13 Filrettigheder

Rettigheder på filer og kataloger

- En listning (`ls -l`) af en fil viser bl.a. rettighederne for de 3 brugerkategorier: **Ejer**, **Gruppe** og **Alle andre**.

`-rw-rw-r-- 1 bruger bruger 1165 Nov 20 09:07 fil`

<code>rwX</code>	<code>rwX</code>	<code>rwX</code>
User	Group	Other

Read, **W**rite, **eX**ecute og **-**.

- En *vandret streg* svarer til ingen rettighed på det pågældende område for for brugerkategorien.
- Man kan **godt slette sin egen fil**, også selvom man ikke har skriverettighed til den! (`rm -f`)



13 Filrettigheder

Rettigheder på kataloger

- Rettigheder på et katalog har en anden betydning:

READ: Man kan få vist lang oplysningslinje om selve kataloget, men ikke om indholdet!

WRITE: Man kan oprette/slette filer her

EXECUTE: Man kan skifte til katalog (med *cd*).

For katalogers vedkommende kaldes

x -rettigheden oftest for *søgerettighed*.

r- og **w**-rettighederne på et katalog er ikke meget værd **uden x**-rettigheden.

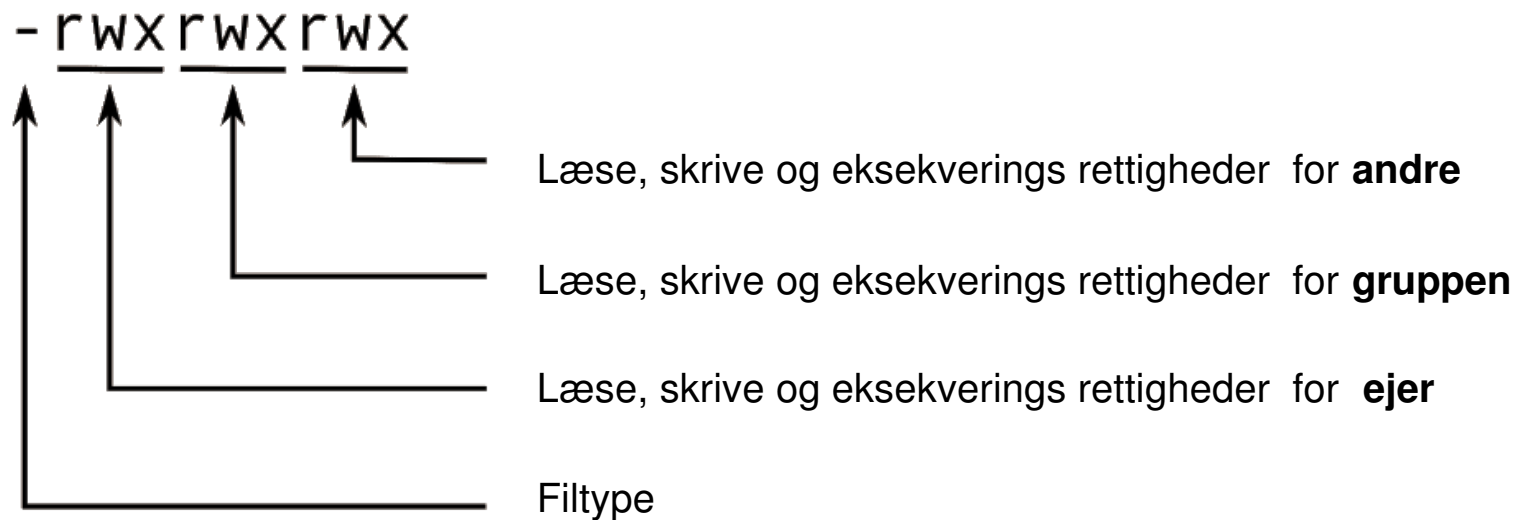
At lave en lang listning (*ls -l*) af et katalog kræver for eksempel både **r** - og **x**-rettighed.



0 Master

Sådan læses rettighederne:

-rw-rw-r-- 1 bruger bruger 1165 Nov 20 09:07 fil



13 Filrettigheder

Ændring af rettigheder på filer/kataloger

- Til ændring af rettigheder benyttes kommandoen ***chmod***. Man kan ændre rettigheder på sine *egne* filer og kataloger – enten med ***symboler*** eller ***oktalt***

eksempelvis:

chmod +x

og

chmod 751



13 Filrettigheder

Ændring af rettigheder – med symboler

Man har følgende symboler til rådighed:

u, g, o, a	user, group, other, all
r, w, x	read, write, execute
+	tilføj rettighed
-	fjern rettighed
=	fjerner gamle rettigheder først og sætter dernæst de(n) specificerede rettigheder

```
$ chmod u+x fil1       #(Brugeren tildeles rettigheder til at eksekvere filen)
$ chmod g-w fil2       #(Gruppen får frataget skriverettigheder til filen)
$ chmod a+r fil3       #(Alle tildeles læserettigheder til filen)
$ chmod +r fil1        #(Brugere, gruppen og andre tildeles læserettigheder)
```



13 Filrettigheder

Ændring af rettigheder – med oktale værdier

- De enkelte rettigheder (uanset brugerkategori) har følgende talværdier

<div>r</div>	<div>w</div>	<div>x</div>	<div>r</div>	<div>w</div>	<div>x</div>	<div>r</div>	<div>w</div>	<div>x</div>
4	2	1	4	2	1	4	2	1
Bruger			Gruppe			Andre		

- Rettighederne for *hver enkelt bruger-kategori* angives med en **talværdi fra 0-7**.
Talværdien kan således stå for en enkelt rettighed eller flere rettigheder (**Kombinationer af summen for 4, 2 og 1**).



13 Filrettigheder

Ændring af rettigheder – med oktale værdier

r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

- Execute-rettighed og skriverettighed: 3 (1 + 2)
- Execute-rettighed og læserettighed: 5 (1 + 4)
- Skriverettighed og læserettighed: 6 (2 + 4)
- Alle 3 rettigheder: 7 (4 + 2 + 1)
- Ingen rettigheder: 0

```
$ chmod 777 fil ;ls -l fil
-rwxrwxrwx 1 bruger bruger 1165 Nov 20 09:07 fil
$ chmod 000 fil ;ls -l fil
----- 1 bruger bruger 1165 Nov 20 09:07 fil
$ chmod 751 fil ;ls -l fil
-rwxr-x--x 1 bruger bruger 1165 Nov 20 09:07 fil
```



13 Filrettigheder

Ændring af ejerskab på fil

- Kommandoen ***chown*** bruges til at skifte ejer på en fil. Kommandoen kan dog kun benyttes af ***root***.

Syntax: `chown <ny ejer> <filnavn>`

Eksempel:

```
$ sudo chown bruger3 fil; ls -l fil
[sudo] password for bruger:
-rwxr-x--x 1 bruger3 bruger 1165 Nov 20 09:07 fil
$ sudo chown bruger fil; ls -l fil
-rwxr-x--x 1 bruger bruger 1165 Nov 20 09:07 fil
```



13 Filrettigheder

Ændring af gruppe-tilhørsforhold på fil

- Kommandoen **chgrp** bruges til at skifte gruppe på en fil. Kommandoen kan dog kun benyttes af **root**.

Syntax: `chgrp <ny gruppe> <filnavn>`

Eksempel:

```
$ sudo chgrp gurues fil1  
$ sudo chgrp staff fil2
```



Lav øvelser 13.1 og 13.2



13 Filrettigheder

Rettigheder på fremtidige filer/kataloger

- Styres med kommandoen ***umask***
- Under UNIX/Linux oprettes nye filer normalt med ***standard-rettighederne***: `rw-rw-rw-`
Filer oprettes ***aldrig*** med ***execute-rettigheder***.

Nye kataloger oprettes med

standard-rettighederne: `rwXrwXrwX`

(Husk: eksekverings/søgeret er nødvendig for at kunne gå til et katalog med `cd`).

- Hvis man ***ikke*** ønsker, at nye filer og kataloger oprettes med alle disse rettigheder, kan man bruge kommandoen ***umask*** til at ændre på standard-rettighederne.



13 Filrettigheder

Rettigheder på fremtidige filer/kataloger

Udgangspunktet for de oktale værdier er:

	For filer	For kataloger

	6 6 6 (rw- rw- rw-)	7 7 7 (rwx rwx rwx)

umask		
000	6 6 6 (rw- rw- rw-)	7 7 7 (rwx rwx rwx)
002	6 6 4 (rw- rw- r--)	7 7 5 (rwx rwx r-x)
022	6 4 4 (rw- r-- r--)	7 5 5 (rwx r-x r-x)
033	6 4 4 (rw- r-- r--)	7 4 4 (rwx r-- r--)
077	6 0 0 (rw- --- ---)	7 0 0 (rwx --- ---)

Umask kan også benyttes med symbolske værdier:

umask u=rwx,g=,o=	Svarer til umask 077
umask u=rwx,g=r,o=r	Svarer til umask 033
umask u=rwx,g=rx,o=rx	Svarer til umask 022



13 Filrettigheder

Rettigheder på fremtidige filer/kataloger

- Eksempler på brug af *umask*

```
$ umask 033 ; touch testrettighed1
-rw-r--r-- 1 bruger bruger 0 Nov 21 14:39 testrettighed1
$ umask 777 ; touch testrettighed5
----- 1 bruger bruger 0 Nov 21 14:56 testrettighed5
$ umask 000 ; touch testrettighed6      #(standardrettigheder)
-rw-rw-rw- 1 bruger bruger 0 Nov 21 14:57 testrettighed6
```



Lav øvelse 13.3

