



# Laboration – Behörigheter

OBS det kan förekomma små skillnader vad gäller sökvägar samt befintliga kommandon mellan olika distributioner. T.ex. mellan CentOS och Ubuntu,

**Material:** För att genomföra laborationen behöver man ha tillgång till en dator (vm) med Linux installerat. Windows Subsystem for Linux går att använda också. För vissa övningar kan lokal administratörsbehörighet behövas. I labben utgår det från att man är inloggad med en användare som heter **sysadmin**

**Mål:** I denna laboration kommer du att utföra följande uppgifter:

- Se och förstå behörigheter på filer och kataloger.
- Använd **chmod**-kommandot för att ändra behörigheter.
- Byt ägare med kommandona **chown** och **chgrp**.

## Behörigheter till filer och mappar

I denna uppgift skapar du filer och kataloger, samt visar och ställer in deras behörigheter.

I det här steget skapar du två kataloger och två filer i **/tmp**-katalogen.

Flytta till **/tmp**-katalogen:

```
cd /tmp
```

```
sysadmin@localhost:~$ cd /tmp
```

Använd kommandot **mkdir** för att skapa två kataloger som kallas **priv-dir** och **pub-dir**:

```
mkdir priv-dir pub-dir
```

```
sysadmin@localhost:/tmp$ mkdir priv-dir pub-dir
```

Skapa två filer, en fil kallad **priv-file** i **priv-dir**-katalogen och en annan fil som heter **pub-file** i **pub-dir**-katalogen:

```
touch priv-dir/priv-file
```

```
touch pub-dir/pub-file
```

```
sysadmin@localhost:/tmp$ touch priv-dir/priv-file  
sysadmin@localhost:/tmp$ touch pub-dir/pub-file
```

Filäggande kan bekräftas med detaljerad listning **-l** med **ls**-kommandot. Använd detta kommando för att se innehållet i de nya katalogerna:

```
ls -l priv-dir
```

```
ls -l pub-dir
```

Ditt utdata bör innehålla följande:



```

sysadmin@localhost:/tmp$ ls -l priv-dir
total 0
-rw-rw-r-- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-file
sysadmin@localhost:/tmp$ ls -l pub-dir
total 0
-rw-rw-r-- 1 sysadmin sysadmin 0 Feb 24 18:20 pub-file

```

Observera att för varje listad fil är det första tecknet i raden bindestrecket, `-`. Detta förmedlar informationen att föremålen är *vanliga filer*. Den första tecknet i annonsen anger filtypen. Dessa filtyper listas i tabellen nedan:

Tecken	Betydelse
<b>d</b>	är en mapp
<b>-</b>	är en vanlig fil
<b>l</b>	är en symbolic link
<b>b</b>	är en block device fil
<b>c</b>	är en character device fil
<b>p</b>	är en pipe fil
<b>s</b>	är en socket-fil

De nästa nio tecknen finns i tre grupper om tre tecken.

Den första gruppen av tre tecken är användarägarens behörigheter:

```
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-file
```

De tre följande tecknen är gruppägarens behörigheter:

```
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-file
```

De tre sista tecknen representerar allas behörigheter (kallade "others"):

```
-rw-rw-r-- 1 sysadmin sysadmin 0 Apr 11 21:27 pub-file
```

När man tittar på behörigheter indikerar `r` *läsbehörigheten*, `w` anger skrivbehörigheten och `x` anger *exekveringsbehörigheten*. Ett `-` tecken indikerar att det tillståndet inte har beviljats.

Efter behörigheterna finns ett länkräkningsnummer som anger hur många filer som är länkade till denna fil. Därefter ser du ägaren, ägargruppen, filstorleken, datum/tid då filen senast ändrades och filnamnet.

Filäggande gäller även dolda filer i systemet. Dolda filer, som börjar med punkten `.` tecken listas med `-a` alternativet i `ls`-kommandot. De två första dolda "filerna" som listas är faktiskt den aktuella (`.`) respektive föräldrakatalogerna (`..`). Ägarskapet till alla filer och underkataloger i den aktuella katalogen kan listas med kommandot `ls -la`.

```

ls -la
sysadmin@localhost:/tmp$ ls -la
total 16
drwxrwxrwt 1 root      root      4096 Feb 24 18:19 .
drwxr-xr-x 1 root      root      4096 Feb 24 18:11 ..
drwxrwxr-x 2 sysadmin sysadmin 4096 Feb 24 18:20 priv-dir
drwxrwxr-x 2 sysadmin sysadmin 4096 Feb 24 18:20 pub-dir

```



Om du vill göra en katalog mer privat, använd `chmod`-kommandot för att ta bort behörigheter som others har på katalogen. Att använda `-d`-alternativet med `ls`-kommandot listar katalogposter istället för innehåll.

Använd kommandot `ls -ld` för att visa behörigheter för `priv-dir`-katalogen, och använd sedan `chmod`-kommandot för att ta bort de andras behörigheter för *läsning* och *exekvering*:

```
ls -ld priv-dir/  
chmod o-rx priv-dir/
```

```
sysadmin@localhost:/tmp$ ls -ld priv-dir/  
drwxrwxr-x 2 sysadmin sysadmin 4096 Feb 24 18:20 priv-dir/  
sysadmin@localhost:/tmp$ chmod o-rx priv-dir/
```

Använd slutligen samma `ls`-kommando för att verifiera ändringen i behörigheter. Resultatet visar nu att andra inte har behörighet eller åtkomst till `priv-dir`-katalogen:

```
ls -ld priv-dir/
```

```
sysadmin@localhost:/tmp$ ls -ld priv-dir/  
drwxrwx--- 2 sysadmin sysadmin 23 Feb 24 18:20 priv-dir/  
sysadmin@localhost:/tmp$
```

Du kan använda `chmod`-kommandot för att ändra behörigheter för andra genom att använda ett `o`-tecken följt av antingen ett `+` eller ett `-` tecken för att lägga till eller ta bort behörigheter. `o`-tecknet kan användas för att sätta en exakt behörighet.

Du kan också använda ett `u`-tecken istället för ett `o`-tecken för att ändra användarägarens behörigheter. Använd en `g`-tecken om du vill ändra behörigheter för gruppägaren. Denna metod för att ändra behörigheter kallas *symbolisk* eftersom den använder bokstäver för att representera behörighetsgrupper. `u`-, `g`-, `o`- eller användar-, grupp-, andra-notationen kan vara lättare att använda än *oktalmetoden* som bygger på en binär translation av ett oktaltal (bas 8) tal:

För att ändra allas behörigheter, använd en `a`-karakteristisk istället för `o`, `u` eller `g`.

#### Exempel:

```
chmod a+x file
```

 Ger alla *exekveringstillstånd*

```
chmod g-w file
```

 Tar bort *skrivbehörighet* för gruppägare

```
chmod go+r file
```

 Lägger till *läsbehörighet* för gruppägare och andra

```
chmod o=rwx
```

 Sätter other behörigheter för att *läsa*, *skriva* och *köra*

Om du vill göra en katalog mer offentlig kan du använda `chmod`-kommandot för att lägga till *skrivbehörighet* för andra:

```
ls -ld pub-dir/  
chmod o+w pub-dir/  
ls -ld pub-dir/
```



Din utdata visar nu att andra har *skrivbehörighet* på katalogen (möjligheten att lägga till eller ta bort filer i katalogen):

```
sysadmin@localhost:/tmp$ ls -ld pub-dir/
drwxrwxr-x 2 sysadmin sysadmin 22 Feb 24 18:20 pub-dir/
sysadmin@localhost:/tmp$ chmod o+w pub-dir/
sysadmin@localhost:/tmp$ ls -ld pub-dir/
drwxrwxrwx 2 sysadmin sysadmin 22 Feb 24 18:20 pub-dir/
```

Använd `chmod`-kommandot för att ta bort alla behörigheter från gruppen och andra till `priv-file`:

```
ls -l priv-dir/priv-file
chmod g-rw,o-r priv-dir/priv-file
ls -l priv-dir/priv-file
```

Ditt resultat bör se ut ungefär så här:

```
sysadmin@localhost:/tmp$ ls -l priv-dir/priv-file
-rw-rw-r-- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-dir/priv-file
sysadmin@localhost:/tmp$ chmod g-rw,o-r priv-dir/priv-file
sysadmin@localhost:/tmp$ ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-dir/priv-file
```

Ge alla användare samma läs- och skrivbehörighet för filen `pub-file`:

```
ls -l pub-dir/pub-file
chmod a=rw pub-dir/pub-file
ls -l pub-dir/pub-file
```

Ditt resultat bör se ut ungefär så här:

```
sysadmin@localhost:/tmp$ ls -l pub-dir/pub-file
-rw-rw-r-- 1 sysadmin sysadmin 0 Feb 24 18:20 pub-dir/pub-file
sysadmin@localhost:/tmp$ chmod a=rw pub-dir/pub-file
sysadmin@localhost:/tmp$ ls -l pub-dir/pub-file
-rw-rw-rw- 1 sysadmin sysadmin 0 Feb 24 18:20 pub-dir/pub-file
```

Exekveringsbehörigheter krävs för att användare ska kunna köra körbara filer. Skapa en `test.sh`-fil i `/tmp` som innehåller "date":

```
echo "date" > test.sh
```

```
sysadmin@localhost:/tmp$ echo "date" > test.sh
sysadmin@localhost:/tmp$
```

Om en fil innehåller kommandon kan dessa kommandon köras eller *exekveras* endast om filen har exekveringsbehörighet för användaren. Processen att göra en fil till en körbar fil kräver att man ger exekveringsbehörighet för filen. Utan detta tillstånd kan filen inte behandlas som ett program.



Försök att köra `test.sh`-filen; Det borde misslyckas. Se behörigheterna på filen för att se varför:

```
./test.sh  
ls -l test.sh
```

```
sysadmin@localhost:/tmp$ ./test.sh  
-bash: ./test.sh: Permission denied  
sysadmin@localhost:/tmp$ ls -l test.sh  
-rw-rw-r-- 1 sysadmin sysadmin 5 Feb 24 18:55 test.sh
```

Endast användare/ägaren av en fil (eller root-användaren) får ändra behörigheter på en fil. Ge dig själv, användare/ägaren, exekveringsbehörighet och kör sedan `test.sh`:

```
chmod u+x test.sh  
ls -l test.sh  
./test.sh
```

Utdata visar den tillagda exekveringsbehörigheten för användare/ägaren samt aktuellt datum och tid från att datumkommandot exekveras i din `test.sh`-skriptfil:

```
sysadmin@localhost:/tmp$ chmod u+x test.sh  
sysadmin@localhost:/tmp$ ls -l test.sh  
-rwxrwx-r-- 1 sysadmin sysadmin 5 Feb 24 18:55 test.sh  
sysadmin@localhost:/tmp$ ./test.sh  
Sat Feb 24 18:56:19 UTC 2026
```

Hittills har du sett hur man använder `chmod`-kommandot med *symbolisk notation*, där symboler används för att representera vem (`u`, `g`, `o` och `a`), hur (`+`, `-`, eller `=`), och vad som ska ändras (`r`, `w` och `x`). `chmod`-kommandot kan också användas med ett numeriskt värde som representerar behörigheterna för användare/ägaren, gruppägaren och andra i det som kallas *oktal notation*.

För att använda `chmod`-kommandot med oktal notation måste du först förstå oktalsvärdet för behörigheterna:

Read (r)     4

Write (w)    2

Execute (x)   1

Från den senaste listningen av `test.sh` filen visades behörigheterna vara `rwx` för användare/ägaren, `rw` för gruppägaren och `r` för other. För att uttrycka dessa behörigheter i oktal notation beräknas en total för varje ägare.

Som ett resultat skulle användarens behörighetstotal beräknas som  $4 + 2 + 1$ , eller 7, där 4 är för läsningen, 2 för skrivningen och 1 för exekveringsbehörigheten.

Gruppens totala behörighet skulle vara  $4 + 2$  eller 6, där 4 är för läsbehörighet och 2 för skrivbehörighet.

De other ägartillstånd skulle helt enkelt vara 4 för det läsa-behörighet de har.

Om man lägger ihop allt skulle det oktala värdet för de nuvarande behörigheterna vara 764.



`stat`-kommandot visar mer detaljerad information om en fil, inklusive att ange gruppägarskap både via gruppnamn och GID-nummer. Använd `stat`-kommandot för att verifiera det oktala värdet för behörigheterna (åtkomsten) till `test.sh`:

```
stat test.sh
```

```
sysadmin@localhost:/tmp$ stat test.sh
File: test.sh
Size: 5          Blocks: 8          IO Block: 4096   regular file
Device: 20006fh/2097263d    Inode: 3943515   Links: 1
Access: (0764/-rwxrw-r--)  Uid:( 1001/sysadmin) Gid:(1001/sysadmin)
Access: 2024-02-24 18:56:19.424596208 +0000
Modify: 2024-02-24 18:55:26.212306137 +0000
Change: 2024-02-24 18:56:06.152523859 +0000
Birth: -
```

Behörigheterna listas i fältet Access: i statistikens utdata. Om du ville ändra dessa behörigheter med oktala notation för att ge gruppen och andra *exekveringsbehörighet*, skulle du använda följande tre siffror:

- 7 (read, write and execute) for the user owner
- 7 (read, write and execute) for the group owner
- 5 (read and execute) for others

Det nya läget, eller oktaltal, för behörigheterna skulle då vara 775.

Använd oktala notation, ändra behörigheterna för `test.sh`-filen, så att alla kan köra filen:

```
chmod 775 test.sh
```

```
ls -l test.sh
```

```
sysadmin@localhost:/tmp$ chmod 775 test.sh
sysadmin@localhost:/tmp$ ls -l test.sh
-rwxrwxr-x 1 sysadmin sysadmin 5 Feb 24 18:55 test.sh
```

Nu kommer alla användare på detta system att kunna köra `test.sh`-filen.

## Ägarskap

I denna uppgift kommer du att se och ändra äganderätten till filer och kataloger.

Det finns två kommandon som kan påverka ägarskapet till filer: `chown`-kommandot och `chgrp`-kommandot. `chown`-kommandot kan endast utföras av `root`-användaren och det kan ändra både användaren och gruppen som äger en fil.

`chgrp`-kommandot kan användas antingen av användaren som äger en fil eller av `root`-användaren.

Kommandot `chgrp` ändrar bara gruppen som äger en fil.

När en icke-`root`-användare använder `chgrp`-kommandot kan de endast ändra gruppägarskapet till en grupp där de är medlemmar. `Root`-användaren kan använda `chgrp` för att ändra gruppägarskapet för vilken fil som helst till vilken grupp som helst.



Byt till root-användaren och använd root-lösenordet när du blir tillfrågad, så att du kan utföra både `chown`- och `chgrp`-kommandon för att ändra gruppägarskap till vilken grupp som helst:

```
su -
```

```
sysadmin@localhost:~/tmp$ su -  
Password:  
root@localhost:~#
```

Byt tillbaka till `/tmp`-katalogen och lista detaljerna för `public-dir` och sedan dess innehåll:

```
cd /tmp  
ls -ld pub-dir  
ls -l pub-dir/pub-file
```

Notera att utdata visar katalogen och filen som ägs av `sysadmin`-användaren, samt `sysadmin`-gruppen:

```
root@localhost:~# cd /tmp  
root@localhost:~/tmp# ls -ld pub-dir/  
drwxrwxrwx 2 sysadmin sysadmin 4096 Feb 24 18:20 pub-dir/  
root@localhost:~/tmp# ls -l pub-dir/pub-file  
-rw-rw-rw- 1 sysadmin sysadmin 0 Feb 24 18:20 pub-dir/pub-file
```

Använd kommandot `chown` för att ändra användaren och gruppägaren av `pub-dir` till root-användaren och root-gruppen. Se sedan detaljerna i katalogen:

```
chown root:root pub-dir  
ls -ld pub-dir
```

Din utdata ska visa att både användaren och gruppägarna har ändrats:

```
root@localhost:~/tmp# chown root:root pub-dir  
root@localhost:~/tmp# ls -ld pub-dir/  
drwxrwxrwx 2 root root 4096 Feb 24 18:20 pub-dir/
```

Använd kommandot `chown` för att ändra användarägaren av `pub-file` till `bin`:

```
chown bin pub-dir/pub-file  
ls -l pub-dir/pub-file
```

Utdata visar nu att användarägaren har uppdaterats till `bin`:

```
root@localhost:~/tmp# chown bin pub-dir/pub-file  
root@localhost:~/tmp# ls -l pub-dir/pub-file  
-rw-rw-rw- 1 bin sysadmin 0 Feb 24 18:20 pub-dir/pub-file
```

Se detaljerna om `priv-dir` och dess innehåll:

```
ls -ld priv-dir  
ls -l priv-dir/priv-file
```

Resultatet ska visa att `priv-dir` ägs av `sysadmin`-användaren och `sysadmin`-gruppen:



```
root@localhost:/tmp# ls -ld priv-dir
drwxrwx--- 2 sysadmin sysadmin 4096 Feb 24 18:20 priv-dir
root@localhost:/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-dir/priv-file
```

För att ändra gruppägandet för alla filer i en katalogstruktur, använd det rekursiva `-R`-alternativet till `chgrp`-kommandot. Byt gruppägaren av `priv-dir` och `priv-file` till användargruppen rekursivt med `chgrp`-kommandot och visa de uppdaterade filerna:

```
ls -ld priv-dir
ls -l priv-dir/priv-file
chgrp -R users priv-dir
ls -ld priv-dir
ls -l priv-dir/priv-file
```

```
root@localhost:/tmp# ls -ld priv-dir
drwxrwx--- 2 sysadmin sysadmin 4096 Feb 24 18:20 priv-dir
root@localhost:/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-dir/priv-file
root@localhost:/tmp# chgrp -R users priv-dir
root@localhost:/tmp# ls -ld priv-dir
-rw----- 1 sysadmin sysadmin 0 Feb 24 18:20 priv-dir
root@localhost:/tmp# ls -l priv-dir/priv-file
-rw----- 1 sysadmin users 0 Feb 24 18:20 priv-dir/priv-file
```

Din utdata speglar att när ändringar tillämpas rekursivt på en katalog, gäller ändringarna katalogen och allt den innehåller. Detta skulle innebära att varje underkatalog under `priv-dir` och varje fil i `priv-dir` och alla dess underkataloger skulle få denna ändring tillämpad.