



# Laboration – Arkivering och komprimering

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:

- Skapa arkivfiler med tar, både med och utan komprimering
- Komprimera och dekomprimera filer till ett gzip-arkiv
- Komprimera och dekomprimera filer till ett bzip2-arkiv
- Komprimera och dekomprimera filer till ett xz-arkiv
- Använd zip och unzip för att komprimera och dekomprimera arkivfiler

## Arkivering

I den här uppgiften ska vi använda `gzip`, `bzip2`, `tar`, `zip/unzip` och `xz/unxz` för att arkivera och återställa filer. Dessa kommandon är skapade för att antingen slå ihop flera filer till en enda fil eller komprimera stora filer så att de blir mindre. I vissa fall utför kommandona båda funktionerna.

Att arkivera data är viktigt av flera skäl, däribland följande:

- Stora filer kan vara svåra att överföra. Genom att göra dessa filer mindre går det snabbare att flytta dem.
- Att överföra många filer mellan olika system kan bli omständligt när filerna är många. Genom att slå ihop dem till en enda fil blir processen enklare.
- Filer kan snabbt ta upp mycket utrymme, särskilt på mindre flyttbara medier som USB-minnen. Arkivering minskar det problemet.

Ett område som ofta kan förvirra nybörjare i Linux är frågan: varför finns det så många olika arkiveringskommandon? Svaret är att kommandona har olika egenskaper (till exempel kan vissa lösenordsskydda arkivfiler) och använder olika komprimeringstekniker.

Det viktigaste att veta just nu är hur de olika kommandona fungerar. Med tiden lär du dig att välja rätt verktyg för varje situation.

Använd följande `tar`-kommando för att skapa ett arkiv av katalogen `/etc/udev`. Spara backupen i katalogen `~/mybackups`:

```
cd
mkdir mybackups
tar -cvf mybackups/udev.tar /etc/udev
ls mybackups
```

Din utdata ska likna följande:



```
sysadmin@localhost:~$ cd
sysadmin@localhost:~$ mkdir mybackups
sysadmin@localhost:~$ tar -cvf mybackups/udev.tar /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/rules.d/
/etc/udev/rules.d/70-persistent-cd.rules
/etc/udev/rules.d/README
/etc/udev/hwdb.d/
/etc/udev/udev.conf
sysadmin@localhost:~$ ls mybackups
udev.tar
sysadmin@localhost:~$
```

`tar`-kommandot används för att slå ihop flera filer till en enda fil. Som standard komprimerar det inte data.

Flaggan `-c` instruerar `tar`-kommandot att skapa en `tar`-fil. Flaggan `-v` står för "verbose", vilket gör att `tar` visar vad det gör. Flaggan `-f` används för att ange namnet på `tar`-filen.

### Fundera på detta

`tar` står för **T**ape **AR**chive. Kommandot användes ursprungligen för att skapa säkerhetskopior på band, men idag används det oftast för att skapa arkivfiler.

### Viktigt

Det är inte ett krav att använda filändelsen `.tar` på arkivfiler, men det underlättar när man vill identifiera filtypen. Det anses också vara "god stil" när du skickar en arkivfil till någon annan.

Visa innehållet i en `tar`-fil genom att använda tillgängliga flaggor (`t` = lista innehåll, `v` = detaljerad information, `f` = filnamn):

```
tar -tvf mybackups/udev.tar
```

Din utdata ska likna följande:

```
sysadmin@localhost:~$ tar -tvf mybackups/udev.tar
drwxr-xr-x root/root    0 2021-02-08 15:07 etc/udev/
drwxr-xr-x root/root    0 2021-02-08 15:08 etc/udev/rules.d/
-rw-r--r-- root/root   306 2021-02-08 15:08 etc/udev/rules.d/70-per
-rw-r--r-- root/root  1157 2021-02-08 15:08 etc/udev/rules.d/README
drwxr-xr-x root/root    0 2021-01-06 21:04 etc/udev/hwdb.d/
-rw-r--r-- root/root   218 2021-02-08 15:08 etc/udev/udev.conf
sysadmin@localhost:~$
```

Lägg märke till att filer har säkerhetskopierats rekursivt med *relativa* sökvägsnamn. Det är viktigt eftersom filerna vid extrahering hamnar i din nuvarande katalog och inte skriver över befintliga filer.

För att skapa en `tar`-fil som är komprimerad använder du flaggan `-z`. Flaggan `-z` använder `gzip`-verktyget för att utföra komprimeringen.

```
tar -zcvf mybackups/udev.tar.gz /etc/udev
ls -lh mybackups
```



Din utdata ska likna följande

```
sysadmin@localhost:~$ tar -zcvf mybackups/udev.tar.gz /etc/udev
tar: Removing leading `/' from member names
/etc/udev/
/etc/udev/rules.d/
/etc/udev/rules.d/70-persistent-cd.rules
/etc/udev/rules.d/README
/etc/udev/hwdb.d/
/etc/udev/udev.conf
sysadmin@localhost:~$ ls -lh mybackups/
total 16K
-rw-rw-r-- 1 sysadmin sysadmin 10K Feb  9 17:34 udev.tar
-rw-rw-r-- 1 sysadmin sysadmin 1.2K Feb  9 17:45 udev.tar.gz
sysadmin@localhost:~$
```

Lägg märke till skillnaden i storlek; den första säkerhetskopian (10K byte) är större än den andra säkerhetskopian (1,2K byte).

Extrahera innehållet från ett arkiv. Data återställs som standard till den aktuella katalogen.

```
cd mybackups
ls
tar -xvf udev.tar.gz
ls
ls etc
ls etc/udev
ls etc/udev/rules.d
```

```
sysadmin@localhost:~$ cd mybackups
sysadmin@localhost:~/mybackups$ ls
udev.tar  udev.tar.gz
sysadmin@localhost:~/mybackups$ tar -xvf udev.tar.gz
etc/udev/
etc/udev/rules.d/
etc/udev/rules.d/70-persistent-cd.rules
etc/udev/rules.d/README
etc/udev/hwdb.d/
etc/udev/udev.conf
sysadmin@localhost:~/mybackups$ ls
etc  udev.tar  udev.tar.gz
sysadmin@localhost:~/mybackups$ ls etc
udev
sysadmin@localhost:~/mybackups$ ls etc/udev
hwdb.d  rules.d  udev.conf
sysadmin@localhost:~/mybackups$ ls etc/udev/rules.d
70-persistent-cd.rules  README
```



Om du ville att filerna skulle "hamna tillbaka" på sin ursprungliga plats, kan du först köra `cd` till `/`-katalogen och sedan köra `tar`-kommandot. Men i det här exemplet måste du vara inloggad som administratör eftersom du bara kan skapa filer i `/etc`-katalogen om du är administratör.

För att lägga till en fil i ett befintligt arkiv använder du alternativet `-r` till `tar`-kommandot. Kör följande kommandon för att utföra denna åtgärd och kontrollera att den nya filen finns i tar-arkivet:

```
tar -rvf udev.tar /etc/hosts
tar -tvf udev.tar
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ tar -rvf udev.tar /etc/hosts
tar: Removing leading `/' from member names
/etc/hosts
sysadmin@localhost:~/mybackups$ tar -tvf udev.tar
drwxr-xr-x root/root      0 2021-02-08 15:07 etc/udev/
drwxr-xr-x root/root      0 2021-02-08 15:08 etc/udev/rules.d/
-rw-r--r-- root/root    306 2021-02-08 15:08 etc/udev/rules.d/70-p
-rw-r--r-- root/root   1157 2021-02-08 15:08 etc/udev/rules.d/READ
drwxr-xr-x root/root      0 2021-01-06 21:04 etc/udev/hwdb.d/
-rw-r--r-- root/root    218 2021-02-08 15:08 etc/udev/udev.conf
-rw-r--r-- root/root    172 2024-02-09 17:33 etc/hosts
sysadmin@localhost:~/mybackups$
```

I följande exempel kommer du att använda `gzip` och `gunzip` för att komprimera och packa upp en fil. Kör följande kommandon för att komprimera en kopia av filen `words`:

```
cp /usr/share/dict/words .
ls -l words
gzip words
ls -l words.gz
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ cp /usr/share/dict/words .
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
sysadmin@localhost:~/mybackups$ gzip words
sysadmin@localhost:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 sysadmin sysadmin 259983 Feb  9 17:59 words.gz
sysadmin@localhost:~/mybackups$
```

Observera att den zippade `words`-filen (259983 bytes i exemplet ovan) är mycket mindre än originalfilen (971578 bytes i exemplet ovan).

### Mycket viktigt

När du använder `gzip` ersätts originalfilen med den zippade filen. I exemplet ovan ersattes filen `words` med `words.gz`.

När du packar upp filen kommer den zippade filen att ersättas med originalfilen.



Kör följande kommandon för att packa upp `words.gz`-filen:

```
ls -l words.gz
gunzip words.gz
ls -l words
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ ls -l words.gz
-rw-r--r-- 1 sysadmin sysadmin 259983 Feb  9 17:59 words.gz
sysadmin@localhost:~/mybackups$ gunzip words.gz
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
sysadmin@localhost:~/mybackups$
```

Linux har ett stort antal komprimeringsverktyg utöver `gzip/gunzip`. Alla har sina för- och nackdelar (snabbare komprimering, bättre komprimeringsgrad, mer flexibilitet, mer portabilitet, snabbare dekomprimering, och så vidare).

Kommandona `gzip/gunzip` är väldigt populära i Linux, men du bör känna till att `bzip2/bunzip2` också är vanliga på vissa Linux-distributioner. Det är bra att de flesta funktioner (hur du kör kommandona) och alternativ är samma som för `gzip/gunzip`.

Användningen av `bzip2` och `bunzip2` för att komprimera och packa upp en fil är mycket lik användningen av `gzip` och `gunzip`. Den komprimerade filen får ändelsen `.bz2`. Ändelsen tas bort när filen packas upp. Kör följande kommandon för att komprimera en kopia av filen `words`:

```
ls -l words
bzip2 words
ls -l words.bz2
```

Din utdata bör se ut ungefär så här:

```
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
sysadmin@localhost:~/mybackups$ bzip2 words
sysadmin@localhost:~/mybackups$ ls -l words.bz2
-rw-r--r-- 1 sysadmin sysadmin 345560 Feb  9 17:59 words.bz2
```

Om du jämför den resulterande `.bz2`-filens storlek (345560) med `.gz`-filens storlek (259983) från tidigare steg, märker du att `gzip` gjorde ett bättre jobb med att komprimera just den här filen.

Kör följande kommandon för att packa upp filen `words.bz2`:

```
ls -l words.bz2
bunzip2 words.bz2
ls -l words
```



```
sysadmin@localhost:~/mybackups$ ls -l words.bz2
-rw-r--r-- 1 sysadmin sysadmin 345560 Feb  9 17:59 words.bz2
sysadmin@localhost:~/mybackups$ bunzip2 words.bz2
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
```

Användningen av `xz` och `unxz` för att komprimera och packa upp en fil är också mycket lik användningen av `gzip` och `gunzip`. Den komprimerade filen får ändelsen `.xz`. Ändelsen tas bort när filen packas upp. Kör följande kommandon för att komprimera en kopia av filen `words`:

```
ls -l words
xz words
ls -l words.xz
```

Din utdata bör se ut ungefär så här:

```
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
sysadmin@localhost:~/mybackups$ xz words
sysadmin@localhost:~/mybackups$ ls -l words.xz
-rw-r--r-- 1 sysadmin sysadmin 198756 Feb  9 17:59 words.xz
```

Om du jämför den resulterande `.xz`-filens storlek (198756) med `.gz`-filens storlek (259983) från tidigare steg, märker du att `xz` gjorde ett bättre jobb med att komprimera just den här filen.

Kör följande kommandon för att packa upp `words.xz`-filen:

```
ls -l words.xz
unxz words.xz
ls -l words
```

```
sysadmin@localhost:~/mybackups$ ls -l words.xz
-rw-r--r-- 1 sysadmin sysadmin 198756 Feb  9 17:59 words.xz
sysadmin@localhost:~/mybackups$ unxz words.xz
sysadmin@localhost:~/mybackups$ ls -l words
-rw-r--r-- 1 sysadmin sysadmin 971578 Feb  9 17:59 words
```

Även om `gzip`-, `bzip`- och `xz`-arkiv ofta används i Linux, är `zip`-arkivtypen vanligare i andra operativsystem, som Windows. Faktum är att programmet Utforskaren i Windows har inbyggt stöd för att packa upp `zip`-arkivfiler.

Därför är det oftast att föredra att använda `zip`-arkivtypen om du planerar att dela ett arkiv med Windows-användare (OBS Sen windows 11 22H2 så stödjer även Windows alla vanliga komprimeringsformat som Linux använder). Till skillnad från `gzip` och `bzip2`, när en fil komprimeras med kommandot `zip`, komprimeras en kopia av originalfilen och originalet förblir okomprimerat.

Använd kommandot `zip` för att komprimera filen `words`:

```
zip words.zip words
ls -l words.zip
```



Din utdata bör se ut ungefär så här:

```
sysadmin@localhost:~/mybackups$ zip words.zip words
  adding: words (deflated 73%)
sysadmin@localhost:~/mybackups$ ls -l words.zip
-rw-rw-r-- 1 sysadmin sysadmin 260119 Feb  9 18:23 words.zip
sysadmin@localhost:~/mybackups$
```

I exemplet ovan är det första argumentet (`words.zip`) till `zip`-kommandot namnet på den fil du vill skapa. De återstående argumenten (`words`) är de filer du vill lägga i den komprimerade filen.

### Viktigt

Du måste inte använda filändelsen `.zip` för den komprimerade filen, men det underlättar att se vilken filtyp det är. Det räknas också som "god stil" när du skickar en arkivfil till någon annan.

Komprimera katalogen `/etc/udev` och dess innehåll med `zip`-komprimering:

```
zip -r udev.zip /etc/udev
ls -l udev.zip
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ zip -r udev.zip /etc/udev
  adding: etc/udev/ (stored 0%)
  adding: etc/udev/rules.d/ (stored 0%)
  adding: etc/udev/rules.d/70-persistent-cd.rules (deflated 29%)
  adding: etc/udev/rules.d/README (deflated 50%)
  adding: etc/udev/hwdb.d/ (stored 0%)
  adding: etc/udev/udev.conf (deflated 24%)
sysadmin@localhost:~/mybackups$ ls -l udev.zip
-rw-rw-r-- 1 sysadmin sysadmin 2000 Feb  9 18:25 udev.zip
sysadmin@localhost:~/mybackups$
```

`tar`-kommandot som diskuterades tidigare i denna labb går automatiskt ner i eventuella underkataloger till den katalog som ska arkiveras. Med `bzip2`-, `gzip`- och `zip`-kommandona måste alternativet *recursive* `-r` anges för att recursion ska ske till underkataloger.

För att visa innehållet i ett `zip`-arkiv, använd alternativet `-l` tillsammans med `unzip`-kommandot:

```
unzip -l udev.zip
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ unzip -l udev.zip
Archive:  udev.zip
  Length      Date    Time    Name
-----
     0  2021-02-08  15:07  etc/udev/
     0  2021-02-08  15:08  etc/udev/rules.d/
   306  2021-02-08  15:08  etc/udev/rules.d/70-persistent-cd.ru
  1157  2021-02-08  15:08  etc/udev/rules.d/README
     0  2021-01-06  21:04  etc/udev/hwdb.d/
   218  2021-02-08  15:08  etc/udev/udev.conf
```



För att extrahera **zip**-arkivet använder du kommandot **unzip** utan några alternativ. I detta exempel behöver vi först ta bort de filer som skapades i det tidigare tar-exemplet:

```
rm -r etc
unzip udev.zip
```

Din utdata bör likna följande:

```
sysadmin@localhost:~/mybackups$ rm -r etc
sysadmin@localhost:~/mybackups$ unzip udev.zip
Archive:  udev.zip
  creating: etc/udev/
  creating: etc/udev/rules.d/
 inflating: etc/udev/rules.d/70-persistent-cd.rules
 inflating: etc/udev/rules.d/README
  creating: etc/udev/hwdb.d/
 inflating: etc/udev/udev.conf
filessysadmin@localhost:~/mybackups$
```