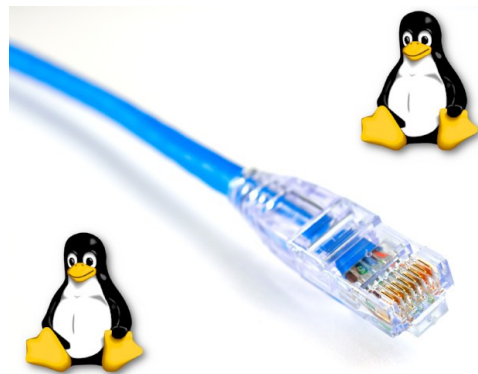


15 Lokalnet, filoverførsel m.m.

Lokalnet, filoverførsel m.m.



SUPERUSERS

15 Lokalnet

Om at logge på andre computere og overførsel af filer:

- **Secure Shell – ssh**
- **Private key - ssh-keygen**
- **Secure Copy - scp**
- **Remote Sync - rsync**
- **Midnight Commander - mc**



15 Lokalnet

Secure Shell – ssh

Secure Shell giver dig mulighed for få krypteret adgang til en anden computer på såvel lokalnet, som på internettet. (client-server).

Vi har prøvet at logge på en lokal computer med en lokal ip-adresse, dette kan også være en ekstern ip-adresse eller domæne eks:

```
$ ssh bruger1@192.168.2.101 #lokal ip-adresse
$ ssh bruger@23.56.184.47   #ekstern ip-adresse
$ ssh bruger@dr.dk          #eksternt domæne
```



15 Client-server

Private key - ssh-keygen

For at øge sikkerheden, kan det være nødvendigt, at man har en *ssh-nøgle* liggende på din klient:

private-key

På hostserveren har man så en “lås”, der ved hjælp af ens public-key, (som har sendt host-udbyderen) giver adgang til serveren.

Dette sikrer at lige netop den, der har nøglen til rådighed sammen med det rette password, kan logge på.



15 Client-server

Private key - ssh-keygen

Sådan opretter du dit eget nøgle-par:

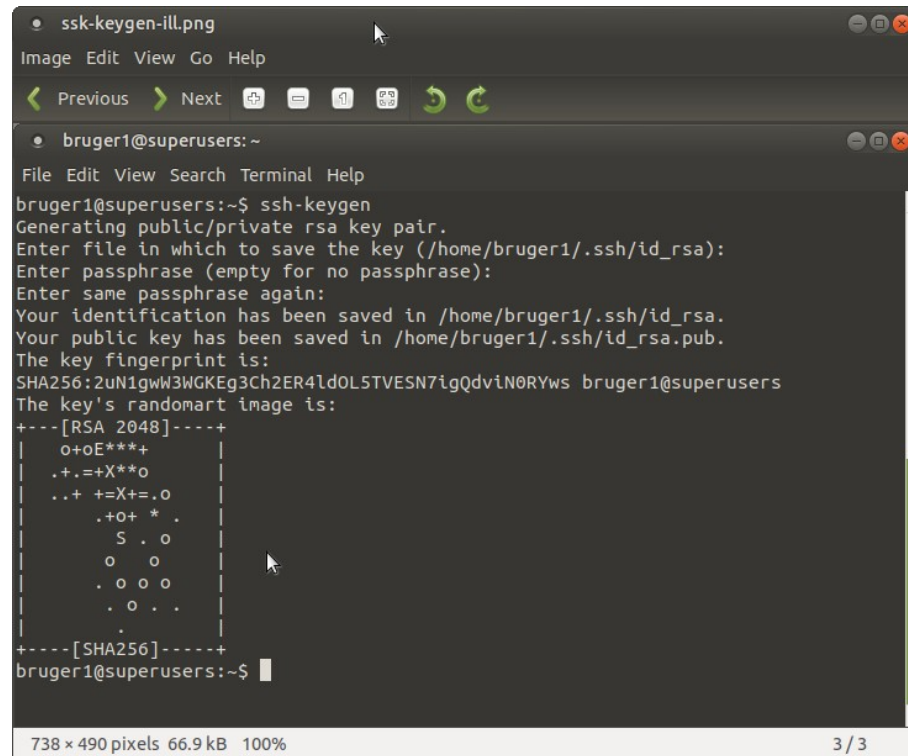
(Som standard bliver lagt i /home/brugenavn/.ssh kataloget)

```
$ ssh-keygen #Danner en rsa-key og lægger den i .ssh
$ ssh-keygen -t rsa -b 2048 -f .ssh/claus-ssh-key
# Beskrivelse af flag:
-b = antal bits
-t = typen (dsa | ecdsa | ed25519 | rsa)
-f = filnavnet
```



15 Client-server

Private key - ssh-keygen



```
ssk-keygen-ill.png
Image Edit View Go Help
< Previous > Next
bruger1@superusers:~
File Edit View Search Terminal Help
bruger1@superusers:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bruger1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bruger1/.ssh/id_rsa.
Your public key has been saved in /home/bruger1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:2uN1gWw3WGKEg3Ch2ER4ldOL5TVESN7igQdviN0RYws bruger1@superusers
The key's randomart image is:
+---[RSA 2048]---+
|  o+oE***+      |
|  .+.=+X**o     |
|  ..+ +=X+=.o   |
|    .+o+ * .    |
|      S . o     |
|      o  o      |
|    . o o o     |
|    . o . .     |
|      .         |
+-----[SHA256]-----+
bruger1@superusers:~$
```



Lav øvelser Modul 15.1



15 Filoverførsel

Secure Copy - scp:

Syntaks: **scp** [flag] [fra] [til]

Kopierer filen "foobar.txt" fra remote host til local host :



```
$ scp bruger@remotehost.dk:remote/directory/foobar.txt /some/local/directory  
# Download
```

Kopierer filen "foobar.txt" fra local host til remote host:



```
$ scp ~/foobar.txt dit_brugernavn@remotehost.dk:/some/remote/directory  
# Upload
```

Kataloger med recursiv kopiering : `scp -r`



15 Filoverførsel

Remote Sync - rsync:



rsync kan arbejde med hele mapper af filer og sørge for, at det kun er nye filer der overføres.

Derved spares en del ressourcer i forhold til at bruge **scp**.

rsync benyttes bl.a. i forbindelse med backup procedurer.

15 Filoverførsel

Remote Sync - rsync:

Syntaks: `rsync [flag] [fra] [til]`

Pull: `rsync [OPTION...] [USER@]HOST:SRC... [DEST]`

Push: `rsync [OPTION...] SRC... [USER@]HOST:DEST`



Pull:

```
$ rsync -avz -e ssh bruger@remotehost.dk:/et/eksternt/katalog ~/lokalt
# Tager eksternt "katalog", pakker det, sender til lokalt "katalog"
pakker det ud og synkroniserer
```

Push:

```
$ rsync -avz -e ssh ~/lokalt/katalog bruger@remotehost.dk:/et/eksternt
# Tager lokalt "katalog", pakker det, sender til eksternt "katalog"
pakker det ud og synkroniserer
```



15 Filoverførsel

Remote Sync - rsync:

Et eksempel på backup (pull):



```
$ rsync --archive --verbose --compress --progress --recursive \  
--times --perms --links --delete \  
-e ssh bruger1@192.168.7.101:/home/bruger1/testRsync ~/mitBackupkatalog
```

og med "enkelt flag":

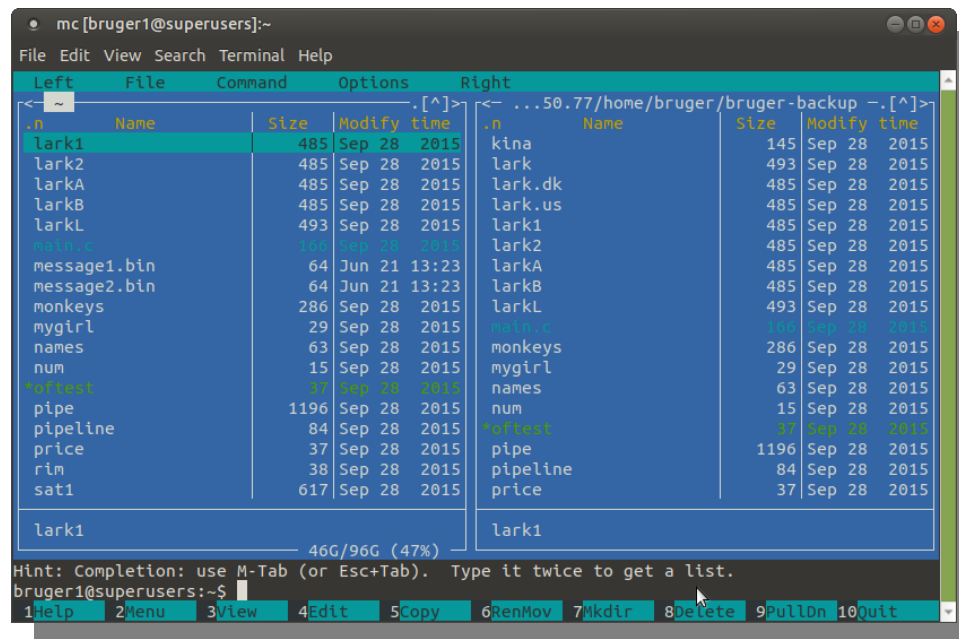
```
$ rsync -avzhrtple ssh bruger1@192.168.7.101:/home/bruger1/testRsync \  
~/mitBackupkatalog
```



15 Midnight commander

Midnight Commander - mc

`$ mc` # starter et semigrafisk brugerinterface
til brug ved filoverførsler m.m.



Lav øvelser Modul 15.2



15 Shellscripts

Hvad er et shellscript?

Et Shellscript er egentlig bare en samling af kommandoer samlet i en fil.

Et shellscript kan indeholde alt fra en enkelt kommando eller to, til flere hundrede kodelinjer.

Shellscript bruges normalt til batch-jobs, cronjobs samt jobs der afvikles i forbindelse med system-boot og bruger login, men man kan også lave sine egne små programmer.

I det følgende er der et par eksempler shellscripts



15 Shellscripts

Shellscript med Pipe Viewer (pv): [<=>]

```
# Eksempel på shellscript der afvikles på 50 sek:
```

```
$ vi 50sek.sh
```

```
#!/bin/bash
#job der hvert sekund gentager ordet test i 50 sekunder
end=$((SECONDS+50))
while [ $SECONDS -lt $end ];
do
    echo "test" ;
    sleep 1;
done
```

```
$ chmod +x 50sek.sh
```

```
$ ./50sek.sh | pv
```

```
$ ./50sek.sh | pv > /dev/null
```

```
35 B 0:00:06 [4.99 B/s] [ <=> ]
```



15 Shellscripts

Shellscript med 'dialog' (En grafisk process-bar)

```
# Eksempel på et shellscript:
```

```
$ vi dialog.sh
```

```
#!/bin/bash
for i in $(seq 0 10 100) ;
do sleep 1;
  echo $i | dialog --gauge "Vent venligst" 10 70 0;
done;
clear
```

```
$ chmod +x dialog.sh
```

```
$ ./dialog.sh
```



Lav øvelse Modul 15.3

