

Varmuuskopiointi: image

- sammuta virtuaalikone ensin - elävästä koneesta otettu kopio ei ole koherentti - ja:

```
cp kone1.img kone1.img.bak
```

- huom. tiedostojen omistaja muuttuu kun virtuaalikone käynnistetään, palautus esim.

```
rm -f kone1.img; cp kone1.img.bak kone1.img
```

```
cp -f kone1.img.bak kone1.img
```

- ei säännöllinen varmuuskopiointitapa, vain poikkeustilanteissa (esim. ennen versiopäivitystä tai jotain "tämä saattaa rikkoa kaiken" -säätöä)

Ubuntun päivitys

- `sudo apt-get update; sudo apt-get dist-upgrade`
 - säännöllisesti, ja aina kun tietoturva-aukkoja tiedossa
 - automatisointi: `/etc/apt/apt.config.d/50unattended-upgrades`,
`dpkg-reconfigure unattended-upgrades`
- Käyttäjärjestelmäversion päivitys: `sudo do-release-upgrade`
 - muistia pitää olla riittävästi, samoin levytilaa
 - jos `do-release-upgrade` -komentoa ei löydy:
`sudo apt-get install update-manager-core`
 - `/etc/update-manager/release-upgrades`: "Prompt=" Its, normal
 - Päivityksen jälkeen poista vanhat kernelit: `dpkg -l|grep linux-image` ja
`apt-get purge linux-image-...` ja `apt-get autoremove`

Tiedostojen aikaleimat

- mtime (modification time)
 - ls -lt [--full-time]
 - touch -m [-t [CC]YYMMDDhhmm[.ss]]
 - find ... -mtime ...
- ctime (status change time, creation time)
 - ls -lc
 - find ... -ctime ...
- atime (access time)
 - ls -lu
 - touch -a ...
 - find ... -atime ...

for-loop, parameter expansion

```
for i in 1 3 5 7 ; do touch koe$i.txt ; done
```

```
for x in a b c ; do echo $x > ${x}koe.txt ; done
```

```
for f in *.txt ; do cp $f $f.bak ; done
```

```
for f in *.txt ; do cp $f ${f%.txt}.bak ; done
```

```
for f in *.txt ; do cp $f ${f/txt/bak} ; done # bash
```

```
for f in a* ; do mv $f b${f#a} ; done
```

```
for n in {0..7} ; do mkdir d$n ; done # bash
```

```
for d in d{0..7} ; do mkdir $d ; done # bash
```

```
touch koe{1..7..2}.txt # bash
```

redirection, read

```
while read suku etu login demo ; do  
    echo $login $etu $suku; id $login  
done < kurssilaiset.txt
```

```
while read suku etu login demo ; do # ei toimi!  
    echo ${login}1; ssh ${login}1 'dpkg -l | grep acpid'  
done < kurssilaiset.txt
```

```
while read <&3 suku etu login demo ; do  
    echo ${login}1; ssh ${login}1 'dpkg -l | grep acpid'  
done 3< kurssilaiset.txt
```

command expansion, eval

- Komennon tulos merkkijonoon:

```
year=$(date +%Y); month=$(date +%m)
```

```
day=`date +%d` # vanha tapa, ei suositeltava
```

```
files=$(ls -l $(grep -l kala *.txt))
```

- Merkkijono komennoksi:

```
eval $(date '+year=%Y month=%m day=%d')
```

- `x=ls; $x` # toimii

```
x='y=z'; $x # ei toimi
```

```
x='y=z'; eval $x; echo $y # toimii
```

Levy täyttyy: miksi

df

du [-s] dir... [| sort -n]

find dir -type f -mtime -1 -size +10000 -user tt

ls -l | sort -k5n | tail

ls -l | grep /var

tail /var/log/syslog.log ...

Levy täyttyy, mitä tehdä

```
rm [-rf] ...
```

```
find dir -mtime -1 -size +10000 -exec gzip {} \;
```

```
... -exec rm {} \;
```

```
find dir -mtime -1 -size +10000 -user tt -print0 |  
xargs -0 gzip
```

Prosessilla auki olevaa tiedostoa ei pidä poistaa tappamatta prosessia ensin (lsof)!

Virtuaalisen kovalevyn lisäys

- Luodaan levyimage:
`qemu-img create kone2b.img 2G`
- Lisätään se virtuaalikoneeseen:
`virsh attach-disk kone $PWD/kone2b.img vdb --persistent`
- Jos acpid toimii ja kyllin uusi kernel (Ubuntu 14.04 ainakin) ja levyohjaimena on virtio, uusi levy ilmestyy lennosta (hotplug), muuten virtuaalikone pitää bootata; tarkista: `dmesg`, `ls -l /dev/vdb`
- Poisto: `virsh detach-disk kone1 vdb [--live|--config|...]`

Uuden levyn käyttöönotto

- Levyn voi (ei ole pakko) partitioida:

`fdisk /dev/vdb`

`parted /dev/vdb`

- Tiedostojärjestelmän luonti, esim:

`mkfs -t ext4 /dev/vdb1`

ellei partitioitu:

`mkfs -t ext4 /dev/dvb`

mkfs optioita

- -t tiedostojärjestelmätyyppi, esim. ext4, xfs, jfs
vaatii yleensä tyyppikohtaisen lisäpaketin,
esim xfsprogs, jfstools
- tyyppikohtaiset man-sivut: man mkfs.ext4 jne
- -m reserved-percentage: kuinka monta prosenttia
varataan superuserille (oletus 5%); säädettävissä
jälkeenpäin komennolla tune2fs (ext*:lle)

Levyn käyttöönotto 2

- Valitse mount point; jos olemassaoleva, data siirrettävä ja niin, ettei sitä siirron aikaan käytetä (jos mahdollista; ellei, esim. /usr, bootti mahdollisimman pian). Esim. /home:

ensin käyttäjät ulos!

`mv /home /oldhome`

`mkdir /home`

`mount /dev/vdb1 /home`

`mv /oldhome/* /home`

/etc/fstab

Jotta uusi levy tulisi käyttöön automaattisesti bootin jälkeen, se pitää lisätä /etc/fstabiin:

```
# laite      polku tyyppi optiot  dump_freq fsck_pass
/dev/vdb1 /home ext4 defaults 0          2
```

Optiot kuten mount-komennossa -o ...

dump_freq nykyisin aina 0

fsck_pass tarkistusjärjestys bootissa, 0=ei tarkisteta
(yleensä root (/) 1, loput 2)

mount optioita 1

- a [-O] # kaikki fstabissa määritellyt (paitsi...)
- r | --readonly
- w | --rw
- U uuid
- t types
- B | --bind
- R | --rbind

mount options 2

-o ... (no-)

atime, noatime, diratime, relatime, strictatime

async, sync

auto

noexec, nosuid, nodev

group, owner, users

remount

ro, rw

loop device

Mountataan tiedosto kuin se olisi levy:

```
mount /tmp/disk.img /mnt/tmp -t ext4 -o loop=/dev/loop3
```

Käytetään tiedostoa kuin se olisi levy:

```
losetup -f /dev/loop0 file.img
```

Partitiot näkyviin:

```
kpartx -a /dev/loop0
```

Mountataan em. tiedoston 1. partitio:

```
mount /dev/mapper/loop0p1 /mnt/tmp
```

Vapautetaan:

```
umount /mnt/tmp; losetup -d /dev/loop0
```


Virtuaalilevyn suurentaminen 1

Edellyttää yleensä aina boottia. Vaarallinen!

Vaiheet:

- (1) Suurennetaan levyimage
- (2) Muutetaan partitointia
- (3) Suurennetaan tiedostojärjestelmä(t) ja swap

Virtuaalilevyn suurentaminen 2

- `virsh shutdown kone # jatkuu alustakoneessa`
- `mv kone.img kone.img.bak; cp kone.img.bak kone.img`
(ensin mv jotta saadaan kirjoitusoikeudet kohdalleen)
- `qemu-img resize kone.img +2GB`
- `virsh start kone # jatkuu virtuaalikoneessa`
- `sudo fdisk -c -u /dev/vda1 # tai parted`
 - poistetaan kaikki partitiot ja luodaan uusiksi, root-partition alun pitää säilyä täsmälleen ennallaan (swap voi siirtyä, muut jos siirtää datankin), samoin tyyppien ja boot-lipun; voi myös lisätä uusia partitioita

Virtuaalikoneen suurentaminen 3

```
sudo shutdown -r now
```

```
sudo resize2fs /dev/vda1; df
```

```
grep swap /etc/fstab
```

```
# cut'n'paste UUID= ... edeltä tai:
```

```
eval $(awk '/UUID.*swap/{print $1}' /etc/fstab)
```

```
sudo mkswap -u $UUID /dev/vda5
```

```
sudo swapon -a; cat /proc/swaps
```

```
shutdown -r now # varmuuden vuoksi
```

```
rm -f kone.img.bak # alustakoneessa (jos kaikki ok)
```

Virtuaalilevyn suurentaminen 4

Jos on root-oikeudet (sudo) alustakoneessa, toinen tapa:

- `virsh shutdown kone`
- `losetup -f # tulos (/dev/loop2 tms) talteen, käytetään alla`
- `losetup -f /dev/loop2 kone.img`
- `kpartx -a /dev/loop2 # partitiot näkyviin`
- `eval $(blkid /dev/mapper/loop2p5 | awk '{print $2}') # UUID`
- `kpartx -d /dev/loop2 # partitiot pois koska ne muuttuvat alla`
- `fdisk /dev/loop2 # partitiot uusiksi, root-partition alku ennallaan`

Virtuaalilevyn suurentaminen 5

- `kpartx -a /dev/loop2 # uudet partitiot näkyviin`
- `e2fsck -f /dev/mapper/loop2p1 # varmuuden vuoksi`
- `resize2fs /dev/mapper/loop2p1`
- `mkswap -u $UUID /dev/mapper/loop2p5`
(swap UUID joka talletettiin edellä)
- `kpartx -d /dev/loop2`
- `losetup -d /dev/loop2`
- `virsh start kone`