Tiedostojen ominaisuuksista

- ls -komennon tärkeimmät optiot:

  -l "long", näyttää omistajan, oikeudet ym

  --full-time aikaleima mikrosekunnilleen

  -a näyttää myös .-alkuiset tiedostot (vrt -A)

  -t tulostus aikajärjestyksessä (vrt -c, -u)

  -r käänteinen tulostusjärjestys

  -R tulosta myös alihakemistot rekursiivisesti

  -d näytä hakemisto (eikä sen sisältöä)

  -i tulosta inodet

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

```
-rw-r--r--. 1 tt users 0 Mar 12 13:50 hauki
```

Normaali tiedosto (-)

Omistajalla luku- ja kirjoitusoikeudet (rw-)

Ryhmällä vain lukuoikeus (r--), samoin muilla (r--)

SELinux käytössä (.); voi olla myös + jos acl käytössä

Yksi inode

Omistaja tt, ryhmä users, koko 0 tavua

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

  ```
  drwxr-xr-x 2 tt users 19 Mar 12 13:50 kala
  ```

  Hakemisto, omistajalla kaikki oikeudet, ryhmällä ja muilla luku- ja suoritusoikeus. Hakemiston suoritusoikeus = oikeus lukea sen alla olevia tiedostoja, lukuoikeus = oikeus listata hakemiston sisältö (tiedostonimet)

Tiedostojen ominaisuuksista

- **ls -ld /tmp** tulostaa:

```
drwxrwxrwt 7 root root 140 Mar 12 13:17 /tmp
```

Hakemisto, kaikilla kaikki oikeudet mutta *sticky-bit* päällä: vain tiedoston omistaja (tai root) voi poistaa sen tai muuttaa sitä. Pieni t = sticky bit + execute bit, iso T = pelkkä sticky bit. Normaaleilla tiedostoilla sticky bitillä ei ole yleistä merkitystä.

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

```
lrwxrwxrwx 1 tt users 4 Mar 12 13:51 fish -> kala
```

Symbolinen linkki: viittaukset kohdistuvat nimettyyn tiedostoon, oikeudetkin tulevat sieltä, itse linkin oikeuksilla ei vaikutusta. Vain jotkin harvat komennot osaavat käsitellä itse linkkiä. Luonti: ln -s

Tiedostojen ominaisuuksista

```
ls -l
   -rw-rw-r-- 2 tt tt 5 Mar 12 15:36 bird
   -rw-rw-r-- 2 tt tt 5 Mar 12 15:36 lintu
ls -li
   22544411 -rw-rw-r-- 2 tt tt 5 Mar 12 15:36 bird
   22544411 -rw-rw-r-- 2 tt tt 5 Mar 12 15:36 lintu
```

”hard link”: sama tiedosto, kaksi hakemistoentryä. Luonti: ln (ei -s:ää)

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

```
-rwsr-xr-x   1 root root       32096 Feb  8  2017 su
```

Normaali tiedosto, omistajalla kaikki oikeudet, ryhmällä ja muilla suoritus- ja lukuoikeus, *suid-bitti* päällä: ajettaessa suoritetaan tiedoston omistajan oikeuksilla (tässä root). Pieni s = suid + execute bitit päällä, iso S = pelkkä suid-bitti päällä. Ei yleistä merkitystä hakemistoille.

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

```
---x--s--x 1 root nobody 306360 Mar 1 2017 ssh-agent
```

Normaali tiedosto, kaikilla suoritusoikeus (eikä muuta), *sgid-bitti* päällä: suoritetaan tiedoston *ryhmän* oikeuksilla (tässä kuten suorittaja olisi nobody-ryhmässä). Hakemistolla sgid-bitti aiheuttaa sen oikeuksien periytymisen sen alle luotuihin tiedostoihin ja hakemistoihin.

Tiedostojen ominaisuuksista

- **ls -l** tulostaa:

```
brw-rw---- 1 root disk      8,   0 Mar  6 14:02 sda
crw-rw-rw- 1 root tty       5,   0 Mar  7 22:00 tty
prw------- 1 root root     0 Mar  6 14:02 fifo
srw------- 1 root root     0 Mar  6 14:02 control
```

Laitetiedostoja, putkia, soketteja… muitakin erikoistapauksia voi olla.
Laitetiedoston koon edellä pilkulla erotettuna laitetyypin (ajurin) tunniste.

Tiedostojen aikaleimat

- mtime (modification time)

  ls -lt [--full-time]

  touch -m [-t [CC]YYMMDDhhmm[.ss]]

  find … -mtime ...

Tiedostojen aikaleimat

atime (access time)

- ls -lu
- touch -a …
- find … -atime …

Tiedostojen aikaleimat

- ctime (status change time, creation time)

  ls -lc

  find … -ctime …

Levy täyttyy: miksi

df

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

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

Levy täyttyy: miksi

ls -l | sort -k5n | tail

lsof -s | grep /var

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

Levy täyttyy, mitä tehdä

rm [-rf] ...

find . -mtime -1 -type f -size +10000 -exec gzip {} \;

tai ... -exec rm {} \;

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

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

Virtuaalisen kovalevyn lisäys

- Luodaan levyimage:

  qemu-img create [-f raw] $HOME/kone2b.img 2G

- Lisätään se virtuaalikoneeseen:

  virsh attach-disk kone $HOME/kone2b.img vdb --persistent

  - Joskus tarpeen --driver qemu --subdriver qcow2
  - Ilman --persistent -optiota katoaa bootissa

Virtuaalisen kovalevyn lisäys 2

- Jos acpid toimii ja kyllin uusi kernel (Ubuntu 14.04 ja uudemmat ainakin) ja levyohjaimena on virtio, uusi levy ilmestyy lennosta (hotplug), muuten virtuaalikone pitää bootata; tarkista: dmesg, ls -l /dev/vdb
- Poisto:

  virsh detach-disk kone vdb [--persistent]

  jos lisätty ilman --persistent, bootti poistaa myös

Uuden levyn käyttöönotto

- Levyn voi (ei ole pakko, yleensä kannattaa) partitioida:

  fdisk /dev/vdb # *tai*

  parted /dev/vdb

Uuden levyn käyttöönotto 2

- Tiedostojärjestelmän luonti, esim:

  mkfs -t ext4 -i 8096 /dev/vdb1

  tai ellei partitioitu:

  mkfs -t ext4 /dev/vdb

mkfs optioita

- -t tiedostojärjestelmätyyppi, esim. ext2, ext4, xfs, jfs, zfs, brtfs, msdos, ntfs, iso9660 (cd-rom)

  muut kuin ext* vaativat yleensä tyyppikohtaisen lisäpaketin, esim. xfsprogs, jfstools, zfsutils
- muut optiot -t:n jälkeen
- tyyppikohtaiset man-sivut: man mkfs.ext4 jne
- mke2fs = mkfs.ext2 = mkfs.ext3 = mkfs.ext4

mke2fs optioita

- -m reserved-percentage: kuinka monta prosenttia varataan superuserille (oletus 5%)
- -i bytes-per-inode: montako tavua per inode, rajaa montako tiedostoa voidaan luoda (oletus vaihtelee, usein 16384, pienennä jos paljon pieniä tiedostoja)
- komennolla tune2fs voidaan tutkia (-l) ja muuttaakin joitakin tiedostojärjestelmän ominaisuuksia; vrt myös df [-hiT...]

Uuden levyn käyttöönotto 3

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

Uuden levyn käyttöönotto 4

- Esim. uudesta levystä /home:
  - *ensin käyttäjät ulos!*

    mv /home /oldhome

    mkdir /home

    mount /dev/vdb1 /home

    mv /oldhome/* /home && rmdir /oldhome

/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
```

(Optio "noauto" estää automaattisen mountin bootissa.)

/etc/fstab 2

Optiot kuten mount-komennossa -o ...

dump_freq nykyisin aina 0

fsck_pass tarkistusjärjestys bootissa, 0=ei tarkisteta

- yleensä root (/) 1, loput 2, cd:t yms 0

mount optioita 1

-a [-O] # kaikki fstabissa määritellyt (paitsi...)

-t types # rajaa myös -a:ta

-r | --readonly

-w | --rw | --read-write

-U uuid

mount optioita 2

-B | --bind # olemassaoleva alihakemisto

-R | --rbind #… monen tiedostojärjestelmän yli

esim.

mount -B /home/tt/testikala /tmp/tt

mount optioita 3

-o ... (no-)

atime, noatime, diratime, relatime, strictatime, lazytime

async, sync

auto, noauto

mount optioita 4

noexec, nosuid, nodev

group, owner, users

remount # halutaan vain vaihtaa optioita

ro, rw # vrt. -r, -w

virt-install - oikeudet

- virt-install osaa luoda tiedoston levyimageksi jos sitä ei ole, mutta se vaihtaa sen omistajaa ja oletuksena jättää lukuoikeudet vain itselleen, joten sitä eivät muut voi kopioidakaan – käyttökelpoinen vain jos olet root. Mutta jos image on valmiiksi tehty, omistaja vaihtuu mutta suojaukset eivät. Imagen luonti etukäteen:

  qemu-img -f qcow2 kone.img 3G

levyformaatit

- KVM-qemu tuntee useita levyformaatteja. Parhaiten tuettu ja yleensä helpoimmin toimiva on *qcow2;* toinen yleinen on *raw*.

  (qcow2 = qemu copy-on-write version 2,

  qemu = Quick EMUlator)