

tune2fs

- Tiedostojärjestelmien ominaisuuksia voi joskus olla tarpeen tutkia tai muuttaa. Välineet riippuvat tiedostojärjestelmätyypistä, historiallinen komento on tunefs, nykyisille ext*-tiedostojärjestelmille **tune2fs**.
- Kaikki säädettävissä olevat asetukset ja paljon muutakin saa näkyviin komennolla **tune2fs -l //laitepolku** (hyvä tehdä ennen kuin muuttaa mitään).
- Tavu/inode -suhteen saa laskemalla tune2fs -l:n tulostuksesta
(Blocks per group)/(Inodes per group)*(Block size)
 - `sudo tune2fs -l /dev/tt1-vg/lvusr | awk '/Blocks per group/{bg=$NF} /Inodes per group/{ig=$NF}/Block size/{bs=$NF}END{print bg/ig*bs}'`

tune2fs

- Joskus hyödyllisiä optioita:
 - m *rootille-varattu-prosenttiosuus*
 - L *volume-label* # aseta nimiö
 - U *UUID* # aseta UUID
 - O +large_file # salli yli 2GB tiedostot (kiellä ^large_file)
 - O +huge_file # salli yli 2TB tiedostot (kiellä ^huge_file)
- jfs:lle on vastaavasti tune_jfs (eri optiot), xfs:lle xfs_admin

Tiedostojärjestelmän muutos

- Tiedostojärjestelmätyyppejä (ext4, xfs &c) ei yleensä voi muuttaa muuten kuin luomalla se kokonaan uudestaan. Poikkeuksena kuitenkin ext2, ext3 ja ext4 ovat muutettavissa ristiin.

- Esim. ext2 → ext4 tai ext3 → ext4:

```
umount ...
```

```
tune2fs -O extents,uninit_bg,dir_index,has_journal /dev/...
```

```
e2fsck -pf /dev/... # varmuuden vuoksi
```

Asennus/päivitys täytti levyn

- Jos jokin kriittinen levyosio (etenkin /, /boot, /var, /usr) on täyttynyt ohjelman asennuksen tai päivityksen yhteydessä ja se on saanut koneen sekaisin:
 - Pakettiluettelon tarkistus: `dpkg -l`
 - Levysyöppöjä voi etsiä: `du, find`
 - `apt-get autoremove`, `apt-get clean`, yritys poistaa (tilapäisesti) paketteja `apt-get purge...` (Ubuntu Xenial alkaen myös `apt purge &c`)

Asennus/päivitys täytti levyn

- Levyosioita voi aina myös yrittää laajentaa (lvextend &c jos käytössä LVM, tai fdisk &c ellei)
- Mikä levyosio täyttyi ei aina ole ilmeistä, tarkista lokit ja virheilmoitukset (ja tietysti df ja df -i)
- Levytilan vapautuksen/lisäämisen jälkeen

apt-get -f install

ja jos oltiin päivittämässä, uudestaan

apt-get update ; apt-get dist-upgrade

Asennus/päivitys täytti levyn

- Jos apt -komennot eivät toimi, paketin voi yrittää poistaa suoraan: `dpkg -r paketti` (tämä rikkoo riippuvuuksia, korjattava myöhemmin)
- Jos `dpkg:kään` ei toimi, voi tiedostoja (varovasti!) poistaa suoraan `rm` -komennolla; tämä on kohtuuturvallista jos esimerkiksi vanhoja kerneleitä on kertynyt, etenkin jos `/boot`-hakemistossa näkyy jämiä kerneleistä joita `dpkg -l` ei näe:

```
ls /boot
```

```
dpkg -l | grep linux-image
```

```
dpkg -l | grep linux-headers
```

Asennus/päivitys täytti levyn

- tilanteen selvittyä pitää siivota sotku:
 - apt-get purge tai dpkg -r rikoituille paketeille
 - apt-get -f install
- lopuksi kannattaa yleensä vielä tehdä
 - apt-get update && apt-get upgrade # tai dist-upgrade

Esimerkki: levytilaongelma

- Tilanne: /usr on täynnä, /home:ssa tilaa vaikka kuinka, ei LVM:ää tai ei vapaata levytilaa VG:ssä.
- Ongelma: /home'n pienentäminen on yleensä vaikeaa, ei juuri koskaan onnistu ilman boottia ja voi olla vaikeaa bootatenkin (esim. jos /home on xfs, jota ei voi pienentää) ja /usr:n laajentaminen on helppoa vain jos käytössä on LVM ja volume groupissa on tilaa.

Esimerkki: levytilaongelma

Ratkaisu: siirretään osa /usr:stä, esimerkiksi /usr/src, /home'n alle ja linkitetään se takaisin:

```
mkdir /home/usr
```

```
mv /usr/src /home/usr
```

```
ln -s /home/usr/src /usr
```

Helppoa ja nopeaa, mutta ylläpidon kannalta jatkossa hankalaa ja virhealtista - paras kohdella tilapäisratkaisuna ja siirtää /usr/src takaisin tai muuttaa se omaksi tiedostojärjestelmäkseen jos ja kun levyn lisäys/suurennus tms myöhemmin sen sallii.

Esimerkki: levytilaongelma

Siirrettävän alihakemiston valinnassa syytä olla varovainen, erityisesti pitää varoa käytössä olevia ja bootissa tarpeellisia hakemistoja. Hyviä vaihtoehtoja /usr:n alla ovat /usr/src, /usr/local ja /usr/share/doc, mahdollisesti jopa koko /usr/share; /var:n alta voi bootin puolesta yleensä siirtää melkein mitä vain, mutta se yleensä edellyttää ao. alihakemistoa käyttävien palveluiden sammuttamista (/var/run ei pidä siirtää). Ennen siirtoa pitää tietysti aina katsoa paljonko tilaa siirrettävän hakemiston alla on (tyhjää hakemistoa ei kannata siirtää).

Esimerkki: inodet lopussa

- Kun `df -i` kertoo, että `IFree` on jollain osiolla pieni (kerneliä päivitettäessä saatetaan tarvita 20000 inodea tai enemmänkin)
- Jos myös levytila (`df -h`) on vähissä on parasta suurentaa levyosiota (jos mahdollista) tai poistaa jotain tarpeetonta levyiltä
- Etsi inode-syöpöt: `du --inodes / | sort -n`
- Kernelistä saattaa olla vanhoja versioita, joita ei enää tarvita. Yleensä `apt autoremove` poistaa ne, mutta aina se ei toimi ja joka tapauksessa se yleensä jättää ainakin kaksi kernel-versiota jäljelle. Jos inodet ovat lopussa voi vanhan varakernelin poistaa, auttaa tilapäisesti (ei pysyvästi koska seuraava päivitys tuo kuitenkin uuden).

Esim: inodet lopussa, kernel-siivous

- Ensin apt autoremove
- Mitä vanhoja versioita on asennettuna?
 - `dpkg -l | grep linux-image`
 - `ls /boot # jos näkyy enemmän kuin yllä, rm ...`
- Mikä on käytössä: `uname -r` (tai `uname -a`); jos vanha, boottaa
- Mitä paketteja vanhaan kerneliin kuuluu: `dpkg -l | grep 4.4.0-116`
 - Poista kaikki po. version paketit (image, headers, extrat)
`apt-get purge ...` (yleensä neljä pakettia)
 - Jos apt / apt-get eivät toimi: `dpkg -r ...`
 - Jos dpkg ei toimi, `rm /boot/...` (varovasti!)
 - Lopuksi `apt-get -f install` (ja ehkä `apt-get update`; `apt-get dist-upgrade`)

Esim. inodet lopussa, uusi osio

- Jos /usr:ssä loppuvat inodet, syyllinen on yleensä /usr/src, ja paras pysyvä ratkaisu on tehdä siitä oma levyosionsa, jolla on enemmän inodeja. Jos levytilaa on ja LVM on käytössä, tämä on helppoa:

```
du -sh /usr/src # katso koko - tehdään vähän isompi
```

```
lvcreate -n lvsrc -L ... vg...
```

```
mkfs -t ext4 -i ... /dev/mapper/... # vrt df -i, tuneefs
```

```
echo "... " >>/etc/fstab
```

```
# huom. seuraava on nopea mutta hieman vaarallinen:
```

```
mv /usr/src /usr/src.old; mkdir /usr/src; mount /usr/src
```

```
mv /usr/src.old/* /usr/src; rmdir /usr/src.old
```

Esim. inodet lopussa, uusi osio

- # Turvallisempi tapa tehdä siirto
(lyhyempi aikaikkuna jona /usr/src rikki):
mkdir /usr/src.new
mount /dev/vg.../lvsrc /usr/src.new
cp -a /usr/src/* /usr/src.new
umount /usr/src.new
mv /usr/src /usr/src.old # /usr/src katoaa
mkdir /usr/src; mount /usr/src # ja palaa
rm -rf /usr/src.{new,old}
- Lopuksi kannattaa tehdä df -i /usr/src

Esimerkki: levytilaa kateissa

- Jos levyn/osion/taltion mounttaa hakemistoon, joka ei ole tyhjä, alle jäävät tiedostot jäävät näkymättömiin mutta vievät yhä levytilaa (ja ilmestyvät taas näkyviin umountin jälkeen)
- Esim. tehdään /tmp:stä erillinen taltio (LV):

```
lvcreate -n lvtmp -L500M vg1
```

```
mkfs -t ext2 /dev/vg1/lvtmp
```

```
mount /dev/vg1/lvtmp /tmp # vanha sisältö jää piiloon
```
- Mahdollisesti auki ollut tiedosto elää piilossa kunnes ao. prosessi kuolee

System startup & shutdown

- Useita erilaisia ja eri-ikäisiä menetelmiä:
 - Vanha Unixin init-script setup (SysV): `/etc/init.d`
 - Upstart (Ubuntu, Fedora, ChromeOS): `/etc/init`
 - Systemd (melkein kaikki nykyisin): `/etc/systemd`
- Vanhat käynnistyskriptit usein viritetty toimimaan uudempien kanssa → kaikkia näkee sekaisin
- Käynnistysasetuksia: `/etc/default/...`

Startup & shutdown: sysV

- SysV-init, "klassinen" run-level -pohjainen init-script setup (RHEL -6 ym)
- /etc/init.d/proggis.sh [start|stop|restart...]
 debugatessa sh -x ...
- /etc/rc?.d/{Snn,Knn}proggis
 - ?=runlevel (0=halt, 1=single user... ks. man telinit)
 - S = start, K = kill, nn = järjestysnumero
- Vanha mutta yhä käytössä uudempien rinnalla

Startup & shutdown: upstart

- upstart: Ubuntu 6.10-14.10, Fedora 9-14, ChromeOS:
- start proggis
stop proggis
service proggis [start|stop|restart...]
- /etc/init/proggis.conf
- lähes kadonnut uusista järjestelmistä mutta jotain saattaa vielä käynnistyä /etc/init/'istä

Startup & shutdown: systemd

- systemd: Fedora 15-, Debian 8-, RHEL 7-, Ubuntu 15.04- ...
init + udev + syslog + cron...
- `systemctl [start|stop|status|...] proggis`
- `/etc/systemd/system/proggis.service`
- käynnistää ohjelmia rinnakkain, riippuvuuksia
`/etc/systemd/system/*.target.wants`
- hyvin monipuolinen, paljon ominaisuuksia

Palveluiden tila

- Palveluiden tilaa voi tarkastella komennoilla
 `service [palvelu status|--status-all] # upstart, sysV`
 `systemctl status [palvelu] # systemd`
- Joskus tarpeen tutkia yksittäisiä prosesseja:
 `ps -ef | grep ...`
 `ps -fu userid # tietyn id:n omistamat prosessit`
 `lsof -p PID # prosessin avoimet tiedostot`

busybox

- Etenkin asennusvaiheessa (avattaessa konsoli Alt-F2:lla jne), joskus muutenkin minimalistisissa asennuksissa käytettävä staattinen binääri, joka sisältää riisutut versiot yleisimmistä komennoista
- Bootin jäädessä (initramfs) -promptiin käytössä on juuri busybox
- Monissa komennoissa vähemmän optioita kuin standardiversioissa (uudemmissa busyboxin versioissa enemmän optioita)
- <https://busybox.net/downloads/BusyBox.html>

VM:n valvonta

- `virsh domstatus kone # vrt. virsh list | grep kone`
`case $(virsh domstatus $KONE) in *running*) echo "$KONE
päällä";; esac`
- `virsh event kone event [--all] [--loop] [--timeout sec] [--list]`
 - odottaa kunnes haluttu *event* tapahtuu
 - **virsh event --list** näyttää käytettävissä olevat *eventit*
 - **virsh event kone --all --loop** näyttää kaikki *eventit* niiden tapahtuessa

VM:n valvonta

- Sammutetaan VM, odotetaan että se sammuu:

```
virsh shutdown kone1
```

```
virsh event kone1 lifecycle --timeout 60
```

```
if virsh domstate kone1 | grep -q running ;then
```

```
    echo "kone1 shutdown failed" >&2
```

```
fi
```

VM:n valvonta

Joskus hyödyllisiä myös

- `virsh dominfo kone`
- `virsh domblklist kone # levyt`
- `virsh domiflist kone # verkkokortit (interfaces)`
- `virsh domif-getlink kone interface`
- `virsh domifstat kone interface`
- `virsh domstats kone`
- `virsh help monitor`

ping

- Lähettää ns. "icmp echo" -paketteja ja seuraa vastauksia; helppo tapa tarkistaa onko kone verkossa, yleensä vain

`ping kone # tai IP-osoite`

- Optioista on hyötyä lähinnä skripteissä mm.

`-c count` lähetä vain *count* pakettia

`-q` älä tulosta jokaista pakettia

ping

- Tutkittaessa verkko-ongelmaa koneen sisältä kannattaa aloittaa "ping localhost"illa, sitten pingailla yhä kauemmas verkossa
- Ei aina toimi palomuurien läpi
- Vrt. traceroute, nmap

file

`file [optiot] tiedosto[t]`

- Yrittää tunnistaa tiedoston tyyppin (sen nimestä välittämättä)
- Paljon optioita tulostuksen muotoiluun ym, usein hyödyllisiä:
 - k tulosta kaikki useista mahdollisuuksista
 - L/-h seuraa/älä seuraa symbolisia linkkejä
(oletus riippuu ympäristömuuttujasta POSIXLY_CORRECT)
 - s avaa myös laitetiedostot yms
 - r älä muuta kontrollimerkkejä oktaalimuotoon
- Opetettavissa omilla säännöillä tiedostossa /etc/magic

Recovery mode

- Jos kone on niin solmussa, ettei se boottaa normaalisti tai ainakaan saa verkkoa/ssh:ta ylös, ensimmäinen keino on yrittää recovery mode -boottia; se on myös käyttökelpoinen jos pitää pienentää levypartitiota, jota normaalisti koko ajan käytetään
- Jos grubissa on (kuten oletuksena on) `HIDDEN_TIMEOUT`, boottivalikon saaminen edellyttää bootin keskeyttämistä sokkona oikealla hetkellä (joka voi olla hyvinkin lyhyt); palvelimissa `HIDDEN_TIMEOUT` kannattaa siksi yleensä poistaa käytöstä (ja pidentää viivettä)

Recovery mode

- Graafisen konsolin voi käynnistää ennen boottia:
virt-viewer --wait kone &
- Tekstikonsolin (virsh console kone) saa päälle vasta VM:n käynnistyttyä
- Boottivalikossa voi myös muokata menuentryjä, kernelin parametreja tai bootata vanhalta kerneliltä
- Vrt. CD:ltä (tms) bootattaessa Rescue Mode

Recovery mode valikko

- Ubuntun (esimerkkinä 16.04) normaali recovery boot tarjoaa valikon yleisimmille pelastustoimille:
 - resume: jatkaa normaalia boottia
 - clean: tekee apt-get autoremove'n ym rutiinitoimia levytilan vapauttamiseksi
 - dpkg: yrittää korjata rikkinäisiä paketteja

Recovery mode valikko

- fsck: tiedostojärjestelmien tarkistus- ja korjausyritys
- grub: asentaa grub'in uudestaan (grub-install)
- network: yrittää käynnistää verkon
- root: shell pelastuksen jatkamiseksi käsin
- system-summary: näyttää perustietoja koneesta (ei aina toimi)

Single-user mode

- Single-user tilaan voi joutua automaattisesti erilaisten virhetilanteiden seurauksena tai tarkoituksella virheiden korjaamista varten, erityisesti kun recovery-valikosta on valittu "root".
 - Koneen kunnosta riippuen osa levyistä tai partitioista saattaa olla kateissa, verkkoyhteydet saattavat olla nurin, useimmat palvelut (mukaanlukien ssh) ovat yleensä nurin

Single-user mode

- Levyjen tilan tarkistus: `df`, `vgs`, `ls /dev/?d?*`, `fdisk -l /dev/sda`, `mount...` jne
 - Jos kriittisiä levyjä (`/usr` tms) puuttuu, tarkista onko `/etc/fstab` rikki, jos kryptausta on käytetty, `/etc/crypttab`; LVM:ää käytettäessä kannattaa kokeilla `vgscan`, ehkä `vgchange -a y ...` (jos käytössä softaRAID, `cat /proc/mdstat`, `/etc/mdadm/mdadm.conf`)
 - LVM-komennot muodossa "`lvm ...`", esim. `lvm vgscan`
- Tarkista lokit (ainakin `/var/log/syslog`), `dmesg`

Single-user mode

- Verkon tilan tarkistus: ip link, ip address, ifconfig, ping
 - Jos verkko on nurin, sen voi yrittää käynnistää käsin komennolla `ifup eth0` (laitenimi voi vaihdella), joskus ensin muokaten `/etc/network/interfaces` tai `/etc/netplan/*.xml` -tiedostoa
- Jos muut editorit eivät toimi, **sed** voi pelastaa päivän (tai **cat >file**)

Recovery: LVM, / broken

- Jos recovery boot onnistuu:
 - root shellissä:
 - ls /dev/mapper; jos lv ei näy: lvm vgscan, lvm vgchange -a y ...
 - Mountataan / read-write, muuten sitä ei voi muokata:
mount -o remount, rw /

Recovery: LVM, / broken

- Mountataan /usr ja ehkä /var, muuten työkaluja on vähän:
mount /usr; mount /var
- Jos mount /usr ei onnistu, esim. /etc/fstab on rikki:
 - Yritetään mountata käsin, esim.
e2fsck /dev/kone-vg/lvusr
mount /dev/kone-vg/lvusr /usr
 - Korjataan mitä voidaan ilman /usr:ää; jos nano/vi eivät toimi, sed usein toimii
- Ei kannata mountata enempää kuin tarpeen

Recovery: LVM, / broken

- Jos jokin levyosio on täynnä:
 - Yritetään tyhjentää, recovery boot -valikosta tai käsin
 - Jos VG:ssä tilaa, suurennetaan LV:tä: lvextend ...
Huom. optio -r ei tässä yleensä toimi vaan pitää tehdä erikseen resize2fs tai xfs_growfs tms
 - VG:n laajentaminen (levyn lisäys tai pv:n kasvatus) myös mahdollista, kone pitää yleensä sammuttaa ensin

Recovery: LVM, / broken 2

- Tilanne: kone1 LVM root (/) niin rikki, ettei edes Recovery Boot onnistu. Liitetään levy toiseen virtuaalikoneeseen kone2 ja korjataan siellä.
 - Varmista ettei kone2:ssa ole samannimistä VG:tä
 - Alustakoneessa:

```
virsh destroy kone1  
virsh attach-disk kone2 $HOME/kone1.img vdb  
# --persistent ei yleensä tarpeen (jos levyajuri virtio),  
# --driver/--subdriver voi olla
```

Recovery: LVM, / broken 2

- Kone2:ssa:

```
sudo apt-get install lvm2 # ellei ole jo
```

```
sudo vgscan
```

```
ls /dev/mapper # näkyykö kone1--vg-root jne?
```

```
sudo vgchange -a y kone1-vg # jos ei näkynyt yllä
```

```
# jos ei vieläkään näy: unohtuiko --driver/--subdriver?
```

Recovery: LVM, / broken 2

```
sudo mkdir /mnt/root
```

```
sudo mount /dev/kone1-vg/lvroot /mnt/root
```

```
sudo nano /mnt/root/etc/fstab # tms, riippuu mitä on rikki
```

```
sudo umount /mnt/root
```

```
sudo vgchange -a n kone1-vg
```

```
ls /dev/mapper # kone1... ei pitäisi enää näkyä
```

Recovery: LVM, / broken 2

- Lopuksi alustakoneessa:
virsh detach-disk kone2 vdb
virsh start kone1