12 Linux (RedHat) -järjestelmä ja sen ylläpito

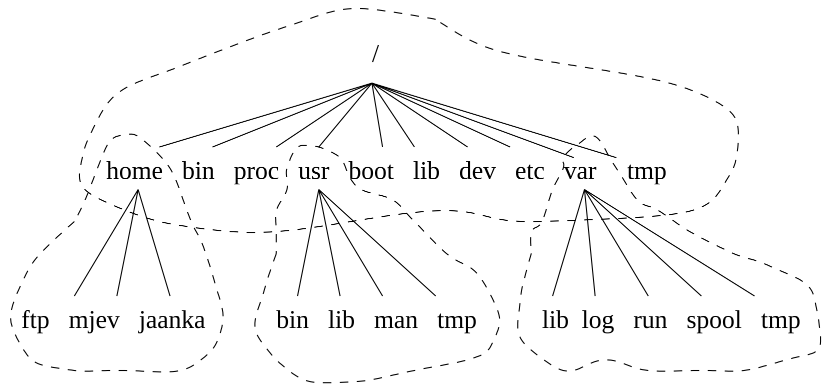
Tässä dokumentissa on tarkoitus käsitellä Linux-järjestelmää ja järjestelmän ylläpitoa. Koska eri Unix-järjestelmät poikkeavat melko paljon toisistaan on mahdotonta kirjoittaa yleispätevää kuvausta Unix-järjestelmän ylläpidosta, mutta monet asiat pätevät kuitenkin yleisemminkin.

Yleisesti järjestelmän ylläpitäjän tehtäviin kuuluu pitää järjestelmä toimivana, huolehtia tiedostojen varmuuskopioinnista (ja mahdollisesta tiedostojen palautuksesta), asentaa uudet sovellusohjelmat, uusien käyttäjätunnusten luonti ja turhien poisto jne.

12.1 Levyn partitioinnista ja hakemistorakenteesta

Linuxin hakemistorakenne pyritään määrittämään ja yhdenmukaistamaan FSSTND:n (filesystem standard) avulla. Kyseinen standardi pyrkii määrittämään kuinka linuxin hakemistorakenne pitäisi järjestää. Tällä pyritään helpottamaan ohjelmien kehitystä ja ylläpitoa.

Linuxin tiedostojärjestelmä on tarkoitettu jaettavaksi pienempiin osiin joista kukin osa sijaitsee omalla levypartitiollaan. Hakemistorakenteen pääosat ovat: juuri `/`, `/usr`, `/var` ja `/home` -hakemistot. Jokaisella hakemistolla on oma tarkoituksensa ja hakemistorakenne on suunniteltu siten, että muutamat hakemistot voivat sijaita sellaisella laitteella johon on vain lukuoikeus. Esimerkiksi verkkolevyllä, jossa useammat linux-järjestelmät voivat jakaa saman hakemiston.



Kuva 8: Linuxin hakemistorakenne (katkoviiva tarkoittaa partitiorajoja)

Jokaisella linux-koneella on oma juurihakemisto, joka normaalisti sijaitsee lokaalilla levyllä. Juurihakemisto voi olla myös *ramdisk* tai se voi sijaita myös verkkolevyllä. Juurihakemistossa sijaitsevat kaikki järjestelmän käynnistyksessä tarvittavat tiedostot, siten että muut hakemistot voidaan *mountata*.

`/usr` hakemisto sisältää suoritettavat ohjelmat, kirjastot, manuaalisivut ja muut normaalisti tarvittavat muuttumattomat tiedostot. `/usr` hakemistossa ei normaalisti ole mitään tietylle koneelle tarkoitettuja tiedostoja, eikä hakemistossa olevia tiedostoja ole tarve muuttaa normaalikäytössä. Tämä mahdollistaa `/usr` hakemiston jakamisen verkossa eri koneiden välillä.

`/var` hakemisto sisältää muuttuvia tiedostoja, kuten tulostusjonot, mailjonot, lokitiedostot jne.

`/home` hakemisto sisältää käyttäjien kotihakemistot. Kotihakemistojen erottaminen omaksi hakemistoksi helpottaa varmuuskopiointia. Suuri `/home` hakemisto voidaan edelleen jakaa useampaan eri partitioon.

12.1.1 Juurihakemiston / osat

Normaalisti juurihakemiston `/` sisältävä levypartitio kannattaa pitää pienenä koska se sisältää järjestelmän käynnistämiseen tarvittavat tiedostot ja pieni harvoin muuttuva tiedostojärjestelmä ei korruptoidu helposti.

Juurihakemisto `(/)` itse ei yleensä sisällä mitään tiedostoja, paitsi ehkä *boot image*n.

/bin
Komennot joita tarvitaan järjestelmän käynnistyksessä ja jotka on tarkoitettu tavallisille käyttäjille.

/sbin
Samoin kuin `/bin` mutta komentoja ei ole tarkoitettu tavallisille käyttäjille. Komentojen käyttö voidaan sallia myös tavallisille käyttäjille.

/etc
Järjestelmäkohtaiset konfiguraatiotiedostot

/root
root käyttäjän kotihakemisto.

/lib
Juurihakemiston ohjelmien tarvitsemat jaetut kirjastotiedostot.

/lib/modules
Ytimen ladattavat modulit.

/dev
Hakemisto `/dev` sisältää laitetiedostot kaikille eri laitteille. Laitetiedostot luodaan järjestelmän asennuksen yhteydessä. Myöhemmin niitä voidaan luoda `/dev/MAKEDEV` komentojonolla.

/tmp
Tilapäistiedostot.

/boot
Järjestelmän käynnistäjän (bootstrap loader), esimerkiksi LILO, tarvitsemat tiedostot. Ytimen kuva (tavallisesti `vmlinuz`) pidetään yleensä `/boot` hakemistossa.

/mnt
Tilapäiset "mount"hakemistot.

/proc
`/proc` tiedostojärjestelmä sisältää kuvitteellisen hakemistojärjestelmän joka ei sijaitsee levyllä vaan ydin luo sen keskusmuistiin. Sen tarkoituksena on antaa tietoa järjestelmän toiminnasta.

Lisätietoa *proc(5)*

/usr, /var, /home
`/usr, /var, /home` hakemistot ovat eri tiedostojärjestelmien "mount" hakemistoja.

12.1.2 `/etc` hakemisto

Seuraavassa esitellään muutamia tärkeitä `/etc` hakemistossa olevia tiedostoja ja alihakemistoja.

`/etc/fstab` tiedosto sisältää tiedot järjestelmässä olevista tiedostojärjestelmistä.

`/etc/motd` (message of the day) tiedosto sisältää viestin joka näytettään automaattisesti onnistuneen järjestelmään kirjoittautumisen jälkeen. Tiedoston avulla järjestelmän ylläpitäjä voi kertoa esimerkiksi aioituista huoltokatkoksista.

`/etc/nsswitch.conf` (Name Service Switch config file) Tiedostossa määritellään millä tavalla eri palveluiden hakeminen toteutetaan. Katso esimerkki alempana.

`/etc/printcap` tiedostossa on tiedot järjestelmään asennetuista kirjoittimista.

`/etc/profile` Komentotulkkien *bash* ja *ksh* alustamiseen tarkoitetut määritykset. Tiedosto suoritetaan vain sisäänkirjottautuessa login shellille.

`/etc/skel/` hakemistossa on tiedostot jotka on tarkotus kopioida käyttäjän kotihakemistoon kun uusi käyttäjä luodaan

`/etc/termcap` tiedostossa listataan erityyppisten terminaalien ominaisuudet.

Esimerkki `/etc/nsswitch.conf` tiedostosta:

```
passwd:     files nisplus nis
shadow:     files nisplus nis
group:      files nisplus nis
hosts:      files nisplus nis dns
services:   nisplus [NOTFOUND=return] files
networks:   nisplus [NOTFOUND=return] files
protocols:  nisplus [NOTFOUND=return] files
rpc:        nisplus [NOTFOUND=return] files
```

siis esimerkiksi salasanaa haetaan ensin tiedostosta, sitten käytetään NIS+ palvelua ja viimeisenä NIS palvelua. Kenttä `[NOTFOUND=return]` tarkoittaa että palvelun etsiminen lopetetaan jos edellisessä kentässä määritelty tapa ei palauta mitään. On kuitenkin huomattava, että jos edellisessä haussa tulee virhetilanne (esim. NIS palvelin ei vastaa) jatketaan seuraavan kentän osoittamalla tavalla.

12.2 Init prosessi

Linux-järjestelmän käynnistyessä ladataan aluksi ydin (kernel) muistiin. Kun ydin on käynnistetty ja kaikki tarvittavat laiteajurit ladattu jne. käynnistyy ensimmäinen käyttäjätason prosessi eli *init* prosessi. Koska init on ensimmäinen prosessi on sen PID (process ID) aina yksi.

Init prosessin käynnistyessä se hoitaa aluksi joitain ylläpidollisia tehtäviä kuten tiedostojärjestelmän tarkistuksen, `/tmp` hakemiston siivouksen, muutamien palvelujen käynnistyksen ja init myös käynnistää jonkinlaisen *login*-prosessin (useimmiten *getty*) jokaiselle terminaalille eli fyysiselle päätteelle ja virtuaalikonsolille joilta käyttäjät voivat kirjoittautua sisään järjestelmään.

Järjestelmän toimiessa normaalisti käynnistää init aina getty prosessin uudestaan terminaalille, jolta käyttäjä on kirjoittautunut ulos. Init prosessin tehtävä on myös huolehtia niistä prosesseista joiden vanhempi prosessi on kuollut ennenaikojaan.

Init prosessin käynnistyessä se lukee `/etc/inittab` konfiguraatiotiedoston. Järjestelmän olessa käynnissä init prosessi lukee `/etc/inittab` tiedoston uudestaan jos se vastaanottaa HUP signaalin (esim. `kill -HUP 1`). Itseasiassa sama toimenpide (`kill -HUP` *`daemon_pid`*) toimii muillekin *daemon*i prosesseille kun niiden konfigurointitiedostoa muutetaan.

12.2.1 `/etc/inittab` tiedosto

`/etc/inittab` tiedosto koostuu riveistä joissa on neljä kaksoispisteellä erotettua kenttää: (Tyhjät rivit ja '`#`'-merkillä alkavat rivit jätetään huomiotta.)

```
id:runlevels:action:prosess
```

id
Yksilöi rivin tiedostossa. *getty* riveille se kertoo terminaalin jolla getty prosessi odottaa. Muille riveille *id*:n pitää olla yksilöllinen (huomioi pituusrajoitus: 1-4 merkkiä).

runlevels
Lista käyttötasoista, joilla annettu tapahtuma suoritetaan.

action
Kertoo, miten haluttu tapahtuma (tai prosessi) suoritetaan.

process
Määrittää prosessin joka suoritetaan.

Esimerkiksi rivi

```
1:12345:respawn:/sbin/mingetty tty1
```

tarkoittaa, että käyttötasoilla 1, 2, 3, 4 ja 5 käynnistetään `/sbin/mingetty`-prosessi terminaalille `tty1` aina uudestaan kun edellinen on kuollut.

12.2.2 Käyttötasot (runlevel)

Käyttötaso on *init* prosessin ja koko järjestelmän taso, joka määrittää mitä palveluita on käytössä.

Käyttötasot

0	halt
1	Yhden käyttäjän tila (järjestelmän ylläpito)
2-5	Normaalit käyttäjätilat
6	Uudelleen käynnistys (reboot)

`/etc/inittab` tiedostossa käyttötasot määritellään seuraavasti:

```
l3:3:wait:/etc/rc.d/rc 3
```

Ensimmäinen kenttä kertoo varsinaisen käyttötason. Toinen kenttä kertoo että ko. riviä käytetään tasolla kolme. Kolmas kenttä tarkoittaa, että neljännen kentän komento suoritetaan välittömästi, kun kyseinen taso käynnistyy ja *init* odottaa komennon suoritusta. `/etc/rc.d/rc` komento suorittaa tarvittavat komennot joilla käynnistetään tai lopetetaan palvelut käyttötasolla kolme.

Init prosessin käynnistyessä se hakee `/etc/inittab` tiedostosta riviä joka määrittää järjestelmän oletuskäyttötason. Kyseinen rivi on muotoa:

```
id:runlevel:initdefault:
```

Järjestelmän toimiessa, käyttötasoa voidaan muuttaa komennolla *telinit*.

Lisätietoa: *init(8), telinit(8) ,inittab(5)*

12.2.3 `/etc/rc.d/` hakemisto.

`/etc/rc.d/` hakemiston ja sen alihakemistojen komentojonoilla määrätään mitä palveluita käynnistetään tai lopetetaan järjestelmän käynnistyessä tai käyttötasoa muutettaessa.

`/etc/rc.d/` hakemistossa on jokaista käyttötasoa vastaava alihakemisto `/etc/rc.d/rcX.d` (X on 0,1,2,3,4,5 tai 6). Näissä alihakemistoissa määrätään mitä palveluja kullekin käyttötasolle käynnistetään. Kyseiset alihakemistot sisältävät symbolisia linkkejä varsinaisiin palvelujen käynnistyskomentojonoihin jotka sijaitsevat tavallisesti `/etc/rc.d/init.d/` hakemistossa. Se käynnistetäänkö vai lopetetaanko palvelu kyseiselle käyttötasolle riipuu symbolisen linkin nimestä seuraavasti: Mikäli nimi alkaa isolla s (S) kirjaimella palvelu käynnistetään kyseiselle käyttötasolle. Mikäli nimi taasen alkaa isolla k (K) kirjaimella palvelu lopetetaan kyseiselle käyttötasolle. Kirjaimen S tai K jälkeen tuleva numero linkin nimessä kertoo palvelun käynnistys- tai lopetusprioriteetin.

Palvelujen käynnistystä, lopetusta ja uusien palvelujen lisäystä voidaan hallita *chkconfig* komentorivityökalulla.

Lisätietoa: *chkconfig(8)*, *ntsysv(8)*, *tksysv(8)*, *runlevel(8)*

12.3 Käyttäjät

Linux järjestelmä (kernel) käsittelee käyttäjiä yksikäsitteisen kokonaisluvun, UID:n (user ID) avulla. Käyttäjän todellinen käyttäjätunnus ja muu tieto käyttäjästä sijaitsee tiedostossa
`/etc/passwd`.

Käyttäjän luomiseksi linux järjestelmään on lisättävä tarvittava tieto `/etc/passwd` tiedostoon ja luotava käyttäjälle kotihakemisto. Yleensä linuxin mukana tulee ohjelma jolla käyttäjä voidaan lisätä järjestelmään, esimerkiksi *useradd* tai *adduser* (lisätietoa *adduser(8)*).

12.3.1 `/etc/passwd` tiedosto

Tieto järjestelmässä olevista käyttäjistä sijaitsee `/etc/passwd` tekstitiedostossa. Tiedostossa on jokaista käyttäjää kohden yksi rivi jossa on seitsemän kaksoispisteellä erotettua kenttää:

- Käyttäjätunnus (username)

- Kryptattu salasana, jos ei käytetä `/etc/shadow` tiedostoa.
- Numeerinen käyttäjätunnus (UID)
- Numeerinen ryhmätunnus (GID)
- Käyttäjän todellinen nimi ja muuta tietoa.
- Kotihakemisto.
- Sisäänkirjoittautuessa suoritettava ohjelma (login shell).

Käyttäjän luominen ilman edellämainittuja apuohjelmia tapahtuu seuraavalla tavalla:

- Lisätään tarvittava tieto `/etc/passwd` tiedostoon. Editoitaessa `/etc/passwd` tiedostoa on syytä olla erityisen varovainen. Editointi on syytä tehdä tätä varten tarkoitetulla editorilla *vipw* joka hoitaa `/etc/passwd` tiedoston lukitsemisen ja suorittaa tarvittavat toimenpiteet kun editointi on tehty. Lisätietoa *vipw(8)*. Salasanakenttään kannattaa tässä vaiheessa laittaa '*' mikä estää tunnuksen käytön. Myös UID:n (user ID) valintaan kannattaa kiinnittää huomiota mikäli järjestelmä on kytketty verkkoon. Muutamat verkkopalvelut (esim. NFS) perustavat käyttäjän tunnistamisen pelkästään UID numeroon, joten samalla käyttäjällä on syytä olla sama UID verkon eri koneissa.
- Luodaan käyttäjälle kotihakemisto *mkdir* komennolla.
- Kopioidaan tiedostot `/etc/skel` hakemistosta käyttäjän kotihakemistoon.
- Laitetaan kotihakemiston omistusoikeudet ja suoritusoikeudet kuntoon *chown* ja *chmod* komennoilla. Esimerkiksi:

  ```
  chown -R user /home/user
  chmod -R 700  /home/user
  ```

- Asetetaan käyttäjälle salasana *passwd* komennolla.

Oletuksena kryptattu käyttäjän salasana sijaitsee `/etc/passwd` tiedoston toisessa kentässä. Usean käyttäjän järjestelmässä tämä ei ole kuitenkaan suositeltavaa koska `/etc/passwd` tiedostoon on oltava lukuoikeus maailmalla. Tällöin kuka tahansa saa tietoonsa kryptatut salasanat. Parempi tapa on ottaa käyttöön `/etc/shadow` tiedosto johon kryptatut salasanat sijoitetaan. `/etc/shadow` tiedostossa jokaiselle käyttäjälle on oma rivinsä joka koostuu yhdeksästä kaksoispisteellä erotetusta kentästä. `/etc/shadow` tiedosto voidaan ottaa käyttöön *pwconv* ohjelmalla. Lisätietoa *shadow(5)* ja *pwconv(8)*

12.3.2 Käyttäjien levytila (quota)

Järjestelmässä jossa on paljon käyttäjiä ja rajallinen määrä levytilaa on syytä määritellä käyttäjän käytössä oleva levytila (quota). Levytilan käyttömääritys voi olla joko käyttäjäkohtainen tai ryhmäkohtainen ja se on tehtävä jokaiselle partitiolle erikseen.

Quota määritellään tiedostossa `quota.user` ja `quota.qroup` joiden pitää olle sen tiedostojärjestelmän juurihakemistossa jolle ne määrittävät quotan.

Quotan voi ottaa käyttöön seuraavalla tavalla: (olettaen että järjestelmän ytimessä (kernel) on quota tuki mukana ja järjestelmästä löytyvät tarvittavat ohjelmat)

Tiedostoon `/etc/fstab` quota määritellään seuraavasti:

```
/dev/hda2  /home ext2 defaults,usrquota,grpquota  1  1
```

Luodaan tiedostot `quota.user` ja `quota.qroup` esim komennolla:

```
touch quota.user quota.group
```

ja koska edellä luotuihin tiedostoihin pitää olla maailmalla lukuoikeus on hyvä antaa myös komento

```
chmod 644 quota.*
```

Quota voidaan laittaa tämän jälkeen päälle komennolla

```
quotaon -a
```

Tämän jälkeen käyttäjien quotaa voidaan editoida komennolla *edquota* jolle annetaan parametrina käyttäjätunnus.

Esimerkiksi komento

```
edquota mjev
```

avaa editorin *vi* johon tulee seuraavanlainen teksti:

```
Quotas for user mjev:
/dev/hda1: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

Käyttäjälle voidaan nyt määrittää quota muuttamalla *soft* ja *hard* parametrien arvoja (arvo 0 tarkoittaa rajoittamatonta quotaa).

HT. Miten laitetaan tämän jälkeen sama quota muillekin käyttäjille?

Parametri *soft* määrittää suurimman levytilan joka on käyttäjän käytössä. Kun myös aikaraja (grace period) on määritelty *soft* toimii rajana jonka ylityksen jälkeen käyttäjää varoitetaan quotan ylityksestä.

NFS mountatuille hakemistoille *nfsd* kutsuu *rcp.rquotad* demonia palvelinkoneella joka ilmoittaa quotan.

Lisätietoja *quota(8), quotactl (2), edquota (8), quotacheck (8), quotaon (8), repquota (8)*

12.4 Komentojen ajastus: cron ja at

12.4.1 Cron

Järjestelmän ylläpidossa on monia sellaisia asioita jotka tulee suorittaa säännöllisesti esimerkiksi varmuuskopiointi, `/tmp` hakemiston tyhjennys jne. Säännöllisesti suoritettavia tehtäviä tehtäviä varten linux järjestelmästä löytyy *cron*-demoni. Cron demoni etsii `/var/spool/cron` hakemistosta *crontab* tiedostoja jotka on nimetty käyttäjätunnusten mukaan. Löydetyt crontab tiedostot ladataan muistiin. Cron demoni etsii myös `/etc/crontab` konfiguraatiotiedostoa.

Minuutin välein cron demoni tutkii pitääkö jokin crontab tiedostoissa määritetty komentojono suorittaa. Cron tutkii myös minuutin välein onko `/var/spool/cron` hakemistossa olevat crontab tiedostot muuttuneet ja lataa muuttuneet crontab tiedostot uudestaan muistiin samoin tutkitaan myös `/etc/crontab` tiedosto. Suoritetuista komentojonoista lähetetään sähköposti viesti kyseisen crontab tiedoston omistajalle tai crontab tiedostossa määriteltyyn MAILTO muuttujan osoittamaan osoitteeseen.

`crontab` tiedostolla kerrotaan *cron* demonille milloin haluttu komento suoritetaan. `crontab` tiedosto koostuu ympäristömuuttuja määrittelyistä ja itse crontab rivistä.

Esimerkki `crontab` tiedostosta:

```
# Käytetään /bin/sh komentotulkkia komentojen suoritukseen
SHELL=/bin/sh
# Viesti 'mjev' osoitteeseen
MAILTO=mjev
# Suoritetaan 'cron.job' jokapaivä 5 minuuttia yli puolenyön
5 0 * * *       $HOME/cron.job
```

Aikamäärittely koostuu seuraavista kentistä: (vasemmalta oikealle)

Kenttä	**Sallitut arvot**
minuutti	0–59
tunti	0–23
kuukaudenpäivä	1–31
kuukausi	1–12
viikonpäivä	0–6 (0 = sunnuntai)

Kuhunkin kenttään voidaan laittaa myös (*), joka tarkoittaa kaikkia kyseisen kentän arvoja.

Kun `crontab` tiedosto on kirjoitettu se voidaan laittaa suorituslistalle käyttäen *crontab* komentoa.

Esimerkki:

```
crontab cron.job
```

12.4.2 at

Tietty komento voidaan ajastaa suoritettavaksi myöhemmin *at*:ta käyttäen. *At* on tarkoittu komentojen kertaluontoiseen suorittamiseen. (*cron*:a käytetään komentojen toistuvaan suoritukseen.) Haluttu kommento voidaan kirjoittaa joko suoraan komentoriville tai se voidaan lukea tiedostosta.

at komento on seuraavaa muotoa:

```
at [-V] [-q queue] [-f file] [-mldbv] TIME
```

Halutun komennon suoritusaika voidaan antaa monella tapaa. Tarkemmat kuvaukset löytyy tiedostosta `/usr/doc/atXXXXX/timespec`.

Esimerkiksi komento:

```
at -f my_at.job teatime
```

Tiedostossa `my_at.job` olevat komennot suoritetaan teeaikaan (kello 16:00)

Lisätietoja *cron(8), crontab(1), crontab(5), at(1)*

12.5 Verkkopalvelut (demonit)

12.5.1 inetd ja `/etc/inetd.conf` tiedosto

Lähes jokaisesta Unix järjestelmästä löytyy ns. "super-server" joka luo palveluiden tarvitsemat portit ja kuuntelee niitä kaikkia yhtäaikaa käyttäen *select()* systeemikutsua. Kun johonkin porttiin tulee kutsu "super-server" huomaa tämän ja käynnistää palvelijan joka on tarkoittu huolehtimaan kyseisen portin palvelusta.

Yleisesti käytetty "super-server" on *inetd* Internet demoni. Inetd käynnistetään järjestelmän käynnistyksen yhteydessä ja se lukee `/etc/inetd.conf` tiedoston, jossa kerrotaan mitä palveluita tarjotaan. Ulkoisten pelvelijoiden lisäksi inetd tarjoaa muutaman triviaalin sisäisen palvelun (siis hoitaa palvelun tarjoamisen itse). Tällaisia palveluita ovat esimerkiksi "chargen"(merkin generointi) ja "daytime" (kellonaika).

`/etc/inetd.conf` tiedosto muodostuu riveistä, jokainen palvelu on omalla rivillään. Merkillä `#` alkavat rivit ovat kommentteja. Rivi muodostuu seuraavista kentistä:

```
service name (service name)
socket-tyyppi (socket type)
protokolla (protocol)
wait/nowait[.max]
user[.group]
palvelin ohjelma (server program)
palvelin ohjelman argumentit (server program arguments)
```

service name Määrittää palvelunnimen. Palvelunnimi yhdistetään portin numeroon etsimällä palvelu `/etc/services` tiedostosta.

socket type Määrittää socketin tyypin. Tyyppi on joko *stream*, *dgram raw*, *rdm* tai *seqpacket*.

protocol Nimeää siirtoprotokollan jota palvelu käyttää. Protokollan nimi on löydyttävä
`/etc/protocols` tiedostosta.

wait/nowait Arvoa wait käytetään vain *dgram* socketeille, muille tulee määritellä arvoksi nowait.

user Määrittää käyttäjätunnuksen jolla palvelun tarjoava prosessi suoritetaan.

server program Palvelun tarjoavan ohjelman koko polku ja nimi. Sisäiset palvelut merkitään avainsanalla *internal*

server program arguments Palvelin ohjelmalle tarkoitetut argumentit. Alkaen argumentista *argv[0]* eli komennon nimestä

Seuraavassa muutama esimerkkirivi `/etc/inetd.conf` tiedostosta:

```
#  inetd.conf
echo   stream  tcp     nowait  root    internal
echo   dgram   udp     wait    root    internal
discard        stream  tcp     nowait  root    internal
discard        dgram   udp     wait    root    internal
```

```
daytime        stream  tcp     nowait  root    internal
daytime        dgram   udp     wait    root    internal

ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
gopher stream  tcp     nowait  root    /usr/sbin/tcpd  gn
```

Verkkoon liitetyn linux-järjestelmän verkkoturvallisuutta on helppo parantaa kommentoimalla `/etc/inetd.conf` tiedostosta kaikki ne palvelut joita ei tarvita. Käytännössä pelkkänä työasemana käytettävän linux koneen `/etc/inetd.conf` tiedostosta voi kommentoida kaikki rivit (paitsi ehkä *auth* rivi), varsinkin jos käytettävissä on *ssh ja sshd*.

Lisätietoa *inetd(8)*, *protocols(5)*, *services(5)*

12.5.2 tcpd ja "kulun valvonta"

Verkkopalveluiden tarjoaminen aiheuttaa järjestelmälle aina turvallisuusriskejä. Verkon kautta tapahtuvia hyökkäyksiä järjestelmää kohtaan voidaan valvoa erilaisilla sovelluksilla. Voi myöskin olla että, tietty palvelu halutaan tarjota vain tiettyyn osaan verkkoa. *Inetd* ei suoraan tarjoa tähan mahdollisuutta.

Yksi tapa valvoa ja rajoittaa palvelujen käyttöä on *tcpd* eli niinsanottu "tcp-wrapper (*daemon wrapper*)". Kun jonkun TCP palvelun käyttöä halutaan valvoa tai rajoittaa käynnistetään tpcd palvelun sijasta. Tcpd kirjaa pyynnön *syslog* demonille ja tarkistaa onko pyynnön esittäjällä lupa käyttää pyytämäänsä palvelua ja vain jos tämä onnistuu, tcpd käynnistää todellisen palvelinohjelman.

Esimerkiksi *telnet* demonin käyttöä voidaan valvoa määrittämällä `/etc/inetd.conf` tiedostossa rivi:

```
telnet stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

Käyttörajoitukset määritellään tiedostostoissa `/etc/hosts.deny` ja `/etc/hosts.allow`. Pyynnön saapuessa tcpd hakee tiedostoista `/etc/hosts.allow` ja `/etc/hosts.deny` (tässä järjestyksessä) sekä palvelun nimeä että kutsun esittänyttä clientosoitetta. Pyyntö hyväksytään jos se täsmää `/etc/hosts.allow` tiedostossa olevaan määritykseen riippumatta `/etc/hosts.deny` tiedoston sisällöstä. Mikäli pyyntö täsmää `/etc/hosts.deny` olevan määrityksen kanssa palvelua ei tarjota. Mikäli täsmäävää määritystä ei löydy kummastakaan tiedostosta palvelu tarjotaan.

Lisätietoa *tcpd(8)*, *hosts.deny(5)*, *hosts.allow(5)*.

12.5.3 ssh, sshd

Lähiverkossa on yleensä useita koneita jotka hoitavat eri tehtäviä ja lähes jatkuvasti joudutaan ottamaan interaktiivinen yhteys toiseen verkon Unix koneeseen tai suorittamaan tiettyjä komentoja toisessa koneessa ei-interaktiivisesti. Perinteisesti tämä tapahtuu esimerkiksi *telnet* tai *rlogin* ohjelmilla tai muilla ns. *r* komennoilla. Nämä ohjelmat eivät kuitenkaan tarjoa minkäänlaista suojausta muodostaessaan yhteyden koneiden välille. Myöskin käyttäjän autentikoinnissa antama salasana välitetään verkossa koneiden välillä sellaisena kuin se on kirjoitettu. Mikäli verkon liikennettä on mahdollista 'kuunnella' saadaan helpohkosti selville käyttäjien salasanoja.

Edellämainittujen perinteisten *telnet* ja *rlogin* ohjelmien sijasta voidaan käyttää *ssh* ohjelmaa (secure shell). *Ssh* ohjelman käyttö vaatii että koneessa johon yhteys otetaan on toiminnassa *sshd* (ssh demoni).

Sshd toimii siten, että jokaisella koneella on oma RSA-avain, jota käytettään koneen yksilöimiseen. Kun asiakas ottaa yhteyden palvelimeen, palvelin lähettää oman tunnusavaimensa (*host key*) ja palvelimen julkisen salausavaimen takaisin asiakasohjelmalle. Asiakas vertaa saatua tunnusavainta omaan tiedostoonsa tarkistaen, ettei avain ole muuttunut. Tämän jälkeen asiakas generoi 256 bittisen satunnaisluvun. Generoitu satunnaisluku kryptataan sekä 'host key':llä että palvelimen julkisella salausavaimella ja lähetetään sen jälkeen palvelimelle (kryptattuna). Palvelin pystyy purkamaan kryptauksen käyttämällä omaa ei-julkista salausavaintaan sekä 'host key':tä ja saa tuloksena kyseisen satunnaisluvun. Tämän jälkeen molemmat osapuolet alkavat käyttää kyseistä istuntoavainta kaiken istuntoon liittyvän kommunikaation kryptaamiseen.

Tämän jälkeen asiakas ja palvelin suorittavat istunnon autentikoinnin. Aluksi asiakas yrittää tehdä `.rhosts` autentikoinnin. Tätä tapaa ei luitenkaan suositella käytettäväksi siihen riittyvien riskien vuoksi. Toisena asiakas yrittää tehdä autentikoinnin joka perustuu yhdistettyyn `.rhosts` ja RSA autentikointiin. Kolmas tapa perustuu niinsanottuun julkiseen avaimeen. Neljäs tapa on käyttää TIS-autentikointi palvelinta. Mikäli kaikki muut tavat suorittaa autentikointi epäonnistuvat suoritetaan salasana-autentikointi. Tämä on kuitenkin turvallinen tapa koska verkossa liikkuva salasana kryptataan edellä mainitulla keinolla.

Lisätietoa *ssh(8), sshd(8), ssh-keygen(1), ssh-agent(1), ssh-add(1), scp(1), make-ssh-known-hosts(1), rlogin(1), telnet(1)*

12.5.4 ftp, ftpd palvelin

Ftp (file transfer protocol) on tarkoitettu tiedostojen siirtoon verkossa olevien koneiden välillä. Esimerkiksi Linux järjestelmä voidaan asentaa koneelle ftp:tä käyttäen, jolloin asennukseen tarvitaan vain kaksi asennuslevykettä ja kiinteä verkkoyhteys. Asennus voidaan tehdä miltä ftp-palvelimelta vain (palvelimen ei siis tarvitse sijaita omassa lähiverkossa).

Lähes jokaisessa Unix-koneessa on vakiona ftp ohjelma (client) jolla tiedostoja voidaan siirtää. Mikäli halutaan että oman järjestelmän tiedostot ovat saatavissa ftp:llä on järjestelmään asennettava *ftpd* (ftp demoni). *ftpd* palvelu toimii *inetd* ja *tcpd* palveluiden kanssa.

Ftpd autentikoi käyttäjän neljän säännön mukaan:

- Annetun käyttäjänimen pitää löytyä `/etc/passwd` tiedostostosta ja salasana ei saa olla tyhjä. Tässä tapauksessa asiakkaan on annettava salasana ennenkuin tiedosto operaatiot ovat mahdollisia.
- Käyttäjätunnus ei saa löytyä `/etc/ftpusers` tiedostosta.
- Käyttäjälle on oltava määriteltynä standardi komentotulkki jonka *getusershell* palauttaa. Komentotulkki pitää olla listattuna myös tiedostossa `/etc/shells`
- Jos käyttäjätunnus on joko "anonymous"tai "ftp", täytyy `/etc/passwd` tiedostosta löytyä anonyymi käyttäjätunnus ("ftp"). Tällöin käyttäjä voi kirjoittautua sisään ilman salasanaa.

Viimeisessä tapauksessa *ftpd* rajoittaa käyttäjän oikeuksia suorittamalla *chroot* komennon "ftp"käyttäjän kotihakemistossa. Jottei järjestelmän turvallisuutta vaaranneta on "ftp"käyttäjän kotihakemiston rakentamiseen syytä kiinnittää erityistä huomiota.

Ainakin seuraavat seikat kannattaa tarkistaa:

- "ftp"käyttäjän kotihakemisto `~ftp` on syytä olla *root* käyttäjän omistuksessa ja siihen ei saa olla muilla käyttäjillä kirjoitusoikeutta.
- `~ftp/bin` hakemiston on myös oltava *root* käyttäjän omistuksessa ja sille on syytä asettaa moodi `111` On syytä muistaa että ko. hakemistossa tulee olla kaikki komennot joita tarvitaan ftp yhteyden aikana.
- `~ftp/etc` hakemisto on myös oltava *root* käyttäjän omistuksessa ja sinne ei saa olla kirjoitusoikeutta kenelläkään. Jos halutaan että *ls* komento näyttää oikeat käyttäjä ja ryhmätunnukset on hakemistossa oltava

myös `passwd` ja `group` tiedostot. Koska salasanakenttiä ei tarvita on ne syytä poistaa `passwd` tiedosta. Tiedostoille `passwd` ja `group` kannattaa määrittää moodi `444` ja ne on syytä pitää *root* käyttäjän omistuksessa.

- Myös `~ftp/pub` hakemisto kannattaa luoda ja asettaa sille tarvittavat oikeudet. Esim. `733` jos halutaan että tavalliset käyttäjät voivat ladata tiedostoja ftp palvelimelle.

Lisätietoja *ftp(1), ftpd(8), ftpaccess(5), chroot(5)*

12.5.5 httpd palvelin

Varmaankin suosituin linuxin käyttökohde on yhdessä Apache httpd-palvelijan kanssa WWW-palvelimena. Tämän yhdistelmän suosio perustuu siihen että molemmat, sekä Linux että Apache, ovat ilmaisia.

Httpd palvelimen toiminta määritellään konfiguraatiotiedostoissa, jotka sijaitsevat normaalisti palvelimen juurihakemiston (ServerRoot) alihakemistossa `conf`. Konfiguraatiotiedostot ovat

```
conf/httpd.conf
conf/srm.conf
conf/access.conf
conf/mime.types
```

`httpd.conf` tiedostossa määritellään *httpd* palvelimen toiminta.

`srm.conf` tiedostossa määritellään asiakkaille tarjottavien dokumenttien spesifikaatiot.

`access.conf` tiedostossa määritellään mitä dokumentteja asiakkailla on oikeus saada ja mistä nämä dokumentit löytyvät.

`mime.types` tiedostossa voidaan määritellä dokumenttien *mime* tyypit.

Lisätietoa *httpd(8), http://www.apache.org/*

12.5.6 NFS (network filesystem)

NFS on RCP:tä käyttävä palvelu, jonka avulla voidaan tarjota pääsy verkossa olevan järjestelmän tiedostoihin aivan kuin tiedostot sijaisisivat koneen lokaalilla levyllä.

NFS:n tarjoamia etuja ovat muunmuassa:

- Kaikkien käyttäjien käyttämä tieto voidaan pitää keskuspalvelin-koneella ja asiakaskoneet mounttaavat tarvittavan hakemiston järjestelmän käynnistyessä. Esimerkiksi käyttäjien kotihakemistot voivat sijaita yhdellä palvelimella ja verkossa olevat koneet mouttaavat `/home` hakemiston ko. palvelimelta.
- Paljon levytilaa käyttävät ohjelmat voivat sijaita yhdellä palvelimella. Esimerkiksi TEX-ladontaohjelman käyttämät makro- ja fonttitiedostot voivat sijaita yhdellä koneella, jossa niitä on myös helpompi hallita.
- Järjestelmän ylläpitoon käytettävät tiedostot voidaan pitää yhdellä koneella eikä samoja tiedostoja tarvitse kopioida kaikkiin koneisiin.

NFS hakemiston mounttaus tapahtuu samaan tapaan kuin minkä tahansa fyysisen laitteen. *mount* komennon syntaksi on seuraavanlainen:

```
mount -t nfs remote_host:/remote_dir local_dir
```

Tällöin *mount* yrittää kutsua *mountd*:a (mount demoni) *remote_host* koneessa RPC:tä käyttäen. Palvelin tarkistaa onko paikallisella koneella lupa mountata haluttu hakemisto. Mikäli hakemiston mounttaus onnistuu, palautetaan sen tiedostokuvaaja (file handle).

Kun tiedostoja käytetään NFS:n yli ydin (kernel) lähettää RPC kutsun *nfsd*:lle (NFS daemon) palvelin koneessa. Kutsu ottaa tiedosto kuvaajan, halutun tiedoston nimen ja käyttäjän UID:n sekä GID:n. Näiden avulla selvitetään käyttäjän oikeudet haluttuun tiedostoon. Koska käyttöoikeudet haluttuun tiedostoon tarkistetaan pelkästään numeeristen UID:n ja GID:n avulla on UID:n ja GID:n oltava samat molemmissa koneissa.

Ennenkuin NFS:ää voidaan käyttää on varmistettava että ydin (kernel) tukee NFS:ää. Uudemmissa järjestelmissä tämä voidaan tarkistaa esimerkiksi kommennolla

```
$ cat /proc/filesystems
          ext2
          msdos
   nodev  proc
   nodev  nfs
```

Mikäli nfs puuttuu listasta on ydin (kernel) rakennettava uudestaan.

Mikäli halutaan että järjestelmä tarjoaa NFS palvelun muille koneille on järjestelmässä käynnistettävä *nfsd* ja *mountd*. RPC-pohjaisina niitä ei voi hallita *inetd* kautta. NFS- ja mount-palvelimet on käynnistettävä järjestelmän käynnistyksessä sen jälkeen kun *portmap*-demoni on käynnistetty.

Sen mitä hakemistoja *nfsd* tarjoaa mountattavaksi määritellään `/etc/exports` tiedostossa. Oletuksena *nfsd* ei salli minkään hakemiston mountausta.

`/etc/exports` tiedoston muoto on seuraava:

```
/dir hosts(options)
```

Esimerkki `/etc/exports` tiedostosta

```
/ jane.math.jyu.fi(ro,no_root_squash)
/cdrom *.jyu.fi(ro) *.*.jyu.fi(ro) *.*.*.jyu.fi(ro)
/spirs/1 *.math.jyu.fi(ro) *.math.public.jyu.fi(ro) *.it.jyu.fi(ro)
```

Lisätietoa *nfs(5)*, *exports(5) nfsd(8)*

12.6 Linux automounter

Isommissa verkkojärjestelmissä tulee usein vastaan tilanteita joissa eri koneiden hakemistoja käytetään ristiin toisissa koneissa NFS:n yli. Muutamat koneet saattavat olla kuitenkin osan aikaa poissa käytöstä, joten pysyvien NFS mounttien hallinta on työlästä. Asia voidaan hoitaa käyttämällä *automount* järjestelmää. Automount järjestelmässä tietyn hakemiston mounttaus hoidetaan automaattisesti kun hakemistossa olevia tiedostoja tarvitaan. Hakemiston mounttaus hoidetaan automount demonilla. Myöskin hakemistojen umounttaus tehdään automaattisesti mikäli hakemistossa olevia tiedostoja ei ole tarvittu tietyn ajan kuluessa. Linuxsissa on käytössä (ainakin) kaksi erilaista mahdollisuutta toteuttaa hakemistojen automaattinen mounttaus: *AMD* ja *autofs*.

AMD (automount daemon) toimii samaan tapaan kuin SunOS:n AMD ja se on toteutettu "käyttäjätilassa"eli se ei ole ydin (kernel) pohjainen järjestelmä. Ytimen ei välttämättä tarvitse olla tietoinen automounttauksesta jos hakemisto mountataan NFS:n yli käyttäen AMD demonia.

Toinen tapa hoitaa hakemistojen automaattinen mounttaus on *autofs*. Autofs systeemissä ydin tietää missä automount hakemistot ovat ja *automount*-ohjelma hoitaa hakemistojen mouttauksen.

Mikäli halutaan käyttää *autofs* systeemiä on ytimessä oltava tuki mukaan otettuna. (autofs tuen pitäisi olla vakaa ytimissä joiden versio numero on suurempi kuin 2.2.xx)

Autofs systeemi tarvitsee myös *automount*-ohjelman ja konfiguraatiotiedostot
`/etc/auto.master` ja esim. `/etc/auto.misc`.

Esimerkki `/etc/auto.master` tiedostosta:

```
/tmp_mnt /etc/auto.misc --timeout 60
```

Ensimmäinen kenttä kertoo mihin toisen kentän osoittamat mount hakemistot sijoitetaan (ensimmäinen kenttä ei siis ole kokonainen mount hakemisto). Kolmas kenttä kertoo että tiedostojärjestelmät yrittävät umountata itsensä 60 sekunnin kuluttua, jos ne eivät ole käytössä.

Esimerkki `/etc/auto.misc` tiedostosta:

```
sources        -ro,soft,intr       ftp.math.jyu.fi:/ftp/pub/linux
cd             -fstype=iso9660,ro  :dev/cdrom
zip            -fstype=auto        :dev/hdd4
floppy         -fstype=vfat        :dev/fd0
```

Ensimmäinen kenttä kertoo mount hakemiston. Tässä tapauksessa se olisi `/tpm_mnt/cd` jne. Toinen kenttä määrittää optiot. Kolmas kenttä määrittää mikä tiedostojärjestelmä mountataan. 'sources' on NFS mount ja : muissa tarkoittaa, että kyseessä on paikallinen laite.

12.7 Varmuuskopiointi

Yksi järjestelmäylläpitäjän tärkeimmistä tehtävistä on tietojärjestelmien tiedostojen varmuuskopiointi. Varmuuskopiointi tulee suunnitella huolella. Suunnittelussa kannattaa kiinnittää huomioita ainakin minkälaiselle laitteelle varmuuskopiot tehdään, mitä työkalua käytetään, mitä tiedostoja halutaan varmistaa jne.

Varmuuskopiontimedian valintaan vaikuttaa ainakin sen luotettavuus, hinta, nopeus, saatavuus ja käytettävyys. Median luotettavuus on tärkeimpiä kriteereitä, koska rikkoutunut varmuuskopio on hyödytön. Usein varmuuskopioiden tulee myös säilyä useita vuosia. Mikäli varmuuskopiointi voidaan tehdä silloin kun koneen kuormitus muuten on vähäistä (esimerkiksi yöllä) ei varmuuskopiointisysteemin nopeudella ole niin suurta merkitystä. Varsinkin kun varmuuskopiointi hoidetaan yleensä siten ettei käyttäjän tarvitse sitä valvoa. Käyttettävyys on myöskin tärkeää varsinkon jos varmuuskopiointi suoritetaan usein. Myöskin tiedon palautuksen kannalta vaivaton käytettävyys on tärkeä.

Työkaluja varmuuskopioiden tekoon on useita. Perinteisesti käytetyjä ovat: *tar*, *cpio* ja *dump*. Myös paljon muita sekä kaupallisia että ilmaisia työkaluja on tarjolla.

tar ja *cpio* ovat käyttäjän kannalta lähes samanlaisia. Molemmilla voidaan kirjoittaa ja lukea tiedostoja nauhalta. Myöskin nykyisin linuxissa käytössä olevat *tar* ja *cpio* versiot kykenevät lukemaan erikoistiedostot oikein (esim laitetiedostot, symboliset linkit, pitkät tiedostonimet jne.)

Yksinkertaisin tapa varmuuskopiointiin on tehdä ensin täydellinen varmuuskopio (full backup) järjestelmästä (tai halutusta osasta järjestelmää) ja sen jälkeen varmuuskopioidaan vain muuttuneet tiedostot (incremental backup).

Esimerkiksi *GNU tar* ohjelmaa käyttäen tämä voidaan tehdä seuraavasti:

```
tar -cvf /dev/tape /home
```

Varmuuskopioinnin onnistuminen voidaan tarkistaa seuraavasti:

```
tar --compare -vf /dev/tape
```

Tämän jälkeen muuttuneet tiedostot voidaan varmuuskopioida komennolla:

```
tar -c --newer='Apr 18 1999' -f /dev/tape /home
```

Edellä kuvattu varmuuskopiointi tapa soveltuu hyvin pieniin järjestelmiin ja esimerkiksi kotona käytettävän mikron varmuuskopioiden tekemiseen.

Suurien järjestelmien varmuuskopiointiin kannattaa käyttää kuitenkin monitasoisia varmuuskopioita (multilevel backups). Yksinkertaisin tapa on siis käyttää kahta tasoa: täysi varmuuskopio ja siihen verrattuna muuttuneet tiedostot. Tätä samaa systeemiä voidaan kuitenkin laajentaa miten moneen tasoon tahansa.

Aluksi tehdään täydellinen varmuuskopioi (0-taso) ja 1-tason varmuuskopioa tehtäessä verrataan muuttuneet tiedostot 0-tasoon ja 2-tason varmuuskopiossa verrataan 1-tasoon jne.

Usean tason varmuuskopioinnin tarkoituksena on, että se mahdollistaa pidemmän varmuuskopiohistorian 'halvemmalla'.

Monitasoinen varmuuskopiointi on sisäänrakennettu *dump* ohjelmaan. *tar* ja *cpio* ohjelmiin se on toteutettava komentojonojen avulla.

Lisätietoja *tar(1), cpio(1L), dump(8)*

12.8 Muita asioita

12.8.1 Linux man sivut

Unix järjestelmässä yleensä kaikki järjestelmän mukana tulleet ohjelmat sisältävät käyttöohjeen. Nämä käyttöohjeet on toteutettu ns. man-sivuina. Man sivuja voi selata *man* ohjelman avulla. Parametrinä ohjelmalle annetaan ohjelma jonka ohjeita halutaan. *man* ohjelma lukee oletuksena konfiguraatiotiedoston `/etc/man.config`. Tiedostossa määritellään mm. mistä hakemistoista man-sivuja oletuksena haetaan. Lisätietoa *man(1)*