

## 1.1 Käsiteltävät asiat:

- Unix monen käyttäjän ympäristönä
- Prosessien tarkastelu ja hallinta
- Miten verkko toimii (perusteet)
- ssh-etäyhteydet
- Tietojen hakeminen itsestä ja muista
- Ympäristömuuttujat
- man-sivujen sisällön ymmärtäminen
- Käyttäjien välinen keskustelu

## 2.1 Monikäyttäjäisyys — yleiskuva

- Koneeseen voi kirjautua sisään eri reittejä: näyttö ja näppis, vanhanaikaiset päätteet, verkon (esim. Internet) yli toimivat etäyhteydet
- Kaikilla sisäänkirjautumistavoilla periaatteessa sama ympäristö, komentotulkki
- Koneessa on yhtaikaa toiminnassa useita *prosesseja*, eli käynnissä olevia ohjelmia (esim. kunkin sisäänkirjautuneen käyttäjän komentotulkki)
- Prosessit toimivat suojattuina toisiltaan, mikä sallii useampien käyttäjien ajaa helposti omia prosessejaan koneessa yhtaikaa

## 2.2 Puuduttavat yksityiskohdat [ylim]

- Kaikki prosessit toimivat jonkin käyttäjän oikeuksilla (tätä käyttäjätunnusta kutsutaan prosessin *vaikuttavaksi käyttäjätunnukseksi*, "euid")
- Erilaiset koneeseen päästävät järjestelmät (ssh-palvelin, sisäänkirjautumisruutu) toimivat ylläpitäjän oikeuksilla
- Käyttäjän tullessa sisään ne varmistavat, että hän on kuka väittääkin olevansa (esim. salasana)
- Tämän jälkeen ne käynnistävät käyttäjän oletuskomentotulkin (login shell) käyttäjän oikeuksilla
- Kaikki tästä komentotulkista käynnistetyt prosessit perivät sen oikeudet, ts. komentotulkista ajettu ohjelma toimii samoilla oikeuksilla kuin komentotulkki

Date: 2004/02/05 20:18:37

## 2.3 Graafinen käyttöympäristö

- Periaatteessa yksi ohjelma muiden joukossa (mutta toimii korotetuilla oikeuksilla voidakseen piirrellä näytölle suoraan)
- Muut ohjelmat, *X-asiakkaat* (esim. mozilla) ottavat siihen yhteyttä DISPLAY-ympäristömuuttujan perusteella (ks. ??, Ympäristömuuttujat) [ylim]
- Tästä syystä graafista käyttöympäristöä kutsutaan *X-palvelimeksi* ja siinä voi olla myös toisella koneella toimivien ohjelmien ikkunoita
- Graafisten ohjelmien kaatuessa havaittava ikkunoiden yhtäkkinen katoaminen on seurausta siitä, että käyttöjärjestelmä sulkee kaatuneen ohjelman kaikki yhteydet

### 3.1 Prosessit — yleiskuva

- Prosessi tarkoittaa ohjelman yhtä käynnissä olevaa instanssia; jokaisella prosessilla on yksiselitteinen tunnusnumero, "pid"
- Kaikkia prosesseja ajetaan periaatteessa toisistaan riippumatta yhtäikaa, mutta ne voivat odotella toisia prosesseja
- Omista ja muiden prosesseista on nähtävillä samat tiedot
- Prosesseja hallitaan lähettämällä niille *signaaleja*, jotka ovat käytännössä pyyntöjä / käskyjä prosessille tehdä jotain
- Käyttäjä voi lähettää signaaleja vain omille (eli omilla oikeuksillaan toimiville) prosesseilleen
- Reagointi signaaleihin vaihtelee jonkin verran ohjelmasta riippuen, mutta on enimmäkseen arvattavaa

## 3.2 Prosessilistaus

- Ensisijainen ohjelma prosessilistauksen tarkasteluun on `ps`.
- Kaziljoona komentolinjaoptiota listauksen muunteluun, jotka (Linuxissa) annetaan ilman viivaa, esim. `ps auwx`
- Listauksen pidennys: `a` = kaikki interaktiiviset prosessit, `x` = myös taustaprosessit (esim. palvelimet)
- Lisää tietoa: `l` = "pitkä" listaus, `u` = näyttää myös, kenen oikeuksilla mikäkin prosessi toimii (pelkkä `ps u` laajentaa myös listauksen kaikkiin omiin prosesseihin)
- Lisää rakennetta: `w` = näyttää pidemmätkin linjat, `f` = näyttää, mikä prosessi on käynnistänyt minkäkin (eli prosessihierarkia)

- Itse asiassa `ps`:lla on myös viivan kanssa annettavia optioita, jotka aiheuttavat (joskus) eri asioita kuin viivattomat (`huoh`)
- Useimmilla komentolinjaoptioilla ei tee mitään
- Muita hyödyllisiä ohjelmia: `pstree`, `top`

★ `alias psg='ps wuxa | grep'` on huomattavan hyödyllinen, kun yrittää esim. selvittää, onko jokin graafinen ohjelma käynnistynyt vai ei

### 3.3 Signaalien lähettäminen

- Signaaleja tarvitaan esim. jumittuneen ohjelman kaatamiseen, joskus johonkin sivistyneempäänkin
- Ensisijainen komento tähän on `kill`, kehittyneempi versio siitä `killall`
- Molemmat ottavat kaksi argumenttia: mikä signaali lähetetään ja mille prosessille, esim. `kill -TERM 2094`
- `kill` ottaa prosessinumeron (pid), joita voi selvittää `ps`:lla; `killall` ottaa prosessin ohjelman nimen ja lähettää signaalin kaikille tästä ohjelmasta käynnistetyille prosesseille.
- Signaaleja lähtee prosesseille myös tietyissä tilanteissa automaattisesti.

Date: 2004/02/05 20:18:37



### 3.4 Käyttäjän lähettämien signaalien merkitys

Monilla signaaleilla on hyvin vakiintunut merkitys.

Signaali	merkitys
TERM	pyyntö lopettaa
INT	Keskeytyspyyntö (Ctrl-C lähettää)
TSTP	Pysäytyspyyntö (Ctrl-Z lähettää) (ks. ??, Työnhallinta)
KILL	lopettaa ohjelman heti

Käytännössä, jos haluaa lopettaa ohjelman, sille kannattaa lähettää ensin TERM- ja sitten KILL-signaali. Joitain taustaprosesseja hallitaan signaaleilla; merkitykset löytyvät kyseisten ohjelmien man-sivulta.

### 3.5 Käyttöjärjestelmän lähettämiä [ylim]

Signaali	milloin	mitä tyypillisesti tekee
HUP	yhteys menee poikki	sama kuin TERM
PIPE	putken vastaanottava pää suljetaan	sama kuin TERM
SEGV	prosessi yritti käyttää muistialuetta, joka ei kuulu sille	prosessin tilasta talletetaan kuva, "core", vikaselvitystä varten
WINCH	ikkunan kokoa on muutettu	ohjelma muuttuu uuteen ikkunaan sopivaksi
TTIN	prosessi on taustalla (ks. ??, Työhallinta) ja yritti lukea syötettä	prosessi pysäytetään

Date: 2004/02/05 20:18:37

### 3.6 Työnhallinta

- “Job control”, komentotulkin ja käyttöjärjestelmän yhteistyönä toteutettu tapa ajaa nätisti useampia ohjelmia yhtäaikaan
- Useimmille ohjelmille läpinäkyvä
- Yksittäisestä komennosta käynnistyneet prosessit ovat yksi *työ* (job)
- Työllä kolme tilaa: edustalla, taustalla ja pysäytetty
- Vain yksi työ (tai komentotulkki itse) on edustalla kerrallaan
- Komentotulkki odottaa edustalla olevan työn päättymistä, taustalla olevat taas toimivat omia aikojaan (voit antaa muita komentoja sillä välin).

Date: 2004/02/05 20:18:37

- Työ käynnistetään komentotulkista taustalle panemalla komennon perään `&`-merkki
- Edustalla olevan työn voi pysäyttää (ja palata komentotulkkiin) painamalla `Ctrl-Z`
- Komentotulkista työt voi listata komennolla `jobs`, ja niihin viitataan merkinnöillä `%1 ... %n`
- Taustalla olevan tai pysäytetyn työn voi tuoda edustalle (ja käynnistää) komennolla `fg` tai antamalla työn numeron (esim. `%2`)
- Pysäytetyn työn voi käynnistää taustalle komennolla `bg`
- Myös `kill` ymmärtää %-määriytyksiä (esim. `kill %1`)

Date: 2004/02/05 20:18:37

## 4.1 Verkko — yleiskuva

- Tietokoneet / ohjelmat keskustelevat keskenään *protokollilla*, jotka ovat tavallaan käyttötarkoitukseltaan rajoitettuja kieliä
- Unixissa (ja Internetissä) käytännössä kaikki tietokoneiden välinen keskustelu hoidetaan TCP/IP-protokollalla (ja UDP/IP)
- TCP/IP-protokollassa jokaisella koneella on oma osoite, esim. 128.214.78.153, sekä jokaisessa koneessa 65536 mahdollista *porttia*. [ylim]
- TCP/IP tarjoaa mahdollisuuden muodostaa *yhteyksiä* kahden koneen kahden portin välille.
- Yhteys syntyy, kun yksi ohjelma kuuntelee tiettyä porttia ja toinen muodostaa yhteyden kyseiseen porttiin.

- Joitain portteja kuuntelee jatkuvasti *palvelinohjelma*, joka tarjoaa joitain tiettyjä palveluita ohjelmille, jotka ottavat siihen yhteyttä.
- Jokaisella palvelulla on oma protokolla, joka on suunniteltu kyseisen palvelun tarjoamiseen, sekä määrätty porttinumero, josta kyseinen palvelu tyypillisesti löytyy.

[ylim] Esimerkkejä yleisesti tunnetuista palveluista:

Palvelu	portti	protokolla
www	80	HTTP
sähköpostin lähetys	25	SMTP
sähköpostin luku	143	IMAP
ssh-etäyhteys	22	SSH
koneiden nimet	53	BIND

## 4.2 Esimerkki: WWW-sivun haku [ylim]

1. Naputtelen selaimen osoitteen, esim.  
`http://www.ling.helsinki.fi/atk/`
2. Selain ottaa yhteyttä nimipalvelimeen porttiin 53 ja kysyy BIND-protokollalla, mikä osoite on koneella  
`www.ling.helsinki.fi`
3. Saatuaan vastauksen (BIND-protokollalla) selain ottaa yhteyden kyseiseen osoitteeseen porttiin 80 ja pyytää (HTTP-protokollalla) sitä antamaan sivun `/atk/`
4. Vastauksena annetaan (HTTP-protokollalla) kyseinen sivu.  
Tyypillisesti sivu on HTML-muotoista tekstiä, jonka selain sitten näyttää ruudulla mielensä mukaan.

## 5.1 ssh — yleiskuva

- Antaa käyttää komentotulkkia palvelimella verkkoyhteyden yli
- Korvaa aiemmat etäyhteyksiprotokollat (rsh ja telnet)
- Lisäominaisuuksia: salakirjoitus (verkkosalakuuntelun estämiseksi), muiden yhteyksien salakirjoitettu tunnelointi, pöljöönat uudet autentikointitavat (siis muutakin kuin käyttäjätunnus ja salasana), tiedostonsiirto
- Monet ohjelmat osaavat myös käyttää `ssh`:a yhteydenmuodostustapana omille protokollilleen (ja siten palveluilleen), millä vältetään autentikoinnin ja salakirjoituksen uudelleentoteutus joka protokollaan.



## 5.2 Tunnelointi [ylim][ylim]

SSH-protokolla sallii *multipleksoida* yhteen SSH-yhteyteen monta merkkivirtayhteyttä. `ssh` käyttää tätä toteuttaakseen muiden yhteyksien "läpinäkyvän" salakirjoituksen, tunneloinnin.

Käyttäjän kone

```
bash
```

`ssh -L 2025:localhost:25 mars`

```
bash
ssh 3398
kuuntelee ssh 2025
```

`telnet localhost 2025`

```
bash
ssh 3398
kuuntelee ssh 2025
ssh 2025
telnet 3399
```

`mars.ling`

```
22 sshd kuuntelee
25 sendmail kuuntelee
```

```
22 sshd kuuntelee
25 sendmail kuuntelee
22 sshd, bash
```

```
22 sshd kuuntelee
25 sendmail kuuntelee
22 sshd, bash
4980 sshd
25 sendmail
```

Date: 2004/02/05 20:18:37

### 5.3 ssh-avainautentikointi

- Autentikointi käy avainparilla: sisään pääsee, jos yksityinen avain ja julkinen avain täsmäävät. Avaimista ei voi päätellä toisiaan.
- Avainpari muodostetaan komennolla `ssh-keygen -t dsa`
- Yksityinen avain on tallessa `.ssh/id_dsa`-tiedostossa, sen pari `.ssh/id_dsa.pub` lisätään kohdekoneen `.ssh/authorized_keys`-tiedostoon.
- Yksityisen avaimen voi suojata salasanalla.
- Salasanalla suojatun avaimen voi ladata muistiin komennoilla `ssh-agent` ja `ssh-add`. `ssh-agent:n` on tarkoitus olla käynnissä koko istunnon ajan.

Date: 2004/02/05 20:18:37

## 5.4 X-tunnelointi [ylim]

`ssh` käyttää tunnelointia (ks. ??, Tunnelointi [ylim][ylim]) graafisten ohjelmien X-yhteyksien suojaamiseen salakirjoituksella. Tähän liittyy jonkin verran lisäsäätöä, joka saa tunneloinnin toimimaan ilman, että tarvitsee tehdä mitään kummempaa.

X-ohjelmat ottavat yhteyttä käyttäjän X-palvelimeen DISPLAY-ympäristömuuttujan perusteella (ks. ??, Graafinen käyttöympäristö). Tunneloitaessa `ssh` perustaa etäkoneelle pseudo-X-palvelimen, joka siirtää X-yhteydet kryptatun ssh-yhteyden yli edelleen käyttäjän koneelle, ja asettaa DISPLAY-muuttujan osoittamaan tähän pseudopalvelimeen, esim. `:10.0`.

X-tunnelointia voi erikseen pyytää optiolla `-X`, siis esim. `ssh -X`.

## 5.5 ssh-tiedostonsiirto

- Yksinkertainen tiedostonsiirto: `scp` (kuten `cp`-komento, mutta sallii koneennimiä lähteessä ja kohteessa, esim. `scp pkalliok@mars:foo /tmp/bar`)
- Myös `scp -r` on mahdollinen
- Monen tiedoston siirtoon `sftp` on parempi (muistuttaa `ftp:a`)
- Kokonaisten hakemistopuiden varmuuskopiointiin ssh-putkitettu `rsync` on parempi, esim. `rsync -e ssh -avz dirika mars:kopiot`
- Kielitieteellä ssh-tiedostonsiirtoa ei tarvita kovin usein jaettujen kotikansioiden takia

## 6.1 Käyttäjätiedot — yleiskuva

- Unixissa käyttäjistä on tietoja monessa vakiintuneessa paikassa: `/etc/passwd`, tietyt tiedostot käyttäjän kotihakemistossa, sisään- ja uloskirjautumislokit (`/var/log/wtmp`), prosessilistaus (ks. `??`, Prosessilistaus)
- Useimpia näistä voi tarkastella suoraan esim. `less`:lla, ja monet ohjelmat käyttävät niitä tietolähteinään.
- Tietoja käyttäjätunnuksista ja niiden ryhmistä: `/etc/passwd`, `/etc/group`, `finger`, `id`, `groups`, `.plan`, `.project`, `.signature`
- Tietoja sisällä olevista / olleista käyttäjistä: `w`, `who`, `ps`, `last`

## 6.2 Yleisen kielitieteen händäystä

- Käyttäjätiedot levitetään yhteiseltä käyttäjätietopalvelimelta
- Niinpä käyttäjätietoja muutettaessa pitää ottaa yhteys tähän käyttäjätietopalvelimeen. `passwd`-komento tekee tämän.  
(Perinteisessä Unix-ympäristössä `passwd`-komennolla vaihdetaan paikallista salasanaa ja `chfn`-komennolla paikallisia käyttäjätietoja.)
- Käyttäjätiedot päivittyvät kaikkiin koneisiin noin tunnin sisällä muutoksesta.

★ Tehkää itsellenne lyhyet ja kuvaavat `.plan`- ja `.signature`-tiedostot!

## 7.1 Ympäristömuuttujat

- Jokaisella prosessilla on oma *ympäristönsä*, joka koostuu useammista *ympäristömuuttujista*. Esim. komentotulkkini ympäristössä muuttujan SHELL arvo on todennäköisesti /bin/bash.
- Ympäristömuuttujat ovat tavallaan kolmas tapa antaa prosessille syötettä. (Muut kaksi ovat komentoriviargumentit ja stdin.) Prosessi perii ympäristönsä vanhemmiltaan.
- Monien ohjelmien toimintaa voi muunnella tietyillä ympäristömuuttujilla. Esim. useat ohjelmat katsovat, mikä editori ajetaan tekstiä muokattaessa, ympäristömuuttujasta EDITOR.

## 7.2 Muuttujien käyttö komentotulkista

- Muuttujan voi viedä pysyvästi komentotulkin ympäristöön (josta se sitten periytyy käynnistettäville ohjelmille) komennolla `export MUUTTUJA=arvo`.
- Jos komennossa on merkkijono `$MUUTTUJA`, komentotulkki korvaa sen MUUTTUJAN arvolla. [ylim] Esim. komento `echo $MUUTTUJA` näyttää MUUTTUJAN arvon.
- Kaikki voimassa olevat ympäristömuuttujat voi listata komennolla `env`. (Tai `set`, joka näyttää muutakin.)
- Yksittäiselle komennolle voi määrittää ympäristömuutoksia näin: `MUUTTUJA1=arvo1 MUUTTUJA2=arvo2 komento...`



### 7.3 Laajalti käytettyjä ympäristömuuttujia [ylim]

PATH	hakemistot, joista suoritettavaa ohjelmaa etsitään, jos komentotulkille ei anneta täydellistä tiedostonimeä
HOME	käyttäjän kotihakemisto, sisältää useita ohjelmien käyttämiä .-alkuisia asetustiedostoja
EDITOR	ohjelma, jota halutaan käyttää tekstin muokkaukseen
PAGER	ohjelma tiedostojen katseluun (esim. <code>less</code> )
LANG	(ja kumppanit) määrittää kieliympäristön
PRINTER	oletustulostin
TERM	Mitä terminaalia käytät. Jos otat etäyhteyden eikä jokin toimi, vika on todennäköisesti TERM-asetuksessa.
TMPDIR	mihin väliaikaistiedostot perustetaan

## 8.1 man-sivut — yleiskuva

- man-sivut jakautuvat 9 *osioon* (ja alaosioiden) sen perusteella, minkä tyyppistä asiaa käsittelevät
- Jokainen sivu noudattaa verrattain vakiintunutta rakennetta ainakin jonkin verran, man-sivuissa on tusinan verran erilaisia vakiintuneita *osia*
- GNU-ohjelmien man-sivut ovat enimmäkseen roskaa (koska GNU-projektin ensisijainen dokumentaatiomuoto on info)

★ Psst. man-sivuilta (ja muutenkin `less`-ohjelmassa) voipi etsiä merkkijonoa kirjoittamalla `/merkkijono`, enter.

## 8.2 man-sivujen osiot

Osio	merkitys
1	tavalliset komennot
2	systemikutsut (ohjelmoijille)
3	kirjastokutsut (ohjelmoijille)
5	tiedostomuodot
8	systemikomennot (ylläpitäjille)

man-sivuja etsitään osiosta järjestyksessä pienestä suureen. Jos samanniminen sivu on useammassa osiossa, joutuu joskus antamaan osion erikseen. Esim. komento `man crontab` tuo `crontab`-komennon man-sivun, kun taas komento `man 5 crontab` tuo `crontab`-tiedostomuodon man-sivun.

### 8.3 man-sivun osat

NAME	lyhyt yleiskuvaus ohjelmasta
SYNOPSIS	lyhyt kuvaus komennon muodosta — hakasulkeissa olevat asiat ovat valinnaisia
DESCRIPTION	selitys siitä, mitä ohjelma tekee ja miten
OPTIONS	komentorivalitsimien merkitykset
DIAGNOSTICS	ohjelman palautteen merkitys
EXAMPLES	käyttöesimerkkejä (lue ensin!)
ENVIRONMENT	toimintaan vaikuttavat ympäristömuuttujat
FILES	ohjelman käyttämät tiedostot
SEE ALSO	muita ohjelmaan liittyviä man-sivuja
BUGS	omituksuuksia, tunnettuja ongelmia, varoituksia

Date: 2004/02/05 20:18:37

## 9.1 Käyttäjien välinen keskustelu

- Toimii perinteisesti siten, että saman koneen käyttäjät kirjoittelevat toistensa päätteille.
- Muiden viestit omalle ruudulle voi sallia tai kieltää komennolla `mesg. [ylim]`
- Yksittäisen viestin kirjoittamiseen on ohjelma `write`, argumentiksi kohdekäyttäjän tunnus. Viesti lopetetaan Ctrl-D:lla.
- Pidempään keskusteluun on ohjelma `ytalk` (tai `talk`). Keskustelu lopetetaan Ctrl-C:lla.
- Jos tarvitset jotain vaativampaa, käytä esim. IRC:a. [ylim]