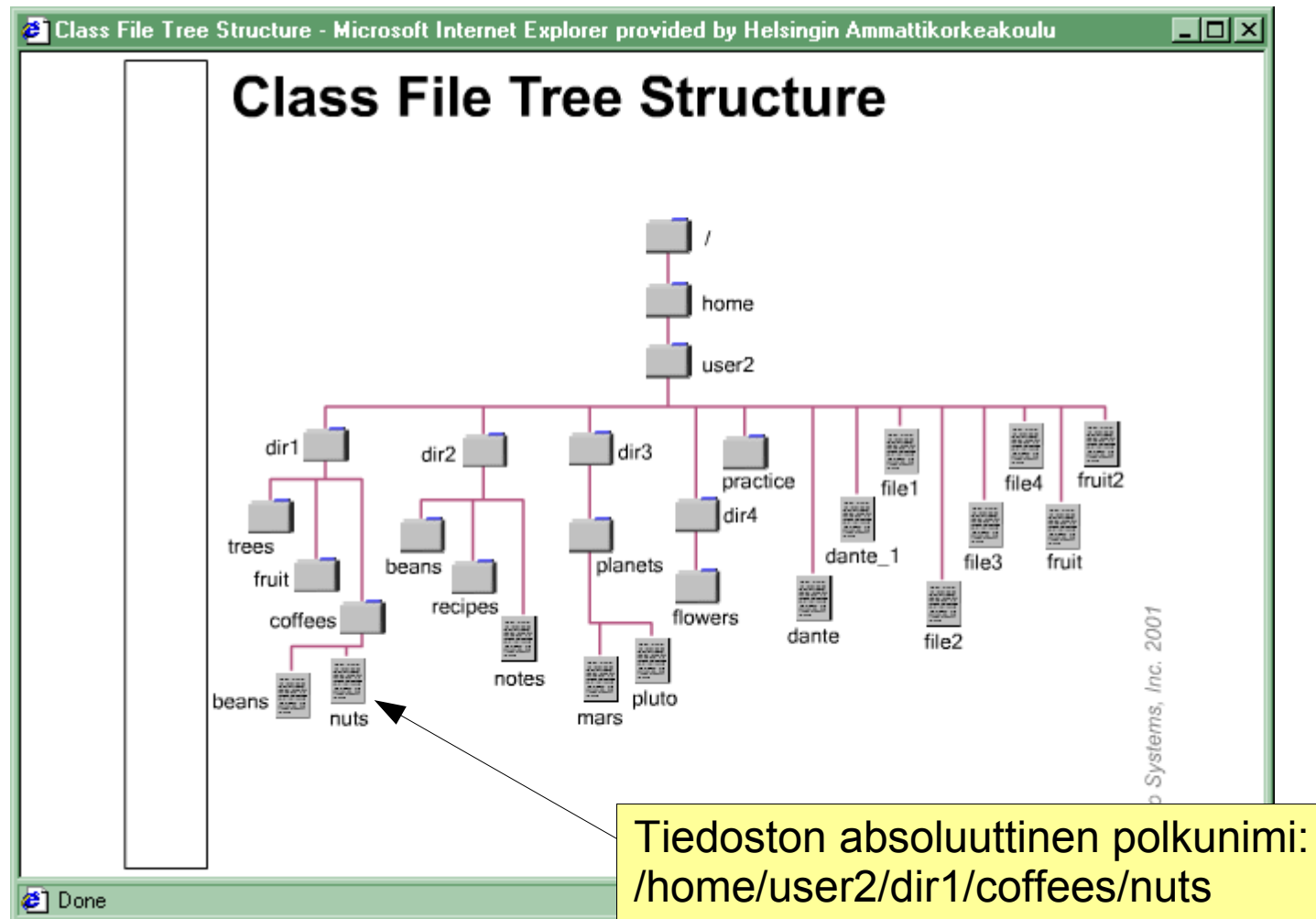


Unix-perusteet

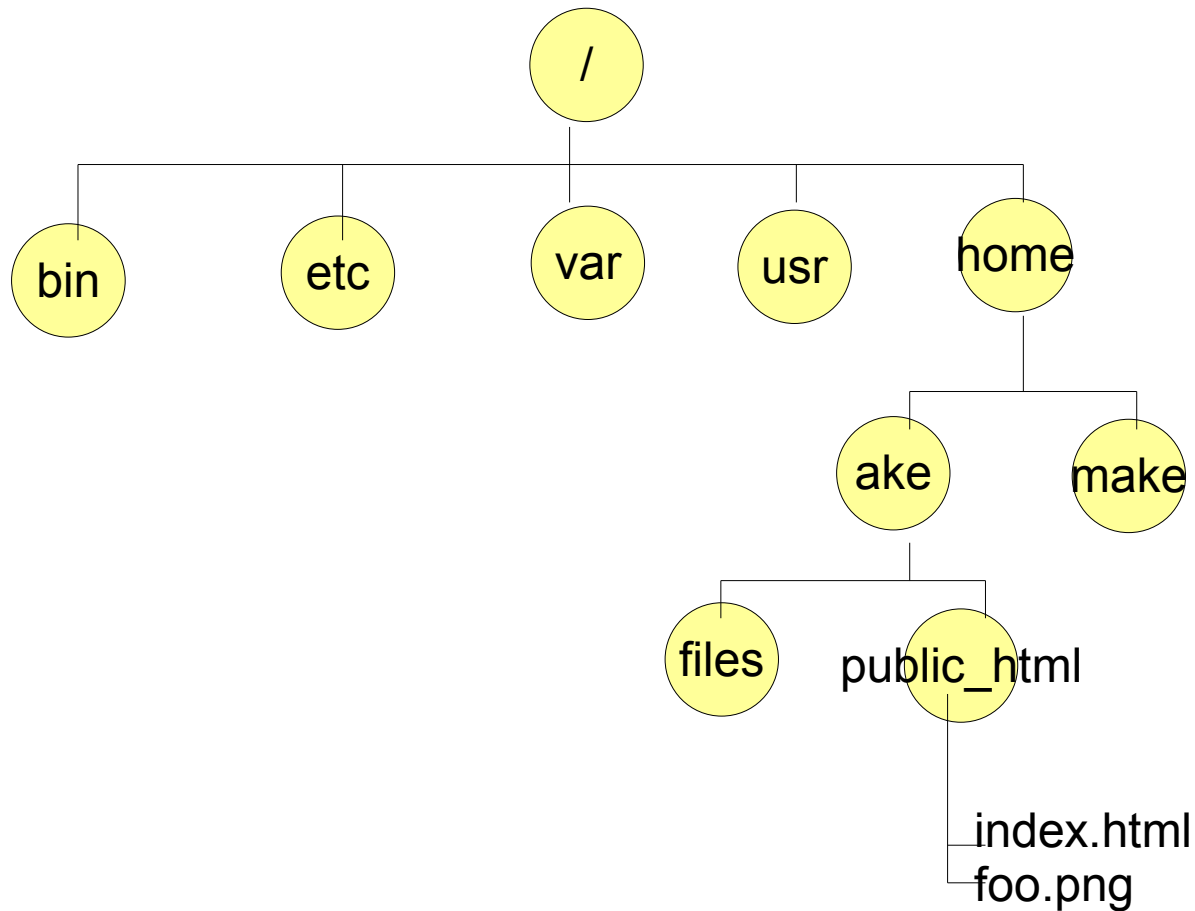
Hakemistot ja tiedostot

Tiedostojärjestelmä



Unixin yleinen hakemistorakenne

- De facto -standardi: <http://www.pathname.com/fhs/>



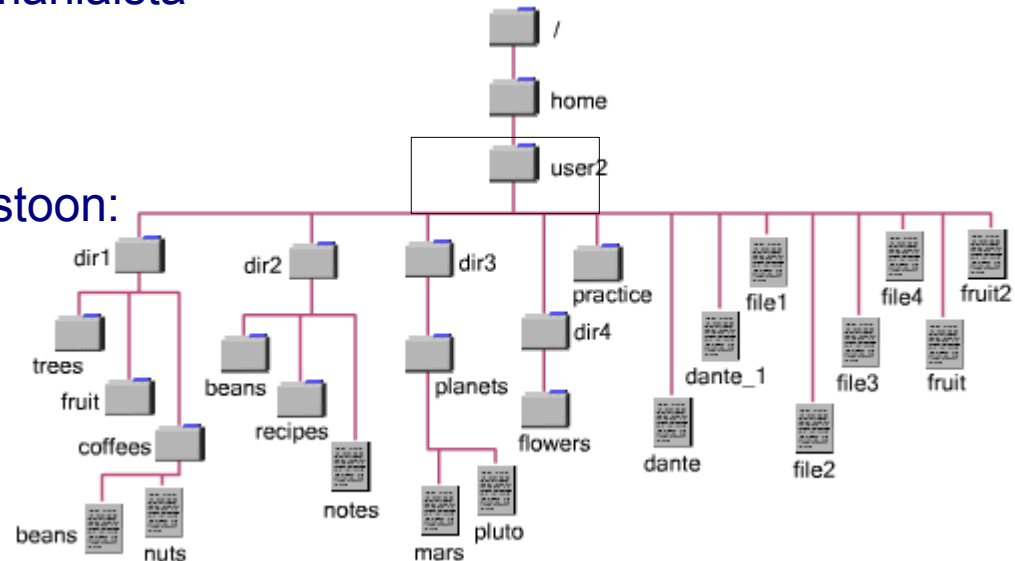
Milloin hakemistoja luodaan?

- Käyttöjärjestelmän asennuksessa
- Ohjelmia asennettaessa
- Pääkäyttäjän toimesta
- Tavallisen käyttäjän toimesta
 - tämä luonnollisesti käyttäjän oikeuksien mukaan

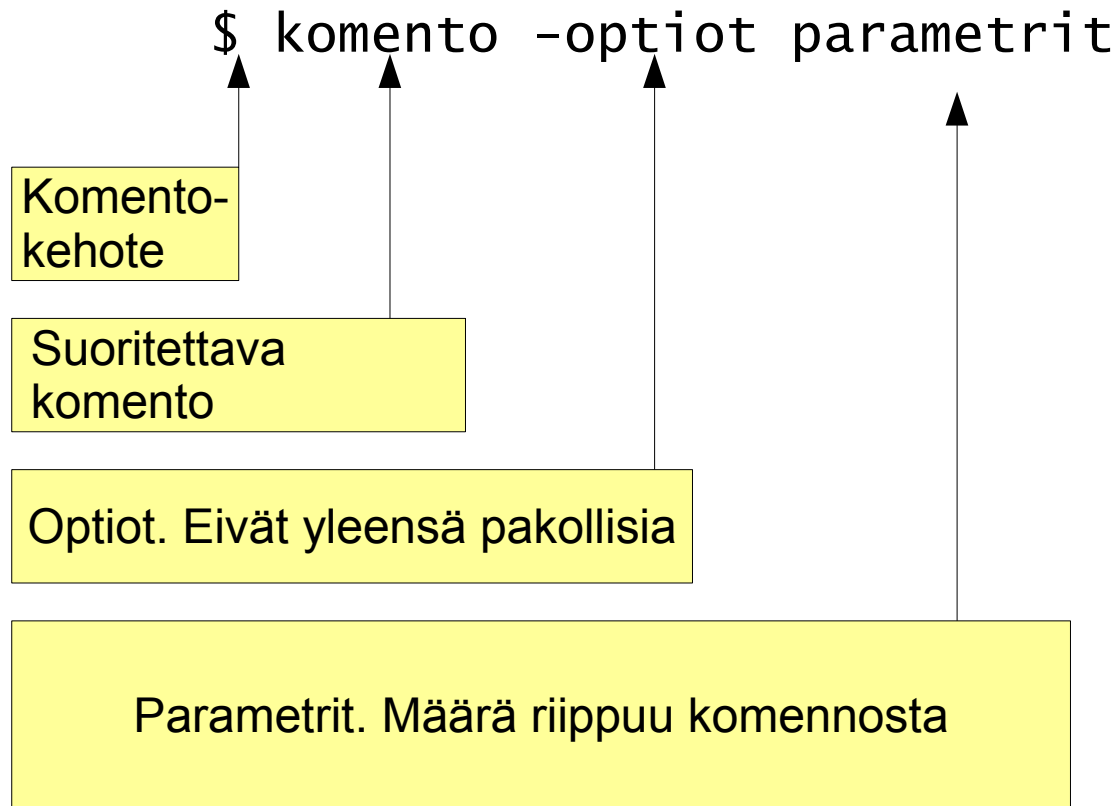
- Käytä hakemistoja datan organisoimiseen
- Periaatteessa hakemisto on vain tiedoston erikoistapaus

Absoluuttinen ja suhteellinen polkunimi

- Absoluuttinen polkunimi
 - koko polku juuresta alkaen
 - /home/user2/dir3/planets/mars
 - yksikäsitteinen, ei kahta samanlaista
- Suhteellinen polkunimi
 - polku suhteessa työhakemistoon:
 - dir3/planets/mars
 - kun työhakemisto user2
 - ei ala koskaan / :lla!



Komentojen yleinen muoto



HUOM! joihinkin distroihiin on luotu runsaasti ns. aliaksia komennoille. Siksi se näyttää usein toimivan näitä sääntöjä vastaan. Ilmeisesti on haluttu matkia Windowsin komentorivin toimintaa.

Toimi kuitenkin unix-sääntöjen mukaisesti! Siitä on sinulle enemmän hyötyä! Opi pois Windows-komentotulkista!

Jokaisen osan väliin välilyönti. Ehdottomasti!

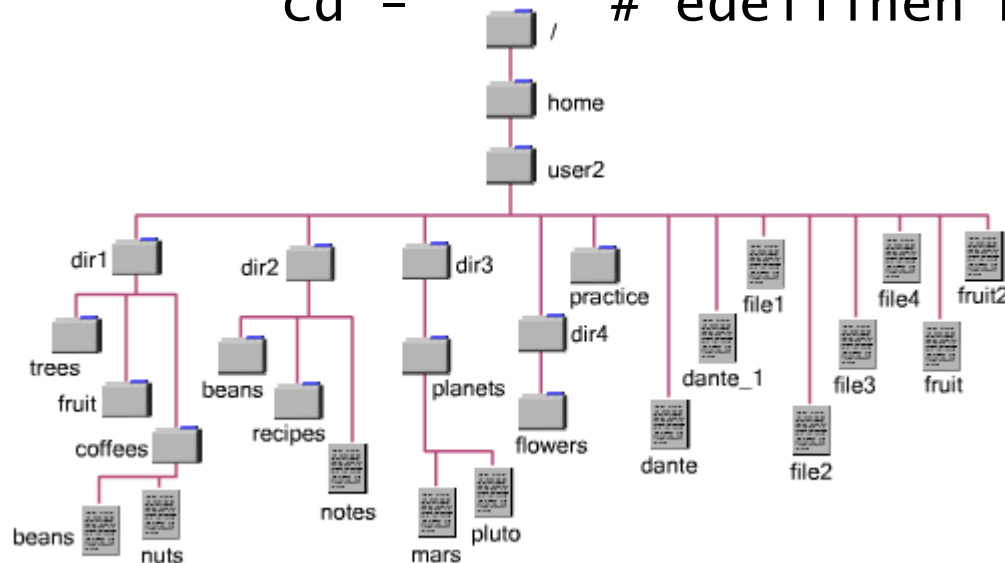
Esimerkkejä

```
ls  
cp file1 file2  
wc -w tiedosto  
cd dir2  
clear; date # puolipisteellä useita komentoja
```

cd

- Vaihtaa hakemistoa

```
cd dir1 # menee hakemistoon dir1
cd      # kotihakemistoon ihan mistä vain
cd /tmp # hakemistoon /tmp
cd ..   # hakemistotaso ylöspäin
cd -    # edellinen hakemisto (bash)
```

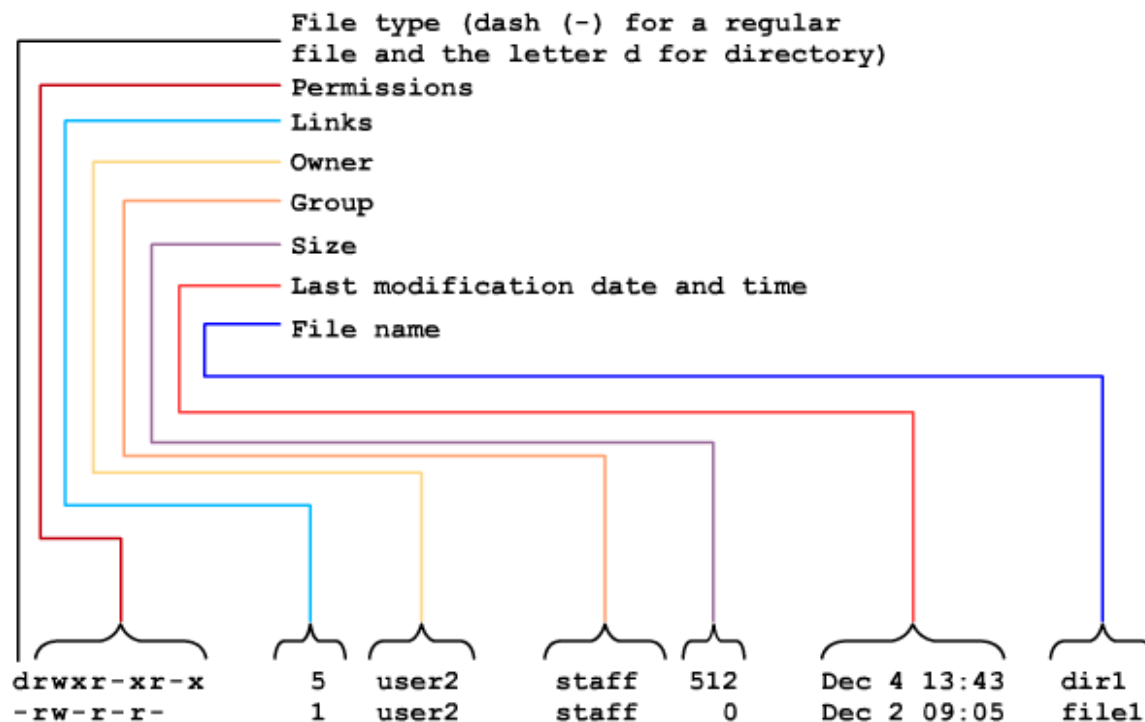


Hakemistotunnus .
viittaa aina työhakemistoon

ls

- Näyttää hakemiston sisällön

- `ls` # hakemiston tiedostonimet
- `ls -a` # myös piilotiedostot
- `ls -l` # pitkä listaus



1. tiedoston tyyppi
2. oikeudet
3. linkit
4. omistaja
5. ryhmä
6. koko
7. muutospäivä
8. tiedoston nimi

Jokeri- eli korvausmerkit

- * - mitä merkkejä tahansa
 - ls *txt - txt-päätteiset tiedostot
 - cat kis* - kis-alkuiset tiedostot
- ? - mikä tahansa merkki
 - ls ??? - kaikki nelimerkkiset tiedostot
- [lista] - mikä tahansa luetelluista merkeistä
 - ls kis[abc] - tiedostot kisa, kisb, kisc, muttei esim. kis1, kisz, kissa
 - ls kis[1-9] - aluettakin voi käyttää.
 - ls kis[!1-9] - huutomerkki aiheuttaa negaation, eli ei nämä

Muita erikoismerkkejä

- ; - komentojen erotin
 - `clear; cd; ls`
- | - putki, eli edellisen komennon tuloste seuraavalle
 - `ls | wc -l` (mitä tekee?)
- > - tulostuksen ohjaus tiedostoon
 - `ls > lista.txt`
- >> - tulostuksen ohjaus tiedoston perään
 - `ls /home >> lista.txt`
- < - lukeminen tiedostosta
- Vältä siis näiden käyttöä tiedostojen nimissä!
- Näiden käyttöön palataan myöhemmin tarkemmin

Muita hyödyllisiä komentoja

- pwd
 - Tulostaa työhakemiston (print working directory)
- more & less
 - Sivutusohjelmia. Näillä voi katsella tulosteita sivu kerrallaan:
 - less file - näyttää tiedoston file sivuttain
 - ls | less - tiedostolistaus sivu kerrallaan

Hakemistot ja tiedostot

Perustoimintoja

Tiedostojen nimeämisestä

- isot ja pienet kirjaimet erotellaan!
- pisteellä alkava tiedosto on ns. piilotiedosto
- maksimipituus 255 merkkiä
 - kuitenkin ei suositella kovin pitkien nimien käyttöä
- aakkosnumeeriset merkit (kirjaimet ja numerot) suositeltavia sekä erikoismerkeistä '.', '-' ja '_'
 - -merkkiä ei kuitenkaan nimen alkuun!
- muutkin sallittuja, nimeäminen erittäin vapaata
 - vihje: käytä vain sellaisia merkkejä, joita varmasti osaat käyttää!
- tiedoston nimet voivat sisältää yhden tai useamman tarkentimen (extension, erotetaan nimestä pisteellä)
 - ei kuitenkaan samanlaista merkitystä kuin Windowsissa
- hakemistojen nimissä ei yleensä ole tarkentimia, mutteivät ne ole kuitenkaan kiellettyjä

Unixissa voi tiedoston nimeen laittaa kahta merkkiä lukuunottamatta minkä tahansa merkin. Kielletyt merkit ovat '/' ja '\0', eli NULL

Kontrollimerkit

- CTRL - S ← Kokeile: kirjoita komentoriville ls -R / ja paina enter. Pysäytä CTRL-S:llä ja jatka CTRL-Q:lla
 - pysäyttää ruudun rullauksen
- CTRL - Q ← Kokeile: kirjoita komentoriville cat ja paina enter. Pois pääset CTRL-C:llä
 - jatkaa
- CTRL - C ← Kokeile: sulje komentoikkuna tällä.
 - keskeyttää sen hetkisen toiminnan
- CTRL - D ← Kokeile: sulje komentoikkuna tällä.
 - osoittaa tiedoston päättymisen tai poistutaan
- CTRL - U
 - komentorivin puhdistus
- CTRL - W
 - edellinen sana pois riviltä
- CTRL - H
 - poistaa edellisen merkin

file

- kertoo tiedostojen tyyppin
- hyvin käyttökelpoinen, koska muuten unixin tiedostonimet harvemmin sisältävät tietoa tiedoston tyyppistä

```
[kuivanen@chinay testikansio]$ file *
a.out: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
for GNU/Linux 2.2.5, dynamically linked (uses shared libs), for GNU/Linux 2.2.5,
not stripped
argentiina250706.ods: Zip archive data, at least v2.0 to extract
attachment.doc: Rich Text Format data, version 1, ANSI
lueminut.txt: ISO-8859 text, with CRLF line terminators
testi.c: ASCII C program text
thunderbird_ase.txt: ASCII text
[kuivanen@chinay testikansio]$
```


strings

- kaivaa tekstin esille mistä tahansa tiedostosta
 - strings tiedosto
 - strings tiedosto | less
 - sama mutta ruutu kerrallaan
 - kätevä esim. voittuneiden tiedostojen käsittelyssä
 - tai tiedoston luontiohjelman selvittämiseen

cat

- ohjelma tekstitiedostojen katseluun ja niiden yhdistämiseen
- oikeasti tekstitiedostojen yhteenliittämiseen (CATenate)
 - cat tiedosto
 - cat tiedosto1 tiedosto2 tiedosto3
 - cat file*
- yhdistäminen toimii näin:
 - cat file1 file2 file3 file4 > isofile
 - taikka siis `cat fle[1-4] > isofile`

katselutoiminto tulee todellisuudessa unixin tietovirtojen periaatteesta: kun ei osoiteta kohdetta, minne liitetään (ed. esimerkissä > isofile), niin liitos tulostetaan ns. standard outputiin, eli näytölle. Näistä lisää tuonnempana....

more ja less

- sivutusohjelmia, joilla katsellaan tekstitiedostoja sivu kerrallaan
 - less tiedosto
 - more tiedosto
 - more file1 file2 file3
 - unix-sanonta: "less is more"! (eli less on monipuolisempi)
- putken avulla näitä voidaan käyttää muiden komentojen tulosteen sivuttamiseen (unixin työkaluajattelua!)
 - missään yksittäisessä ohjelmassa ei ole sivutus-optiota
 - ls -l /etc | less
 - tulostaa /etc-hakemistosta pitkän listauksen sivu kerrallaan

head ja tail

- head: rivejä tiedoston alusta (oletuksena 10 riviä)
 - head -5 /usr/share/dict/words
 - viisi ensimmäistä riviä tiedostosta
- tail: rivejä tiedoston lopusta tai alkaen riviltä
 - tail -30 /usr/share/dict/words
 - 30 viimeistä riviä
 - tail +40 /usr/share/dict/words
 - rivistä 40 alkaen loppuun asti
- mihin mm. käytetään?
 - lokitiedostojen tutkimiseen; niistä usein kiinnostava osa vain jokunen rivi lopusta

WC

- Word Count

- kertoo oletuksena tekstitiedoston merkkien, sanojen ja rivien määrän
- tulostusta rajoitetaan optioilla

- `wc teksti`

- `wc -l teksti` # vain rivien lukumäärä

- `wc -wl teksti` # sanat ja rivit

- `wc -c teksti` # tavujen (merkkien) määrä

- `-m` -optiota ei ole GNU-versiossa (Linux), Solariksessa on

diff

- kahden tekstitiedoston sisällön vertailemiseen
 - kertoo, miten tiedostoja pitää muuttaa, että ne olisivat samanlaiset
 - versioiden hallintaan ym. erojen selvittämiseen

```
[kuivanen@chimay coursefiles]$ diff -c fruit fruit2
*** fruit      1999-03-07 09:52:10.000000000 +0200
--- fruit2     1999-03-07 09:53:12.000000000 +0200
*****
*** 2,8 ****
    orange
    apple
    banana
-   pear
-   mango
    tomato
    pomegranate
--- 2,8 ----
    orange
    apple
    banana
    tomato
+   guava
+   mango
-   pomegranate
[kuivanen@chimay coursefiles]$
```

Muita vastaavia:

- comm
- cmp

touch

- muuttaa tiedoston aikaleiman
 - jos vaikkapa halutaan joku tiedosto mukaan varmistukseen
 - ei käsittele muuten tiedostoa
- jos tiedostoa ei ole, luo tyhjän tiedoston
- esim:
 - touch testi testi2
 - luo joko kaksi tyhjää tiedostoa taikka vaihtaa näiden aikaleiman

mkdir

- luo uusia hakemistoja
 - mkdir eka
 - luo yhden uuden hakemiston
 - mkdir eka toka kolmas
 - kolme uutta hakemistoa
 - mkdir -p eka/toka/kolmas
 - luo koko hakemistorakenteen, jos sitä ei ole
 - mkdir eka toka toka/kolmas toka/neljas
 - toimii myös näin

rm

- poistaa tiedostoja
- perus-unixeissa ei ole "roskakoria", joten on syytä olla varovainen
 - Gnomen ja KDE:n roskikset pystyvät palauttamaan vain näiden ympäristöjen kautta poistettuja tiedostoja!
 - `rm tied`
 - poistaa tiedoston "tied"
 - `rm -i tied`
 - kysyy varmistuksen ennen poistoa
 - `rm -r hakem`
 - poistaa hakemiston "hakem" kaikkine tiedostoineen ja alihakemistoineen. EI VÄLTTÄMÄTTÄ KYSY MITÄÄN VARMISTUKSIA!
 - `rm -ir hakem`
 - edelliset yhdessä
 - `rmdir hakem`
 - poistaa tyhjän hakemiston "hakem"

cp

- kopioi tiedostoja ja hakemistoja paikasta toiseen
 - cp file1 file2
 - kopioi file1-tiedoston tiedostoksi file2
 - cp file* hakemisto
 - kopioi file-alkuiset tiedostot hakemistoon. Aina, kun kopioidaan useita tiedostoja, kohteen on oltava hakemisto
 - cp -r hak /tmp
 - kopioi koko hakemiston hak alihakemistoinen hakemistoon /tmp
 - cp /tmp/foo* .
 - kopioi /tmp-hakemistosta kaikki foo-alkuiset tiedostot työhakemistoon
 - huomaa piste!

mv

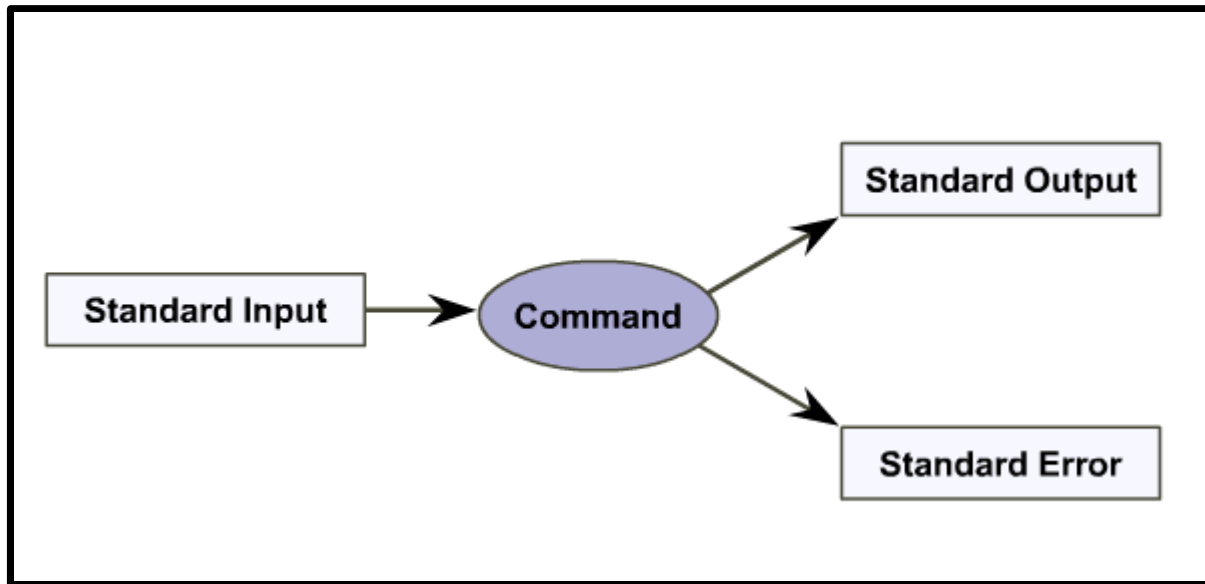
- siirtää (taikka uudelleen nimeää) tiedostoja
 - mv vanha uusi
 - nimeää tiedoston nimeltä "vanha" nimelle "uusi"
 - mv tied* /tmp
 - siirtää tied-alkuiset tdstot /tmp -hakemistoon
 - mv -i vanha uusi
 - kysyy varmistuksen
 - mv hakem uusiahakem
 - osaa siirtää myös hakemistoja
- myös hakemistosta toiseen
 - mv vanha /tmp/uusi

In

- luodaan vaihtoehtoinen nimi (linkki) tiedostoon.
 - voidaan siis viitata kahdella eri nimellä samaan tiedostoon
- kaksi tapaa: ns. kova (hard) ja symbolinen linkki
 - ln nimi uusinimi
 - luo uusinimi-nimisen linkin nimi-nimiseen tiedostoon.
 - ln -s nimi uusinimi
 - sama symbolisella linkillä
- ja mitä eroa:
 - osiolta toiselle voi tehdä vain symbolisia linkkejä
 - periaatteessa samalla osiolla oleviin tiedostoihin pitäisi tehdä kovia linkkejä, mutta voi myös käyttää sym. linkkiä.
 - jos alkuperäisen tiedoston poistaa, kovan linkki jää jäljelle kopiona, sym. linkki joutuu tilaan "broken link"

syötön ja tulostuksen ohjaus

- Unixissa kolme ns. standardivirtaa:
 - standard input, standard output, standard error
 - esimerkki: jos tulostusta ei erikseen ohjata muualle, se ohjataan näytölle (esim. cat tiedosto -komento tulostaa tiedoston sisällön näytölle)
 - samoin muiden standardivirtojen kanssa: input == näppäimistön, output ja error == näyttö.



syötteen lukeminen muualta

- < -operaattorilla:
 - Mail kuivanen@stadia.fi < viesti
 - lähettää tiedostossa "viesti" olevan tekstin sähköpostilla.
- "poistunut" käytöstä monen komennon kohdalla:
 - sort < tiedosto
 - sort tiedosto

tiedostoon ohjaaminen

- > - kirjoittaa tiedostoon
 - jos tiedosto olemassa, kirjoitetaan vanhan yli
 - `ls > list.txt`
 - tiedostolistaus list.txt -tiedostoon
 - `set -o noclobber`
 - estää ylikirjoituksen korn- ja bash-komentotulkeissa, ei mitään muuta suojaa!
 - `ls -l >| list.txt`
 - eston kiertämistapa. Huomaa putkimerkki.
 - `set +o noclobber`
 - sallii ylikirjoittamisen
- >> - kirjoittaa olemassa olevan tiedoston perään
 - jos tiedostoa ei ole olemassa, se luodaan.

virheiden käsittely

- Unixissa virhevirta on oma erillinen tulostusvirtansa, siksi ohjaukset on tehtävä sille erikseen, jos on tarvetta:

- `ls /dir 2> fiba`

- ohjaa syntyneen virheilmoituksen tiedostoon "fiba".
- mikä onkaan virhe kyseisessä tilanteessa?

- `find / -name 'foo' > tulos 2> /dev/null`

- etsii tiedostoa foo alkaen juuresta. Kirjoittaa löydöt tulos-tiedostoon ja heittää kaikki virheilmoitukset "roskakoriin"
- muita unixin erikoistiedostoja: /dev/zero, /dev/random,...

standardivirtojen numerot

ovat siis:

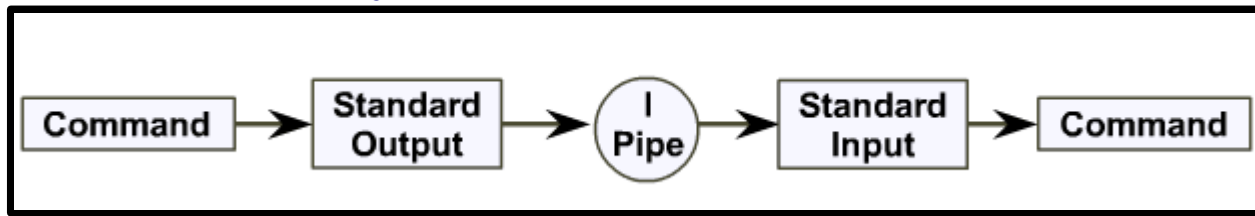
0: standard input

1: standard output

2: standard error

putket (pipe)

- putket edustavat unixin työkaluajattelua parhaimmillaan:
 - yksi komento tekee jotakin ja tämä komennon tuloste siirretään suoraan toiselle komennolle työstettäväksi.



- `ls -l | less`
- `ls -l /etc > temp` # sama, jos putkia ei olisi
`less temp` # tarvii siis kolme komentoa!
`rm temp`
- `ps -ef | grep user2 | sort -k 2`
 - listaa kaikki prosessit, hakee niistä käyttäjän "user2" prosessit ja lajittelee ne toisen kentän mukaan

tee

- komento, jolla voi "haarauttaa" tulostuksen sekä tiedostoon että standard outputiin
- käytetään yleensä juuri putkien kanssa
 - `ls -l | tee list.txt`
 - tulostaa pitkän listauksen näytölle sekä kirjoittaa tiedostoon list.txt

Hakemistot ja tiedostot

tiedostojärjestelmä

Tiedostojärjestelmä (file system)

- Oikeastaan kaksi merkitystä:
 - Hakemistojärjestelmä
 - "directory tree"
 - tiedostot + hakemistot
 - "käyttäjän näkökulma"
 - Levyn tallennusjärjestelmä
 - osiot
 - levykirjanpito
 - "käyttöjärjestelmän näkökulma"

Osioinnin etuja ja ominaisuuksia

- jakaa kiintolevyn pienempiin osiin
- datan organisointi
- pitää liittää järjestelmään ennen käyttöä
 - mount
- voidaan liittää joko luettavaksi tai luettavaksi ja kirjoitettavaksi
- helpottaa varmistamista
 - esim. voidaan varmistaa koko osio kerrallaan
- pienentää hakuaikoja
- tukee turvallisuusoptioita
- parantaa luotettavuutta
- säästää käyttäjän datan järjestelmän päivityksissä

tavanomaisia linux-osioita

- /
 - juuri (root). Tähän liittyvät muut hakemistot/osiot
- /boot
 - kernel & käynnistyksessä tarvittavat tiedostot
- /home
 - käyttäjien kotihakemistot
- /media
 - ulkopuoliset levyt (korppu + cd) liitetään tänne
- /opt
 - kolmannen osapuolen (3rd party) ohjelmat, esim. OpenOffice
- swap
 - virtuaalimuisti. Ei liitetä erikseen mihinkään hakemistoon
- /tmp
 - tilapäistiedostot
- /usr
 - Linux-ohjelmat ja data
- /var
 - muuttuva järjestelmädata (posti, lokitiedostot, printterijonot)

Miten osioin oikein?

- ei ole yhtä oikeaa tapaa osioida
- riippuu eri asioista
 - tietokoneen roolista (palvelin vaiko työasema)
 - olemassaolevasta levymäärästä
- palvelin
 - tarkka osiointi hyödyllistä
 - parantaa luotettavuutta
- työasema
 - usein riittää juuriosio (/) ja kotihakemistot (/home) sekä virtuaalimuisti
- paljonko tilaa kullekin osiolle?
 - tähän ei ole yksikäsitteistä vastausta. Jokin LVM voikin olla käyttökelpoinen apuväline

Levytekniikoista, EIDE

- Levyjen toiminnot keskusyksikköriippuvaisia
 - kaksi laitetta per liitäntä:
 - master
 - slave
- EIDE
 - normaalisti neljä laitetta kahteen linjaan (2 + 2)
 - suurempi tiedonsiirtokapasiteetti kuin IDE:ssä
 - siirtymässä historiaan vähän kerrassaan...

SCSI

- laiteriippumaton
- useampia laitteita
- nopeampi
- standardiliitäntä kaikille laitteille
 - tarvitaan vain yksi adapteri
 - luku ja kirjoitus samanaikaisesti usealle levyille (vrt. IDE, jossa vain toiselle väylän levyille kerrallaan)
- SCSI-3
 - 40 Mbps siirtonopeus
 - max 15 laitetta samaan väylään
 - valittava id-numero
 - mitä suurempi, sitä suurempi prioriteetti
 - Lisätietoa: <http://www.t10.org/>

Linux-osiot

- EIDE
- /dev/hda1
 - hd = EIDE-levy
 - a = ensimmäisen liittimen ensimmäinen levy
 - b = toinen
 - c ja d vastaavat toisessa liittimessä
 - 1 = ensimmäinen primääriosio
 - 1 - 4 primääriosioita
 - 5 - jatko-osioita (extended)
 - levyllä ei ole pakko olla primääriosiota!
 - Tilanne nykyisin: kaikki levyt näytetään kuten SCSI-levyt, /dev/sda, /dev/sdb,...

...Linux-osiot

- SCSI
- /dev/sdb2
 - sd = scsi-levy
 - b = toinen laite ohjaimessa
 - 2 = toinen osio tällä levyllä
- nykyiset kaikki kiintolevyt tyypistä riippumatta, kuten SATA-kiintolevyt "näkyvät" samoin kuin scsi-levyt.
- kuten myös muistitikut

Kommentti Solariksesta:
Solariksen osiointi ja tapa esittää levyt poikkeaa huomattavasti Linuxista. Ks. CNA:n Fundamentals of Unix.

df

- df = disk free
- näyttää vapaan levytilan
- jotkut unix-versiot näyttävät 512:n tavun blokkeina (varausyksikön koko aikoinaan)
- linux näyttää suoraan kiloina:

```
[kuivanen@serveriu203 kuivanen]$ df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/hdc7            497829        81748   390379   18% /
/dev/hdc1            15522         2476    12245   17% /boot
/dev/hdc6           3028080       1791044  1083216  63% /home
/dev/hdc9           5803236       1415736  4092712  26% /opt
/dev/hdc5           10080488      1180328  8388092  13% /usr
[kuivanen@serveriu203 kuivanen]$
[kuivanen@serveriu203 kuivanen]$
```

find

- ohjelma, joka hakee tiedostoja levyiltä
 - `find path expression [action]`
 - `path`: mistä hakemistosta aloitetaan
 - `expression`: mitä haetaan
 - `action`: mitä tehdään hakutulokselle?

...find...

- expression: mitä haetaan:

- -name nimi - hakee tiedostoa nimeltään 'nimi'
- -type tyyppi - hakee tietyntyyppisiä tiedostoja, esim d = hakemisto
- -mtime [+|-]n - montako päivää sitten muutettu. + on n päivää vanhempi, - taas n päivää tai uudempi
- -atime [+|-]n - milloin viimeksi käytetty (kuten edellä)
- -perm mode - hakee tiedostot, joiden oikeudet ovat muotoa 'mode'
 - tiedosto-oikeuksista tulee asiaa myöhemmin...
- -user userid - käyttäjän 'userid' tiedostoja
- -group grpuid - ryhmään 'grpuid' kuuluvia tiedostoja
- -size [+|-]n[c] - tiedoston koko joko suurempi (+) kuin n blokkia (512 t) tai pienempiä (-). C perässä vaihtaa yksiköksi tavuja

...find

- Action - toiminta

- `-exec command {} \;`
 - suorittaa komennon 'command'. Pitää päättää `{}` `\;`, jos halutaan vaikuttavan jokaiseen löydettyyn tiedostoon
 - huomaa muoto: välilyönti sulkujen ja kenoviivan väliin!
- `-ok command {} \;`
 - tarvitaan silloin, jos käytetään komentoa, joka kysyy jotain käyttäjältä, esim. `rm -i`
- `-ls`
 - tulostaa löydetyn tiedoston tiedot pitkässä muodossa
- `-print`
 - tulostaa löydöt std outputiin. Tätä ei enää tarvita välttämättä Linuxissa eikä Solariksessa, mutta joku Unix-versio saattaa tarvita tämän.

find, esimerkkejä:

- `find /usr -name 'file'`
 - etsii /usr -hakemistosta alkaen tiedostot, joiden nimi on 'file'
- `find ~ -name '*gif'`
 - etsii käyttäjän kotihakemistosta alkaen gif-päätteiset tiedostot
 - huomaa heittomerkit: Estetään komentotulkin puuttuminen jokerimerkin käsittelyyn
- `find ~ -name 'core' -exec rm {} \;`
 - etsii käyttäjän kotihakemistosta alkaen tiedostot nimeltään core ja poistaa ne samalla
- `find . -type d`
 - haetaan kaikki hakemistot työhakemiston alta

grep

- Hakee tekstitiedostoista rivejä, jotka täyttävät annetut ehdot
- grep = Global Regular Expression Print
 - grep 'user2' /etc/passwd
 - hakee /etc/passwd-tiedostosta rivit, joilta löytyy teksti 'user2'
 - grep 'teksti' *
 - hakee tekstiä 'teksti' kaikista hakemistossa olevista tiedostoista (milloin on mielekästä?)
 - grep 'teksti' ./*
 - sama kuin edellä, mutta korostetaan työhakemistoa

Säännölliset lausekkeet

- eräänlaisia korvausmerkintöjä hakulauseissa
- älä sekoita jokerimerkkeihin, vaikka samat merkit näissäkin esiintyvät!
- jaettu yleensä kahteen osaan: peruslausekkeisiin ja laajennettuihin
- pääsääntöisesti Linux-grep:t (GNU) eivät tee eroa näiden välillä, muut versiot unixista tekevät
- useat eri ohjelmat osaavat käyttää näitä merkintöjä, tosin kaikki ohjelmat eivät osaa kaikkia näistä

Säännöllisissä lausekkeissa käytetyt merkit

- merkki taikka merkkijono:
 - haku sellaisenaan: `grep 'a' foo`, hakee tiedostosta `foo` sellaiset rivit, joilla on merkki `a`
- `.`
 - yksi mikä tahansa merkki
 - `grep 'kis.a' foo => kissa, kiska, kista, kisia, ...`
- `*`
 - edeltävä merkki 0 – n kertaa:
 - `grep 'kis*a' foo => kia, kisa, kissa, ksssssa,....`
- `^` - sitoo rivin alkuun
- `$` - rivin loppuun

...merkinnät

- [lista]
 - luettele merkit, jotka sopivat
 - `grep 'kiss[abcde]' foo` (myös `'kiss[a-e]'`)
- tästä eteenpäin sellaisia, joita eivät kaikki ohjelmat välttämättä hallitse:
- ?
 - mikä tahansa merkki 0 tai 1 kertaa
 - `kiss?a => kissa tai kisa`
- +
 - 1 – n kertaa. Huomaa ero tähteen.

...merkinnät

- $\{N\}$
 - edeltävä täsmälleen N kertaa
- $\{N,\}$
 - vähintään N kertaa
- $\{N, M\}$
 - N – M kertaa
- $|$
 - joko tai
- $()$
 - ryhmittely
 - $\text{Matti}(|\text{a|nen}) \Rightarrow$ joko Mattila tai Mattinen

...merkinnät

- \
 - sitoo haun sanan alkuun
- \>
 - sanan loppuun
- ohjelmia
 - grep – perus-grep
 - egrep – extended grep
 - fgrep – fast (fixed) grep. Ei huomioi ollenkaan säänn. lausekkeita.
- lisää tietoa: info:grep
 - konqueror-selain tarjoaa järkevämmän liittymän info-komentoon. Kannattaa käyttää sitä... ;-)

sed

- lukee tekstitiedostoa rivi kerrallaan
- suorittaa kullekin riville halutut toimenpiteet
- ei koske alkuperäiseen tiedostoon!
 - ei siis tee mitään muutoksia siihen ==> turvallinen!
 - muutokset pitää ohjata tiedostoon itse
- jos tulosta ei ohjata minnekään, tulostaa std outputiin
 - ==> pitää siis käyttää > - tai >> -ohjauksia, jos halutaan muutoksista pysyviä.

sed, esimerkkejä

- sed -n '10,30p' tied
 - näyttää tiedostosta rivit 10-30
- sed '3d' tied
 - poistaa rivin 3
- ls -l | sed '/[Tt]esti/d' > tied
 - poistaa ls -l:n tulosteesta tiedostot testi ja Testi ja kirjoittaa tuloksen tiedostoon 'tied'
- sed 's/vanha/uusi/g' tied > tied2
 - korvaa tiedostosta 'tied' jokaisen (g = global) sanan "vanha" sanalla "uusi" ja kirjoittaa tuloksen tiedostoon 'tied2'
- ls -l | sed 's/ */:/g'
 - korvaa ls -l:n tulosteesta kaikki vähintään yhden välilyönnin pituiset tyhjät yhdellä kaksoispisteellä. Huomaa kaksi välilyöntiä!
 - ks info:sed, niin saat selville, miten sed hallitsee säännölliset lausekkeet.

sort

- lajittelee tiedostoja
- monipuolinen:
 - runsaasti optioita
 - minkä sarakkeen perusteella tahansa
- optioita:
 - **-n**
 - numeerinen lajittelu. Tarvitaan, jos järjestetään lukuja
 - **+|-luku**
 - monesko sarake alusta (+) vaiko lopusta (-). Ensimmäinen on 0. sarake. Sarake-erottimeksi käy välilyönti tai tabulaattori (oletuksina)
 - **-k**
 - vaihtoehto edelliselle
 - oikeastaan tätä pitäisi käyttää POSIX-sääntöjen mukaisesti
 - **-f**
 - isot ja pienet kirjaimet samanarvoisia
 - **+nM**
 - lajitellaan n:s sarake kuukauden kolmemerkkisen lyhenteen mukaisessa järjestyksessä.
 - **-r**
 - käännetty järjestys
 - **-d**
 - sanakirjajärjestys (vain kirjaimet, numerot ja välilyönnit otetaan huomioon)
 - **-o**
 - mihin tiedostoon lajitellaan
 - **-b**
 - jos sarakkeiden välissä eri määrä välilyöntejä, käsittelee näitä kuin yhtä välilyöntiä

sort, esimerkkejä

- `sort eka > toka` (tai: `sort eka -o toka`)
 - lajittelee tiedoston eka tiedostoon toka
- `sort -k 3n -b luvut`
 - lajittelee tiedoston ”luvut” kolmannen sarakkeen mukaan. -b poistaa ylimääräiset tyhjät lajittelussa.
- `ls -l | sort +5M +6n`
 - lajittelee ls -l:n listauksen kuukauden mukaisessa järjestyksessä (jan, feb, mar, apr, ...) Huomaa kaksi ehtoa (kk + pvm)
- `sort +4 -t: /etc/passwd > toinen`
 - -t -option jälkeen voi kertoa kenttien erotinmerkin, jos se on jotain muuta kuin välilyönti tai tabulaattori