file

file [*optiot*] *tiedosto[t]*

- Yrittää tunnistaa tiedoston tyypin (tiedostonimestä 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

regular expressions

- Regular expression: yleinen merkkijonomalli, jossa erikoismerkeillä esitetään vaihtoehtoja
- Glob pattern: yksinkertaistettu regexp, shellin yms jokerimerkkinotaatio:

  ? yksi mielivaltainen merkki

  * mielivaltainen määrä mitä tahansa merkkejä

  [acx1-9] yksi kirjain a, c tai x tai numero 1-9
- Basic regular expression:

  . (piste) yksi mielivaltainen merkki

  * mielivaltainen määrä edeltäviä merkkejä

  \{*m,n*\} edellistä merkkiä toistetaan m-n kertaa

  ^ rivin alku

  $ rivin loppu

  [...] kuten yllä

  \(...\) rajoittaa osan lausekkeesta viitattavaksi myöhemmin

  n *viittaa n*:nteen \(...\) -osalausekkeeseen

  \ suojaa erikoismerkin, esim. \. = piste

grep

(”g/re/p”: get regexp and print)

- grep [options] {[-e] 'malli'|-f mallifile} [file...]

  tulostaa mallia vastaavat rivit
- Esim.

  grep hauki kalat.txt

  grep '[0-9][0-9]*\.[0-9][0-9]*' file.lst # kaksi lukua joiden välissä piste

  grep '^\([0-9][0-9]*\)[^0-9].*\1$' file1 > file2 # rivin alussa ja lopussa sama luku

  grep -e kala -e lintu file # sekä kala- että linturivit
- Optioita mm:

  -i ignore case (isot ja pienet kirjaimet samanarvoisia)

  -v tulosta rivit joilla mallia *ei* esiinny

  -E extended regepx (egrep)

  -F fixed strings (fgrep): malli on vakiomerkkijono (* jne esittävät vain itseään)

extended regular expressions

- egrep = grep -E, käyttää extended regexp'ejä:
  - + edeltävän merkin toisto vähintään kerran
  - ? edeltävän merkin toisto nolla tai useamman kerran
  - () rajaavat osalausekkeen ilman kenoviivoja
  - {*m,n*} edeltävän merkin toisto m-n kertaa (ei kenoviivoja)
  - ^ ja $ toimivat myös osalausekkeissa
  - | ”tai” (myös osalausekkeen sisällä) – yleisin syy käyttää egrepiä
  - osalausekeviittauksia \1 jne ei ole (standardiversiossa)
- Usein lyhyempi kuin grep, esim.

  ```
  egrep '[-+]?[0-9]+'          # vrt. grep '[-+]\{0,1\}[0-9][0-9]*'
  ```
- Seuraavat eivät onnistu basic regexpillä:

  ```
  egrep '((kissa|koira)[ ;,])+'  # paljon kissoja tai koiria
  egrep 'iso (lintu|kala) (lensi|ui)'
  ```
- Erityisesti Gnu grep tuntee kaikenlaisia laajennuksia...

sed

- ”stream editor”: komentorivieditori, sopii käytettäväksi skripteissä yksinkertaisiin tiedoston muutoksiin
- Käsittelee tiedostoa rivi kerrallaan (monen rivin muutoksia voi tehdä mutta vaikeasti)
- Nopea ja tehokas, tarvitsee hyvin vähän muistia
- Syntaksi:

  sed [optiot] {-f scriptfile|[-e] 'script'...} [file...] [>outfile]

- Oletuksena kirjoittaa stdout'iin, optiolla -n ei tulosta muuta kuin erillisillä tulostuskomennoilla, optiolla -i muuttaa tiedostoa paikalla (-i.bak tallettaa alkuperäisen tarkenteella .bak)
- Editointikomennot voi antaa komentorivillä (optio -e, ei tarvita jos vain yksi) tai tiedostossa (-f)

sed 2

- Komentojen yleinen syntaksi:

  <osoite>[,<osoite2>]komento[argumentit]

  - Osoitteet rivinumeroita tai malleja (/regexp/), $=loppu, !=ei
  - Komennot yksikirjaimisia, seuraavassa yleisimmät
- s: substitute, korvaa merkkijono toisella

  s/malli/korvaus/[liput]

  - Oletuksena vain ensimmäinen per rivi, lippu g = kaikki, numero = korvaa n:s esiintymä
  - p = tulosta jos korvattiin, w *file* = kirjoita tiedostoon jos korvattiin
  - Malli basic regular expression (vähemmän laajennuksia kuin grepissä)
  - Korvausmerkkijonossa & = korvattava jono, *n* = *n*:s \(...\)

  ```
  sed '5,6s/lintu/kala/g'          # korvaa kaikki linnut kaloilla riveillä 5-6
  sed 's/./& /g'                   # lisää jokaisen merkin perään välilyönti
  sed '/kala/s/^\(.\)\(.\)/\2\1/'  # vaihda kaksi ensimmäistä merkkiä kalariveillä
  ```

sed 3

- d: delete
  - Ei optioita eikä argumentteja, mutta yleensä aina osoite, esim.

    ```
    sed 5d              # poista rivi 5
    sed /kala/d         # poista rivit joilla esiintyy kala
    sed '1,5d;9,15d'  # poista rivit 1-5 ja 9-15
    sed '/kala/,$d'     # poista rivit kalan ensimmäisestä esiintymästä loppuun
    sed '/kala/!d'       # poista kaikki rivit joilla ei ole sanaa 'kala'
    ```
- p: print
  - Tulostaa rivin (senhetkisessä muodossaan); ei optioita eikä argumentteja
  - tarpeen erityisesti käytettäessä sed'in -n optiota

    ```
    sed -n 15,20p       # tulosta (vain) rivit 15-20
    sed -n /kala/p       # sama kuin sed '/kala/!d'
    sed -n 1,/kala/p   # tulosta alusta ensimmäiselle kalariville
    ```

sed 4

- w: write to file

  sed '/kala/w kalafile.txt' infile >outfile # vrt: grep kala infile > kalafile.txt

  - Tiedostonimen edessä oltavat tasan yksi välilyönti
- r: read from file

  sed '5r file' infile # tiedoston ”file” sisältö lisätään rivin 5 jälkeen
- {}: ryhmittely, esim.

  sed '1,10{/^#/d;}' # poistetaan riveistä 1-10 #:lla alkavat

  sed '1,50{s/kala/hauki/g;s/lintu/kana/g;}'

  # riveiltä 1-50 vaihdetaan kalat hauiksi ja linnut kanoiksi

  sed '1,/kala/{/hauki/s/a/b/g;}'

  # alusta ensimmäiseen kalariviin sanan hauki-sisältäviltä vaihdetaan a->b
- Paljon muitakin komentoja ja optioita, ks.
  http://www.mit.jyu.fi/opiskelu/kurssit/unixshell01/sed.html

ping

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

  ping *kone*

- Optioista on hyötyä lähinnä skripteissä mm.

  -c *count* lähetä vain *count* pakettia

  -q älä tulosta jokaista pakettia

- Tutkittaessa verkko-ongelmaa koneen sisältä kannattaa aloittaa ”ping localhost”illa, sitten pingailla yhä kauemmas verkossa
- Ei aina toimi palomuurien läpi

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, yms
- Jos grubissa on (kuten oletuksena on) HIDDEN_TIMEOUT, boottivalikon saaminen edellyttää bootin keskeyttämistä oikealla hetkellä (joka voi olla hyvinkin lyhyt); palvelimissa HIDDEN_TIMEOUT kannattaa siksi poistaa käytöstä
- Boottivalikossa voi myös muokata menuentryjä tai bootata vaikkapa vanhalta kerneliltä

Recovery mode valikko

- Ubuntun (esimerkkinä 14.04) 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 (esim levyn täytyttyä kesken paketin asennuksen) ja päivittää vanhoja
  - fsck: tiedostojärjestelmän korjausyritys (ei yleensä onnistu edellisten jälkeen)
  - grub: asentaa grub'in uudestaan (grub-install)
  - network: yrittää käynnistää verkon
  - root: shell pelastuksen jatkamiseksi käsin
  - system-summary: yrittää näyttää jotain perustietoja koneesta (ei useinkaan toimi)

Single-user mode

- 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
  - Verkon tilan tarkistus: ifconfig, ping
    - Jos verkko on nurin, sen voi yrittää käynnistää käsin komennolla ifup eth0 (laitenimi voi vaihdella), joskus ensin muokaten /etc/network/interfaces -tiedostoa
  - Levyjen tilan tarkistus: df, vgs, ls /dev/?d?*, fdisk -l /dev/sda 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/...)
  - Tarkista lokit (ainakin /var/log/syslog), dmesg
  - Jos muut editorit eivät toimi, sed voi pelastaa päivän

Single-user mode: levy täynnä

- Jos jokin kriittinen levyosio (etenkin /, /boot, /var, /usr) on täyttynyt ohjelman asennuksen tai päivityksen yhteydessä ja se on saanut koneen niin sekaisin, ettei se enää boottaa normaalisti:
  - Pakettiluettelon tarkistus: dpkg -l
  - Ensin apt-get autoremove, apt-get clean, yritys poistaa (tilapäisesti) paketteja apt-get purge...
  - 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 (ls /boot, dpkg -l |grep linux-image), tilanteen selvittyä pitää siivota sotku (apt-get purge tai dpkg -r rikotuille paketeille, lopuksi apt-get update && apt-get upgrade)

Ftp server

- ”file transfer protocol” - vanha ja pikkuhiljaa katoava, mutta yhä laajalti käytössä
- Ei salausta tiedonsiirrolle eikä edes autentikoinnille, turvallinen vain julkisen tiedon anonyymiin jakeluun (anymous upload myös OK jos sellaista tarvitaan); nyttemmin toimii myös SSL:n kanssa (aika turha koska saman tien voi yleensä käyttää https:ää)
- Käyttää kahta tcp-porttia (20 ja 21), erikseen ohjaus- ja tiedonsiirtoväylät, kaksi toimintatapaa (active & passive – jälkimmäisessä palvelin ottaa datayhteyden takaisinpäin), vaatii erikoissäätöjä palomuureissa
- Ftp-palvelinohjelmia paljon, asiakasohjelmina nykyisin lähinnä web-selaimet (mukaanlukien wget), mm. vsftpd (yleisin Linuxeissa nykyisin), Pure-FTPd, ProFTPD

vsftpd

- Asennus:

  apt-get install vsftpd

  nano /etc/vsftpd.conf

  - Tarkista ainakin rivit
    anonymous_enable
    anon_upload_enable
    anon_mkdir_write_enable
    local_enable
    chroot_local_user
    ls_recurse_enable
    write_enable
    xferlog_enable
    nopriv_user

  – Jos anonymous -käyttö on sallittu, sitä varten pitää tehdä hakemisto (ftp-käyttäjälle)