

Osa IV: Tekst, otsing ja automatiseerimine — täisspikker

Tase: Töövood

Teksti otsimine: grep ja sugulased

Eesmärk: Vali tekstist sobivad read; erista täpset teksti regulaaravaldisest.

Otsingukuju ja tulemus

- `grep 'kass' loomad.txt` — ka pikema sõna sees; väljastab terve rea
- `grep -in 'kass' loomad.txt` — eira tähesuurst, lisa reanumbrid
- `grep -x 'kass' loomad.txt` — ainult tervenisti sobiv rida
- `grep -v '^#' seadistus.conf` — jäta välja # märgiga algavad read
- `grep -F '.json' nimed.txt` — punkt tähendab päris punkti, mitte suvalist märki
- `grep -rn 'TODO' .` — otsi failipuust koos failinimede ja reanumbritega
- `grep -c 'kass' loomad.txt` — loenda vastega ridu, mitte kõiki esinemi

grep-i tulemuskoodid

- 0 — vähemalt üks vaste
- 1 — vasteid pole, mitte tingimata viga
- 2 — tavaliselt viga; loe veateadet

Pane tähele: `-F` otsib täpset teksti ka pikema rea seest. Terve rea sobitamiseks lisa `-x`.

Teksti teisendamine: tr, cut, paste, column, strings

Eesmärk: need väikesed filtrid muudavad tekstivoo kuju kiiresti: märgid, väljad, veerud ja tabelid ilma eraldi skriptita

Põhikujud

- `echo 'tere maailm' | tr '[:lower:]' '[:upper:]'` — muuda suurtäheks
- `echo 'a,b,c' | tr ',' '\n'` — tee ridadeks
- `cut -d ':' -f 1 andmed.txt` — võta esimene väli
- `cut -d ':' -f 1,3 andmed.txt` — võta mitu välja
- `printf 'nimi vanus\nMari 20\n' | column -t` — joonda tabel
- `strings /bin/ls | head -n 20` — loe binaarist tekst
- `paste` — kleebi veerge

Olulisemad lipud, märgid ja kiirnupud

- `[:lower:]` — väiketähed
- `[:upper:]` — suurtähed
- `cut -d` — vali eraldaja
- `cut -f` — vali väljad
- `column -t` — joonda tabel
- `strings` — tekstijupid

Pane tähele: Kui väljund muutub imelikuks, kontrolli kõigepealt eraldajat ja sisendi kuju; need tööriistad on väikesed, aga väga sõna-sõnalt loetavad.

Vood ja tabelid: `sort`, `uniq`, `wc`, `pr`, `join`

Eesmärk: tüüpiline töövoog on: sorteerida read, koondada kordused ja loe tulemused kokku

Põhikujud

- `sort viljad.txt` — sordi read
- `sort viljad.txt | uniq` — eemalda kordused
- `sort viljad.txt | uniq -c` — loe kordused
- `wc -l data/sample-text.txt` — loe read
- `wc -w data/sample-text.txt` — loe sõnad
- `join nimed.txt linnad.txt` — ühenda võtme järgi
- `pr` — jaga veergudeks

Olulisemad lipud, märgid ja kiirnupud

- `sort -n` — numbrid
- `sort -r` — tagurpidi
- `uniq -c` — loenda kordused
- `wc -l` — ridade arv
- `wc -w` — sõnade arv
- `pr -2` — kaks veergu

Pane tähele: `uniq` ühendab ainult kõrvutised võrdsed read. Kui algjärjekord pole oluline, koondada võrdsed read esmalt käsuga `sort`.

`sed`, `awk` ja `perl` praktiliselt

Eesmärk: kasutada `sed` lihtsaks asenduseks, `awk` väljade jaoks ja `perl` siis, kui vajad tugevamat regulaaravaldiste loogikat

Põhikujud

- `echo 'kass koer kass' | sed 's/kass/rebane/'` — asenda esimene
- `echo 'kass koer kass' | sed 's/kass/rebane/g'` — asenda kõik

- `printf 'Mari:20\nJaan:21\n' | awk -F: '{print $1}'` — võta esimene väli
- `printf 'Mari:20\nJaan:21\n' | awk -F: '{print $1, $2}'` — võta kaks välja
- `echo 'kass koer kass' | perl -pe 's/kass/rebane/g'` — asenda Perlga
- `printf 'Mari\nJaan\n' | perl -ne 'print if /Ja/'` — filtreeri Perlga

Olulisemad lipud, märgid ja kiirnupud

- `s/vana/uus/` — üks asendus
- `/g` — kõik vasted
- `-F:` — koolon väljade vahel
- `$1` — esimene väli
- `$2` — teine väli
- `-pe / -ne` — read läbi

Pane tähele: Kui vajad ainult üht lihtsat asendust, alusta sed-ist; ära hüppa kohe `awk` või `perl` juurde enne, kui lihtsam tee on ammendunud.

`find` ja `xargs` ohutumalt

Eesmärk: otsi failid `find`-iga ja anna need ohutult edasi järgmisele käsule, isegi siis, kui nimedes on tühikuid

Põhikujud

- `find . -type f -name '*.txt'` — otsi failid
- `find . -type f -name '*.txt' -print0 | xargs -0 wc -l` — loe read ohutult
- `find . -type f -name '*.log' -exec ls -lh {} +` — käivita otse
- `find data -type f -name '*.txt' -print0 | while IFS= read -r -d '' fail; do head -n 1 "$fail"; done` — töötle ükshaaval

Olulisemad lipud, märgid ja kiirnupud

- `-type f` — `find`: ainult tavafailid
- `-name '*.txt'` — nime järgi
- `-print0` — `find`: failinimede vahel nullmärk
- `xargs -0` — `xargs`: eralda argumendid nullmärgiga
- `-exec käsk {} +` — käivita otse
- `read -r -d ''` — Bash/zsh: loe nullmärgini, säilita kaldkriipsud

Pane tähele: Kui failinimes võib olla tühikuid või reavahetusi, eelista paari `-print0` ja `xargs -0`.

Esimene shelliskript

Eesmärk: esimese shelliskripti tuum on viis asja: shebang, käivitavus, argumentid, lihtne tingimus ja lõpetuskood

Põhikujud

- `chmod +x tervita.sh` — tee käivitavaks
- `./tervita.sh Mari Jaan` — käivita argumentidega
- `./tervita.sh` — näe veajuhtumit
- `echo $?` — vaata lõpetuskoodi
- `bash skript.sh` — käivita Bashiga
- `sh skript.sh` — käivita sh-ga
- `./skript.sh` — kasuta shebang-rida

Olulisemad lipud, märgid ja kiirnupud

- `#!/usr/bin/env bash` — vali Bash
- `$1` — esimene argument
- `"$@"` — kõik argumentid
- `if ... fi` — tingimus
- `for ... done` — kordus
- `exit 0 / exit 1` — edu või viga

Pane tähele: Kõige tähtsam vahe on see: `./skript.sh` kasutab shebang-rida, `bash skript.sh` valib tõlgendaja käsurealt.

cron ja ajastatud tööd

Eesmärk: Ajasta läbiproovitud käsk; kasuta täisteid ja talleta väljund logisse.

Käsud ja näited

- `crontab -l` — vaata praeguse kasutaja ajastusi
- `crontab -e` — muuda tabelit, säilitades muud tööd
- `/bin/date >> "$HOME"/cron-naide.log 2>&1` — testi ajastatavat käsku käsitsi
- `tail -n 5 "$HOME"/cron-naide.log` — kontrolli, kas töö lisab kirjeid

Märgid ja valikud

- `*/15 * * * *` — minutitel 0, 15, 30 ja 45
- `0 8 * * 1-5` — tööpäeviti kell 08.00 süsteemi ajavööndis
- `crontab fail` — asendab kogu kasutaja ajastustabeli
- `crontab -r` — eemaldab kogu kasutaja ajastustabeli

Pane tähele: Ära kasuta `crontab fail` ühe rea lisamiseks. Cron ei käivita tavaliselt arvuti uneajal vahele jäänud töid tagantjärele.