

Osa IV: Tekst, otsing ja automatiseerimine — täisspikker

Tase: Töövood

Teksti otsimine: grep ja sugulased

Peatüki täisspikker

Tase: Töövood

Eesmärk: grep valib sisendist välja ainult need read, mis sobivad mustriga; see on filtritööriist, mitte tekstiredaktor

Põhikujud

- `grep 'kass' loomad.txt` — otsi ühest failist
- `grep -i 'kass' loomad.txt` — otsi tõstuta
- `grep -n 'koer' loomad.txt` — näita reanr
- `grep -r 'TODO' .` — otsi puust
- `grep -F 'https://example.com' fail.txt` — otsi täpset sõnet
- `grep -v '^#' seadistus.conf` — jäta kommentaarid välja

Olulisemad lipud, märgid ja kiirnopud

- `-i` — tõstutundetud
- `-n` — reanumbrid
- `-r` — rekursiivne
- `-v` — jäta vasted välja
- `-F` — täpne sõne
- `-E` — laiendatud regex

Pane tähele: Kui sa ei ole mustris kindel, alusta `grep -F` või lihtsa sõnega; regulaaravaldise keerukus olgu järgmine samm, mitte esimene.

Edasi: Järgmine loomulik samm: Teksti teisendamine: `tr`, `cut`, `paste`, `column`, `strings`.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

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

Peatüki täisspikker

Tase: Töövood

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 kiirnopud

- `[: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.

Edasi: Järgmine loomulik samm: Vood ja tabelid: `sort`, `uniq`, `wc`, `pr`, `join`.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

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

Peatüki täisspikker

Tase: **Töövood**

Eesmärk: tüüpiline töövoog on: sorteeri read, koonda 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 kiirnopud

- `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` töötab loogiliselt alles pärast `sort-i`; vastasel juhul loendab ta ainult järjestikuseid kordusi.

Edasi: Järgmine loomulik samm: `sed`, `awk` ja `perl` praktiliselt.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

sed, awk ja perl praktiliselt

Peatüki täisspikker

Tase: **Töövood**

Eesmärk: kasuta `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 Perliga
- `printf 'Mari\nJaan\n' | perl -ne 'print if /Ja/'` — filtreeri Perliga

Olulisemad lipud, märgid ja kiirnopud

- `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.

Edasi: Järgmine loomulik samm: `find` ja `xargs` ohutumalt.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

find ja xargs ohutumalt

Peatüki täisspikker

Tase: **Töövood**

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 kiirpud

- `-type f` — ainult failid
- `-name '*.txt'` — nime järgi
- `-print0` — ohutu eraldus
- `xargs -0` — loe ohutult
- `-exec {} +` — käivita otse
- `read -r -d ' ' —` loe nullmärgini

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

Edasi: Järgmine loomulik samm: Esimene shelliskript.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

Esimene shelliskript

Peatüki täisspikker

Tase: **Töövood**

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

Põhikujud

- `chmod +x tervita.sh —` tee käivitatavaks
- `./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'i

Olulisemad lipud, märgid ja kiirnopud

- `#!/usr/bin/env bash` — vali Bash
- `$1` — esimene argument
- `"$@"` — kõik argumendid
- `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.

Edasi: Järgmine loomulik samm: cron ja ajastatud tööd.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)

cron ja ajastatud tööd

Peatüki täisspikker

Tase: **Töövood**

Eesmärk: ajastatud töö tähendab, et käsk jookseb kindlal ajal ilma sinu avatud terminalita; tähtsad on ajaväljad, täisrajad ja logifail

Põhikujud

- `crontab -l` — vaata olemasolevat
- `crontab naide.cron` — paigalda fail
- `/bin/date >> "$HOME"/cron-naide.log 2>&1` — testi käsitsi
- `tail -n 5 "$HOME"/cron-naide.log` — vaata logi

Olulisemad lipud, märgid ja kiirnopud

- `*/15 * * * *` — iga 15 min
- `>>` — lisa logisse
- `2>&1` — vead samasse
- `$HOME` — kasutaja kodu
- `/bin/date` — täisrada

Pane tähele: Kui käsk ei tööta täpselt samal kujul käsitsi, ei tööta see tavaliselt ka cronis.

Edasi: Järgmine loomulik samm: Git, GitHub ja töövoog.

Osa PDF: [./osa-iv-tekst-otsing-ja-automatiseerimine-spikker.pdf](#)