

# Kõik osade spikrid

Tegemist on eraldi referentsiväljundiga, mis koondab peatükkide lõpus olevad täisspikrilaadsed kokkuvõtted osade kaupa ühte faili. Buildi aeg: **2026-04-30 22:10:34 EEST**.

## Osa I: Esimesed sammud — täisspikker

Tase: Algaja

Kuidas seda õpikut kasutada

### Peatüki täisspikker

Tase: Algaja

**Eesmärk:** Saa aru, mis on terminal, shell, CLI ja GUI, ning vali rahulik algusrada enne keerulisemaid töövooge.

#### Põhitee

- Terminali esimesed sammud — alusta siit
- Abi leidmine — otsi tuge
- Kataloogid ja failid — tee muudatusi
- Teksti vaatamine — loe rahulikult

#### Tüüpilised kujud

- `pwd` — kontrolli asukohta
- `ls` — vaata sisu
- `man ls` — loe abi
- `cd ~/tmp` — mine tmp-kausta; sobib harjutamiseks

#### Põhimõisted

- CLI — käsurida
- GUI — graafiline liides
- `terminal` — aken käsureaks
- `shell` — tõlgendab käske

**Pane tähele:** Sa ei pea esimeses ringis veel teadma Git-i, Dockerit ega keerulisemaid töövooge.

**Edasi:** Järgmine loomulik samm: Õpitee ja õppetunnid.

**Osa PDF:** `./osa-i-esimesed-sammud-spikker.pdf`

## Õpitee ja õppetunnid

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** See peatükk aitab valida, mis järjekorras lugeda. Raamatut saab kasutada ka käsiraamatuna, kuid alguses on lihtsam liikuda kindla õpitee järgi.

### Õpiteed

- Täiesti algaja — loe esimesena
- Süsteemi pildi loomine — pärast esimest ringi
- Igapäevane Linuxi ja serveri kasutaja — kaugühendus ja failid
- Tekst, filtrid ja automatiseerimine — käsurea jõud
- Arendaja suund — Git, Docker, build
- Andmeteaduse stardirada — failid, vormingud, SQL

### Soovitatud järjestused

- Terminali esimesed sammud → Abi leidmine → Kataloogid ja failid → Teksti vaatamine ja liikumine — algaja põhirada
- Failide muutmine → Käskude kuju → Torud ja suunamine → Esimene tervikharjutus — seo alus tervikuks
- Failisüsteemi kaart → Kettaruum → Õigused → Võrgu põhitööriistad — ehita süsteemi pilt
- grep → Teksti teisendamine → sort, uniq, wc → Kompileerimine ja käivitamine — filtrid ja arendus

### Märgendid ja kontroll

- vajalik — loe nüüd
- soovitatav — hea järgmine
- hiljem — ära kiirusta
- Pärast seda rada oskad ... — kontrolli edenemist

**Pane tähele:** Ära püüa kogu raamatut korraga läbi võtta; vali kõigepealt üks rada ja liigu selle sees rahulikult edasi.

**Edasi:** Järgmine loomulik samm: Terminali esimesed sammud.

**Osa PDF:** [./osa-i-esimesed-sammud-spikker.pdf](#)

### Terminali esimesed sammud

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** Vaata kõigepealt rahulikult, kus sa oled, kes sa oled ja mis selles kaustas on; alles siis tee esimene väike muudatus.

## Käsud ja esimesed sammud

- `pwd` — vaata asukohta
- `ls` — vaata sisu
- `whoami` — vaata kasutajat
- `date` — vaata praegust kuupäeva ja aega
- `history` — vaata käsuajalugu
- `cd ~/tmp` — mine tmp-kausta; sobib harjutamiseks
- `mkdir proov && cd proov` — loo proovikaust

## Olulisemad lipud, märgid ja kiirnopud

- `Tab` — lõpeta nimi
- `Ctrl-c` — katkesta programmi töö
- `Ctrl-r` — otsi ajaloost
- `Ctrl-a` — rea algus
- `Ctrl-e` — rea lõpp
- `~/proov %` — näide viibast ehk promptist

**Pane tähele:** Ära kopeeri viipa käsu ette kaasa; kopeeri ainult käsk ise.

**Edasi:** Järgmine loomulik samm: Abi leidmine: `man`, `-help` ja `info`.

**Osa PDF:** `./osa-i-esimesed-sammud-spikker.pdf`

**Abi leidmine:** `man`, `-help` ja `info`

## Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** Kui sa ei mäleta käsu kuju või lippu, vaata kõigepealt käsu enda abi: `man`, `-help`, `whatis`, `apropos` ja vajadusel `info`.

## Põhikujud

- `man ls` — loe põhijuhendit
- `grep --help` — kiire lippude abi
- `whatis ls` — üks lause
- `apropos ssh` — otsi teema järgi
- `info coreutils 'ls invocation'` — GNU sügavam abi

## Olulisemad lipud, märgid ja kiirnopud

- `q` — välju `man-ist`
- `/tekst` — otsi `man-ist`
- `n` — järgmine vaste
- `g` — algusesse
- `G` — lõppu

**Pane tähele:** Kui man tundub “terminali üle võtvat”, siis vajuta lihtsalt q; see on normaalne täisekraani vaade, mitte ummik.

**Edasi:** Järgmine loomulik samm: Kataloogid ja failid.

**Osa PDF:** ./osa-i-esimesed-sammud-spikker.pdf

## Kataloogid ja failid

### Peatüki täisspikker

Tase: Algaja

**Eesmärk:** Failitöö põhiharjumus on: kontrolli asukohta, tee üks väike muudatus, vaata tulemus üle ja kustuta alles pärast teadlikku pausi.

#### Põhikujud

- `pwd` — kontrolli asukohta
- `ls` — vaata sisu
- `mkdir -p ~/tmp/faili-naited` — loo terve puuduv tee
- `cd ~/tmp/faili-naited` — mine harjutama
- `touch fail.txt` — loo fail
- `cp algne.txt koopia.txt` — tee koopia
- `mv vana.txt uus.txt` — nimeta ümber
- `mv fail.txt kaust/` — liiguta kausta
- `rm fail.txt` — kustuta fail
- `rmdir tyhi-kataloog` — eemalda tühi kataloog
- `find . -name '*.txt'` — otsi ohutult
- `shasum -a 256 a.txt b.txt` — võrdle sisu macOS-is

#### Olulisemad lipud, märgid ja kiirrupud

- `.` — see kaust
- `..` — ülemine kaust
- `~` — kodukataloog
- `ls -a` — ka peidetud
- `ls -A` — peidetud ilma `.` ja `..` kirjeteta
- `ls -la` — detailne vaade
- `"fail nimi.txt"` — nimi tühikuga
- `rm -r` — rekursiivne kustutus

**Pane tähele:** Enne `rm` või `rm -r` tee väike paus: kontrolli `pwd`, vaata `ls`, alles siis kustuta; `rm` ei vii faili prügikasti.

**Edasi:** Järgmine loomulik samm: Teksti vaatamine ja liikumine.

**Osa PDF:** ./osa-i-esimesed-sammud-spikker.pdf

## Teksti vaatamine ja liikumine

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** lühikese faili jaoks kasuta `cat`; pika faili sirvimiseks ja otsimiseks kasuta `less`; faili alguse vaatamiseks kasuta `head`

#### Põhikujud

- `less fail.txt` — ava lehitseja
- `head -n 7 fail.txt` — esimesed read
- `tail -n 7 fail.txt` — viimased read
- `tail -f app.log` — jälgi logi
- `less *` — mitu faili korraga
- `cat` — näita faili
- `seq` — numbrijada
- `touch` — loo või aja tempel

#### `less`-i klahvid

- `q` — välju
- `/tekst` — otsi edasi
- `n` — järgmine vaste
- `N` — eelmine vaste
- `v` — ava redaktoris
- `:n / :p` — järgmine/eelmine fail

**Pane tähele:** `cat` sobib lühikese faili jaoks; pika faili puhul vali `less`, `head` või `tail`.

**Edasi:** Järgmine loomulik samm: Failide muutmine: nano ja esimene kokkupuude vim-iga.

**Osa PDF:** [./osa-i-esimesed-sammud-spikker.pdf](#)

## Failide muutmine: nano ja esimene kokkupuude vim-iga

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** `less` aitab faili sisu enne muutmist mugavalt üle vaadata; `nano` on alguses kõige lihtsam tekstiredaktor; `vim` ja `vi` on sama pere klassikalised tekstiredaktorid

#### Põhikujud

- `less fail.txt` — vaata faili sisu mugavalt

- `nano fail.txt` — lihtne tekstiredaktor
- `vim fail.txt` — klassikaline redaktor
- `vi fail.txt` — sama pere variant
- `command -v nano` — kontrolli olemasolu

### vi/vim minispikker

- `Esc` — välju sisestusest
- `:q` — välju
- `:q!` — välju salvestamata
- `:wq` — salvesta ja välju

**Pane tähele:** Kui `nano` puudub või `vim` hirmutab, siis peamine algaja oskus on see, et sa oskad redaktorist kindlalt väljuda.

**Edasi:** Järgmine loomulik samm: Käskude kuju ja argumentide loogika.

**Osa PDF:** [./osa-i-esimesed-sammud-spikker.pdf](#)

## Käskude kuju ja argumentide loogika

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** Õpi ühe näite pealt eristama käsku, lippu ja argumenti; siis ei tundu uus käsk iga kord täiesti uus keel.

### Üks näide

- `ls -la /etc` — terve kuju
- `ls` — käsk
- `-la` — lipud
- `/etc` — argument
- `cp vana.txt uus.txt` — kaks argumenti

### Tüüpilised kujud

- `grep -n 'root' fail.txt` — lipp + muster + fail
- `cp 'fail nimi.txt' koopia.txt` — nimi tühikuga
- `rm -- --algab-kriipsuga.txt` — ära tõlgenda lipuna
- `ls *.txt` — kõik `.txt` lõpuga failid
- `ls words.0?.txt` — üks märk keskel

### Olulised märgid

- `-` — lühike lipp
- `--` — pikk lipp või stop
- `"..."` — hoia koos; muutujad toimivad

- '...' — täht-tähelt nagu kirjas
- \* — failimuster
- ? — üks märk

**Pane tähele:** Kui nimes on tühikuid või erimärke, pane nimi jutumärkidesse; see on üks kõige tavalisemaid algaja komistuskohti.

**Edasi:** Järgmine loomulik samm: Sisend, väljund, torud ja suunamine.

**Osa PDF:** [./osa-i-esimesed-sammud-spikker.pdf](#)

## Sisend, väljund, torud ja suunamine

### Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** Ühenda käske nii, et ühe väljund saab järgmise sisendiks, või saada väljund faili; vea- ja tavaväljundit tasub mõelda eraldi.

### Põhikäsud

- `echo` — prindi tekst
- `printf` — vorminda tekst
- `cat` — näita faili
- `tee` — ekraan ja fail
- `wc` — loe kokku

### Tüüpilised kujud

- `käsk > valjund.txt` — stdout faili
- `käsk >> logi.txt` — lisa stdout faili lõppu juurde
- `käsk 2> vead.txt` — stderr eraldi
- `käsk > valjund.txt 2> vead.txt` — eralda mõlemad
- `käsk 2> /dev/null` — peida veateated
- `cat fail.txt | wc -l` — toru loendus
- `käsk ; järgmine` — järgmine alati
- `käsk && järgmine` — järgmine ainult edul
- `käsk || varuplaan` — varuplaan vea korral
- `käsk > koik.txt 2>&1` — stdout ja stderr koos

### Olulisemad lipud, märgid ja kiirrupud

- `>` — stdout faili
- `>>` — lisa stdout faili lõppu juurde
- `|` — anna edasi järgmisele
- `;` — käivita lihtsalt järjest
- `&&` — tee edasi edu korral
- `||` — tee edasi vea korral

- `2>` — `stderr` eraldi faili
- `2>&1` — `stderr` samasse kohta
- `/dev/null` — must auk väljundile

**Pane tähele:** Õpi kõigepealt rahulikult `>`, `»` ja `|`; alles siis lisa sellele `&&`, `||` ja `2>&1`.

**Edasi:** Järgmine loomulik samm: Esimene tervikharjutus: 30 minutit.

**Osa PDF:** `./osa-i-esimesed-sammud-spikker.pdf`

**Esimene tervikharjutus: 30 minutit**

## Peatüki täisspikker

Tase: **Algaja**

**Eesmärk:** kontrolli alguskohta; tööta eraldi harjutuskaustas; lisa faili read väikeste sammudena

### Lühike töövoog

- `pwd && ls` — kontrolli alguskohta
- `mkdir proov && cd proov` — loo töökaust
- `echo tere > sonad.txt` — tee esimene rida
- `echo maailm >> sonad.txt` — lisa järgmine rida
- `cat sonad.txt && wc -l sonad.txt` — kontrolli tulemust
- `cp sonad.txt koopia.txt` — tee koopia

### Olulised märgid

- `>` — kirjuta üle
- `>>` — lisa faili lõppu juurde
- `Ctrl-c` — katkesta programmi töö
- `man` — vaata abi

**Pane tähele:** Kui mõni käsk on vahepeal hägusaks läinud, peatu ja vaata `man` või eelmised Osa I peatükid üle.

**Edasi:** Järgmine loomulik samm: Linux, Unix, GNU, macOS, Windows ja shellid.

**Osa PDF:** `./osa-i-esimesed-sammud-spikker.pdf`

## Osa II: Süsteemi pilt ja haldus — täisspikker

Tase: **Baas ja süsteempilt**

## Linux, Unix, GNU, macOS, Windows ja shellid

### Peatüki täisspikker

Tase: **Baas ja süsteemipilt**

**Eesmärk:** erista süsteemi, shelli ja käsureatööriistu; sama terminaliaken ei tähenda veel, et Linux, macOS ja Windows käituksid ühtemoodi

#### Kontrollkäsud

- `uname -a` — süsteemi ja kerneli rida
- `echo "$SHELL"` — praegune shell
- `ps -p $$` — käivitud shelliprotsess
- `command -v ls` — kust käsk leitakse
- `sw_vers` — macOS-i versioon
- `wsl -l -v` — WSL-i jaotused

#### Olulised mõisted

- **Linux** — kernel ehk tuum
- **GNU** — paljud põhikäsud
- **shell** — tõlgendab käske
- **WSL** — Linux Windowsis
- **BSD** — macOS-i tööriistapere
- **PowerShell** — Windowsi käsu keskkond

**Pane tähele:** macOS-is ja Windowsis võivad sama nimega käsud käituda teistmoodi kui Linuxis.

**Edasi:** Järgmine loomulik samm: Failisüsteemi kaart.

**Osa PDF:** [./osa-ii-süsteemi-pilt-ja-haldus-spikker.pdf](#)

#### Failisüsteemi kaart

### Peatüki täisspikker

Tase: **Baas ja süsteemipilt**

**Eesmärk:** õpi eristama süsteemi juuri, oma kodukataloogi ja ajutisi ning süsteemseid kaustu; siis on lihtsam aru saada, kus tohid rahulikult katsetada

#### Põhikujud

- `pwd` — kontrolli asukohta
- `echo "$HOME"` — vaata oma kodu
- `cd /` — mine juurkataloogi
- `cd ~` — mine kodukataloogi
- `ls /etc | head` — piilu seadistusi

- `ls -ld /tmp` — vaata ajutist kausta

### Olulised teed

- `/` — kogu failipuu juur
- `~` — sinu kodukataloog
- `/etc` — süsteemi seadistused
- `/var` — logid ja muutuv sisu
- `/tmp` — ajutiste failide koht
- `/Users` / `/home` — kasutajate kodud

**Pane tähele:** `/` ja `~` ei ole sama asi: esimene on süsteemi juur, teine sinu enda tööala.

**Edasi:** Järgmine loomulik samm: Kettaruum ja süsteemi maht.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

### Kettaruum ja süsteemi maht

#### Peatüki täisspikker

Tase: **Baas ja süsteempilt**

**Eesmärk:** erista kogu failisüsteemi seis ja ühe konkreetse kausta mahtu; `df` ja `du` vastavad eri küsimustele

#### Põhikujud

- `df -h` — kõigi failisüsteemide maht
- `df -h .` — siinse failisüsteemi maht
- `du -sh .` — praeguse kausta summa
- `du -sh * 2> /dev/null` — alamkirjed ilma vigadeta
- `du -sh * 2> /dev/null | sort -h` — alamkirjed inimloetava mahu järgi
- `du -a . 2> /dev/null | sort -nr | less` — suurimad kirjed detailsemalt

#### Olulised võtmed ja vood

- `-h` — inimloetavad ühikud
- `-s` — ainult kokkuvõte
- `-a` — näita ka üksikuid kirjeid
- `2> /dev/null` — peida ligipääsuvead
- `| sort -h` — sordi inimloetavad mahud väiksemast suuremani
- `| sort -nr | less` — suurimad ette

**Pane tähele:** `df` ei ütle, milline kaust ruumi sööb; selleks vaata `du` väljundit.

**Edasi:** Järgmine loomulik samm: Õigused, omanikud ja täitmisbitid.

**Osa PDF:** ./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf

## Õigused, omanikud ja täitmisbitid

### Peatüki täisspikker

Tase: **Baas ja süsteempilt**

**Eesmärk:** loe õiguste rida, mõista rwx tähendust ja erista faili käivitatavust kataloogi läbikäigust

#### Põhikujud

- `ls -l fail.txt` — loe õiguste rida
- `chmod u+x skript.sh` — anna omanikule täitmine
- `chmod 644 naide.txt` — sea tekstifaili õigused
- `chmod 755 skript.sh` — sea käivitatava õigused
- `sudo chown kasutaja:grupp fail.txt` — muuda omanikku

#### Olulised õigusekujud

- `r` — loe sisu
- `w` — muuda sisu
- `x` — käivita või sisene
- `d` — tegu on kataloogiga
- `644` — tüüpiline tekstifail
- `755` — tüüpiline käivitatav

**Pane tähele:** Kataloogi x-õigus tähendab läbikäiku; see ei ole sama asi mis faili käivitamine.

**Edasi:** Järgmine loomulik samm: Kasutajad, grupid ja sudo.

**Osa PDF:** ./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf

## Kasutajad, grupid ja sudo

### Peatüki täisspikker

Tase: **Baas ja süsteempilt**

**Eesmärk:** õpi nägema, millise kasutajana sa töötad, millistes gruppides oled ja millal on sudo päriselt vajalik

#### Põhikujud

- `whoami` — kontrolli kasutajat
- `id` — loe grupid välja
- `groups` — vaata grupinimesid
- `sudo -l` — vaata sudo õigusi

- `sudo apt update` — üks kõrgem käsk
- `su - kasutaja` — vaheta kasutajat

### Olulised märgid ja rollid

- `$` — tavaline kasutaja
- `%` — teine levinud tavaviip
- `#` — root või kõrgemad õigused
- `root` — administraatori kasutaja
- `sudo -i` — ava kõrgem shell
- `sudo käsk` — eelista ühte käsku

**Pane tähele:** Ära kasuta `sudo-t` lihtsalt harjumusest; enne kontrolli, kas käsk tõesti vajab kõrgemaid õigusi.

**Edasi:** Järgmine loomulik samm: Muutujad, keskkond, `PATH` ja aliased.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

### Muutujad, keskkond, `PATH` ja aliased

#### Peatüki täisspikker

Tase: **Baas ja süsteempilt**

**Eesmärk:** erista shellimuutujat, keskkonnamuutujat, käsuotsingu teed ja aliaist; `PATH-i` järjekord otsustab, milline väline käsk esimesena leitakse.

#### Põhikujud

- `nimi='Mari'; echo "$nimi"` — shellimuutuja kehtib selles shellis
- `export DEMO=1` — anna muutuja edasi käivitavatele programmidele
- `env | grep DEMO` — kontrolli, kas muutuja on keskkonnas
- `echo "$PATH"` — vaata käskude otsinguteed
- `command -v python3` — leia esimene väline vaste
- `type cd` — vaata käsu liiki
- `type -a grep` — näita kõik vasted
- `export PATH="$HOME/bin:$PATH"` — pane oma käskude kataloog otsingutee ette
- `alias ll='ls -lah'` — loo lühinimi sagedasele käsule
- `source ~/.zshrc` — loe `zsh` seadistus uuesti

#### Olulised mõisted

- `nimi='Mari'` — shellimuutuja
- `export NIMI='Mari'` — keskkonnamuutuja
- `PATH` — otsinguteede järjekord
- `$HOME/bin` — oma käskude koht
- `type` — alias, builtin või programm

- `~/.zshrc` — zsh seadistusfail
- `~/.bashrc` — bash seadistusfail

**Pane tähele:** PATH-i järjekord loeb: vasakul olev vaste leitakse enne; ära lisa tundmatut kataloogi ettepoole lihtsalt sellepärast, et juhend nii ütleb.

**Edasi:** Järgmine loomulik samm: Paketihaldus: apt, dnf, pacman, brew.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

**Paketihaldus: apt, dnf, pacman, brew**

## Peatüki täisspikker

Tase: **Baas ja süsteemipilt**

**Eesmärk:** vali õige tase: süsteemi paketihaldur tööriistadele, keele paketihaldur teekidele ja projektisõltuvused võimalusel projekti sisse

## Põhikujud

- `apt search ripgrep` — otsi Debianis või Ubuntu
- `sudo apt install ripgrep` — paigalda Debianis või Ubuntu
- `sudo dnf install ripgrep` — paigalda Fedoras
- `sudo pacman -S ripgrep` — paigalda Archis
- `brew install ripgrep` — paigalda Homebrew'ga
- `python3 -m pip install requests` — paigalda Pythoni pakett aktiivsesse keskkonda
- `npm install` — paigalda projekti Node.js sõltuvused
- `command -v rg` — kontrolli, kas käsk leitakse

## Tööriistad ja tasemed

- `apt, dnf, pacman` — Linuxi süsteemipaketid
- `brew` — macOS-i kasutajataseme tööriistad
- `update` — värskenda nimekirja või retsepte
- `upgrade` — uuenda paigaldatud paketid
- `--cask` — Homebrew graafiline rakendus
- `venv` — Pythoni projekti eraldi keskkond
- `package.json` — Node.js projekti sõltuvused

**Pane tähele:** Ära aja segi süsteemi paketihaldurit ja pip/npm-i; need lahendavad eri taseme probleeme.

**Edasi:** Järgmine loomulik samm: Lihtne veaotsing käsureal.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

## Lihtne veaotsing käsureal

### Peatüki täisspikker

Tase: **Baas ja süsteemipilt**

**Eesmärk:** kitsenda viga enne parandamist: täpne veateade, asukoht, olema-solu, käsu leitavus, õigused ja shelli/süntaksi kontroll

#### Kontrollrada

- `pwd` — kontrolli, kus oled
- `ls -lah` — vaata faile, õigusi ja peidetud kirjeid
- `command -v rg` — kas käsk leitakse
- `type -a rg` — näita kõik sama nimega vasted
- `ls -l skript.sh` — vaata skripti õigusi
- `head -n 1 skript.sh` — vaata skripti shebang'i
- `bash skript.sh` — käivita teadlikult Bashiga
- `grep --help` — kinnita käsu süntaks

#### Levinud veateated

- `command not found` — käsk puudub, nimi vale või pole PATH-is
- `No such file or directory` — tee, failinimi või asukoht on vale
- `Permission denied` — õigus ei luba lugeda, kirjutada või käivitada
- `syntax error` — vale shell või vale käsukuju
- `package not found` — paketinimi või nimekiri on vale
- `database is locked` — teine paketi haldur töötab

**Pane tähele:** Pane kõigepealt kirja täpne käsk ja täpne veateade; ilma nendeta muutub veaotsing oletamiseks.

**Edasi:** Järgmine loomulik samm: Võrgu põhitööriistad.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

#### Võrgu põhitööriistad

### Peatüki täisspikker

Tase: **Baas ja süsteemipilt**

**Eesmärk:** erista nelja küsimust: kas host vastab, kas veeb vastab, millised liidesed on ja kas kohalik teenus kuulab porti

#### Kontrollrada

- `curl -I https://example.com/` — kontrolli veebiteenust
- `ping -c 4 example.com` — kontrolli hosti ICMP vastust
- `ip a` — vaata Linux'i liideseid

- `ifconfig` — vaata macOS-i liideseid
- `ss -ltn` — vaata Linuxi kuulavaid TCP-porte
- `lsof -iTCP -sTCP:LISTEN -n -P` — vaata macOS-i kuulavaid porte
- `nc -vz example.com 443` — kontrolli TCP ühendust porti

### Olulised võtmed ja vaated

- `-I` — küsi ainult HTTP päised
- `-L` — järgi HTTP suunamisi
- `LISTEN` — port kuulab
- `127.0.0.1` — ainult oma masin
- `0.0.0.0` — kõik IPv4 liidesed
- `ip/ss` — Linuxi tööriistad
- `ifconfig/lsof` — macOS-i põhivariandid

**Pane tähele:** ping ja HTTP ei ole sama asi; veeb võib töötada ka siis, kui host pingile ei vasta.

**Edasi:** Järgmine loomulik samm: Failide kopeerimine ja sünkroonimine.

**Osa PDF:** [./osa-ii-susteemi-pilt-ja-haldus-spikker.pdf](#)

## Osa III: Failid, võrk ja süsteemitöö — täisspikker

Tase: **Praktiline kasutus**

### Failide kopeerimine ja sünkroonimine

#### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** vali tööriist allika järgi: `cp` samas masinas, `scp` üle SSH, `rsync` korduvaks sünkroonimiseks ja `curl/wget` veebist toomiseks

#### Põhikujud

- `cp fail.txt koopia.txt` — tee koopia
- `cp -R kaust kaust-koopia` — kopeeri kaust
- `scp fail.txt kasutaja@server:/tmp/` — saada serverisse
- `rsync -avn kaust/ kasutaja@server:/tmp/kaust/` — kontrolli enne
- `rsync -av kaust/ kasutaja@server:/tmp/kaust/` — sünkrooni sisu
- `wget https://example.com/arhiiv.tar.gz` — laadi fail
- `curl -L -O https://example.com/arhiiv.tar.gz` — järgi ja salvesta

### Olulisemad lipud, märgid ja kiirpud

- `cp -R` — kopeeri kaust
- `scp -r` — saada kaust

- `rsync -a` — säilita struktuur
- `rsync -n` — proovi enne
- `curl -L` — järgi suunamist
- `curl -O` — salvesta nimega

**Pane tähele:** Vali tööriist allika järgi: `cp` samas masinas, `scp` üle SSH, `rsync` korduvaks sünkroonimiseks, `curl/wget` veebist toomiseks.

**Edasi:** Järgmine loomulik samm: Kauglogimine ja SSH.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## Kauglogimine ja SSH

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** tee SSH korduvkasutatavaks: kontrolli olemasolevaid võtmeid, lisa avalik võti serverisse, kasuta `~/.ssh/config` alias't ja macOS-is Keychaini

#### Põhirada

- `ssh kasutaja@server.example.org` — proovi tavalist sisselogimist
- `ls -la ~/.ssh` — vaata olemasolevaid võtmefaile
- `ssh-add -l` — vaata agendis olevaid võtmeid
- `ssh-keygen -t ed25519 -C 'kasutaja@arvuti'` — loo võti ainult vajadusel
- `ssh-add --apple-use-keychain ~/.ssh/id_ed25519` — macOS: salvesta paroolifraas Keychaini
- `ssh-copy-id -i ~/.ssh/id_ed25519.pub kasutaja@server.example.org` — lisa avalik võti serverisse
- `ssh opik-server` — kasuta config-alias't
- `ssh -v opik-server` — diagnostika

#### Olulised failid ja mõisted

- `id_ed25519` — privaatvõti; ära jaga
- `id_ed25519.pub` — avalik võti serverisse
- `~/.ssh/config` — ühenduse alias
- `~/.ssh/authorized_keys` — lubatud võtmed
- `known_hosts` — serverite sõrmejäljed
- `AddKeysToAgent` — lisa võtmed agent
- `UseKeychain` — macOS paroolifraasi hoidmine
- `IdentitiesOnly yes` — kasuta nimetatud võtit

**Pane tähele:** Alusta sisselogimisest ja võtmetest; port forwarding nagu `ssh -L` ... on järgmine, mitte esimene samm.

**Edasi:** Järgmine loomulik samm: Veebist sisu toomine ja tekstivaade: curl, wget, lynx.

**Osa PDF:** ./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf

**Veebist sisu toomine ja tekstivaade:** curl, wget, lynx

## Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** vali curl ühe vastuse või päiste jaoks, wget allalaadimise jaoks ja lynx HTML-i rahulikuks tekstivaateks

### Põhikujud

- curl -I https://example.com/ — kontrolli URL-i
- curl -L -o leht.html https://example.com/ — salvesta HTML
- wget -O leht.html https://example.com/ — laadi faili
- curl -sL https://example.com/ | lynx -stdin -dump — loe tekstina
- lynx -dump -listonly https://example.com/ — kogu lingid

### Olulisemad lipud, märgid ja kiirnopud

- curl -I — vaata päiseid
- curl -L — järgi suunamist
- curl -O — salvesta nimega
- wget -c — jätkta tõmmet
- lynx -dump — tekstvaade
- lynx -listonly — ainult lingid

**Pane tähele:** Alusta alati ühest URL-ist; suur crawl ja automaatne kraapimine olgu järgmine, mitte esimene samm.

**Edasi:** Järgmine loomulik samm: Arhiivid ja pakkimine.

**Osa PDF:** ./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf

## Arhiivid ja pakkimine

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** arhiiv koondab faili- või kaustapuu üheks tervikuks ja pakkimine teeb selle terviku väiksemaks või lihtsamini jagatavaks

## Põhikujud

- `tar -cf proov.tar kaust/` — loo arhiiv
- `tar -tf proov.tar` — vaata sisu
- `tar -czf proov.tar.gz kaust/` — loo gzip-arhiiv
- `tar -xzf proov.tar.gz -C siht/` — paki lahti sihtkohta
- `zip -r proov.zip kaust/` — loo zip
- `unzip -l proov.zip` — vaata zip sisu

## Olulisemad lipud, märgid ja kiirpud

- `tar -cf` — loo arhiiv
- `tar -tf` — vaata sisu
- `tar -czf` — gzip-iga kokku
- `tar -xzf` — paki gzip lahti
- `zip -r` — zip kaust
- `unzip -l` — vaata zip sisu

**Pane tähele:** Enne lahtipakkimist vaata võimalusel arhiivi sisu; see aitab näha, kas sees on oodatud juurkaust ja failipuud.

**Edasi:** Järgmine loomulik samm: Tervete kataloogipuude haldus ja jagamine.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## Tervete kataloogipuude haldus ja jagamine

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** terve projekti või kaustapuu puhul vali tööriist selle järgi, kas tahad koopiat, korduvat sünkroonimist, arhiivi või versioonialalugu

## Põhikujud

- `cp -R projekt projekt-koopia` — tee kohalik koopia
- `rsync -avn projekt/ projekt-varu/` — kontrolli enne
- `rsync -av projekt/ kasutaja@server:/srv/projekt/` — sünkrooni serverisse
- `scp -r projekt kasutaja@server:/tmp/` — saada kiirelt
- `tar -czf projekt.tgz projekt/` — tee hetkearhiiv
- `git` — jälgi ajalugu

## Olulisemad lipud, märgid ja kiirpud

- `projekt/` — sisu sees
- `projekt` — kaust tervikuna
- `rsync -n` — proovi enne
- `cp -a` — säilita metaandmed

**Pane tähele:** Kui vajad korduvat sünkroonimist, eelista tavaliselt rsync-i; scp -r sobib pigem üheks kiireks ülekandeks.

**Edasi:** Järgmine loomulik samm: Protsessid, tööd ja signaalid.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## Protsessid, tööd ja signaalid

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** erista protsessi, shelli tööd ja signaali: vaata, mis jookseb, peata vajadusel ja too pikk töö taustale

#### Põhikujud

- `ps aux | grep python` — otsi protsess
- `ps aux | sort -nrk 3 | head` — CPU sööjad
- `ps aux | sort -nrk 4 | head` — mälusööjad
- `sleep 300 &` — käivita taustal
- `jobs` — vaata töid
- `fg %1` — too töö ette
- `top` — jälgi reaalajas
- `kill` — saada signaal
- `bg` — jätkata taustal

#### Olulisemad lipud, märgid ja kiirrupud

- `Ctrl-c` — katkesta programmi töö
- `Ctrl-z` — peata ajutiselt; fg ette, bg taustale
- `&` — käivita taustal
- `%1` — töö number
- `kill -15` — viisakas lõpp
- `kill -9` — jõuga lõpp

**Pane tähele:** Kasuta `kill -9` alles siis, kui tavalisem `kill` või `kill -15` ei lõpeta protsessi viisakalt.

**Edasi:** Järgmine loomulik samm: Logid ja teenused.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## Logid ja teenused

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** kui teenus ei tööta, alusta kahest küsimusest: kas teenus on üldse aktiivne ja mida logid selle kohta ütlevad

### Põhikujud

- `systemctl status ssh` — vaata seis
- `journalctl -u ssh -n 50` — loe viimast logi
- `journalctl -u ssh -f` — jälgi teenust
- `tail -n 50 /var/log/syslog` — loe faili lõppu
- `tail -f /var/log/syslog` — jälgi faili
- `tail -n 50 /var/log/syslog | less` — sirvi viimaseid ridu

### Olulisemad lipud, märgid ja kiirnopud

- `status` — teenuse seis
- `-u nimi` — üks teenus
- `-n 50` — viimased read
- `-f` — jälgi reaajas
- `--since today` — ainult tänane

**Pane tähele:** Alusta teenuse puhul tavaliselt `systemctl status-est` ja alles siis mine sügavamale `journalctl` või logifailide juurde.

**Edasi:** Järgmine loomulik samm: Püsivad terminalisessioonid: `tmux` ja `screen`.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

### Püsivad terminalisessioonid: `tmux` ja `screen`

#### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** kasuta püsivat sessiooni siis, kui SSH võib katkeda või kui pikk töö peab jätkuma ka pärast akna sulgemist

### Põhikujud

- `tmux new -s opik` — loo `tmux`
- `tmux attach -t opik` — naase `tmux-i`
- `screen -S opik` — loo `screen`
- `screen -r opik` — naase `screeni`
- `nohup pikk-kaik > logi 2>&1 &` — jäta töö käima
- `disown` — seo shellist lahti

### Olulisemad lipud, märgid ja kiirnopud

- `Ctrl-b d` — eraldu `tmux-ist`
- `Ctrl-a d` — eraldu `screenist`

- `tmux ls` — sessioonide loend
- `screen -ls` — sessioonide loend

**Pane tähele:** Kui pead valima ühe harjumuse, vali `tmux`: see jätab alles terve sessiooni, mitte ainult ühe käsu.

**Edasi:** Järgmine loomulik samm: Graafilised rakendused kaugmasinast.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## Graafilised rakendused kaugmasinast

### Peatüki täisspikker

Tase: **Praktiline kasutus**

**Eesmärk:** kaugelt graafika kasutamisel eelista lihtsaimat toimivat teed: veebiliides enne X11-edastust, port forwarding enne toorest kaugtöölauda.

### Põhikujud

- `ssh -L 8888:localhost:8888 kasutaja@server` — too veebiliides kohale
- `http://localhost:8888` — ava edasi suunatud teenus
- `ssh -X kasutaja@server` — proovi lihtsat X11
- `code --remote ssh-remote+server /tee/projektini` — IDE üle SSH

### Valiku rusikareeglid

- **veebiliides** — tavaliselt mugavam
- `ssh -L` — port edasi
- `ssh -X` — X11 erijuht
- **Remote SSH** — arenduseks parem

**Pane tähele:** Kui sul on valida, eelista brauserit või Remote SSH-d; X11 forwarding olgu pigem varuplaan, mitte esimene mõte.

**Edasi:** Järgmine loomulik samm: Teksti otsimine: `grep` ja sugulased.

**Osa PDF:** [./osa-iii-failid-vork-ja-susteemitoo-spikker.pdf](#)

## 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 kiirpupud

- 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 üksshaaval

### Olulisemad lipud, märgid ja kiirnopud

- `-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 kiirrupud

- `#!/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 kiirrupud

- `*/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](#)

## Osa V: Arendus ja töövood — täisspikker

Tase: Edasijõudnu

## Git, GitHub ja töövoog

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** Git hoiab muutuste ajalugu; erista tööpuud, stage'i ja commit'i ning vaata diff enne üle.

#### Põhikujud

- `git status` — vaata seis
- `git diff` — muutused tööpuus
- `git diff --cached` — muutused stage'is
- `git add fail.txt` — pane stage'i
- `git commit -m '...'` — tee commit
- `git switch -c parandus` — loo haru
- `git pull --ff-only` — uuenda puhtalt
- `git push -u origin parandus` — saada haru
- `git restore --staged fail.txt` — võta stage'ist

#### Olulisemad lipud, märgid ja kiirrupud

- `HEAD` — praegune tipp
- `main` — põhirida
- `origin` — kaugrepo
- `stage` — järgmine commit

**Pane tähele:** Uus jälgimata fail ei ilmu tavalisse `git diff` vaatesse; pärast `git add` kontrolli esimese commit'i sisu käsuga `git diff --cached`.

**Edasi:** Järgmine loomulik samm: Pythoni venv ja eraldatud keskkonnad.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

#### Pythoni venv ja eraldatud keskkonnad

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** venv hoiab ühe projekti Pythoni paketid eraldi; kõige olulisem on, et terminal, pip ja IDE näeksid sama keskkonda.

#### Põhikujud

- `python3 -m venv .venv` — loo keskkond
- `source .venv/bin/activate` — aktiveeri shellis
- `python -m pip install -U pip` — uuenda pip
- `python -m pip install requests` — paigalda pakett

- `python -m pip list` — vaata pakke
- `command -v python` — kontrolli tõlgendit
- `deactivate` — välju keskkonnast

### Olulised märgid

- `(.venv)` — aktiivne keskkond
- `.venv/` — projekti sees
- `python -m pip` — kindlam kui pip
- Docker — terve keskkond

**Pane tähele:** Kui pip install läks valesse kohta, kontrolli kohe `command -v python` ja `command -v pip`; enamasti on probleem just vales aktiivses keskkonnas.

**Edasi:** Järgmine loomulik samm: Docker'i alused.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

### Docker'i alused

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** image on käivitusmall; konteiner on sellest tehtud töötav eksemplar; andmed püsivad ainult hostis, volume'is või Git-repos.

### Põhikäsud

- `docker --version` — kontrolli olemasolu
- `docker run --rm alpine echo tere` — ohutu esimene käivitus
- `docker build -t rakendus .` — ehita image
- `docker run --rm rakendus` — käivita image
- `docker ps` — töötavad konteinerid
- `docker logs -f nimi` — jälgi logi
- `docker exec -it nimi sh` — sisene töötavasse konteinerisse
- `docker compose up -d` — teenused taustale
- `docker compose down` — peata komplekt

### Olulisemad lipud, märgid ja kiirpud

- `image` — käivitusmall
- `konteiner` — töötav eksemplar
- `Dockerfile` — image'i retsept
- `compose.yaml` — mitu teenust
- `volume` — püsiv andmesisu
- `registry` — image'ite hoidla
- `--rm` — kustuta lühikatske konteiner pärast lõppu

- `-p 8000:8000` — seo hosti port konteineri pordiga
- `--mount type=bind,src="$PWD",dst=/app` — jaga praegune kaust konteinerisse
- `-w /app` — määra konteineri töökaust
- `down -v` — peatab ja kustutab ka volume'id

**Pane tähele:** Kui konteiner ei tööta, vaata enne docker logs; ära käivita tundmatut image'it oma kodukataloogi või saladuste bind mount'iga.

**Edasi:** Järgmine loomulik samm: IDE-d ja arenduskeskkonnad.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

## IDE-d ja arenduskeskkonnad

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** hea arenduskeskkond tähendab, et terminal, Git, interpreter ja vajadusel konteiner osutavad samale projektile, mitte eri maailmadele.

#### Kontrollid ja valikud

- `python3 --version` — kontrolli Pythonit
- `python3 -m pip --version` — kontrolli pip-i
- `git --version` — kontrolli Git-i
- `docker --version` — kontrolli Dockerit
- `node --version` — kontrolli Node'i
- `npm --version` — kontrolli npm-i

#### Olulised mõisted

- `interpreter` — tegelik käivitaja
- `sisseehitatud terminal` — sama projektivaade
- `Remote SSH` — tööta kaugmasinas
- `devcontainer` — IDE konteineris

**Pane tähele:** Kui IDE-s “miski ei tööta”, kontrolli kõigepealt, kas IDE kasutab sama Pythonit, Node'i või konteinerit mis sinu terminal.

**Edasi:** Järgmine loomulik samm: Andmeteaduse eelteadmised käsurea vaates.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

## Andmeteaduse eelteadmised käsurea vaates

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** andmeteaduse stardis ei piisa ühest tööriistast; vaja on korraga failivormingute tunnetust, SQL-i mõtteviisi, programmeerimist ja statistilist mõtlemist.

### Põhikujud

- fail -> head/less -> column/jq -> sqlite3 -> python3 — alusta väikese vaatusega
- Pythoni venv — projekti töölaud
- CSV, JSON ja XML — andmete kuju
- SQLite ja Python — esimene andmebaas
- Teksti teisendamine — väiksed filtrid
- Vood ja tabelid — koondamine

### Põhiteljed

- programmeerimine — korduv töö loogikaks
- SQL — küsi ja seo andmeid
- vormingud — CSV, JSON, XML
- statistika — anna tähendus

**Pane tähele:** Ära mine otse mudeli või notebook'i juurde enne, kui oled vähemalt korra kontrollinud, mis kujul andmed sul tegelikult on.

**Edasi:** Järgmine loomulik samm: CSV, JSON ja XML käsuraal.

**Osa PDF:** ./osa-v-arendus-ja-toovood-spikker.pdf

### CSV, JSON ja XML käsuraal

#### Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** vormingu tüüp ütleb, milline tööriist mõistab andmeid kõige loomulikumalt: tabelit, puud või märgendistruktuuri.

### Põhikujud

- head fail.csv — vaata CSV algust
- column -s, -t < fail.csv — joonda lihtne CSV
- cut -d, -f1 fail.csv — võta üks veerg
- python3 -m json.tool fail.json — vorminda JSON
- jq '.students[].name' fail.json — vali JSON välju
- xmllint --format fail.xml — vorminda XML

### Olulised mõisted

- CSV — lihtne tabel

- JSON — objektide puu
- XML — märgendipuu
- jq — JSON filtriks

**Pane tähele:** Kui CSV-s on jutumärgid, komad välja sees või reavahetused, ei pruugi `cut -d`, enam piisata; siis mine pigem Pythonisse või spetsiaalsema parseri juurde.

**Edasi:** Järgmine loomulik samm: Andmebaasi algus: `sqlite` ja `Python`.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

## Andmebaasi algus: `sqlite` ja `Python`

### Peatüki täisspikker

Tase: **Edasijõudnu**

**Esmärk:** `SQLite` on sild tekstifailide ja päris andmebaasimõtte vahel: failina lihtne, `SQL`-i mõttes siiski relatsiooniline andmebaas.

#### Põhikujud

- `sqlite3 andmed.db` — ava või loo fail
- `sqlite3 andmed.db '.tables'` — vaata tabeleid
- `sqlite3 andmed.db '.schema'` — vaata struktuuri
- `sqlite3 andmed.db 'select * from students limit 5;'` — piilu ridu
- `sqlite3 andmed.db 'select city, count(*) from students group by city;'` — koonda read
- `sqlite3 andmed.db 'select s.name, r.score from results r join students s on s.id = r.student_id;'` — ühenda tabelid
- `python3 naide.py` — kasuta `Python`ist

#### Olulised mõisted

- `primary key` — rea unikaalne id
- `foreign key` — viide teise tabelisse
- `JOIN` — seo tabelid
- `GROUP BY` — koonda read

**Pane tähele:** Ära tee `JOIN`-i ainult käsu pärast; enne joonista enda jaoks välja, milline väli viitab millisele tabelile.

**Edasi:** Järgmine loomulik samm: Kompileerimine ja käivitamine: `shell`, `Python`, `C`, `C++`, `Go`, `Rust`, `Java`.

**Osa PDF:** [./osa-v-arendus-ja-toovood-spikker.pdf](#)

**Kompileerimine ja käivitamine: shell, Python, C, C++, Go, Rust, Java**

## Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** mõni fail käivitatakse tõlgendiga otse; mõni fail kompileeritakse kõigepealt teise vormi; mõni tulemus on päris binaar

### Põhikäsud

- `go` — Go tööriist
- `cargo` — Rusti tööriist
- `python3` — käivita Python
- `cc` — kompileeri C
- `javac` — kompileeri Java
- `java` — käivita JVM-is

### Tüüpilised kujud

- `chmod +x hello.sh` — õigused
- `python3 hello.py` — käivita Python
- `cc hello.c -o hello-c` — kompileeri C
- `go run hello.go` — Go tööriist
- `go build -o hello-go hello.go` — Go tööriist
- `mkdir -p hello-rust` — loo puuduv rada

**Pane tähele:** Kui käivitatav fail “ei tööta”, kontrolli kõigepealt, kas ta üldse ehitati valmis, kus ta asub ja kas pead kasutama kuju `./fail`.

**Edasi:** Järgmine loomulik samm: LaTeX käsuraal.

**Osa PDF:** `./osa-v-arendus-ja-toovood-spikker.pdf`

### LaTeX käsuraal

## Peatüki täisspikker

Tase: **Edasijõudnu**

**Eesmärk:** LaTeX-i mõte on lihtne: lähtetekst jääb tekstifailiks, aga buildist sünnib väga täpselt juhitud PDF.

### Põhikujud

- `pdflatex tere.tex` — kompileeri PDF
- `xelatex tere.tex` — parem Unicode tugi
- `latexmk -pdf tere.tex` — korda vajalikud käigud
- `latexmk -xelatex tere.tex` — xelatex automaatselt

- `ls tere.*` — vaata abifaile
- `open tere.pdf` — ava tulemus

### Olulised failid

- `.tex` — lähtefail
- `.pdf` — tulemus
- `.aux` — ristviidete abi
- `.log` — kompileerimislogi

**Pane tähele:** Kui ristviited või sisukord ei ilmu kohe õigesti, ei tähenda see tingimata viga; LaTeX vajab sageli rohkem kui üht kompileerimiskäiku.

**Edasi:** Järgmine loomulik samm: Lisa A: kopeeritavad minitestid.

**Osa PDF:** `./osa-v-arendus-ja-toovood-spikker.pdf`

## Lisad — täisspikker

Tase: **Referents**

**Lisa A: kopeeritavad minitestid**

### Peatüki täisspikker

Tase: **Referents**

**Eesmärk:** Siia koondame lühikesed peatükkide kaupa jaotatud copy-paste harjutused.

### Põhikujud

- `mkdir -p ~/tmp/test1` — loo puuduv rada
- `cd ~/tmp/test1` — vaheta kaust
- `touch a.txt b.txt` — loo või aja tempel
- `mkdir kaust` — loo kaust
- `mv b.txt kaust/` — liiguta/nimeta
- `ls -la` — pikk + peidetud
- `printf` — vorminda tekst

### Olulisemad lipud, märgid ja kiirrupud

- `|` — toru edasi
- `>` — kirjuta üle
- `>>` — lisa faili lõppu juurde

**Pane tähele:** Kui kahtled, kontrolli enne `pwd` ja `ls`, siis tee alles järgmine samm.

**Edasi:** Järgmine loomulik samm: Lisa B: spikrite register.

**Osa PDF:** ./lisad-spikker.pdf

## **Lisa B: spikrite register**

### **Peatüki täisspikker**

Tase: **Referents**

**Eesmärk:** kõige lühemad meespead teemade kaupa: failid, tekst, võrk, arendus, loogika ja protsessid

#### **Mida siit leiad**

- **Failid** — liigu ja vaata
- **Tekst** — loe ja filtreeri
- **Võrk** — ühendu ja kopeeri
- **Arendus** — Git, Docker, build
- **Loogika** — torud ja vead
- **Protsessid** — tööd ja signaalid

**Pane tähele:** See lisa on kõige lühem meespea; kui mõni käsk tundub hägune, mine tagasi vastava peatüki juurde.

**Edasi:** Järgmine loomulik samm: Lisa C: sõnastik ja terminoloogia.

**Osa PDF:** ./lisad-spikker.pdf

## **Lisa C: sõnastik ja terminoloogia**

### **Peatüki täisspikker**

Tase: **Referents**

**Eesmärk:** ühtlusta sõnad nagu CLI, GUI, viip, shell, lipp, stdin, stdout ja repo kogu õpiku ulatuses

#### **Põhimõisted**

- **CLI** — käsurida
- **GUI** — graafiline liides
- **viip** ehk **prompt** — ootab sisestust
- **shell** — tõlgendab käske
- **lipp** — käsu valik
- **repo** — Git-i hoidla

**Pane tähele:** Kui valid kahe sõna vahel, eelista seda lisa kui kogu õpiku terminoloogilist alust.

**Edasi:** Järgmine loomulik samm: Lisa D: edasised võimalused ja lugejate soovid.

**Osa PDF:** ./lisad-spikker.pdf

## Lisa D: edasised võimalused ja lugejate soovid

### Peatüki täisspikker

Tase: **Referents**

**Eesmärk:** järgmine voor peaks pigem süvendama olemasolevaid tugevaid teemasid kui lisama juhuslikke uusi baaskäske

### Tugevad järgmised teemad

- R ja notebook'id — analüüsi järgmine aste
- Git järgmine aste — harud ja konfliktid
- Shelliskriptid II — veakindlam loogika
- Statistika sild — andmetöö mõtteviis
- systemd timerid — ajastus Linuxis
- Varukoopiad — taastamine ja kontroll

**Pane tähele:** Siin nimekirjas ei ole “mida iga hinna eest kohe lisada”, vaid teemad, mis annaksid järgmises ringis kõige rohkem väärtust.

**Edasi:** Järgmine loomulik samm: Lisa E: usaldusväärsed viited ja lisalugemine.

**Osa PDF:** ./lisad-spikker.pdf

## Lisa E: usaldusväärsed viited ja lisalugemine

### Peatüki täisspikker

Tase: **Referents**

**Eesmärk:** ametlikud käsiraamatud ja getting started allikad shelli, SSH, Git-i, Dockeri, JSON-i, SQLite'i ja LaTeX-i jaoks

### Edasiõppimise teemad

- Shellid ja põhikäsud — bash, zsh, coreutils
- SSH ja võrk — OpenSSH, curl, wget
- Git ja GitHub — docs ja Pro Git
- Docker — get started ja compose
- JSON, CSV, XML, SQLite — ametlikud viited
- LaTeX — projekt ja CTAN

**Pane tähele:** Alusta iga teema puhul esimesest viitest; reference-manuaal on tavaliselt järgmine samm alles siis, kui põhikujutlus on olemas.

**Edasi:** Järgmine loomulik samm: Lisa F: shelli seadistusfailid bash ja zsh jaoks.

**Osa PDF:** ./lisad-spikker.pdf

**Lisa F:** shelli seadistusfailid bash ja zsh jaoks

## Peatüki täisspikker

Tase: **Referents**

**Eesmärk:** See lisa annab kopeeritavad algusfailid kahe levinud shelli jaoks:

### Põhikujud

- `echo "$SHELL"` — prindi tekst
- `ls -la "$HOME"` — pikk + peidetud
- `cp ~/.zshrc ~/.zshrc.backup` — kopeeri
- `cp ~/.bashrc ~/.bashrc.backup` — kopeeri
- `alias ll='ls -lh'` — lühinimi
- `alias la='ls -la'` — lühinimi
- `mv` — liiguta/nimeta
- `rm` — kustuta

### Olulisemad lipud, märgid ja kiirnopud

- `;` — järjest
- `|` — toru edasi
- `2>` — vead faili
- `-h` — lühike abi
- `&&` — ainult õnnestumisel

**Pane tähele:** Ära pane kõiki tasemeid korraka samasse faili. Alusta ühest tasemest ja lisa hiljem juurde.

**Edasi:** Selle osa viimase peatüki järel tasub vaadata järgmise osa algust või osa-spikrit.

**Osa PDF:** [./lisad-spikker.pdf](#)