

# Linux – käske

# Käsu parameetrid

- Enamus käske tunnistab parameetrit `--help`. Viimase abil kuvatakse lühiülevaade käsust ja tema parameetritest.
- `ls --help` – Käsu `ls` lühikirjeldus ja parameetrid.

# Kataloogipuu modifitseerimise käsud

- **mv** – failide ja kaustade teisaldamine;
- **cp** - kopeerimine;
- **rm** - kustutamine;
- **mkdir** – kausta loomine;
- **rmdir** – kausta kustutamine;
- **ln; ln -s** – viitade loomine.

# Kataloogipuus liikumine

- **cd; popd; pushd** – aktiivse kausta vahetamine.

# Info failide ja kaustade kohta

- **find** – otsimine;
- **ls** – kaustaloend;
- **cat** – failisisu kuvamine;
- **file** – faili tüübi kindlaks tegemine;
- **strings** – failist kuvatavate täheühendite kuvamine;
- **od** – kuvab faili sisu kaheksandsüsteemis (või teistes).

# Info vaba kettaruumi kohta

- **du** – kettakasutuse kuvamine;
- **df** – kettaruumi kasutuse kuvamine.

# Failidele juurdepääsuõiguste loomine

- **chmod**
- **chown**
- **chgrp**
- **umask**
- **UID**
- **GID**

# Ketaste ühendamine, partitsioonid

- **mount** – seadme (võrguketta) külge ühendamine;
- **mkfs** – linuxi failisüsteemi ehitamine;
- **badblocks** – loetamatute blokkide otsimine;
- **fsck** – kontrollitakse ja parandatakse linuxi failisüsteemi;
- **fdisk** – partitsioonitabeli redigeerimine;
- **ln** – linkide loomine;
- **dd** – failide kopeerimine ja konvertimine.



# Protsessid

- **ps; pstree; pidof; jobs; top; free** – info protsesside kohta;
- **kill; killall; ctrl+c; ctrl+z** – protsessi tapmine;
- **nice; renice; time; ulimit** – prioriteedi seadmine;
- **at; atq; atrm** – töö hilisemaks käivituseks seadmine.

# Administreerimine

- **uname** – süsteemiinfo;
- **finger; w; who** – hetkel arvutisse loginud isikute loend;
- **dmesg; last; /etc/syslog.conf; /var/log** – logifailid;
- **top; vmstat; free; ps; pstree** – mälu ja protsessid;
- **crontab** – programmide regulaarne käivitamine.

# Kasutajad

- **useradd (adduser)** – kasutaja lisamine;
- **userdel** – kasutaja kustutamine;
- **quota; quotaon; quotaoff, ...** – kettakvoot;
- **su; sudo** – muu kasutaja (näiteks süsteemiülema) õigustes programmide (terminali) käivitamine.

# UNIXi kest (*shell*)

Kest on kasutajaprogramm:

- Tuuma ja kasutaja vaheliseks suhtluseks
- Käivitab käske, mida loetakse sisendist või failist
- Ei ole tuuma osa, vaid kasutab seda programmide töötamiseks
- bash, tcsh, sh, csh, ...

# Kestaskript

- Kestaskript on järjestikune käskude kogum.
- Käsud kirjutatakse tekstifaili.
- Käivitamiseks tuleb failile anda käivitamisõigus (chmod 755 failinimi).
- Esimene rida skriptis sisaldab teed skriptis kasutatavale kestatüübile:  

```
#! /bin/sh
```
- Skriptikeel on lihtne programmeerimiskeel, millega saab skripte kirjutada.

# Milleks kestackripti vaja on?

- Kestaskript võtab klaviatuurilt või failist sisendi ja väljastab selle ekraanil
- Sellega saab ise oma käske luua
- Sellega säästab aega (üks käsk vs mitu käsku)
- Saab automatiseerida igapäevaseid ülesandeid
- Süsteemi administreerimiseks

# UNIXI käsud vt algus

- Kataloogipuus liikumine:
  - `cd`; `popd`; `pushd` – aktiivse kataloogi vahetamine
  - **`cd kataloogi_nimi`** - liigutakse vastavasse kataloogi
  - **`cd`** ja `cd ~` - liigutakse kasutaja kodukataloogi
  - **`cd ..`** - liigutakse ülemkataloogi
  - **`pwd`** - kuvatakse antud kataloogi täistee

# UNIXI käsud

- **cat faili\_nimi** - kuvab faili sisu
- **cp fail\_1 fail\_2** - kopeerib faili 1 ja nimetab selle fail 2-ks
- **mv fail\_1 fail\_2** - tõstab ümber või nimetab fail 1 ümber
- **rmdir fail / kataloog** - kustutab faili või kataloogi
- **man käsk** - kuvab manuaali käsu kohta



# Failidele õiguste andmine

- UNIXi süsteemis on kasutajal identifikaator ehk ID
- Kasutaja omakorda kuulub gruppi
- Grupile vastab grupi identifikaator ehk ID
- ID on numbriline info faili atribuutides
- Vajalik, et saaks määrata, kes saavad faili lugeda, kirjutada või käivitada.

# Failidele õiguste andmine

## Omadused:

- r(Read) – võib lugeda – sellele vastab 4
- w(Write) – võib kirjutada – sellele vastab 2
- x-(eXecute) – võib käivitada – sellele vastab 1

Numbreid liites saadakse tulemuseks erinevad õigustehulgad.

- $4+2+1 = 7$  - kõik õigused

# Failidele õiguste andmine

## Kasutajad:

- u(User) – faili omanik
- g(Group) – faili gruppi kuuluv kasutaja
- o(Other) – keegi muu kasutaja süsteemis, erinevas grupis

# Failidele õiguste andmine

- Fail nimega **'info'** kõigile ainult loetavaks:

```
chmod a=r info
```

- Lisatakse omanikule faili kirjutamise ja käivitamise õigus:

```
chmod u+wx info
```

- Soovitakse jätta grupi liikmed lugemisõigusest ilma:

```
chmod g-r info
```

- Tulemusena on faili **'info'** õigused sellised:

```
rwX --- r--
```

# Kestaskriptide kirjutamine

- Kasutada võib suvalist tekstiredaktorit.
- Kestaskript algab reaga, mis näitab tee kasutatavale kestatüübile: `#!/bin/sh`
  - Juhul, kui skripti alguses tee kasutatava shellini puudub, kasutatakse käivitusaeget keskkonda.
- Käivitamisõigused: `chmod 755 faili_nimi.sh`

- Käivitada skript nagu süntaks:

```
bash "faili_nimi"
```

```
sh "faili_nimi"
```

- `./"faili_nimi"`

# Muutujad kesta

- Kesta oluliseks funktsiooniks on keskkonnamuutujad
- Pääsevad ligi kõik kesta poolt käivitatud programmid
- Seal hoitakse palju programmide seadeid
- Süsteemimuutujad on suurte tähtedega, näiteks USER
- Kasutaja poolt defineeritud (*User Defined Variables*) muutujad on väikeste tähtedega
- Defineerimine: `muutuja_nimi=väärtus`

# Muutujad keestas

## Reeglid:

- muutujas võivad olla ASCII tähed ja numbrid ning alakriips
- võrdusmärgi ette ega taha ei tohi panna tühikuid
- suured ja väiksed tähed on erinevad
- NULL muutuja saab defineerida näiteks:  
\$kolmas= või \$kolmas=""
- muutujate defineerimisel ei tohiks kasutada erisümboleid: ?, \* ja muid taolisi märke

# Muutujad keestas

## Süsteemimuutujad:

- USER - Kasutajanimi, millega sisse logisite.
- PATH - Koolonitega eraldatud nimekiri kataloogidest, kust tuleks binaarfaile otsida.
- SHELL - Kasutuses olev shell.
- TERM - Kasutaja terminali nimi.



# Aritmeetiline väärtustamine

Võimaldab teha tehteid keskkonnamuutujate väärtustega

- `expr 1 + 3` - liitmine
- `Expr 10 / 2` - jagamine
- `expr 20 % 3` - jäägi leidmine
- `expr 10 \* 3` - korrutamine
- `echo `expr 6 + 3`` - skripti sees arvutab tulemuse  
oluline `echo` käsu olemasolu

# Ülakomade tähtsus

- Jutumärgid - " " - "tekst" - kõiki märke jutumärkide vahel tõlgendatakse stringina, välja arvatud, märke \ ja \$.
- Ülakomad - ' ' - 'tekst' - kõik ülakomade vahel jääb muutumatuks.
- Tagurpidi ülakomad - ` ` - `tekst` - käsu käivitamiseks teksti sees

# Käsk ECHO

- Käsku echo kasutatakse teksti või väärtuse kuvamiseks ekraanile.

```
echo "Kooriku skriptide kirjutamine on  
tegelikult lihtne!!"
```

- Kuvab:

```
Kooriku skriptide kirjutamine on  
tegelikult lihtne!!
```

# Käsk EXIT

- Unixis ja Linuxis tagastatakse käskude ja kestackriptide käivitamisel kahte sorti väärtusi.
- Vajalik, et jõuda selgusele toimunud protsessi edukuses.
- Kui tagastatakse number null (0), siis on käsk edukas.
- Kui tagastatakse nullist erinev arv (*nonzero*), siis käsk ei ole edukas või oli käsu või on skripti käivitamisviga.
- Seda väärtust nimetatakse väljumisväärtuseks (*Exit Status*).

# Torud (pipes)

Sisendi ja väljundiga manipuleerimiseks saab kasutada torusid:

- `<` – failist loetakse sisendväärtused

```
käsk ("parameeter") < faili_nimi
```

- `>` – tulemus lükatakse faili.

```
who > kes_logis.txt
```

- Toru, kus programm saadab andmeid teisele programmile

```
ls | more
```

# Torud (*pipes*)

- Toru, milles kasutatakse veaväljundit (*standard error*).  
Veaväljundi puhul pannakse süntaksis ette ka 2.

```
käsk ("parameeter") 2> faili_nimi
```

- Toru, mis lisab andmed faili lõppu.

```
käsk ("parameeter") >> faili_nimi
```

- Toru, mis võimaldab ühe protsessi väljundit samaaegselt 'sisse sööta' mitmele muule protsessile. Selleks on programm *tee*.

```
käsk1 | käsk2 | tee faili_nimi
```

# Tingimused

- Vastavalt seatud tingimustele saab programmi edasist käiku suunata.
- Selleks on vajalikud tingimuslaused.
- Arvulisi suurusi võrreldakse järgmiste loogiliste operaatoritega, näiteks:
  - `-eq` - võrdne (inglise keeles equal)
  - `-ne` - mittevõrdne (inglise keeles not equal)

# Tingimused

- IF – tingimus

Kui *If* – tingimus on täidetud, siis vastav käsk täidetakse.

- IF – ELSE – FI – tingimus

Kui *If* – tingimus on tõene, täidetakse esimene käsk, muul juhul teine käsk.

- CASE – tingimus

*Case* – tingimus annab võimaluse valida mitme tegevuse vahel.



# Tingimused

- FOR – tingimus

*For*– tingimus võimaldab tegevust teatud arv kordi korrata.

- WHILE – tingimus

*While* – tingimuse korral korratakse tegevust, kuni tingimus on tõene.

- UNTIL– tingimus

*Until* – tingimuse korral korratakse tegevust, kuni tingimus saab tõeseks.

# Sisend

- Kestaskriptidele saab anda sisendit:
  - Käsurealt
  - Teise programmi väljundist
  - Failist
- Kestaskriptides sisendfailidest ja käsurealt lugemiseks kasutatakse käsku `read`.

# Väljund

Kestaskriptide väljundi saab:

- suunata faili
- anda mõne programmi sisendisse
- kuvada ekraanile kasutades käsku echo

# Käsurea parameetrid

- Kestaskripti käivitamisel saab anda käsurealt ka argumente
- Sisendisse saab anda numbreid ja stringe, näiteks:  
faili nimi, sünniaasta, kasutaja
- Nende väärtused on skriptis muutujates \$1 , \$2 jne.

# Pikemalt leiab abi:

- Bash Guide for Beginners
  - <http://tldp.org/LDP/Bash-Beginners-Guide/html/>
- Advanced Bash-Scripting Guide
  - <http://www.tldp.org/LDP/abs/html/>
- Koorik Bash
  - <http://kuutorvaja.eenet.ee/kasutamine/os/shell/bash.html>
- Mujalt Internetist

# Näide

- Kirjutada skript, mis saab ühe parameetri ja otsustab, kas tegemist on kausta või failiga. Kui on tegemist kaustaga – antakse vastuseks KAUST, kui fail, siis FAIL.

# Ülesande analüüs

- Kontrollida, kas failile antakse sisend;
- kontrollida, kas sisendina esitatud asi on ka tegelikult olemas;
- kontrollida, kas on tegemist faili või kaustaga.

# Kestaskript, mis kontrollib sisendit (fail/kaust)

vt. Introduction to if

```
#!/bin/bash

if [ -a $1 ];
then
  if [ -d $1 ]
  then
    echo "\"$1\" on KAUST."
  fi
  if [ -f $1 ]
  then
    echo "\"$1\" on mingi fail."
  fi
else echo "\"$1\" ei eksisteeri
      ilmselt."
fi
```