

## UNIX-i ajalugu

1969 Ken Thompson Bell Laboratories'st hakkab tegelema UNIX-i loomisega. Kuna teda ükski olemasolevatest programmeerimiskeeltest ei rahuldanud siis tuli luua ka uus programmeerimiskeel, millega hakkas tegelema D.Richie. Keele nimi võeti tähestikust 'C' (eelmine keel millega see mees tegeles, oli 'B').

1970 Saadakse valmis esimese realisatsiooniga (arvutil PDP 7)

1975 Bell Laboratories annab UNIX-i u"likoolidele kasutada.

1977 Antakse välja Unixi ver. 7

1983 UNIX System V, mis ilmub AT&T kaubama"rgi all.

## Arvutisse logimine

Selleks et üldse midagi UNIX-is teha, pead sa süsteemi sisse logima. Kui sa astud mõne vaba arvuti juurde millel töötab UNIX süsteem on seal ees kiri:

login:

Selle järel pead sa sisestama oma kasutajanime (*account*). Järgmiseks ilmub ekraanile text:

password:

Nüüd pead sa sisestama oma parooli. Nüüd ilmub ekraanile kas prompt, mis tähendab et süsteem on edasiisek tööks valmis, või teade :

login incorrect

login:

See tähendab, et te sisestasite oma parooli või kasutajanime valesti. (Võibolla läks mõni täht valesti või jäi vahelt ära) Proovige paar korda uuesti sisestada kasutajanime ja parooli. Ja kui te ikka sisse ei saa siis pöörduge süsteemiadministraatori juurde.

NB! UNIX-is on suured ja väikesed tähed erinevad asjad!

## ls - kataloogi sisu listimine

Süntaks:

```
ls [-adfgilrt] [kataloog]
```

ls väljastab infot iga antud kataloogis oleva faili kohta.

Infot, mida näidatakse, võid sa määrata järgmiste lippudega:

- a - näidata peidetud faile
- g - näidata faili gruppi
- l - pikk listing
- R - näidata ka kõiki alamkatalooge
- t - sorteerida aja, aga mitte nime järgi

Näide:

```
> ls -l
drwx----- 2 argo  users  1024 Feb 22 17:49 commands/
-rw----- 1 argo  users  4794 Feb  2 12:23 failisüsteemid.html
-rw----- 1 argo  users  4760 Feb  2 13:43 fsyst.html
drwx----- 2 argo  users   512 Feb 22 14:20 fsystem/
-rw----- 1 argo  users   648 Feb  2 12:07 history.html
-rw----- 1 argo  users   511 Feb 20 18:13 index.html
-rw----- 1 argo  users  4106 Feb  7 08:33 midagi.veel
```

## Juurdepääsu õiguste vaatamine

Faili juurdepääsu õiguste vaatamiseks on olemas järgmine käsk:

```
ls -l failinimi
```

See käsk valjastab iga faili kohta ühe rea. Näiteks:

```
-rwxr-xr-x 1 erpl08 1995 Feb 22 15:51 minu.html
```

Esimene element `-rwxr-xr-x` näitab juurdepääsu õigusi. Järgmine element näitab, mitu linki on sellele failile, siis tuleb faili omaniku kasutajanimi, faili suurus ja kellaaeg ning kuupäev, millal faili viimati muudeti, ja lõpuks ka faili nimi.

### Õigused:

On olemas kolme sorti õigusi. Õigused käituvad kataloogide ja failide puhul kergelt erinevatl.

r - lugemis õigus.

Faili võib lugeda. (Trükkida välja faili sisu)

Saab näha kataloogis olevate failide listingut.

w - kirjutamis õigus.

Faili võib kirjutada.

Kataloogis võib tekitada, ümbernimetada ja kustutada faile.

x - käivitamis õigus.

Faili võib käima lasta.

Kataloogis olevaid faile saab lugeda, käivitada ja neisse võib kirjutada.

Kõiki neid õigusi saab eraldi panna järgmist tüüpi omanikele:

u - Faili omanik.

g - Faili omaniku gruppi kuuluvad kasutajad.

o - Koik teised kasutajad.

Õigused antakse 9-tähelise stringina:

kasutaja	grupp	teised
<b>r w x</b>	<b>r w x</b>	<b>r w x</b>

Näiteid:

1. -rw----- Faili omanik võib seda faili lugeda/kirjutada/kustutada. Keegi teine seda faili isegi ei näe!
2. -rw-r--r-- Faili omanik võib jällegi lugeda ja kirjutada. Koik teised, nii gruppi liikmed kui kõik ülejäänud liikmed võivad ainult faili lugeda.
3. -rwx--x--x Faili omanik võib selle failiga kõike teha. Teised võivad seda faili ainult käivitada.
4. drwxr--x--- See on kataloog ( d ). Kataloogi omanikul on olemas kõik õigused. Grupil on lugemise ja kirjutamise õigused, ning ülejäänutel pole mingeid õigusi selle kataloogi suhtes.

## Standardsisend ja -väljund

UNIX peab kõiki seadmeid, mis on ühendatud süsteemi külge, failideks. Vaikimisi kasutatakse sinu klaviatuuri standardsisendina, millest informatsiooni sisse loetakse, ja sinu terminali satandardväljundina, millesse informatsiooni saadetakse.

Standard sisendit ja väljundit on võimalik ümber suunata:

### Standard sisendi ümber suunamine

Andes käsu taha märgi < millele järgneb faili nimi kust peab tulema standard sisend.

Näide: mail mart <memo

## Standard väljundi ümbersuunamine

Andes käsu taha märgi `>` millele järgneb faili nimi kuhu peab minema standard sisend. Kui väljundiks on fail, mida veel olemas ei ole, siis ta luuakse automaatselt.

Näide: `grep Tarmo /etc/passwd > tulem`

## Standard väljundiga faili lõppu lisamine

Andes käsu taha kahekordselt märgi `>>` millele järgneb faili nimi mille otsa tuleb lisad standard väljund.

Näide: `cat osa1 osa2 >> text`

## Käskude kokku ühendamine

Unix võimaldab toru kasutades kahte ja enamat käsku kokku ühendada. Toru võtab ühe käsu standard väljundi ja suunab selle teise käsu standard sisendiks:

`käsk1 | käsk2 | käsk3`

Märki `|` kasutatakse toru tähisena, mis ühendab omavahel kaks järjestikust käsku.

Näiteid:

`who | wc -l`

See käsk ütleb, mitu kasutajat on parajasti süsteemi sisse logitud.

`ps -aux | grep argo | more`

See käsk toob ekraanile lehekülgede kaupa kasutaja argo protsessid.

## **cp - Kopeerimine**

Süntaks:

```
cp fail1 fail2
```

```
cp fail1 ... failn kataloog1
```

```
cp -r kataloog1 kataloog2
```

Kui fail, milleks kopeeritakse, on juba olemas, siis ta kirjutatakse üle.

Näiteid:

faili kopeerimine:

```
> cp failike failike.vana
```

```
> ls failike*
```

```
failike failike.vana
```

failide kopeerimine kataloogi:

```
> cp ~/src/* /tmp
```

kataloogi kopeerimine:

```
> ls -R ~/bkup
```

```
/usr/example/fred/bkup not found
```

```
> cp -r ~/src ~/bkup
```

```
> ls -R ~/bkup
```

```
x.c y.c z.sh
```

## **mv - failide liigutamine ja ümbernimetamine**

Süntaks:

mv fail1 fail2	//nimetab faili fail1 failiks fail2
mv fail1 kataloog1	//tõstab faili fail1 kataloogi kataloog1
mv kataloog1 kataloog2	//nimetab kataloogi kataloog1 // kataloogiks kataloog2

## **rm , rmdir - kataloogide ja failide kustutamine**

Süntaks:

rm [-i] failinimi...	//kustutab failid
rm -r [-i] katalooginimi...	//kustutab kataloogi koos temas // asuvate alamkataloogide ja failidega
rmdir katalooginimi	//kustutab tühja kataloogi

**NB!** *Undo*'d ei ole!

Lisades käsule rm lipu -i , küsitakse enne iga faili kustutamist, kas see ikka kustutada.

## **file - faili tüübi kindlaks tegemine**

Süntaks:

```
file failinimi...
```

See käsk püüab kindlaks teha, mis tüüpi on mingi fail.

Näide:

```
> file *
a.out:      ELF 32-bit MSB executable SPARC Version 1,
            dynamically linked, not stripped
bla:        directory
win.c:      c program text
```

## **cat - Failide ühendamine ja kuvamine**

Süntaks:

```
cat failinimi...
```

cat loeb järjest etteantud failidest infot ja väljastab selle standard väljundisse.

Näide: cat minufail

Käsk trükib faili minufail ekraanile.

Näide: cat fail1 fail2 > fail3

Kõigepealt väljastatakse fail1 faili fail3 ja siis fail fail2 faili fail3 otsa, saades niimodi fail1 ja fail2 ühendatuna faili fail3.

Kui ei ole antud ühtegi argumenti siis cat loeb infot standardsisendist.



## **more - failide kuvamine**

Süntaks:

```
more failinimi
```

more väljastab ekraanile faili failinimi sisu lehekülgede kaupa. Kui teksti juba näidatakse, on võimalik more-ile edastada järgmisi käske klahvivajutustega:

tühik - lehekülg edasi

d - pool lehekülge edasi

b - lehekülg tagasi

q, Ctrl-C - vaatamise katkestamine

## **find - failide otsimine**

Süntaks:

```
find kataloog -name failinimi -print
```

find otsib rekursiivselt etteantud kataloogi alamkataloogidest failinimi-ga määratud faile.

Lipp *-print* näitab et tulemused tuleb väljastada ekraanile. Selle asemel võib kasutada ka muid lippe. Täpsema info saamiseks uuri

*man find*

Ette antavas failinimes võib kasutada ka metamärke.

**lpr - failide trükkimine**  
**grep - failist regulaaravaldise otsimine**  
**sort - failide sisu sorteerimine**  
**ln - kataloogide ja failide linkimine**

## **cd - Aktiivse kataloogi muutmine**

Süntaks:

`cd [ kataloog ]`

Kataloogi kataloog muutmine aktiivseks kataloogiks.

Kui kataloog-i ei ole öeldud, kasutatakse käsu parameetrina keskkonnamuutuja **\$HOME** väärtust.

Kataloog-i võib ette anda nii relatiivsena kui täielikuna.

## **pwd - aktiivse kataloogi nimi**

Süntaks:

`pwd`

Annab aktiivse kataloogi täieliku tee

## **mkdir - kataloogi tegemine**

# PROTSESSID

## Protsess esiplaanil

Kui sa sisestad mingi käsu, siis *shell* käivitab selle programmi oma alamprotsessina.

Ülmprotsess(sinu shell) ootab kuni, alamprotsess jõuab oma tööga lõpule ja sureb. Seni ei saa sa sisestada ühtegi järgmist käsku - alamprotsess on esiplaani protsessiks. Kui alamprotsess lõpetab oma töö aktiveeritakse jällegi ülemprotsess.

### 1.Eesplaanil töötava protsessi katkestamine

Vajutades klahvikombinatsiooni:

Ctrl-C

on võimalik katkestada eesplaani protsessi. Pärast seda klahvivajutust enam seda protsessi ei eksisteeri. Tööjärg antakse tagasi ülemprotsessile.

### 2.Eesplaani protsessi peatamine

Eesplaani töö peatamiseks vajuta klahve:

Ctrl-Z

Bourne shellis võid sa nüüd selle protsessi tappa. Teised shellid võimaldavad aga ka seda protsessi hiljem jälle esiplaanile tuua, või siis asetada see töö taustale.

### 3. Eesplaanil töötava protsessi panemine taustale

Kõigepealt tuleb eesplaani töö peatada ja seejärel asetada ta käsuga **bg** taustale.

Näide:

```
lpr part[1-8].ps  
^Z  
Suspended  
bg  
[1] lpr part1.ps &  
jobs  
[1] Running lpr part1.ps
```

Kõigepealt pannakse arvuti mõningaid faile välja trükkima. Siis see töö peatatakse ja viiakse taustale. Viimase käsuga vaadatakse millised tööd on parajasti käimas.

## Taustaprotsessid

Lisades käsu lõppu märgi **&**, on võimalik selle käsu töö panna taustale.

```
find . -name '*.txt' -print > tekstid &  
[I] 2473
```

Shell paneb töö (käsu) alamprotsessina käima ja tagastab tema töö numbri ([n]) ja PID-i (Process IDentifier) numbri. Shelli prompt ilmub tagasi ekraanile, ja sa võid jätkata oma tööd sisestades järgmise käsu. Antud käsu täitmist jätkatakse taustal.

Kui mõni taustaprotsess kinni jääb siis tuleb ta hävitada tema tapmisega. Kinni jooksevad kõik sellised protsessid, mis tahavad kasutajalt midagi standard sisendist lugeda (juhul kui neile ei ole antud mõni muu sisend standardsisendiks).

Käsuga **jobs** on võimalik näha millised taustaprotsessid on parajasti käimas.

Käsuga **fg** on võimalik tuua protsessi taustalt esiplaanile.

## **ps - Protsesside vaatamine**

Süntaks

ps [ -lipud ]

Ilma parameetriteta see käsk annab nimekirja sinu protsessidest, mis on seotud antud terminaliga. Lisaks muule infole annab see käsk ka protsessi numbri, mida on vaja sel juhtul, kui soovite mõnda protsessi tappa.

Antud käsu lipud sõltuvad sellest, millist UNIX-i süsteemi sa kasutad. Täpsema info saamiseks pöördu man-i poole.

## **kill - Protsessi tapmine**

# Failide tüübid UNIX-is

## Tavalised(Ordinary) failid

Seda tüüpi kasutatakse tavalise informatsiooni hoidmiseks, näiteks mingi tekst või mõni pilt. Seda tüüpi failidega tulebki tavaliselt kõige rohkem kokku puutuda.

## Kataloogid(Directories)

Kataloog on fail mis sisaldab teisi faile ja katalooge. s.t. See on fail milles on viidad teistele failidele. Siin on kirjas iga antud kataloogis asetseva faili nimi ja tema inode number.

### *Inode number:*

Iga faili nimega on ühendatud ainulaadne number, mida nimetatakse inode numbriks. Selle numbri järgi leitakse sissekanne inode tabelist, mis annab informatsiooni faili tüübi, suuruse, asukoha ja faili omaniku kohta.

Inode number viitab ühele kindlale failile. Inode numbrile aga võib viidata mitu faili nime. Ühendust faili nime ja inode numbri vahel nimetatakse lingiks.

## Spetsiaalsed(Special) failid

Need on failid, mis viitavad füüsilistele väljunditele ja sisenditele. Näiteks printerid, floppyd, terminalid. Esialgu võib küll tunduda vahe imelik, kui mõelda mingist ekraanist või printerist kui failist. See aga annab väga mugava võimaluse käsurealt info suunamiseks.

Näiteks:

```
cat scream.au > /dev/audio
```

See käsk suunab faili scream.au sisu audio väljundisse.

Spetsiaalsed failid asuvad UNIX-is kataloogis /dev .

UNIX-is on olemas prügikasti väljund /dev/null. Sinna saadetud info hävineb.

## **Torud(Pipes)**

UNIX võimaldab ühendada käske kokku , nii et ühe käsu väljund läheb otse teise faili sisendiks. Toru käitub ajutise failina, mis eksisteerib ainult niikaua kui on vaja hoida andmeid, et järgmine käsk suudaks need sisse lugeda.

## **Failisüsteemi struktuur**

UNIX-i failisüsteem on organiseeritud hierarhiliselt kataloogidest.

Alustatakse kataloogist **root** mida tähistakse märgiga /

Failisüsteemi võib endale ette kujutada puuna, kuid on ka link'id!

## **UNIX-i süsteemsed kataloogid**

UNIX-is on root kataloogis olemas kataloogid, mis sisaldavad kindlat tüüpi informatsiooni.

### **/bin**

See kataloog sisaldab käivitatavaid faile mida UNIX-is loetakse käskudeks.

### **/dev**

See kataloog sisaldab spetsiaalseid faile, mis vastavad füüsilistele väljunditele ja sisenditele.

**/etc**

Selles kataloogis asuvad kasud mis on vajalikud süsteemi administreerimiseks.

**/lib**

Siin kataloogis asuvad failid mida kasutavad erinevad programmid oma töös.

**/tmp**

See on kataloog kuhu luuaks ajutisi faile.

**/usr**

Siin kataloogis asuvad ülejäänud kataloogid.(manuaalid, mängud...)