

# Salvestusseadmed ja kettahaldus

# Salvestusseadmed

- Lindid
- Kettad
- Väikmäluuseadmed (*flash memory*)

# Lindid

- Lindiseadmeid kasutati esmalt salvestusseadmena
  - Lindilt andmete kätte saamine on aeglane
- Praegusel ajal kasutatakse magnetlinte suurte andmemahutude varundamisel.
  - Lindile mahub palju andmeid (sadu GB)
  - Lindilt lugemine ja lindile kirjutamine on kiire
  - Õige koha leidmine võtab aega
- **NB!** Kõvakettamahutude oluline kasv ning otsejuurdepääs tähendab seda, et linte kasutatakse varundamiseks üha vähem.

# Kettad

- Peamine andmete hoidmise vahend tänapäeval
  - Oma ehituse tõttu on võimalik kiiresti suvalise kettapiirkonna poole pöörduda.
  - Adresseeritud ühedimensioonilised loogiliste plokkide kogumid, milles ühe loogilise ploki suurus on väikseim andmeühik.
- Magnetkettad.
- Optilised kettad.

# Välkmäluseadmed

- Liik püsिमälused, millel andmed püsivad ka peale toite väljalülitamist
  - Mälupulgad jms,
  - SSD kõvakettad.

# Partitsioon

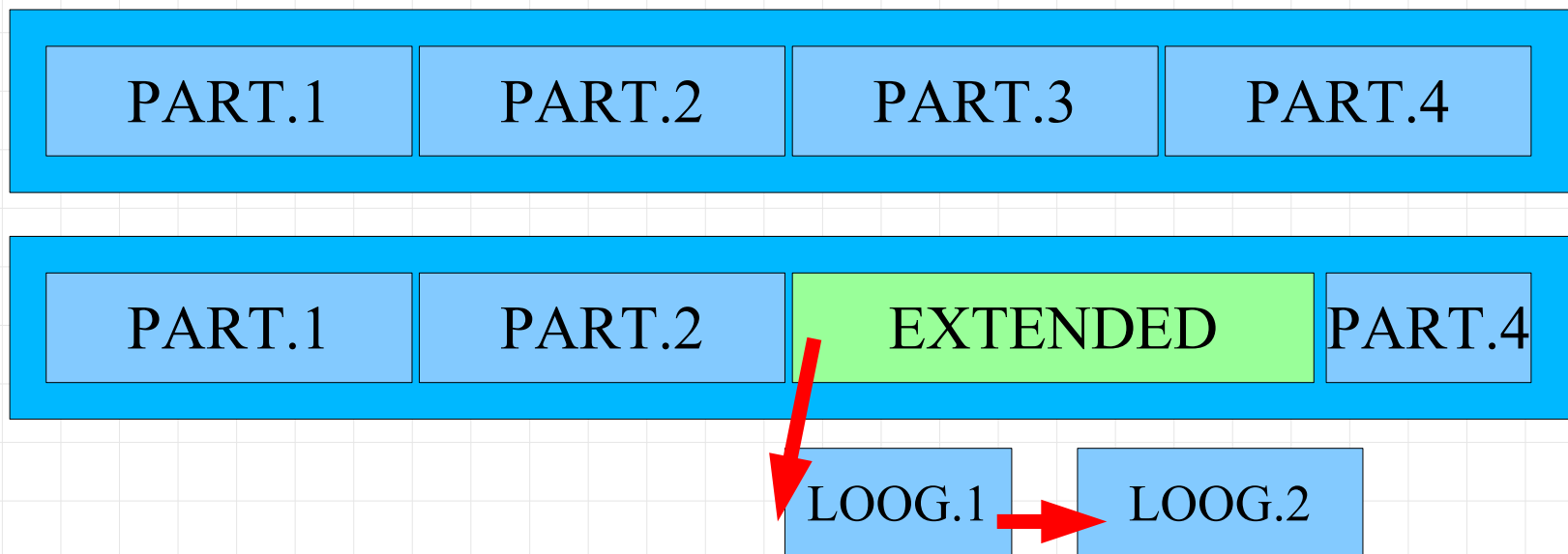
- Selleks, et andmekandjatel andmeid paremini hallata kasutatakse partitsioneerimist.
- Kõvakettad, CD-plaadid, ühe füüsilise kettana paistvad kettamassiivid (RAID)
- Partitsioon on salvestusseadme salvestatava pinna loogiline jaotis
- Partitsioone saab omakorda jagada väiksemateks loogilisteks üksusteks kasutades selleks LVMi
  - See eeldab, et juba partitsioneerimisel on kasutatud LVM võimalusi.

# Piirangud

- Iga partitsioon koosneb ainult külgnevast kettaruumist;
- Iga failisüsteem on piiratud ühe partitsiooniga;
- Standardsed võimalused partitsioonide suuruste dünaamiliseks muutmiseks üldjuhul puuduvad
- Uuemad operatsioonisüsteemid lubavad teatud partitsioone **suurendada** – juhul kui füüsiliselt sellele järgneb vaba ruumi.

# MS-DOS-tüüpi partitsioonitabel

- Klassikaline MSDOS partitsioonitabel lubas luua **neli** partitsiooni.
  - Piirang, kui palju partitsioone luua saab (sõltub kettatüübist: IDE/SCSI).
- Praegu lubatakse rohkem partitsioone lubades loogilisi partitsioone.





# MS-DOS-tüüpi partitsioonitabel

- Esmane partitsioon
  - Kuni 4 tk kettal.
  - Info nende kohta on MBR alas.
- Laiendatud partitsioon
  - Kuni 1 tk kettal.
  - Üks esmane partitsioon võib olla laiendatud partitsioon.
- Loogiline partitsioon
  - Laiendatud partitsiooni sees.
  - Järjest.
  - Info on loogiliste partitsioonide juures (algus-lõpp).

# GPT partitsioonitabel

- Tuntud ka kui GUID partition table.
  - Globally unique identifier (GUID).
  - Iga partitsiooni kohta on oma unikaalne tunnus.
- Info partitsioonide kohta on dubleeritud ketta alguses ja lõpus.
- Enamus praeguseid operatsioonisüsteeme toetab seda (2010 ja uuemad).
- Saab teha palju partitsioone (siiski mitte lõputult).
- UEFI süsteemis kasutatakse.
  - Siiski ka vanemates BIOS süsteemides, kui on selleks vajadus (rohkem esmaseid partitsioone jms).

# Köidete haldus (UNIX tüüpi OS)

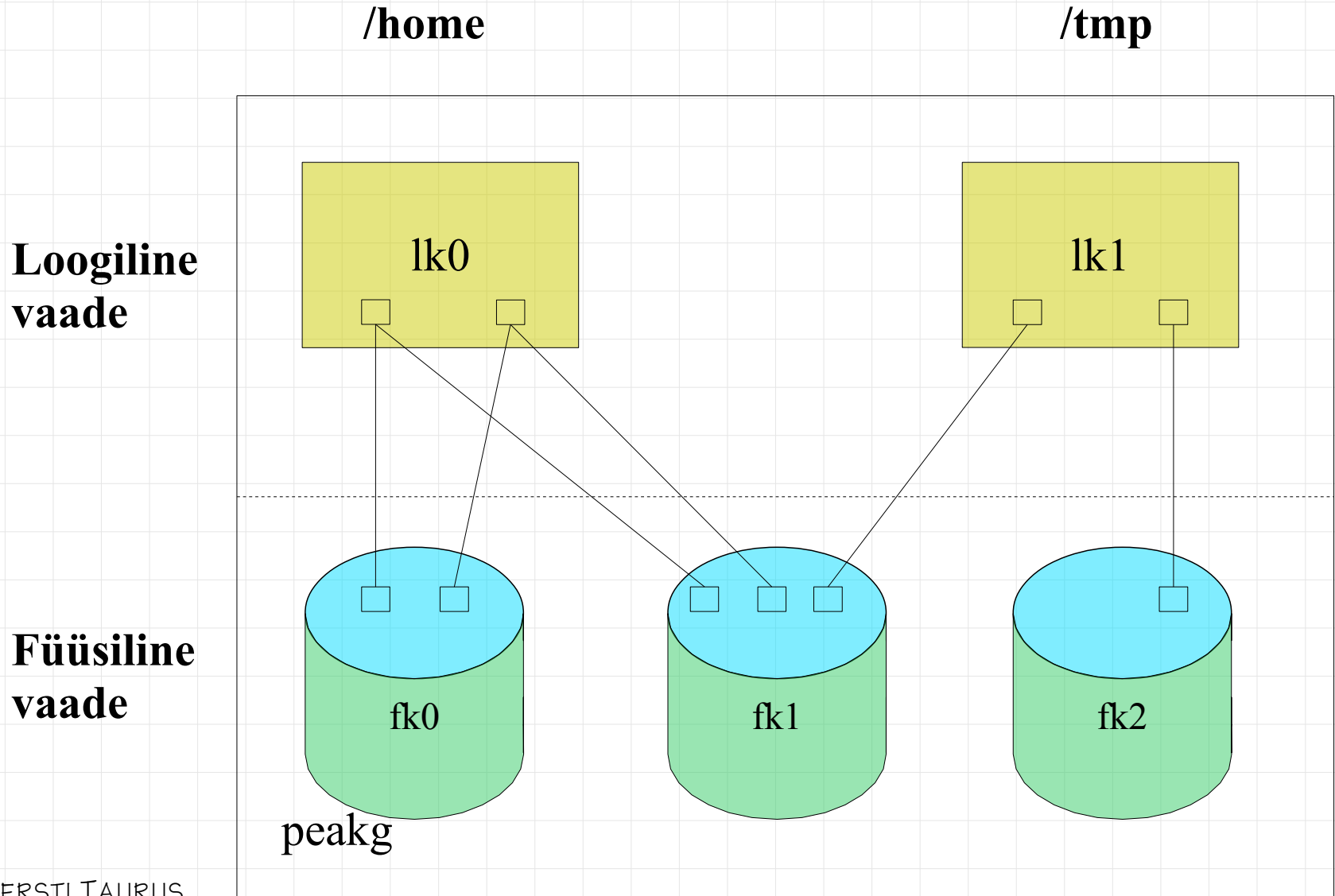
## *(Logical Volume Manager - LVM)*

- On vahekihiks failisüsteemide ja füüsilise ketta partitsioonide vahel
  - Paindlikud loogilised köited
  - Dünaamiline haldus
  - Hea käideldavus
  - Jõudlus
- Kettaruum on jagatud köitegruppideks
  - Kettaruumi lõppemise korral lisatakse füüsiline ketas ja suurendatakse olemasolevat loogilist köidet.

# LVM

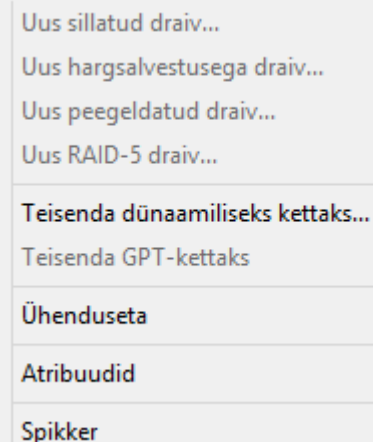
- Köitegrupp on füüsiliste ja loogiliste köidete grupp.
- Füüsilised köited:
  - Kõvakettad, RAID, partitsioonid
  - Kõik füüsilised köited on jagatud sama suurusega tükkideks (näiteks 16 MB)
- Loogiline köide
  - Nähtav kui plokkseade (ketas)
  - Võib sisaldada failisüsteemi

# Köidete haldus: näide



# Dünaamiline/elementaarne ketas

- *Elementaarne ketas (Basic Disk)* võimaldab traditsioonilist partitsioonide kasutamist (MS Dos partitsioonitabel).
- *Dynamic Disk*: kasutab dünaamilisi köiteid (*volumes*)
  - *Simple volumes, spanned volumes, striped volumes, mirrored volumes ja RAID-5 volumes*



Uus sillatud draiv...  
Uus hargsalvestusega draiv...  
Uus peegeldatud draiv...  
Uus RAID-5 draiv...  
Teisenda dünaamiliseks kettaks...  
Teisenda GPT-kettaks  
Ühenduseta  
Atribuudid  
Spikker

# *Dynamic Volumes*

- *Simple volume* on tervikuna ühel kettal.
- *Spanned volume* on mitmel kettal (max 32 ketast)
- *Striped volume* andmed on vööditud mitmele kettale (RAID-0).
- *Mirrored volume* andmed on peegeldatud mitmele kettale (RAID-1)
- *RAID-5 volume*. Seda ei saa peegeldada ega laiendada.
- *System volume* ja *Boot volume*.

# RAID (*Redundant Array of Independent Disks*)

- Sõltumatute ketaste liiasmassiiv
- Mitmest füüsilisest kettast koosnev üks loogiline salvestusseade
  - Töökindlus kasvab (veakindlus, tõrkekindlus)
  - kiirus/maht kasvab
- Samu andmeid kirjutatakse erinevatesse kohtadesse/andmeid kirjutatakse korruga mitmetele ketastele
- Erinevad RAID tasemed – ketaste liitmine ja kasutamine võimaldab täita erinevaid eesmärke



# Probleemidest andmete ja andmekandjatega

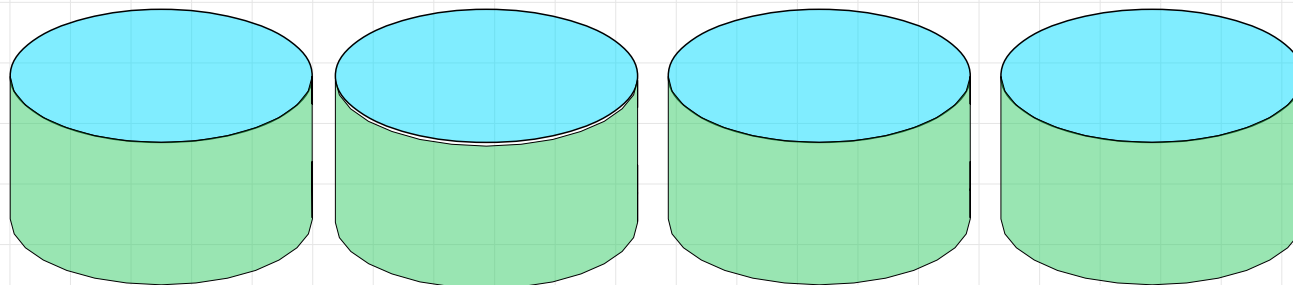
- Ketta tõrge (*disk failure*) – seadmelt enam ei saa andmeid lugeda
- Ketta viga (*disk error*) – seadme mingi osa andmed on vigased/ei saa lugeda (tarkvaralisel või ka riistvaralisel põhjusel)

# Hargsalvestus ja vöötimine (*striping*)

- Selline salvestussüsteem, kus andmed (fail) jaotatakse mitme kettaseadme peale.
- Bititaseme vöötimine
  - Andmeid kirjutatakse ketastele bittide kaupa – ühe baidi sisu on mitmel erineval füüsilisel kettal.
- Plokkasemel vöötimine
  - Andmeid kirjutatakse ketastele plokkide kaupa.

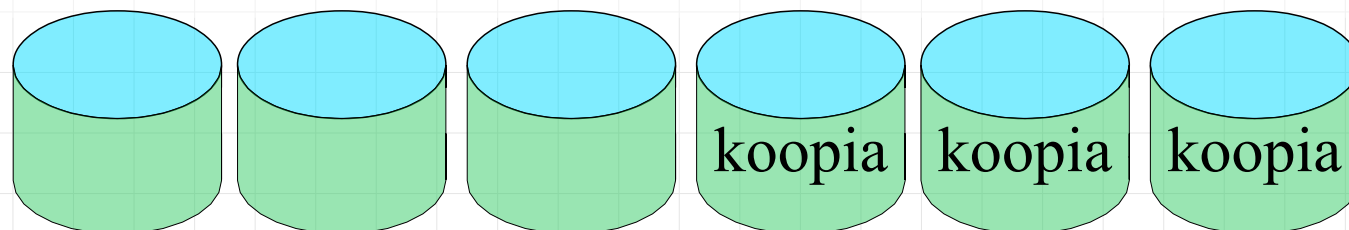
# RAID 0 (*striping*)

- Kettad jaotatakse (plokkides) vöötideks, andmete liiasust pole.
- Ketastelt loetakse ja neile kirjutatakse vaheldumisi
  - Jõudlus paraneb
  - Veakindlus ja tõrkekindlus EI suurene (vs üks ketas)



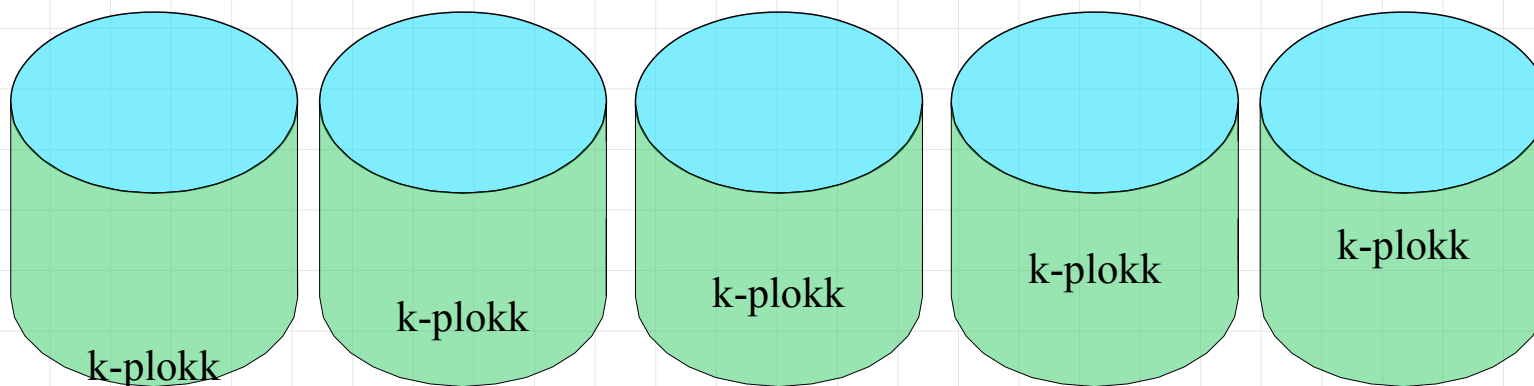
# RAID 1 (*mirror*) - peegel

- Kasutuses on kaks või enam ketast
- Andmete kirjutamine käib neile samaaegselt
  - Vöötideks jagamist ei ole
  - Paraneb lugemiskiirus (saab lugeda kaht ketast samaaegselt)
  - Kirjutamise kiirus ei parane
  - Kettaseadme tõrkekindlus kasvab



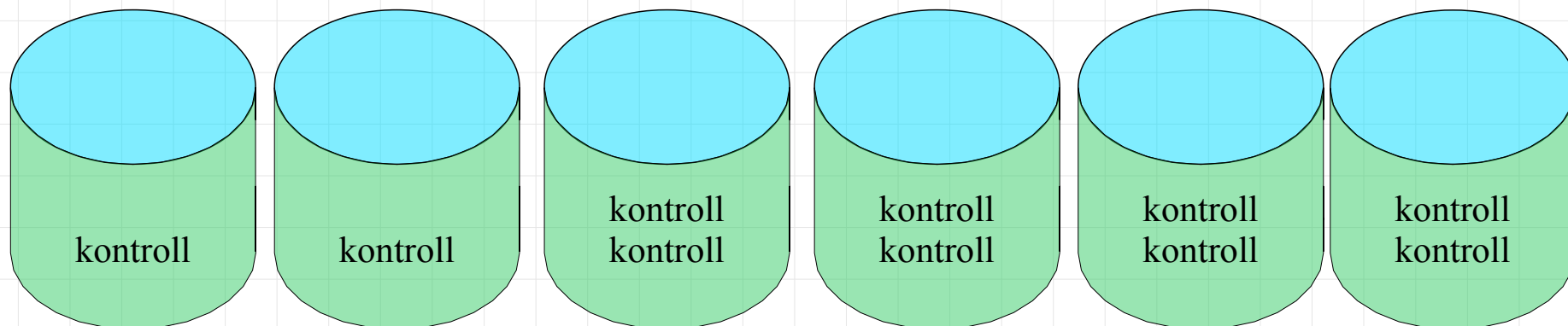
# RAID 5

- Plokitasemel vöötimine (*striping*) koos jagatud paarsuskontrolliga.
  - Kui RAID 4 hoiab paarsuskontrolli ainult ühel kettal, siis siin on koormus hajutatud. Kirjutamise kiirus paraneb.
  - Vajab oma tööks vähemalt kolme (soovitavalt viite) kettaseadet
  - Elab üle ühe kõvaketta väljalangemise süsteemist.



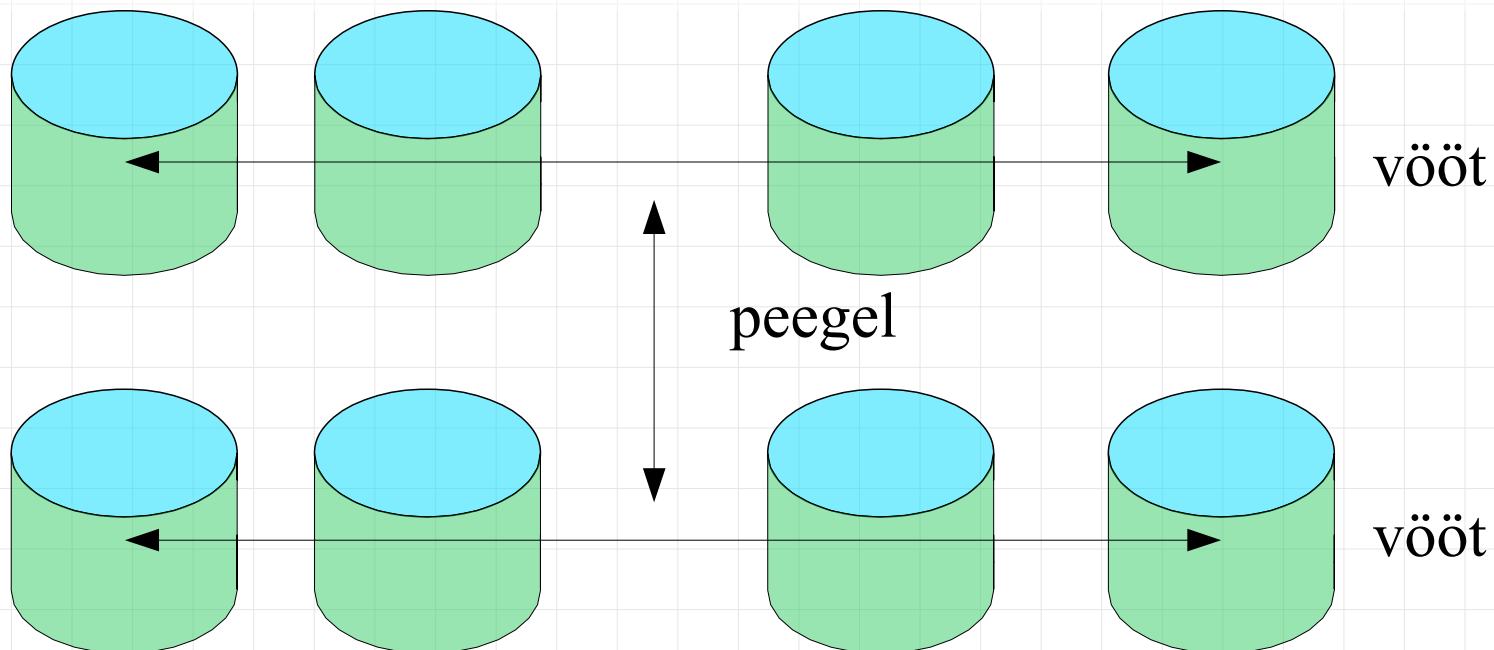
# RAID 6

- Sarnaneb RAID 5-le, kuid sisaldab ka teist paarsusskeemi, mis on jaotatud üle erinevate kettaseadete.
  - Tugev veakindlus – tagatakse, et kettal bitid omapäi ei muutu.
  - Kettaseadmete tõrkekindlus – elab üle kahe ketta väljalangemise süsteemist.



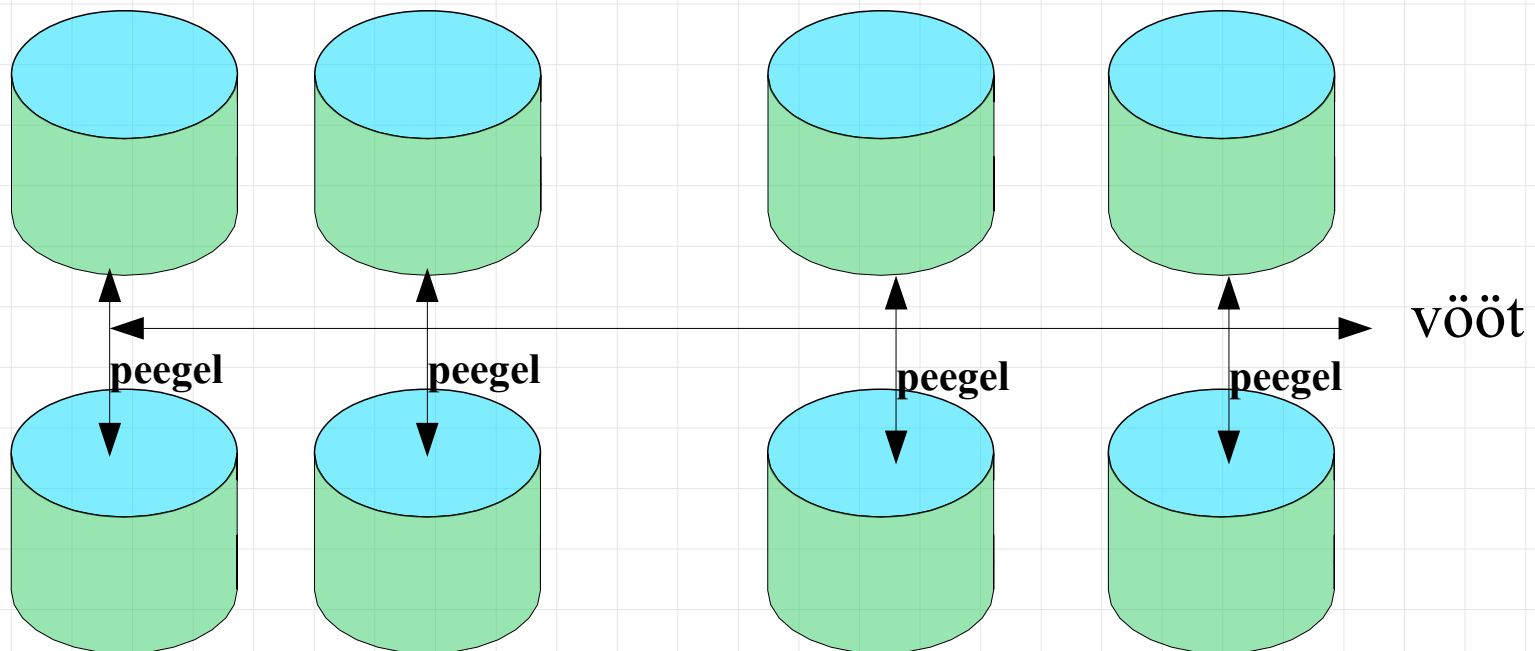
# RAID 0+1

- RAID 0 ja RAID 1 kombinatsioon. (vöödid peegeldatakse)
  - Jõudlus paraneb aga on kallim



# RAID 1+0

- Peegeldatud kettad vööditakse





# Failisüsteem

- Faili mõiste
- Pöördusmeetodid
- Kaustastruktuurid
- Failisüsteemi ühendamine
- Failide jagamine
- Kaitse

# Faili mõiste

- välismälu salvestatud seotud infokogum
- näiteks:
  - andmed
    - numbrilised
    - sümbolid
    - binaarkood
  - programmid

# Faili struktuur

- pole – lihtsalt sõnade või baitide jada
- Lihtne kirjestrüktuur
  - read
    - fikseeritud pikkusega
    - erineva pikkusega
- Keerukamad struktuurid
  - vormindatud dokumendid (näiteks OLE objektivoog)
  - segmentidest koosnev käivitav fail

# Faili struktuur

- Keerulisemaid formaate saab teha lihtsamate abil
- Kes otsustab formaadi üle:
  - operatsioonisüsteem
  - programm

# Faili atribuudid

- 1 Nimi
  - 2 Identifikaator
  - 3 Tüüp – enamustes OS'ides
  - 4 Asukoht – kuskohas fail kettal asub
  - 5 Suurus (maht) – faili pikkus
  - 6 Kaitse – juurdepääsuõigused
  - 7 Kuupäev, kellaaeg ja kasutajainfo
- Infot failide kohta hoitakse kettal kaustastruktuurides

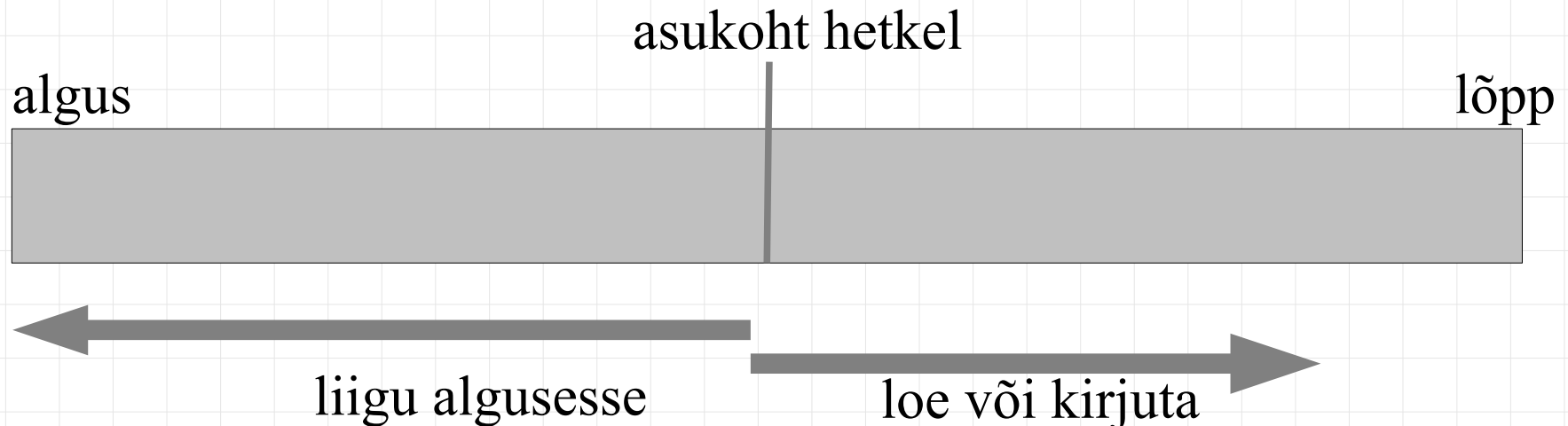
# Operatsioonid failidega

- 1 Loomine
  - 2 Kirjutamine
  - 3 Lugemine
  - 4 Positsioneerimine failis (seek)
  - 5 Kustutamine
  - 6 Kärpimine (pikkuse muutmine)
  - 7 Avamine ja sulgemine
- Kõrgema taseme operatsioonid on tehtavad nende kaudu

Faili tüüp	Laiendid
käivitatav	exe, com, bin
objektifail	obj, o
lähtekood	c, cc, java, pas, asm, a
pakkfail	bat, cmd, sh
tekstiredaktor	txt, tex, rtf, doc, sxw, odt
teek	lib, a, so, dll
arhiiv	zip, tar

# Faili pöördusmeetodid

- järjestikpöördus (samm-sammult lõpu poole – linnid)
  - loe järgmine (*read next*)
  - kirjuta järgmine (*write next*)
  - liigu tagasi faili algusesse (*rewind*)





# Faili pöördusmeetodid

- otsepöördus – ketta mudelil
  - fail koosneb fikseeritud pikkusega nummerdatud kirjetest, mis lubab neid kiiresti lugeda/kirjutada suvalises järjekorras

read n

write n

position to n

    read next

    write next

rewrite n

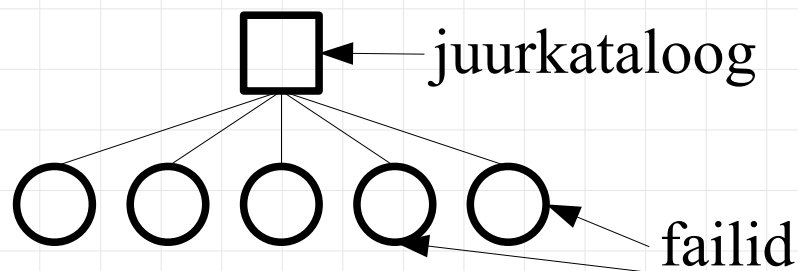
- n = ploki number (faili algusest)

# Pöördusmeetodid

- indekseerimine – otsepöördusmeetodi põhjal
  - faili poole pöördumisel otsitakse kõigepealt indeksist kirje asukoht;
  - sealt liigutakse otse vastavale asukohale.

# Kaustad

- faile on vaja kettal organiseerida
  - Sektsioonid ehk partitsioonid (*partition*)
  - kaustad (*directory*)
- lihtsamal juhul on selleks iga seadme alguses kaust selle seadme failide kohta



# Operatsioonid kataloogidega

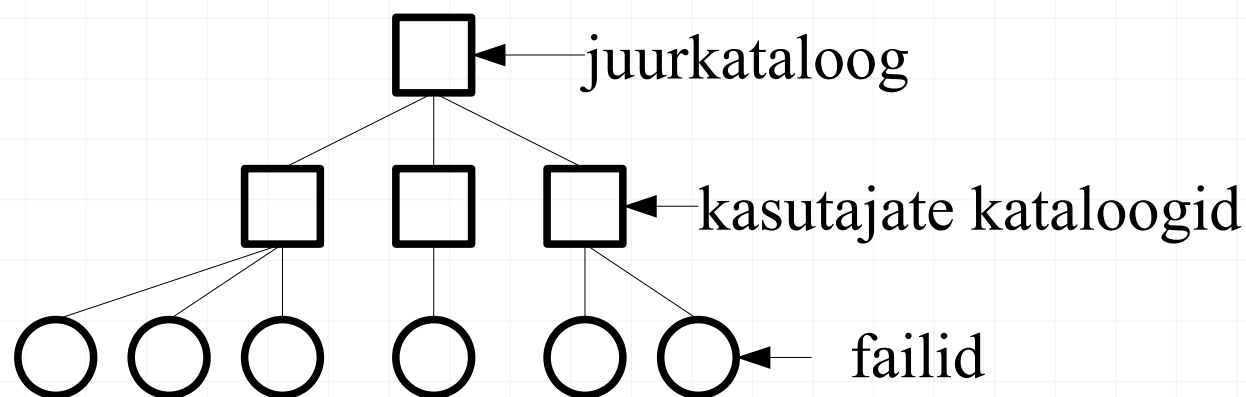
- 1 faili otsimine nime järgi
- 2 faili loomine
- 3 faili kustutamine
- 4 kausta sisu kuvamine
- 5 faili ümber nimetamine
- 6 failisüsteemi läbimine

# Kataloogisüsteemi organiseerimine

- Ühest kataloogist jääb väheks
- mitu kasutajat võivad tahta kasutada sama failinime
- sama fail mitme erineva nime all
- soov sarnaste omadustega faile grupeerida

# Kahetasemeline kataloogistruktuur

- Igal kasutajal on oma kataloog, mis on ära määratud vastava teega (*path*)
- kõrgemal tasemel on metakataloog, kus on viited kasutajate kataloogidele

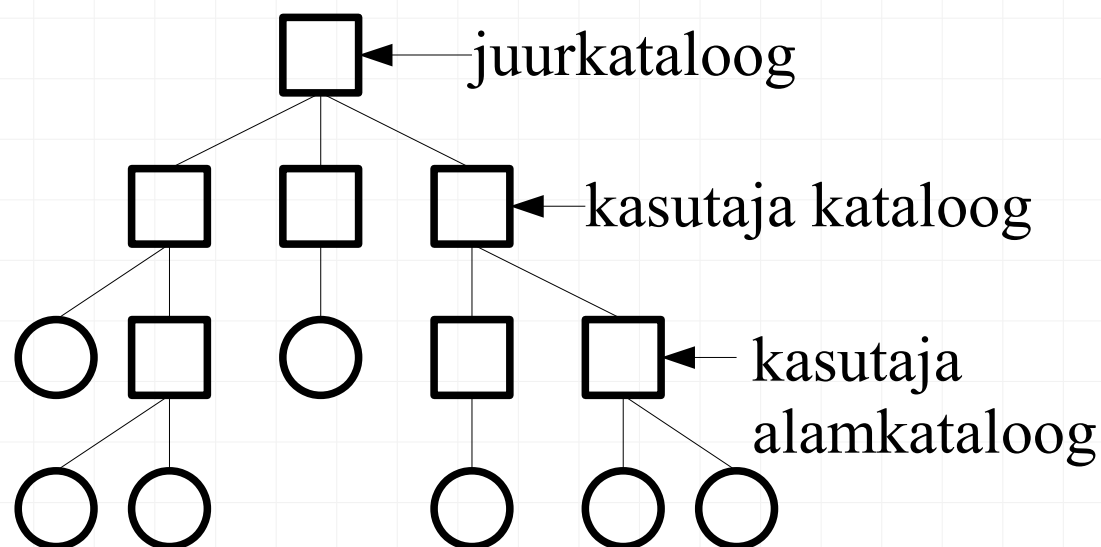


# Kahetasemeline kaustastruktuur

- Tekivad mõned küsimused:
- Kas kasutajad näevad ka teiste kasutajate kaustu?
  - Kui näeb – kuidas sinna pöörduda?
- Kus hoitakse süsteemseid faile?

# Puustruktuuriga kataloogid

- Otsimisvõimalused paranevad;
- mitu kasutajat saavad kasutada sama faili;
- töökaust (hetkel aktiivne kaust)





# Puustruktuuriga kaustad

- Kaust on eriline fail, mille sees võivad olla uued kaustakirjed.
- Jooksva kataloogi mõiste.
- Kataloogidele saab viidata kas absoluutse või suhtelise tee abil (*path*).
- Küsimus kustutamisel (kui leidub alamkaustu): kustutada rekursiivselt kõik või jätta üldse kustutamata.

# Tsükliteta graafid

- Sarnane puustruktuuriga kataloogidele; saab lisaks:
  - viidata samale failile/kaustale mitmest erinevast kaustast;
  - lubab faile/kaustu jagada mitme kasutaja vahel.
- Link – meetod alias-nimede loomiseks
  - Nimeviit (*symbolic link, symlink*)
  - Viit (*hard link*)
- Mis saab viidatava objekti kustutamisel?
  - Tagasiviidad linkide kustutamiseks

# Üldine graaf

- Viidad võivad tekitada kataloogistruktuuri tsükleid.
- Seega võib failisüsteemi läbimisel sattuda tsüklisse.
- Graafist tsüklite leidmise algoritmid on kulukad, arvestada tuleb igal sammul kettalt lugemisega.
- Lahendus: mitte lubada viiteid kataloogidele.
- Nimeviitade korral on tsüklid tavaliselt siiski lubatud:
  - failisüsteemi läbimisel piiratakse nimeviitade läbimise sügavust.

# Näide

- Nimeviida tegemine UNIXis võib Windowsi segadusse ajada:
- `kasutaja@math:~/yks/kaks> ln -s ../yks yks`
- Windowsis paistab see selge kataloogina – kustutamisel kustutatakse ära tegelik kataloog ja tema sisu: (häda tekib kui link peaks juhuslikult juurkataloogile viitama – kogu koduketas koheselt tühi)
- Kustutada selliseid linke UNIXis

# Failisüsteemi ühendamine

- Monteerimine, ühendamine (*mounting*).
- Failisüsteem tuleb enne kasutamist ühendada, et ta nimeruumis nähtavaks saaks.
- Failisüsteem ühendatakse mingisse kindlasse punkti olemasolevas nimeruumis (enamasti on selleks tühi kaust).
- Mis juhtub, kui kaust ei ole tühi?
  - I Veateade
  - II Peidetakse vana sisu kuni lahtiühendamiseni
  - III Andmed seal taga lähevad kaotsi

## Failisüsteemide ühendamine (2)

- Erinevatel protsessidel võib nimeruum olla erinev (monteeritud on erinevad failisüsteemid)
- Monteerida saab failisüsteemi
- Failisüsteem võib olla mingil plokkseadmel, võrgus või üldse käigu pealt kokku pandud
- Kokku üks suur virtuaalne failisüsteem (VFS)
- Eriotstarbelised failisüsteemid – procfs, kernfs, usbfs ...

# Näide

- Windowsi operatsioonisüsteemiga arvutis (PURK) on välja jagatud kataloog „KURK“
- Linux arvutis tuleb selle kausta kasutamiseks teha järgmist:
  - su -
  - mkdir /mnt/kurk
  - mount -t cifs -o username=PURK\kasutaja,password=parool //purk/kurk /mnt/kurk

# Failide jagamine

- Sama faili tahavad mõnikord korraga kasutada mitu kasutajat.
- Selleks kasutatakse erinevaid semantikaid:
  - Lukustamine (kohustuslik ja soovitatav; *opportunistic locking*)
  - Eksklusiivne avamine
  - Atomaarsus - kas üksikute lugemis-  
kirjutamisoperatsioonide tasemel või sessiooni (open  
ja close vahel) tasemel



# Failide lukustamine - Windows

- OS haldab lukustamist
- Programmifailid lukustatakse käivitamisel – neid ei saa töötamise ajal kustutada ega muuta.
- Failide lukustamise õigused seatakse vastava parameetriga – mida lubatakse failiga teha (eksklusiivne, jagatud lugemine/jagatud kirjutamine)
- Faililukke (*file handles*) saab näha ja jälgida **Process Exploreri** abil
- *opportunistic locking*

# Failide lukustamine - Unix

- Avatud faile ja kaustu vaikimisi ei lukustata
  - faili kirjutamine on koheselt nähtav;
- Lukustamine on realiseeritud programmi tasemel

# Kaitse

- Kaks kategooriat:
  - kindlus (*reliability*)
    - failide duplikaadid, ka RAID vahendid;
    - tagavarakoopiate tegemine (*backup*);
  - kaitse (*protection*)
    - ligipääsuõigused;
    - ketaste füüsiline kaitse (hoiame seifis);
    - andmete krüpteerimine.

# Kaitse

- Igal failisüsteemi objektil (failil/kaustal) on omanik;
- omanikku tähistatakse numbrilise ID abil (Unix – UID, Windows – SID, ...);
- omanik saab määrata objektile juurdepääsu; õiguseid
- Unixi loabitid;
- **ACL** (*Access Control List*) – üldisemalt.

# UNIXi loabitid

- Kasutaja – grupp – ülejäänud
- drwxrwxrwx
- d – (*directory*) kaust
- R – (*read*) lugemisõigus 4
- W – (*write*) kirjutamisõigus 2
- X – (*execute*) käivitusõigus 1
- Käsud: getfacl, ls, setfacl, chmod, attributes

# NTFS

- Failisüsteem, milles on toetatud erinevatele kasutajatele ja gruppidele õiguste andmine.
- Ta omab ka vahendeid failisüsteemi taastamiseks logifailidest.
- Mitu versiooni
  - Nt. Win10 NTFS erineb WinXP NTFS failisüsteemist.
- Windows
  - Failide ja kaustade õigused on eraldi
  - Krüpteerimine
  - Kettakvoodid
  - Pakkimine

# Näide: MS-DOSi failisüsteem

- Kettaruumi haldamiseks kasutatakse FAT-i
  - FAT – File Allocation Table
- Failisüsteemi paigutus kettal:



- Kolm versiooni – FAT-12, FAT-16, FAT-32
  - Ketta-aadressi pikkus vastavalt 12, 16 ja 32 bitti
  - FAT-32 korral on tegelikult kasutusel 28 bitti
- Kettaploki suurus on 512 B – 32 kB

# Näide: MS-DOSi failisüsteem

- Maksimaalne partitsiooni (seksiooni) suurus:

Ploki suurus	FAT-12	FAT-16	FAT-32
0,5 kB	2 MB		
1 kB	4 MB		
2 kB	8 MB	128 MB	
4 kB	16 MB	256 MB	1 TB
8 kB		512 MB	2 TB
16 kB		1024 MB	2 TB
32 kB		2048 MB	2 TB



# Windows

- NTFS
  - Turvainfo salvestatakse
  - Suured failid, palju faile
  - Väikesed partitsioonid – aeglane, suured - kiire
- FAT32
  - Ei hoita turvainfot
  - Väikesed partitsioonid – kiire, suured - aeglane
- ExFAT
  - ACL vähesel määral (turvainfo)
  - Väga suuri faile saab hoida
  - Jõudlus on hea

vt veel: [http://www.ntfs.com/ntfs\\_vs\\_fat.htm](http://www.ntfs.com/ntfs_vs_fat.htm)

# Kaitse

- Saab määrata juurdepääsuõigused kasutajate, gruppide ja ülejäänute lõikes;
- muud meetodid:
  - võib iga failiga siduda parooli:
    - kasu on vaid siis, kui parooli tihti muudetakse;
    - suur hulk erinevaid paroole;
  - siduda parooli kaustaga;
  - failide krüpteerimine.

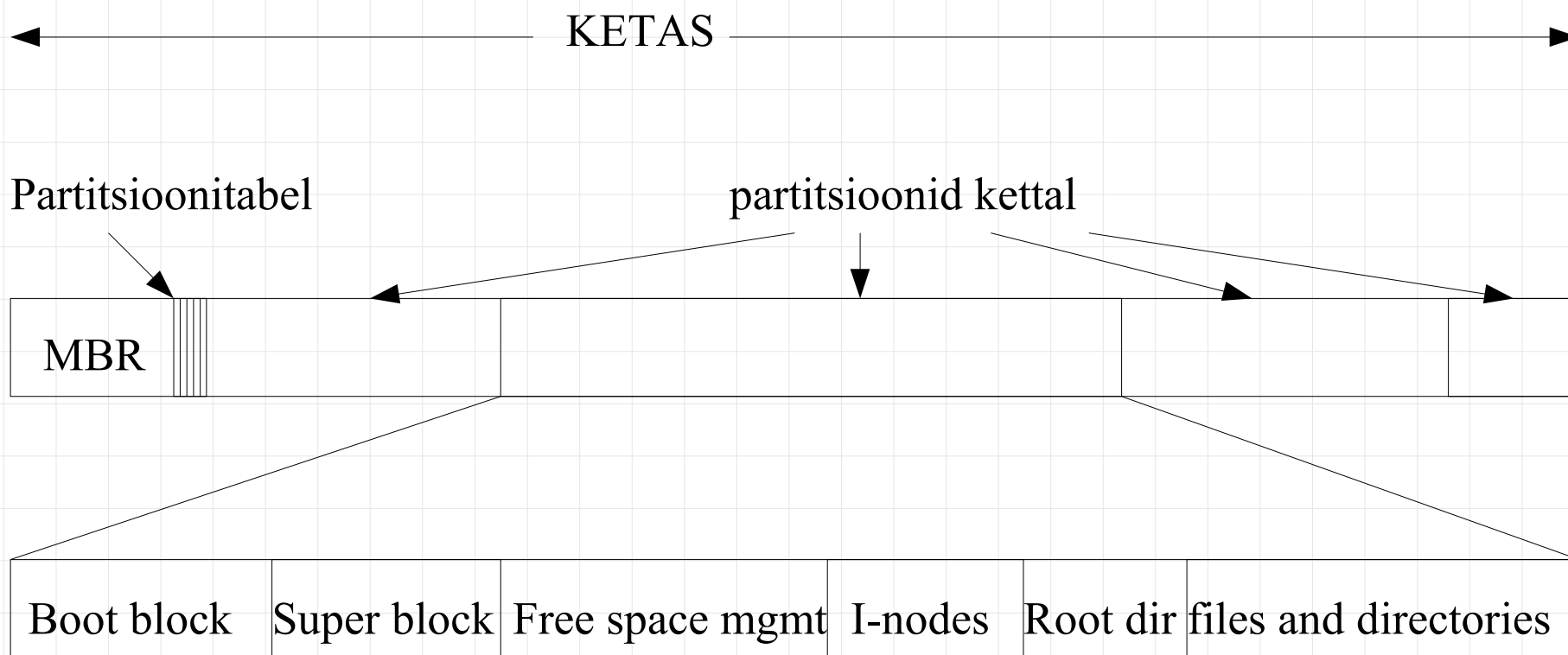
# Mida hoitakse kettal?

- (Partitsioonitabel – väljaspool konkreetset failisüsteemi);
- algladeplokk (UFS: *boot block*, NTFS: *partition boot sector*) – operatsioonisüsteemi alglaadimiseks vajalik info;
- failisüsteemi juhtplokk – konkreetse failisüsteemi detailne info:
  - (plokkide arv, vabade plokkide arv ja asukoht, vabade failikirjete arv ja asukoht, . . . ). UFS puhul superblock, NTFS puhul MFT (*Master File Table*)

# Mida hoitakse kettal?

- Kataloogistruktuur – kaustapuu hoidmiseks;
- failide kontrollplokid (FCB – *File Control Block*)
  - detailid iga konkreetse faili kohta.
  - UFS puhul i-kirje
  - NTFS puhul asub see MFT tabelis.

# Võimalik failisüsteemi laotus kettal



# *Master Boot Record (MBR)*

- Asub alati kettaseadme 0-silindri 0-lugemispera esimeses sektoris
- Lisaks neljale esmasele partitsioonile (MS-DOS tüüpi partitsioonitabeli puhul) sisaldab ka alglaadimiskoodi, mis võimaldab käivitada aktiivse partitsiooni.
- MBRi pikkus on 512 baiti ketta algusest.

# Failisüsteem hoiab mälus:

- (Ühenduspunktide tabelit)
- ühendatud failisüsteemide tabelit;
- viimati kasutatud kataloogistruktuuri kirjeid (ja muu metainfo) puhverdamise mõttes;
- puhverdatud andmeplokke;
- süsteemne avatud failide tabel – avatud failide FCB-de koopiaid;
- iga protsessi avatud failide tabel:
  - UNIX: failikirjeldused (file descriptor);
  - Win32: failipidemed (file handle).

# Fragmenteerumine

- Ühe faili andmed kettal ei ole järjest
  - Lugemiskiirus väheneb
- Defragmenteerimine
  - On olemas programmid, mis failide laialiasuvaid tükke järjestikku tagasi seavad.



# Kõvaketaste piirangud

- 1024 silindri (504 MB) piirang
- 4096 silindri (1,97 GB) piirang
  - Need piirangud on seotud kõvaketaste füüsilise adresseerimisega (vanemates) BIOSides.
- Tänapäeval kasutatakse **LBA** (*logical block addressing*) selleks et kasutada suuremaid kettaid.
- LBA aadressid ATA ketastel võivad olla 28 bitised (ketas kuni 128 GB) või 48 bitised.

# Tänaseks kõik