



IT KOLLEDŽ  
TALLINNA TEHNIKAÜLIKOOL

# Failisüsteemid

Andmesalvestustehnoloogiad ICA0006

Edmund Laugasson

[edmund.laugasson@itcollege.ee](mailto:edmund.laugasson@itcollege.ee)

[https://wiki.itcollege.ee/index.php/User:Edmund#eesti\\_keeles](https://wiki.itcollege.ee/index.php/User:Edmund#eesti_keeles)

Käesoleva dokumendi paljundamine, edasiandmine ja/või muutmine on sätestatud ühega järgnevatest litsentsidest kasutaja valikul:

\* GNU Vaba Dokumentatsiooni Litsentsi versioon 1.2 või uuem

\* Creative Commons Autorile viitamine + Jagamine samadel tingimustel 4.0 litsents (CC BY-SA)

# Loengu teemad

- fail
- failisüsteem

# Fail

- ühetüübiline ja üheselt töödeldav andmete kogum
- andmed salvestatakse faili kujul
- on salvestatav, loetav, kopeeritav ja kustutatav arvuti välismälust
- võib avada, muuta, kopeerida lõpmata arv kordi
- tööks failidega on vaja operatsioonisüsteemi kus asub kestprogramm (*shell*), tegevusi lihtsustab failihaldur (CLI, GUI)
- on olemas erinevad tüübid, nt Linuxis vaatamiseks käsk *file*
  - internetis leiab failitüübi kohta infot <https://filext.com>
- mõned failitüübid saavad salvestada erinevaid andmetüüpe, nt arhiivifailid võivad sisaldada teksti, pilti, videot jne

# Fail

- UNIXilaadsetes süsteemides virtuaalsed failid
  - */dev/\** - riistvara seadmefailid (*block, character device*) kas siis kohalikus masinas või võrgus
  - */proc* – protsessid (*/proc/PID/\**), *procfs* ühenduspunkt
    - *cat /proc/cmdline* – süsteemi laadinud tuum ja selle parameetrid
    - *cat /proc/cpuinfo* – info protsessori kohta
    - jne
  - */sys* – seadmed, juhtprogrammid, osaliselt ka tuumaga seotud funktsionaalsus
- võib asuda erinevatel andmekandjatel, sh magnetlindil, optilised andmekandjad, USB-andmekandjad, võrgus, SSD jne

# Fail

- töötava programmi poolt nähakse faili:
  - FCB (*File Control Block*, faili juhtblokk) - faili struktuuriandmeid, pikkust, nime jms sisaldav kirje; varasem lähenemine
  - failisang, failipide (*file handle*) - faili ühene lühiidentifikaator, mille opsüsteem moodustab faili avamisel; tänapäevane lähenemine

# FCB (*File Control Block*, faili juhtblokk)

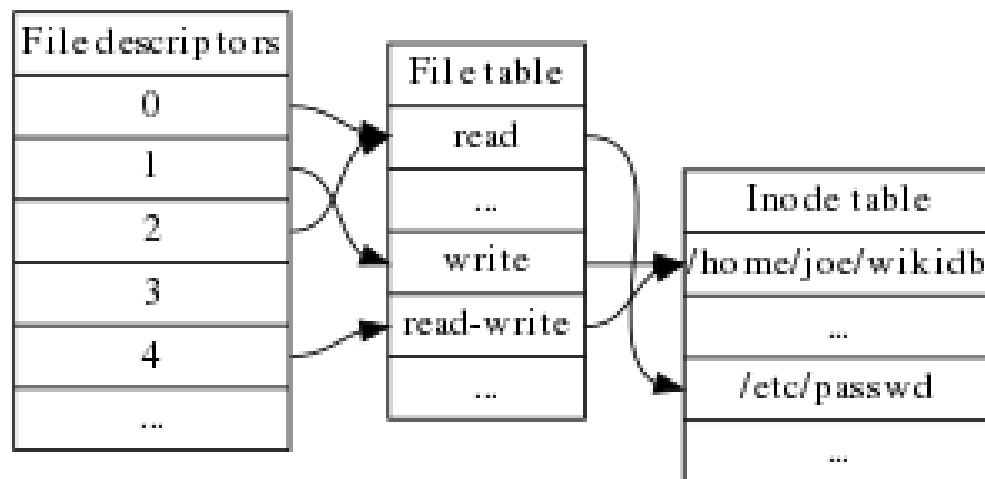
- hallatakse OSi poolt
- asub rakenduse mälus, mis faili kasutab (mitte OSi mälus)
- võimaldab protsessil korraga avada nii palju faile kui vaja kuniks suudetakse piisavalt muutmälu eraldada iga faili FCB jaoks
- pärineb CP/M (*Control Program/Monitor*) CLI-põhisest operatsioonisüsteemist, loodud 1974 (arenduses kuni 1983), oli piiratud 8-bit ja hiljem 16-bit kuni 64 kB RAMi jaoks. Algselt suletud, nüüd avatud lähtekoodiga; levinud ka DOS'is
- 36 B pikk, algses CP/M versioonis oli 33 B – see piiratus takistab tulevikus FSB'd kasutamast failide ligipääsumeetodina (puudus ka võimalus adresseerida alamkatalooge, faile võrgus)
- CP/M, 86-DOS, PC DOS 1.x/MS-DOS 1.xx puhul ainus failide ligipääsumeetod

# failisang, failipide (*file handle, file descriptor, FD*)

- tänapäevane viis failidele ligipääsuks
- abstraktne indikaator (sang, pide) failidele vms sisend/väljundressursile (toru, võrgusokkel) ligipääsuks
  - toru (*pipe, pipeline*): ühe protsessi *stdout* suunatakse teise *stdin*'i
  - võrgusokkel (*network socket*): IP-aadressist ja võrguteenust määrava pord i numbrist koosnev identifikaator
  - positiivne täisarv, nt C-keeles *int* (negatiivsed väärtused tähistavad väärtuse puudumist või viga)
- iga UNIXi protsess (teenus ehk välja arvata) omab kolme standardset POSIX failipidet
  - *stdin* (standardsisend), täisarvu (*int*) väärtus 0
  - *stdout* (standardväljund), täisarvu väärtus 1
  - *stderr* (standardveaväljund), täisarvu väärtus 2

# failisang, failipide (*file handle, file descriptor, FD*)

- failipidemed ühe protsessi, failitabeli ja infosõlme (*inode*) kohta
- mitu failipidet võivad viidata samale failitabeli kirjele (*duplicate system call* – käitub sisuliselt failipideme *alias'ena*)
- failitabeli kirje võib viidata mitmele infosõlmele – fail on avatud mitu korda



**infosõlm** (*inode*) UNIX'i failisüsteemides andmestruktuur, mis sisaldab informatsiooni individuaalsete failide kohta. Igal failil on oma infosõlm ja faile identifitseeritakse infosõlme numbri (i-number) järgi failisüsteemis, kus fail asub. Infosõlm sisaldab failide kohta olulist infot - kes on faili omanik, faili pöördusrežiim (lugemise, kirjutamise või täitmise load), faili tüüp jne. Infosõlmed luuakse samaaegselt failisüsteemi loomisega. Infosõlmi on failisüsteemis kindel arv ja see arv näitab, kui palju faile süsteem suudab maksimaalselt säilitada. Faili infosõlme numbrit võib teada saada käsuga `ls -li` ning käsk `ls -li` toob välja infosõlmes sisalduva informatsiooni; paljude failide korral: `ls -U`, `ls -f`. `df -li` näitab kasutust, haldamine: `tune2fs`



# infosõlme (*inode*) haldus

- failisüsteemipõhine suurus
- määratakse üldjuhul failisüsteemi loomisega
- ext4 puhul vaikimisi 1 inode 16384 B kohta
  - */etc/mke2fs.conf* failis määratud
- arvu saab muuta kuid mitte lihtsalt:
  - varunda andmed
  - loo failisüsteem uuesti suurema arvuga
  - taasta andmed
- võib siiski raisata ülemäära ruumi

# Faili omadused

- nimi – enamik OS'e vajab unikaalsust, mis asuvad ühes kataloogis
  - GNU/Linuxis jt UNIXilaadsetes OSides tehakse vahet tähesuurusel, MS Windows'is mitte – ühest süsteemist teise minnes võib tekkida probleeme
  - üldiselt on soovitatav nimes kasutada vaid neid tähti, mis on inglise tähestikus, lisaks araabia numbreid
  - muid sümboleid ei soovita kasutada
  - punkt on nime ja laiendi (tüübi) eraldajaks
  - punkt nimes soovitatav asendada sidekriipsuga
  - tühik soovitatav asendada alakriipsuga
  - eesti jm täpitähed asendada sarnaste tähtedega inglise tähestikust või ka sobivate numbritega
  - pikkuse piirangud, seotud sageli otsiteekonna (*path*) maksimaalse pikkusega

# Faili omadused

- laiend (tüüp) – liik metaandmeid, mida kasutatakse enamuse OS'ides kasutuse määramiseks
  - üldjuhul 1...3 sümbolit
  - üldiselt rohkem oluline MS Windows'is; UNIX'ilaadsetes vähemolulisem (avamisel loetakse failipäist) ja võib ka puududa
  - graafilised failihaldurid sageli seovad kindla rakendusega
  - MS Windows'is vaikimisi peidetud (risk pahavara käivitada)
- aeg – millal fail loodi, faili poole pöörduti
- omanik, grupp
- juurdepääsuõigused
- võib olla võimalik ka versioneerimine

# Toimingud failidega

- avamine – fail avatakse kohalikult või võrgus asuvalt andmekandjalt ja loetakse muutmällu kus rakendused saavad failis olevaid andmeid omakorda lugeda
- sulgemine – sõltuvalt rakendusest küsitakse kas muutmälus olevad failiga seotud andmed salvestada andmekandjale, fail suletakse ja muutmälust (RAM) hõivatud osa vabastatakse
- kirjutamine – muutmälus olevate andmete salvestamine andmekandjale
- lugemine – failist muutmällu loetud andmete poole pöördumine mõne rakenduse poolt

# Failide haldamine

- üldiselt pöörduakse faili poole nimepidi, mis peab tavaliselt olema unikaalne samas kataloogis
- asukoht (*path*) suhteline või absoluutne (alates juurkataloogist UNIXilaadsetes OSides või MS Windowsis kettajao tähisest)
  - MS Windows: UNC (Universal/Uniform Naming Convention)
    - \\ComputerName\SharedFolder\Resource (võrgus, nime asemel võib olla ka IP-aadress)
    - võib sisaldada ka SSL, TCP/IP pordinumbrit
    - WebDAV'i aadress  
http[s]://HostName[:Port]/SharedFolder/Resource on siis \\HostName[@SSL][@Port]\SharedFolder\Resource
    - mõned haldusliidesed toetavad ka pikka UNC'd:  
\\?\UNC\ComputerName\SharedFolder\Resource

# Failide haldamine

- MS Windows:
  - C:\File (absoluutne); \File, .\File (suhteline) – kohalikud failid
  - \\Server\Volume\File  
või ka  
/<internet resource name>[\Directory name] – vähemalt Win7 ja uuemad
  - pikk UNC  
\\?\C:\File või ka \\?\UNC\Server\Volume\File
- UNIX'ilaadsed (POSIX)
  - /file (absoluutne); .file ./file (suhteline) – kohalik (sh viide), ka ühendatud võrguressurss
  - hostname:/directorypath/resource (scp, rsync jt)
  - smb://hostname/directorypath/resource – analoogselt veebile (protocol://[user:psw@]host[:port]/directorypath/resource)

# Failide haldamine

- failid asuvad kataloogides, mis olemuselt fail kuid grupeerivad „päris” faile; kataloogid võivad sisaldada ka viiteid (linke) failidele
- juurkataloog luuakse failisüsteemi loomisega, hiljem loodud kataloogid on juba juurkataloogi alamkataloogid
- UNIXilaadsetes süsteemides FHS (Filesystem Hierarchy Standard), kindel struktuur ka MS Windowsis
- kodeerimine – millist kooditabelit kasutatakse failide, kataloogide nimede kirjutamisel. Tänapäeval kasutusel Unicode (levinuim UTF-8) – universaalne: toetab erinevaid tähestikke, sh erinevaid märgikirju (nt hieroglüüfid jm). Tugi MS Windows NT ja selle uuemad versioonid (UTF-16 alates W2K), UNIXilaadsed süsteemid, veebipõhised süsteemid. Siiski soovitav vältida erisümboleid, täpitähti faili-, katalooginimedes

# Failide kaitse

- paljudes failisüsteemides on selleks õigused
  - keelatakse ära kirjutamine (*read-only*)
  - vaid piiratud kasutajatele antakse õigus muuta
- mõned failisüsteemid kasutavad ka peitmist (*hidden flag*)
  - näiteks süsteemsed failid MS Windowsis
  - punktiga algavad failid/kataloogid UNIXilaadsetes süsteemides
- krüpteerimine – takistatakse volitamata kasutajate ligipääs, lisaks võidakse rakendada ka peitmist (vt VeraCrypt)



# Kataloogi metafoor

- kataloog kui paberilehti siduva kausta analoog, kasutusel tänapäeva OS'ides ehk siis failisüsteemi kontseptsioon, üldiselt suurusega 4 kB
- on erinevus failisüsteemi kontseptsiooni ja GUI-poolsele kataloogi interpretatsioonil
  - nt MS Windows'is erikaustad, nt kesta kaustad, nt töölaud (%USERPROFILE%\Desktop), rakenduste andmed (%APPDATA%), süsteemikaust (%WINDIR%) jne, mis võivad eri MS Windowsi versioonide või ka konkreetsete paigalduste lõikes erineda olla (keskkonnamuutujad saavad viidata eri asukohtadele sõltuvalt valitud sätetest)
  - mitmed OSid omavad **virtuaalkataloogi** kontseptsiooni – need ei ilmu otseselt kataloogi struktuuri, sisupõhine lähenemine, andmeid (faile) siia ei saa salvestada
    - otsingut kuvatakse virtuaalse kaustana – võivad olla ka võrgus, nii failisüsteemi kui ka nt e-postikliendi, andmebaasi, meediamängija otsing

# Failisüsteem

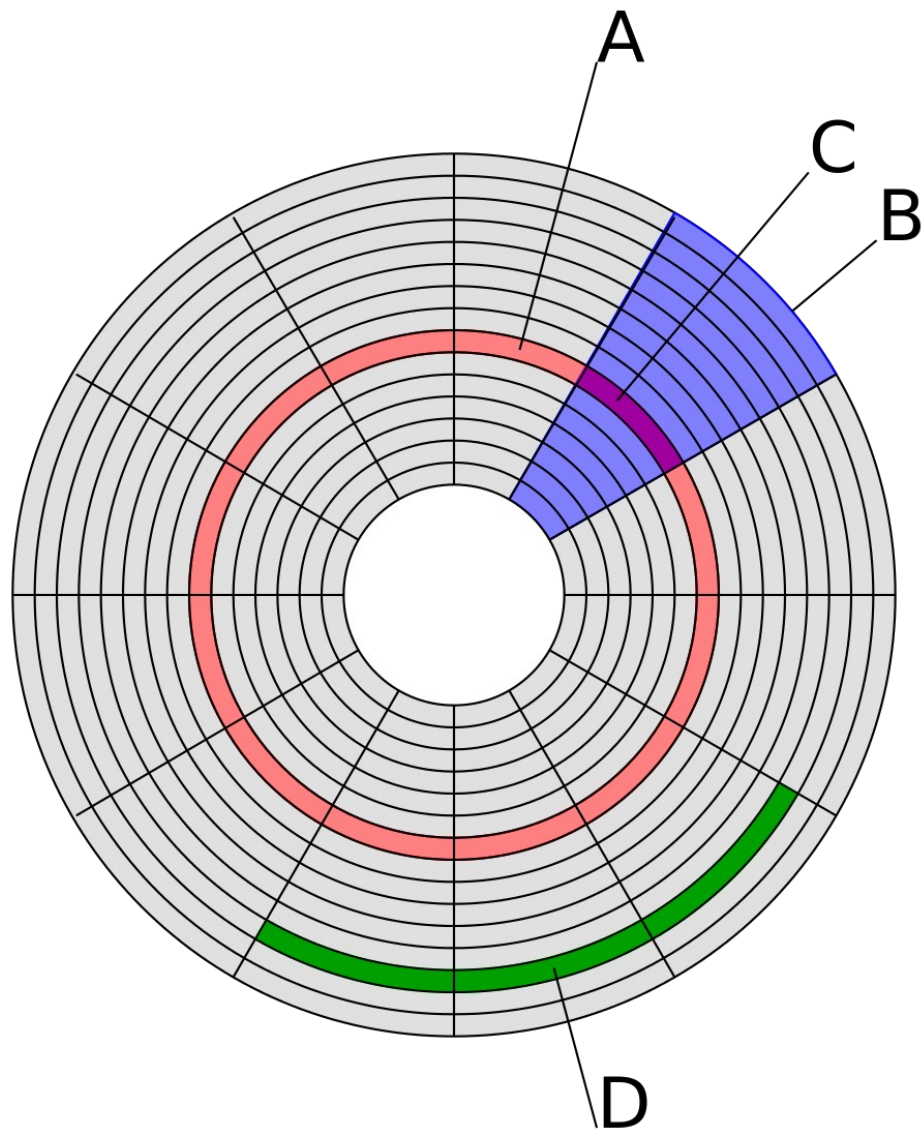
- andmestruktuuride, algoritmide ja tarkvara kogum
- eesmärk: salvestusseadme peale andmete organiseeritud paigutamine hilisemaks kasutamiseks:
  - andmed peavad olema leitavad
  - andmed peavad olema kättesaadavad

# Failisüsteem

- ülesanded
  - määrab failide paiknemise ja struktuuri füüsilisel kettal
  - seab piirangud failinimedele ja nende suurustele
  - võimaldab faile:
    - salvestada
    - kustutada
    - nimetada
    - organiseerida
    - pakkida
    - määrata failidele ligipääsuõigusi

# Failisüsteemi tööpõhimõte

- andmekandjal olev ruum on jaotatud sektoriteks (*sector*), üldjuhul suurusega 512 baiti
- klaster on mingi täisarvuline hulk järjestikku asuvaid sektoreid kuna enamasti ei ole failisüsteemil nii väikeste andmeüksustega praktiline töötada
  - 512 B sektori puhul levinumad klastrid baitides: 2048 (4x512), 4096 (8x512), 8192 (16x512) 16384 (32x512) ja 32768 (64x512)
- suurema klatri korral väheneb suurte failide puhul fragmentatsioon kuid väiksemate failide puhul suureneb raisatud ruum
  - kui fail on väiksem kui klatri suurus siis see jääb tühjaks
  - kui fail on suurem siis jaotub mitmesse klastrisse



Ketta struktuur:

A – Rada

B – Geomeetiline sektor

C – Raja sektor

D – Klaster

# plokk vs sektor

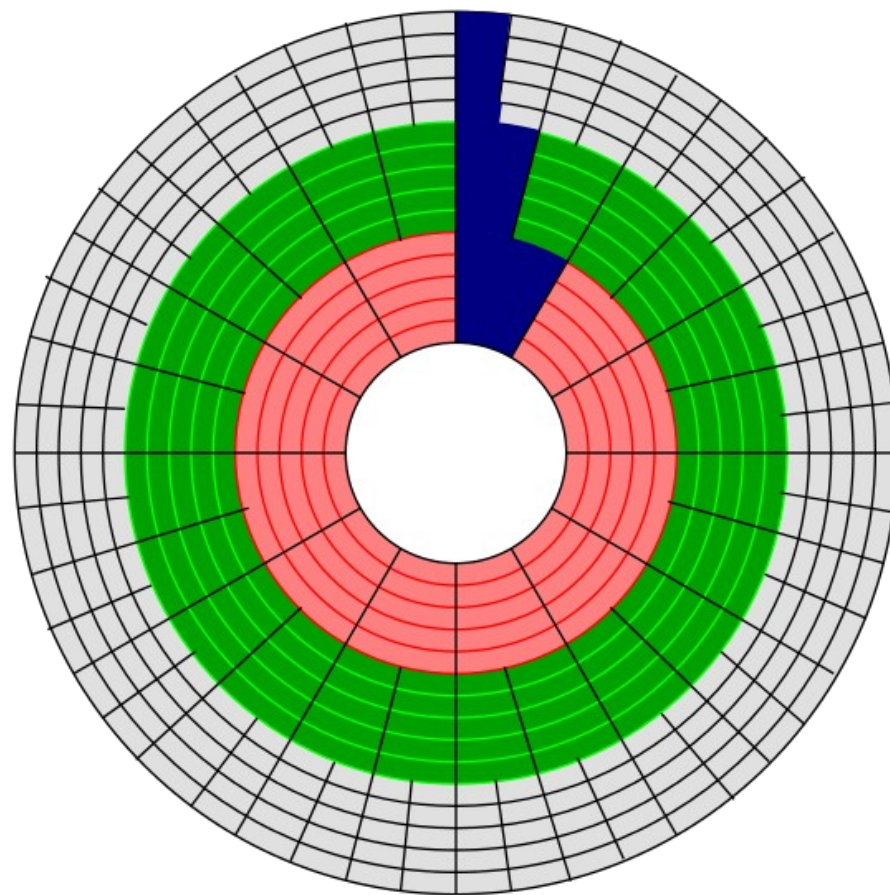
- kui *sektor* määrab kindla ala siis *plokk* vabalt valitud andmetüki
- tähendus sõltub kontekstist
  - abstraktsioon, omaduste määramine üle mitme sektori
  - ühik, andmevoog rakenduses; nt dd puhul bs=bytes – määrab andmetüki (*chunk of data*) suuruse ja ei ole seotud failisüsteemi sektori või plokiga
- Linuxis
  - sektori suurus
    - `fdisk -l | grep "Sector size" -> 512 B`
  - ploki suurus (vt *man blockdev*)
    - `blockdev --getbsz /dev/sdx #sdx` asemele tegelik ketas
      - 4096 B

# zone bit recording (ZBR)

- andmete tsoonsalvestus?
- meetod andmete salvestustiheduse optimeerimiseks
- pöörleva andmekandja keskpunktist välimistele radadele (*track*) paigutatakse rohkem sektoreid (*sector*) võrreldes sisemiste radadega
- erineb teistest lahendustest, nt konstantne nurkkiirus (*CAV* – *Constant Angular Velocity*) kus sektorite arv raja kohta on sama
- sama keskpunktiga (kontsentrilised) rajad – kas realiseeritakse eraldi ringikujuliste radadena või spiraalina – viimasel juhul suureneb raja pikkus kui see keskpunktist kaugemal
- sisemistel radadel pakitakse andmed nii tihedalt kui andmekandja tehnoloogia lubab

# zone bit recording (ZBR)

Füüsiline sektorite paigutus andmekandjal tsoonidena: keskkohast kauguse kasvades kasvab ka sektorite arv etteantud nurga all ühest (punane), kaheni (roheline) kuni neljani (hall)



■ Sector 0



# zone bit recording (ZBR)

- püsiva nurkkiirusega (*CAV, Constant Angular Velocity*) kettal on välimised rajad sama nurga all, mis keskmised ja hõredamalt pakitud andmed
- kasutades andmete tsoonsalvestust (ZBR) sisemisi tsoone kasutatakse lugemis-, kirjutamiskiiruse määramiseks – on samad ka välimistele radadele
- võimaldab rohkem andmeid välimistele radadele salvestada võrreldes sisemistega
- kokkuvõttes suurem andmetihedus sama kettapinna puhul
- välimistel radadel kõrgeim andmeedastuskiirus (*data transfer rate*)
- radade nummerdamine algab tavaliselt sisemistest ja ka andmed paigutatakse nummerdamise algusest – saaleala näiteks soovitav teha seetõttu ketta algusesse
- kasutusala: disketid, DVD-RAM, HD DVD, pöörlevad kõvakettad alates 1990-ndatest

# Failisüsteemi tööpõhimõte

- sõltuvalt failisüsteemist kasutab see erinevaid süsteemseid andmestruktuure (nt tabelid, puud) andmete hoidmiseks, nt:
  - failinimede nimekiri
  - kataloogipuu kirjeldus
  - failide asukohad – nt FAT puhul klastri number, millelt fail algab
  - tabel tühjadest klastritest
  - ketta vigased sektorid (*bad sector*) kuhu andmeid enam ei kirjutata – füüsiliselt vigastatud kohad
  - tabel failide omanike ja juurdepääsuõigustega

# Vigased sektorid

- OS'i tasandil
  - tuvastamiseks chkdsk (MS Windows), Disk Utility (macOS), badblocks (Linux, osa *e2fsprogs* projektist)
- kettakontrolleri (riistvara) tasandil
  - kui vigane sektor leitud siis kettakontroller ühendab loogilise sektori teise füüsilise sektoriga, tüübid:
    - P-LIST vastavuse teisendamine tehase testimise käigus
    - G-LIST vastavuse teisendamine (*mapping*) kasutaja tegevuse käigus kettakontrolleri mikrokoodi poolt (kui G-LIST täis siis tuli andmekandja välja vahetada)
- haldusvahendid suudavad lugeda SMART (*Self-Monitoring, Analysis and Reporting Technology*) andmeid – toetab pöörlevaid kettaid, pooljuhtkettaid (SSD), eMMC (embedded Multi Media Card), et öelda mitu sektorit on teisendatud

# vigased plokid

- Linuxis rakendus badblocks
  - *badblocks -nvs /dev/sdb*
    - kontrollib failisüsteemi vigaste plokkide suhtes mittepurustavas lugemis-, kirjutamisrežiimis (*non-destructive read-write mode*)
    - sobib failisüsteemidele kus on andmed – teeb andmetest eelnevalt varukoopia ja pärast kontrollimist taastab, plokikaupa

# vigased plokid

- Linuxis rakendus badblocks
  - *badblocks -wsv -t random /dev/device*
    - *-w write, -s show, -v verbose, -t test\_pattern*
    - purustav režiim koos juhuslike andmetega ülekirjutamisega (ei loeta siiski piisavalt turvaliseks)
    - nt enne failisüsteemi loomist
  - *badblocks -wsv -o /root/badblocks.txt /dev/sdx*
    - *mkfs.filesystem-type -l /root/badblocks.txt /dev/sdx*
  - *fsck (file system check)*
    - *fsck -vcck /dev/sdx*
      - *-cc non-destructive mode, -v verbose, -k preserve old bad sectors*
- MS Windows: [chkdsk](#) (*check disk*)

# Andmete eesmärgipärane kaitsmine

- sektoreid võidakse märkida „halvaks” ka eesmärgipäraselt sooviga kaitsta andmeid
  - pöörlevate andmekandjate puhul
  - erinevad tehnikad
    - nõudis üldjuhul vähemalt ühte andmekandja täispööret igal lugemisel – sihilikult lugemise aeglaseks muutmine
    - päiseinfo dubleeritud selliselt, et igal lugemisel loeti erinevad andmed välja
    - andmeid ei loetud korrektselt välja
    - jne
  - neid tehnikaid sai siiski üle kavaldada kuna see kood asus üldjuhul alglaadimissektoris ja selle ülekirjutamisega sai tegelikele andmetele ligi

# Failisüsteemid

- FAT – File Allocation Table, failipaigutustabel on failisüsteemi ehitus, mille puhul operatsioonisüsteem paigutab failid klastritesse. Iga fail kasutab minimaalselt üht klastrit. Klastrid (loogilised üksused) koosnevad fikseeritud suurusega sektoritest (füüsilised üksused) ja on adresseeritud n-bitiste kannetega aadressiruumi (tabelisse), kus n on sõltuvalt FAT versioonist 12 (FAT12), 16 (FAT16) või 32 bitti (FAT32)
- Failipaigutustabel sisaldab iga kettal oleva faili algusklastri kannet, mis omakorda sisaldab viita järgmisele failiga seotud klastrile ja nii edasi, kuni faililõpu klastrini.
- 1996. aastast kuulub FAT-i patent Microsoftile.

# Failisüsteemid

- FAT16 loodi 1987 Compaq'i poolt
  - hea tugi eri OSide poolt
  - kettajao suurus kuni 2 GB
  - failinimede pikkus kuni 11, millest 3 on failitüüp
  - piirangu ületamiseks VFAT (Virtual FAT) laiendus, võimaldas kuni 255-kohalisi failinimesid



# Failisüsteemid

- FAT32 on edasiarendus FA16'st
  - maksimaalsete klastrite arvu suurendamine
  - kuni 2 TB kettajagu
  - samuti hea tugi eri OSide poolt
- NTFS (*New Technology File System*)
  - Microsofti failisüsteem
  - eesmärk vabastada FAT'i piirangutest
  - ei ole head tuge OSide poolt kuna spetsifikatsioone ei ole välja antud, siiski ntfs-3g loeb-kirjutab
  - kuni 255 failinimes, UTF-16 kodeeringus
  - suurus kuni 16 EB
  - laiendatud failiomadused, sisseehitatud pakkimine
  - päevikuga (journaling) failisüsteem

# Failisüsteemid

- EXT-failisüsteemid
  - extended filesystem, kasutusel GNU/Linuxis
  - EXT2, EXT3, EXT4
  - EXT2 puudub *journaling* (muudatuste logi, mille abil saab andmekao ohtu oluliselt vähendada) tugi, ka päevikuga failisüsteem
  - EXT3 ühildub EXT2'ga
  - EXT4 vähendab fragmenteerumist suuremate failide puhul, lisab journaling'ile kontrollsummad ja võimaldab kiiremat failisüsteemi kontrollimist
  - kõik EXT-failisüsteemid ühilduvad POSIX-standardi ligipääsuõigustega

# Failisüsteemid

- HFS+, APFS
  - Apple'i failisüsteemid
  - alates macOS 10.2.2 on toetatud ja päevik (journaling)
  - kuni 255 sümbolit failinimes
  - UTF-16
  - toetatud ka UNIXi failiõigused
  - <https://wiki.itcollege.ee/index.php/APFS>

# Võrgufailisüsteemid

- NFS – Network File System
  - läbipaistvalt nagu asuksid kohalikus arvutis
  - sobivad failiõigused peavad olema määratud
- SMB – Server Message Block
  - läbipaistvalt nagu asuksid kohalikus arvutis
  - tundud kui Microsoft Windows Network

# GNU/Linux vs MS Windows failisüsteem

- kataloogide struktuur
  - MS Windows: \Windows, \Program Files, \Users
  - GNU/Linux: /lib /usr /home
- tõstutundlikkus:
  - MS Windows: ei ole (file, FILE võimatu samas kaustas)
  - GNU/Linux: on (file, FILE võimalik samas kaustas)
- kaldkriips vs tagurpidi kaldkriips
  - MS Windows: C:\Users\Name
  - GNU/Linux: /home/name

# GNU/Linux vs MS Windows failisüsteem

- andmekandjate tähistus
  - MS Windows: ladina täht + koolon, nt C:
  - GNU/Linux: kõik asub / (juurkataloogis)
- kõik on fail (lihtsustatud kujul)
  - MS Windows: /dev/random analoog ksecdd.sys kuid \Device\KsecDD ei toimi nii nagu Linuxis
  - GNU/Linux:
    - füüsiline riistvara on fail /dev kaustas, nt /dev/sda esimene kõvaketas jne (vt *lsblk*)
    - nt fail /proc/cpuinfo sisaldab infot CPU kohta

# GNU/Linux vs MS Windows failisüsteem

- kõik on fail (lihtsustatud kujul)
  - GNU/Linux:
    - spetsiifilised failid
    - /dev/null - kustutab saadetud andmed, nt veavoo suunamine: *command 2>/dev/null*
    - /dev/random - juhuandmete generaator (krüptograafias)
    - /dev/zero - pidev voog nulle – kasutatakse andmekandjate madalatasemeliseks vormindamiseks  
*dd if=/dev/zero of=/dev/sdx1*  
*low-level format* – sisuliselt andmete taastamisvõimaluseta hävitamine, sageli aitab ka failisüsteemist tulenevate probleemide vastu (nt *bad block*)

# GNU/Linux vs MS Windows failisüsteem

- saab kustutada, muuta avatud faile
  - MS Windows: üldiselt ei – faili kustutamisel tuleb veateade, et see on kasutuses. Faili kustutamiseks tuleb selle kasutamine esmalt lõpetada. Siiski on võimalik nt Process Explorer'i jt abil seda teha.
  - GNU/Linux: üldiselt jah – siin rakendused ei lukusta faile selliselt nagu MS Windows'is ja on võimalik isegi hetkel kasutuses olevaid faile kustutada. Sellisel juhul sageli võimalik siiski veel muutmälus olev fail salvestada.



# Viited 1

- [https://en.wikipedia.org/wiki/Computer\\_file](https://en.wikipedia.org/wiki/Computer_file)
- <https://en.wikipedia.org/wiki/Filename>
- [https://en.wikipedia.org/wiki/Filename\\_extension](https://en.wikipedia.org/wiki/Filename_extension)
- [https://en.wikipedia.org/wiki/File\\_system](https://en.wikipedia.org/wiki/File_system)
- [https://en.wikipedia.org/wiki/Unix\\_filesystem](https://en.wikipedia.org/wiki/Unix_filesystem)
- <http://refspecs.linuxfoundation.org/fhs.shtml>
- [https://en.wikipedia.org/wiki/Directory\\_\(computing\)](https://en.wikipedia.org/wiki/Directory_(computing))
- [https://en.wikipedia.org/wiki/Directory\\_structure](https://en.wikipedia.org/wiki/Directory_structure)
- [https://en.wikipedia.org/wiki/Path\\_\(computing\)](https://en.wikipedia.org/wiki/Path_(computing))
- <https://en.wikipedia.org/wiki/WinFS>

## Viited 2

- [https://en.wikipedia.org/wiki/Special\\_folder](https://en.wikipedia.org/wiki/Special_folder) - MS Windowsi erikaustad
- [https://en.wikipedia.org/wiki/Virtual\\_folder](https://en.wikipedia.org/wiki/Virtual_folder)
- <https://en.wikipedia.org/wiki/Unicode>
- [https://en.wikipedia.org/wiki/Unicode\\_in\\_Microsoft\\_Windows](https://en.wikipedia.org/wiki/Unicode_in_Microsoft_Windows)
- [https://en.wikipedia.org/wiki/Disk\\_sector](https://en.wikipedia.org/wiki/Disk_sector)
- <https://en.wikipedia.org/wiki/Badblocks>
- <https://en.wikipedia.org/wiki/E2fsprogs> - ext2/3/4 failisüsteemi haldamise vahendid
- [https://en.wikipedia.org/wiki/Self-Monitoring,\\_Analysis,\\_and\\_Reporting\\_Technology](https://en.wikipedia.org/wiki/Self-Monitoring,_Analysis,_and_Reporting_Technology) -S.M.A.R.T.
- <https://en.wikipedia.org/wiki/Procfs>

## Viited 3

- [https://en.wikipedia.org/wiki/File\\_descriptor](https://en.wikipedia.org/wiki/File_descriptor)
- [https://en.wikipedia.org/wiki/Pipe\\_\(Unix\)](https://en.wikipedia.org/wiki/Pipe_(Unix))
- [https://en.wikipedia.org/wiki/Network\\_socket](https://en.wikipedia.org/wiki/Network_socket)
- [https://en.wikipedia.org/wiki/Block\\_\(data\\_storage\)](https://en.wikipedia.org/wiki/Block_(data_storage))

# Küsimused? Tänan tähelepanu eest!



IT KOLLEDŽ  
TALLINNA TEHNIKAÜLIKOOL



**TTÜ IT KOLLEDŽ**

**Raja 4C, 12616 Tallinn**

**tel +372 628 5800**

**info@itcollege.ee**

**<http://www.itcollege.ee/>**