

## Failisüsteemi liides

- Faili mõiste
- Juurdepääsumeedodid
- Kataloogistruktuur
- Failisüsteemide monteerimine
- Failide jagamine
- Kaitse

## Faili struktuur

- Ei mingit – baitide või sõnade jada
- Lihtne kirjestruktuur
  - Read
  - Fikseeritud pikkusega kirjed
  - Erineva pikkusega kirjed
- Keerukamad struktuurid
  - Formaaditud dokumendid (OLE objektivoog, ASN.1, ...)
  - Segmentidest koosnev käivitata fail
- Keerulisemaid formaate saab lihtsamate abil esitada
- Kelle kontrolli all on formaat?
  - Opsüsteemi
  - Rakendusprogrammi

## Faili atribuudid

- Nimi – ainus inimloetaval kujul olev metainfo
- Tüüp – vajalik süsteemides, mis toetavad erinevat tüüpi faile (enamus)
- Sisemine identifikaator – opsüsteemi sisemine identifikaator (näiteks i-kirjed UNIXis)
- Asukoht – kuskohas fail kettal asub
- Maht – faili pikkus
- Kaitse – juurdepääsuõigused
- Omanikuinfo – turvalisuse ja arvepidamise tarvis
- Muutmise, lugemise, loomise ajad – monitooringuks
- Metainfot hoitakse enamasti kataloogistruktuuris
- Harud (*streams*)?

## Operatsioonid failidega

- Loomine
- Kirjutamine
- Lugemine
- Positsioneerimine failis (*seek*)
- Kustutamine
- Pikkuse muutmine
- Kõrgema taseme operatsioonid on realiseeritavad nende kaudu
- Avamine (→ failipide)
- Sulgemine

## Pöördusmeetodid

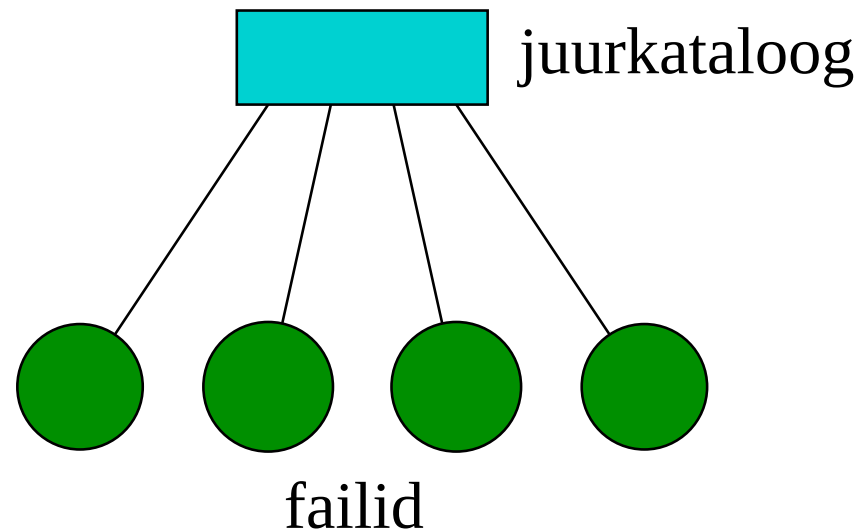
- Järjestikune pöördus – modelleeritud lindiseadme järgi
  - read next
  - write next
  - rewind
  - Puudub rewrite (ülekirjutamine nii, et tagapool andmed säilivad)

## Pöördusmeetodid

- Otsepöördus – modelleeritud kettaseadme järgi
  - read  $n$
  - write  $n$
  - position to  $n$
  - read next
  - write next
  - rewrite  $n$
- $n$  = suhteline ploki number (faili alguse suhtes)
- Järjestikpöördust saab simuleerida otsepöörduse abil

## Kataloogistruktuur

- Kusagil peab olema nimekiri failidest
- Lihtsaimal juhul on iga seadme alguses kataloog selle seadme failide kohta



- Kataloogis on iga faili kohta kataloogikirje (nimi, pikkus, aadress, omanik jne)

## Operatsioonid kataloogiga

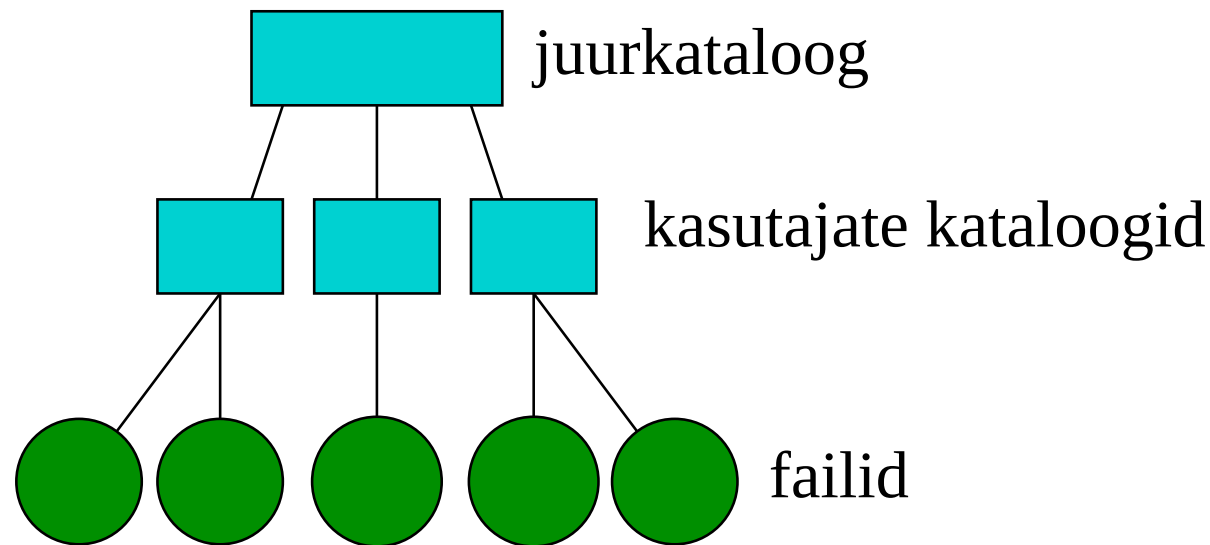
- Faili otsimine nime järgi
- Faili loomine
- Faili kustutamine
- Kataloogi nimekirja vaatamine
- Faili ümber nimetamine
- Failisüsteemi läbimine

## Kataloogisüsteemi organiseerimine

- Ühest üldisest kataloogist kipub väheks jääma
- Efektiivsus – tahame faili kergesti üles leida
- Nimetamine – kasutajatele mugavuse jaoks
  - Mitu kasutajat võivad tahta sama failinime kasutada
  - Sama fail võib esineda mite erineva nime all
- Grupeerimine – tahame faile loogiliselt grupeerida mingite omaduste järgi

## Kahetasemeline kataloogistruktuur

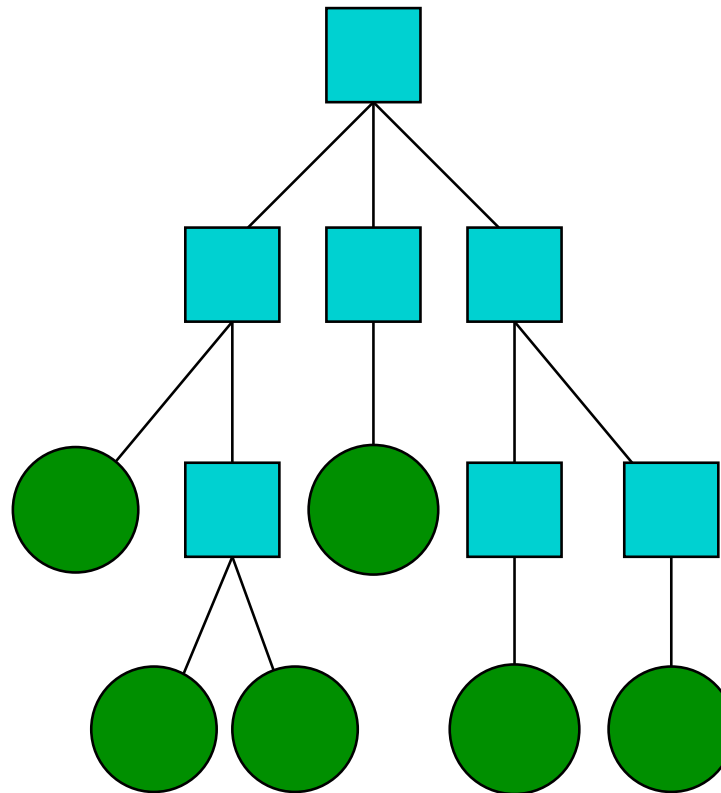
- Igale kasutajale oma kataloog; tipptasemel on metakataloog, kus on vaid viited kasutajate kataloogidele



- Igal kasutajal oma sõltumatu nimeruum ainult oma failidega
- Kas teiste kasutajate kataloogidesse näeb?
- Kui näeb, siis kuidas sinna pöörduda? → kataloogitee
- Kuskohas on süsteemsed programmid, mis kõigile näha on?

## Puustruktuuriga kataloogid

- Arendame ideed edasi: laseme kataloogipuu suvalise sügavuse peale



- Saavutame loogilise grupeerimise

## Puustruktuuriga kataloogid

- Kataloog on eritüübiline fail, mille sees on uued kataloogikirjed
- Kataloogipuu manipuleerimise operatsioonid olid algselt failioperatsioonid, hiljem eraldati nad kataloogioperatsioonideks:
  - Kataloogi loomine (mkdir)
  - Kataloogi kustutamine (rmdir)
- Jooksva kataloogi mõiste (igal protsessil või globaalne)
  - Jooksva kataloogi muutmine (chdir), jooksva kataloogi küsimine
- Kataloogitee (absoluutne ja suhteline)
- Otsimistee (PATH), *Desktop File*

## Tsükliteta graaf

- Puustruktuur keelab failide ja kataloogide jagamise erinevate harude vahel
- Kasutajad tahavad vahel jagamist (sama fail või kataloog on nähtav mitmes kataloogist)
- Puu asemel võtame kasutusele suunatud tsükliteta graafi
- Link – meetod alias-nimede tekitamiseks
  - Nimeviit (*symbolic link*, *symlink*)
  - Viit (*hard link*)
- Mis saab, kui viidatav objekt kustutatakse?
  - Kasutuskordade loendamine
  - Tagasiviidad linkide kustutamiseks

## Üldine graaf

- Viidad võivad tekitada kataloogistruktuuris tsükleid
- Tsüklite haldamine on problemaatiline, näiteks:
  - Failisüsteemi läbimisel
  - Faili kustutamisel – kasutuskordade loendamine ei aita
- Graafist tsükli leidmise algoritmid on kulukad, eriti kui arvestada igal sammul kettalt lugemist
- Prügikoristus kettal on samuti kulukas
- Lahendus: mitte lubada viitamist kataloogidele
- Nimeviitade korral on tsüklid tavaliselt siiski lubatud
  - Failisüsteemi läbimisel piiratakse nimeviitadega saadavat sügavust

## Failisüsteemide monteerimine

- Monteerimine, ühendamine, ingl. k. *mounting*
- Failisüsteem tuleb enne kasutamist monteerida, et ta nimeruumis nähtavaks saaks
- Failisüsteem monteeritakse mingisse kindlasse punkti olemasolevas nimeruumis
- Enamasti on monteerimispunktiks tühi kataloog
- Mis saab, kui monteeritakse mittetühja kataloogi peale?
  - Viga
  - Katab vana sisu kinni kuni lahti monteerimiseni
- Erinevatel protsessidel võib nimeruum (*namespace*) olla erinev
  - monteeritud on erinevad failisüsteemid
- Ühendmonteerimine (*union mount*)

## 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, autofs, ...

## Failide jagamine

- Sama faili tahavad vahel mitu kasutajat/protsessi korraga kasutada
- Kuidagi tuleb faile konsistentsena hoida
- Siin kasutatakse erinevaid semantikaid:
  - Lukustamine (kohustuslik ja soovitatav; *opportunistic locking*)
  - Eksklusiivne avamine
  - Atomaarsus - kas üksikute lugemis-kirjutamisoperatsioonide tasemel või sessiooni (open ja close vahel) tasemel

## Kaitse

- Igal failisüsteemi objektile (failid, kataloogid, ...) on omanik
- Omaniku tähistamiseks mingi numbriline ID (UNIXis UID, Windowsis SID, ...)
- Omanik saab määrata objektile juurdepääsuõigused
- Loabitid UNIXis
- ACL (*Acces Control List*) – üldisem
- Juurdepääs kasutajate, gruppide lõikes ja ülejäänutele
- Alternatiiv: failide või kataloogidega paroolide sidumine