

Operacinės sistemas

N. Sarafinienė
2011m.

Kalbėsime

- Failų sistemą: failus, jų aprašus, aplankus, jų struktūrą
- Failų fizinius talpinimo diske metodus
- Žurnalinės failų sistemos
- Failų sistemų pavyzdžiai
- Atsarginės kopijos

Failų sistema

- Dauguma kompiuterinių sistemų vartotojų tiesiogiai dirbdami su kompiuteriu yra įgavę tam tikrą supratimą apie tai kas tai yra failas, katalogas ar failų sistema.
- Tačiau pažiūrėkime į tai iš informacijos saugojimo kompiuteryje pusės.
- Pagrindinis kompiuterių tikslas yra kurti duomenis, jais manipuluoti, juos saugoti ar esant reikalui pateikti.
- Failų sistema ir suteikia mechanizmą, skirtą šių užduočių vykdymui.
- *Failų sistema ir pateikia būdą kaip saugoti, pateikti ir valdyti informaciją nuolatinio saugojimo įrenginyje koku yra diskas.*

Failas ir failų sistema

- Failais vadinamas bet koks, turintis vardą duomenų rinkinys, kuriuo yra manipuluojama kaip atskiru vienetu.
- Operacinė sistema gali sukurti vartotojui sąsają, kuri palengvina jam navigacijos veiksmus su failais. Šia sąsaja yra failų sistema.

Failų sistema

- Failų sistema teikia metodą, skirtą failų saugojimui ir organizavimui
 - Teikti loginį (abstraktų) požiūrį į failus ir katalogus
 - Užtikrina loginę failų organizaciją,
 - Padėti efektyviai naudoti atmintinės įrenginius
 - Palaikyti bendrą naudojimąsi duomenimis

Failų sistema ir diskai

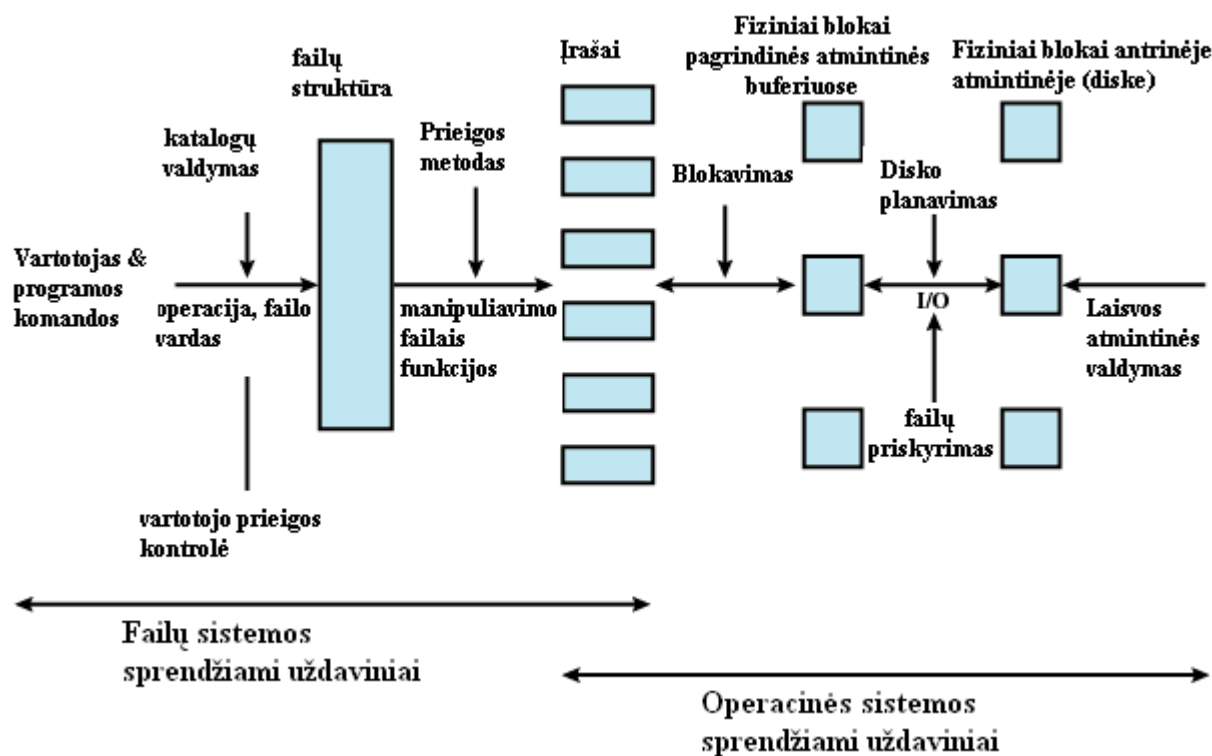
- Diskai yra atmintinės įrenginiai, skirti failų sistemos saugojimui.
- Formatuodami diską, mes “įkurdiname” jame atitinkamą failų sistemą.
- Ji gali būti:
 - FAT (FAT16, FAT32)- file allocation table
 - NTFS(Windows NT/2000/XP)
 - UNIX
 - ext2, ext3, ext4
 - ...

Failų sistema

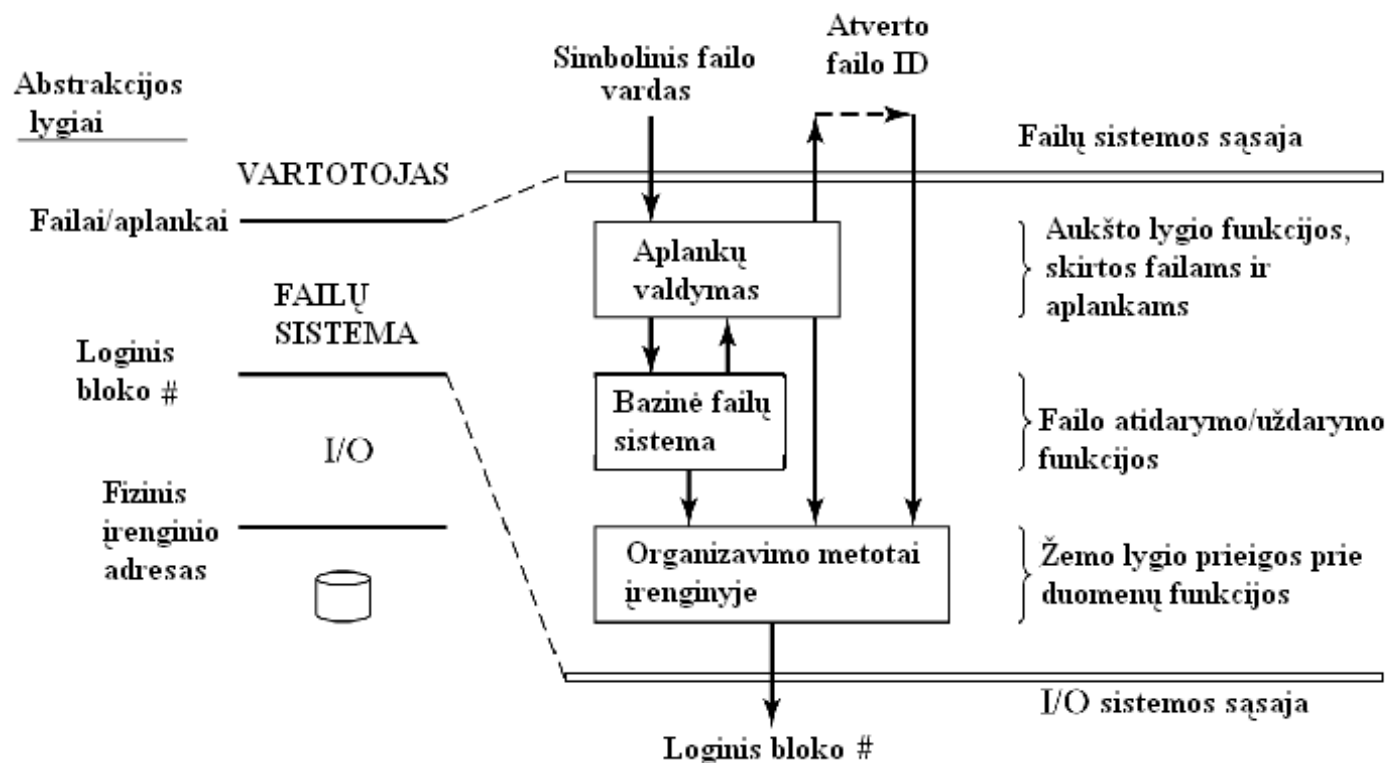
■ Failų sistema apima:

- diske saugomų *failų visumą*;
- *duomenų struktūras*, reikalingas failams tvarkyti (aplankai, failų deskriptoriai, informacija apie vietą diske);
- *failus tvarkančias programas* (skirtas failų kūrimui, naikinimui, kopijavimui, perkėlimui, įrašymui ir t.t.).

Failų sistemos sprendžiami uždaviniai



Abstrakcijos lygiai



Failų sistemos hierarchinis modelis

- Abstrakti vartotojo sąsaja
 - Vartotojui yra pateikiama vartotojo lygio sąsaja, kuria naudodamasis jis gali naudotis *failų* bei *aplankų* sąvokomis operuodamas jomis tiek savo programose, tiek veiksmuose su duomenimis.
- Aplankų (katalogų) valdymas
 - Aplankų valdymo pagrindinė paskirtis yra pagal *loginį failo* vardą pateikti šį failą atitinkantį *unikalų failo identifikatorių*, atrasti failo aprašą (descriptor)
- Bazinė failų sistema
 - Atverti, užverti (open/close) failus
- Fiziniai organizavimo metodai
 - Atvaizduoti failo duomenis į disko blokus

Failas

- Failas vartotojo požiūriu, tai - įrašų grupė, turinti savo vardą ir laikoma ar apdorojama kaip visuma.
- Failo vardas ir tipas
 - Galiojantis vardas
 - Simbolių kiekis
 - Apatinių-viršutinių raidžių registrai
 - Plėtiniai
 - Surišti su failo tipu
 - Naudoja taikomosios programos
 - Baziniai tipai: tekstiniai, objektiniai, užkrovimo failai; aplankai
 - Taikomųjų programų failai, pav., .doc, .c, .html

Failo charakteristikos:

- Informacija apie failus yra laikoma **aplankų struktūroje** (diske). Tai:
 - Failo tipas – reikalingas tik sistemoms, kurios palaiko skirtingus failų tipus
 - Failo savininkas,
 - Šie duomenys naudojami apskaitai, prieigos prie failų nustatymui
 - Leidimai prieigai prie failo,
 - Nusako kas ir kokius veiksmus gali atlikti su failu
 - Data, laikas
 - Failo sukūrimo, modifikavimo, prieigos data
 - Failo vieta diske
 - Failo dydis ir kitos

Veiksmai su failais

- įvairūs, kaip, pavyzdžiui:
 - ☐ sukurti failą,
 - ☐ jį išmesti,
 - ☐ ieškoti failo,
 - ☐ atverti failą,
 - ☐ jį uždaryti,
 - ☐ nustatyti jo charakteristikas,
 - ☐ modifikuoti failo turinį.

Failai ir disko blokai

- OS saugo informaciją apie tai, kuriuose disko blokuose yra patalpintas failas.
 - OS naudoja **loginius** blokų numerius, (nuo 0 ir eina iki maksimalaus galimo).
 - Loginiai blokų numeriai yra verčiami į **fizinius** disko adresus, (takeliai, cilindrai, plokštelės, sektoriai).
 - Šį vertimą vykdo diskų tvarkyklės, kurios perduoda atitinkamas žemo lygio specifines įrenginiui komandas disko kontrolieriui.

Prieigos prie failų metodai

- Failų sistemos gali teikti skirtingus prieigos prie failų metodus, tai:
 - ☐ Nuosekli prieiga
 - ☐ Tiesioginė prieiga
 - ☐ Atsitiktinė
 - ☐ Asociatyvi

Asociatyvus išrinkimas

■ Asociatyvi atmintis

- Dar vadinama adresuojama pagal turinį atmintis (Content –addressable memory-CAM)
 - Atmintinė adresuojama pagal turinį ne pagal adresą)
 - Naudojama labai greitos paieškos taikomosioms programoms
 - Pagal pateiktą žodį CAM ieško per visą atmintinę ir radus grąžina adresą, kur tas žodis rastas
 - Asociatyvi atmintis yra brangi

Prieiga prie failų

■ Nuosekli prieiga

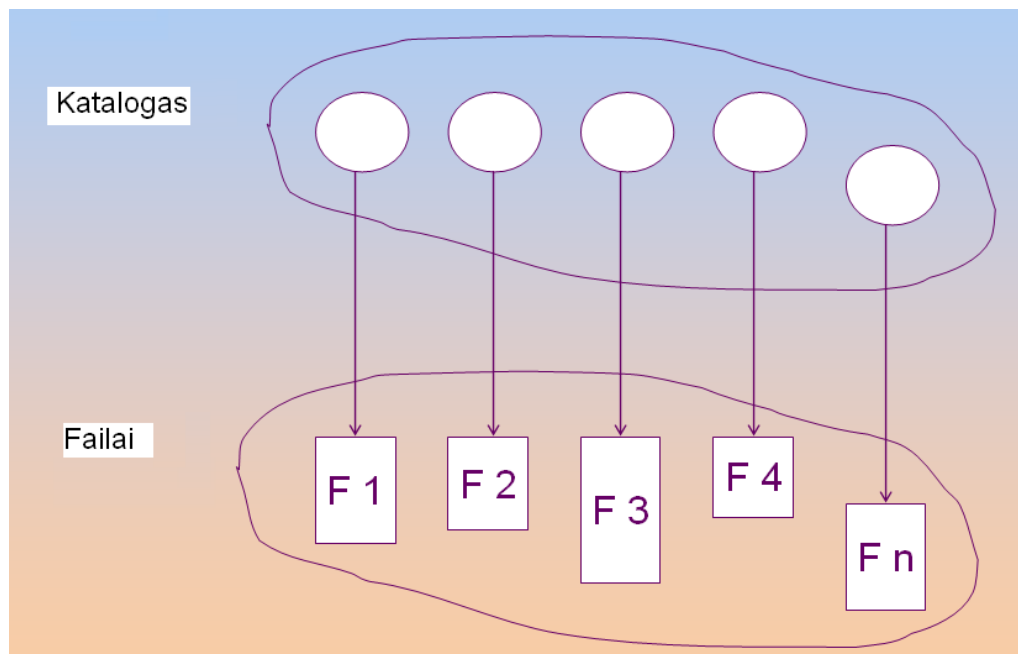
- Visi baitai/jrašai skaitomi nuosekliai nuo pradžios
- Negalima šokti atgal skaitymo metu (reikia persukti juostą)
- Yra taikoma magnetinių juostų atveju

■ Atsitiktinis išrinkimas

- Baitai arba įrašai skaitomi bet kuria tvarka
- Tai svarbu duomenų bazių sistemose

Katalogai (aplankai)

- Pats aplankas taip pat yra failas, tik su specialia struktūra.
 - Kiekvienas įėjimas aplanke – tai simbolinis failo vardas kartu su nuoroda į tai, kur failas yra patalpintas diske.
 - Be to dar gali būti saugoma informacija apie failo dydį, tipą, prieigos galimybes, modifikavimo ir sukūrimo laikai.

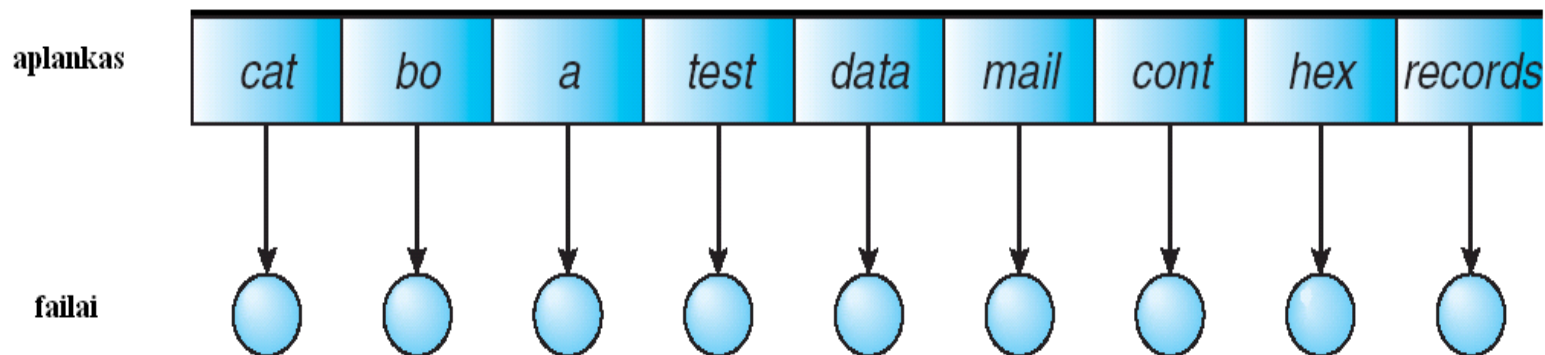


Aplankai (katalogai)

- Aplankai turi dvi paskirtis.
 - Vartotojams jie leidžia struktūrizuoti failų talpinimą failų sistemoje, **grupuojant** juos pagal tam tikras savybes, leidžia skirtinguose aplankuose saugomiems failams naudoti vienodus vardus.
 - Failų sistemos požiūriu aplankai leidžia atskirti **loginę** failų struktūrą nuo **fizinio** failų patalpinimo diske.
- Aplankai teikia informaciją, kurios reikia norint greitai surasti failo duomenų blokus diske.

Aplankų struktūros

- Dauguma failų sistemų palaiko daugelio lygių aplankų struktūras.
- Dažniausiai naudojamos medžio tipo struktūros, nes naudojant vieno lygio struktūrą atsirastų problema įvardinant failus (negalima būtų naudoti tų pačių vardų), negalima būtų panaudoti grupavimo.

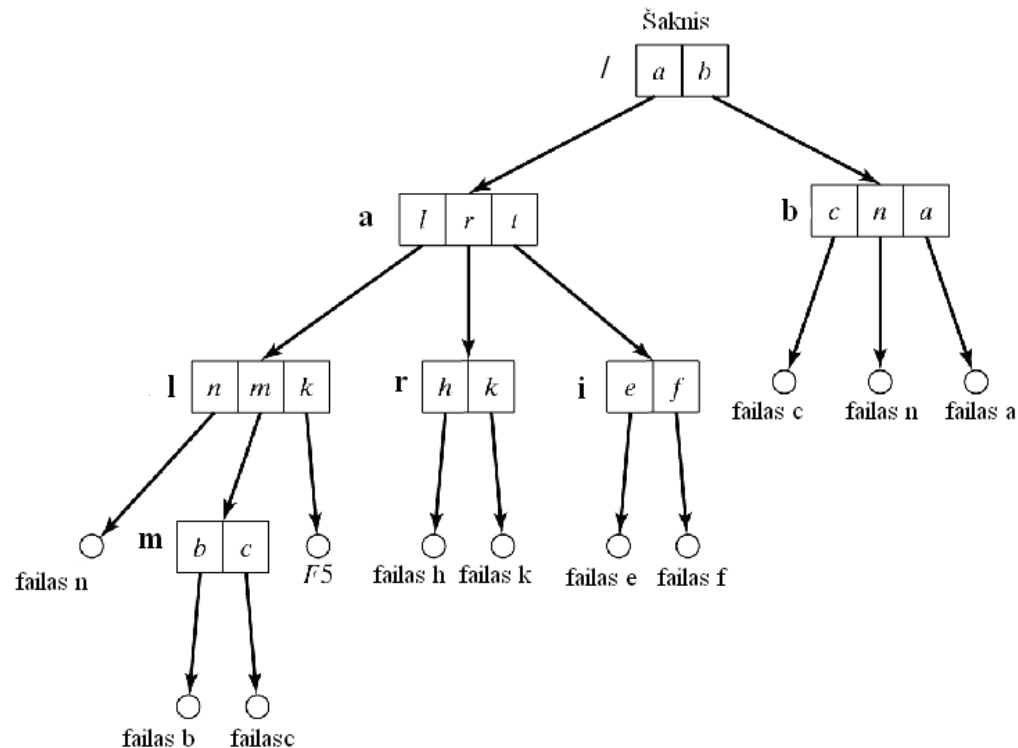


Hierarchinės aplankų struktūros

Desktop

- My Computer
 - 3 1/2 Floppy (A:)
 - (C:)
 - Program Files
 - Windows
 - ...
 - (D:)
 - ...
- Network Neighborhood
- Recycle Bin
- ...

a)



b)

Operacijos su aplankais

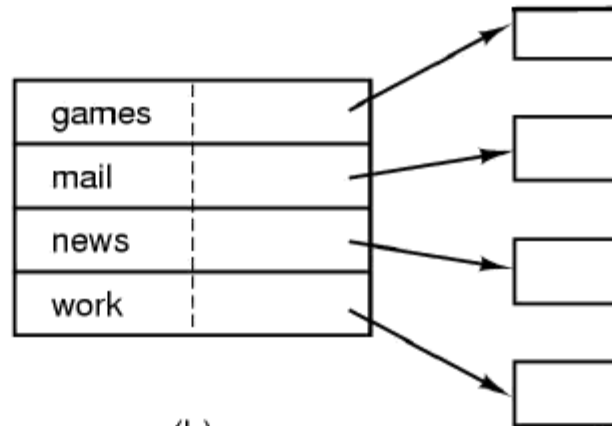
- Galimos panašios operacijos kaip ir su failais:
 - ☐ juos galima sukurti,
 - ☐ išmesti,
 - ☐ galima peržiūrėti aplankų turinį,
 - ☐ galima pereiti į kitą aplanką naudojant kelius (santykinius arba absoliučius),
 - ☐ galima pervardinti,
 - ☐ perkelti aplankus,
 - ☐ pakeisti leidimus prieigai,
 - ☐ sukurti simbolinius ryšius į kitus aplankus.

Aplankų įrašuose saugoma informacija

- Galima saugoti tik simbolinius failų vardus ir metaduomenis
- Galima saugoti tik simbolinius failų vardus ir nuorodą į aprašą.
 - Tokiu atveju reikia papildomo kreipinio į diską, norint paimti informaciją, esančią failo deskriptoriuje.

games	attributes
mail	attributes
news	attributes
work	attributes

(a)



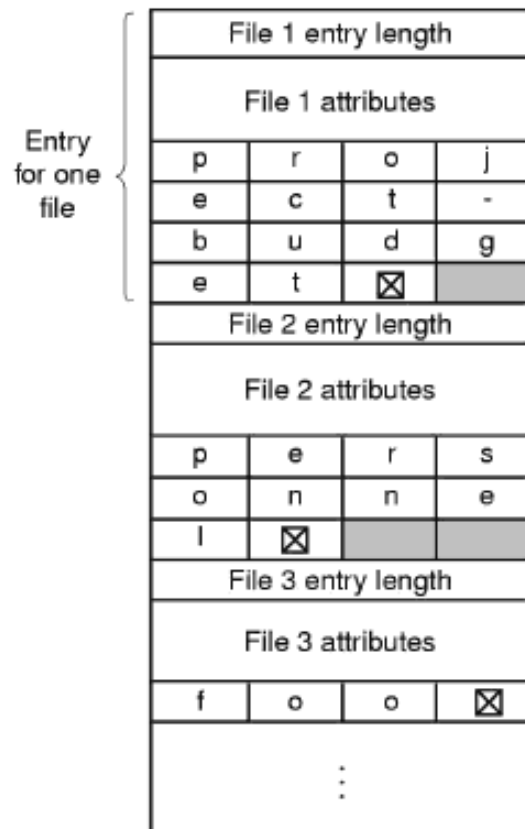
(b)

Data structure
containing the
attributes

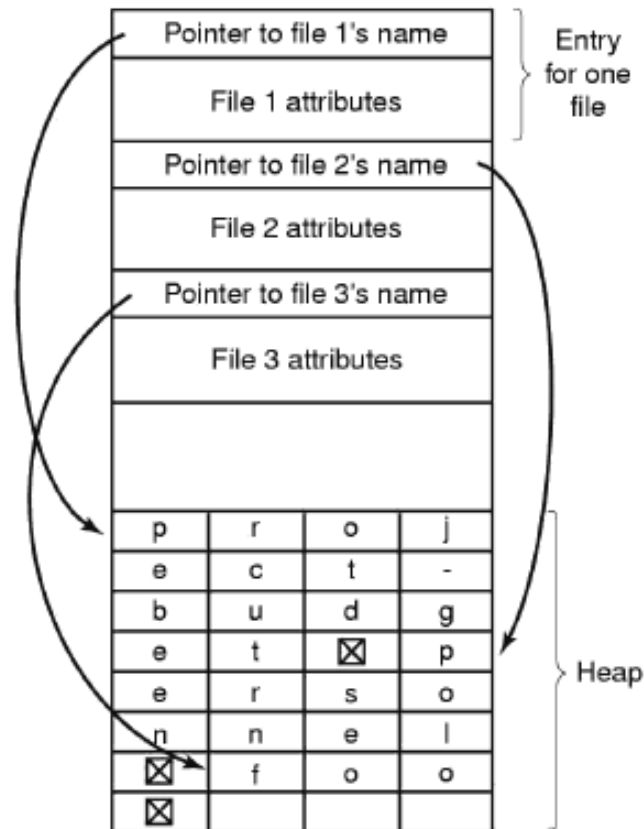
Aplankų įrašuose saugoma informacija

- Įėjimų į aplanką saugojimui irgi gali būti parenkami įvairūs variantai:
 - ☐ Tai gali būti ir fiksuoto ilgio masyvai,
 - ☐ surišti sąrašai,
 - ☐ „hash“ lentelės
 - ☐ B medžiai.
- Bazinė failų sistema, vykdydama failų atidarymo/uždarymo veiksmus paima arba nustato failą aprašančią informaciją, naudojamą efektyviai prieigai prie failo.

Aplankai ir ilgi failų vardai



(a)



(b)

Bazinė failų sistema

■ Bazinė failų sistema

- Aktyvuoja arba deaktivuoja failus juos atverdama arba užverdama
 - Vykdo failų atidarymo/uždarymo (open/close) veiksmus
 - Jei reikia vykdo vartotojo prieigos teisių tikrinimą
 - Paima arba nustato failą aprašančią informaciją, kuri naudojama prieigai prie failo.

■ Failo aprašas (i-node Unix)

- savininko id
- Failo tipas
- Apsaugos informacija
- Atvaizdavimas į fizinius disko blokus
- Sukūrimo laikas, paskutinio panaudojimo laikas, paskutinės modifikacijos laikas
- Kreipinių (ryšių) kiekis

Bazinė failų sistema

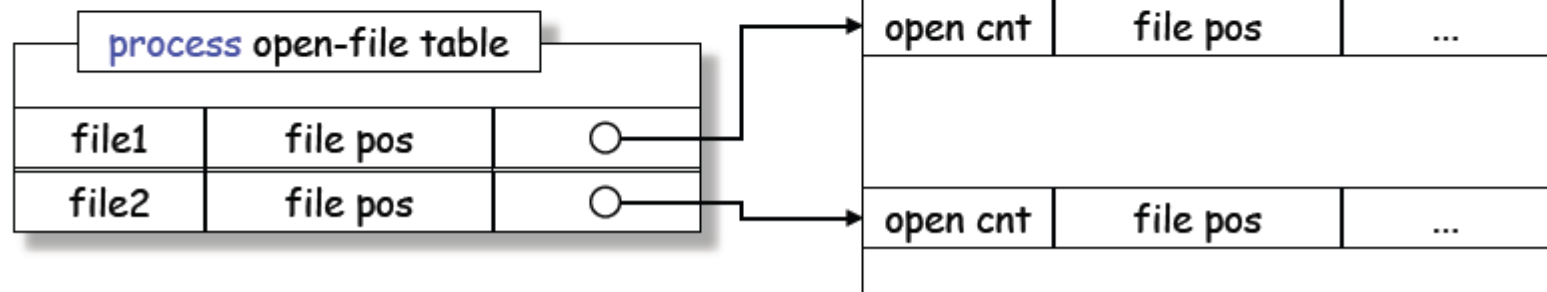
- Atvertų failų lentelė (OFT)
- open komanda:
 - Tikrina prieigos teises
 - Priskiria vietą atvertų failų lentelėje
 - Priskiria (read/write) buferius
 - Užpildo OFT lentelės įrašą
 - inicializacija (pav., einamoji pozicija)
 - Informacija iš failo aprašo (pav. failo ilgis, vieta diske, ...,)
 - Nuorodos į priskirtus buferius
 - Gražina OFT indeksą

Failų lentelės

- Vieno vartotojo sistema

open-file table		
file1	file pos	file location
file2	file pos	file location

- Daugelio vartotojų sistema



Bazinė failų sistema

■ close komanda:

- Perkelia modifikuotus buferius į diską
- Atlaisvina buferius
- Atnaujina failo aprašą:
 - Failo ilgis, vieta diske, apskaitos informacija
- atlaisvina OFT įėjimą

Failų fizinė organizacija

- Fizinė failų organizacija aprašo failo išdėstymo išorinėje atmintyje, pvz., diske, taisyklės.
- Failas susideda iš fizinių įrašų – blokų.
- Blokas – tai mažiausias duomenų vienetas, kuriuo išorinis įrenginys apsieičia su pagrindine atmintine.
- Priskiriant disko blokus failams yra siekiama efektyviai išnaudoti disko erdvę.

Fizinio organizavimo metodai

- Problema – paskirti failams vietą diske: Failai – tai blokų eilė
 - Disko erdvė turi būti išnaudota efektyviai
 - Failus turi būti lengva išrinkti iš disko
- Naudojamos talpinimo strategijos:
 - Į nuosekliai einančius blokus.
 - Į blokus, sujungiant juos į sąrašą.
 - Į blokus, formuojant indeksų sąrašą

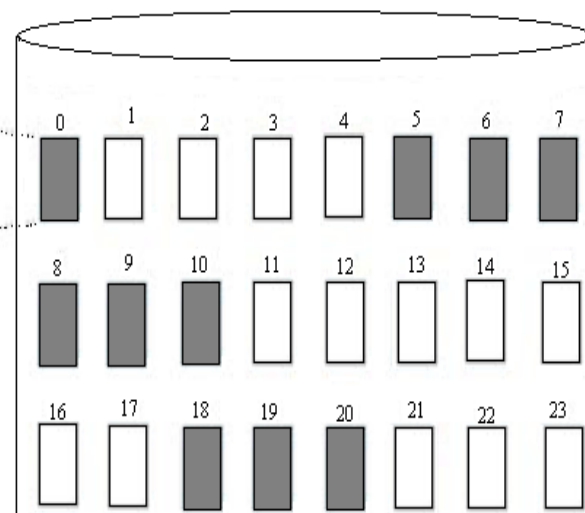
Fizinio organizavimo metodai: nuoseklus talpinimas

- Failas diske talpinamas į nuosekliai einančius disko blokus. Šis talpinimo metodas reikalauja failams priskirti tiek nuosekliai einančių blokų, kiek reikalauja to failo dydis

- Paprastai įgyvendinamas
- Greita **nuosekli** prieiga (minimalus galvutės judesys)
- Sunku atlikti įterpimą/išmetimą
- Sunku nuspręsti, kiek vietos priskirti pradžioje
- Išorinė fragmentacija galima

Aplankas

Failas	Pradžia	Dydis
CIT593doc	5	6
..		
CIT595doc	18	3



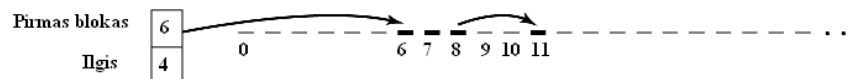
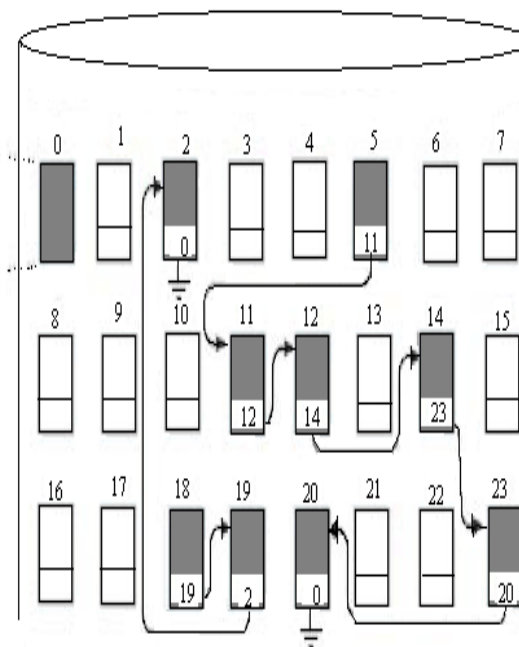
Fiziniai organizavimo metodai: surištas sąrašas

- Šiuo atveju aplanke yra nurodoma failo pradžios blokas, o jau tame bloke patalpinama nuoroda į sekantį failo bloką

- Paprasta įterpti/išmesti, nėra išorinės fragmentacijos
- Nuoseklus išrinkimas mažiau efektyvus (paieška, vėlinimas)
- Nėra galimas tiesioginis kažkurio bloko išrinkimas
- Mažas patikimumas (suirus grandinėlei)

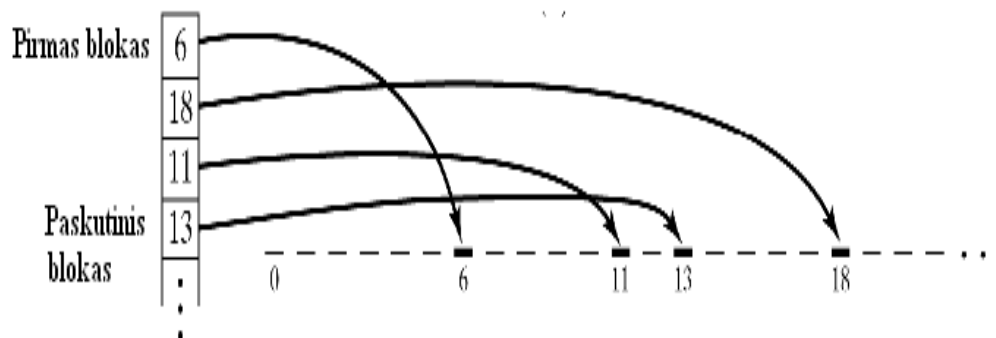
Aplankas

Failas	Pradžia
CIT593doc	5
..	
CIT595doc	18



Fiziniai organizavimo metodai: indeksinė organizacija

- suformuojama indeksų lentelė, rodanti, kuriame fiziniame bloke yra patalpintas atskiras failo blokas .
- Paprasčiausiu atveju ši indeksų lentelė gali būti laikoma failo apraše.
- Esant tokiam talpinimui paprasta atlikti įterpimo, išmetimo veiksmus, galimas tiek **nuoseklus**, tiek **tiesioginis** failo blokų išrinkimas.
- Trūkumas pasireiškia tame, kad failo dydžius gali riboti indeksų lentelės dydžiai.



Fiziniai organizavimo metodai: indeksinė organizacija

■ Indeksavimo variantai

□ Kelių lygių indeksų hierarchija

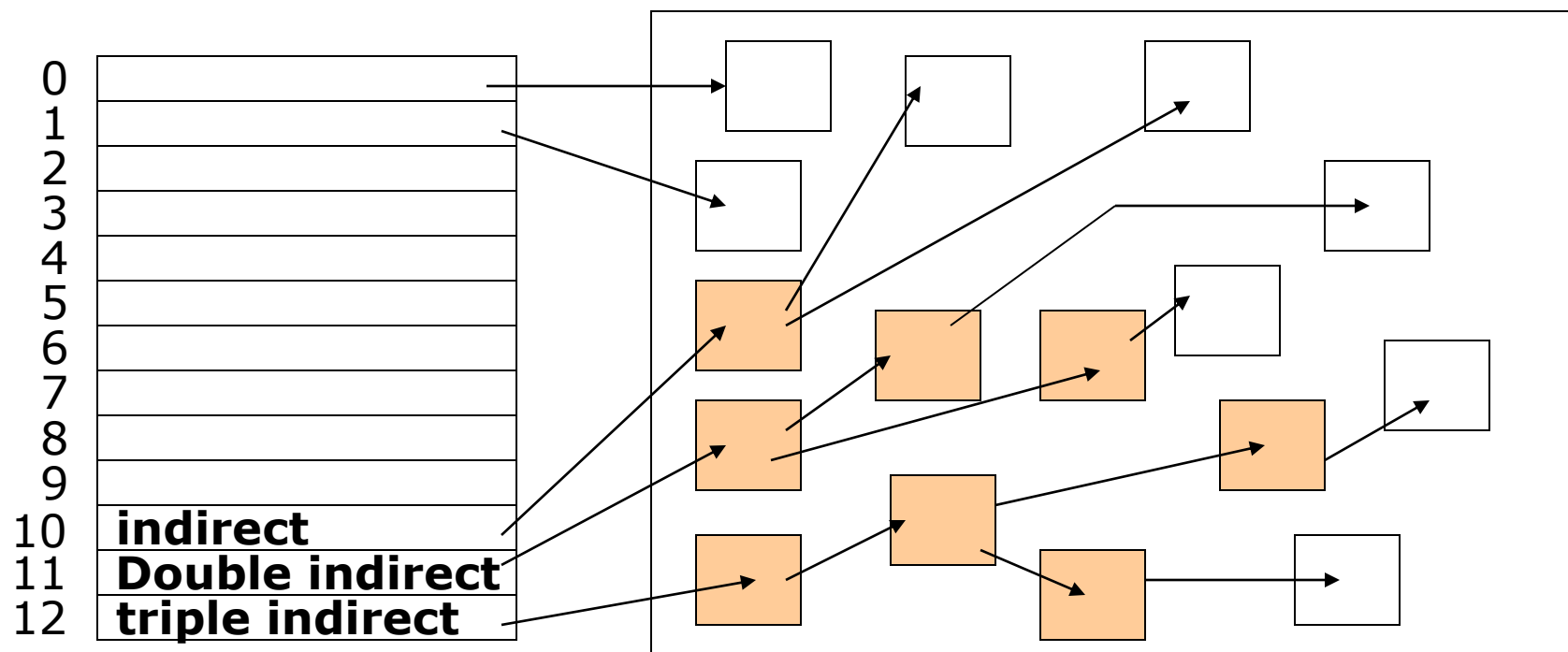
- pirmo lygio indeksai rodo **ne tiesiogiai** į disko blokus, o į antro lygio indeksus
- problema: auga kreipinių į diską kiekis augant hierarchijos lygių kiekiui

□ priauginimo tipo indeksacija

- dalis indeksų rodo tiesiogiai į disko blokus, o jei jų nepakanka, yra įtraukiami papildomi indeksų lygiai, panaudojama dviguba, triguba indeksacija

□ Tarkim jei indekso bloke galim turėti 256 nuorodas, tai bus 65536 nuorodų esant dviejų lygių indeksams, 1K blokas, 64M failas

Adresacija UNIX

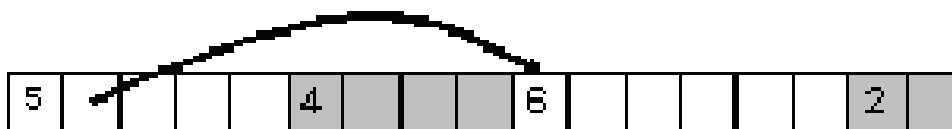


Laisvos disko atmintinės valdymas

- Panašios problemos kaip ir valdant pagrindinę atmintinę
 - Saugoma informacija apie laisvus blokus **surišto sąrašo** metodu
 - Surišti individualius blokus mažai efektyvu :
 - Nėra taikomas blokų apjungimas kad būtų mažinamas paieškos laikas
 - Surišamos nuoseklių blokų grupės
 - Grupė blokų yra priskiriama arba atlaisvinama vienu metu
 - Dvejetainio žemėlapiu formavimas (bitmap)
 - Analogiškai kaip ir pagrindinės atmintinės atveju.
 - Trūkumas dvejetainio žemėlapiu yra tame, kad ieškant laisvo bloko failų sistemai gali tekti peržiūrėti visą dvejetainį žemėlapią
 - Reliatyviai paprastas metodas – lengva rasti n laisvų gretimų blokų
 - Naudoja :Intel 80386, Motorola 68020/30, Apple Macintosh
 - Nėra efektyvu, jei visas žemėlapis nėra laikomas pagrindinėje atmintinėje
 - Sunku naudoti, jei diskai yra labai didelės talpos

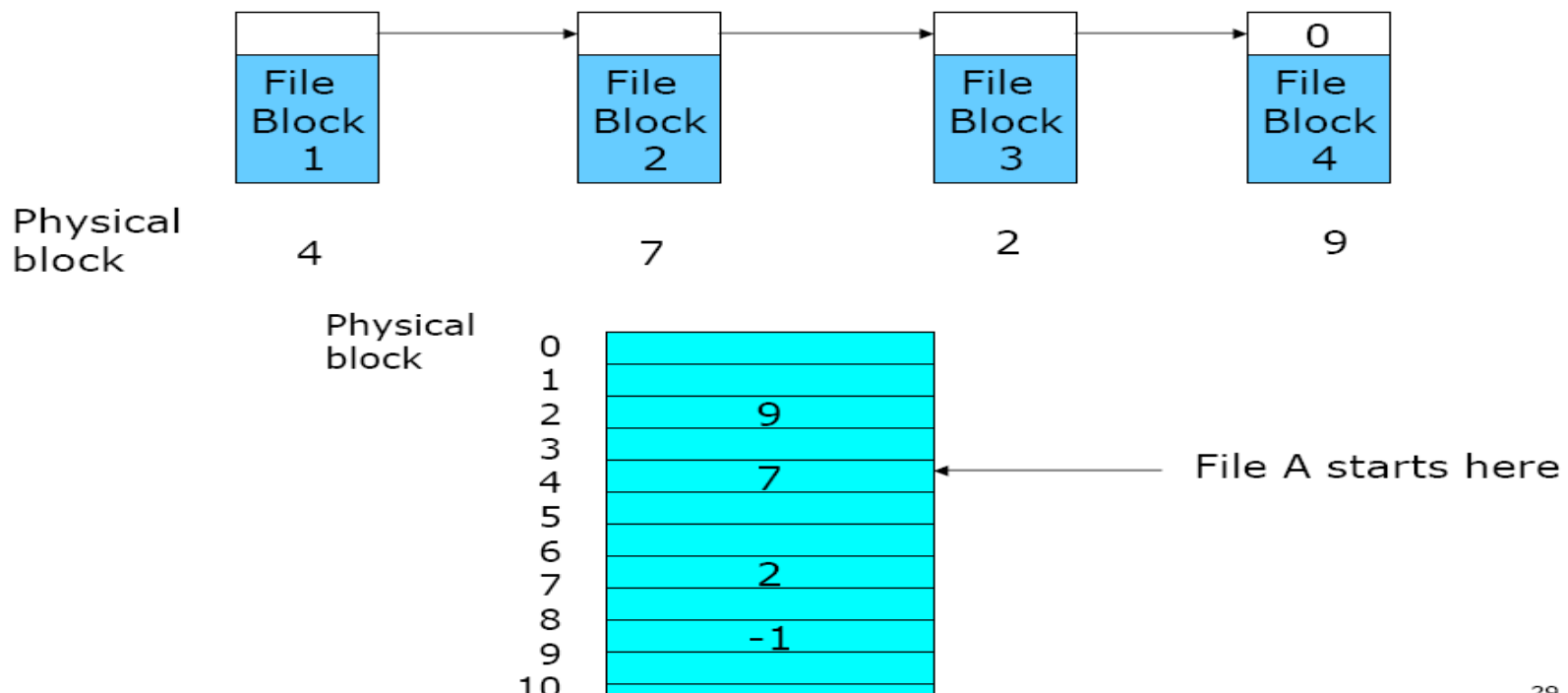
Laisvos disko atmintinės valdymas

- Nėra tikslinga surišti į sąrašą visus atskirus laisvus blokus po vieną.
- Jei yra keli iš eilės einantys laisvi bloakai, tai jie galės būti ir priskiriami esant poreikiui, todėl laisvų blokų sąrašas ir formuojamas surišant į sąrašą laisvų blokų grupes
- Laisvi bloakai yra priskiriami failams imant juos iš sąrašo pradžios, o atsilaisvinę bloakai yra įjungiami į šio sąrašo pabaigą

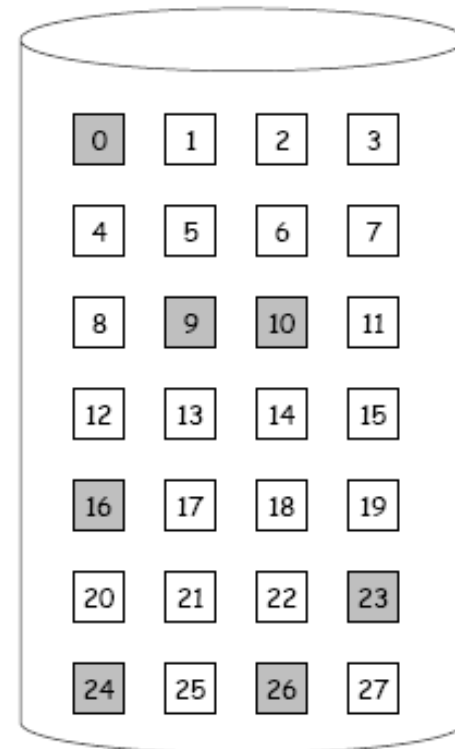
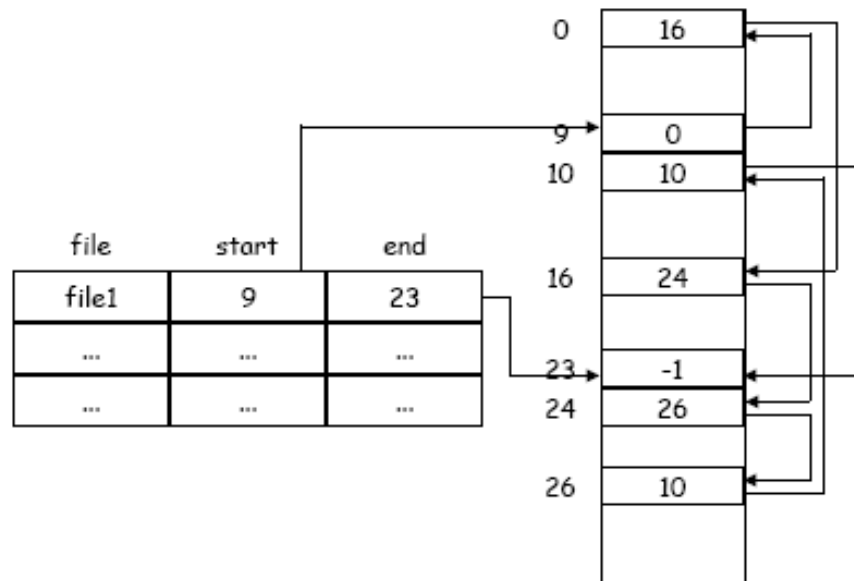


Windows failų sistema

- Įrašus apie talpinimo vietą deda į failų talpinimo lentelę, vadinamą FAT (File Allocation Table).



FAT



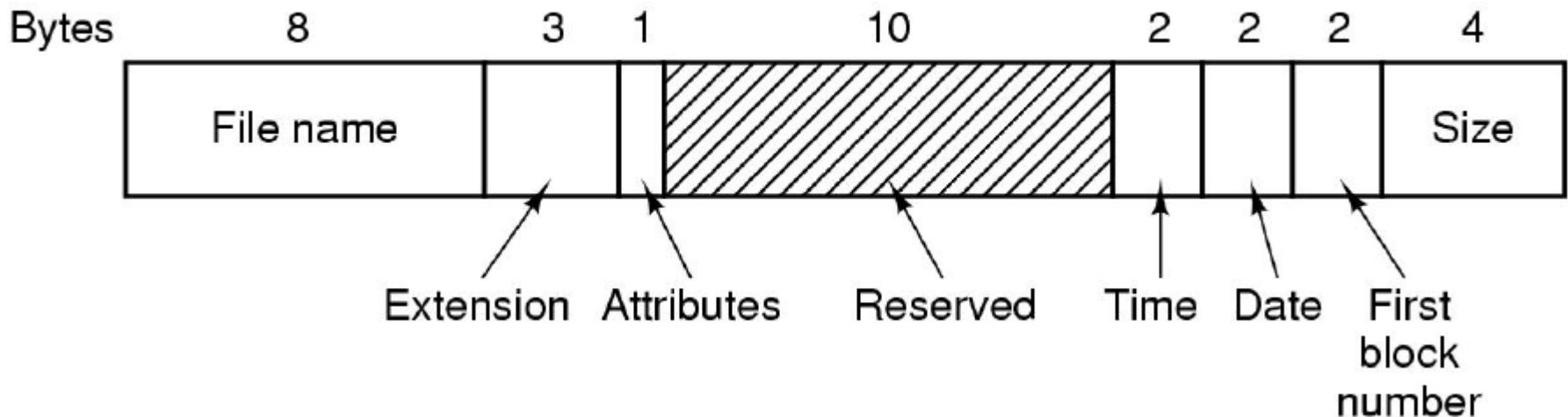
File-Allocation Tables (FAT)
MS-DOS, OS/2.

FAT

- FAT privalumai yra tame, kad informacija apie disko blokų užimtumą yra saugoma nedidelėje diske esančioje lentelėje.
- Lentelė indeksuojama blokų numeriais – vienas įėjimas-vienas blokas.
- Užimti blokai tarpusavyje surišami sąrašu
- Neužimti blokai indikuojami 0-nės reikšmės įėjimu.
- Patogus yra laisvų blokų suradimas ir priskyrimas failui – šiuo tikslu pakanka surasti neužimtus FAT lentelės įrašus.
- Kiek sudėtingesnis yra nuoseklus failo blokų skaitymas.
- naudoja MS-DOS ir OS/2

MS DOS direktorijos įėjimas

- Naudoja FAT12 arba FAT16 lenteles
- Naudoja 8+3 simbolių failų vardus
- Failų atributai: read only, archived, hidden, system



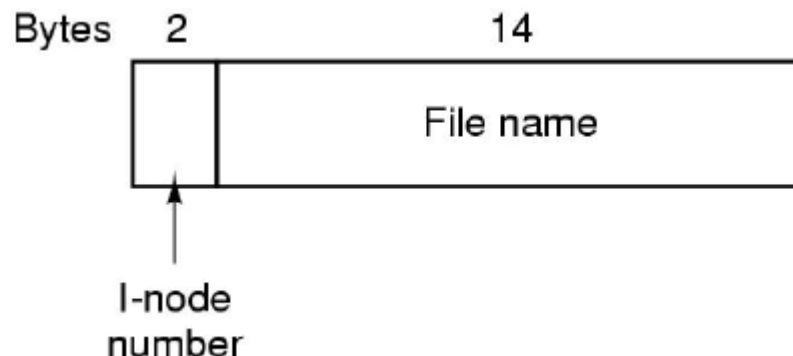


Windows 98 failų sistema

- Naudoja FAT 32 failų patalpinimo lenteles
- Naudoja ilgus 255 simbolių failų vardus

Unix/Linux failų sistemos

- Naudoja įvairias failų sistemas:
 - UFS, NFS
 - Ex, ext2, ext3, ext4
 - UV7 failų sistemos direktorijos įėjimas:

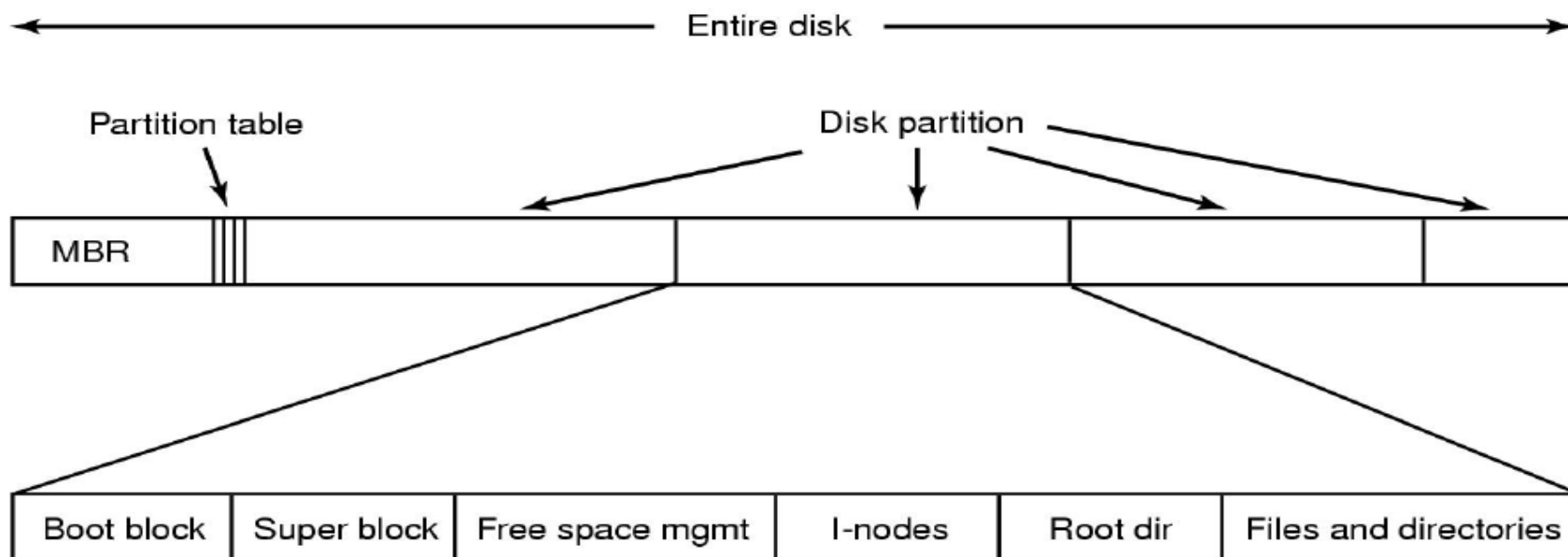


UNIX S5 failų sistemos disko struktūra

Užkrovimo sektorius - operacinės sistemos paleidimo kodas

Superblokas – failų sistemos metaduomenys : failų sistemos dydis, inode sąrašo dydis, skaičius laisvų blokų/ inode blokų.

inode sąrašas- 64baitų ilgio inode struktūrų masyvas



Žurnalinė failų sistema (JFS)

- Operacijos, apie kurias rašomi įrašai:
 - – create, link, mkdir, truncate, allocating write, ...
- Kiekviena sistemos vykdoma operacija gali būti surišta su eilės metaduomenų keitimais (naujinimais).
- Realūs įrašymai į diską gali būti vykdomi :
 - asinchroniškai,
 - prieš darant įrašą

Žurnalus naudojančios failų sistemos

- Failų sistemose, kurios naudoja žurnalus, visi vykdomi pakeitimai yra fiksuojami žurnaluose.
- Šie įrašai paprastai laikomi **atskirame** failų sistemos skyriuje.
 - Įrašas žurnale yra daromas prieš atliekant realų failų sistemos duomenų struktūros pakeitimą.
 - Po to yra atliekami pakeitimai realiose duomenų struktūrose
 - ir tai užbaigus yra padaromas įrašas apie veiksmų pabaigą.

Žurnalus naudojančios failų sistemos

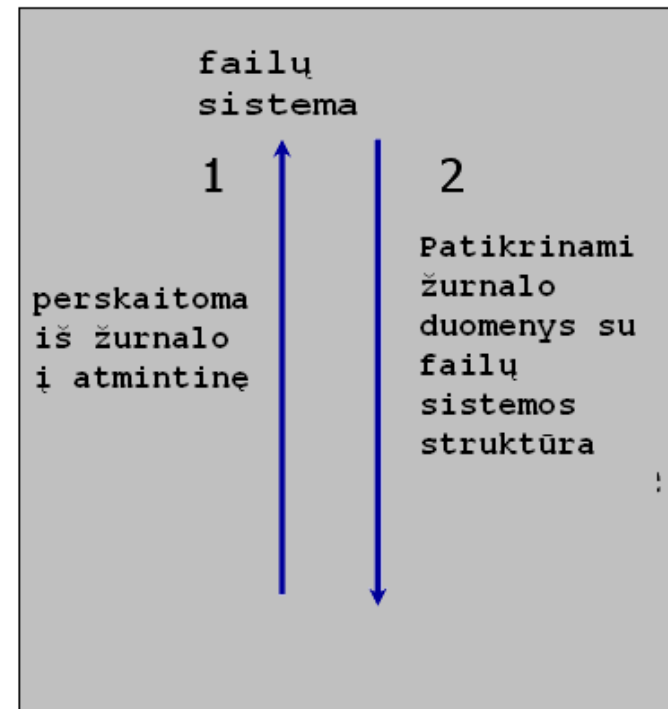
- Kokie duomenys gali būti rašomi į žurnalus?
- Yra dvi galimybės:
 - pilnas duomenų įrašas
 - į žurnalą yra įrašomi visi duomenys
 - Tai duoda didelę failų sistemos neprieštaringumo garantiją
 - reikalauja daug pastangų ir darbo sąnaudų
 - metaduomenų įrašas.
 - yra greitesnis,
 - įrašomi tik metaduomenys (failo ar aplanko).

Žurnalų palaikymas

- Išlaikoma nuorodos (cursor) vieta žurnale
- Ji yra pakeičiama tik tada, kai pakeisti blokai, susiję su šia operacija yra įrašomi į diską
- Šie įrašai gali būti ištrinami iš žurnalo
- Atstatymo atveju visos operacijos nuo paskutinės nuorodos reikšmės yra vykdomos iš naujo.

Žurnalų panaudojimas

- Sumontavus failų sistemą, failų sistemos tvarkyklė atlieka patikrinimą, nustatydama ar failų sistema yra stabili.
- Jei dėl kažkurių priežasčių reikia sutvarkyti metaduomenis, vietoje to kad atlikti pilną meta-duomenų skanavimą kaip fsck, yra peržiūrimas žurnalas.



Žurnalų panaudojimas

- Kadangi žurnale yra chronologiški logai visų nesenų meta-duomenų pakeitimų, tai ji paprasčiausiai patikrina tuos meta-duomenų keitimo įrašus, kurie neseniai yra užregistruoti.
- Šis procesas yra žymiai greitesnis, nei pilnos failų sistemos struktūros analizė.
- Taigi sistema gali būti atstatoma per keletą sekundžių ir vėl tampa prieinama klientams.

Žurnalinės failų sistemos

■ Privalumai:

- ☐ Asinchroninis metaduomenų rašymas
- ☐ Greitas atstatymas: jis priklauso tik nuo žurnalo dydžio o ne nuo failų sistemos dydžio

■ Trūkumai

- ☐ Papildomi įrašymai į diską
 - ☐ Reikia vietos žurnalų saugojimui (nereikšminga)
- Dauguma modernių Linux failų sistemų naudoja žurnalines failų sistemas.

Žurnalus naudojančios failų sistemos

- Linux operacinės sistemos atveju galimas pasirinkimas tarp keturių žurnalinių failų sistemų: Ext3, XFS, JFS, ir ReiserFS.
- Tik metaduomenis įrašančios žurnalinės failų sistemos, tai pavyzdžiui: Ext3fs, JFS, NTFS, Reiserfs, XFS.
- Ext3fs failų sistema gali būti duomenis ir metaduomenis įrašančios failų sistemos pavyzdžiu.

XFS failų sistema

- XFS (high-performance journaling ***file system***) yra žurnalinė failų sistema, sukurta 1994m SGI kompanijoje Unix system-V sistemai.
 - Ji atsirado dėl poreikio :
 - didelei disko talpai ir jo pralaidumo.
 - greito sistemos atsistatymo po nusilaužimo poreikis
 - poreikis didelių failų sistemų,
 - direktorijų su dideliu failų kiekiu poreikis.
- XFS yra taip pat ir Linux versija

XFS savybės

- 18000 Petabaitų failų sistemos dydis (1PB=10⁶ GB), 9000 PB failo dydis
- Įrašai apie failų sistemos metaduomenis daromi į žurnalą **asinchroniškai**, sistema **atsistato** po keleto sekundžių
- Direktorijų medis B+, failų vardai konvertuojami į 4 baitų hash reikšmes naudojamas indeksuojant direktoriją, greita paieška direktorijoje.

XFS savybės

- Naudoja extent (plėtinius):
 - Plėtiniai – tai nuosekliai einantys loginiai disko blokai
 - Plėtinys aprašomas trim elementais: pradžia, plėtinio ilgis ir poslinkis)
 - Plėtinio dydis 512B- 1GB
- Jei yra naudojama Raid (5) technologija tai sudalinimo dydis (stripe) gali būti nusakomas failų sistemos montavimo metu
- Atidėtas vietos diske išskyrimas failui

Ext4 failų sistema

- Tai žurnalinė Linux failų sistema, stabili versija nuo 2008m
- Gali palaikyti skirsnius, kurių dydis iki 1 EB
 - 1EB atitinka 10^{18} B
- Failų dydžiai iki 16 terabaitų
- Failų atvaizdavimas disko blokais pakeistas atvaizdavimu plėtiniais (extent) :
 - vienas ext4 plėtinys gali atitikti iki 128 MB nuosekliai einančios disko erdvės
 - i-node bloke iki 4 plėtinių, jei jų yra daugiau, jų indeksavimui naudojamas H medis
 - Failams leidžiama iš anksto palikti erdvės diske (preallocation)
- EXT4 naudoja “allocate-on-flush” techniką, kuri dar vadinama suvėlintu vietoje diske išskyrimu

Ext4 failų sistema

- **Pakatologių** kiekis išplėstas iki 64000
- Taikomas *kontrolinių sumų* skaičiavimas žurnalui
- Greitesnis failų sistemos tikrinimas:
 - Nepriskirtos blokų grupės ir i-node blokai atžymėti kaip tokie ir nereikalauja tikrinimo
- Sumažinta fragmentacija
 - Vieta skiriama ne po vieną bloką o po keletą blokų
- Pagerinta laiko atžyma – išplėsta iki nanosekundžių eilės, dadėti 2 bitai prie reikšmingiausių sekundžių lauko skilčių kad būtų galima apeiti 2038metų problemą.



NTFS failų sistema

- (New Technology Filesystem)

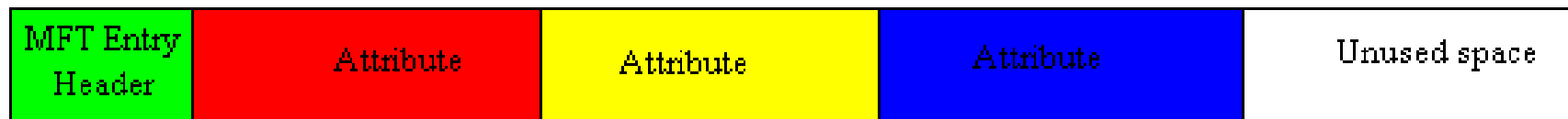
NTFS architektūra

NTFS užkrovimo sektorius	"Master" failų lentelė	Failų sistemos duomenys	"Master" failų lentelės kopija
--------------------------------	---------------------------	-------------------------	-----------------------------------

- NTFS failų sistema kiekvieną failą (arba katalogą) traktuoja kaip *failo atributų rinkinį*.
- **Atributais** g.b. tokie elementai kaip failo vardas, informacija susijusi su failo apsauga, ir net patys failo duomenys.
- Kiekvienas atributas identifikuojamas jo kodu (arba vardu).
- Kai failo atributai telpa į MFT failo įrašą, jie vadinami **rezidentiniais**, pvz failo vardas, laiko atžyma (time stamp).
- Atributai, kurie netelpa į įrašą yra nerezidentiniai ir jie talpinami viename ar keliuose disko "klasteriuose"
 - NTFS sukuria atributą **List** aprašymui, kur yra patalpinti visi failo atributų įrašai.

MFT įrašo struktūra

- Kiekvienas įėjimas yra 1KB dydžio
- Įėjime saugoma
 - Failo atributai
 - Talpinimo duomenys



MFT įrašai

- Maži failai (<900B) yra pilnai patalpinami MFT įrašė.
- Standartinė informacijaryšių kiekis, laiko atžyma
- Indeksinė struktūra – tai atvaizdavimas į kitus klasterius (naudojami B medžiai)

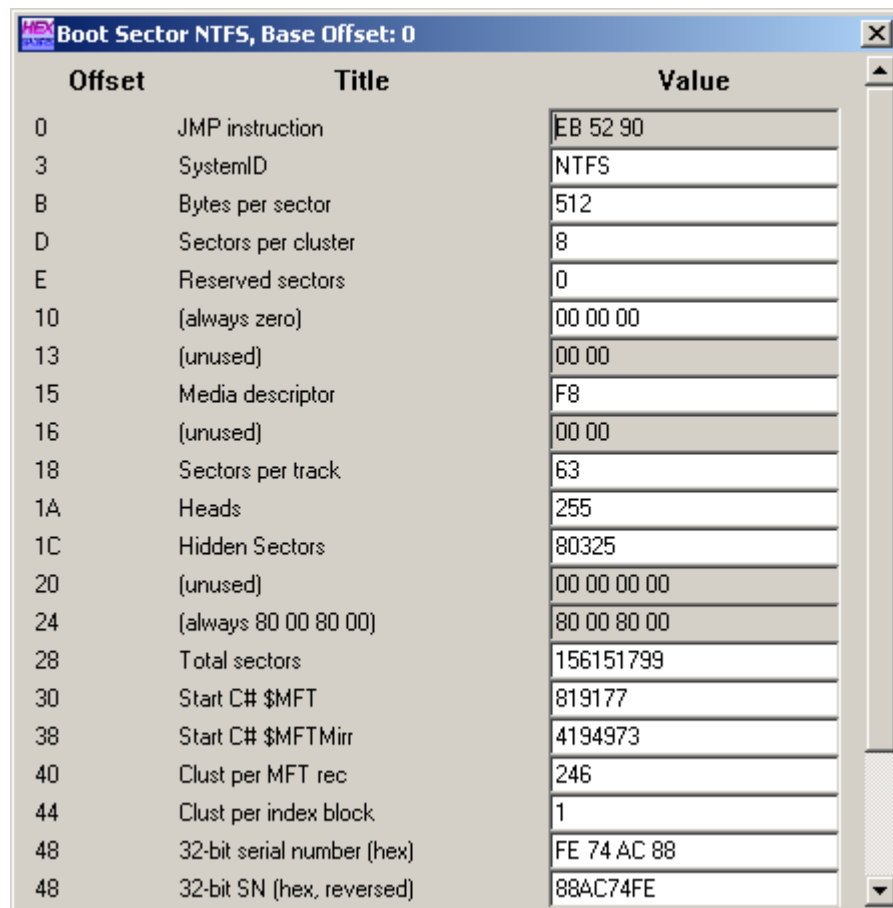
Standartinė informacija
failo ar katalogo vardas
Duomenys arba indeksai
Nenaudojama erdvė

Naudojami atributai

Table 5-3 File Attributes Defined by NTFS

Attribute Type	Description
Standard Information	Includes information such as timestamp and link count.
Attribute List	Lists the location of all attribute records that do not fit in the MFT record.
File Name	A repeatable attribute for both long and short file names. The long name of the file can be up to 255 Unicode characters. The short name is the 8.3, case-insensitive name for the file. Additional names, or hard links, required by POSIX can be included as additional file name attributes.
Security Descriptor	Describes who owns the file and who can access it.
Data	Contains file data. NTFS allows multiple data attributes per file. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes, each using a particular syntax.
Object ID	A volume-unique file identifier. Used by the distributed link tracking service. Not all files have object identifiers.
Logged Tool Stream	Similar to a data stream, but operations are logged to the NTFS log file just like NTFS metadata changes. This is used by EFS.
Reparse Point	Used for volume mount points. They are also used by Installable File System (IFS) filter drivers to mark certain files as special to that driver.
Index Root	Used to implement folders and other indexes.
Index Allocation	Used to implement folders and other indexes.
Bitmap	Used to implement folders and other indexes.
Volume Information	Used only in the \$Volume system file. Contains the volume version.
Volume Name	Used only in the \$Volume system file. Contains the volume label.

NTFS užkrovimo sektoriaus informacija



The screenshot shows a window titled "Boot Sector NTFS, Base Offset: 0". It contains a table with three columns: "Offset", "Title", and "Value". The table lists various fields from the NTFS boot sector, including instructions, system IDs, sector and cluster counts, reserved sectors, media descriptors, hidden sectors, total sectors, and start cluster numbers for the MFT and its mirror. The values are displayed in hexadecimal or decimal as appropriate.

Offset	Title	Value
0	JMP instruction	EB 52 90
3	SystemID	NTFS
8	Bytes per sector	512
D	Sectors per cluster	8
E	Reserved sectors	0
10	(always zero)	00 00 00
13	(unused)	00 00
15	Media descriptor	F8
16	(unused)	00 00
18	Sectors per track	63
1A	Heads	255
1C	Hidden Sectors	80325
20	(unused)	00 00 00 00
24	(always 80 00 80 00)	80 00 80 00
28	Total sectors	156151799
30	Start C# \$MFT	819177
38	Start C# \$MFTMirr	4194973
40	Clust per MFT rec	246
44	Clust per index block	1
48	32-bit serial number (hex)	FE 74 AC 88
48	32-bit SN (hex, reversed)	88AC74FE

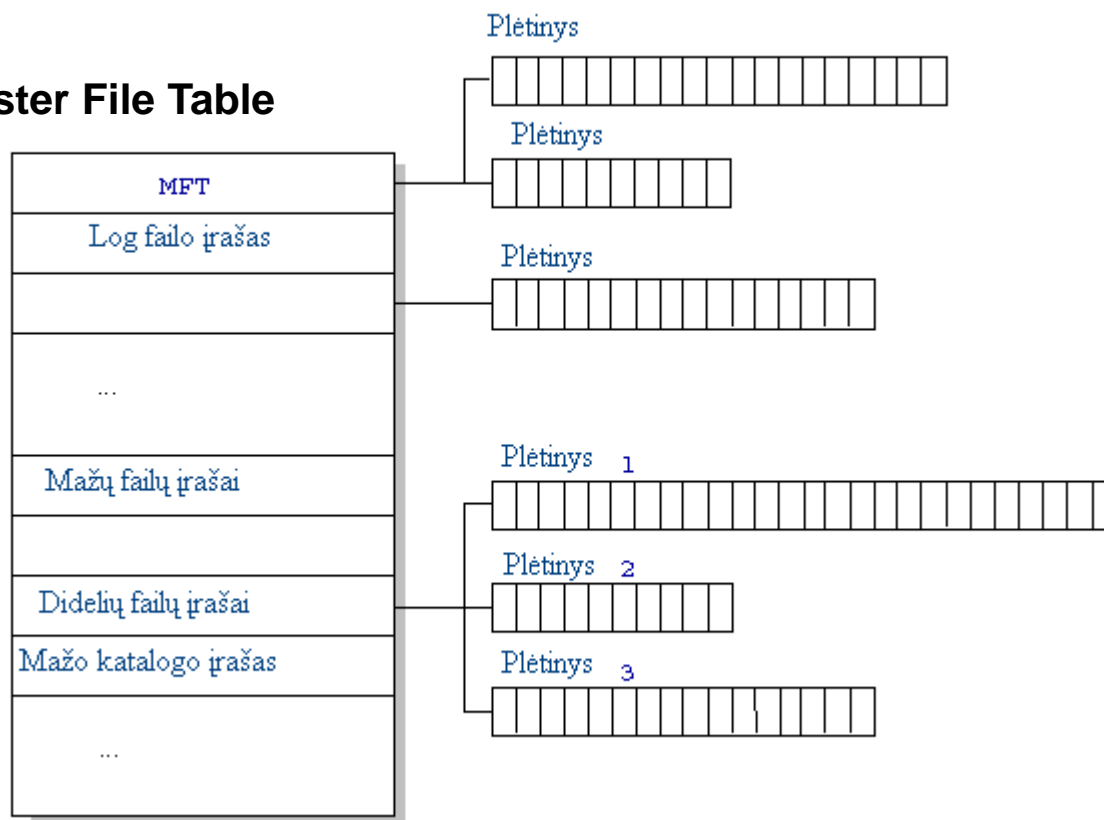
NTFS “Master” failų lentelė

Ji turi 16 sisteminių įrašų

1. Įrašas aprašo pačią master failų lentelę \$MFT failas.
2. Įrašas tai 1-o įrašo kopija (mirror) \$MftMirr failas.
3. Log įrašų failas \$LogFile.
4. Disko skirsnio (Volume) aprašymo failas \$Volume.
5. Atributų aprašymų lentelė – failas \$AttrDef
6. Šakninis katalogas “.” failas \$
7. Klasterių dvejetainio žemėlapijo (bitmap) \$Bitmap failas.
8. Užkrovimo sektorių aprašantis failas \$Boot
9. Blogų klasterių failas \$BadClus
10. Apsaugos failas \$Secure
11. Upcase table \$Upcase (Failas, surištas su mažųjų raidžių konvertavimu į Unikodą
12. NTFS plėtinių failas \$Extend, (pvz g.b. kvotos).
13. 13-16 įrašai rezervuoti ateičiai

NTFS struktūra

Master File Table



Duomenų atstatymas naudojant NTFS

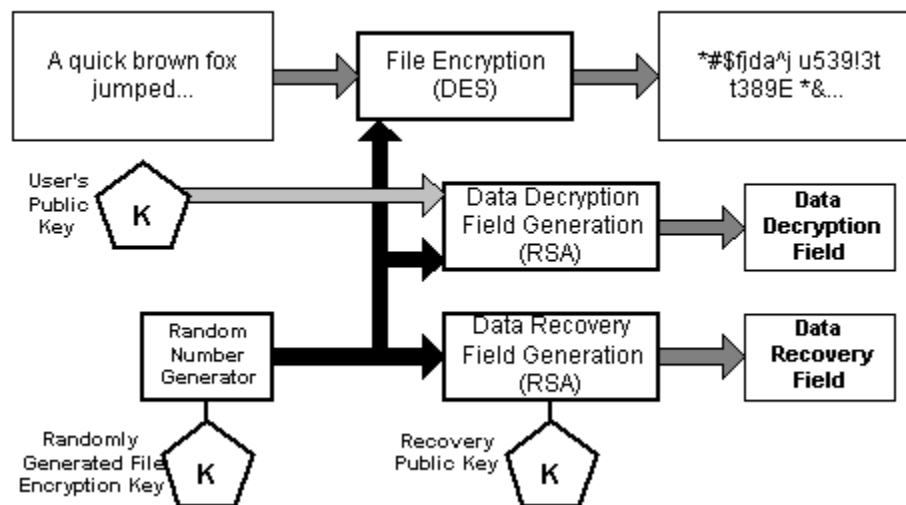
- NTFS traktuoja kiekvieną I/O operaciją, kurios metu modifikuojamas sisteminio failo turinys kaip vieningą užduotį, kuri gali būti **pilnai atlikta** arba kurios veiksmas **grąžinti į išeities padėtį**, esant disko nusimušimui ar kito tipo klaidai.
- Šiuo tikslu:
 - Į užduotį įeinantys veiksmas yra įrašomi į log failą prieš juos įvykdant diske.
 - Tada NTFS vykdo šiuos užduoties veiksmus disko cache.
 - Kai NTFS atnaušina cache turinį, užduotis yra baigiama ir log faile įrašoma, kad užduotis yra atlikta.
- Atlikus pilnai užduotį, NTFS garantuoja, kad tie užduoties veiksmas atsiras diske net ir diskui nusimušus.
 - Vykdam atstatymo operacijas, NTFS pakartoja visas užduotis, kurios įrašytos log faile.
 - Jei NTFS randa užduotis log faile, kurios nebuvo užbaigtos iki nusimušant sistemai, tai tų užduočių veiksmas yra grąžinami atgal.
- **Svarbu**
NTFS naudoja veiksmų įrašymą (logging and recovery) ir atstatymą tam, kad būtų galima garantuoti, kad disko struktūra nėra korumpuota.
 - Visi sisteminiai failai išlieka prieinami po sistemos nusimušimo. Tačiau vartotojo duomenys gali būti prarandami.

Failų sistemos šifravimas

- EFS (The Encrypting File System) teikia šifravimo technologiją, skirtą šifruotų failų saugojimui NTFS diskuose.
 - EFS apsaugo failus nuo atakuotojų.
- Vartotojai dirba su šifruotais failais ar katalogais taip pat kaip ir su kitais failais (katalogais)
 - Šifravimas yra skaidrus atžvilgiu vartotojo, kuris darė šifravimą.:
 - sistema atšifruoja failą tam vartotojui, kai jis į jį kreipiasi.
 - Kai failas yra išsaugomas, jis vėl yra šifruojamas.
 - Vartotojai, kurie nėra autorizuoti prieigos prie šifruotų failų atžvilgiu gaus pranešimą "Access denied" jei jie bandys atlikti open, copy, move, or rename šifruotam failui ar katalogui.
- EFS privalumai lyginant su trečios šalies šifravimo paslaugas tiekiančia įranga:
 - Jei failą pažymėjom kaip šifruojamą, jis visad bus užšifruotas, vartotojui nereiks apie tai rūpintis, jam nereikės atsiminti slaptažodžio atšifravimui.
- Kieta sauga: raktai nėra grindžiami vartotojo įvesta pass-phrase, EFS generuoja raktus, kurie atsparūs žodyno atakoms.
- Visi šifravimai vykdomi branduolio būvyje.
- EFS teikia gerą duomenų atkūrimo mechanizmą – leidžia atstatyti duomenis net jei darbuotojas, kuris paliko organizaciją yra juos užšifravęs..

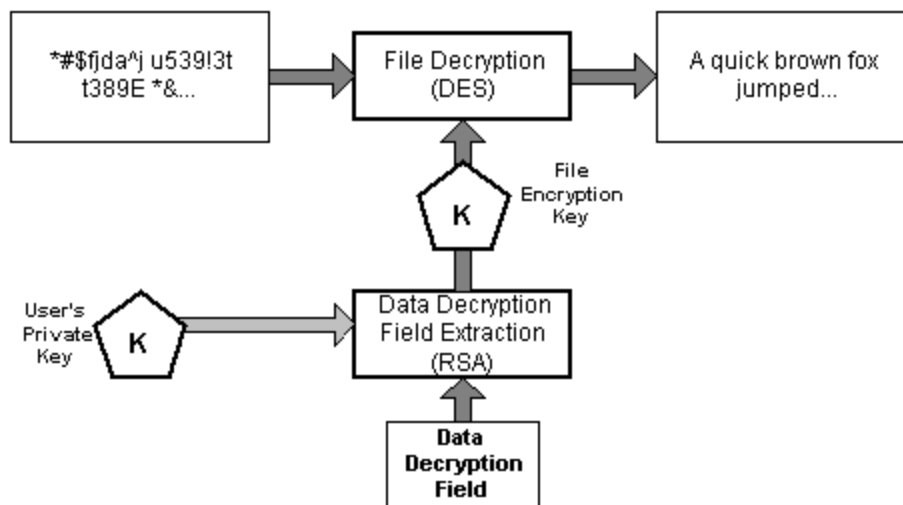
Kodavimas-šifravimas

- EFS naudoja simetrinio rakto šifravimą kartu su viešo rakto technologija failų apsaugai. Failų duomenys šifruojami naudojant DESX algoritmą.
- Šifravimui naudojamas raktas yra šifruojamas naudojant viešą/slaptą raktą ir yra saugomas kartu su failu. Dviejų algoritmų naudojimą skatina šifravimo greitis.
- Užšifravus failą yra kuriamas specialus įrašas vartotojui “Data Decryption Field (DDF)”, kuriame šifravimo raktas išsaugomas užšifruotas vartotojo viešu raktu.
- Taip pat yra sukuriamas duomenų atstatymo įrašas “Data Recovery Field (DRF)”, kuriame raktas patalpinamas jį užšifravus atstatymo agento viešu raktu.
- Baigus šifravimą duomenys yra pakeičiami šifruotais..
- DESX naudoja 128-bitų raktą, gali naudoti 3DES su 168-bitų raktu.



Dešifravimas

- Po to, kai failas yra užšifruotas, tik tas vartotojas, kuris turi laukus DDF arba DRF gali turėti prieigą prie failo.
- Gali būti organizuojama ir taip, kad keli vartotojai turėtų prieigą prie failo. Tai daroma sujungiant kelių vartotojų raktus į raktų grandinę ir užšifruojant raktą šių vartotojų viešais raktais.
- Dešifravimas yra atliekamas kaip atvirkščias veiksmas šifravimui.



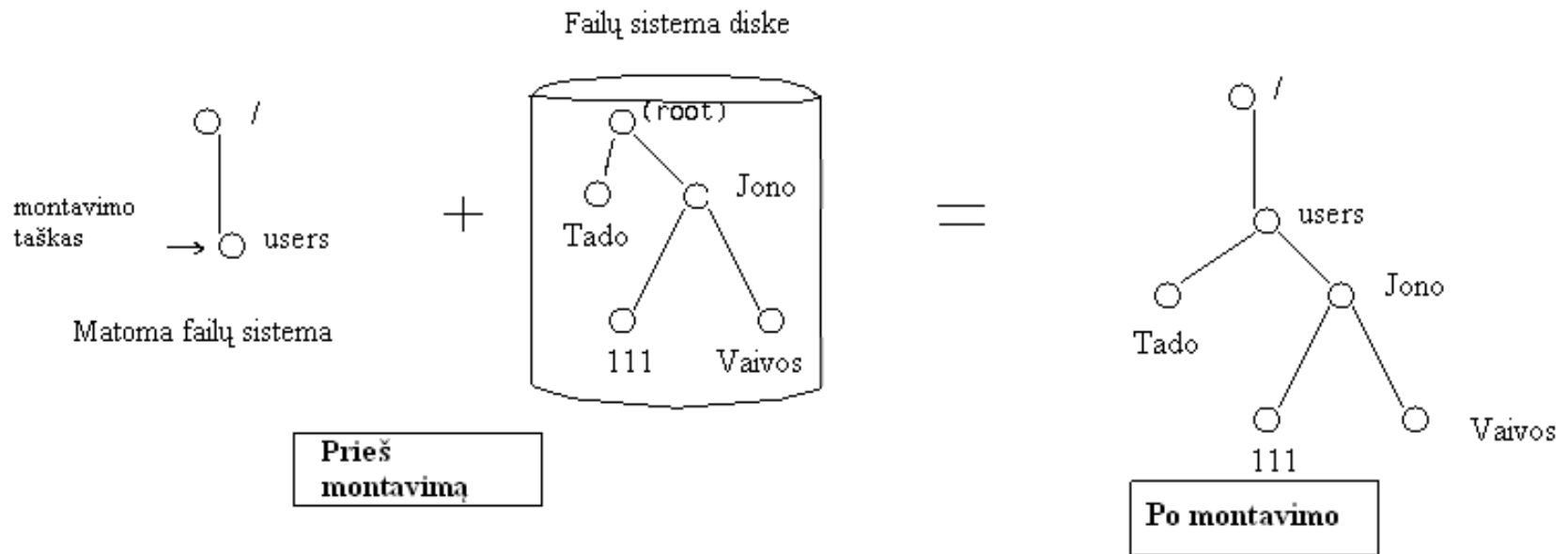
Failų sistemos montavimas UNIX

- Kiekviena diske esanti failų sistema turi savo šakninį katalogą
- Galima sukurti bendrą failų sistemą ją sumontuojant iš atskirų failų sistemų
 - Tai yra sukurti ryšį tarp katalogo vienoje failų sistemoje ir šakninio katalogo kitoje failų sistemoje.
- Tai leidžia sukurti bendrą medį iš kelių failų sistemų.
- Tai gali būti išplečiama ir per tinklą, sumontuojant kelių kompiuterių failų sistemas.

Failų sistemos montavimas

- Atskirame diske esančios failų sistemos įjungimas į egzistuojančią katalogų struktūrą yra žinomas kaip failų sistemos montavimas
- Primontuojama failų sistema gali rasti viename iš tiesiogiai prijungtų diskų (lokalioji failų sistema) arba būti nutolusios tinklinės failų sistemos dalimi.
- Montavimas susieja primontuojamą failų sistemą su tam tikru egzistuojančios failų sistemos katalogu.
- Iki montavimo failai, esantys primontuojamame diske nėra pasiekiami vartotojams.
- Katalogas, kuriame vykdomas montavimas yra vadinamas montavimo tašku.
- Montavimo tašku turi būti tuščias egzistuojančios sistemos katalogas.

Failų sistemos montavimas



Failų sistemos montavimas Linux

- Linux sistemos atveju yra galimybė sujungti skirtingo tipo failų sistemas į vieną bendrą failų sistemą.
 - Tai gali būti tokios failų sistemos: FAT, FAT32, NTFS, ir HPFS
- Jei šakninis katalogas viename iš diskų gali būti c:\, tai tas pats diskas Linux sistemoje gali būti pasiekiamas keliu /mnt/xdir; čia xdir tai montavimo taškas.
- Įrenginį sumontavus, jis tampa pasiekiamas vartotojams, turintiems leistiną prieigą prie jo.
 - Pvz diskelio primontavimui taške /mnt/floppy, reikės komandos
mount /dev/fd0 /mnt/floppy
- Linux instaliavimo metu, informacija apie montavimo taškus yra sukuriamą ir patalpinama faile /etc/fstab

Failų sistemos išmontavimas

- Išmontavus failų sistemą ji tampa laikinai neprieinama vartotojams.
- Išmontavimas neišmeta failų sistemos iš disko
- Sumontuota failų sistema yra automatiškai išmontuojama po kompiuterio stabdymo komandos “shutdown”.



Failų sistema ir įvykiai, dėl kurių prarandami duomenys

- Techninės įrangos sukeltos problemos
- Programinės įrangos klaidos
- Atsitiktiniai praradimai
- Priešiškai nusiteikę vartotojai ar įsilaužėliai
- Natūralios ar žmogaus sukeltos nelaimės

Atstatymas

- Atstatyti prarastus failus atstatant duomenis iš atsarginių kopijų.
 - Leidžia sumažinti duomenų praradimo riziką.
 - Yra vertinga, jei iš jų galima atstatyti duomenis
- Naudojamos sisteminės programos disko atsarginių kopijų darymui (*back up*) į kitus įrenginius.

Duomenų saugojimas atstatymo tikslas – vartotojų politika

- Vartotojai (duomenų savininkai) turi :
 - Suprasti kurie jo duomenys reikalauja apsaugos;
 - Rūpintis tuo, kaip reikės atstatyti duomenis, jei to prireiks;
 - Koordinuoti backup planus su sistemos administratoriumi;
 - Ar daryti savus backups;
 - Jei daromi backup tai būtų tikslinga retkarčiais patikrinti, kad iš jų tikrai galima atkurti duomenis

Saugojimas -backup

- Kopijų darymas:

- Ką saugoti?

- Operacinę sistemą – gal saugoti jos įvaizdį (image)
 - Duomenis

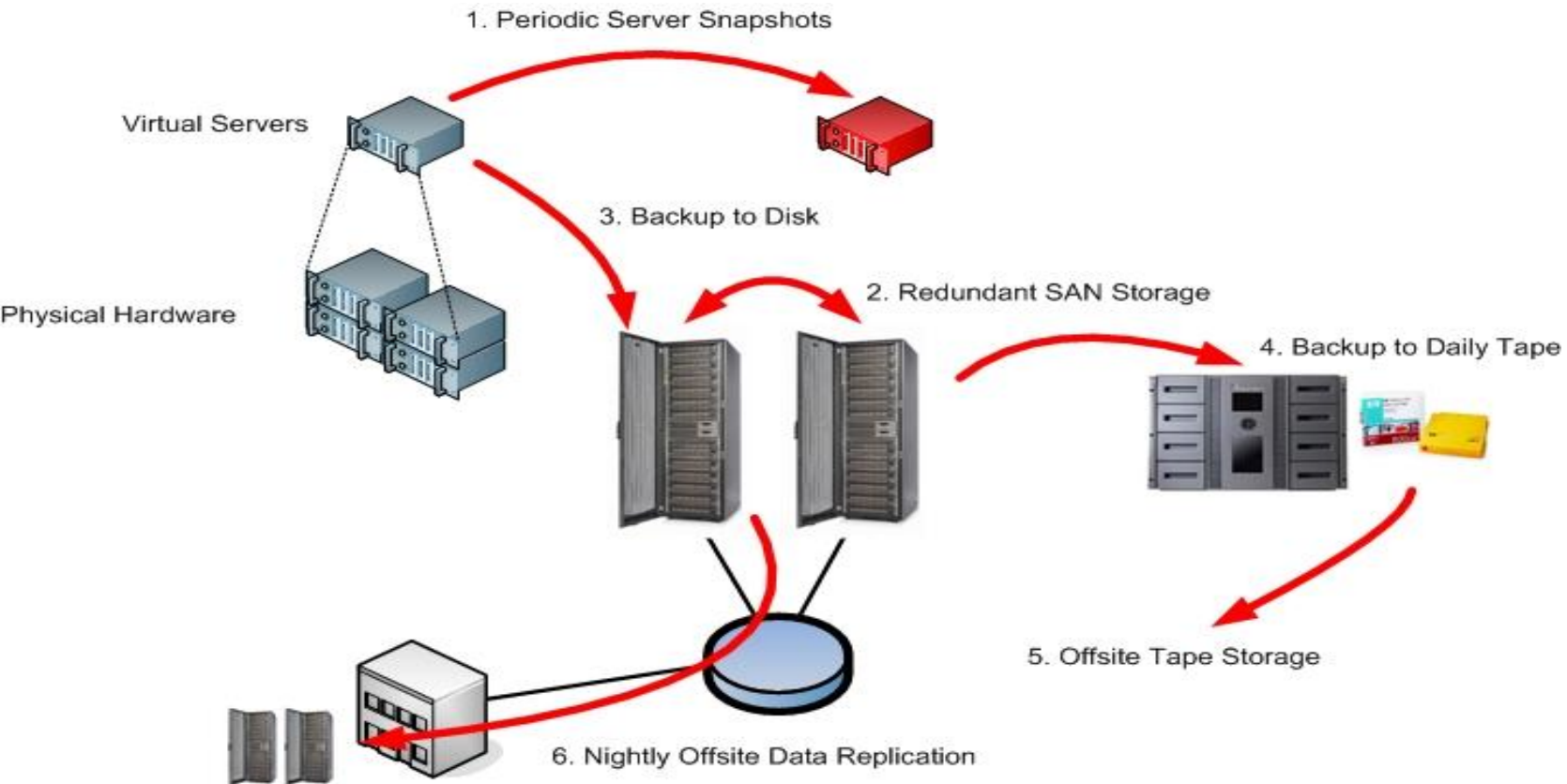
- Kur saugoti?

- Atskiri diskai
 - USB flash tipo įrenginiai
 - USB išoriniai diskai

- Įvesti tam tikrą tvarką saugant duomenis, kuriuos norima atstatyti

- /var ; c:\data
 - UNIX – vartotojų failai, /etc /usr/local /var

Kopijos



Ka saugoti? (Windows 7)

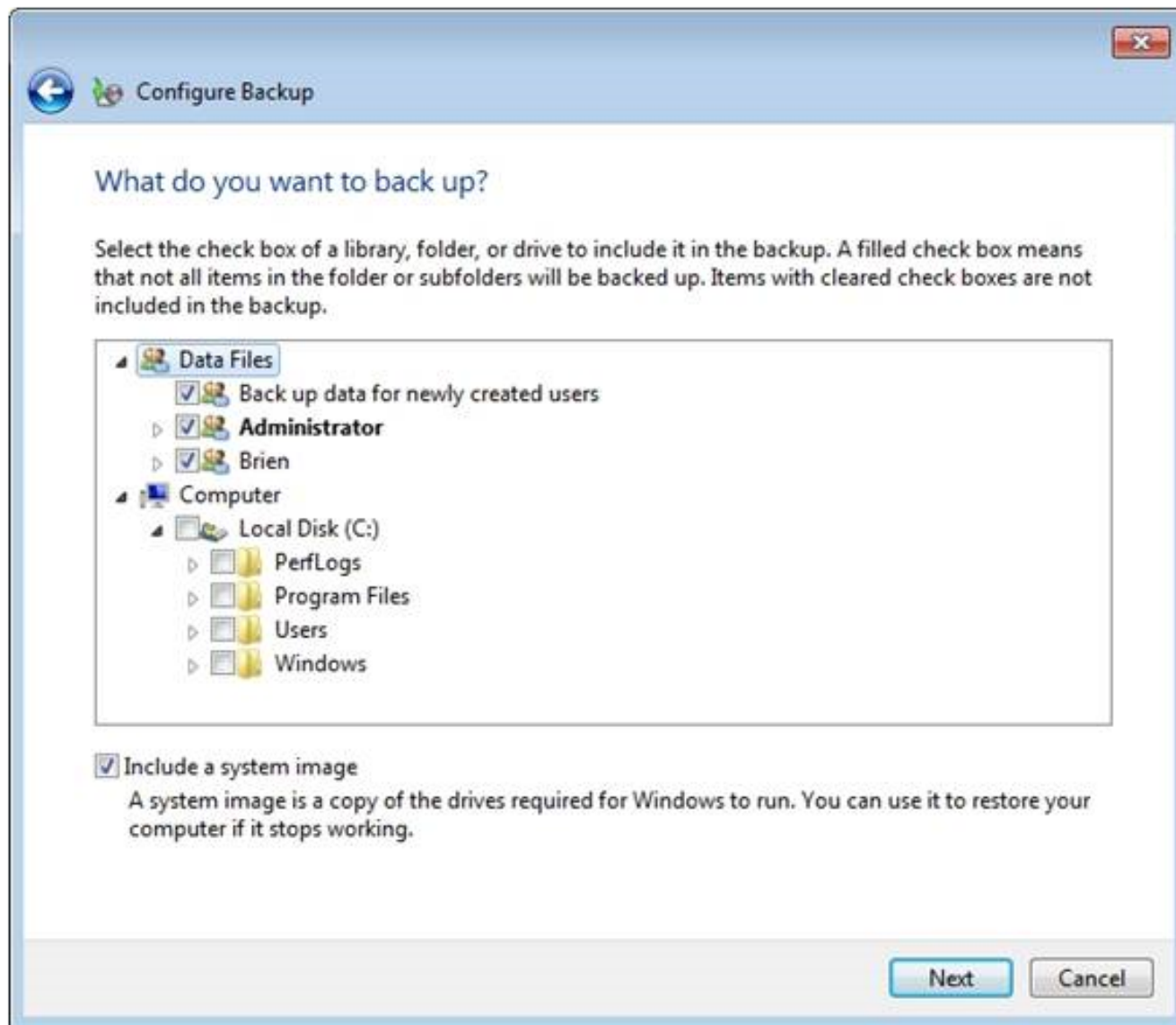


Figure B: Windows 7 allows you to choose what you want to back up