

LUENTO 16

# TIEDOSTOJÄRJESTELMÄ

Stallings, Luku 12.1-12.7

1

## Sisältöä

- Peruskäsitteistöä, tehtäviä ja tavoitteita
- Järjestelmän rakenne
- Tiedostojen organisointi
  - saantimenetelmät
- Hakemistot
- Yhteiskäyttö
- Levykirjanpito
- UNIX-esimerkki

2

## Peruskäsitteistöä

3

## Perustermit

- Kenttä (field)
  - bittien / tavujen muodostama looginen kokonaisuus
  - yksi arvo, jolla tyyppi ja pituus
- Tietue (record)
  - joukko yhteenkuuluvia ja yhdessä käsiteltäviä kenttiä
    - esim. työntekijä-tietue: nimi, osoite, tehtävä ...
- Tiedosto (file)
  - tietueiden muodostama hallinnollinen kokonaisuus
    - levykirjanpito ja pääsyoikeudet tiedostoittain
- Tietokanta (database)
  - joukko yhteenkuuluvia tiedostoja
  - sujuvat elementtien väliset viittaukset (hakuavaimin)

4

## Perustermit

- Tietokantojen hallintaa varten yleensä erillinen tiedonhallintajärjestelmä joka käyttää KJ:n tdstojärjestelmää
- Perusoperaatiot tietokannoille:
  - Hae yksi / kaikki tietueet
  - Hae tietyt ehdot täyttävät tietueet
  - Hae seuraava / edeltävä tietue
  - Lisää / muuta / poista tietue
- Ei yleensä lasketa KJ:n perusosaksi
  - paremminkin palveluohjelmistoa (utility programs)

5

## KJ:n tiedostojärjestelmä

- Tdstojen käsittely suojaussyistä aina KJ:n tdstojärjestelmän kautta
  - käyttöoikeudet osa tdstomääreitä (attribuutteja)
- Huolehtii tallennuksen ja käytön yks.kohdat
  - kirjanpito vapaista lohkoista (taltiolla!)
  - hierarkkinen hakemistorakenne: kirjanpito tdstoista
    - nimeäminen ja attribuutit, tdstoon kuuluvat lohkot
  - tdstojen käyttö
    - avoimet tdstot, luku / kirjoituspositio
    - yhteiskäyttö
- Ei ota kantaa tdston loogiseen rakenteeseen
  - tdsto KJ:lle vain potkko tavuja / lohkoja
  - sovellus 'tietää' tulkinnan

6

## Tehtäviä ja tavoitteita

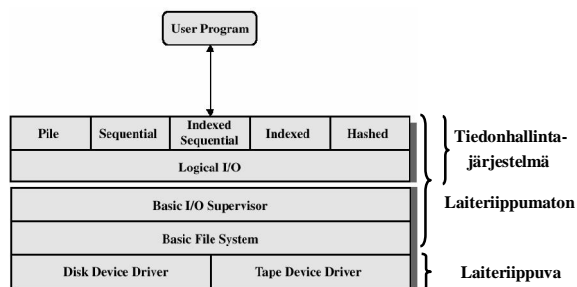
- Tiedon pysyvä tallennus
  - luonti, poisto, muuttaminen
  - varmistuskopiot
- Tiedon oikeellisuus
- Tehokkuus
- Yleisyys: sama idea OK erilaisille laitteille
- Standardoidut I/O-operaatiot
- Monenkäyttäjän moniajojärjestelmän huomiointi
  - eri tdstojen yhtäaikainen käyttö
  - samojen tdstojen yhteiskäyttö, lukitukset
  - käyttöoikeudet

7

## Järjestelmän rakenne

8

## Järjestelmän rakenne Kuva 12.1



## KJ:n laiteriippuva taso

### Laiteajuri (device driver)

- Alin ohjelmallinen taso
  - tuntee laitteen yksityiskohdat ja käyttötavat
- Ei hajulla esim. miten lohkot liittyvät tdstoihin
  - saa ylempää: lohkonumeron, lohkopuskurin muistiosoitteen, operaatiokoodin (R/W)
- Loogiset osoitteet fyysisiksi osoitteiksi
  - esim. lohkonumero  $\rightarrow$  levynta, ura, sektori
- Pyyntöjen uudelleenjärjestely
  - vähentää ylimääräisiä hakuvarren siirtoja
  - missä hakuvarsi nyt, minne seuraavaksi

10

## KJ:n laiteriippuva taso

### Laiteajuri

- Fyysisen siirränän aloitustoimet
  - käyttää laiteohjaimen rekistereitä
  - operaatio, mistä, minne, paljonko
  - ohjain voi toimia sen jälkeen itsenäisesti
- 'Siirto valmis' -keskeytys
  - laiteno  $\rightarrow$  etsi laitekuvaaja  $\rightarrow$  ajuri
- Fyysisen siirränän lopetustoimet
  - tarkistaa onnistumisen ohjaimen rekistereistä
  - 'herättää' KJ:n ylempät osat toimimaan

11

## KJ:n laiteriippumaton taso

- Ylläpitää tdston käytössä tarvittavia ajonaikaisia tietorakenteita
  - PCB:ssä oleva tdstokuvaajataulu
    - mitä tdstoja avannut
    - luku / kirjoituspositio
    - avaamis / käyttötapa (R / W / RW)
  - globaali avoimet tdstot taulu / lista
    - Tdstomääreet, yhteiskäyttö
- Vastaa taltion kirjanpidosta
  - kirjanpitorakenteet taltiolla, käsittelyä varten muistiin
  - missä vapaat / varatut alueet, mitkä lohkot kuuluvat tdstoon
  - tdstohakemistot

12

## KJ:n laiteriippumaton taso

- Huolehtii esim. lohkopuskuroinnista
  - puskuroi muistiin luetuja lohkoja
  - välittäjänä sovelluksen tavuittain / tietueittain käsittelyn ja levyn lohkoittain käsittelyn välillä
  - kopioi tavuja puskurista sovelluksen muuttuihin tai toisinpäin
- Huolehtii siirränän alustus & lopetustoimista
  - jos kaivatut tavut ei puskureissa, laita ajuri töihin
- Jokainen pyyntö ei aiheuta siirrantää
- Yhdistää loogiseen siirrantään oikean laitteen ja oikean ajurin
  - laitenumero Ø etsi laitekuvaaja Ø ajuri

13

## Looginen I/O

- Huolehtii mahd. tietueason käsittelystä
  - lue seuraava tietue / lue seuraavat tavut
  - hae kaikki ehdon täyttävät tdstot
- KJ:n laiteriippumaton osa huolehtii tdston tallentamisen ja käytön yksityiskohdista lohko- ja tavutasolla
- Looginen taso vastuussa tdston sisäiseen rakenteeseen liittyvistä yksityiskohdista
  - miten tietueet sijoittuvat lohkoon / lohkoihin
  - hakua helpottavat rakenteet, saantimenetelmät
  - ei lasketa aina kuuluvaksi peruskJ:hin

14

## Tiedostonhallinnan elementit

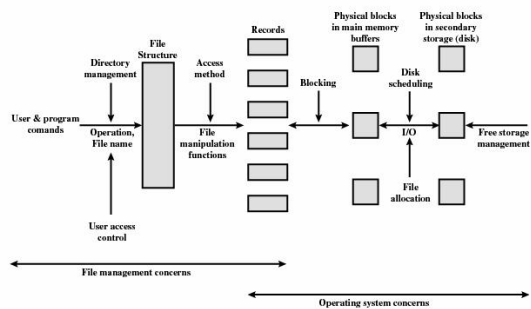


Figure 12.2 Elements of File Management

## Tiedostojen organisointi, saantimenetelmät

ks. Tietokannan hallinta -kurssi

16

## Tiedostojen organisointi

- Nopea saanti
  - tärkeä yksittäistä tietuetta käsiteltäessä
  - ei tärkeä koko tdstoa erätynä päivitettäessä
- Päivittämisen helppous
  - esim. CD-ROMilla oleva tdsto ei muutettavissa
- Talletustilan tehokas käyttö
  - ei moninkertaista / ylimääräistä talletusta
  - mutta ylimääräiset indeksit nopeuttavat käyttöä
- Ylläpidon yksinkertaisuus
- Luotettavuus

17

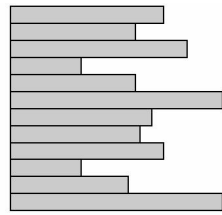
## Saantimenetelmät

- Sovelluksella standardoidut operaatiot tdston käsittelemiseksi
  - open(), close(), read(), write(), seek() ...
- Tdstojen erilaista käsittelyä varten voi olla niitä tukevia lisätietorakenteita
  - kasarakenne
  - peräkkäisrakenne
  - indeksoitu peräkkäisrakenne
  - indeksoitu rakenne
  - hajautettu rakenne
- Vaikuttavat tdstojen talletustapaan, käyttötapaan ja käytön tehokkuuteen

18

## Kasarakenne (pile file) Kuva 12.3

- Tallennus siinä järjestyksessä kuin tietueet saadaan
- Tarkoituksena on tallentaa data, käsittely myöhemmin
- Tietueiden välissä erottimek, tai tietueilla vakioipituus



Variable-length records  
Variable set of fields  
Chronological order

(a) Pile File

19

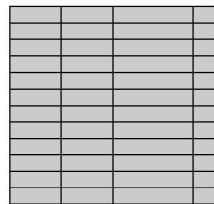
## Kasarakenne

- Kenttien lkm voi vaihdella ja / tai kentät voivat olla eri järjestyksessä eri tietueissa
  - talletettava kentän nimi ja arvo
  - kenttien välissä erottimek tai niillä vakioipituus
- Tietueen haku peräkkäishakuna
  - ei hakuä helpottavaa rakennetta / järjestystä
- Säästää tilaa, jos
  - paljon puuttuvia kenttiä
  - kenttien pituudet vaihtelevat

20

## Peräkkäisrakenne Kuva 12.3

- Tietueilla sama rakenne ja pituus
  - sama kenttien järjestys ja pituus kaikissa tietueissa
    - maxpituuden mukaan
  - kenttien nimii ja pituuksia ei tarvitse tallettaa
- Kenttien nimet ja pituudet tdston attribuutteja
  - talletettu tai muuten vaan käsittelijän tiedossa



Fixed-length records  
Fixed set of fields in fixed order  
Sequential order based on key field

(b) Sequential File

21

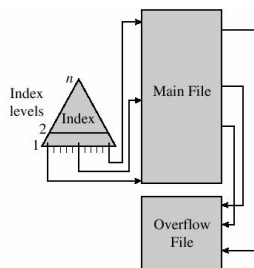
## Peräkkäisrakenne

- Yksi kentistä avain
  - yksikäsitteinen
  - tietueet avaimen mukaan järjestyksessä
- Sopii sekä levyille että nauhoille
- Käytetään lähinnä vain eräkäsittelyssä
  - päivitystapahtumat erillisessä tdstossa (avainjärjestys)
- Päivitysprosessi tuottaa kokonaan uuden tdston
  - kantatdsto ja päivitystapahtumat järjestetty avaimen mukaan

22

## Indeksoitu peräkkäisrakenne

- Järj. tietueiden lisäksi hakuä helpottavaa tietoa, hakemisto (indeksi)
- Hstosta osoittimia tdston keskelle
  - avain, missä tietue
- Etsi hstosta avaimen perusteella suunnilleen oikea kohta, jatka peräkkäishakuna vars. talletusalueelta



(c) Indexed Sequential File

23

## Indeksoitu peräkkäisrakenne

- Lisäyksiä varten erillinen lisäysalue
  - jokaisessa tietueessa ylimääräinen linkkikenttä
  - uusi tietue lisäysalueelle, ja 'kotilohkon' viimeiseen tietueeseen viite
- Aika-ajoin koko tdston uudelleenjärjestely
  - lisäysalueen lomitus

24

## Haun tehokkuudesta

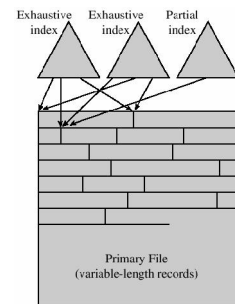
Kun tdstossa miljoona tietuetta

- Peräkkäishaku
  - käytävä läpi km. 500000 tietuetta
- Käytössä 1000 alkion hakemisto
  - hstossa joka tuhannen alkion avain + viite tietueeseen
  - käytävä läpi km. 500 alkioita hakemistosta ja löytyneestä lohkoista km. 500 tietuetta
- Voi tehostaa monitasoisen hakemiston avulla
  - 2-tason indeksi 10000 alkioita (joka 100:s avain+viite)
  - 1-tason indeksi 100 alkioita (joka 100:s 2-tason avain+viite)
  - haku: 50 + 50 + 50 tietuetta

25

## Täysin indeksoitu rakenne

- Useita hstoja, useita hakuavaimia
- Käsittely vain hstojen (indeksien) kautta
- Koko tdston läpikäyntiä varten pääavaimen mukainen täyd. hsto



(d) Indexed File

26

## Täysin indeksoitu rakenne

- Tietueiden ei tarvitse olla fyysisesti avaimen mukaisessa järjestyksessä
  - läpikäynti vain hstojen kautta
- Lisäykset / poistot helppoja
- Käytetään esim. jos haun nopeus tärkeää, ja vain harvoin peräkkäiskäsittelyä

27

## Suorasaantirakenne

- Kukin tietue / lohko saatavissa missä tahansa järjestyksessä
  - sopiva laite levy
- Sijainti selvitetävissä avaimen perusteella
  - voi perustua myös hajautuksen käyttöön
- Tdston tietueet / lohkot sijoitellaan taltiolle vapaaseen järjestykseen
- Sopii kiinteänmittaisille tietueille / lohkoille
- Sopii hyvin interaktiiviseen reaaliaikakäyttöön
  - tietueen päivitys välittömästi omalla paikallaan
    - lue muistiin, päivitä, kirjoita taltiolle

28

Table 12.1 Grades of Performance for Five Basic File Organizations [WIED87]

File Method	Space		Update		Retrieval		
	Attributes		Record Size		Single record	Subset	Exhaustive
	Variable	Fixed	Equal	Greater			
File	A	B	A	E	E	D	B
Sequential	F	A	D	F	F	D	A
Indexed sequential	F	B	B	D	B	D	B
Indexed	B	C	C	C	A	B	D
Hashed	F	B	B	F	B	F	E

A = Excellent, well suited to this purpose  
 B = Good  
 C = Adequate  
 D = Requires some extra effort  
 E = Possible with extreme effort  
 F = Not reasonable for this purpose

where  
 $r$  = size of the result  
 $o$  = number of records that overflow  
 $n$  = number of records in file

## Hakemistot

30

## Hakemisto

= Tiedosto, jossa tdstoihin liittyy tietoa

- Vain KJ:lla hakemistotdston suora käyttöoikeus
  - hakemiston muutokset aina KJ:n palvelupyyntöjä
    - listaa hakemiston tiedot, etsi tdstonimi, lisää / poista tdstonimi, ...
- Hakemistoalkiossa mm. (ks. taulu 12.2)
  - tiedostonimi
  - attribuutit: omistaja (uid), ryhmä (gid), tyyppi, käyttöoikeudet, aikaleimat
  - mitkä lohkot kuuluvat tdstoon, koko tavuina
  - laite, jolla tdsto majoituu

31

Table 12.2 Information Elements of a File Directory

Basic Information	
File Name	Name as chosen by creator (user or program). Must be unique within a specific directory.
File Type	For example: text, binary, load module, etc.
File Organization	For systems that support different organizations
Address Information	
Volume	Indicates device on which file is stored
Starting Address	Starting physical address on secondary storage (e.g., cylinder, track, and block number on disk)
Size Used	Current size of the file in bytes, words, or blocks
Size Allocated	The maximum size of the file
Access Control Information	
Owner	User who is assigned control of this file. The owner may be able to grant/deny access to other users and to change these privileges
Access Information	A simple version of this element would include the user's name and password for each authorized user.
Permitted Actions	Controls reading, writing, executing, transmitting over a network

## Taulu 12.2. jatkuu

Usage Information	
Date Created	When file was first placed in directory
Identity of Creator	Usually but not necessarily the current owner
Date Last Read Access	Date of the last time a record was read
Identity of Last Reader	User who did the reading
Date Last Modified	Date of the last update, insertion, or deletion
Identity of Last Modifier	User who did the modifying
Date of Last Backup	Date of the last time the file was backed up on another storage medium
Current Usage	Information about current activity on the file, such as process or processes that have the file open, whether it is locked by a process, and whether the file has been updated in main memory but not yet on disk

## Yksitasoinen hakemisto

- Vain lista tdstonimistä ja attributeista
- Yksinkertainen peräkkäisrakente
  - avaimena tdstonimi
- Ei anna apua tdstojen organisointiin
  - ei hierarkiaa
  - kaikkien käyttäjien tdstot siinä ainoassa hstossa
- Tdstonimien oltava yksikäsitteisiä koko järjestelmän sisällä
  - eri käyttäjillä ei voi olla samannimisiä tdstoja
- ks. kadonnut kansanperinne; ei käytössä

34

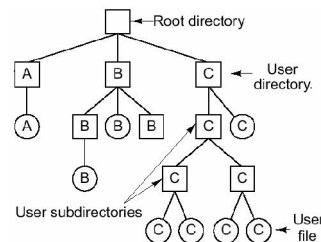
## 2-tasoinen hakemisto

- Jokaiselle käyttäjälle oma käyttäjähakemisto
  - eri käyttäjillä voi olla samannimisiä tdstoja
- Päähakemisto, jossa tiedot käyttäjähstoista
- Hakemisto edelleen vain lista tdstonimiä, ei käyttäjäkohtaista hierarkiaa
  - käyttäjällä ei voi olla kahta samannimistä tdstoa
  - Käyttäjä ei voi luoda hakemistoja
- ks. kadonnut kansanperinne; ei käytössä

35

## Hierarkinen puurakenne

- Juurihsto, kotihakemistot, alihakemistot
- Myös prosessit voivat luoda alihakemistoja
- Juurihstolla kiinteä paikka levyllä



36

## Hierarkinen puurakenne

- Prosessiin liittyy työhakemisto
  - kun prosessi käynnistyy, työhakemistona oma kotihakemisto (kirjattu PCB:hen)
  - työhakemistoa voi vaihtaa
- Tdstonimi muodostuu polkunimestä, jonka osina hstonimet ja tdstonimi
  - absoluuttinen polkunimi: /hsto/hsto/tdstonimi
  - suhteellinen polkunimi: hsto/hsto/tdstonimi, tdstonimi
  - hakemistot . ja ..
- Kaikki nykyiset järjestelmät käyttävät hierarkista hakemistorakennetta

37

## Tiedostojen yhteiskäyttö

38

## Tiedostojen yhteiskäyttö

- Käyttöoikeudet merkitty attributteihin
- Käyttäjien jaottelu esim. (UNIX)
  - u omistaja
  - g samaan ryhmään kuuluvat
  - o muut käyttäjät
- Oikeuksien jaottelu ryhmittäin esim. (UNIX)
  - - ei mitään
  - r lukuoikeus
  - w kirjoitusoikeus (oikeus muuttaa)
  - x suoritusoikeus
- Omistaja voi muuttaa käyttöoikeuksia

39

## Tiedostojen yhteiskäyttö

- Hakemiston oikeudet esim. (UNIX)
  - r oikeus listata hston sisältö
  - w oikeus poistaa tdsto hakemistosta
  - x oikeus käyttää hstonimeä polkunimessä
- Muita mahdollisia oikeuksia
  - lisääminen vain tdston loppuun
  - käyttöoikeuksien muuttaminen
  - oikeus listata tdstonimi ja omistaja
  - ...
- Eräissä järjestelmissä mahdollista määritellä myös käyttäjäkohtaisia oikeuksia
  - Pääsyylistat ja valtakirjalistat (access control/capability lists)

40

## Tiedostojen yhteiskäyttö

- KJ tarkistaa käyttöoikeudet yleensä vain tdstoa avattaessa
  - PCB:ssä omistaja (uid) ja ryhmä (gid)
  - hstoalkiossa omistaja (uid) ja ryhmä (gid)
- Oikeudet oltava kaikkiin polkunimen osiin
- Käyttäjän uid ja gid käyttäjätietokannasta
  - UNIX: passwd-tdsto: uid ja ensisijainen gid
  - group-tdsto: käyttäjän muut ryhmänumerot
- uid ja gid periytyvät lapsiprosesseille ja edelleen luoduille tdstoille

41

## Tiedostolukitus

- Vain yksi saa muuttaa kerrallaan tdstoa, siitä voi lukea usea yhtäaikaan
- Kullakin kjällä oma luku/kirjoituspositio
  - Tdstokuvaajataulu PCB:ssä
  - muut KJ:n rakenteet yhteiskäytössä
    - KJ:n avoimet tdstot taulu, tdstolukot
    - KJ:n lohkopuskurit, ...
- KJ voi tarjota lukituksen tai lukitus jää sovelluksen tehtäväksi
  - koko tiedoston lukitus
  - vain osan lukitus (esim. 80 tavua tai levylohko)
- KJ tarkistaa lukituksen tdstoa avattaessa tai tdstoon viitattaessa

42

## Tilanvaraus

43

## Tilanvaraus

- Ennaltavaraus
  - maksimikoko kerrottava etukäteen
    - onnistuu, jos esim. kopioidaan tdsto
    - luojilla taipumus arvioida yläkantiin
    - hukkaa tilaa
  - sijoitettavissa käytön kannalta tehokkaasti
    - minimoi hakuajan
- Varaus tarvittaessa
  - järkevämpää ja yleisempää
  - ei hukkaa tilaa
  - hakuajan minimoimiseksi joskus uudelleenjärjestelyä

44

## Tilanvaraus

- Pidettävä kirjaa taltion vapaista alueista
  - kirjanpito taltiolla
  - lue muistiin muuttamista varten
- Pidettävä kirjaa tdstolle varatuista alueista
  - kirjanpito tdstokohtaisesti
  - löytyy hstoalkiosta / tdston attributeista
- Varaus yhtenä tai useampana erillisenä alueena
- Kerralla varattavan alueen koko vaikuttaa
  - kirjanpitoalueiden kokoon
  - hukkatilan määrään

45

## Varausyksikön koko

- Jos tdston osat sopivasti peräkkäin, käsittely tehokasta
  - väh. hakuvarren siirtoja ja pyörähdysviipeitä
- Pientä varausyksikköä käytettäessä kirjanpitoalueet vievät enemmän tilaa
- Kiinteän varausyksikkökoon käyttö yksinkertaistaa varausta ja vapautusta
  - vrt. virtuaalimuisti ja sivutus
- Vaihtelevanmittaisen varausyksikön käyttö minimoi hukkatilaa
  - vrt. segmentointi
  - kirjanpito työläämpää

46

## Koko tdsto yhdelle alueelle

- Hstoalkiossa vain ens. lohkon numero sekä tdston koko (pituus)
- Koon muuttaminen vaikeaa
  - arvioitava varausta tehtäessä
  - saatetaan joutua kopioimaan uudelle alueelle

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

47

## Koko tdsto yhdelle alueelle

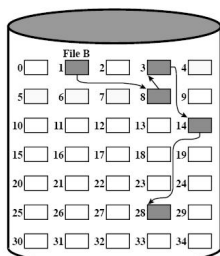
- Pirstoutuminen
  - tdstojen väliin jää tyhjää tilaa
- Vapaista alueista linkitetty lista
  - ens. vapaa lohkonumero, montako vapaata peräkkäin
- Vrt. yksinkertainen muistinhallinta
  - samat ongelmat ja heikkoudet

48



## Lohkojen ketjutus

- Varaus lohko kerrallaan vasta tarvittaessa
- Hstoalkiossa viite tdston ensimm. lohkonumeroon sekä tdston koko



File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...

Fig 12.9 [Stal05]

49

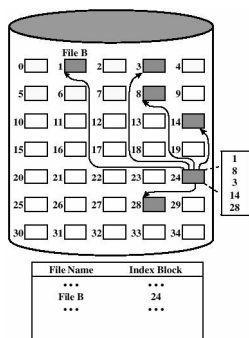
## Lohkojen ketjutus

- Ketjutukselle tav. erillinen varaustaulu FAT, file allocation table
  - ketjutus ei syö tilaa datalohkoista
  - tdston lohkot muodostavat hstoalkiosta alkavan ketjun
    - alkiossa sran lohkon numero tai loppumerkki
  - myös vapaat lohkot voidaan merkitä samaan tauluun
- Lohkojen varaus ja vapautus helppoa
  - mikä tahansa vapaa lohko kelpaa,
  - ei yhdistelemistä, ei järjestelemistä
- Saattaa aiheuttaa tdston pirstoutumista
  - peräkkäiset lohkot kaukana toisistaan
    - hidastaa hakua
  - lohkojen järjestäminen lähemmäs toisiaan (consolidation)
  - levyn tiivistäminen (defragmentation)

50

## Lohkohakemisto

- Erillinen hakemisto tdstolle varatuista lohkoista
- Usein erillään omassa lohkoissa
  - hstoalkiossa vain hstolohkon numero

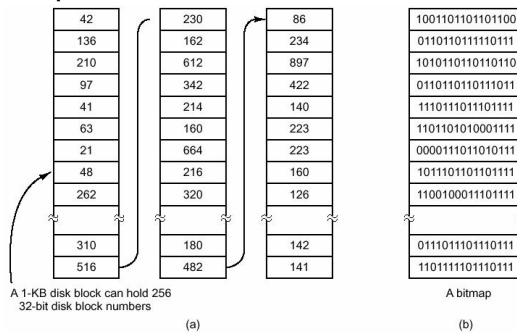


File Name	Index Block
...	...
File B	24
...	...

51

## Vapaan tilan hallinta

Tan01 6-21



A 1-KB disk block can hold 256 32-bit disk block numbers

(a)

A bitmap

(b)

Lohkolista

Bittikartta

52

## Vapaan tilan hallinta

- Bittikartta
  - 0=vapaa, 1=varattu
- Vapaiden lohkojen listat
  - lohkoja, joihin talletettu vapaiden lohkojen numeroita
  - lohkon viimeinen numero viite seuraavaan lohkoon, jossa lisää vapaiden numeroita
  - esim. vanha UNIX
- Lohkojen varaus ja vapautus helppoa
- Saattaa aiheuttaa tdston pirstoutumista

53

Esimerkki:  
Wanha perinteinen UNIX

54

## UNIX

- Hstoalkiossa vain tdstonimi ja i-solmun numero (inode, information node)
  - tdstolla voi olla useita nimiä!
- Tdston muut attribuutit i-solmussa (64 B)
  - omistaja (uid), ryhmä (gid)
  - linkkien lukumäärä
  - koko tavuina
  - tdstotyyppi
    - hsto, tavallinen, lohkolaite, merkkilaite, putki...
  - käyttöoikeusbitit
    - omistaja rwx, ryhmä rwx ja muut rwx
  - aikaleimoja: muutettu, käytetty, inoidea muutettu

55

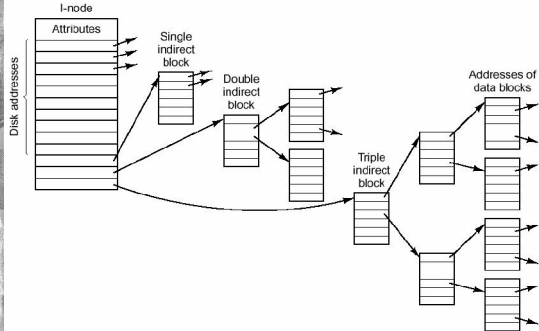
## UNIX

- i-solmussa myös 13 alkion lohkokohakemisto
  - 10 suoraa tdston lohkonroa
  - lohkonro, jossa 256 tdston lohkonroa
  - lohkonro, jossa 256 lohkonroa, joissa 256 tdston lohkonroa
  - lohkonro, jossa 256 lohkonroa, joissa 256 lohkonroa, joissa 256 tdston lohkonroa
- Pääosa UNIX-tdstoista kooltaan alle 10 kiloa
  - lohkonumerot nopeasti selvillä
- Suurin tiedostokoko > 16 GB
  - mutta i-nodessa vain 32-bittisiä koolle => max 4GB

56

## UNIX

Tan01 6-38



57

## UNIX

Tan01 6-39

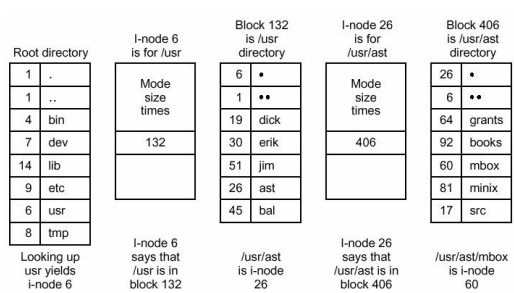


Fig. 6-39. The steps in looking up /usr/ast/mbox.

58

## UNIX

Tan01 10-31

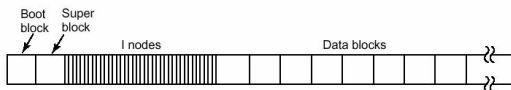
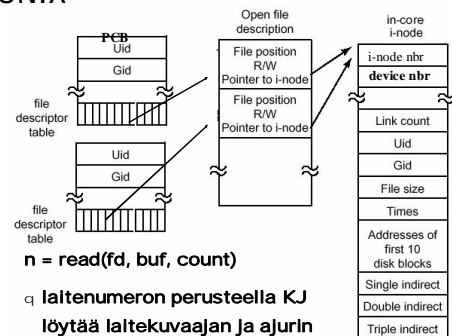


Fig. 10-31. Disk layout in classical UNIX systems.

- vapaat lohkot ketjutettu aiemmin kuvatulla tavalla talliota alustettaessa
- i-solmussa merkintä vapaa/varattu
- superlohkossa mm. laiteno, partition koko, vapaiden lohkojen listan alku, vapaiden i-solmujen numeroita

59

## UNIX



60

