

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

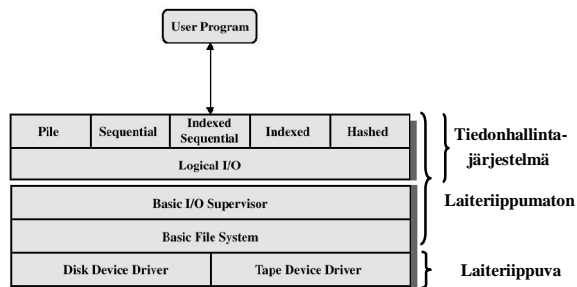
- Tiedon pysyvä tallennus
  - luonti, poisto, muuttaminen
  - varmistuskopiot
- Tiedon oikeellisuus
- Tehokkuus
- Yleisyys: sama idea OK erilaisille laitteille
- Standardoidut I/O-operaatiot
- Monenkäyttäjän moniajojärjestelmän huomiointi
  - eri tdstojen yhtäaikainen käyttö
  - samojen tdstojen yhteiskäyttö,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

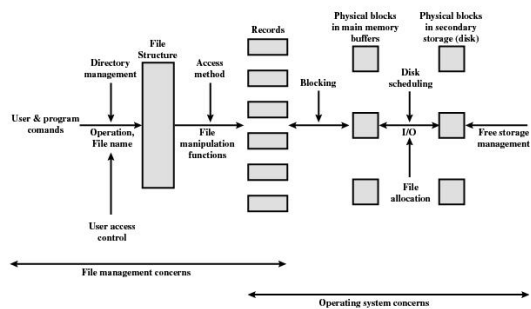
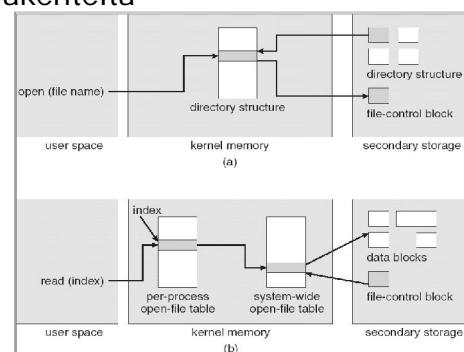


Figure 12.2 Elements of File Management

--

## Tiedostojen käyttöön liittyviä rakenteita

[SGG07] Fig 11.3



16

## Tiedostojen organisointi, saantimenetelmät

ks. Tietokannan hallinta -kurssi

17

## Tiedostojen organisointi

- Nopea saanti
  - tärkeä yksittäistä tietuetta käsiteltäessä
  - ei tärkeä koko tdstoa erityyönä 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

18

## Saantimenetelmät

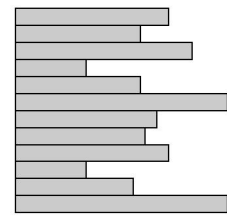
- Sovelluksella standardoidut operaatiot tdston käsittelemiseksi
  - open(), close(), read(), write(), seek() ...
- Tdstojen erilaista käsittelyä varten voi olla niitä tukevia lisätietorakenteita
  - kasarakenne
  - peräkkäisrakenne
  - indeksoitu peräkkäisrakenne
  - indeksoitu rakenne
  - hajautettu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

19

## Kasarakenne (pile file)

Kuva 12.3

- Tallennus siinä järjestyksessä kuin tiedueet saadaan
- Tarkoitus koota ja tallettaa data, käsitteily myöhemmin
- Kenttien lkm ja koko voi vaihdella ja / tai kentät voivat olla eri järjestyksessä eri tiedueissa
  - kentän nimi ja arvo
  - Erotin: merkki tai vakiopituus
- Tietueen haku peräkkäishakuna
- Säästää tilaa, jos
  - puuttuvia kenttiä tai vaihteleva pituus



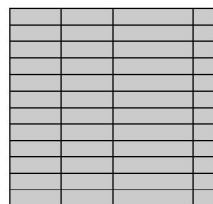
(a) Pile File

20

## Peräkkäisrakenne

Kuva 12.3

- Tietueilla sama rakenne ja pituus
  - sama kenttien järjestys ja pituus
  - nimia ja pituuksia ei tarvitse tallettaa
- Yksi kentistä yksik. avain
  - tiedueet avaimen mukaan järjestyksessä
- Sopii sekä levyille että nauhoille
- Käytetään lähinnä vain eräkäsitt.
- Päivitystapahtumat erillisessä tdstossa (avainjärjestys)
- Päivitysprosessi tuottaa uuden tdston
  - kantatdsto ja päivitystapahtumat järjestetty avaimen mukaan

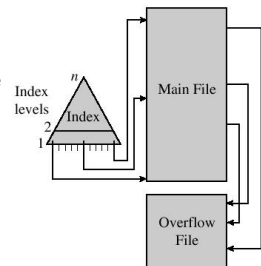


(b) Sequential File

21

## Indeksoitu peräkkäisrakenne

- Tietueiden lisäksi hakemisto (indeksi) haku helpottamaan
  - Hstosta osoittimia tdston keskelle
    - avain, missä tietue
- Etsi hstosta avaimen perusteella suunnilleen oikea kohta, jatka peräkkäishakuna vars. tdstosta
- Lisäyksiä varten erillinen alue
  - jokaisessa tiedueessa ylimääräinen linkkikenttä
  - uusi tietue lisäysalueelle, ja 'kotiloikon' viimeiseen tiedueeseen viite
- Aika-ajoin koko tdston uudelleenjärjestely - lomitus



(c) Indexed Sequential File

22

## Haun tehokkuudesta

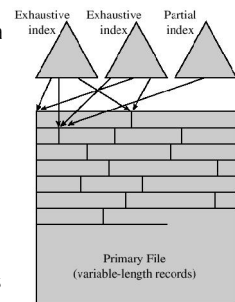
Kun tdstossa miljoona tietuetta

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

23

## Täysin indeksoitu rakenne

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



(d) Indexed File

24

## Suorasaantirakenne

- Kukin tietue / lohko saatavissa missä tahansa järjestyksessä
  - sopiva laite levy
- Sijainti selvitettävissä 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

25

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			
Pile	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  $\approx O(r)$   
 B = Good  $\approx O(r \times r)$   
 C = Adequate  $\approx O(r \log n)$   
 D = Requires some extra effort  $\approx O(n)$   
 E = Possible with extreme effort  $\approx O(r \times n)$   
 F = Not reasonable for this purpose  $\approx O(n^2)$

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

## Hakemistot

27

## Hakemisto

- = Tiedosto, jossa tdstoihin liittyvää 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 majoilee

28

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 attribuuteista
- Yksinkertainen peräkkäisrakenne
  - 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

31

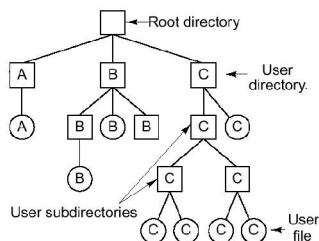
## 2-tasoinen hakemisto

- 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

32

## Hierarkinen puurakenne

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



33

## Hierarkinen puurakenne

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

34

## Tiedostojen yhteiskäyttö

35

## Tiedostojen yhteiskäyttö

- Käyttöoikeudet merkitty attribuutteihin
- 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

36

## Tiedostojen yhteiskäyttö

- Hakemiston oikeudet esim. (UNIX)
  - r oikeus listata hstn sisältö
  - w oikeus poistaa tdsto hakemistosta
  - x oikeus käyttää hstnimeä 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)

37

## 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

38

## Tiedostolukitus

- Vain yksi saa muuttaa kerrallaan tdstoa, siitä voi lukea usea yhtäaikaan
- Kullakin kJ:llä 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 viittaessa

39

## Tilanvaraus

40

## Tilanvaraus

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

41

## Tilanvaraus

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

42

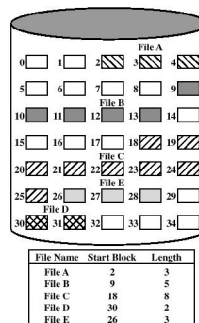
## Varausyksikön koko

- 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

43

## Koko tdsto yhdelle alueelle

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



44

## 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

45

## Lohkojen ketjutus

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

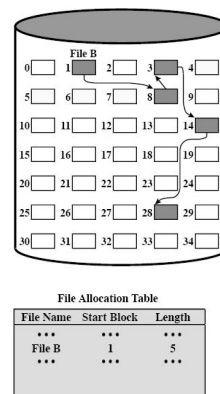


Fig 12.9 [Stal05]

46

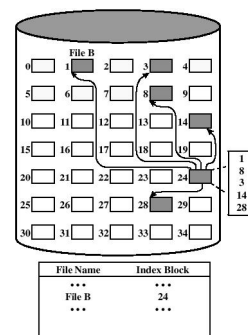
## Lohkojen ketjutus

- Ketjutukselle tav. erillinen varaustaulu FAT, file allocation table
  - ketjutus ei syö tilaa datalohkoista
  - tdston lohkot muodostavat hstoalkiosta alkavan ketjun
    - alkiossa srvan 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)

47

## Lohkohakemisto

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



48

### Vapaan tilan hallinta

Tan01 6-21

The diagram illustrates free space management. On the left, three columns of block numbers are shown, with arrows indicating their mapping to a bitmap on the right. The bitmap is a 16x16 grid of bits, where 0 represents free space and 1 represents allocated space.

**Lohkolista**

42	230	86
136	162	234
210	612	897
97	342	422
41	214	140
63	160	223
21	664	223
48	216	160
262	320	126
310	180	142
516	482	141

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

**Bittikartta**

1001101101101100
0110110111110111
1010110110110110
0110110110110111
1110110110110111
1101101010001111
0000110110101111
1011101101101111
1100100011101111
0111011011011011
1101111101101111

(a) (b)

### 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

49

### Esimerkki:

## Wanha perinteinen UNIX

51

## 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, lohkolaitte, merkkilaitte, putki...
  - käyttöoikeusbitit
    - omistaja rwx, ryhmä rwx ja muut rwx
  - aikaleimoja: muutettu, käytetty, inodea muutettu

52

## 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-bittiiä koolle => max 4GB

53

## UNIX

[SGG07] Fig 11.9

The diagram shows the structure of an inode. It includes fields for mode, owners (2), timestamps (3), and size block count. Below these are pointers to data blocks: direct blocks, single indirect, double indirect, and triple indirect. Each indirect pointer points to a block containing pointers to data blocks.

54

