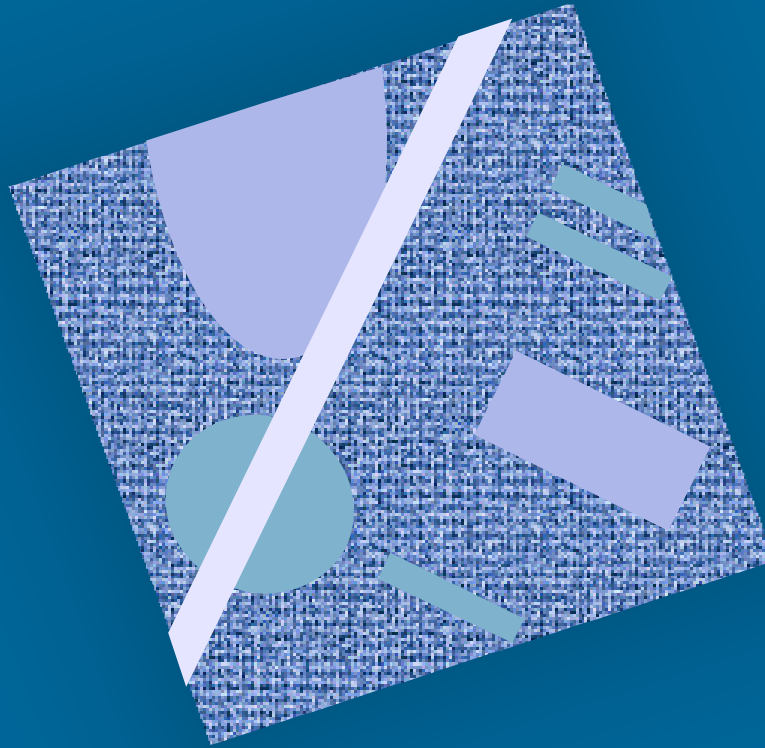


# Luento 9

## Järjestelmän ulkoinen muisti



Muistihierarkia  
Virtuaalimuisti  
Kiintolevyt  
Muut pyörivät levyt

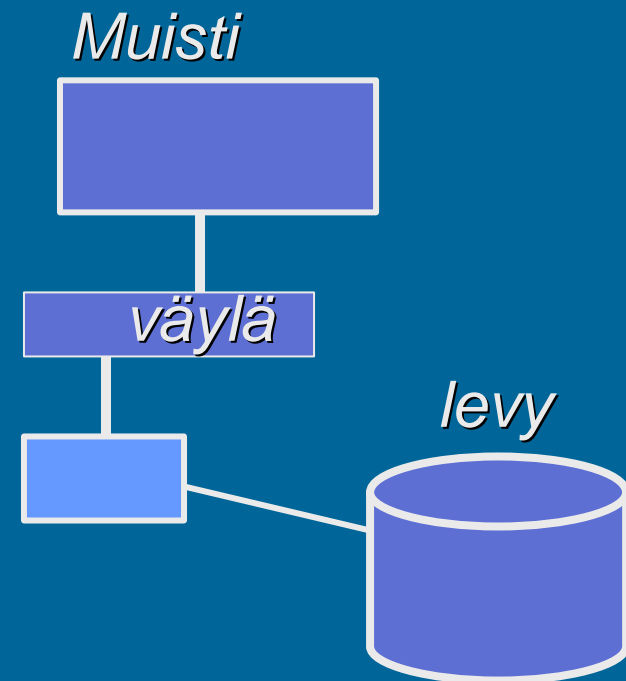
# Muistihierarkia <sup>(4)</sup>

ks. Fig. 2-18 [Tane99]

- Ulkoinen muisti on halvempaa toteuttaa per tavu
- Ulkoinen muisti on hyvin paljon hitaampaa kuin sisäinen muisti
- Aika/tila optimointi
  - suuret tietomäärät täytyy (kannattaa) kustannussyistä pitää ulkoisessa muistissa
  - pienet tietomäärät täytyy (kannattaa) tehokkuussyistä pitää sisäisessä muistissa
- Kaiken viitatus tiedot tulee suoritusaikana olla sisäisessä muistissa!

# Virtuaalimuisti <sup>(3)</sup>

- Osa muistihierarkiaa
- Vastaus ongelmaan: miten tehdä suoritusajaisesta muistista yhtä suuri kuin levymuisti ja yhtä nopea kuin keskusmuisti
- Kaksitasoinen:
  - keskusmuistissa kulloinkin käytössä alueet
  - levyllä kaikki tiedot
  - kopiointi tarvittaessa



# Ohjelman muistiosoitteet <sup>(3)</sup>

- Ohjelmassa loogiset nimet
  - muuttujat, oliot, aliohjelmat
- Ladatussa ohjelmassa loogiset osoitteet
  - staattinen osoitteiden sidonta
  - joskus myös suorat fyysiset osoitteet käytössä
    - KJ:n osat (esim. keskeytyskäsitteijät)
- Suoritusaikana tarvitaan fyysisiä osoitteita
  - dynaaminen osoitteiden sidonta
  - tehdään joka konekäskyllä jokaiselle tarvittavalle osoitteelle
    - käskyn osoite
    - operandien osoitteet

# Heittovaihto

(swapping)

- Pidä kaikki ohjelman muistialueet muistissa
  - Suorituksessa oleva ohjelmalle
  - Ready-to-Run jonossa oleville ohjelmille
- Uusi prosessi?
  - Etsi vapaa yhtenäinen muistialue ja ”swappaa” prosessi sinne
  - Ei tilaa? Siirrä joku (alemmman prioriteetin) työ levyille
- Joskus iso ohjelma pitää jakaa pienempiin ajokelpoisiin heittovaihdettaviin osiin
- Osoitteen muunnos: kanta- ja rajarekisterit

Miksi?

# Virtuaalimuistin toteutus

- Toteutustavat
  - kanta- ja rajarekisterit
  - sivutus
  - (segmentointi ja sivuttava segmentointi)
- Laitteistotuki
  - MMU - muistinhallintayksikkö
  - rakenne ja toiminta vaihtelee virtuaalimuistin toteutustavan mukaan

# Kanta- ja rajarekistereihin perustuva virtuaalimuisti <sup>(3)</sup>

- Yhtenäiset keskusmuistialueet
  - yksi tai useampi per prosessi
  - erilliset segmentit (ja siis kanta ja rajarekisterit) koodille, datalle, I/O-porteille, yhteiselle datalle, ...
- MMU:ssa BASE ja LIMIT rekisterit
  - segmentti joko oletusarvoinen tai nimettykonekäskyssä
  - kaikki osoitteet loogisia virtuaaliosoitteita
  - helppo tarkistus ja muunnos

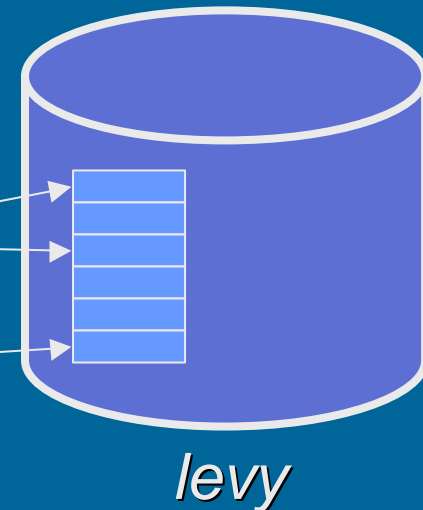
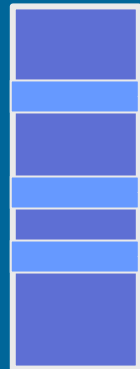
ttk-91: 1 segmentti

Tarkista:	$x < \text{LIMIT}$
Fyysinen osoite:	$\text{BASE} + x$

# Sivuttava virtuaalimuisti

- Looginen osoiteavaruus jaettu saman kokoisiin sivuihin, esim. 1 KB
- Fyysinen muisti jaettu saman kokoisiin sivuraameihin (page frame)
- Jokainen looginen sivu voidaan sijoittaa mihin tahansa (vapaaseen) sivuraamiin
- Sivutaulu pitää kirjaa loogisten sivujen sijainnista

*muisti*



*levy*



# Sivuttavan virtuaalimuistin osoitteenmuunnos (7)

- Loogiset osoitteet sisältävät kaksi kenttää
  - sivun osoite
  - siirtymä sivun sisällä
- Sivutaulun avulla looginen sivun osoite korvataan fyysisellä sivuraamin osoitteella
- Ongelmatilanteet
  - looginen sivun osoite voi olla virheellinen
  - kyseinen looginen sivu ei ole missään sivuraamissa (vielä)  $\Rightarrow$  sivunpuutoskeskeytys

01100110 10110000

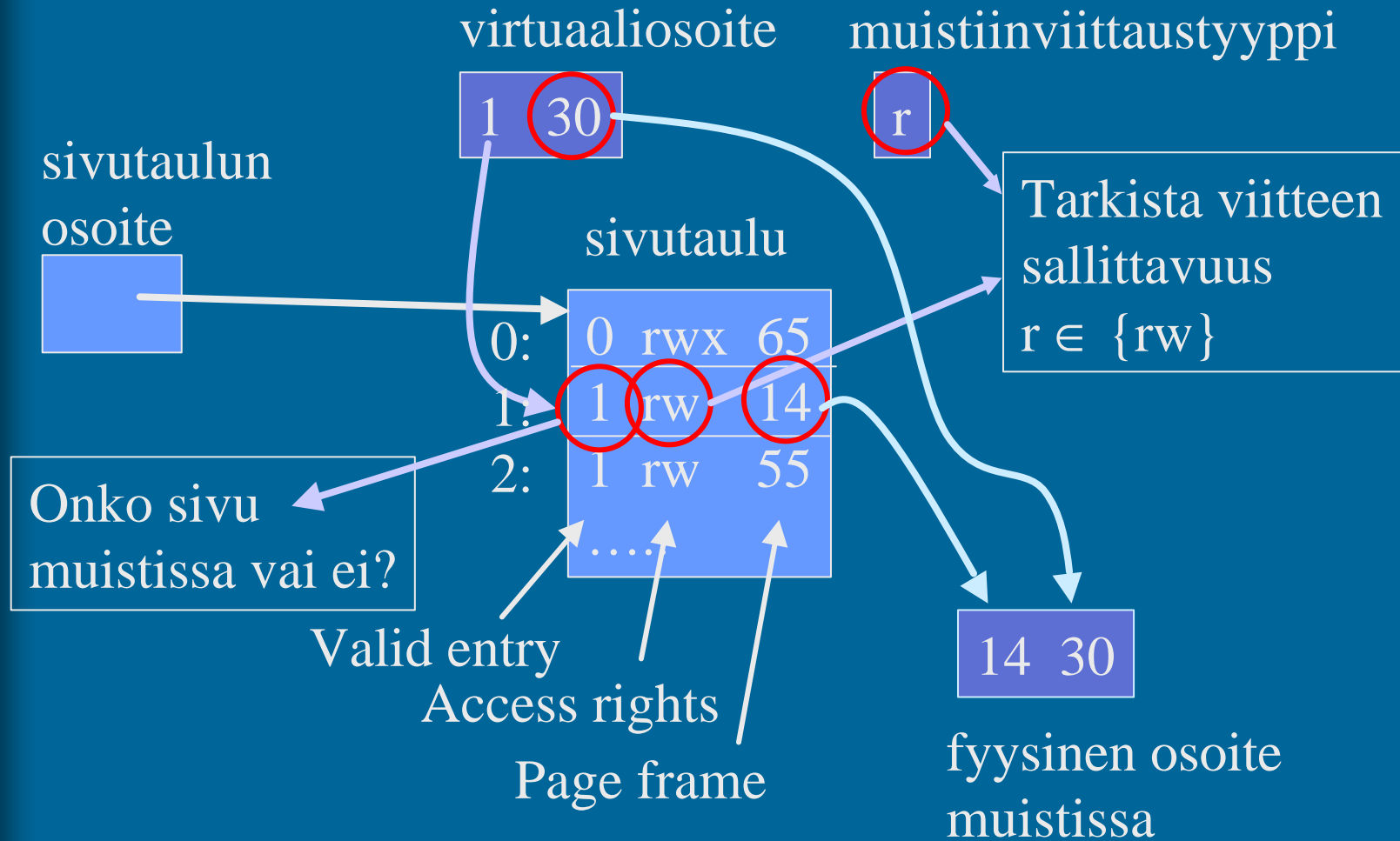
sivu

siirtymä

00111011 10110000

sivuraami siirtymä

# Sivuttava virtuaalimuistin osoitteenmuunnos (4)



# Sivunpuutoskeskeytys

## Sivunpuutos (12)

Lopeta suoritus

Aloita luku levyltä

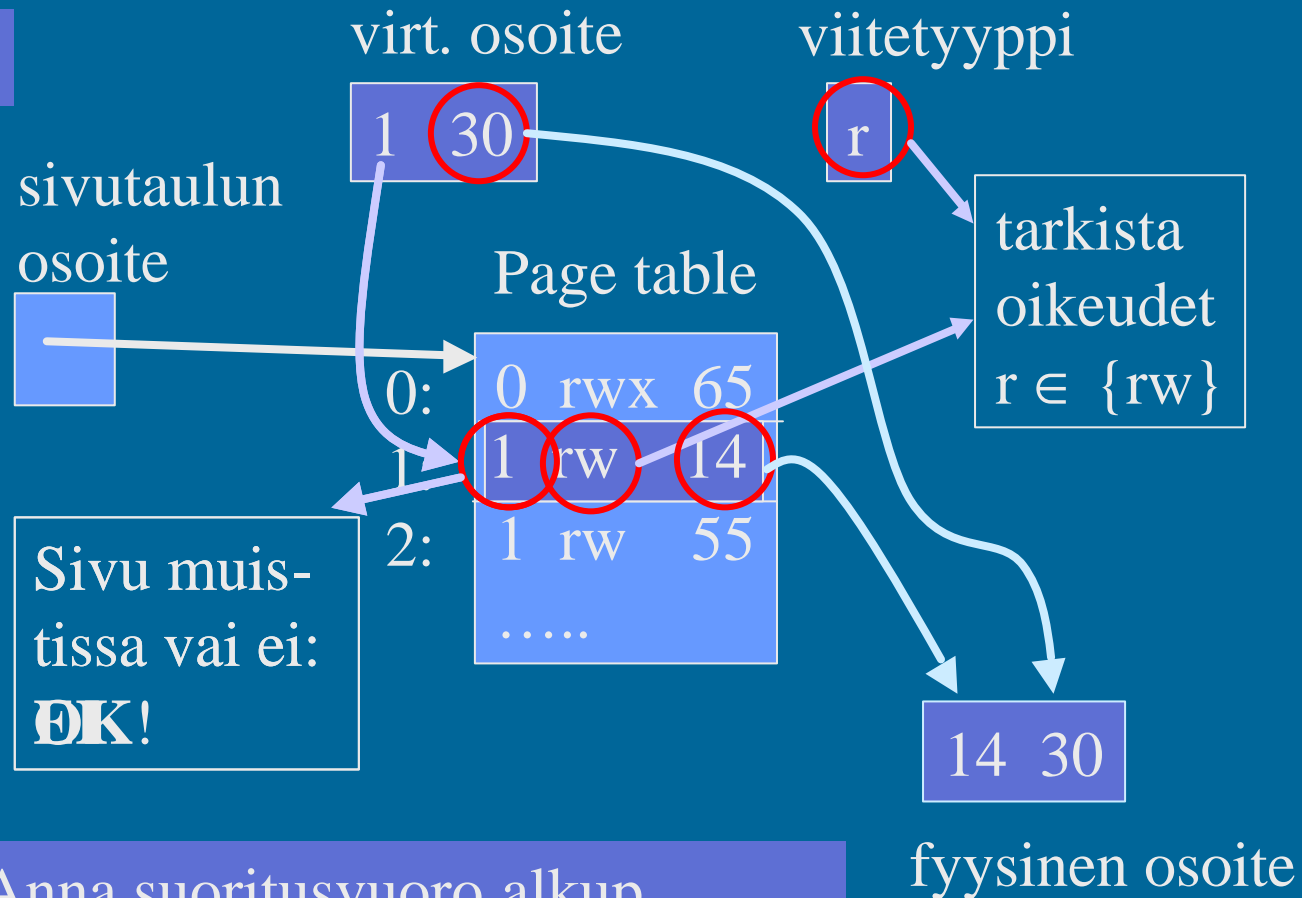
Anna suoritus-  
vuoro seuraavalle  
prosessille

I/O keskeytys

Sivu 1 luettu,  
päivitä sivutaulu

Siirrä alkuper.  
prosessi  
ready-to-run  
jonoon

Anna suoritusvuoro alkup.  
prosessille, samaan konekäskyyn



# Sivuttava virtuaalimuisti (2)

- Fyysinen muisti jaettu sivuihin
  - prosessin looginen muistiavaruus on fyysisesti talletettu erillisissä samankokoisissa sivuissa
- Sivutaulu
  - joka prosessilla omansa (PCB:ssä linkki siihen)
  - sijaitsee muistissa
  - voi olla hyvin iso  
(alkio per virtuaalisen osoiteavaruuden sivu)

# Osoitteen muunnos (3)

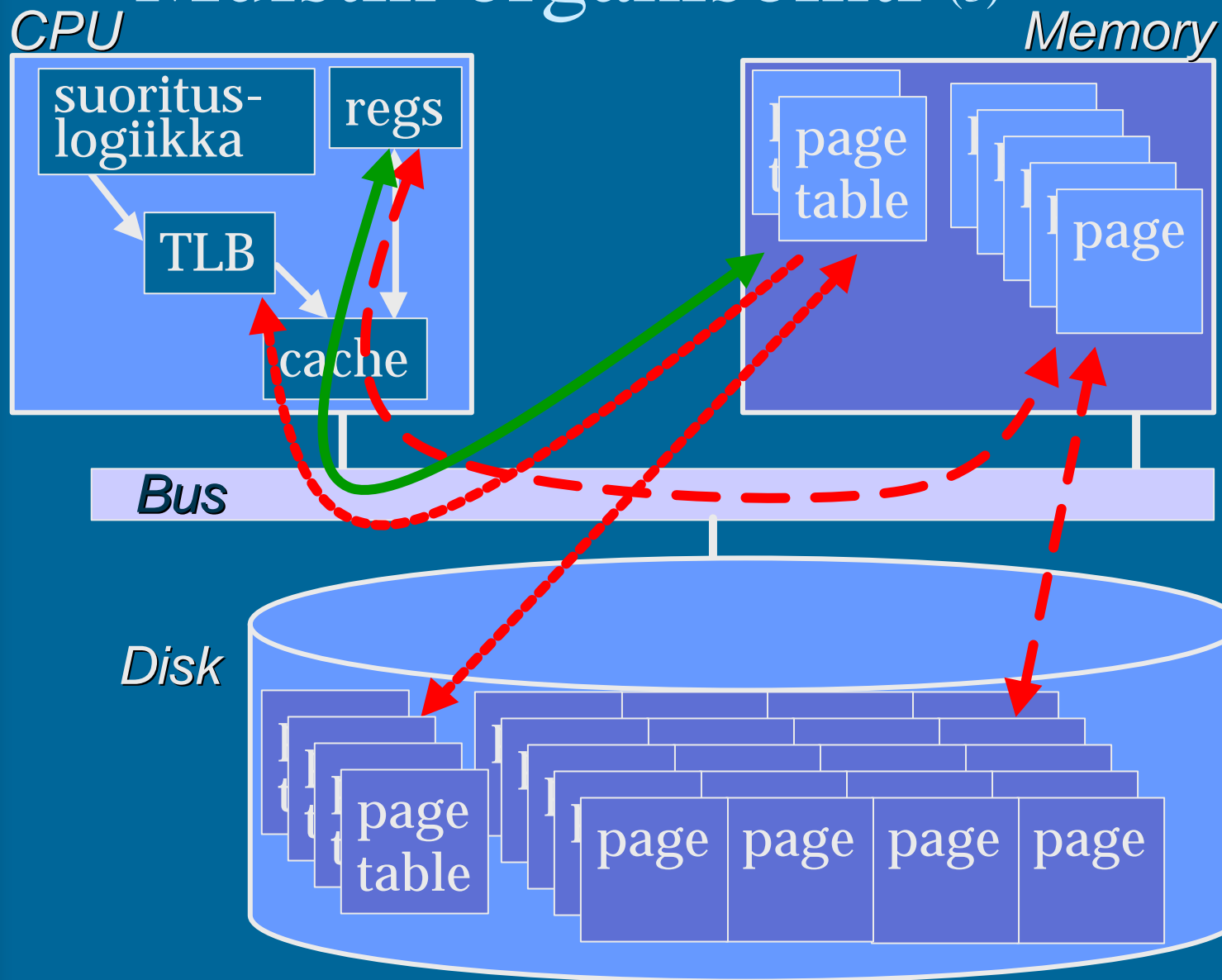
- MMU tekee jokaiselle osoitteelle
  - koodi, data
  - useammin kuin kerran per käsky!
- Ei voi hakea osoitteenmuutostietoja muistista sivutaulusta joka kerta!
  - liian kallis hinta virtuaalimuistista?
- MMU pitää tallessa (”välimuistissa”) viimeisimmät osoitteenmuunnokset
  - osoitteenmuunnostaulukko TLB (Translation Lookaside Buffer)
  - 99.9% osumasuhde?

# TLB - osoitteenmuunnostaulukko <sup>(3)</sup>

- TLB “osuma” ?
  - Osoitteenmuunnos löytyy TLB:stä - nopea
- TLB “huti”?
  - täytyy lukea tiedot sivutaulusta muistista
  - vie aikaa vähän
  - suoritin odottaa (tyhjäkäynnillä)
- Toteutus
  - samalla tavalla kuten tavallinenkin välimuisti
  - välimuistin rivin (datan) asemesta löytyykin haluttu tiedon fyysinen osoite

Fig. 7.19 [Stal99]

# Muistin organisointi (3)



# TLB ja välimuisti (4)

- Yleensä osoitteenmuunnos ensin ja sitten välimuistihaku
- Välimuisti voi perustua myös virtuaalisosoitteisiin
  - voi tehdä TLB-haun ja välimuistahaun samanaikaisesti
  - nopeuttaa
- Toteutukset hyvin samankaltaisia
  - TLB toteutus optimoitu ajalliselle paikallisuudelle
- Kumpikin ovat tuntumattomia (”näkymättömiä”) niitä hyödyntäville ohjelmille
  - KJ rutiinit voivat ”nollata” molemmat, mutta eivät lukea niiden sisältöä



# TLB vs. Välimuisti

## TLB huti

- CPU odottaa
- HW toteutus
- Tuntumaton prosessille
- Tiedot kopioidaan muistista TLB:hen
  - sivutaulusta

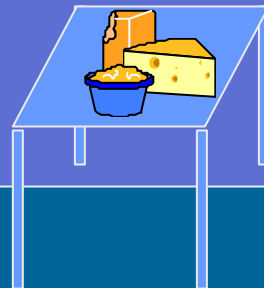
## Välimuistihuti

- CPU odottaa
- HW toteutus
- Tuntumaton prosessille
- Tiedot kopioidaan muistista välimuistiin
  - sivuraamista

# TLB hudit vs. sivunpuutoshudit

## TLB huti

- Suoritin odottaa tyhjäkäynnillä
- HW toteutus
- Data kopioidaan sivutaulusta TLB:hen
- kesto  
4 ns (?)



## Sivunpuutos

- Prosessi odottaa ja suoritusvuoro toisella prosessilla
- SW toteutus
- Data kopioidaan levyltä muistiin
- kesto  
30 ms (?)



# Tiedostojärjestelmä (5)

- KJ:n osa, hallitsee kaikkia tiedostoja
- Valvoo oikeuksia tiedostoa avattaessa
- Muuntaa tiedostonimet fyysisiksi osoitteiksi
- Ylläpitää taulukoita, joista näkee mitä kohtaa mistäkin tiedostosta kukin prosessi on käsittelemässä
- Tiedostojärjestelmä lukee ja kirjoittaa tiedostoja suurina kerralla käsiteltävinä lohkoina (0.5-8 KB?)
  - käyttäjätason prosessit käsittelevät tiedostoja tavuittain, niiden ei tarvitse tietää tiedoston todellista fyysistä rakennetta (laiteajuri huolehtii siitä)

# Levymuisti <sup>(9)</sup>

ks. Fig.2.20 [Tane99]

- Levykkö

0.5MB-16GB/levykkö

- pyörii nopeasti (koko ajan?)

~ 3600-10800 rpm

- luku/kirjoituspäät liikkuvat kaikki yhtä aikaa

- monta levyä

~ 1-16 levyä/levykkö

- Levypinta

ks. Fig.2.19 [Tane99]

- 2 per levy (tai 1)

~ 2000-3000 uraa/pinta

- ura

~ 20-100 sektoria/ura

- sektori: pienin kerralla osoitettavissa oleva alue

~ 0.5-8KB/ura

- sylinteri: päällekkäin olevat urat (luku/kirj. pää samalla kohtaa)

~ 1-31 uraa/sylinteri

# Levymuistin saantiaika <sup>(2)</sup>

- Tiedon osoite: levyypinta + ura + sektori

- laiteajuri etsii KJ-taulukoista loogisen osoitteen perusteella

ks. Fig.2.20 [Tane99]

- Saantiaika:

- hakuvarren siirtoaika

(seek time)

- odota kunnes sektori kohdalla (pyörähdysviive)

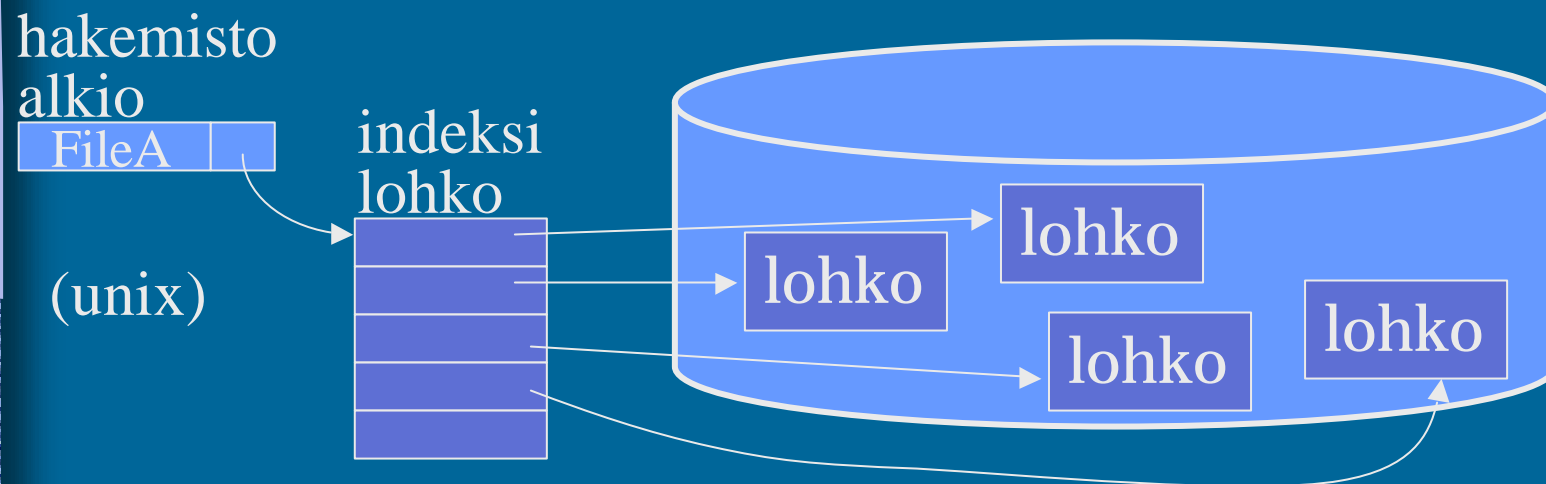
(rotational delay)

- siirrä sektorin verran tietoa

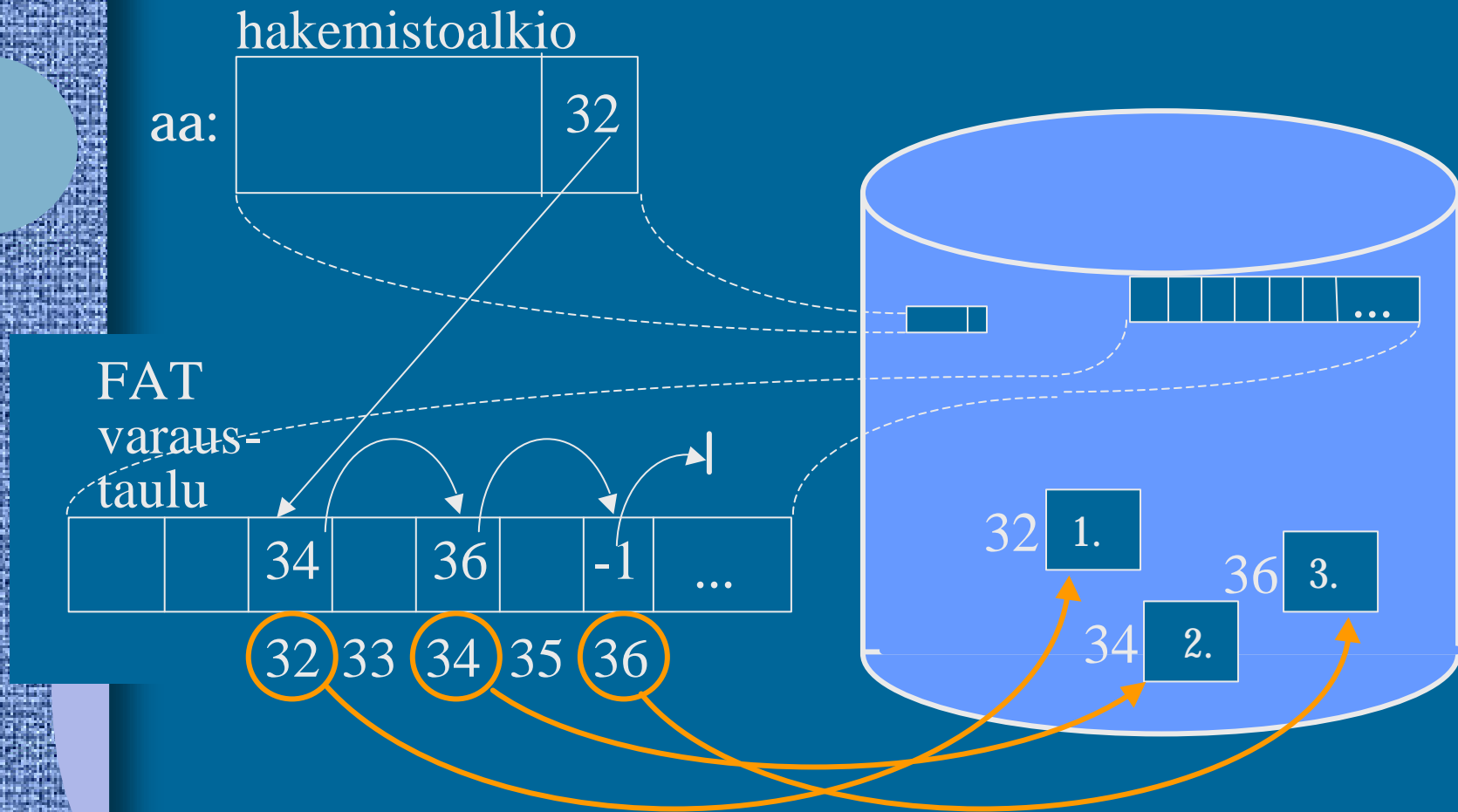
(data transfer time)

# Tiedoston talletus levyille (2)

- Tiedosto koostuu useista lohkoista
  - lohko per sektori
- Levyn hakemistossa on tieto kunkin tiedoston käyttämistä lohkoista
  - luetaan lohkot annetussa järjestyksessä



# FAT - File Allocation Table (DOS)



DOS levykkeet: 1.44MB, lohko 512 B, 2.9K lohkoa  
entä: 1 GB, lohko 64 KB, 64K lohkoa OK?

# Mikä on hyvä levylohkon koko?

- Mihin käyttöön?
- Videokuvan talletus/playback?

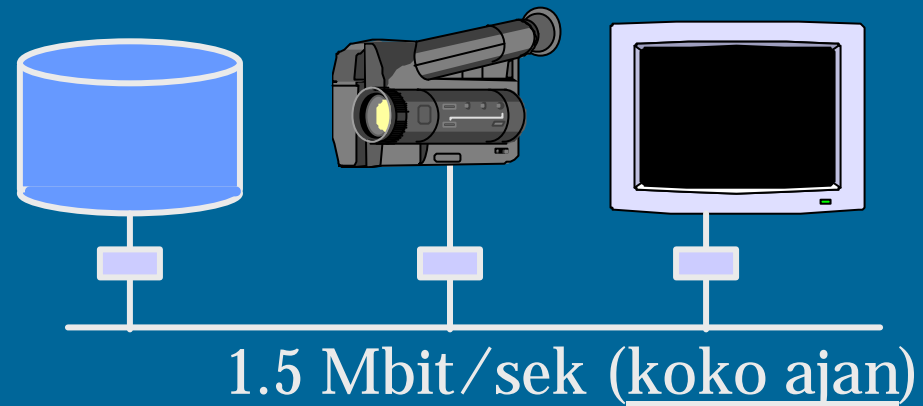
- 1 minuutti

- 12.3 MB

- lohko 4KB?  
osoitetaulu?

- epäsuorat viittaukset? riittääkö nopeus?

- entä jos 1 tunti?





# Erilaisia levyjä

- Kiintolevy
- ZIP levyke
- Levyke
- CD-ROM, CD-R, CD-RW
- DVD

# Kiintolevy (7)

- Kiinteä, ei vaihdettavissa oleva levy
- Väylä: IDE, EIDE, ATA, SCSI, PCMCIA
- Tila: 150 MB - 70 GB
- Haku aika: 5-15 ms
- 1-10 levyä
- Pyörimisnopeus: 4500-10800 rpm
- Siirtonopeus: 5-50 MB/sec



16 g



43 x 36 x 5 mm



# Zip & Jaz levykeasemat <sup>(6)</sup>



- Vaihdeettava levyke
- Väylä: USB, rinnakkaisväylä, SCSI, ...
- Tila: 100 MB - 2 GB
- Hakuaika: 10-30 ms
- Pyörimisnopeus: 3000-5400 rpm
- Siirtonopeus: 1-6 MB/sec

# Levykeasema (6)

- Vaihdeettava levyke
- Väylä: ...
- Tila: 1.44 MB
- Hakuaika 90 ms
- Pyörimisnopeus: 300 rpm
- Siirtonopeus 0.05 MB/sec



# CD - Compact Disc <sup>(9)</sup>

- Vaihdelevy
- CD-R (Recordable)
- CD-RW (Rewritable)
- Yksi pitkä spiraalimainen ”ura”
- Väylä: SCSI, USB, ...
- Tila: 650 MB
- Haku-aika 90 ms
- Pyörimisnopeus 200-9000 rpm
- Siirtonopeus 0.1-2 MB/sec



# DVD - Digital Versatile Disk (9)

- Vaihdelevy
- DVD-ROM
- DVD-R (Recordable)
- DVD-RAM (kuten tavallinen kovalevy)
- Väylä: EIDE, ATAPI, SCSI
- Tila: 4.7-17 GB
- Haku-aika 100-180 ms
- Pyörimisnopeus 2000-8000 rpm
- Siirtonopeus 2-8 MB/sec





# -- Luennon 9 loppu --

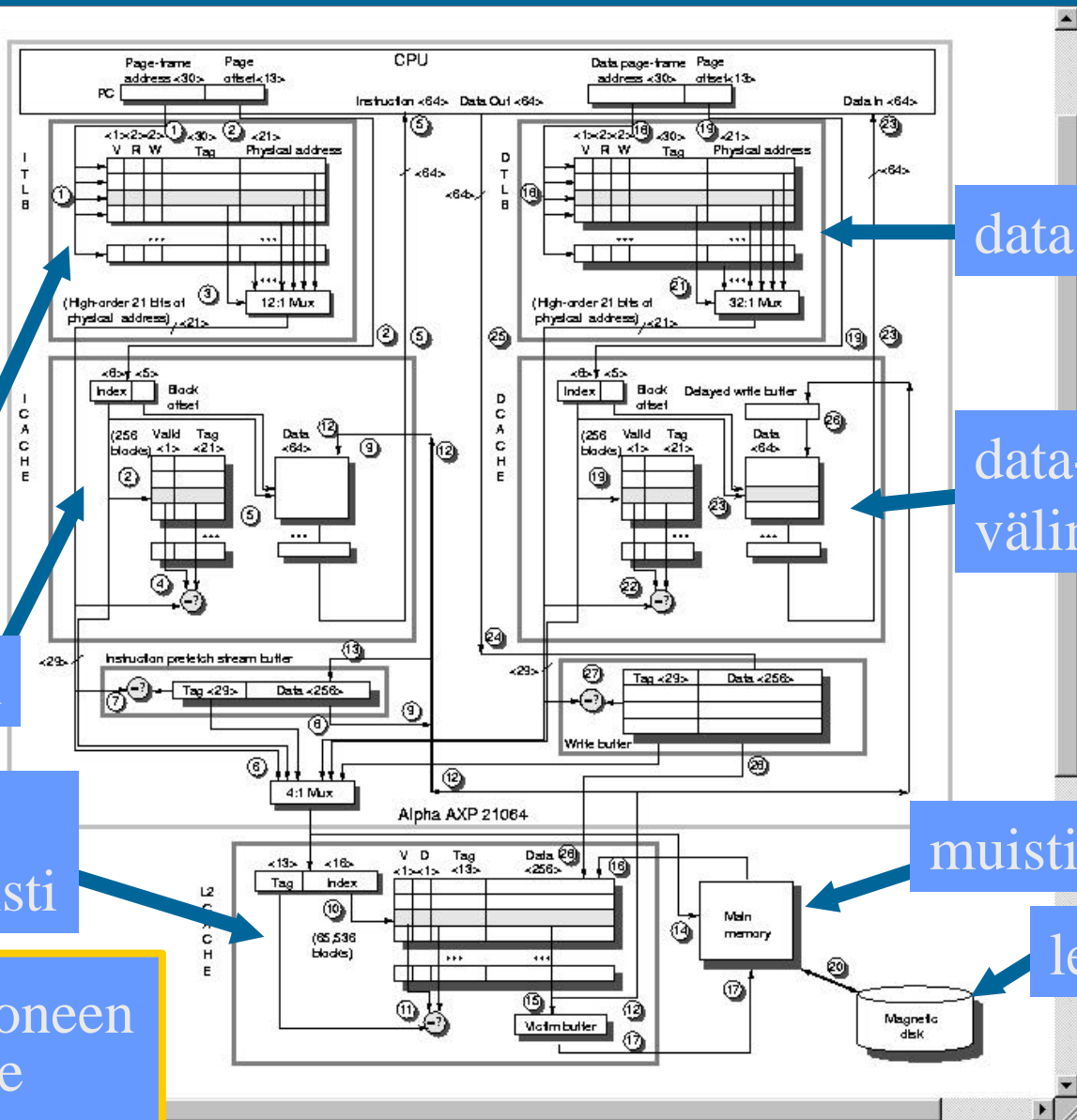
Fig. 5.47 from  
Hennessy-Patterson,  
Computer Architecture  
Alpha AXP 21064  
memory hierarchy

käskyjen TLB

käskyvälimuisti

Käskyjen ja datan  
yhteinen L2 välimuisti

Lisää tietoa?  Tietokoneen rakenne



data TLB

data-  
välimuisti

muisti

levy