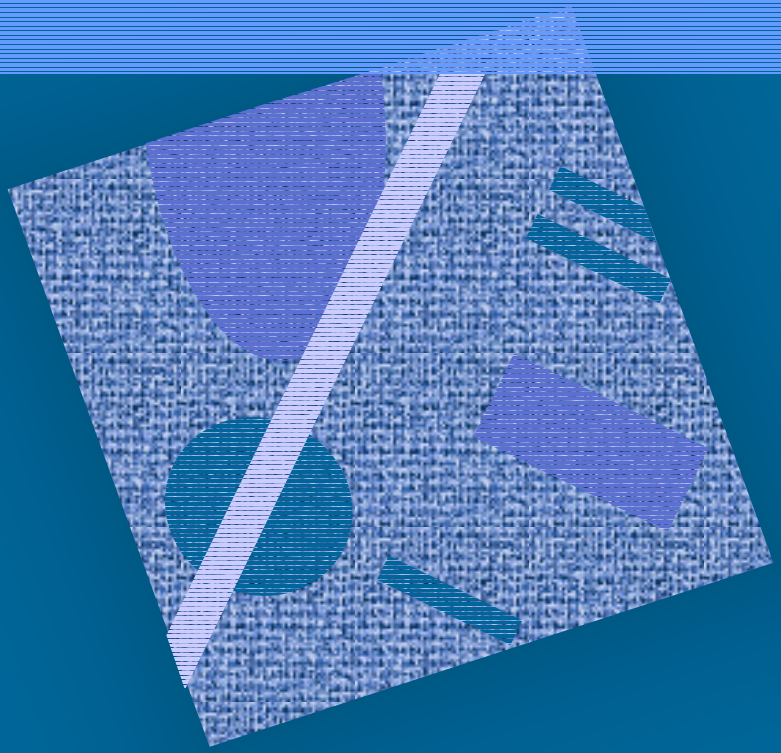
Luento 12
Yhteenveto



Keskeiset asiat

Mitä hyötyä tästä on?

Mitä seuraavaksi?

Kurssit?

Asiat?

Tavoitteet (4)

- Ymmärtää tietokonejärjestelmän keskeiset piirteet sillä suoritettavan ohjelman näkökulmasta
- Miten tietokonejärjestelmä suorittaa sille annettua ohjelmaa?
- Minkälaista koodia suoritin ymmärtää?
- Mikä on käyttöjärjestelmän rooli?

Mitä hyötyä tästä on? (2)

- Ohjelman suoritusnopeus perustuu suorittimen (CPU) suorittamiin konekäskyihin, ei pelkästään ohjelman korkean tason esitysmuotoon
- Ylemmän tason asioiden ymmärtäminen on helpompaa (mahdollista), kun ymmärtää alemman tason asiat

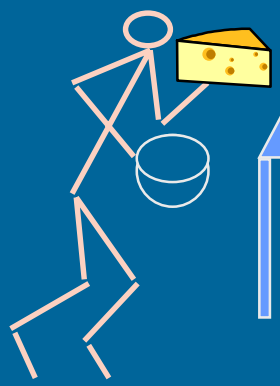
Keskeisiä asioita (10)

- Järjestelmä kokonaisuudessaan, nopeuserot
- Esimerkkikone ja sen käyttö
- Konekielinen ohjelmointi
- Suoritin, rekisterit, väylät, muisti
  - konekäskyjen suoritussykli, keskeytykset
- Aktivointitietuepino, aliohjelmien toteutus
- Tiedon esitysmuodot (ohjelma vs. laitteisto)
- Prosessi ja sen toteutus (PCB)
- I/O laitteet
  - laiteajurit, laitekeskeytykset, levymuisti
- Ohjelmien suoritus järjestelmässä
  - käännös, linkitys, lataus, tulkinta, emulointi, simulointi

Nopeuserot: Teemun juustokakku

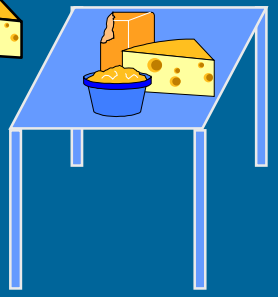
Rekisterien, välimuistin, muistin, levymuistin ja magneettinauhan nopeudet suhteutettuna juuston hakuaikaan juustokakkua tehdessä?

käsi



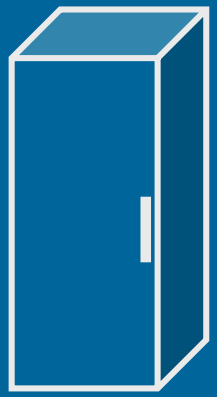
0.5 sek
(rekisteri)

pöytä



1 sek
(väli-muisti)

jääkaappi

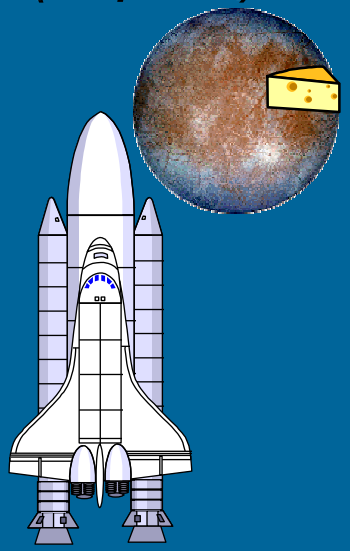


10 sek
(muisti)

kuu

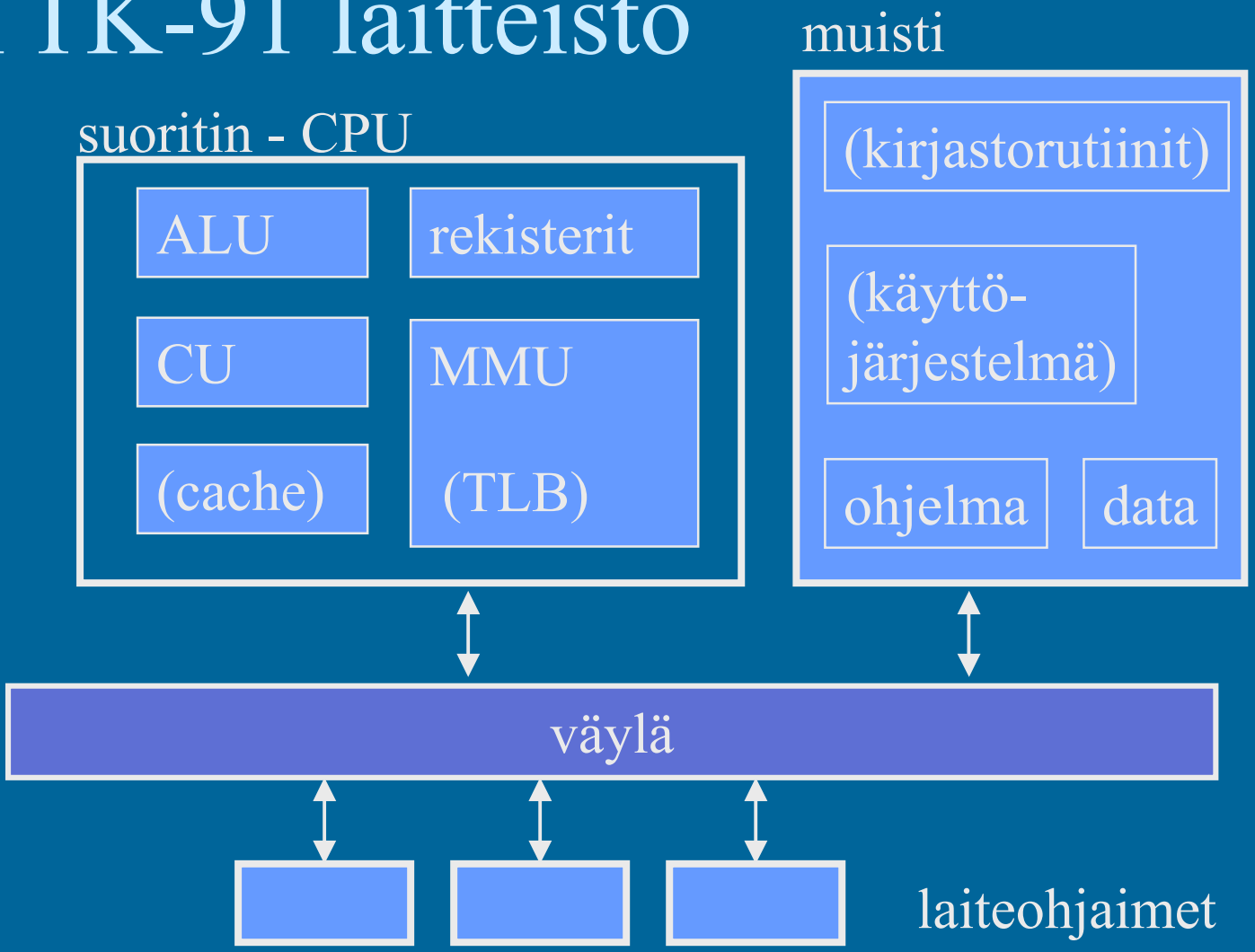
12 pv
(levy)

Europa (Jupiter)



4 v
(nauha, ihminen)

Esimerkkikone: TTK-91 laitteisto



Konekielinen ohjelmointi

```
for (int i=20; i < 50; ++i)
    T[i] = 0;
```

```
I       DC      0
        …
        LOAD  R1, =20
        STORE R1, I

Loop    LOAD  R2, =0
        LOAD  R1, I
        STORE R2, T(R1)

        LOAD  R1, I
        ADD   R1, =1
        STORE R1, I

        LOAD  R3, I
        COMP  R3, =50
        JLES  Loop
```

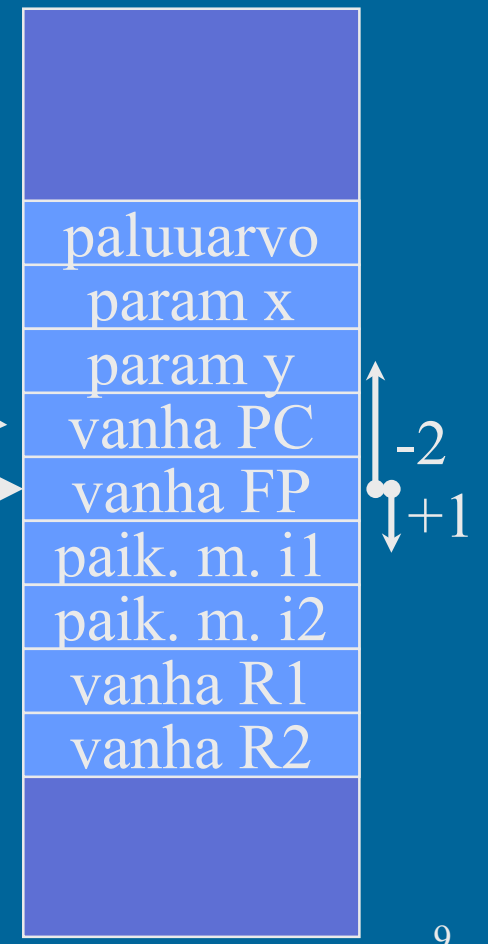
Käskyjen nouto- ja suoritussykli

Aktivointitietue

(activation record, activation frame)

int funcA (int x,y);

- Aliohjelman toteutusmuoto (ttk-91)
  - funktion paluuarvo (tai kaikki paluuarvot)
  - kaikkien (sisäänmeno- ja ulostulo-) parametrien arvot
  - paluuosoite
  - kutsukohdan aktivointitietue
  - kaikki paikalliset muuttujat ja tietorakenteet
  - aliohjelman ajaksi talletettujen rekistereiden alkuperäiset arvot



IEEE 32-bit FP Standard

sign exponent mantissa or significand

1/8 = 0.1250
1/16= 0.0625
0.1875

- 23 bittiä mantissalle, siten että ...

1) Binääripiste (.) on heti ensimmäisen bitin jälkeen

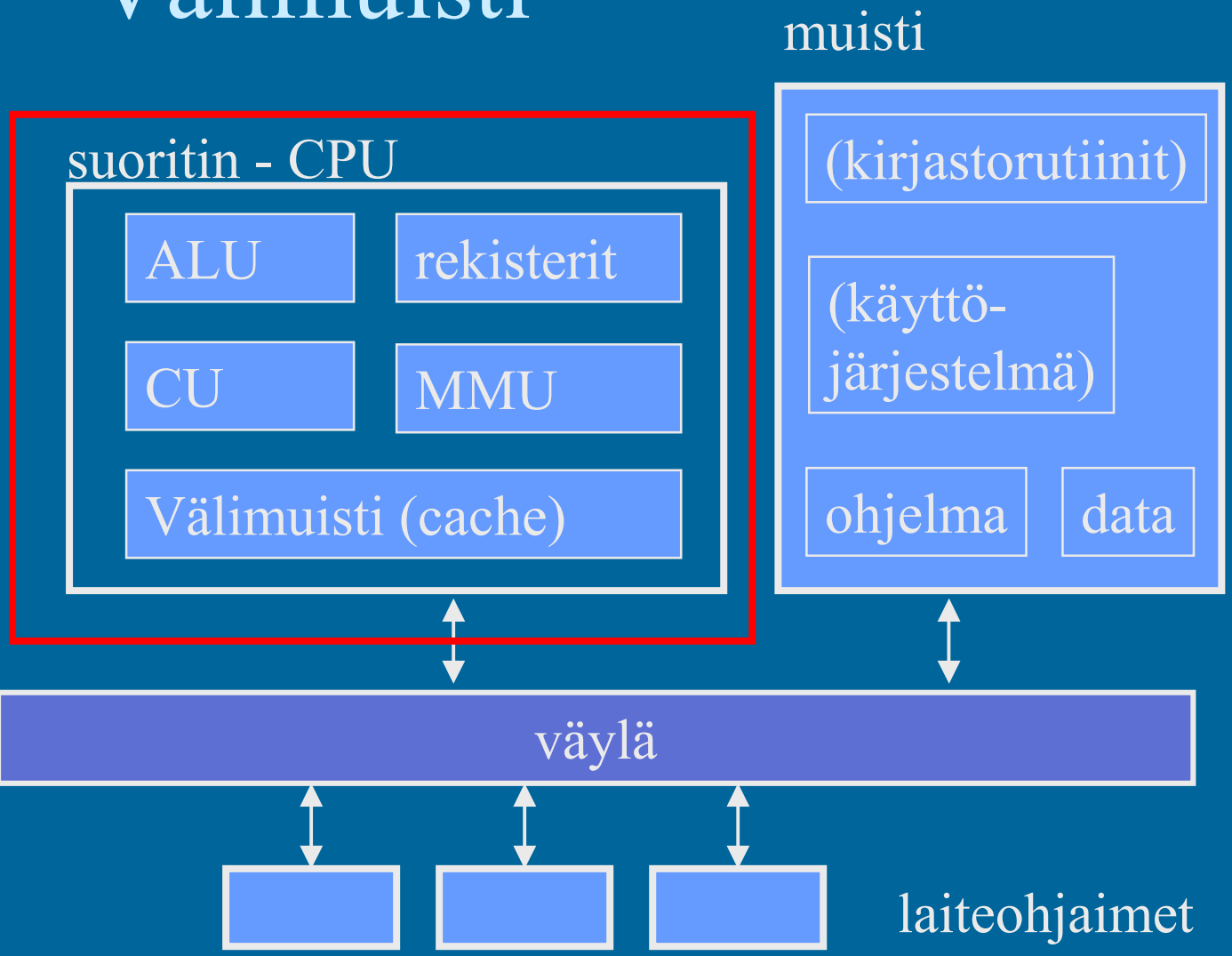
2) Mantissa on normalisoitu: vasemmanpuolimmainen bitti on1

3)Vasemmanpuolimmaista (eniten merkitsevä) bittiä (1) ei talleteta (implied bit)

mantissa	eksponentti
0.0011	"15"
1.1000	"12"
1000	"12"

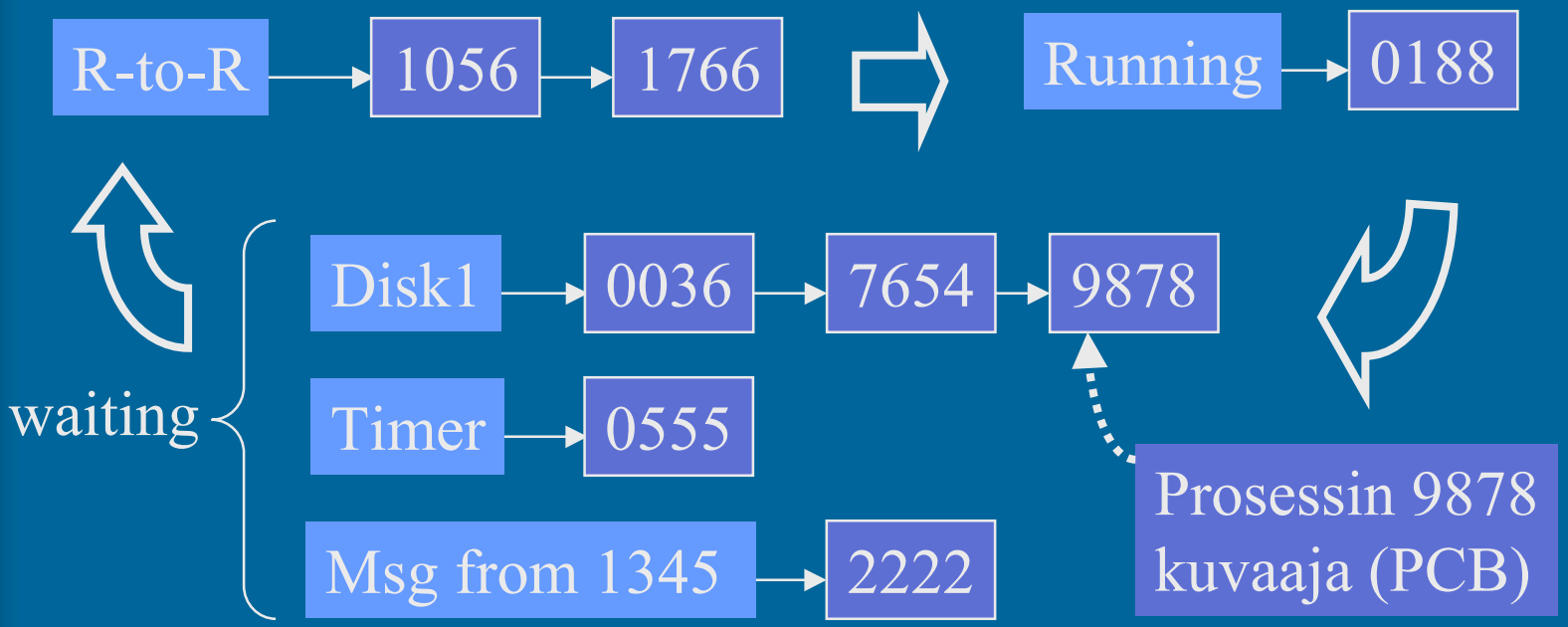
24 bitin mantissa!

Välimuisti



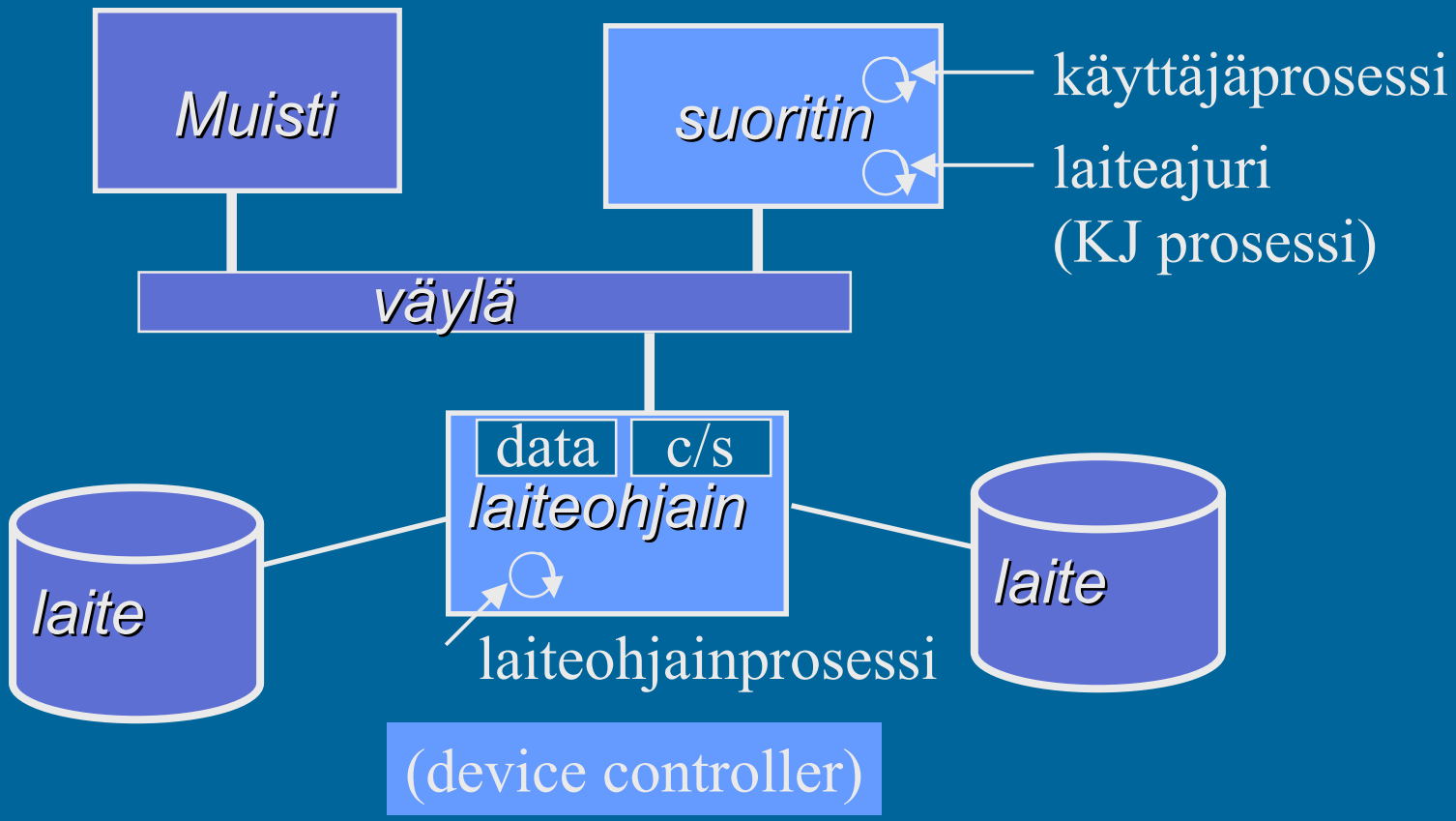
Prosessin tilat ja elinkaari

Prosessit jonoissa ja PCB



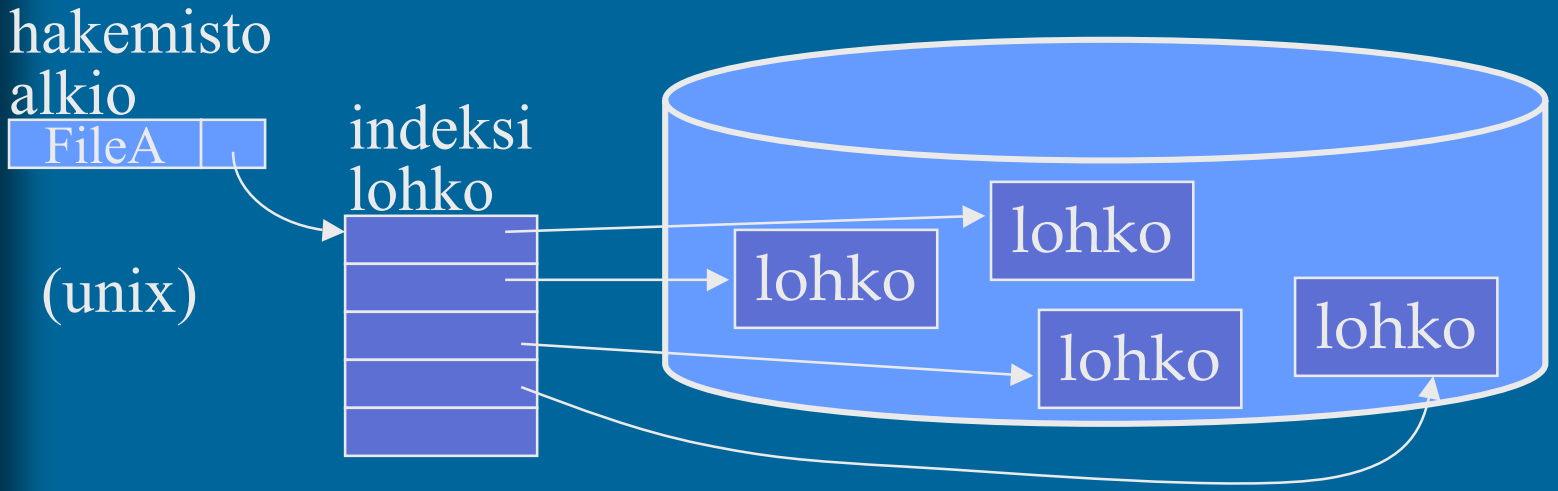
Vuoronanto:
valitse seuraava prosessi Ready-to-Run -jonosta ja
siirrä se suoritukseen CPU:lle
(kopioi tämän prosessin suorittimen tila suorittimelle)

Laiteohjain ja laiteajuri



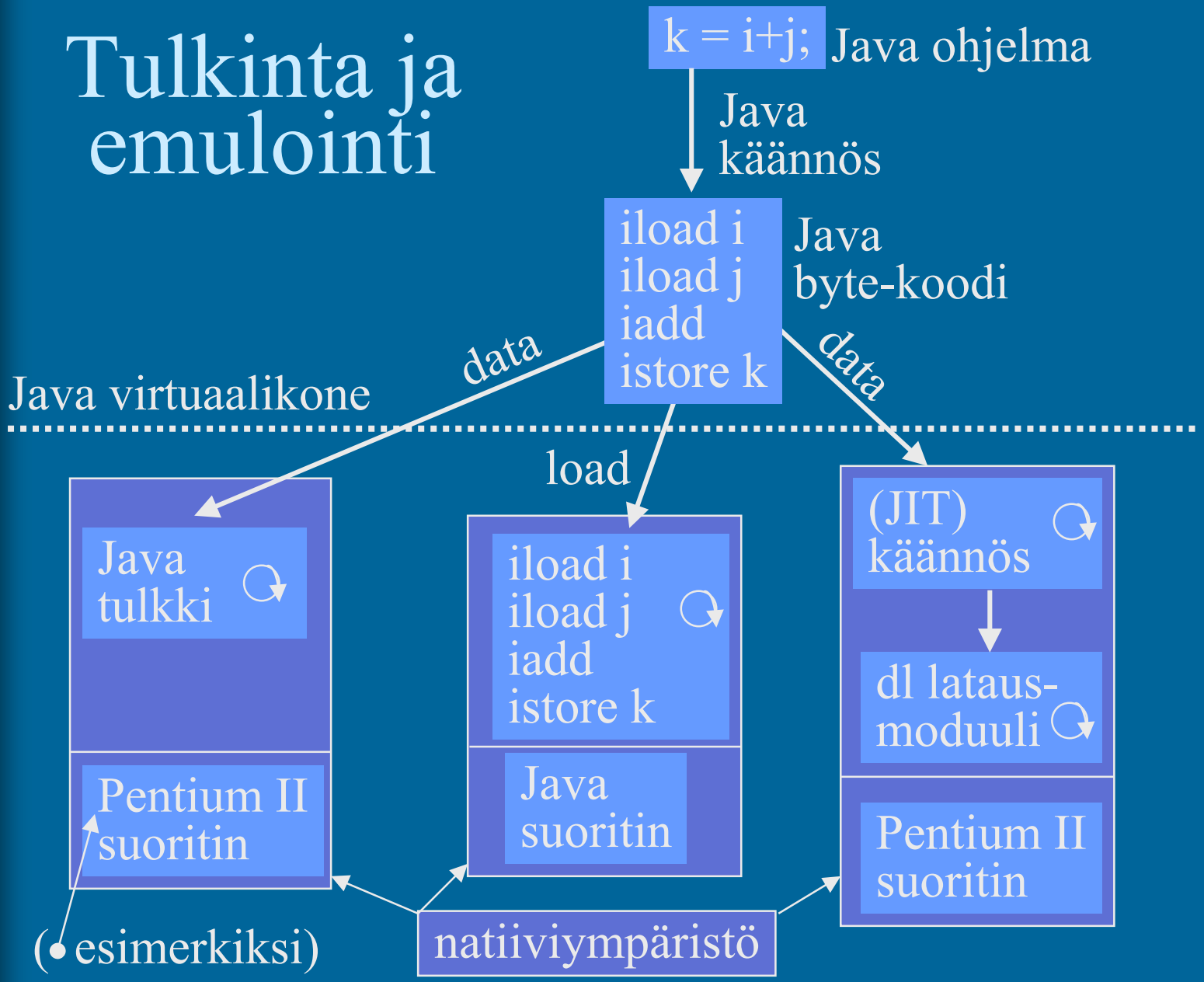
Levyjen käyttö

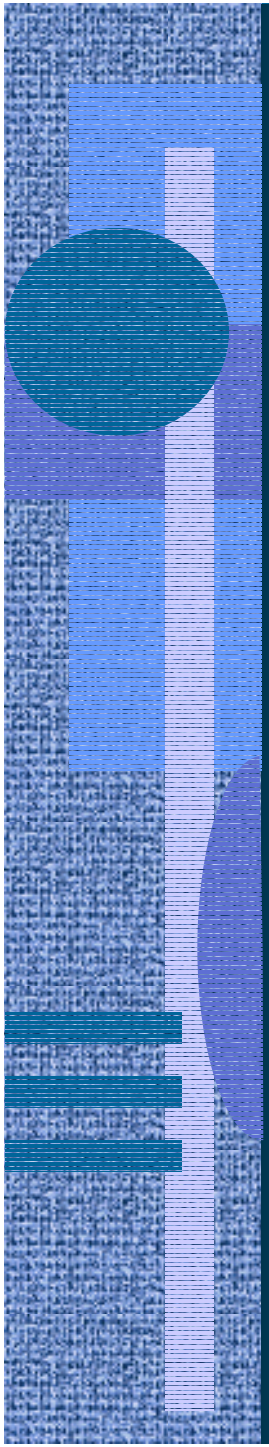
- Tiedosto koostuu useista lohkoista
  - lohko per sektori
- Levyn hakemistossa on tieto kunkin tiedoston käyttämistä lohkoista
  - luetaan lohkot annetussa järjestyksessä



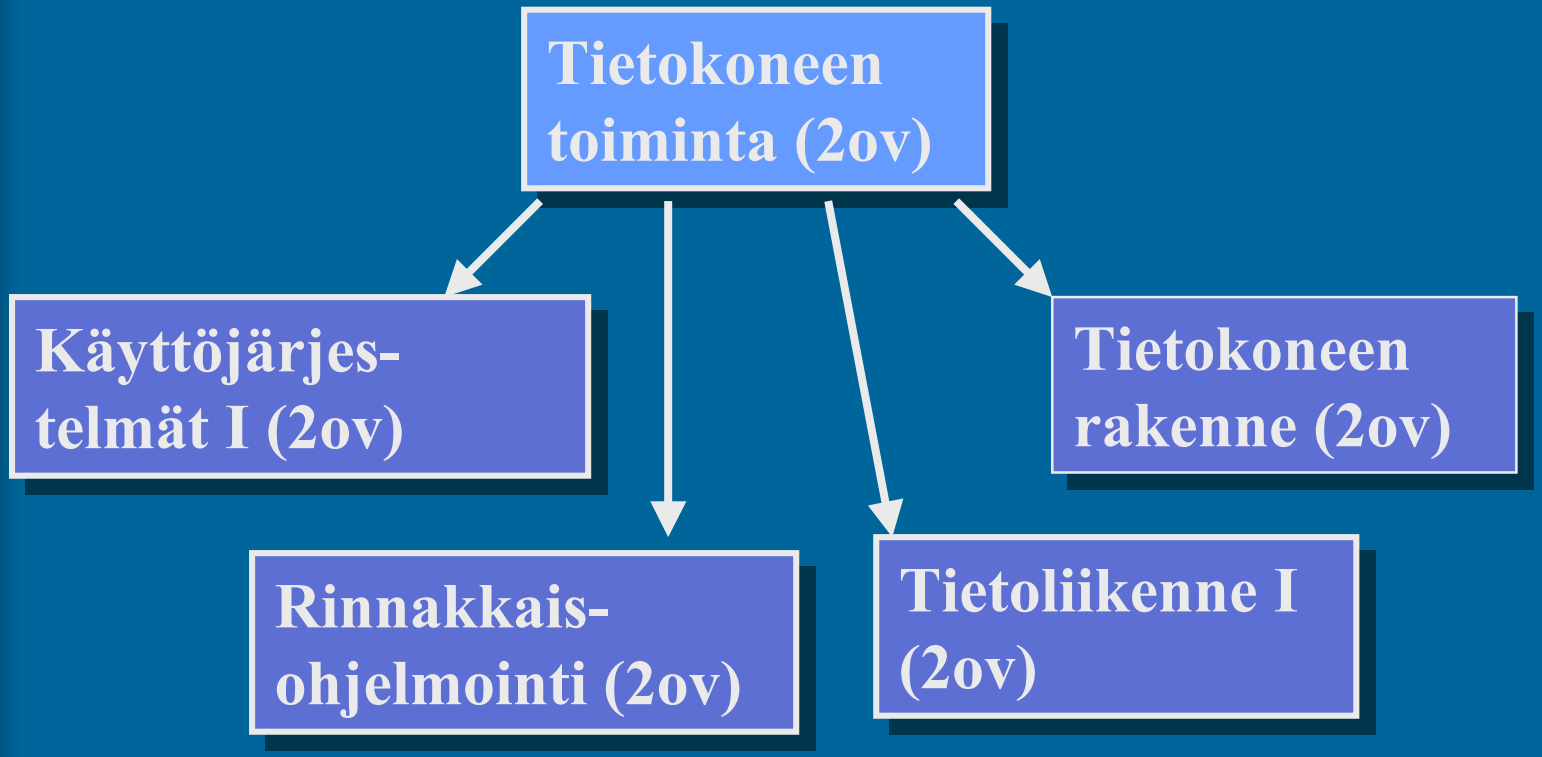
Lausekielestä suoritukseen

Tulkinta ja emulointi

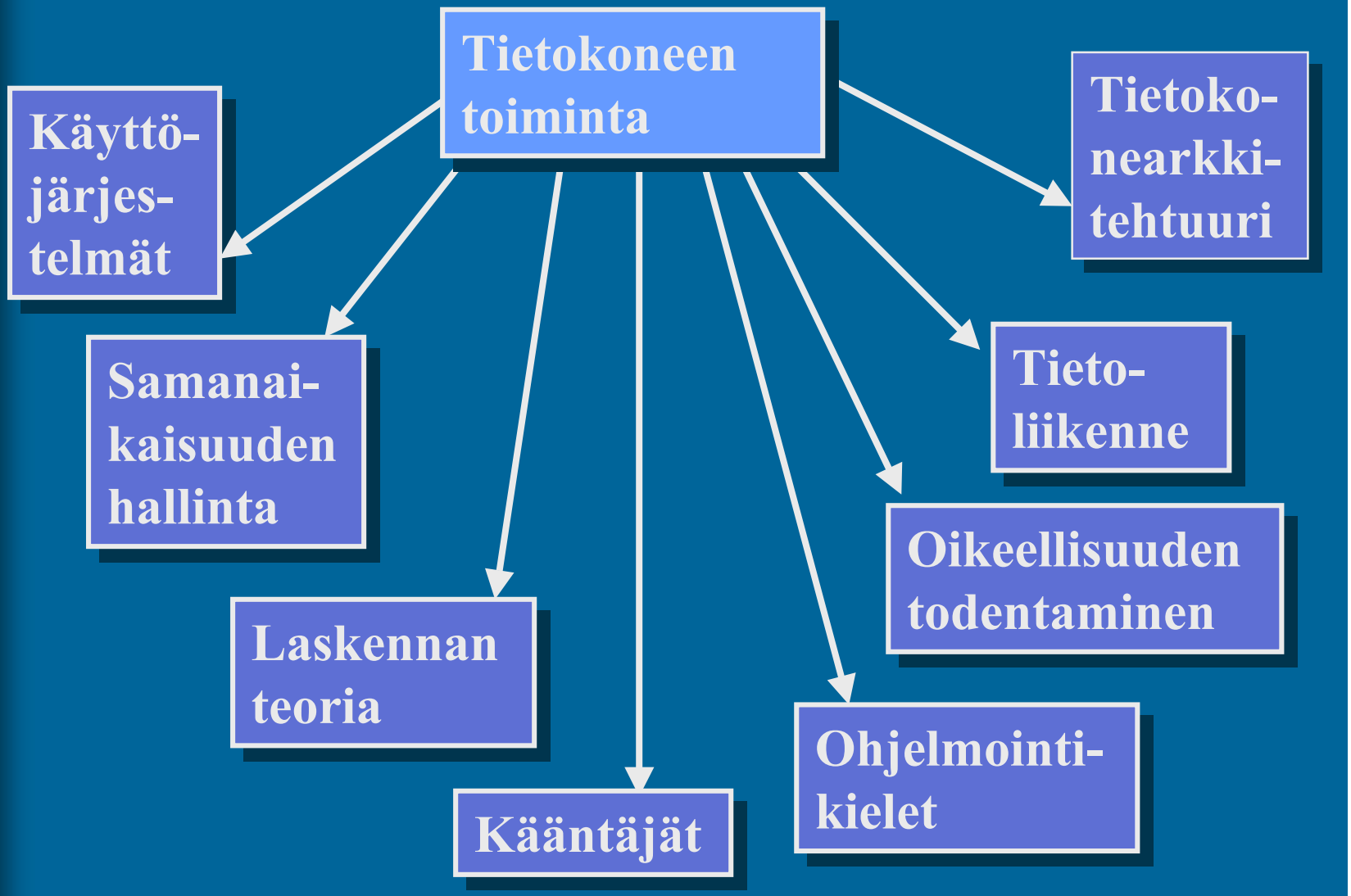




Kurssien välisiä suhteita



Asioiden välisiä suhteita (8)



Tietokoneen rakenne, 2 ov

- Yksi taso alaspäin TITOsta
- Sopiva 2. vuoden opiskelijalle
- Useissa yliopistoissa yhdistetty TITOon
- ”Miten kellopulssi saa suorittimen suorittamaan konekäskyjä?”
- ”Miten suorittimen aritmetiikka on toteutettu?”
- Usea käsky on todellisuudessa suorituksessa samanaikaisesti
  - Miten tämä toteutetaan, mitä ongelmia siitä seuraa ja miten noita ongelmia ratkotaan?
- Jatkoa syventävällä tasolla
  - Tietokonearkkitehtuurit, 4 ov

TiKRa ....

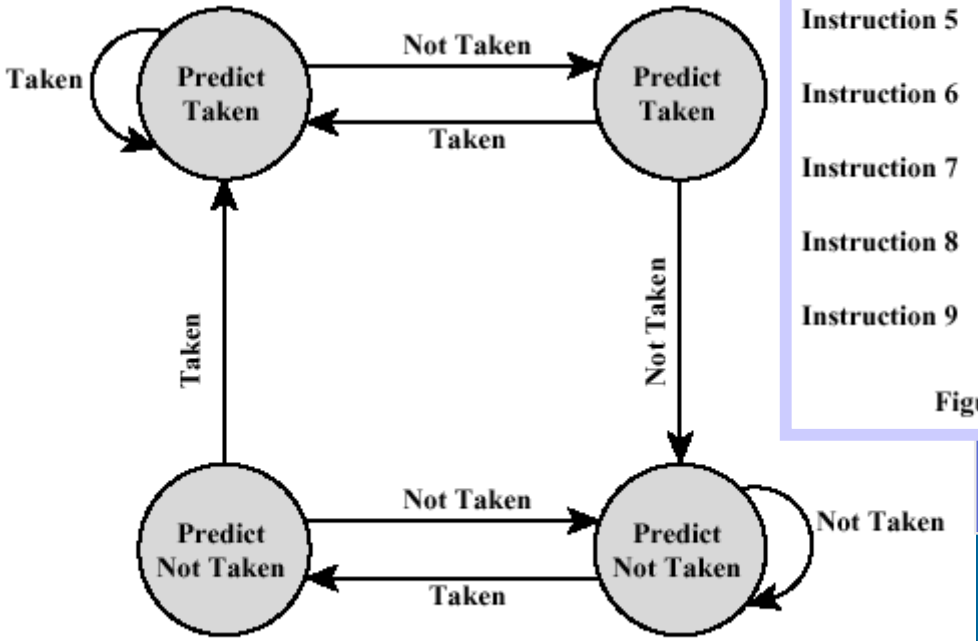


Figure 11.16 Branch Prediction State Diagram

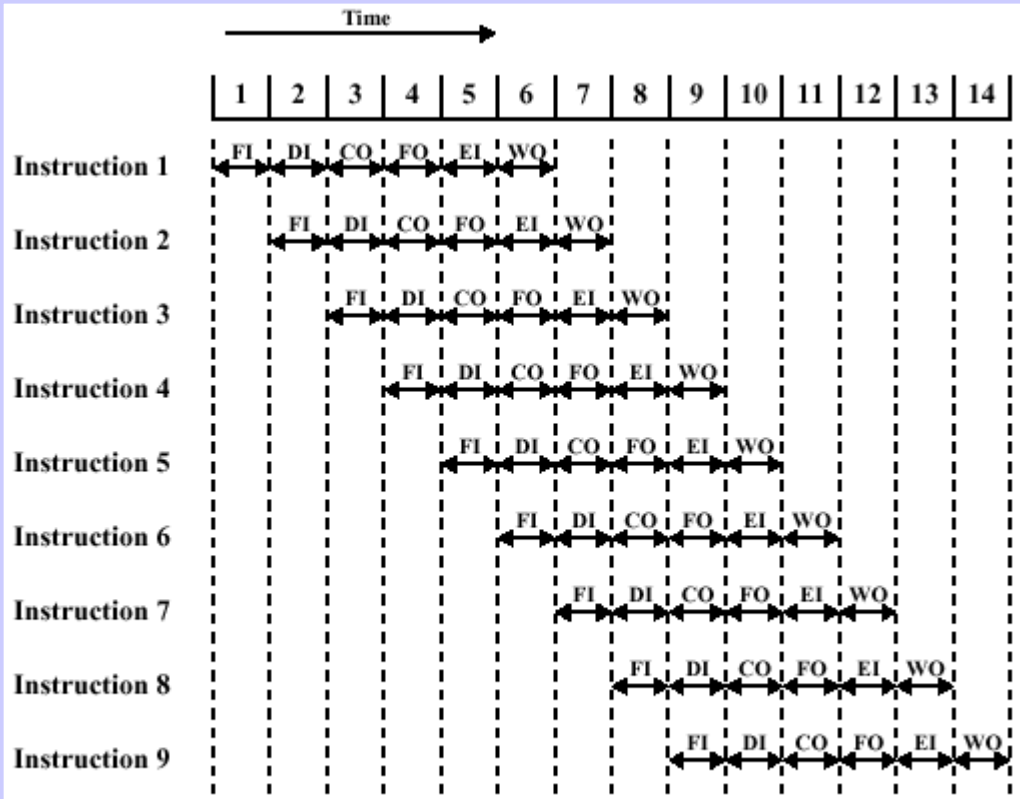


Figure 11.11 Timing Diagram for Instruction Pipeline Operation

[Stal99]

Käyttöjärjestelmät I, 2 ov

- Sopiva 2. vuoden opiskelijalle
- Käyttöjärjestelmän rooli yhden prosessin valvojana
- Täsmentää ja jatkaa TITOn käyttöjärjestelmien piirteiden esittelyä
- Samanaikaiset prosessit resurssien käyttäjinä
- Systeemin resurssien jakelu
- Prosessien vuoronanto (skedulointi)
- Jatkoa perustasolla ja syventävällä tasolla
  - Käyttöjärjestelmät II, 2 ov
  - Käyttöjärjestelmämetodiikka, 3 ov

KJ ...

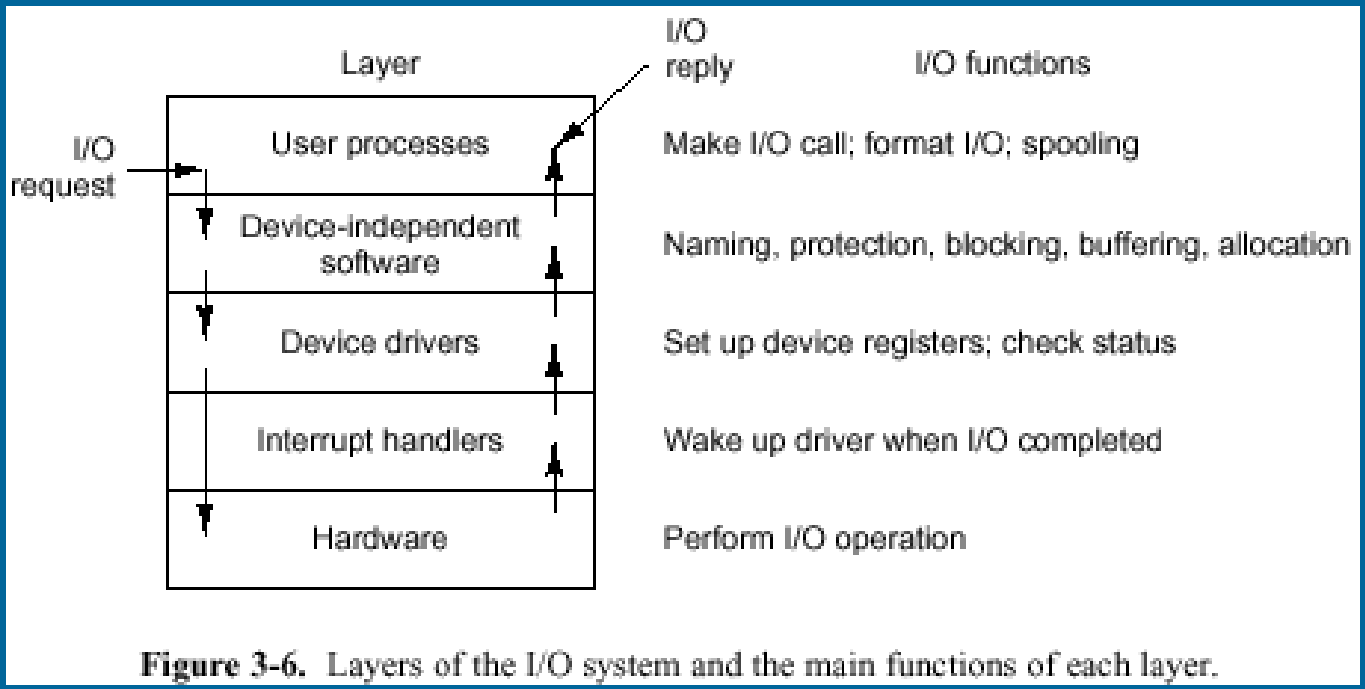
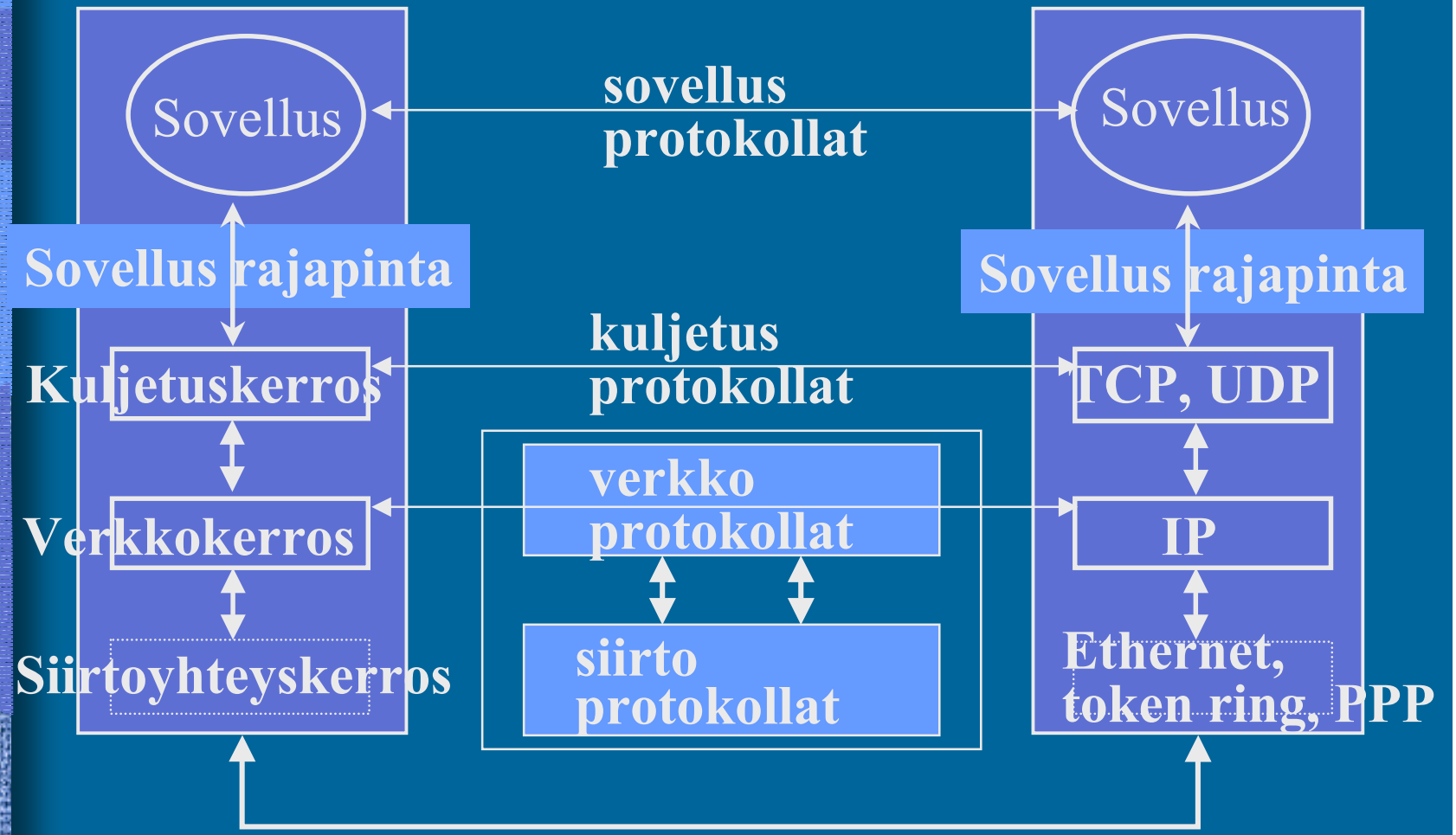


Figure 3-6. Layers of the I/O system and the main functions of each layer.

Tietoliikenne I, 2 ov

- Sopiva 2. vuoden opiskelijalle
- Tietokoneverkkojen peruspalvelut käyttäjälle ja sovelluksille
- Verkkojen tiedonsiirron perusvälineistö
- Verkkoarkkitehtuurin kerrosrakenne ja kunkin tason palvelut
- Jatkoa perustasolla ja syventävällä tasolla
  - Tietoliikenne II, 2 ov
  - Tietoliikennejärjestelmät, 3 ov

Tietoliikenne … TCP/IP -kerrosmalli



Rinnakkaisohjelmointi, 2 ov

- Sopiva: 2. vuoden opiskelijoille
- Samanaikaisuuden aiheuttamat ongelmat
  - järjestelmä kaatuu … miksi niin kävi?
- Samanaikaisuuden aiheuttamat vaatimukset systeemille
- Prosessien synkronointi eri tapauksissa
  - ”busy wait” vai prosessin vaihto?
- Prosessien kommunikointi eri tavoin
  - yhteinen muistialue? viestit?
  - verkon ylitse?
- Jatkoa syventävällä tasolla
  - Hajautetut järjestelmät, 3 ov

RIO: synkronointiongelman ratkaisu Test-and-Set -käskyllä

- TAS Ri, L (ttk-91:n laajennus)
- Kriittinen vaihe

Ri := mem[L]
if Ri==1 then
*{Ri := 0, mem[L] := Ri, jump *+2}*

```
LOOP:  TAS     R1, L      # L:  1 (vapaa)   0 (varattu)
       JUMP    LOOP
       ...
       kriittinen vaihe: yksi prosessi kerrallaan
       ...
       LOAD    R1,=1
       STORE   R1,L
```

- Toimiiko, jos tulee keskeytys pahassa kohtaa?
  - Mikä on “paha kohta”?

Ohjelmointikielten periaatteet, 4 ov

- Lähtötiedot: OLPM, TiKi, ohjelmointilabrat
- Sopiva: 3. vuoden opiskelijat
- Ohjelmointikielten määrittelyn välineistö
- Erilaiset ohjelmointiparadigmat esimerkkikielten avulla
  - proseduraaliset kielet
  - oliokielet
  - funktionaaliset kielet
  - logiikkaohjelmointikielet
- Jatkoa syvemmällä tasolla:
  - ??

C, Pascal

Smalltalk

Scheme, ML

Prolog

Ohjelmointikielten kääntäjät, 5 ov

- Lähtötiedot: OLPM, ohjelmointilabrat
- Sopiva: 3. vuoden opiskelijat
- Ohjelmointikielten kääntäjien tyypit
  - rekursiivisesti etenevä jäsentelijä
- Kääntäjän osat
  - selaaja
  - jäsentelijä
  - semantiikan analyysi
  - koodin generointi
- Jatkoa syvemmällä tasolla:
  - ??

lex

yacc

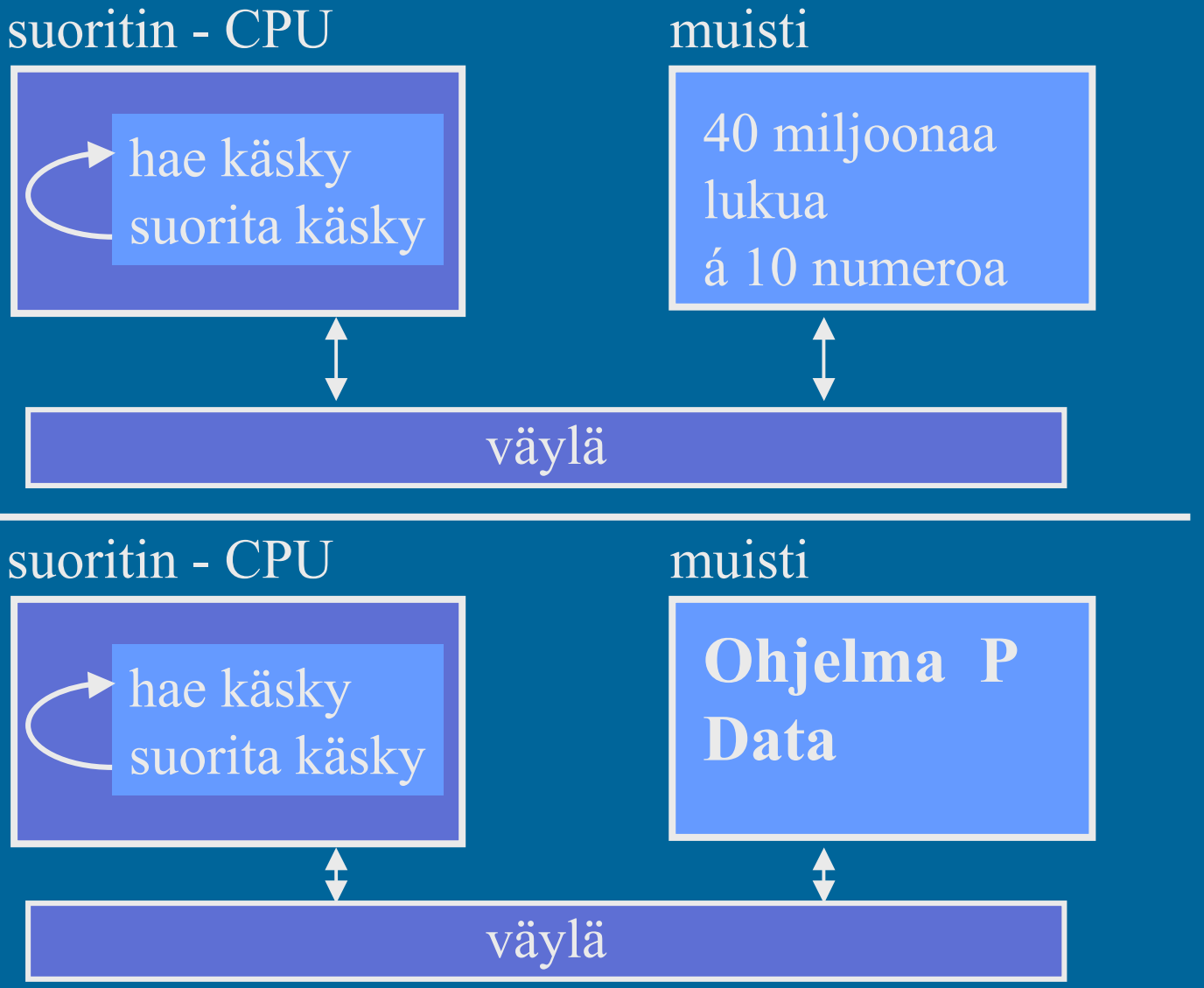
Spesifioinnin ja verifioinnin perusteet, 2 ov

- Lähtötiedot: hajautuksen ja samanaikaisuuden problematiikka
- Sopiva: 2. tai 3. vuoden opiskelijalle
- Mallinnetaan prosesseja siirtymäsysteemeillä
  - askel: konekäsky? metodi? tapahtuma? ohjelma?
- Automaattisen verifioinnin periaatteet
- Yksinkertaisia protokollien verifiointi
- Jatkoa syventävällä tasolla
  - Algoritmien oikeellisuus ja johtaminen, 3 ov
  - Automaattinen verifiointi, 3 ov

Ohjelmoinnin ja laskennan perusmallit (OLPM), 2 ov

- Lähtötiedot: matematiikkaa
  - appro tai diskr. mat., ... + tira?
- Sopiva: 1. vuoden (2. vuoden?) opiskelijalle, joka on opiskellut jo matematiikkaa
- Laskennalliset ongelmat, niiden luokittelu
- Äärelliset automaatit ja säännölliset kielet
- Kieliopit
- Turingin kone
- Jatkoa syventävällä tasolla
  - Laskennan teoria, 3 ov

Laskennan teorian perusta (1)



Laskennan teoriaa … (4)

Muistin sisältö ennen P:n suoritusta:

X = hyvin iso kokonaisluku (200M numeroa?)

Muistin sisältö P:n suorituksen jälkeen:

Y = joku toinen hyvin iso luku

P on kokonaislukuarvoinen funktio $P: N \rightarrow N$

Ohjelman P esitysmuoto muistissa: iso kokonaisluku $P \in N$

Laskennan teoriaa … (5)

- Mielivaltaisten ohjelmien ominaisuuksia voi päätellä kokonaislukujen ja niiden välisten funktioiden ominaisuuksista

- Todistettuja lauseita ohjelmien ominaisuuksista
  - pätevät kaikille tietokoneille
  - nyt ja tulevaisuudessa

Laskennan teoriasta ja algoritmianalyysistä todistettuja lauseita (3)

- Valitaanpa mikä tahansa aikaraja, niin aina on olemassa sellainen ongelma, että
  - (1) siihen on olemassa ratkaisu ja
  - (2) kaikki ongelman ratkaisevat ohjelmat vievät enemmän aikaa tai muistitilaa kuin ennalta annettu raja
- On olemassa sellaisia ongelmia, että niitä ei voi ratkaista millään tietokoneella
- On olemassa suuri joukko tunnettuja vaikeita ongelmia, joista ei vielä tiedetä, kuinka vaikeita ne oikeastaan ovat

$$\mathbf{P} \stackrel{?}{=} \mathbf{NP}$$

--
Luennon 12
ja
koko kurssin
loppu
--

http://lue.kurssikokeeseen.edu/ajoissa.html

http://www.retroweb.com/apollo_retrospective.html