

## Luento 12 Yhteenveto

Keskeiset asiat  
Mitä hyötyä tästä on?  
Mitä seuraavaksi?  
Kurssit?  
Asiat?

27.8.2002

Teemu Kerola, Copyright 2002

1

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

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

27.8.2002

Teemu Kerola, Copyright 2002

2

## Mitä hyötyä tästä on? <sup>(2)</sup>

- 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

27.8.2002

Teemu Kerola, Copyright 2002

3

## Keskeisiä asioita <sup>(10)</sup>

- 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

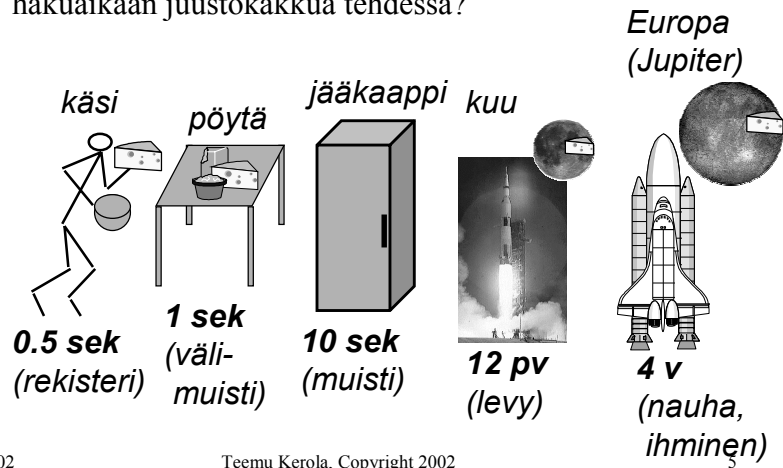
27.8.2002

Teemu Kerola, Copyright 2002

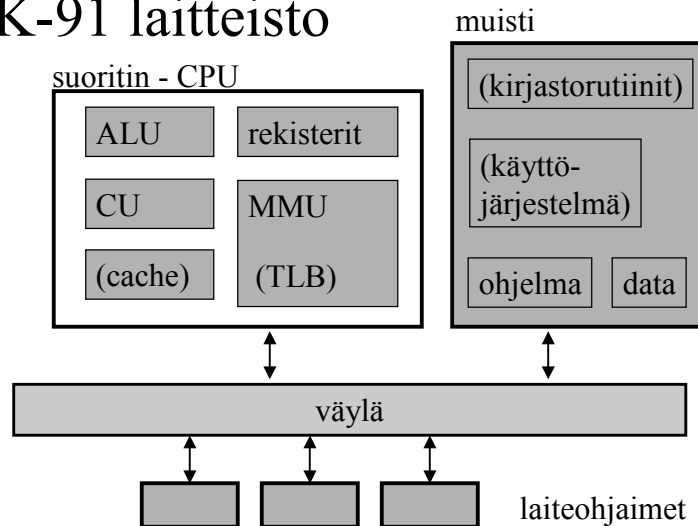
4

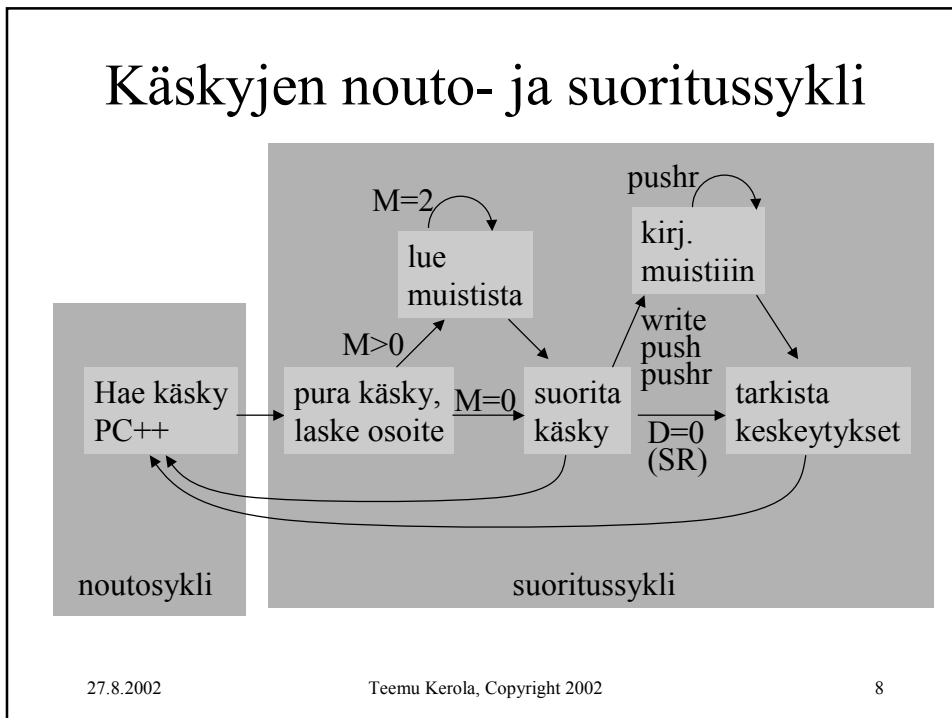
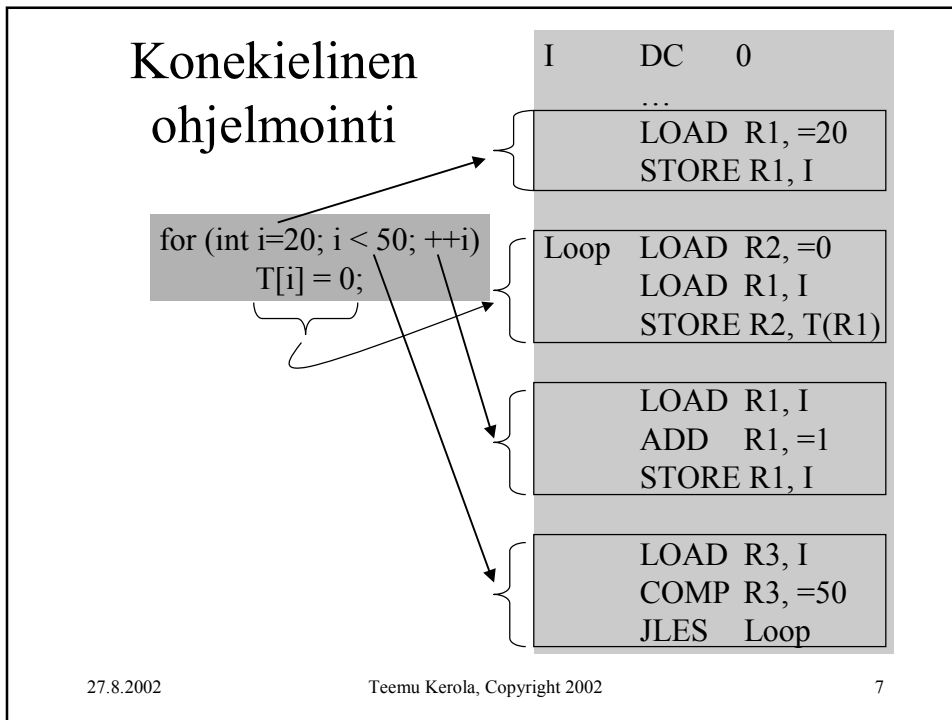
## Nopeuserot: Teemun juustokakku

Rekisterien, välimuistin, muistin, levymuistin ja magneettinauhan nopeudet suhteutettuna juuston haku-aikaan juustokakkuja tehdessä?



## Esimerkkikone: TTK-91 laitteisto





## Aktivoititietue

(activation record,  
activation frame)

```
int funcA (int x,y);
```

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

27.8.2002
Teemu Kerola, Copyright 2002

## IEEE 32-bit FP Standard

“+”

“15”

“0.1875” = “0.0011”

sign    exponent    mantissa or significand

• 23 bittiä mantissalle, siten että ...

1) Binääripiste (.) on heti ensimmäisen bitin jälkeen

2) Mantissa on normalisoitu: vasemmanpuolimmainen bitti on 1

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

mantissa eksponentti

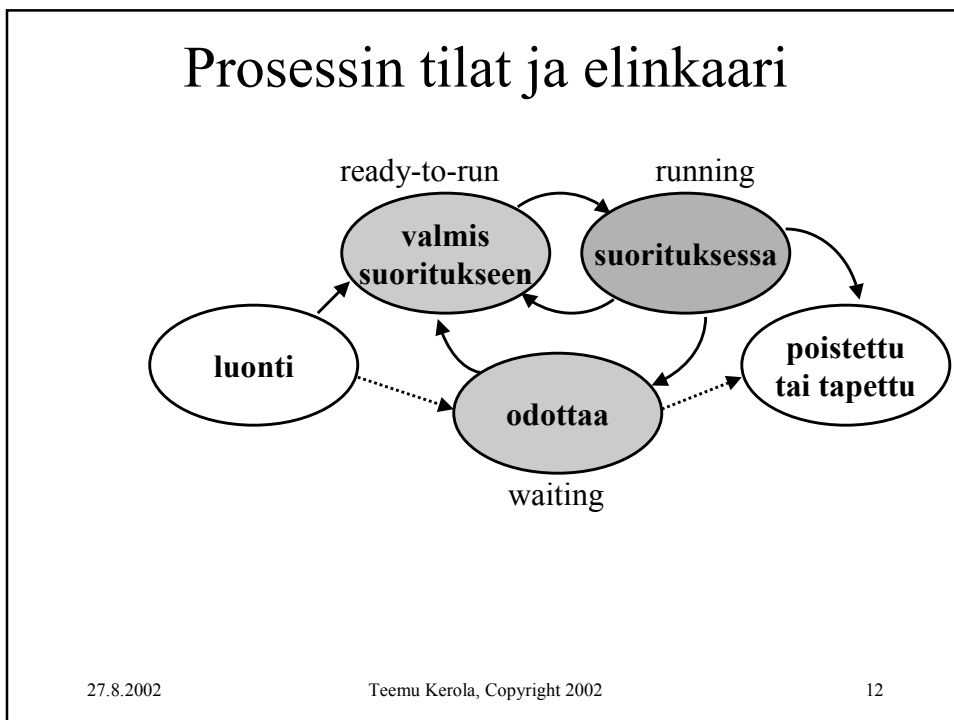
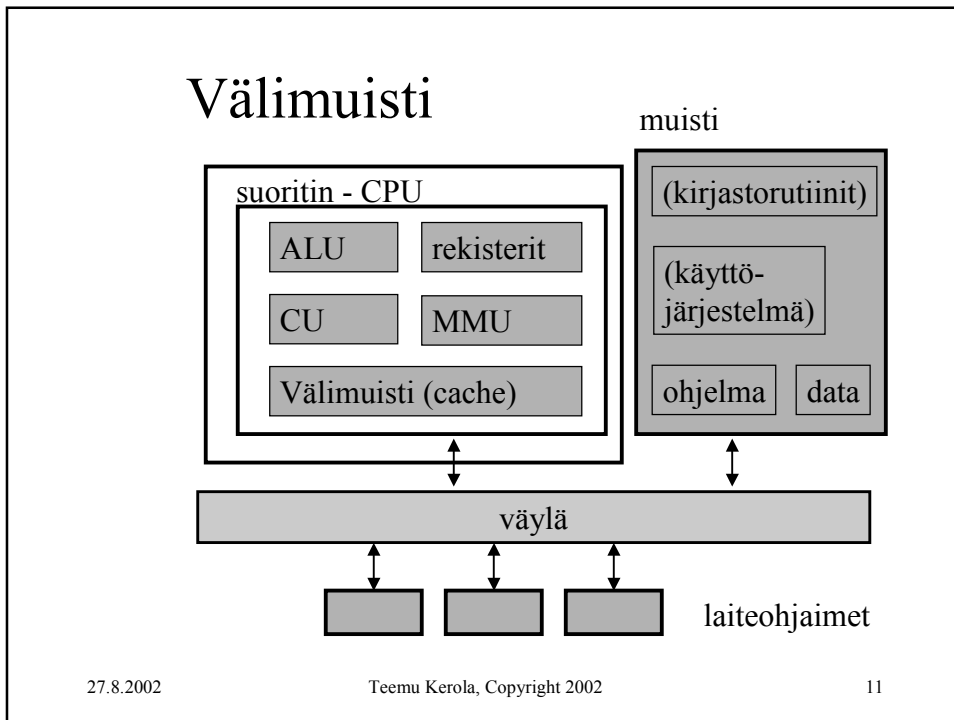
0.0011    “15”

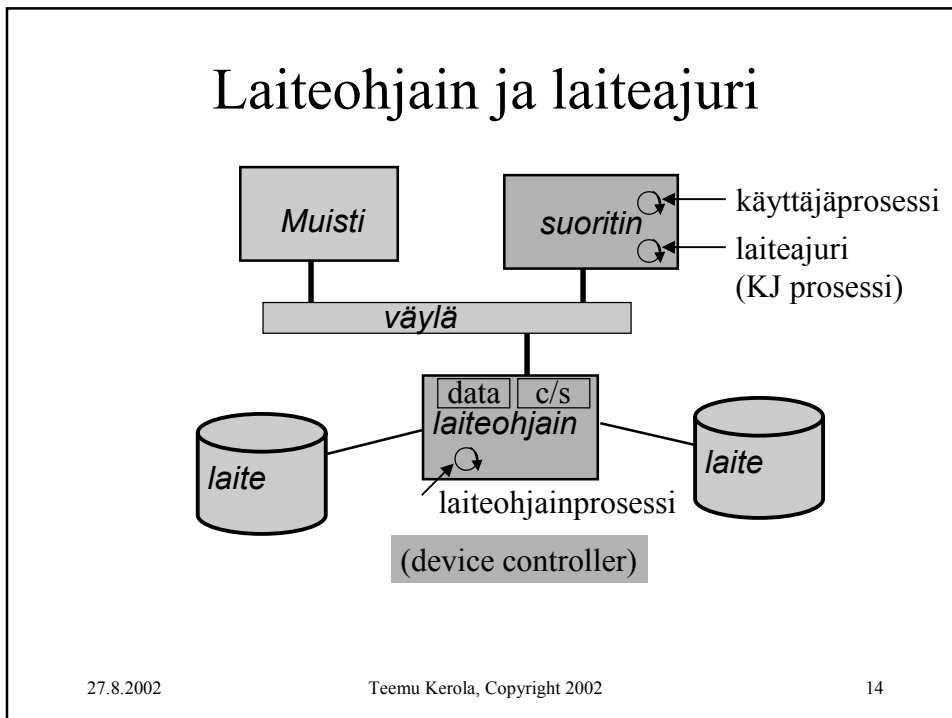
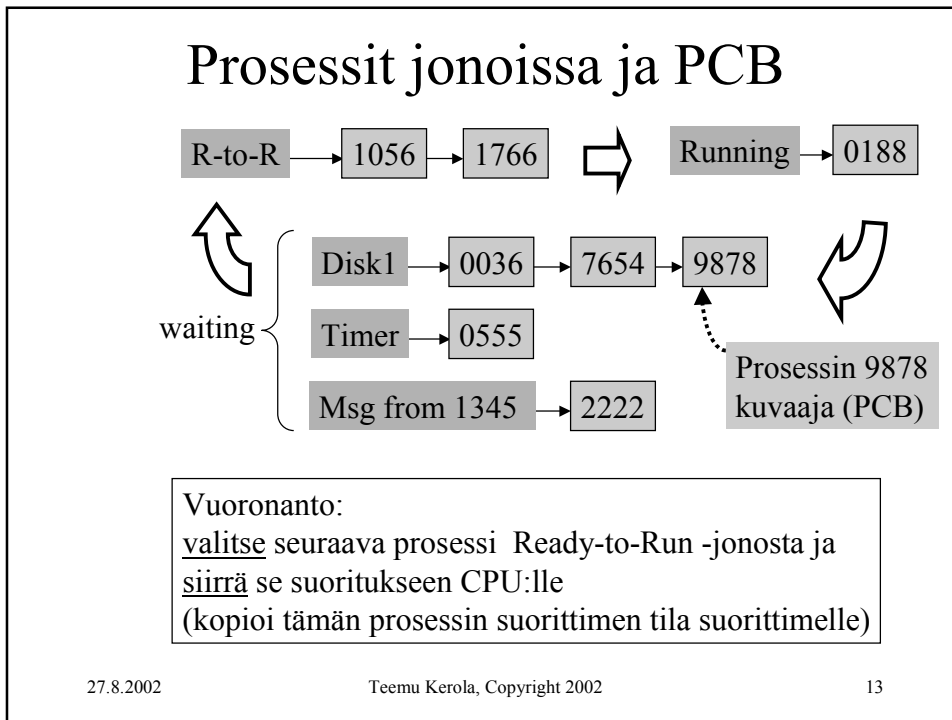
1.1000    “12”

1000    “12”

24 bitin mantissa!

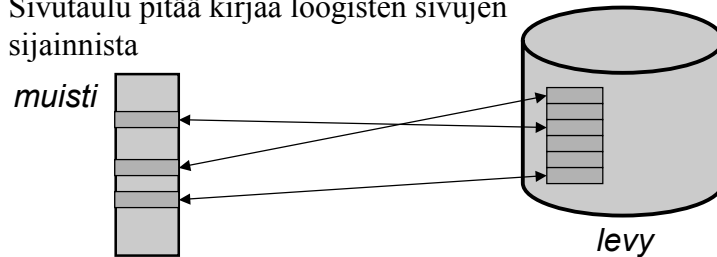
27.8.2002
Teemu Kerola, Copyright 2002
10





## Sivuttava virtuaalimuisti

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

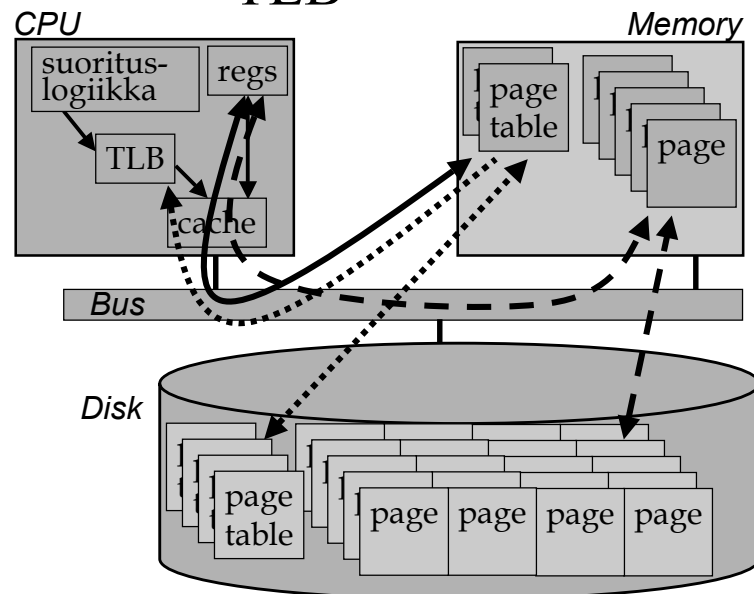


27.8.2002

Teemu Kerola, Copyright 2002

15

## TLB



27.8.2002

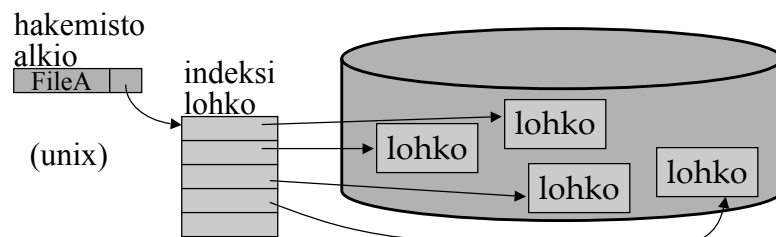
Teemu Kerola, Copyright 2002

16



## Levyjen käyttö

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

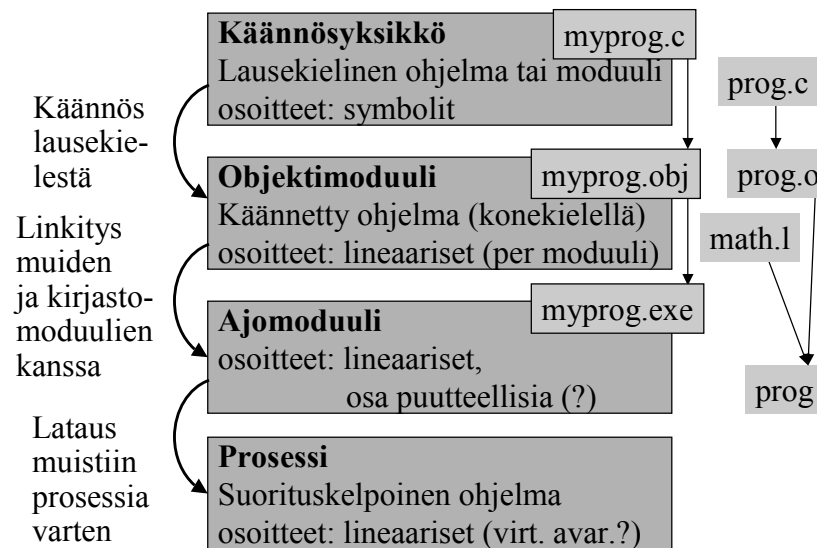


27.8.2002

Teemu Kerola, Copyright 2002

17

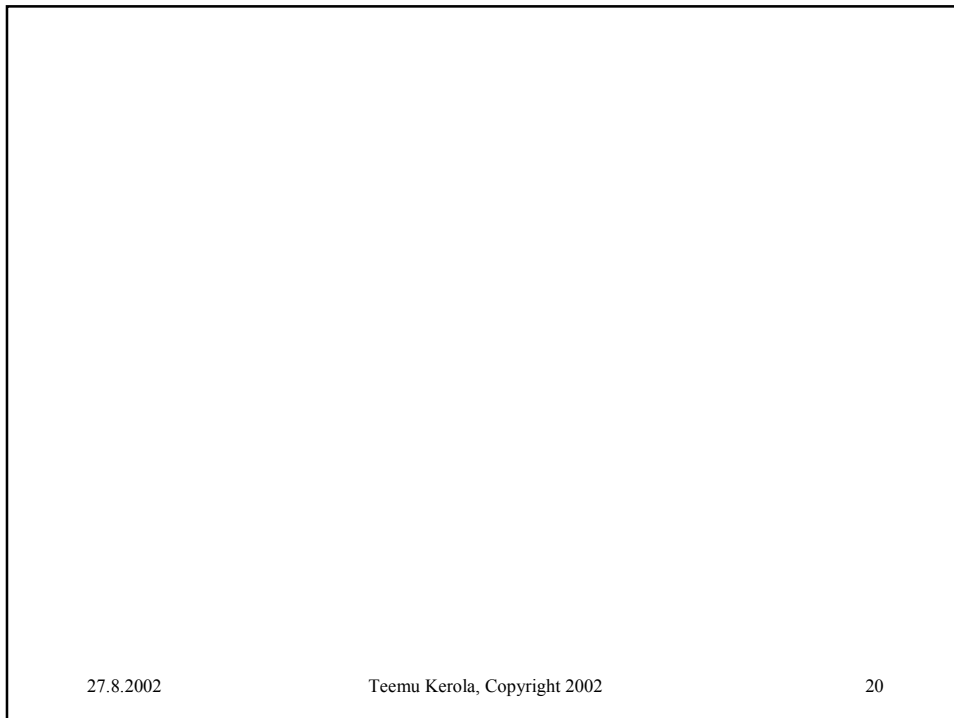
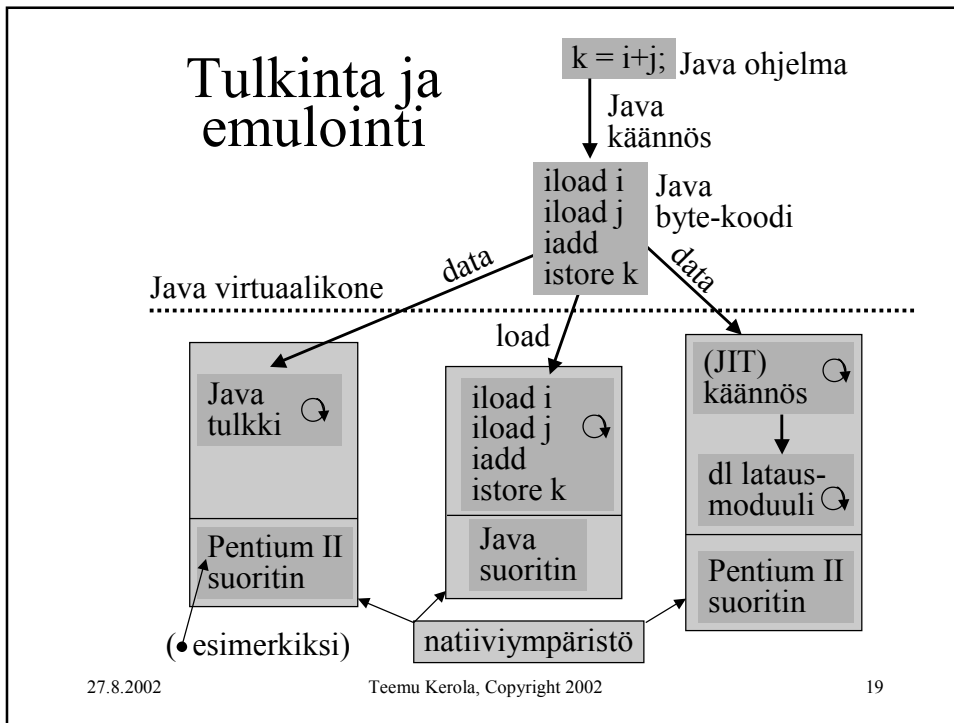
## Lausekielestä suoritukseen

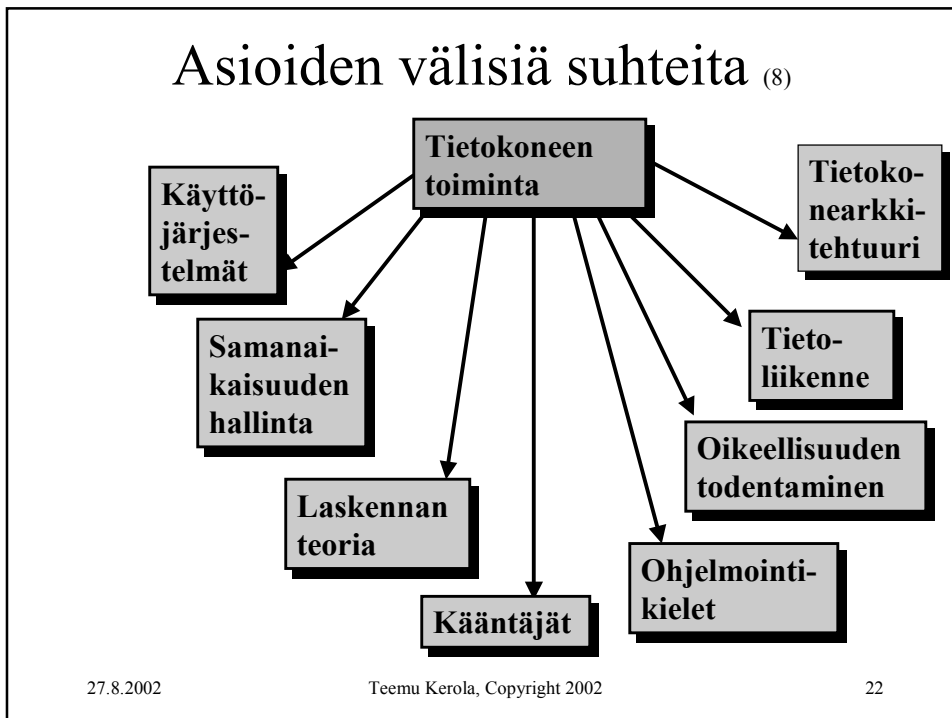
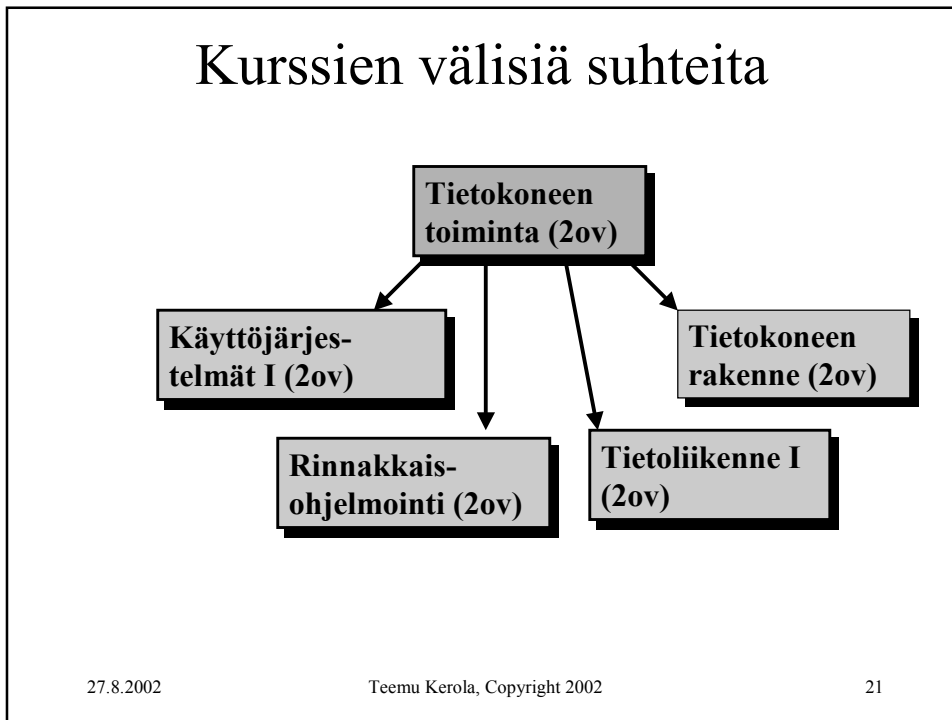


27.8.2002

Teemu Kerola, Copyright 2002

18





## 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

27.8.2002

Teemu Kerola, Copyright 2002

23

## TiKRra ....

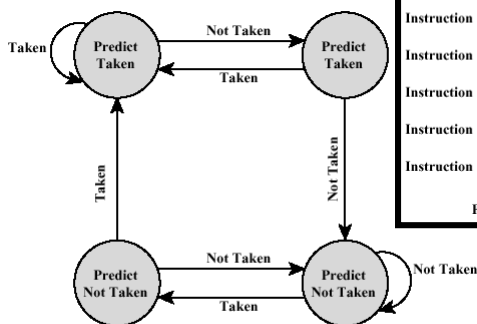


Figure 11.16 Branch Prediction State Diagram

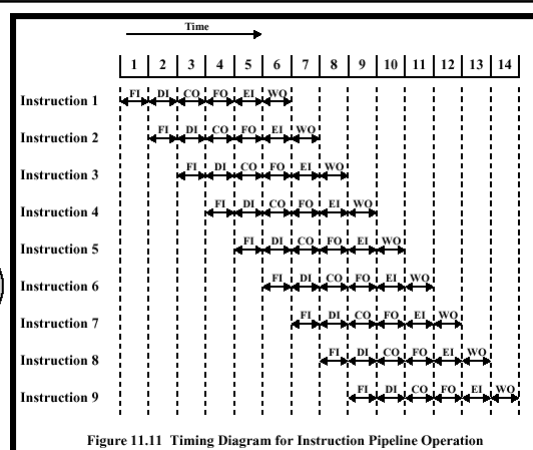


Figure 11.11 Timing Diagram for Instruction Pipeline Operation

[Stal99]

27.8.2002

Teemu Kerola, Copyright 2002

24

## 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

27.8.2002

Teemu Kerola, Copyright 2002

25

## KJ ...

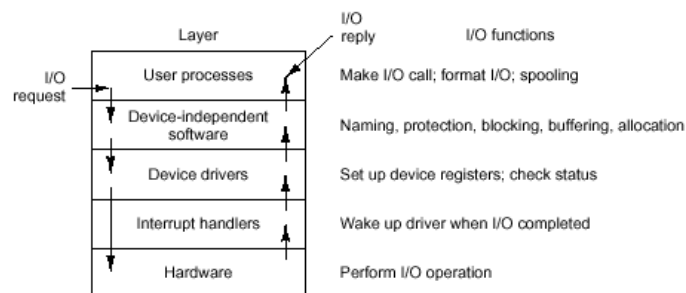


Figure 3-6. Layers of the I/O system and the main functions of each layer.

27.8.2002

Teemu Kerola, Copyright 2002

26

## Tietoliikenne I, 2 ov

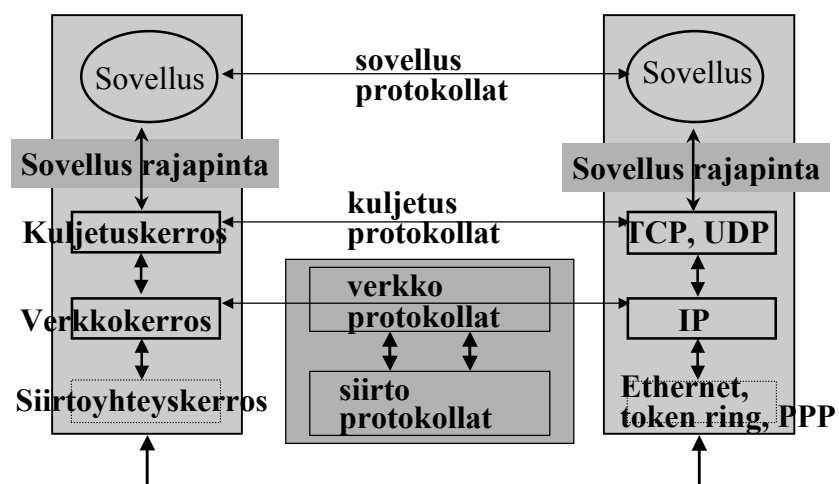
- Sopiva 2. vuoden opiskelijalle
- Tietokoneverkkojen peruspalvelut käyttäjälle ja sovelluksille
- Verkkojen tiedonsiirron perusvälineistö
- Verkkoarkkitehtuurin kerrosrakenne ja kunkin tason palvelut
- Jatkoa perustasolla ja syventävällä tasolla
  - Tietoliikenne II, 2 ov
  - Tietoliikennejärjestelmät,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

27.8.2002

Teemu Kerola, Copyright 2002

27

## Tietoliikenne ... TCP/IP -kerrosmalli



27.8.2002

Teemu Kerola, Copyright 2002

28

## 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

27.8.2002

Teemu Kerola, Copyright 2002

29

## RIO: synkronointiongelman ratkaisu Test-and-Set -käskyllä

- TAS  $R_i, L$   
(ttk-91:n  
laajennus)

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

- Kriittinen vaihe

```
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”?

27.8.2002

Teemu Kerola, Copyright 2002

30

## Ohjelmointikielten periaatteet, 4 ov

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

C, Pascal

Smalltalk

Scheme, ML

Prolog

27.8.2002

Teemu Kerola, Copyright 2002

31

## 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 syvemmillä tasolla:
  - ??

lex

yacc

27.8.2002

Teemu Kerola, Copyright 2002

32



## Spesifioinnin ja verifioinnin perusteet, 2 ov

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

27.8.2002

Teemu Kerola, Copyright 2002

33

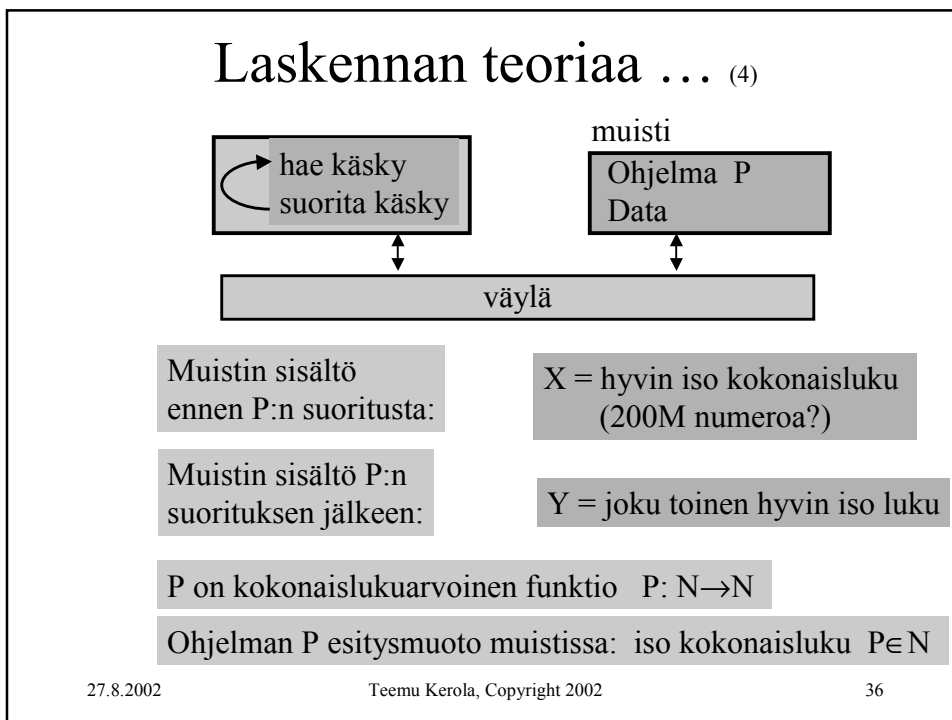
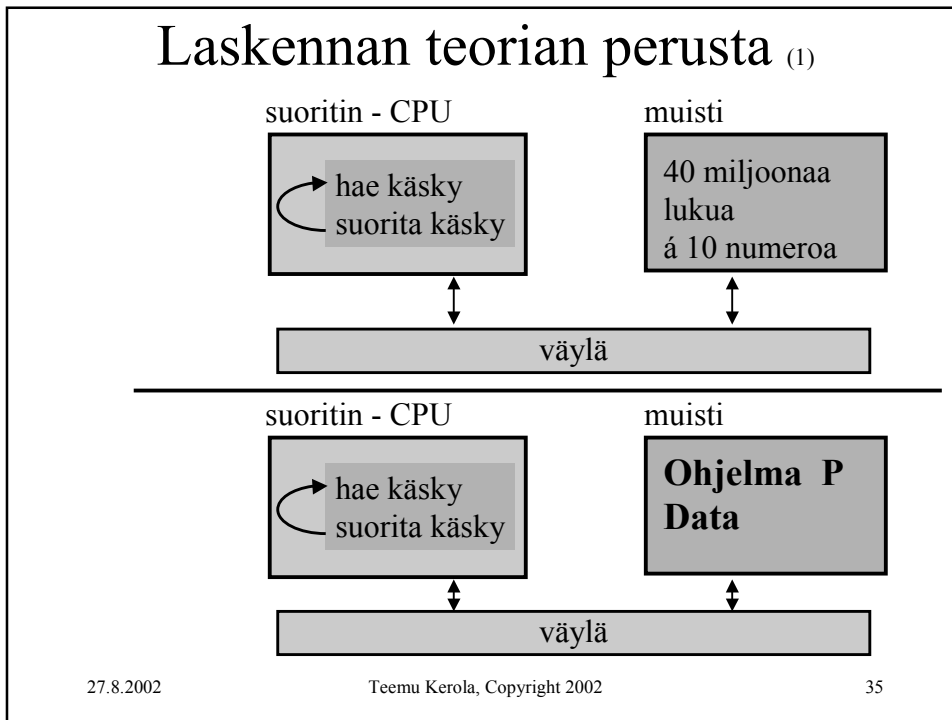
## Ohjelmoinnin ja laskennan perusmallit (OLPM), 2 ov

- Lähtötiedot: matematiikkaa
  - appro tai disk. 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

27.8.2002

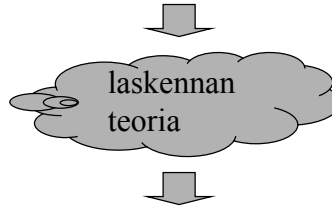
Teemu Kerola, Copyright 2002

34



## Laskennan teoriaa ... (5)

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



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

27.8.2002

Teemu Kerola, Copyright 2002

37

## 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

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

27.8.2002

Teemu Kerola, Copyright 2002

38

--  
Luennon 12  
ja  
koko kurssin  
loppu  
--

[http://www.retroweb.com/apollo\\_retrospective.html](http://www.retroweb.com/apollo_retrospective.html)



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



27.8.2002 Teemu Kerola, Copyright 2002 39