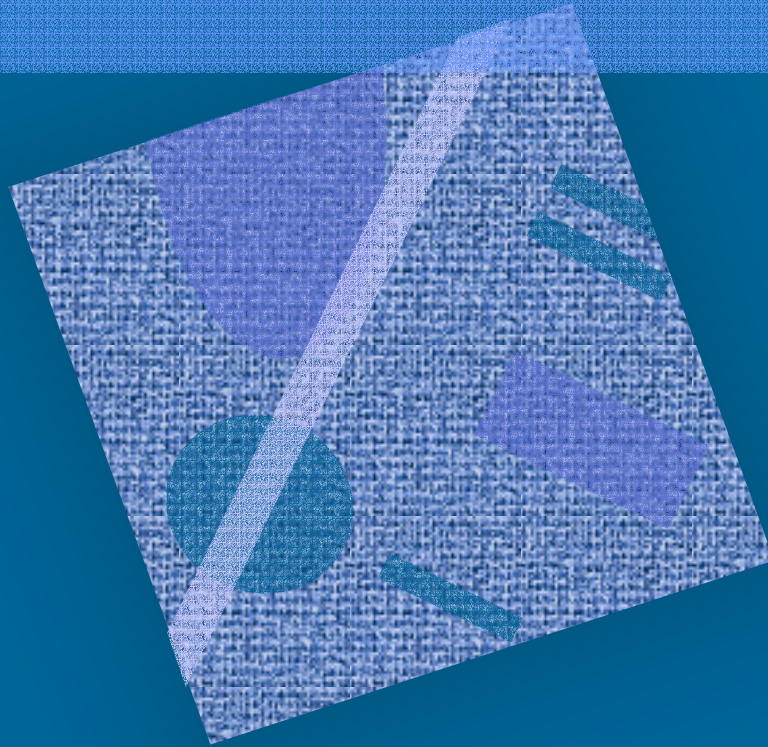


# Jakso 12

## Yhteenveto



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

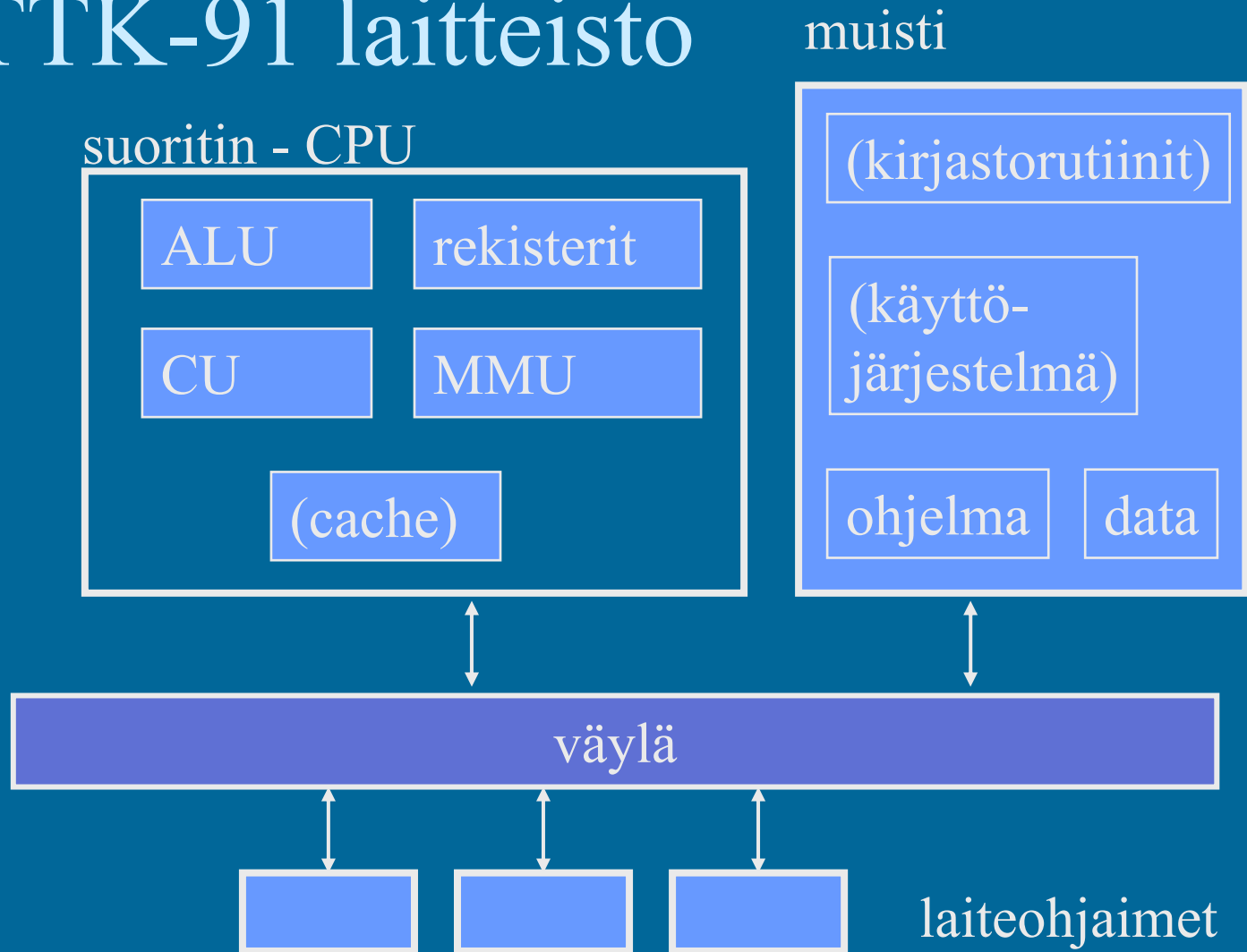
- Järjestelmä kokonaisuudessaan, nopeuserot
- Esimerkkikone ja sen käyttö
- Konekielinen ohjelmointi
- Suoritin, rekisterit, väylät, muisti
  - konekäskyjen suoritussykli, keskeytykset
- Aktivointitietuepino, aliohjelmien toteutus
- Tiedon esitysmuodot (ohjelma vs. laitteisto)
- Prosessi ja sen toteutus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

# Nopeuserot: juustokakku

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



# Esimerkkikone: TTK-91 laitteisto



# Konekielinen ohjelmointi

I DC 0

...

```
LOAD R1, =20  
STORE R1, I
```

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

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

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

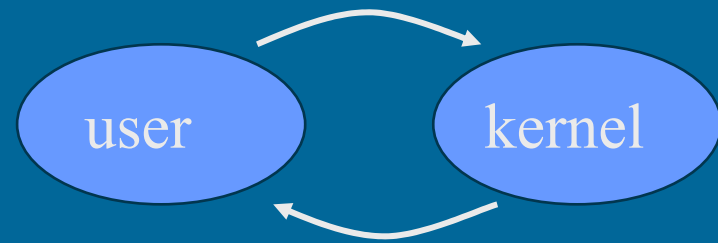
```
LOAD R3, I  
COMP R3, =50  
JLES Loop
```

# TTK-91 Nouto- ja suoritusyykli





# Suorittimen tilat



- Käyttäjätila
  - voi käyttää vain tavallisia käskyjä
  - voi viitata vain käyttäjän omaan muistiavaruuteen (MMU valvoo)
- Etuoikeutettu tila tai (KJ:n) ytimen tila
  - voi käyttää kaikkia konekäskyjä myös etuoikeutettuja (esim. `clear_cache`, `iret`)
  - voi viitata kaikkialla muistiin, myöös käyttöjärjestelmän ytimeen (kernel)
    - voi käyttää myös suoria muistiosoitteita (PA)

# Aktivointitietue

(activation record,  
activation frame)

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

- Aliohjelman toteutusmuoto (ttk-91)

- funktion paluuarvo (tai kaikki paluuarvot)

- kaikkien (sisäänmeno- tai ulostulo-) parametrien arvot

- paluuosoite

- kutsukohdan aktivointitietue **FP**

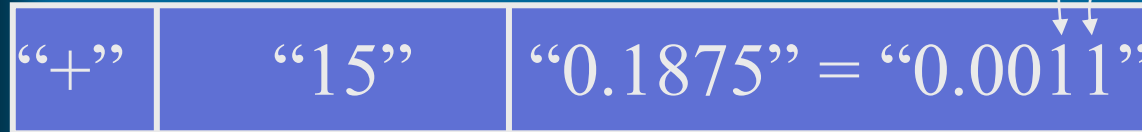
- kaikki paikalliset muuttujat ja tietorakenteet

- aliohjelman ajaksi talletettujen rekistereiden arvot



# Tiedon esitysmuodot

IEEE 32-bit FP Standard



$$\begin{array}{r} 1/8 = 0.1250 \\ 1/16 = 0.0625 \\ \hline 0.1875 \end{array}$$

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”

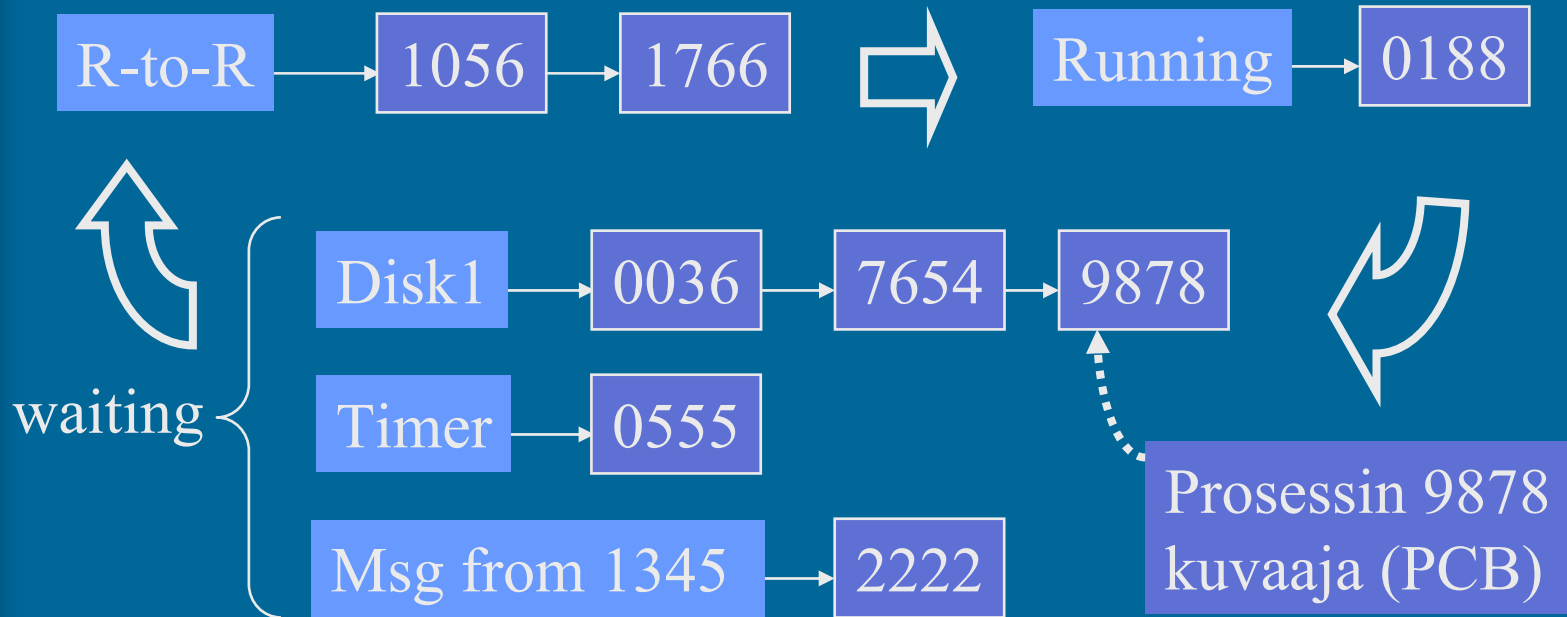
1000    “12”

24 bitin mantissa!

# Prosessin elinkaari



# Prosessit jonoissa ja PCB

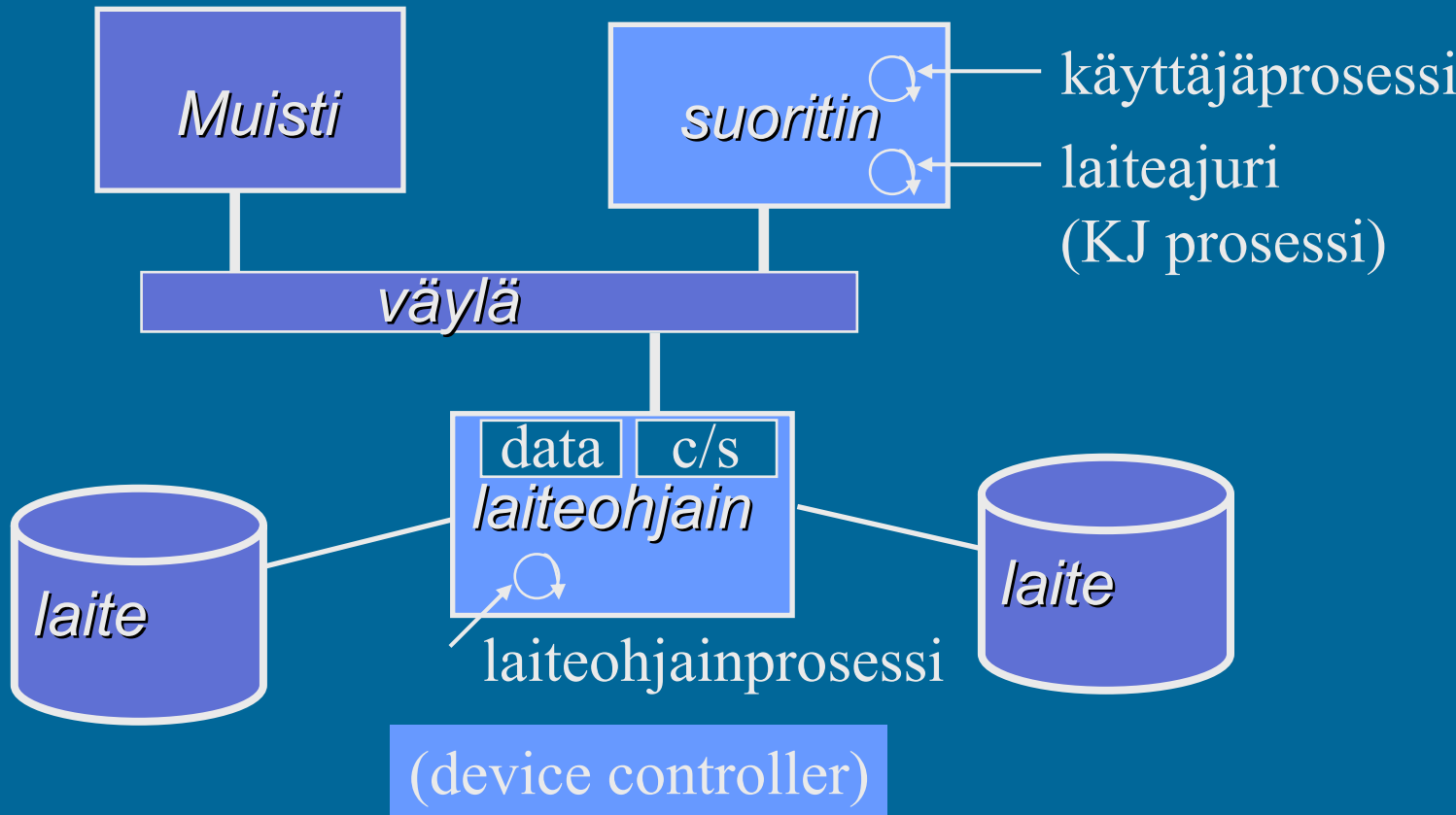


Vuoronanto:

valitse seuraava prosessi Ready-to-Run -jonosta ja  
siirrä se suoritukseen CPU:lle

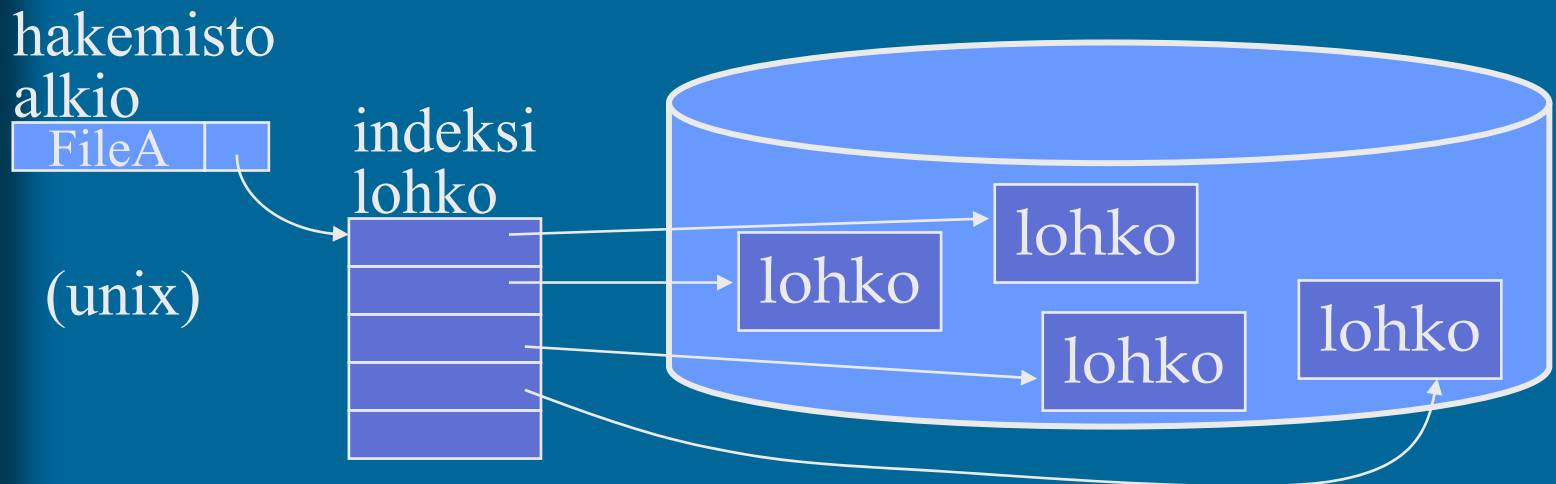
(kopioi tämän prosessin suorittimen tila suorittimelle)

# Laiteohjain ja laiteajuri

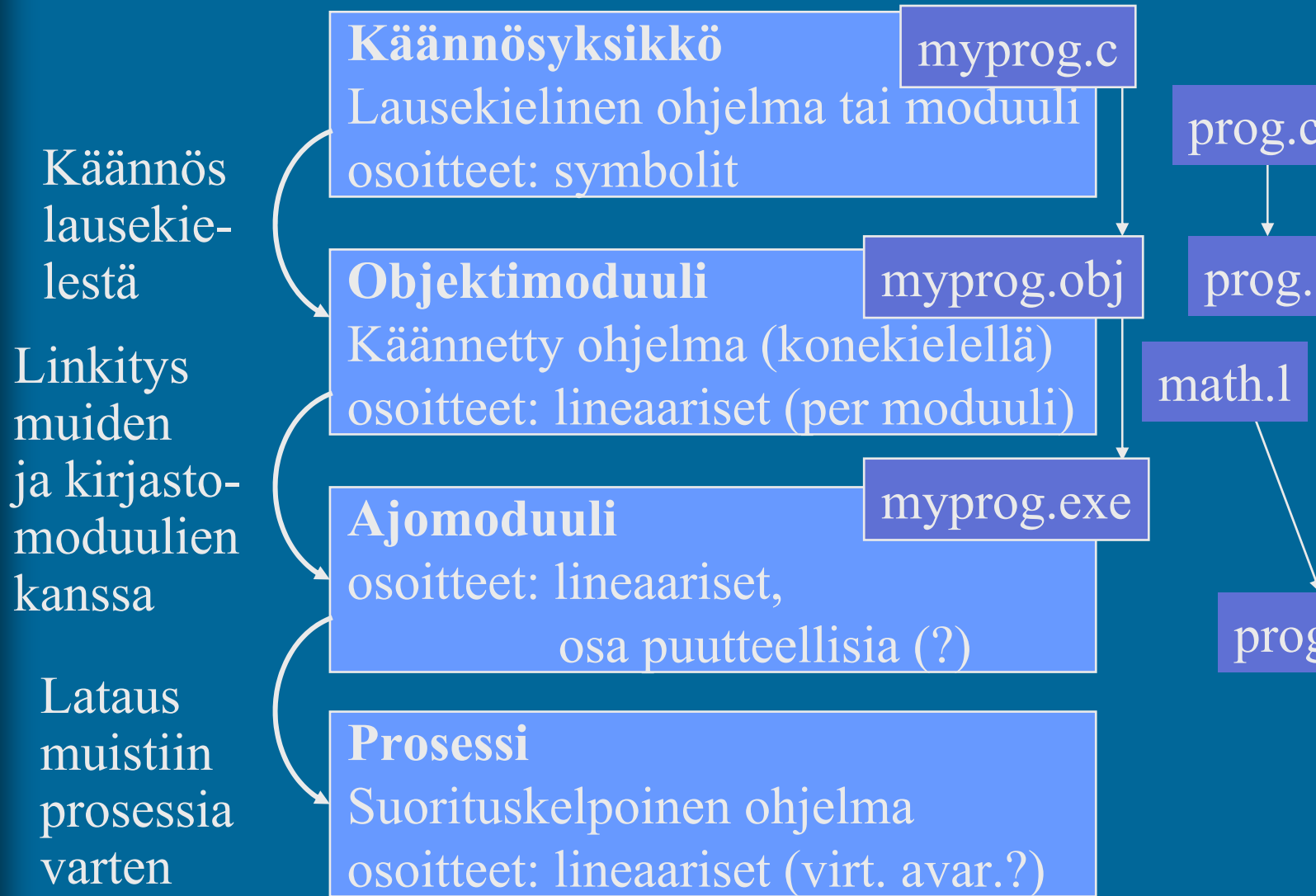


# Levyjen käyttö

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



# Lausekielestä suoritukseen





# Tulkinta ja emulointi

`k = i+j;` Java-ohjelma

Java-käännös

`iload i`  
`iload j`  
`iadd`  
`istore k`

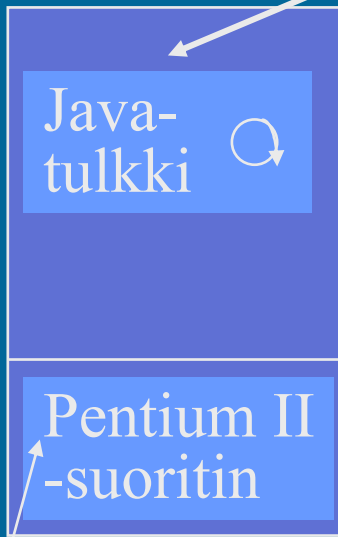
Java byte-koodi

Java-virtuaalikone

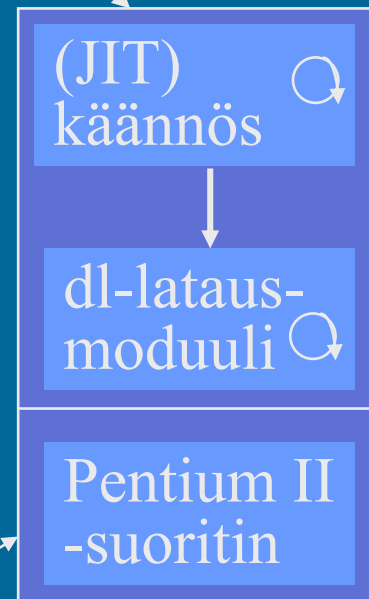
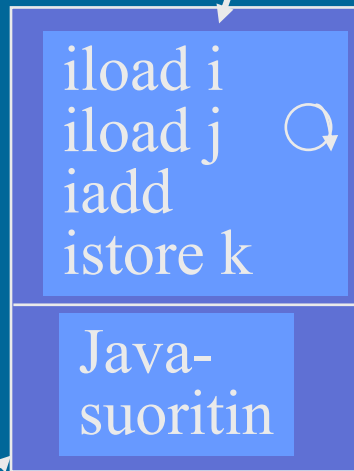
data

data

load



(• esimerkiksi)



natiiviympäristö

--  
Jakson 12  
ja  
koko kurssin  
loppu  
--



<http://cookpages.com/MuriBeachcomber/>

31/05/2004



Teemu Kerola, K2003