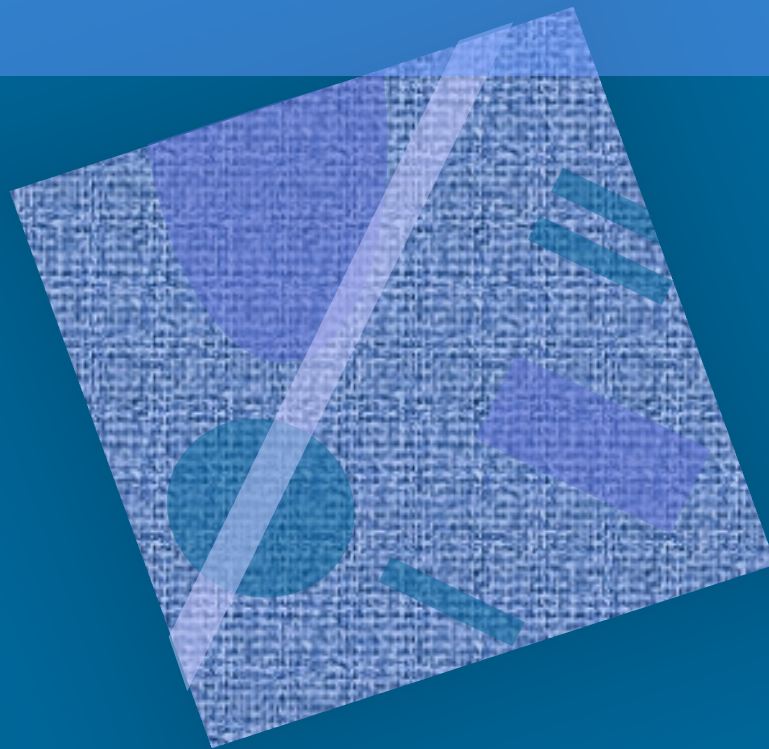


Luento 12

Yhteenveto



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

Tavoitteet

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

Mitä hyötyä tästä on?

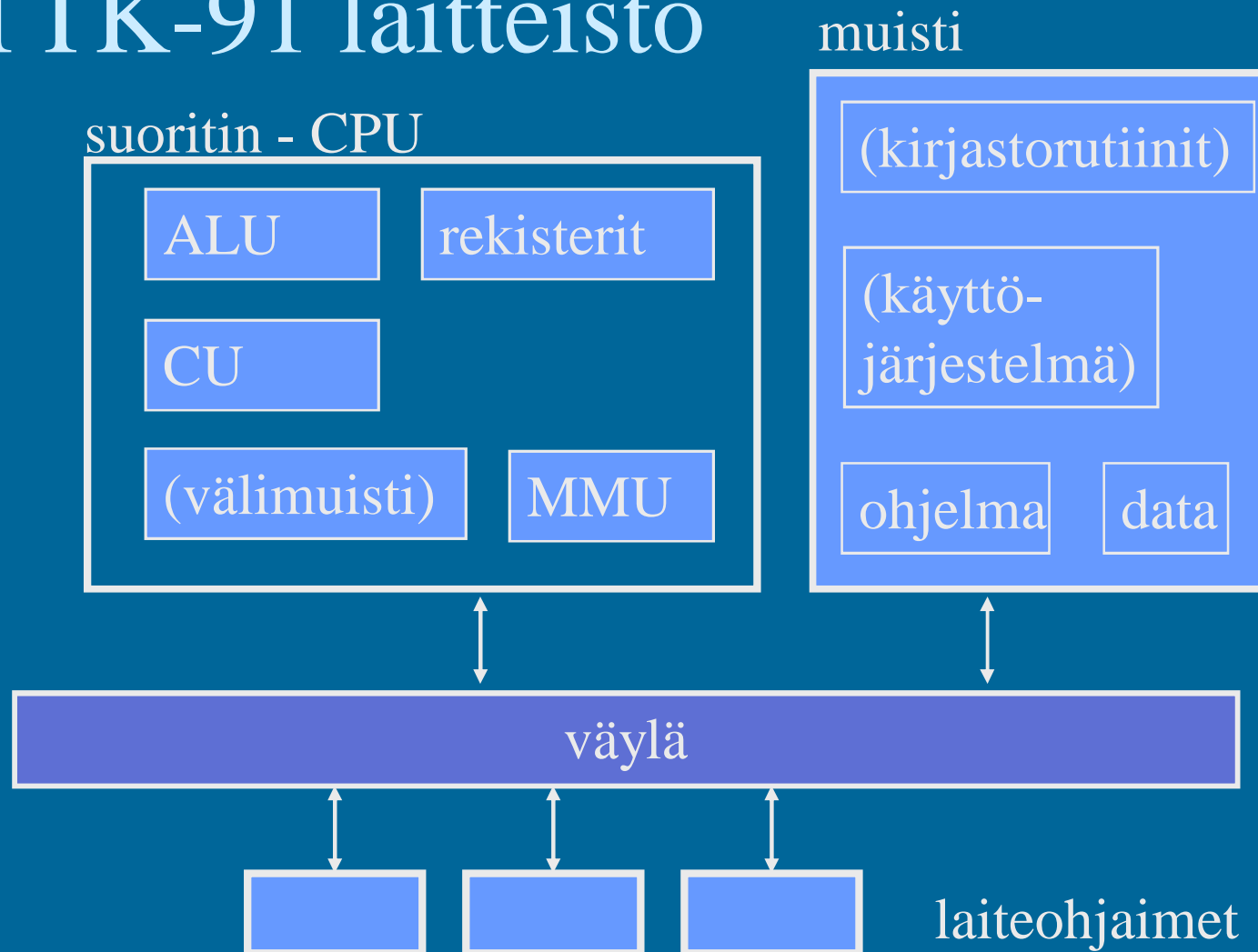
- Ohjelman suoritusnopeus perustuu suorittimen (CPU) suorittamiin konekäskyihin, ei pelkästään ohjelman korkean tason esitysmuotoon
 - Korkean tason kielen esitys on silti tärkeä!
- Ylemmän tason asioiden ymmärtäminen on helpompaa (mahdollista), kun ymmärtää alemman tason asiat

Keskeisiä asioita

- Järjestelmä kokonaisuudessaan, nopeuserot
 - esimerkkikone ja sen käyttö
- Ohjelman suoritus konekielen tasolla
 - suoritin, rekisterit, väylät, muisti
 - konekäskyjen suoritussykli, keskeytykset
 - aktivointitietuepino, aliohjelmien toteutus
- Tiedon esitysmuodot (ohjelma vs. laitteisto)
- I/O laitteet ja I/O:n toteutus
 - laiteajurit, laitekeskeytykset, levymuisti
- Käyttöjärjestelmän perustoiminnot
 - prosessi ja sen toteutus (PCB)
 - ohjelmien suoritus järjestelmässä
 - käänös, linkitys, lataus
 - tulkinta, emulointi, simulointi

Esimerkkejä keskeisistä asioista seuraavilla kalvoilla

Esimerkkikone: TTK-91 laitteisto



Nopeuserot: Teemun juustokakku

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



Konekielinen ohjelmointi

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

muuttujat, vakiot
taulukot (2D), tietueet
muistissa, rekisterissä?

valinta, loopit
aliohjelmat, SVC:t
parametrit,
paikalliset muuttujat

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

Aktivointitietue (Aktivointitietuepino)

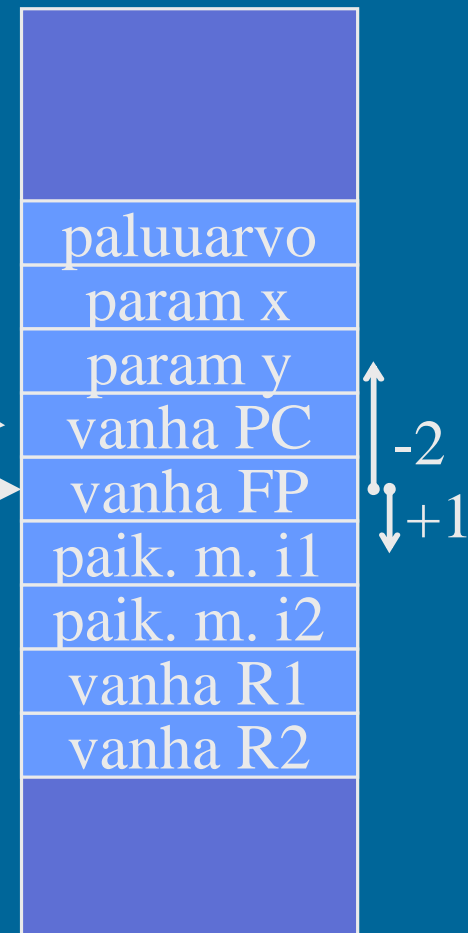
(activation record,
activation frame)

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

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

Parametrien
tyyppi?

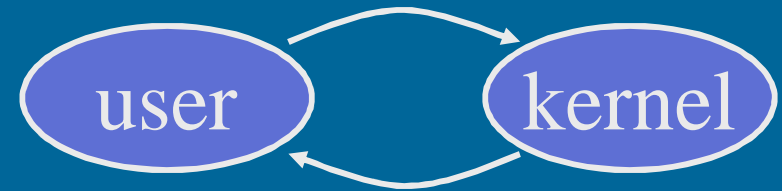
arvo-, viite- ja
nimiparametrit



Käskeyjen nouto- ja suoritusssykli



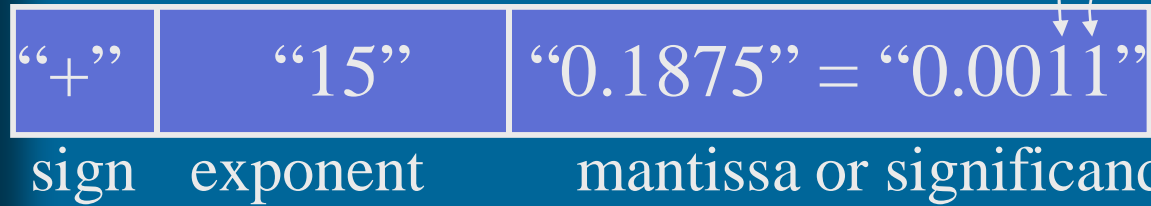
Suorittimen suoritusilat



- Käyttäjätila (user mode, normal mode)
 - voi käyttää vain tavallisia käskyjä
 - voi viitata vain käyttäjän omaan muistiavaruuteen (MMU valvoo)
- Etuoikeutettu tila tai (KJ:n) ytimen tila (kernel mode, privileged mode)
 - voi käyttää kaikkia konekäskyjä, myös etuoikeutettuja (esim. clear_cache, iret)
 - voi viitata kaikkialle muistiin, myös käyttöjärjestelmän ytimeen (kernel)
 - voi käyttää (myös) suoria muistiosoitteita (PA, physical address)

Miten ja milloin tila vaihtuu?

Tiedon esitysmuodot



$$\frac{1}{8} = 0.1250$$

$$\frac{1}{16} = \frac{0.0625}{0.1875}$$



kokonaisluvut
 liukuluvut
 merkit
 merkkijonot
 2 kuvat
 vääntet
 3 ei-standardoitu tieto?
 n suorittimen ymmärtämä tieto?

ten että ...

mantissa eksponentti

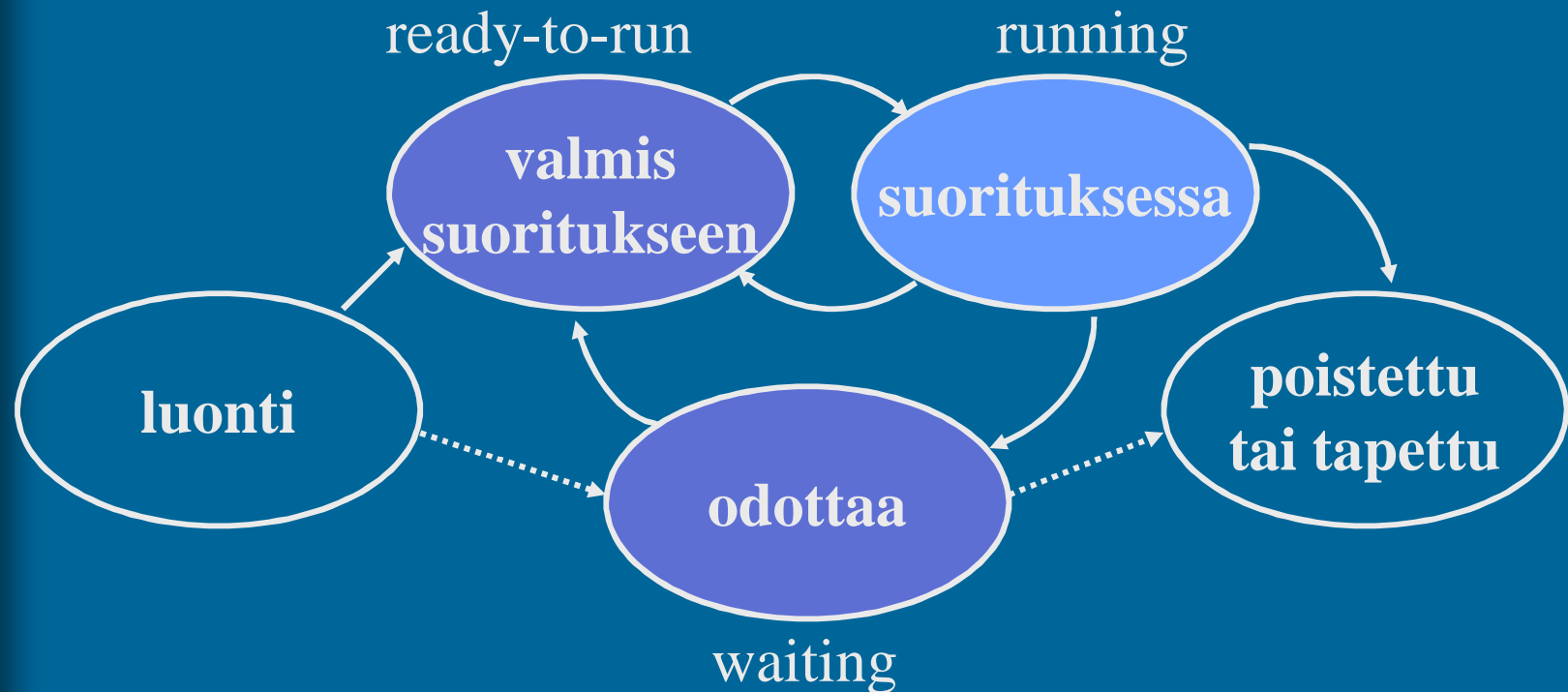
0.0011 “15”

1.1000 “12”

1000 “12”

24 bitin mantissa!

Prosessi, prosessin tilat ja elinkaari

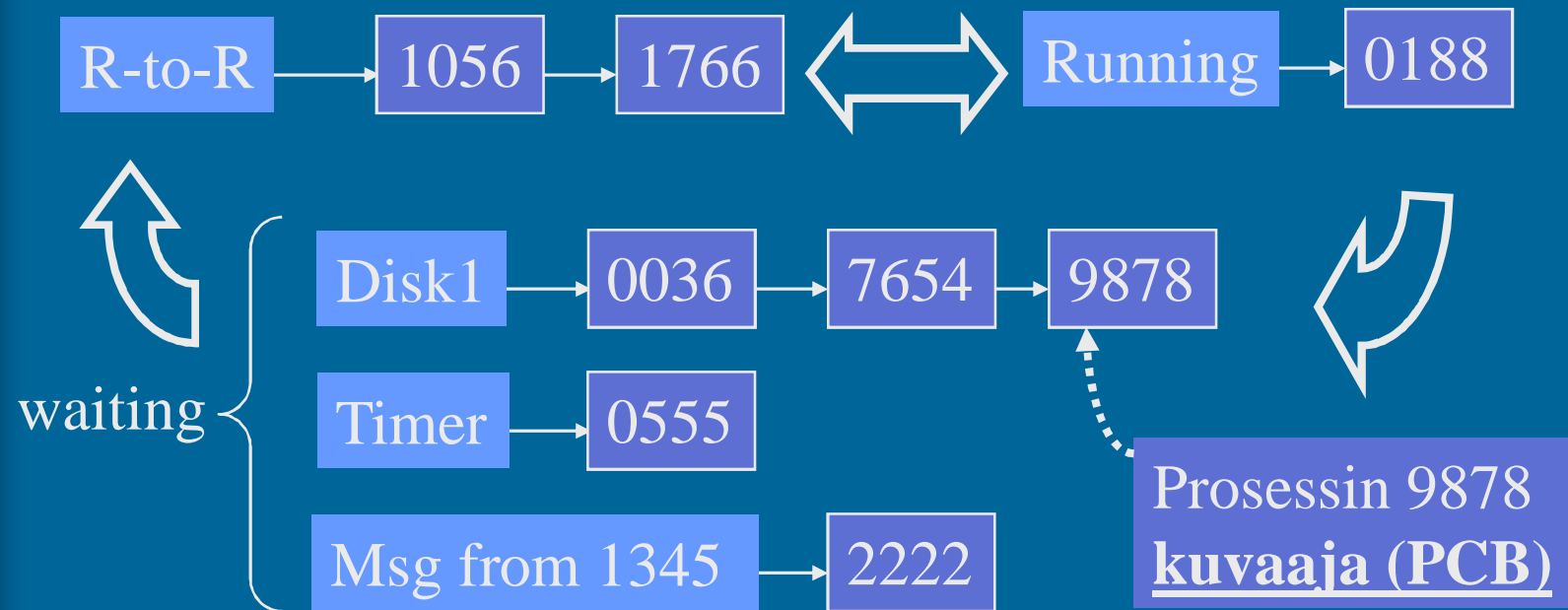


Milloin tila vaihtuu?

Mitä silloin tapahtuu (konekäsky tasolla)?

Kuka tai mikä aiheuttaa tilan vaihdon?

Prosessit jonoissa ja PCB

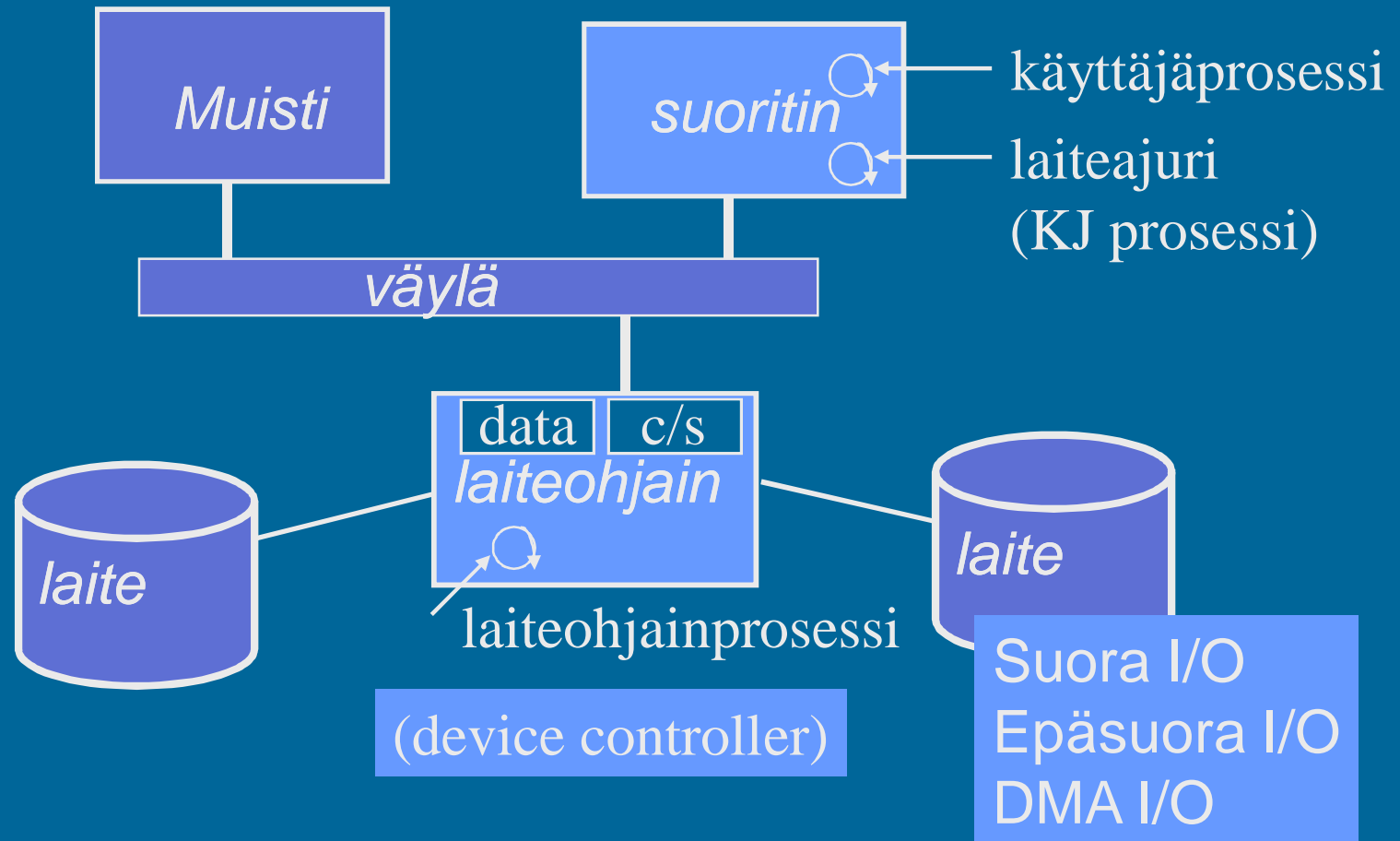


Vuoronanto:

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

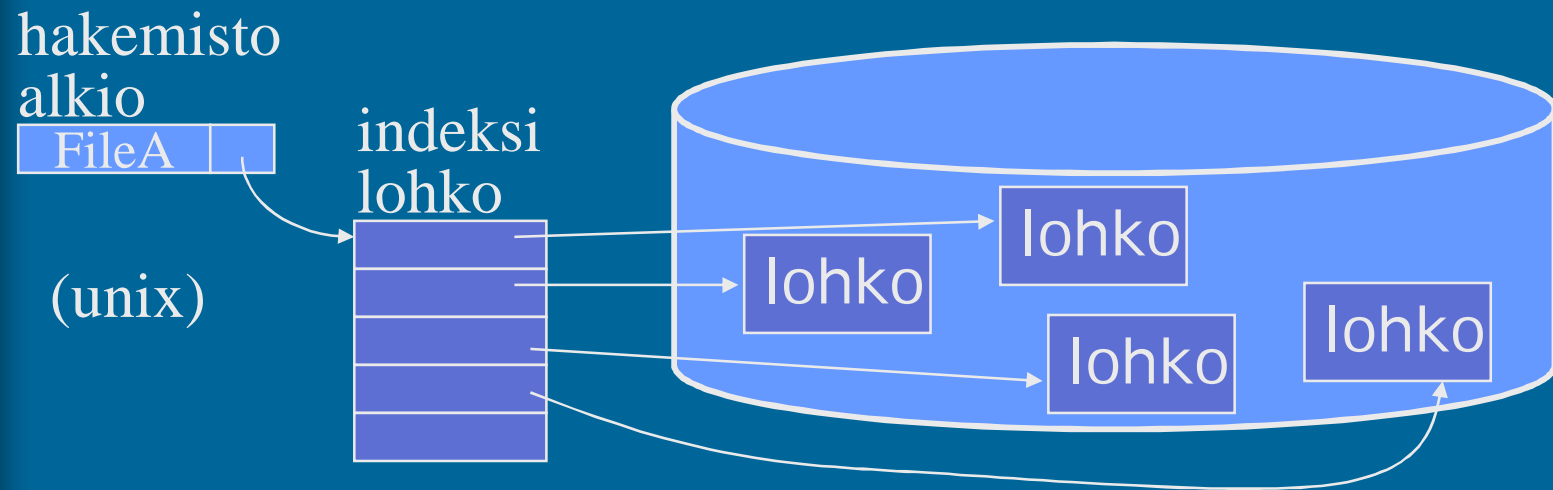
(kopioi tämän prosessin suorittimen tila suorittimelle)

I/O:n toteutus, laiteohjain ja laiteajuri



Levyjen käyttö

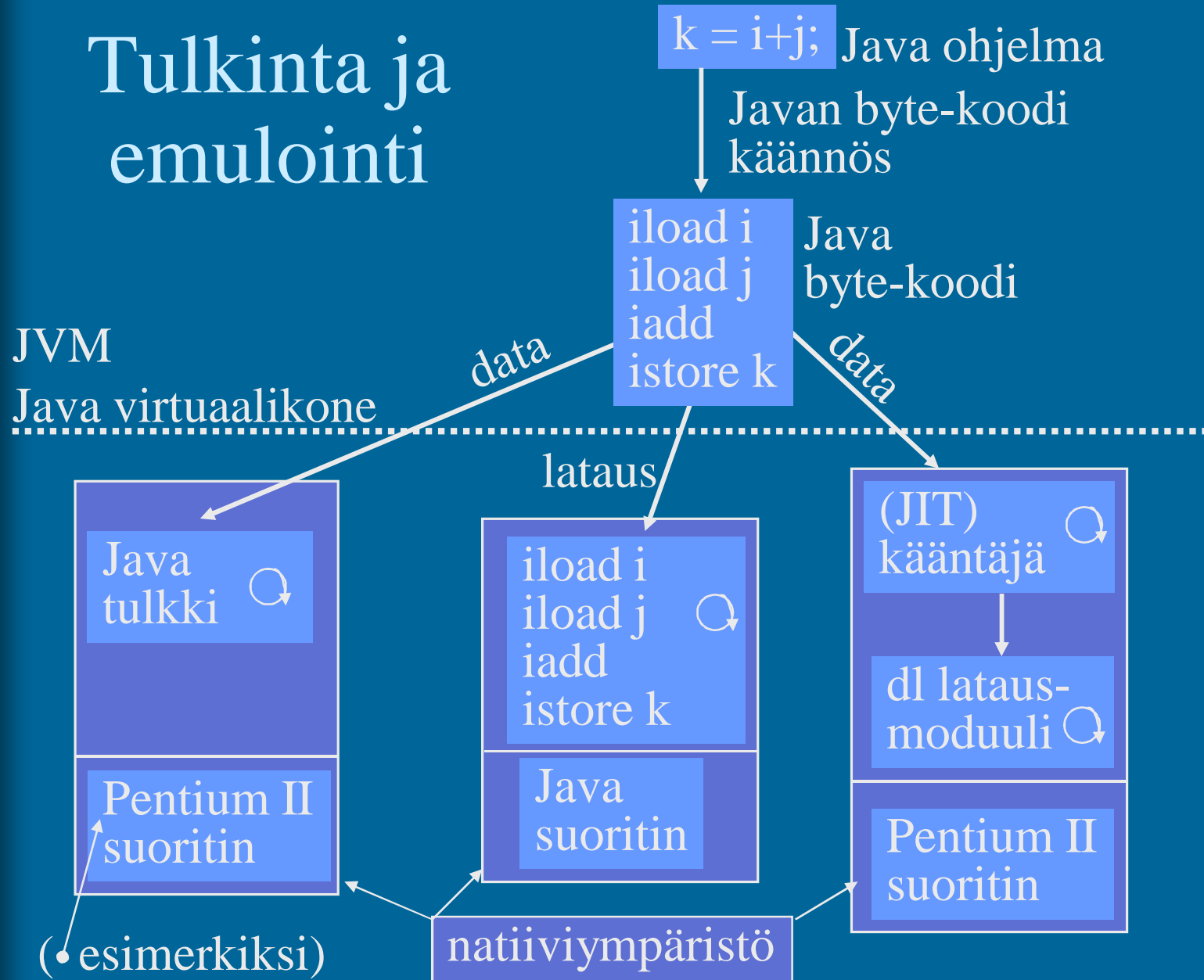
- Tiedosto koostuu useista lohkoista
 - lohko per sektori (tai usea sektori)
- Levyn hakemistossa on tieto kunkin tiedoston käyttämistä lohkoista
 - luetaan lohkot annetussa järjestyksessä



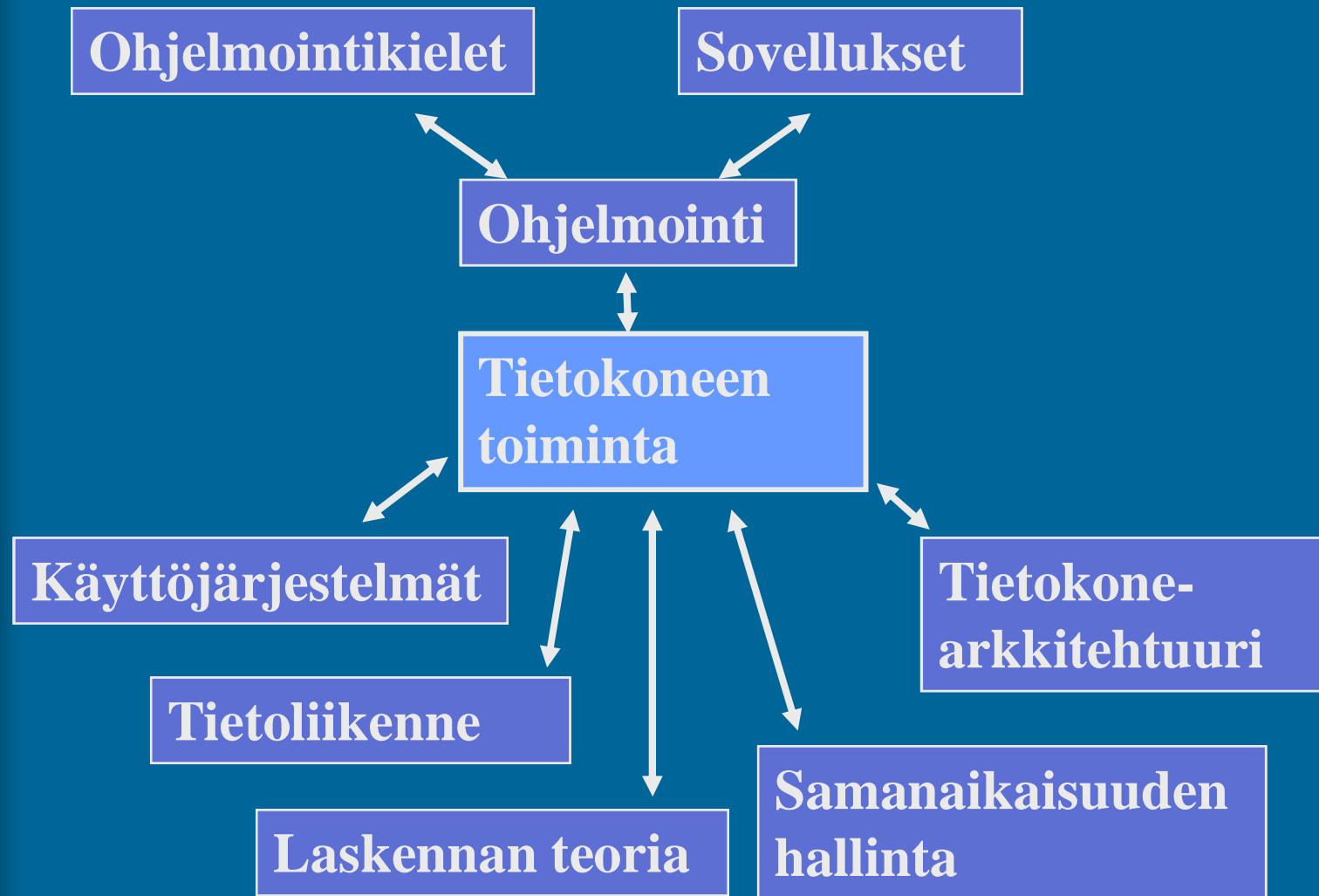
Lausekielestä suoritukseen



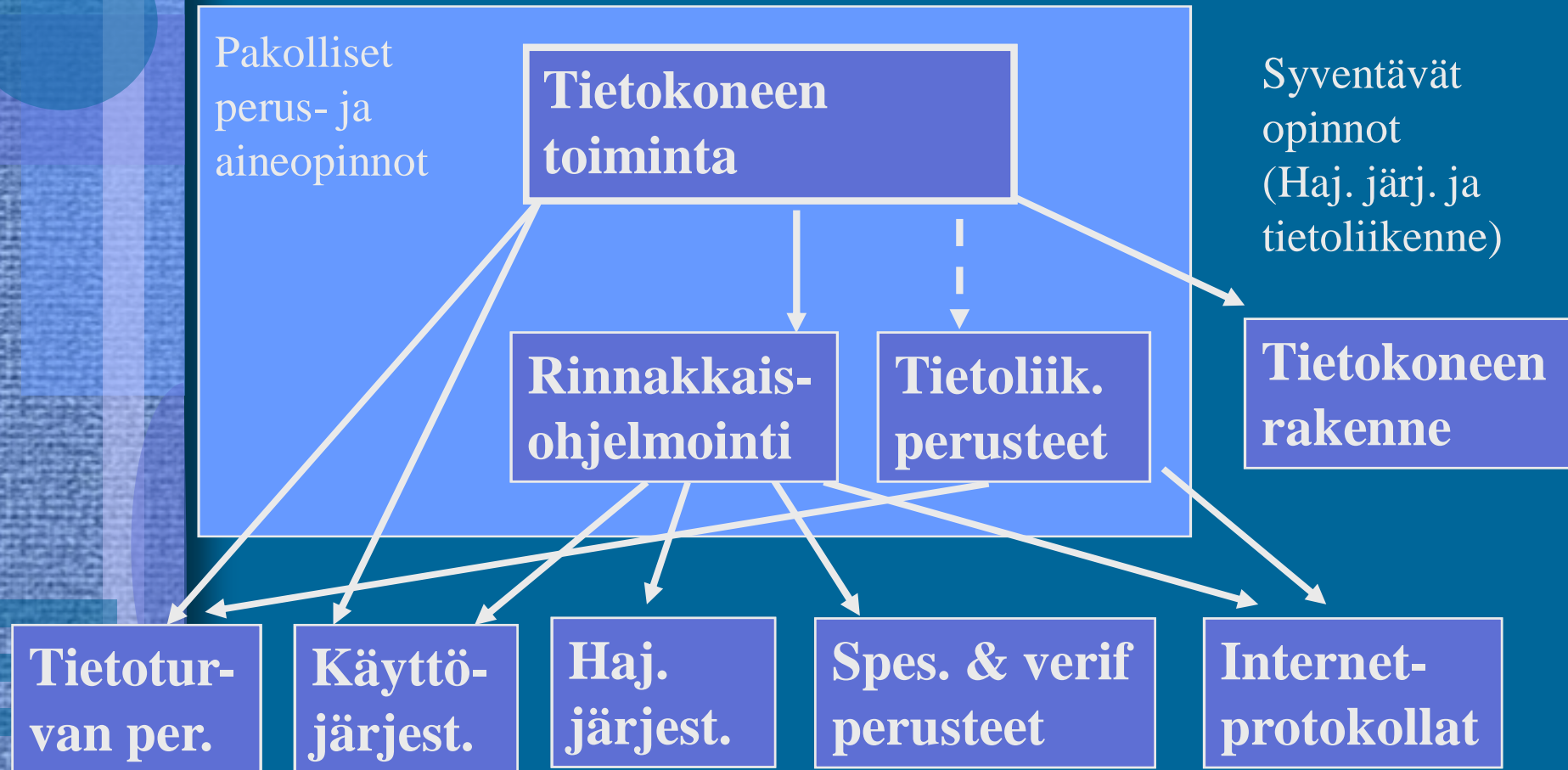
Tulkinta ja emulointi



Aihepiirien välisiä riippuvuuksia



Kurssien välisiä riippuvuuksia



Tietokoneen rakenne, 4 op

- Yksi taso alaspäin TITOSTa
 - LuK ja FM opintojen valinnainen kurssi
- Sopiva 2. vuoden opiskelijalle, heti TITOn jälkeen
- Useissa yliopistoissa yhdistetty TITOon
- ”Miten kellopulssi saa suorittimen suorittamaan konekäskyjä?”
- ”Miten suorittimen aritmetiikka on toteutettu?”
- Usea käsky on todellisuudessa suorituksessa samanaikaisesti (monellakin tavalla!)
 - Miten tämä toteutetaan, mitä ongelmia siitä seuraa ja miten noita ongelmia ratkotaan?

Käyttöjärjestelmät, 4 op

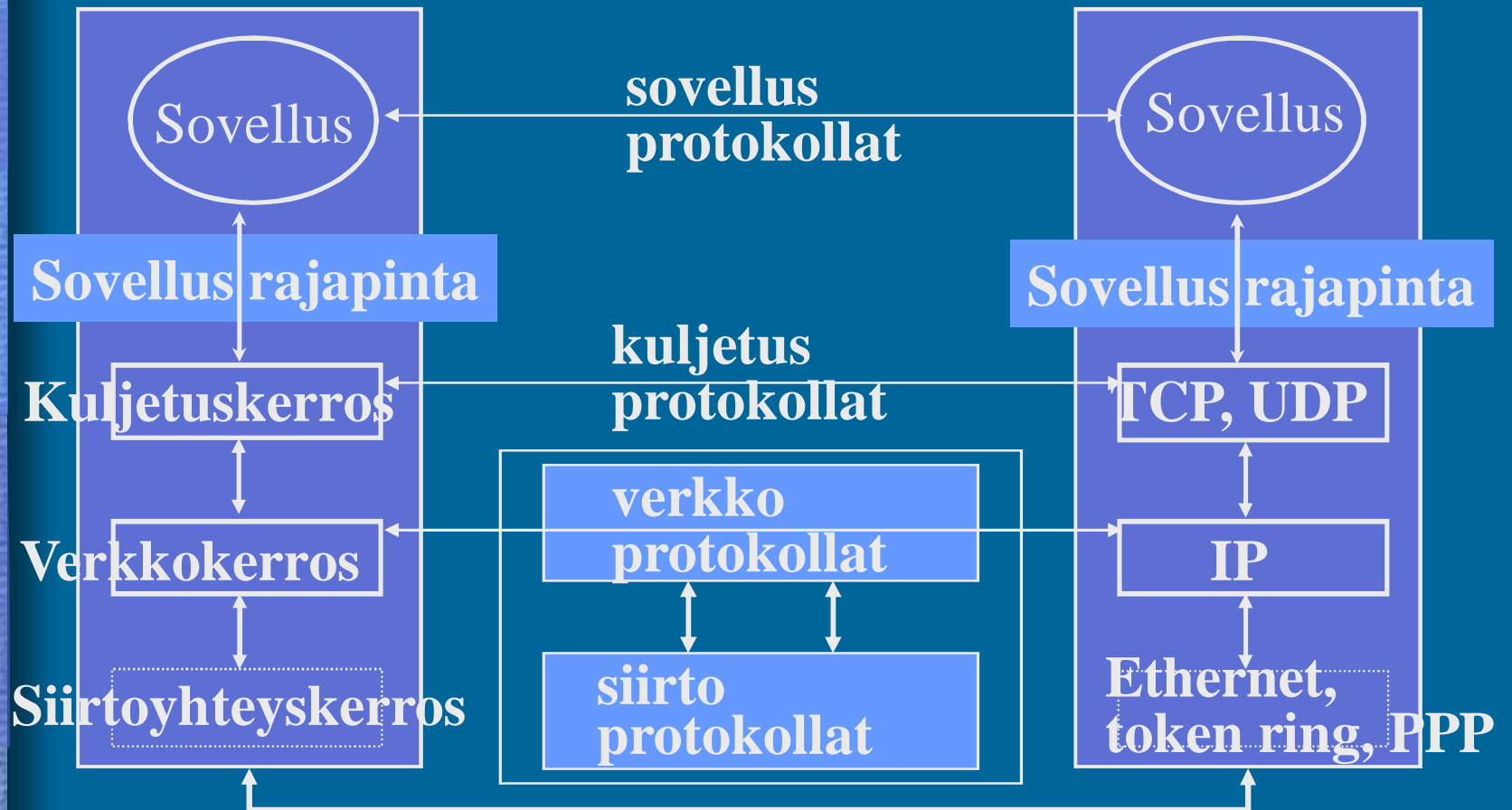
- Sopiva 4. vuoden opiskelijalle
 - Hajautettujen järjestelmien ja tietoliikenteen erikoistumislinjan maisteriopintojen pakollinen opintojakso
- Käyttöjärjestelmän rooli prosessien ja resurssien valvojana
- Samanaikaiset prosessit resurssien käyttäjinä
- Systemin resurssien jakelu
- Prosessien vuoronanto (skedulointi)
- Jatkoa
 - Hajautetut järjestelmät, 4 op

Tietoliikenteen perusteet, 4 op

- Sopiva 2. vuoden opiskelijalle
 - Aineopintojen pakollinen kurssi
- Tietokoneverkkojen peruspalvelut käyttäjälle ja sovelluksille
- Verkkojen tiedonsiirron perusvälineistö
- Verkkoarkkitehtuurin kerrosrakenne ja kunkin tason palvelut
- Jatkoa syventävällä tasolla
 - Internet-protokollat, 4 op

Tietoliikenteen perusteet ...

TCP/IP -kerrosmalli



Rinnakkaisohjelmointi, 4 op

- Sopiva: 2. vuoden opiskelijoille
 - Aineopintojen pakollinen kurssi
- Samanaikaisuuden aiheuttamat ongelmat
 - järjestelmä kaatuu ... miksi niin kävi?
- Prosessien synkronointi eri tapauksissa
 - odottamalla vai prosessia vaihtamalla? miksi? milloin?
- Prosessien kommunikointi eri tavoin
 - yhteinen muistialue? viestit? miksi? milloin?
 - verkon ylitse?
- Jatkoa syventävällä tasolla
 - Hajautetut järjestelmät, 4 op

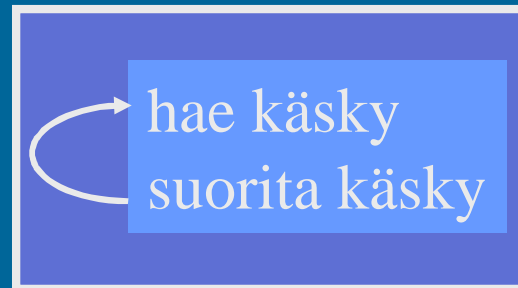
semaforit
monitorit
kohtaaminen
vartioitu rinn. odotus
rpc, viestit
Java rinnakkaisohjelmointi

Spesifioinnin ja verifioinnin perusteet, 4 op

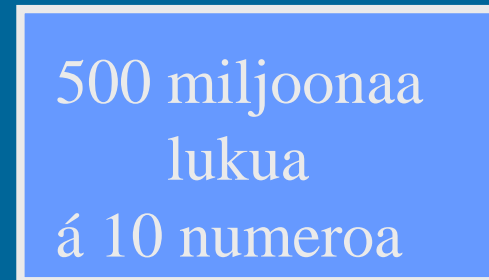
- Sopiva: 4. vuoden opiskelijalle
 - Maisteriopintojen valinnainen erikoiskurssi
- Lähtötiedot: hajautuksen ja samanaikaisuuden problematiikka, TiLi, Rio
- Mallinnetaan prosesseja siirtymäsystemeillä
 - askel: konekäsky? metodi? tapahtuma? ohjelma?
- Automaattisen verifioinnin periaatteet
- Yksinkertaisien protokollien verifiointi
- Jatkoa syventävällä tasolla
 - Ohjelmien semantiikka (?), 6 op
 - Automaattinen verifiointi (?), 6 op

Laskennan teorian perusta (2)

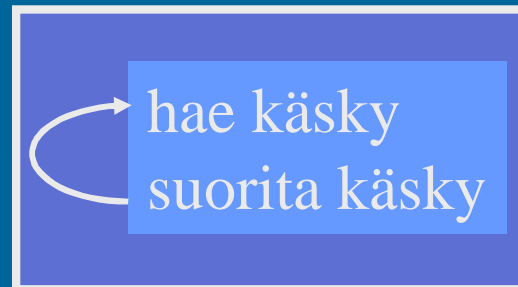
suoritin - CPU



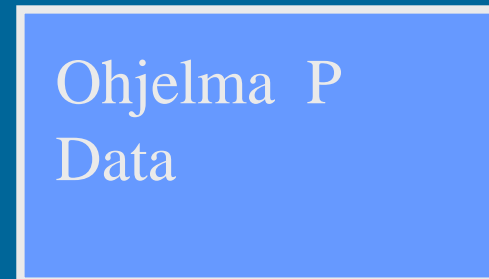
muisti



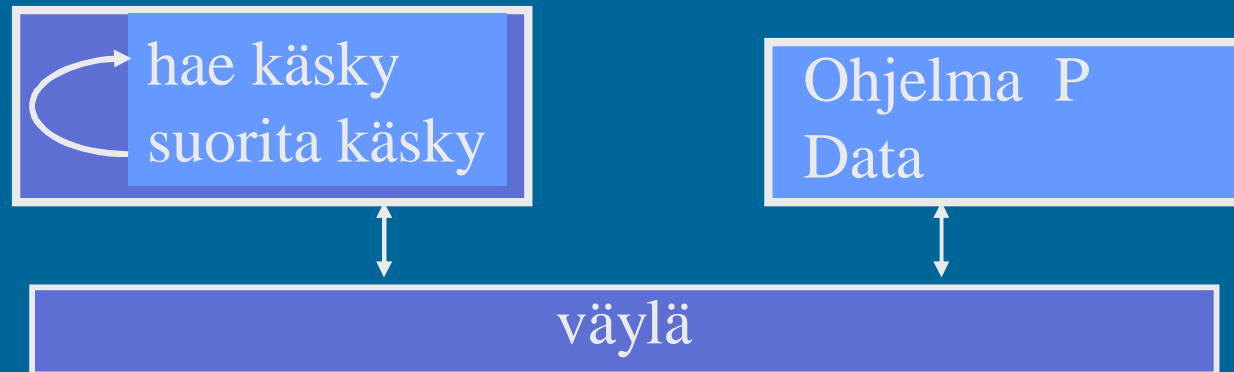
suoritin - CPU



muisti



Laskennan teoriaa ... (5)



Muistin sisältö
ennen P:n suoritusta:

X = hyvin iso kokonaisluku
(500M numeroa?)

Muistin sisältö P:n
suorituksen jälkeen:

Y = joku toinen hyvin iso luku

P on kokonaislukuarvoinen funktio $P: \mathbb{N} \rightarrow \mathbb{N}$

Ohjelman P esitysmuoto muistissa: Iso kokonaisluku, $P \in \mathbb{N}$

Laskennan teoriaa ... (5)

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



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

Laskennan teoriasta ja algoritmianalyysistä todistettuja lauseita ⁽⁴⁾

- Valitaanpa mikä tahansa aikaraja tai muistin koko, 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

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

--
Luennon 12
ja
koko kurssin
loppu
--



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

