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)

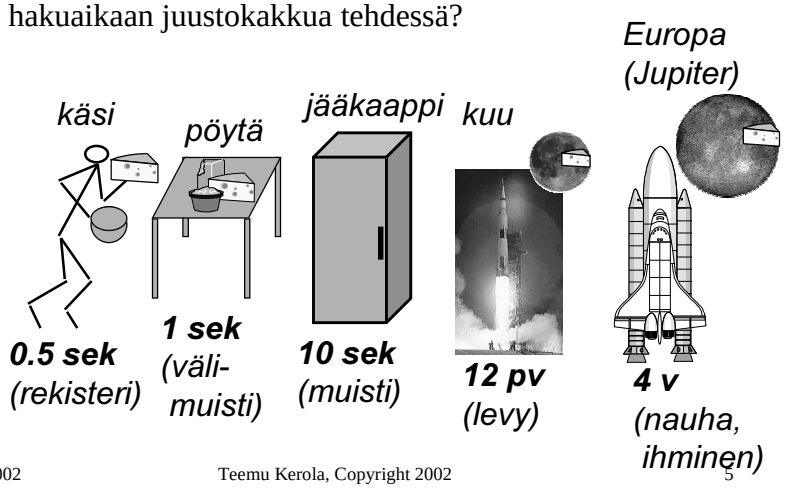
- Ohjelman suoritusnopeus perustuu suorittimen (CPU) suorittamiin konekäskyihin, ei pelkästään ohjelman korkean tason esitysmuotoon
- Ylemmän tason asioiden ymmärtäminen on helpompaa (mahdollista), kun ymmärtää alemman tason asiat

Keskeisiä asioita

- 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
- *Esimerkkejä keskeisistä asioista seuraavilla kalvoilla*

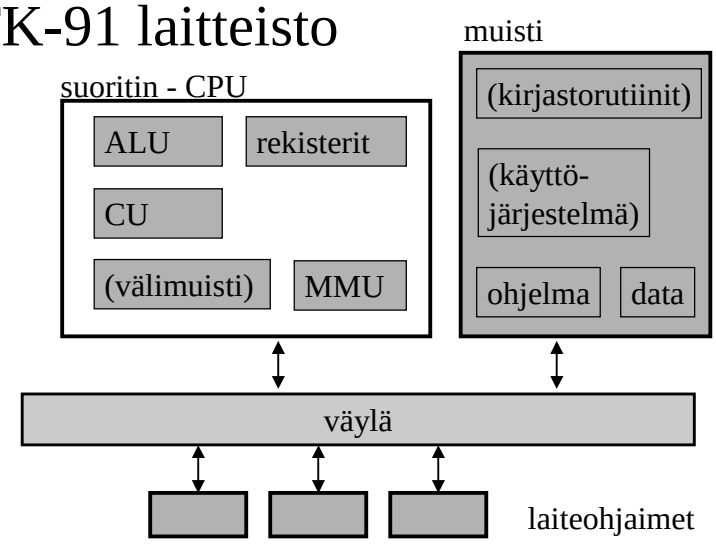
Nopeuserot: Teemun juustokakku

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



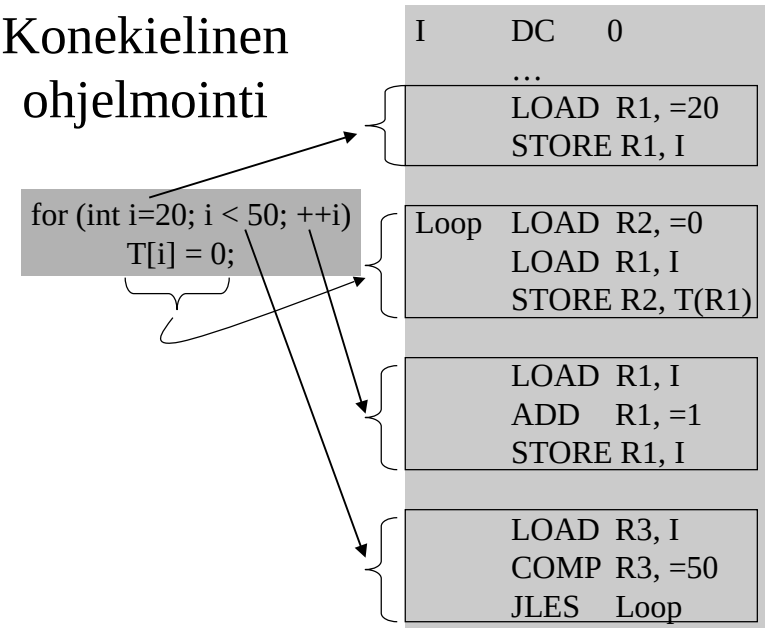
12/04/2002 Teemu Kerola, Copyright 2002 5

Esimerkkikone: TTK-91 laitteisto

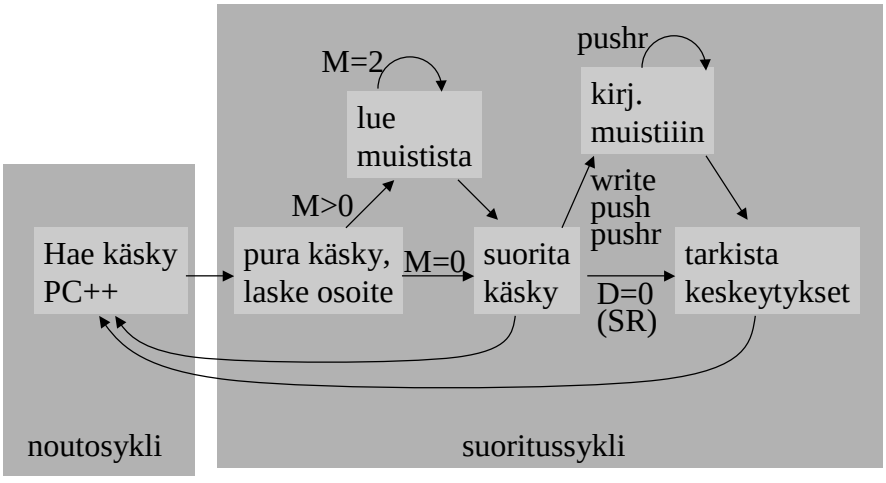


12/04/2002 Teemu Kerola, Copyright 2002 6

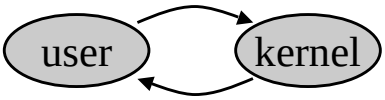
Konekielinen ohjelmointi



Käskyjen nouto- ja suoritussykli



Suorittimen tilat



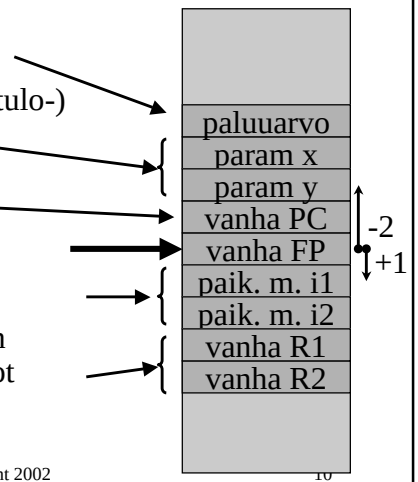
- Käyttäjätila (user mode, normal mode)
  - voi käyttää vain tavallisia käskyjä
  - voi viitata vain käyttäjän omaan muistiavaruuteen (MMU valvoo)
- Etuoikeutettu tila tai (KJ:n) ytimen tila (kernel mode, privileged mode)
  - voi käyttää kaikkia [illegible] myös etuoikeutettuja (esim. clear_cache, iret)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

Aktivointitietue

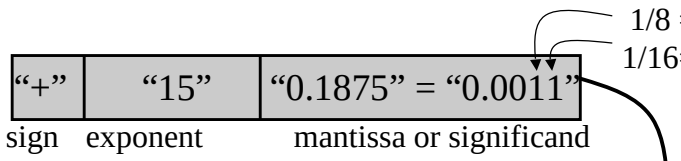
(activation record, activation frame)

int funcA (int x,y);

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



Tiedon esitysmuodot



- 23 bittiä mantissalle, siten että ...

1) Binääripiste (.) on heti ensimmäisen bitin jälkeen

2) Mantissa on normalisoitu: vasemmanpuolimmainen bitti on1

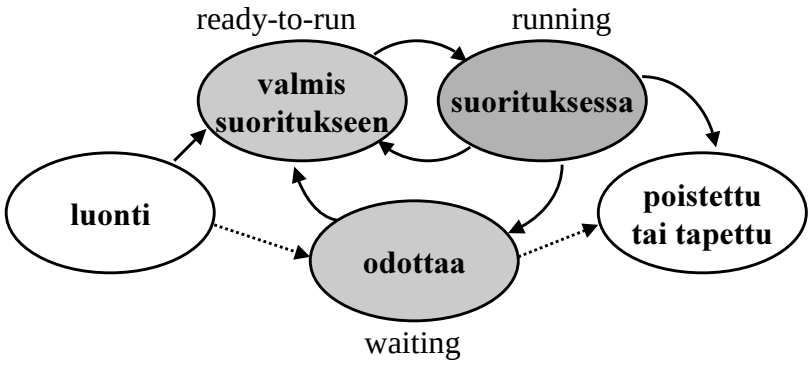
3)Vasemmanpuolimmaista (eniten merkitsevä) bittiä (1) ei talleteta (implied bit)

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

24 bitin mantissa!

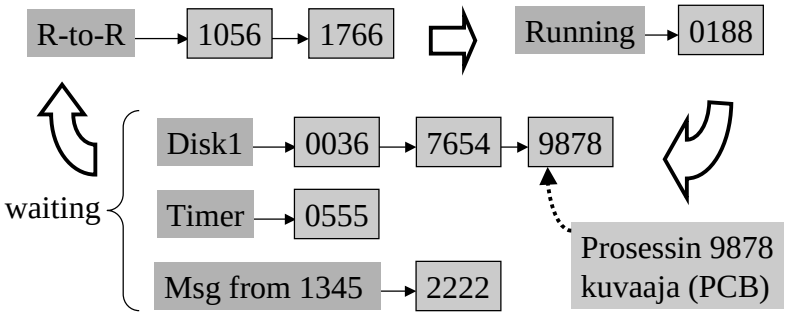
12/04/2002 Teemu Kerola, Copyright 2002 11

Prosessin tilat ja elinkaari



12/04/2002 Teemu Kerola, Copyright 2002 12

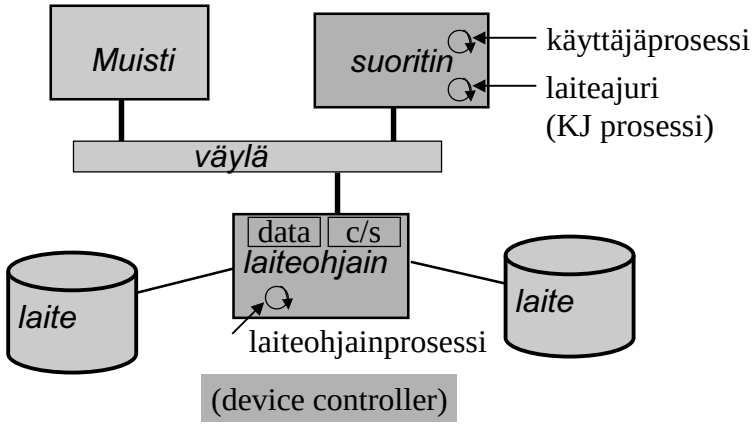
Prosessit jonoissa ja PCB



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

12/04/2002 Teemu Kerola, Copyright 2002 13

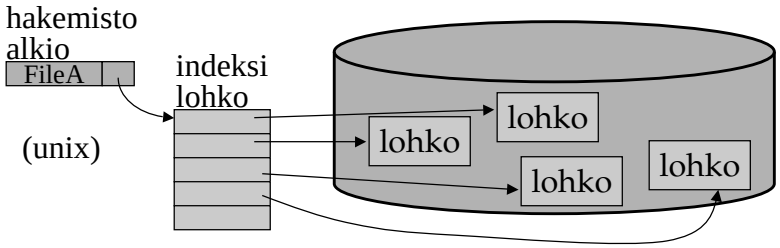
Laiteohjain ja laiteajuri



12/04/2002 Teemu Kerola, Copyright 2002 14

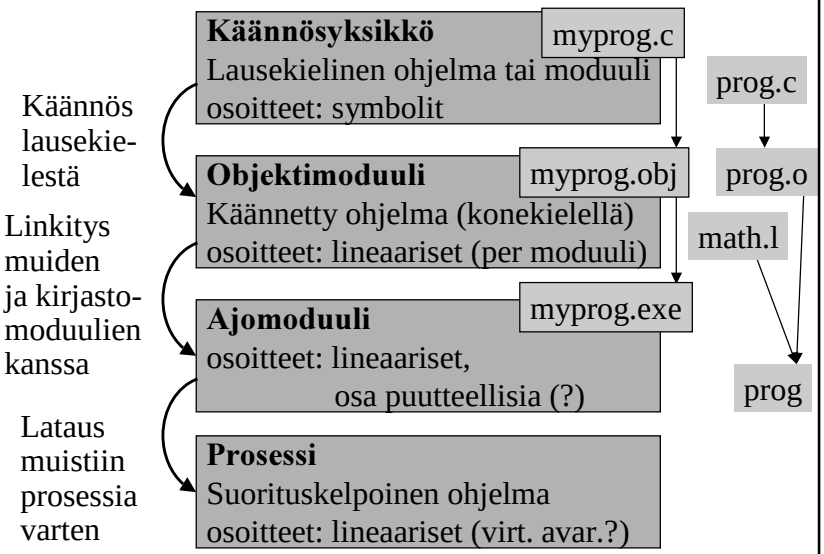
Levyjen käyttö

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

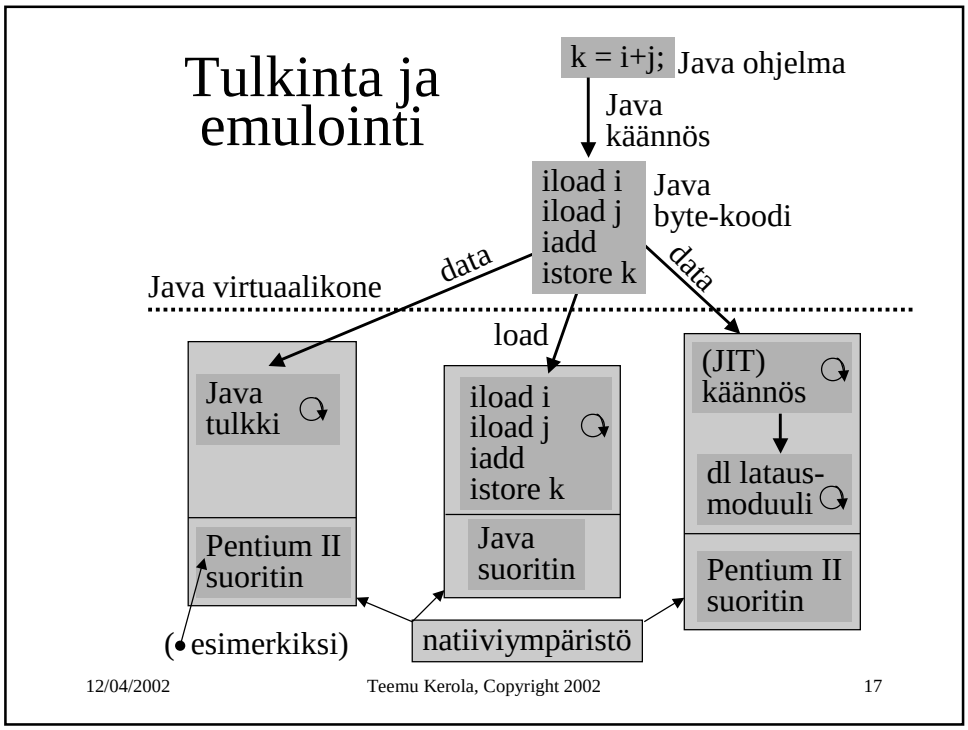


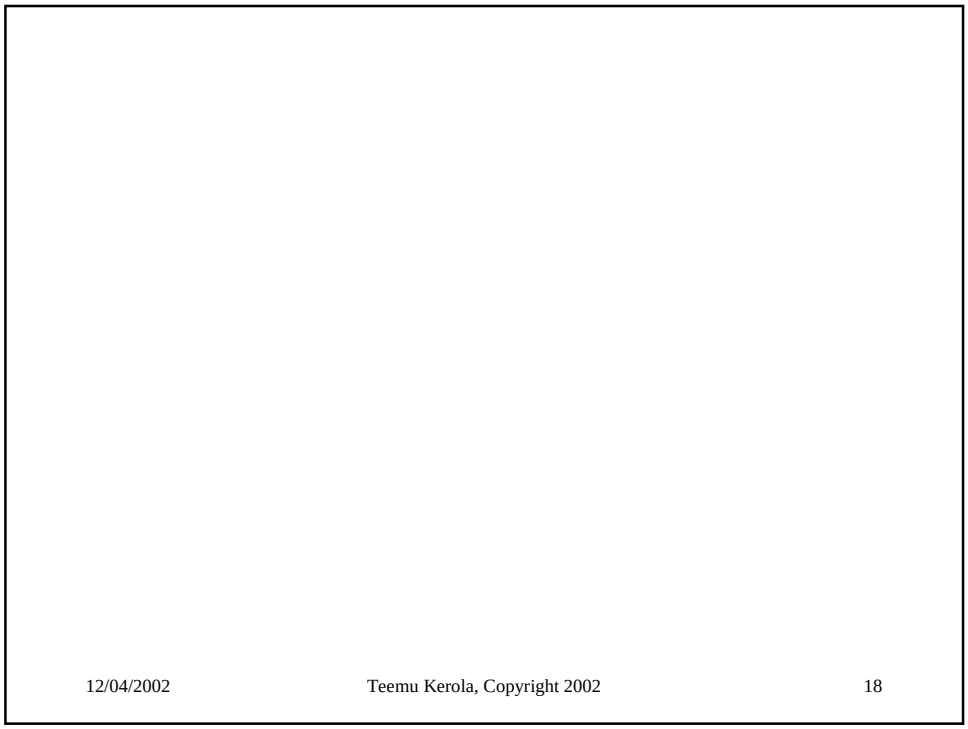
12/04/2002 Teemu Kerola, Copyright 2002 15

Lausekielestä suoritukseen

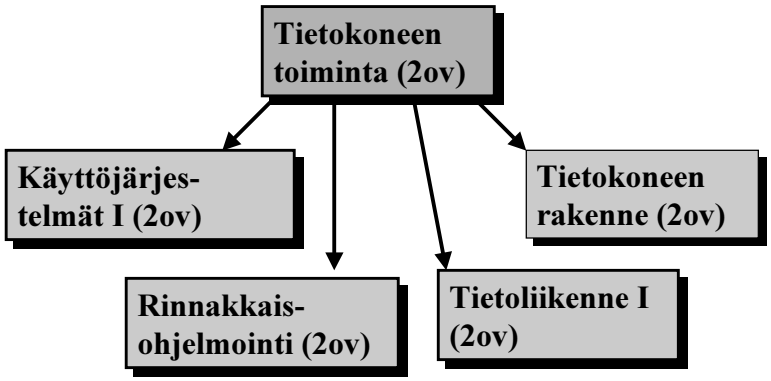


12/04/2002 Teemu Kerola, Copyright 2002 16


Tulkinta ja emulointi
k = i+j;
Java ohjelma
Java käännös
iload i
iload j
iadd
istore k
Java byte-koodi
data
data
Java virtuaalikone
load
Java tulkki
Pentium II suoritin
iload i
iload j
iadd
istore k
Java suoritin
(JIT) käännös
dl lataus-moduuli
Pentium II suoritin
(esimerkiksi)
natiiviympäristö
12/04/2002
Teemu Kerola, Copyright 2002
17

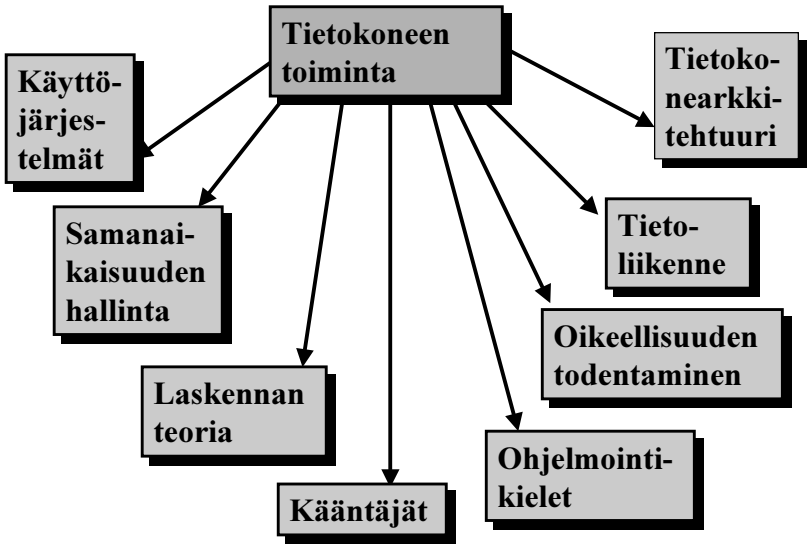

12/04/2002
Teemu Kerola, Copyright 2002
18

Kurssien välisiä suhteita



12/04/2002 Teemu Kerola, Copyright 2002 19

Asioiden välisiä suhteita (8)



12/04/2002 Teemu Kerola, Copyright 2002 20

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 ….

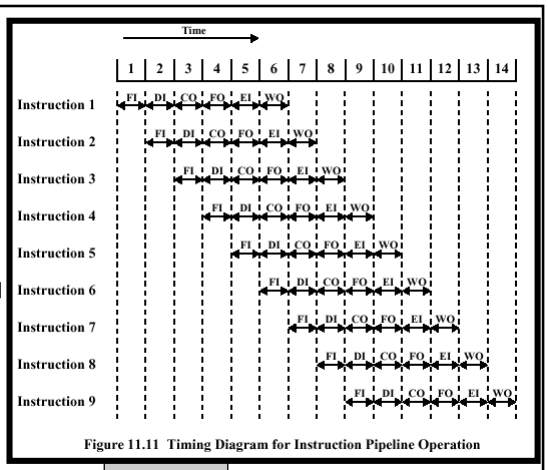


Figure 11.11 Timing Diagram for Instruction Pipeline Operation

[Stal99]

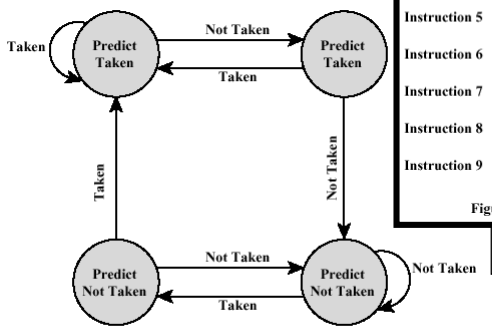


Figure 11.16 Branch Prediction State Diagram

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 ...

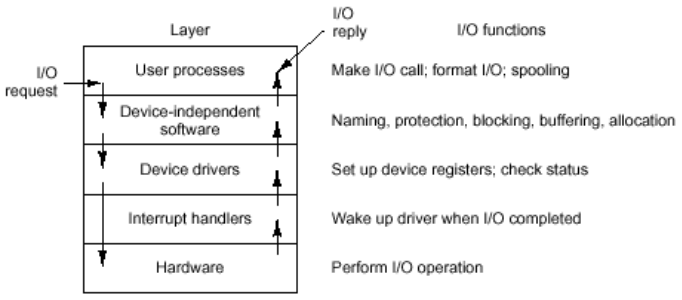
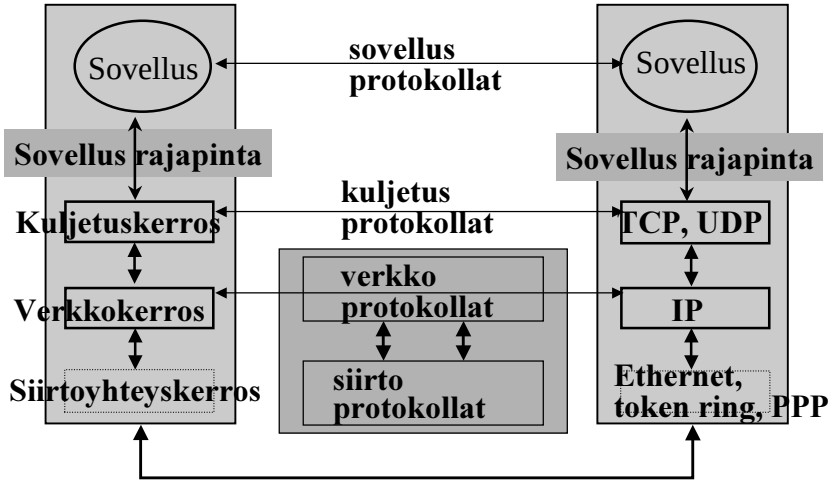


Figure 3-6. Layers of the I/O system and the main functions of each layer.

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
  - odottamalla vai prosessia vaihtamalla?
- 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

tämän käskyn osoite

```
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      # odota kunnes lukko auki
      ...
      kriittinen vaihe: yksi prosessi kerrallaan
      ...
      LOAD  R1,=1     # avaa lukko L
      STORE R1,L
```

- Toimiiko, jos tulee keskeytys paikassa kohta?
  - 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 — C, Pascal
  - oliokielet — Smalltalk
  - funktionaaliset kielet — Scheme, ML
  - logiikkaohjelmointikielet — Prolog
- Jatkoa syvemmällä tasolla:
  - ??

12/04/2002 Teemu Kerola, Copyright 2002 29

Ohjelmointikielten kääntäjät, 5 ov

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

12/04/2002 Teemu Kerola, Copyright 2002 30

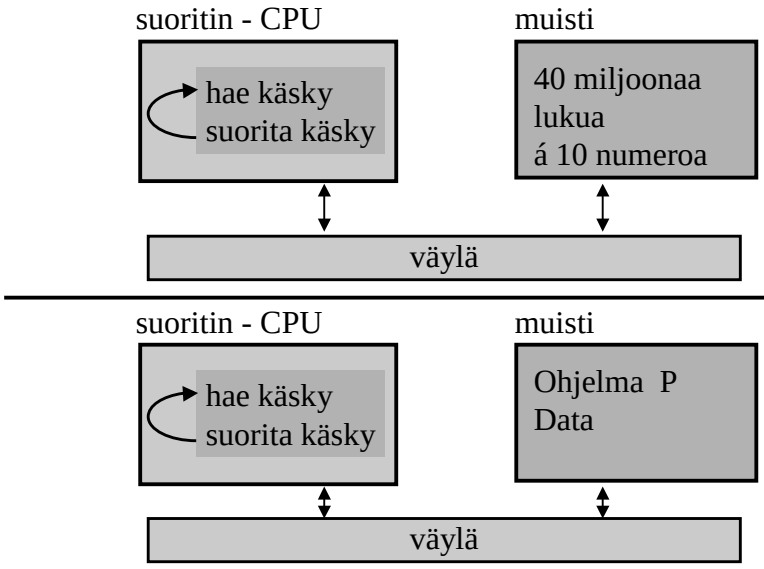
Spesifioinnin ja verifioinnin perusteet, 2 ov

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

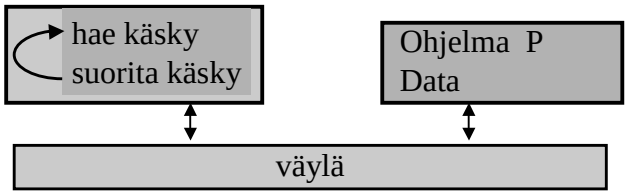
Ohjelmoinnin ja laskennan perusmallit (OLPM), 2 ov

- 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:

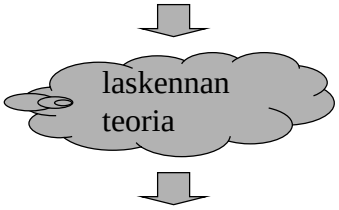
Y = joku toinen hyvin iso luku

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

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

Laskennan teoriaa ... (4)

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



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

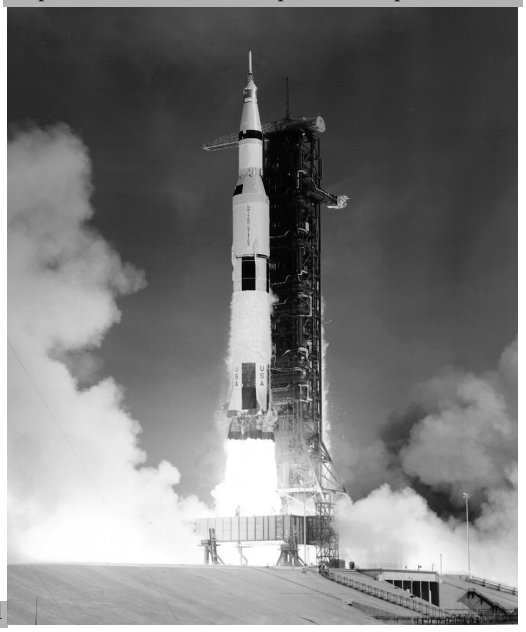
Laskennan teoriasta ja algoritmianalyysistä todistettuja lauseita (3)

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



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

12/04/2002 Teemu Kerola, Copyright 2002 37