

Kertausluento 1 (lu01, lu02, lu03)

Tietokonejärjestelmän rakenne ttk-91 ja sillä ohjelmointi

Järjestelmän eri tasot
Laitteiston nopeus
ttk-91 rakenne ja käskykanta-
arkkitehtuuri
Konekielinen ohjelmointi
ttk-91:llä
(Titokone, TitoTrainer)

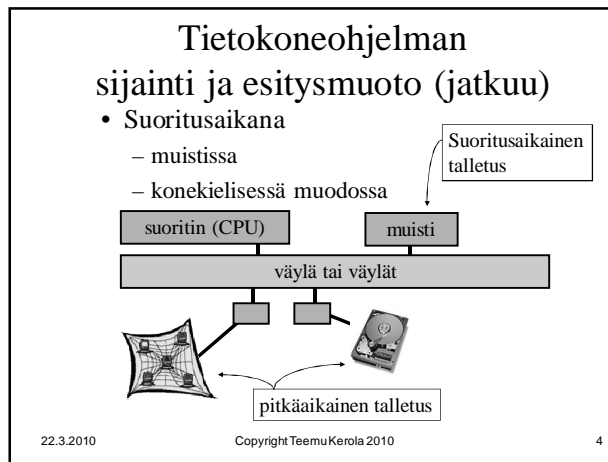
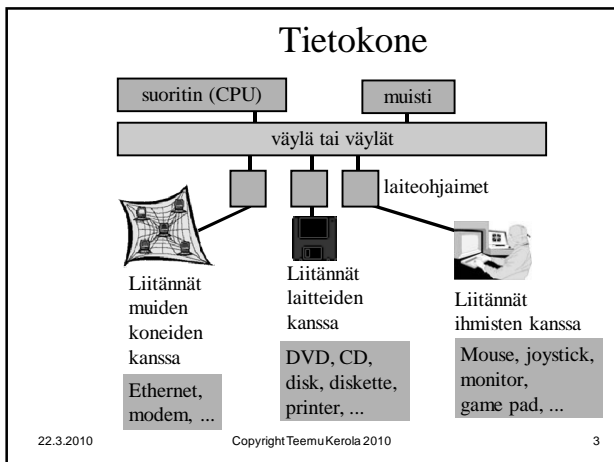
22.3.2010 Copyright Teemu Kerola 2010 1

Luento 1

Tietokonejärjestelmän rakenne

Järjestelmän eri tasot
Laitteiston nopeus

22.3.2010 Copyright Teemu Kerola 2010 2



Konekieli

- Suorittimen konekielen käskykanta määrittelee tietokoneen käskykanta-arkkitehtuurin
 - ISA - Instruction Set Architecture
- Kukin käsky on esim. 10-numeroinen kokonaisluku

2234563212
5437658756
- Usein esitetty symbolisella konekielellä
 - käsky jaettu osiin (kenttiin) **LOAD R1,Summa**
 - joidenkin kenttien arvot kuvattu symboleilla
 - helpompi ihmisten lukea ja kirjoittaa

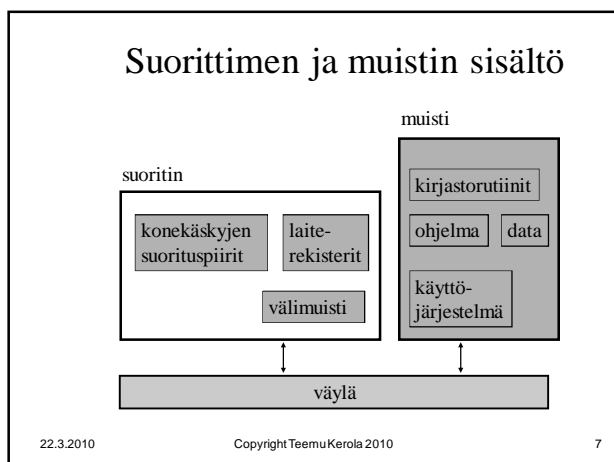
22.3.2010 Copyright Teemu Kerola 2010 5

Symbolinen konekieli

- Yleinen esitystapa konekielisille ohjelmille
 - luettavassa muodossa oleva konekieli
- Helppo muuttaa konekieleksi
 - suora vastaavuus konekieleeseen
 - usein mielletään (vähän väärin, muttei paljon):
symbolinen konekieli ≈ konekieli

129543876	≈	LOAD	R2, Summa	;	R2 ← Mem(Summa)
439874387	≈	ADD	R2, =5	;	R2 ← R2 + 5
544399765	≈	JUMP	Loop	;	PC ← Loop
		(koodi)			(; kommentti)

22.3.2010 Copyright Teemu Kerola 2010 6



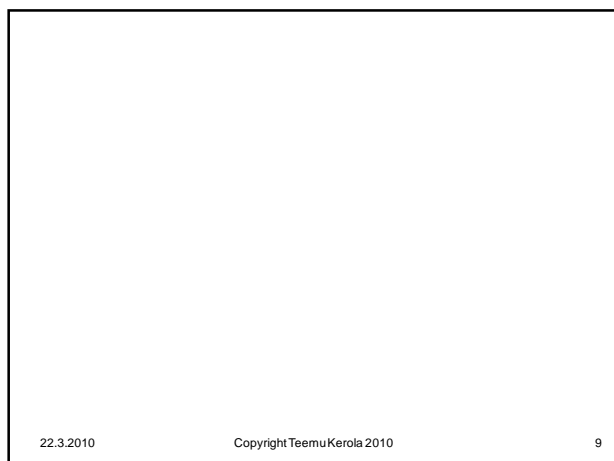
Teemun juustokakku (5)

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

käsi	pöytä	jääkaappi	kuu	Europa (Jupiter)
0.5 sek (rekisteri)	1 sek (välimuisti)	10 sek (muisti)	12 pv (levy)	4 v (nauha, ihminen)

oikea: 10 ms?

22.3.2010 Copyright Teemu Kerola 2010 8



Luento 2

TTK-91 tietokone ja sen KOKSI simulaattori

Miksi TTK-91?

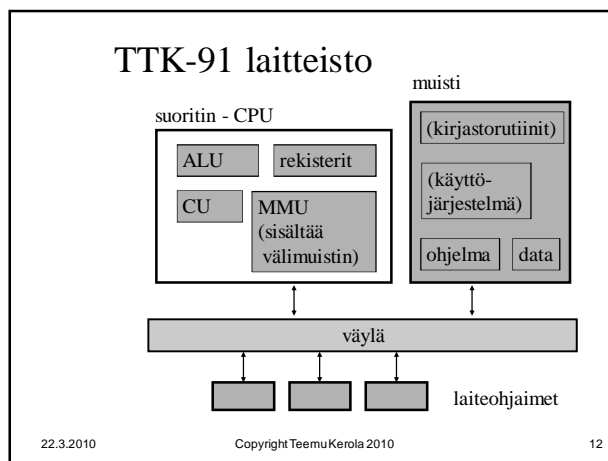
TTK-91 rakenne ja käskykanta-arkkitehtuuri

Mikä on simulaattori?

Miten TTK-91 ohjelmia suoritetaan simulaattorissa?

22.3.2010 Copyright Teemu Kerola 2010 10

- ### Tietokone TTK-91
- Laitteisto, hardware (HW)
 - suoritin, muisti, väylät, oheislaitteiden liitännät
 - Käskykanta - konekieliarkkitehtuuri
 - käyttöliittymä laitteistoon
 - konekäsky, tiedon esitysmuodot, tietotyypit
 - Symbolinen konekieli
 - luettavampi muoto konekielestä
 - kullakin symbolilla yksikäsitteiset arvot
 - Titokone Simulaattori
 - TTK-91 koneen laitteiston simulaattori
 - symbolisen konekielen kääntäjä
 - graafinen käyttöliittymä, debugger-ympäristö
- 22.3.2010 Copyright Teemu Kerola 2010 11



TTK-91 Käskykanta

- Tietotyypit
- Konekäskyjen tyypit
- Konekäskyn rakenne
 - montako bittiä, minkälainen sisäinen rakenne
- Muistissa olevan tiedon osoitustavat
 - konekielessä
 - symbolisessa konekielessä
- Operaatiot

22.3.2010

Copyright Teemu Kerola 2010

13

TTK-91 tietotyypit

- 32 bittinen kokonaisluku
 - noin 10-numeroinen desimaaliluku
- EI:
 - liukulukuja
 - merkkejä
 - totuusarvoja
 - ...

22.3.2010

Copyright Teemu Kerola 2010

14

Luento 3 Konekielinen ohjelmointi (TTK-91, KOKSI)

Muuttujat
Tietorakenteet
Kontrolli
Optimointi
Tarkistukset

22.3.2010

Copyright Teemu Kerola 2010

15

22.3.2010

Copyright Teemu Kerola 2010

16

Tiedon sijainti suoritusaikana

- Muistissa (=keskusmuistissa)
 - iso Esim. 256 MB, tai 64 milj. 32 bitin sanaa
 - hidas Esim. 10 ns
 - data-alueella vai konekäskyssä vakiona?
- Rekisterissä
 - pieni Esim. 256 B, tai 64 kpl 32 bitin sanaa
 - nopea Esim. 1 ns TTK-91: 8 kpl + PC + ...
- Probleemi: milloin muuttujan X arvo pidetään muistissa ja milloin rekisterissä?
 - missä päin muistia? miten siihen viitataan?

22.3.2010

Copyright Teemu Kerola 2010

17

Miten tietoon viitataan? ⁽³⁾

- Tieto muistissa
 - muistiosoitteen (esim. 0x6F123456 tai 3459321) avulla
 - symbolin (esim. HenkTunn tai X) avulla symbolista konekieltä käytettäessä – symbolin arvo on muistiosoite
 - HenkTunn = 0x6F123456, X = 3459321 (heksadesimaali) (desimaali)
- Tieto välimuistissa
 - samalla tavalla kuin jos tieto olisi muistissa
 - viittaushetkellä ei tiedetä, kummasta paikasta tieto lopulta löytyy tai kauanko viittaamiseen kuluu aikaa!
- Tieto rekisterissä
 - rekisterin osoitteen (esim. nro 6 tai 18) avulla
 - symbolinen konekieli: R3, FP, F5, I2, jne
- Tieto konekäskyssä (vakiona)
 - oletusarvoisesti, käskyssä on vain yksi paikka tiedolle

22.3.2010

Copyright Teemu Kerola 2010

18

Tieto ja sen osoite (2)

muuttujan X osoite on symbolin X arvo

X DC 12

 LOAD R1, =X
 LOAD R2, X

symbolin X arvo
 muuttujan X osoite?

muisti

230
12345
12556
128765
12222
12
X=230: 12
12998

int x = 12; muuttujan X arvo

- Muuttujan X osoite on 230
- Muuttujan X arvo on 12
- Symbolin X arvo on 230
 - symbolit ovat yleensä olemassa vain käännösaikana
 - virheilmoituksia varten symbolitaulua pidetään joskus yllä myös suoritusaikana

22.3.2010 Copyright Teemu Kerola 2010 19

Muistitilan käyttö yhdelle ttk-91 ohjelmalle P (6)

0: Ohjelmakoodi

0: Globaalit muuttujat ja tietorakenteet

Pino (stack) aliohjelmien toteuttamista varten (myös paikalliset muuttujat)

Vapaa muistitila

Keko (heap) dynaamisesti (esim. Javan "new") varattavia muistialueita varten (ei ttk-91:ssa...)

BASE LIMIT P SP HP iso:

22.3.2010 Copyright Teemu Kerola 2010 20

Tiedon sijainti suoritusaikana (3)

- Rekisteri (nopein)
 - kääntäjä päättää milloin muuttujan arvo on rekisterissä
- Välimuisti (nopea)
 - laitteisto hoitaa automaattisesti joillekin muistialueille
- Muisti (hidas)
 - kääntäjä/lataaja valitsee sijaintipaikan
 - globaali data ohjelman latauksen yhteydessä
 - vakiot konekäskyssä
 - ohjelma sijoittaa suoritusaikana
 - aliohjelmien paikalliset muuttujat, parametrit
 - käyttäjärjestelmä sijoittaa suoritusaikana
 - dynaaminen data keossa suorituksen aikana
- Levy, levypalvelin (liian hidas, ei mahdollista)
 - vaatii käyttäjärjestelmän varusohjelmien apua

22.3.2010 Copyright Teemu Kerola 2010 21

Ohjelmoinnin peruskäsitteet

- Aritmeettinen lauseke
 - miten tehdä laskutoimitukset?
- Yksinkertaiset tietorakenteet
 - yksiulotteiset taulukot, tietueet
- Kontrolli - mistä seuraava käsky?
 - valinta: if-then-else, case
 - toisto: for-silmukka, while-silmukka
 - aliohjelmat, virhetilanteet
- Monimuuttuimaiset tietorakenteet
 - listat, moniulotteiset taulukot

22.3.2010 Copyright Teemu Kerola 2010 22

For lauseke (4)

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

Olisiko parempi pitää i:n arvo rekisterissä? Miksi? Milloin?

Mikä on i:n arvo lopussa? Onko sitä olemassa?

Entä jos toisenlainen loop-semantiikka?

22.3.2010 Copyright Teemu Kerola 2010 23

While-do -lauseke (2)

X = 14325;
 Xlog = 1;
 Y = 10;
 while (Y < X) {
 Xlog++;
 Y = 10 * Y
 }

LOAD R1, =14325
 STORE R1, X
 LOAD R1, =1 ; R1=Xlog
 LOAD R2, =10 ; R2=Y

While COMP R2, X
 JNLES Done

ADD R1, =1
 MUL R2, =10
 JUMP While

Done STORE R1, Xlog ; talleta tulos
 STORE R2, Y

Mitä kannattaa pitää muistissa?

Mitä kannattaa pitää missä rekisterissä ja milloin? X in R3?

22.3.2010 Copyright Teemu Kerola 2010 24

Koodin generointi ⁽⁷⁾

- Kääntäjän viimeinen vaihe
 - voi olla 50% käännösajasta
- Tavallisen koodin generointi
 - alustukset, lausekkeet, kontrollirakenteet
- Optimoidun koodin generointi
 - käännös kestää (paljon) kauemmin
 - suoritus tapahtuu (paljon) nopeammin
 - milloin globaalin/paikallisen muuttujan X arvo kannattaa pitää rekisterissä ja milloin ei?
 - missä rekisterissä X:n arvo kannattaa pitää?
 - joskus R1:ssä, joskus R5:ssä?

22.3.2010 Copyright Teemu Kerola 2010 25

Taulukon indeksitarkistus

```

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

I	DC	0
T	DS	50 ; data
Tsize	DC	50 ; koko
...		

```

LOAD R1, =20
STORE R1, I
Loop
LOAD R2, =0
LOAD R1, I
JNNEG R1, ok1
SVC SP,=BadIndex
ok1 COMP R1, Tsize
JLES ok2
SVC SP,=BadIndex
ok2 STORE R2, T(R1)
LOAD R1, I
ADD R1, =1
STORE R1, I ; 50 OK!
LOAD R3, I
COMP R3, =50
JLES Loop
    
```

Voisiko loopin kontrollia ja indeksin tarkistusta yhdistää?
Optimoiva kääntäjä osaa!

22.3.2010 Copyright Teemu Kerola 2010 26

Moni-ulotteiset taulukot ⁽³⁾

- Talletus riveittäin
 - C, Pascal, Java?
- Talletus sarakeittain
 - Fortran
- 3- tai useampi ulotteiset
 - vastaavalla tavalla!

T	T[0][0]
T	T[0][1]
T	T[0][2]
T	T[1][0]
T	T[1][1]
T	T[1][2]
T	T[2][0]
T	T[...][...]

T	T[0][0]
T	T[1][0]
T	T[2][0]
T	T[3][0]
T	T[0][1]
T	T[1][1]
T	T[2][1]
T	T[...][...]

22.3.2010 Copyright Teemu Kerola 2010 27

Linkitetty lista ⁽⁸⁾

R1: -1
R2: 132

First →

Data Next → Data Next → Data Next →

2-sanainen tietue

First=200:

211
33
255
55
-1
44
222

```

list_sum.k91
Data EQU 0 ; suht. osoite
Next EQU 1
Sum DC 0
Main LOAD R1, First ; ptrRec
JNEG R1, Done
LOAD R2, =0 ; sum
Loop ADD R2, Data(R1)
LOAD R1, Next(R1)
JNEG R1, Loop
JNNEG R1, Loop
STORE R2, Sum
SVC SP,=HALT
Done
    
```

Virhe, bugi! Missä?

22.3.2010 Copyright Teemu Kerola 2010 28

-- Luennon loppu --

Konrad Zuse: Z1 (1938)

- mekaaninen "laskin", kellotaajuus 1 Hz (käännä kampea!)
- kertolasku 5 s
- datamuisti 64W à 24b
- ohjelma reikänauhalta (filmiltä)

Components of the Z1

Memory - 64 words à 22 bits

Selection Unit

Memory Block 64 x Exponent + Sign à 8 bits

Memory Block 32 x Mantissa à 14 bits

Memory Block 32 x Mantissa à 14 bits

Memory Control

Arithmetic Unit (Floating Point)

Exponent

Mantissa

Mantissa

Register R1

Register R2

Control Unit

Output

Input

http://irb.cs.tu-berlin.de/~zuse/Konrad_Zuse/en/Rechner_Z1.html

22.3.2010 Copyright Teemu Kerola 2010 29