

Tietokonejärjestelmän rakenne ttk-91 ja sillä ohjelmointi

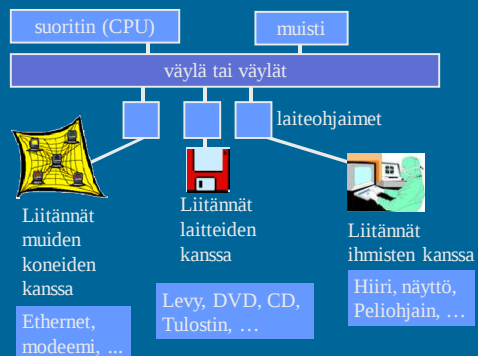
27.1.2011

27.1.2011

Copyright Kerola & Niklander

1

Tietokone



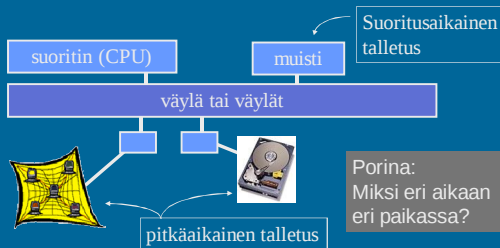
27.1.2011

Copyright Kerola & Niklander

2

Tietokoneohjelman sijainti

- Suoritusaikana muistissa
- Muuna aikana esim. levyllä, verkossa, tms.



27.1.2011

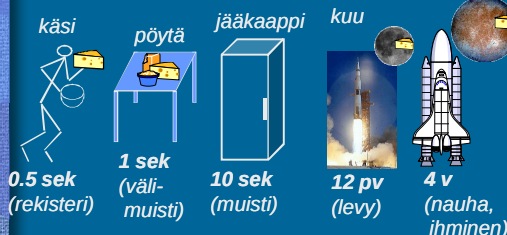
Copyright Kerola & Niklander

3

Teemun juustokakku

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

Europa (Jupiter)



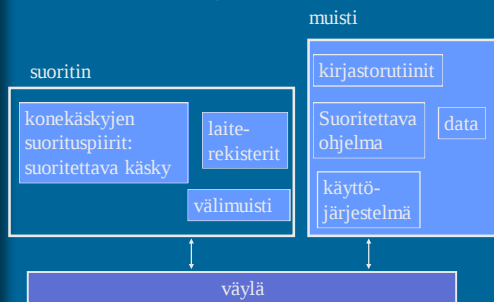
Äänestys: Mikä on eniten pielessä 5 vuoden päästä?

27.1.2011

Copyright Kerola & Niklander

4

Suorituksen aikainen suorittimen ja muistin sisältö



27.1.2011

Copyright Kerola & Niklander

5

Ohjelman esitysmuoto: Konekieli

- Suorittimen konekielen käskykanta määrittelee tietokoneen käskykanta-arkkitehtuurin
 - ISA - Instruction Set Architecture
- Ohjelma keskusmuistissa konekielisenä (TTK-91)

	Konekielinen käsky	lukuna
0:	0000 0010 0000 00 000 0000 0000 0110 0100	DEC: 33554532
1:	0000 0010 0001 01 000 0000 0000 0110 0100	DEC: 36175972
2:	0001 0100 0001 00 000 0000 0000 0000 0000	DEC: 337641472
3:	0010 0010 0000 00 000 0000 0000 0000 0110	HEX: 22000006
4:	0000 0001 0010 00 000 0000 0000 1110 0100	HEX: 012000E8
5:	0000 0000 0000 00 000 0000 0000 0000 0000	HEX: 00000000

27.1.2011

Copyright Kerola & Niklander

6

Ohjelman esitysmuoto: symbolinen konekieli

- Usein symbolisella konekielellä
 - käskey jaettu osiin (kenttiin)
 - joidenkin kenttien arvot kuvattu symboleilla
 - helpompi ihmisten lukea ja kirjoittaa

Symb. konekieli	Konekielinen käskey
LOAD R0, =100	0000 0010 0000 0000 0000 0000 0110 0100
LOAD R1, 100	0000 0010 0001 0100 0000 0000 0110 0100
DIV R1, R0	0001 0100 0010 0000 0000 0000 0000 0000
JZER 6	0010 0010 0000 0000 0000 0000 0000 0110
STORE R1, 228	0000 0001 0010 0000 0000 0000 1110 0100
NOP	0000 0000 0000 0000 0000 0000 0000 0000

27.1.2011

Copyright Kerola & Niklander

7

Esitysmuotoja

osoite	sisältö		
Address	Contents	Address	Contents
101	0010 0010 0000 0001	101	2201
102	0001 0010 0000 0010	102	1202
103	0001 0010 0000 0011	103	1203
104	0011 0010 0000 0100	104	3204
201	0000 0000 0000 0010	201	0002
202	0000 0000 0000 0011	202	0003
203	0000 0000 0000 0100	203	0004
204	0000 0000 0000 0000	204	0000

(a) Binary program

(b) Hexadecimal program

Address	Instruction	Address
101	LDA 201	201
102	ADD 202	202
103	ADD 203	203
104	STA 204	204
201	DAT 2	2
202	DAT 3	3
203	DAT 4	4
204	DAT 0	0

(c) Symbolic program

Label	Operation	Operand
FORMUL	LDA	I
	ADD	J
	ADD	K
	STA	N
I	DATA	2
J	DATA	3
K	DATA	4
N	DATA	0

(d) Assembly program

Symboli
(nimi)

27.1.2011

Computer Organization II, Spring 2009, Tiina Niklander

(Sta06 Fig 10.11)

8

TTK-91 konekieli

Operaatio	Rj	M	Ri	Attribuutti (arvo tai osoite)
8 bit	3 b	2b	3 b	16 bittia

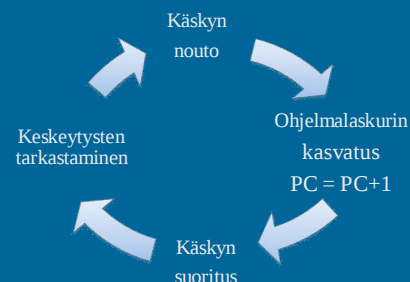
- Kukin käskey 32 bittia. Oma operaatiokoodi kullakin käskeyllä
- Käskeyn rekisterien ja attribuuttien tulkinta riippuu käskeystä ja osoitusmuodosta (M)
- Tietotyyppi:
 - Vain 32-bittinen kokonaisluku
 - Ei: merkkejä, liukulukuja, totuusarvoja

27.1.2011

Copyright Kerola & Niklander

9

Suorittimen toiminta



27.1.2011

Copyright Kerola & Niklander

10

Tiedon sijainti suoritusaikana

- Muistissa (=keskusmuistissa)
 - iso Esim. 256 MB, tai 64 milj. 32 bitin sanaa
 - hidas Esim. 10 ns
 - data-alueella vai konekäskeyssä vakiona?
- Rekisterissä
 - pieni Esim. 256 B, tai 64 kpl 32 bitin sanaa
 - nopea Esim. 1 ns TTK-91: 8 kpl + PC + ...

Milloin muuttujan X arvo pidetään muistissa ja milloin rekisterissä? Missä päin muistia? Miten siihen viitataan?

27.1.2011

Copyright Kerola & Niklander

11

Tiedon sijainti suoritusaikana

- 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 konekäskeyssä
 - ohjelma sijoittaa suoritusaikana
 - aliohjelmien paikalliset muuttujat, parametrit
 - käyttöjärjestelmä sijoittaa suoritusaikana
 - dynaaminen data keossa suorituksen aikana
- Levy, levypalvelin (liian hidas, ei mahdollista)
 - vaatii käyttöjärjestelmän varusohjelmien apua

27.1.2011

Copyright Kerola & Niklander

12

Porinaa

Kuka päättää ja milloin, josko muistipaikan 0x4A86BB00 (muuttuja Big) sisältö on jonkin tietyn konekäskyn suoritushetkellä välimuistissa tai ei?

Mitä suoritin tekee sillä aikaa, kun välimuistista puuttuva tieto haetaan muistista?

27.1.2011

Copyright Kerola & Niklander

13

Miten tietoon viitataan?

- 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

27.1.2011

Copyright Kerola & Niklander

14

Tieto ja sen osoite

```
X DC 12
....
LOAD R1, =X
LOAD R2, X
```

int x = 12;

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

muisti
230
12345
12556
128765
12222
12
12998

27.1.2011

Copyright Kerola & Niklander

15

Tiedon osoitusmuodot TTK-91

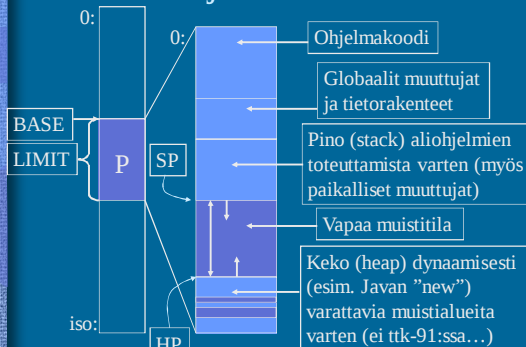
- Välitön operandi (ei muistiosoitusta)
 - OPER Rj, =ADDR(Ri) M=00
- Suora indeksoitu muistiosoitus
 - OPER Rj, ADDR (Ri) M=01
- Epäsuora indeksoitu muistiosoitus
 - OPER Rj, @ADDR(Ri) HUOM: ei R0 M=10

27.1.2011

Copyright Kerola & Niklander

16

Muistitilan käyttö yhdelle ttk-91 ohjelmalle P

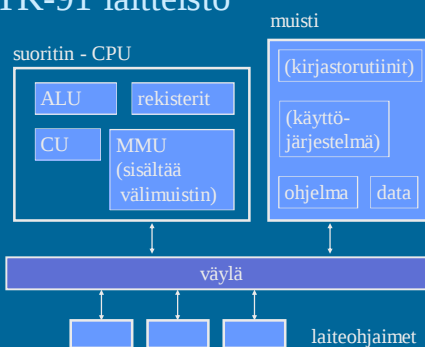


27.1.2011

Copyright Kerola & Niklander

17

TTK-91 laitteisto



27.1.2011

Copyright Kerola & Niklander

18

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
- Monimutkaiset tietorakenteet
 - listat, moniulotteiset taulukot

27.1.2011

Copyright Kerola & Niklander

19

For lauseke

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

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

Mikä on i:n arvo lopussa?
Onko sitä olemassa?
Entä jos toisenlainen
toisto-semantiikka?

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

27.1.2011

Copyright Kerola & Niklander

20

While-do -lauseke

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

Mitä kannattaa
pitää muistissa?

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

	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

27.1.2011

Copyright Kerola & Niklander

21

Koodin generointi

- Kääntäjän viimeinen vaihe
 - voi olla 50% käännösjästa
- 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?

27.1.2011

Copyright Kerola & Niklander

22

Taulukon indeksitarkistus

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

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

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

	LOAD	R1, =20 (kesk.)
	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

27.1.2011

Copyright Kerola & Niklander

23

Moniulotteiset taulukot ⁽³⁾

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

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

27.1.2011

Copyright Kerola & Niklander

24

R1: -1
R2: 132

First → Data Next → Data Next → Data Next →

2-sanainen tietue

First=200:

211

33

255

55

-1

44

222

R1 → 211:

R2: 0

R1 → 222:

R2: 77

R1 → 255:

R2: 33

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

STORE R2, Sum

SVC SP, =HALT

Done

Virhe, bugi! Missä?

27.1.2011

Copyright Kerola & Niklander

25

-- Luennon loppu --

Konrad Zuse: Z1 (1938)

• mekaaninen "laskin", kelloaajuus 1 Hz (käännä kampea!)

• kertolasku 5 s

• datamuisti 64W à 24b

• ohjelma reikänauhalta (filmiltä)

Components of the Z1

Memory - 64 words a 22 bits

Memory Block 64 x Exponent + Sign a 8 bits

Memory Block 32 x Mantissa a 14 bits

Memory Block 32 x Mantissa a 14 bits

Arithmetic Unit (Floating Point)

Exponent

Mantissa

Mantissa

Register R1

Register R2

Control Unit

Output

Input

Selection Unit

Punch Tape Reader

Program Unit

Sign Unit

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

27.1.2011

Copyright Kerola & Niklander

26

Kertausluento luennoista 1-3

5