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

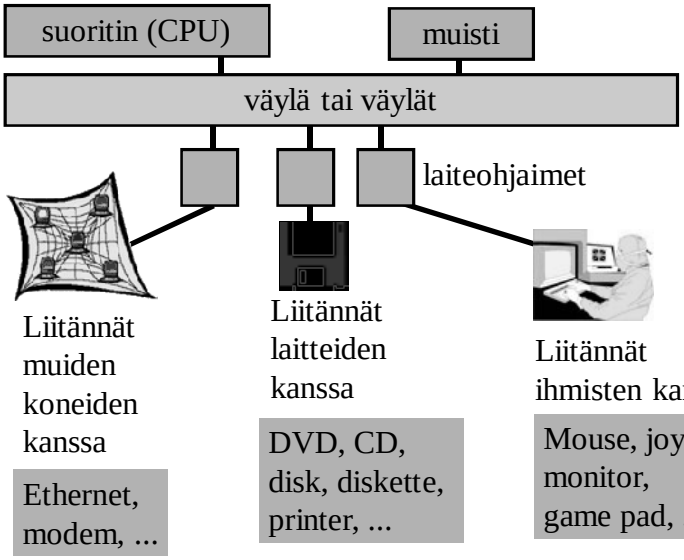
(Titokone, TitoTrainer)

Luento 1 Tietokonejärjestelmän rakenne

Järjestelmän eri tasot

Laitteiston nopeus

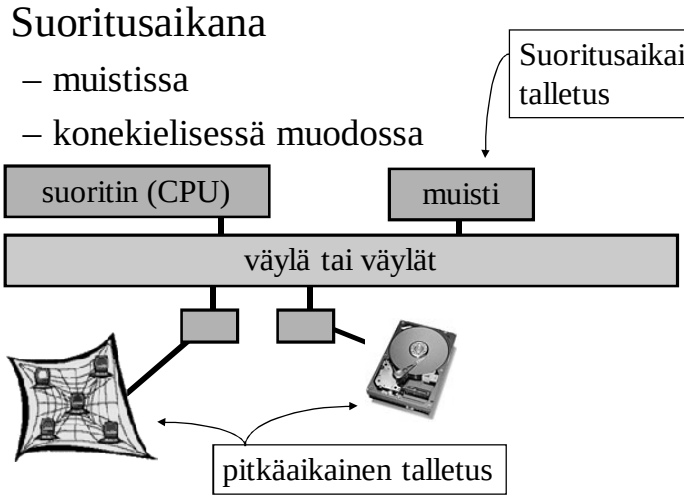
Tietokone



22.3.2010 3

Tietokoneohjelman sijainti ja esitysmuoto (jatkuu)

- Suoritusaikana
  - muistissa
  - konekielisessä muodossa



22.3.2010 4

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

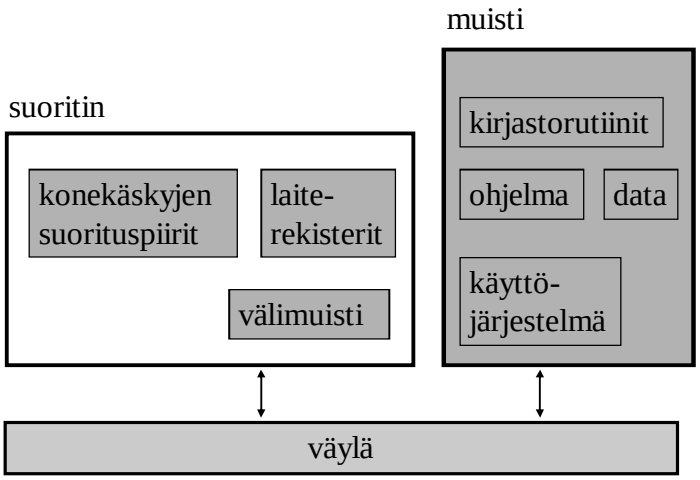
Symbolinen konekieli

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

symbolinen konekieli ≈ konekieli

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

Suorittimen ja muistin sisältö



Teemun juustokakku (5)

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

oikea: 10 ms?

Luento 2
TTK-91 tietokone ja sen KOKSI simulaattori

Miksi TTK-91?

TTK-91 rakenne ja käskykanta-arkkitehtuuri

Mikä on simulaattori?

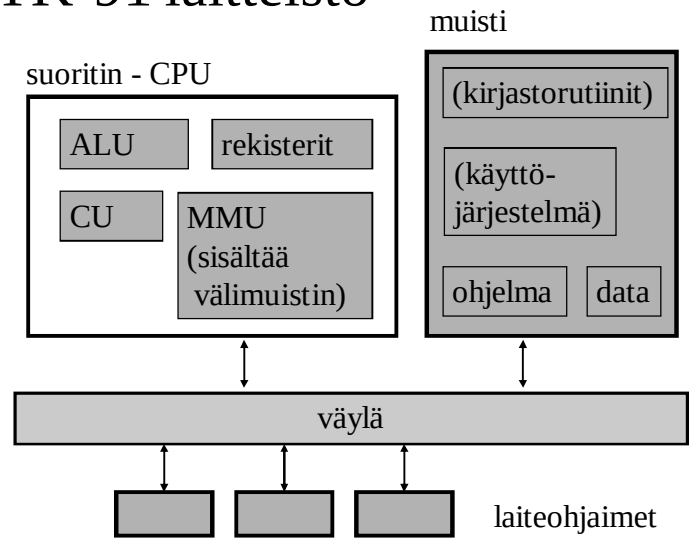
Miten TTK-91 ohjelmia suoritetaan simulaattorissa?

Tietokone TTK-91

- Laitteisto, hardware (HW)
  - suoritin, muisti, väylät, oheislaitteiden liitännät
- Käskykanta - konekieliarkkitehtuuri
  - käyttöliittymä laitteistoon
  - konekäskyt, 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 laitteisto



22.3.2010 Copyright Teemu Kerola 2010 12

TTK-91 Käskykanta

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

TTK-91 tietotyypit

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

Luento 3
Konekielinen ohjelmointi (TTK-91, KOKSI)

Muuttujat
Tietorakenteet
Kontrolli
Optimointi
Tarkistukset

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?

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

Tieto ja sen osoite (2)

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

int x =12;

muuttujan X osoite on symbolin X arvo

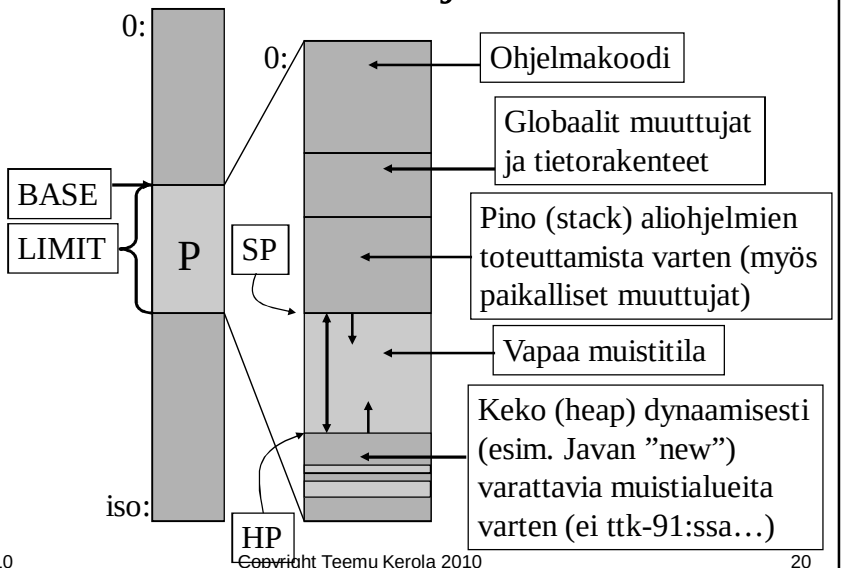
symbolin X arvo
muuttujan X osoite?

muuttujan X arvo

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

- 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

Muistitilan käyttö yhdelle ttk-91 ohjelmalle P (6)



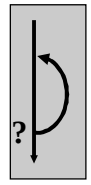
Tiedon sijainti suoritusaikana (3)

- 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
  - käyttöjärjestelmä sijoittaa suoritusaikana
    - dynaaminen data keossa suorituksen aikana
- Levy, levypalvelin (liian hidas, ei mahdollista)
  - vaatii käyttöjärjestelmän varusohjelmien apua

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

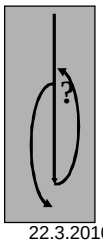
For lauseke (4)




```
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
```

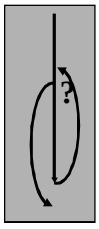
Olisiko parempi pitää i:n arvo rekisterissä? Miksi? Milloin?



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

Entä jos toisenlainen loop-semantiikka?

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?

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:ssa?

Taulukon indeksitarkistus

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

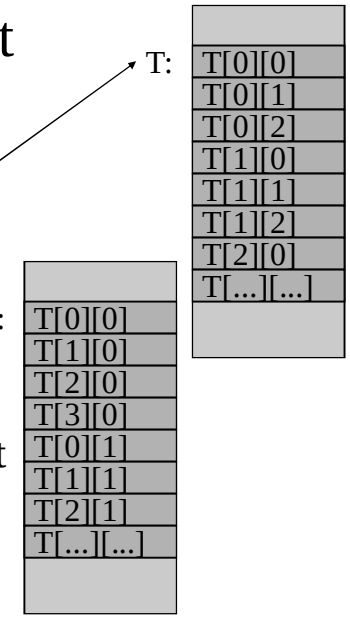
```
I       DC    0

T       DS    50  ; data
Tsize   DC    50  ; koko
        …
```

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

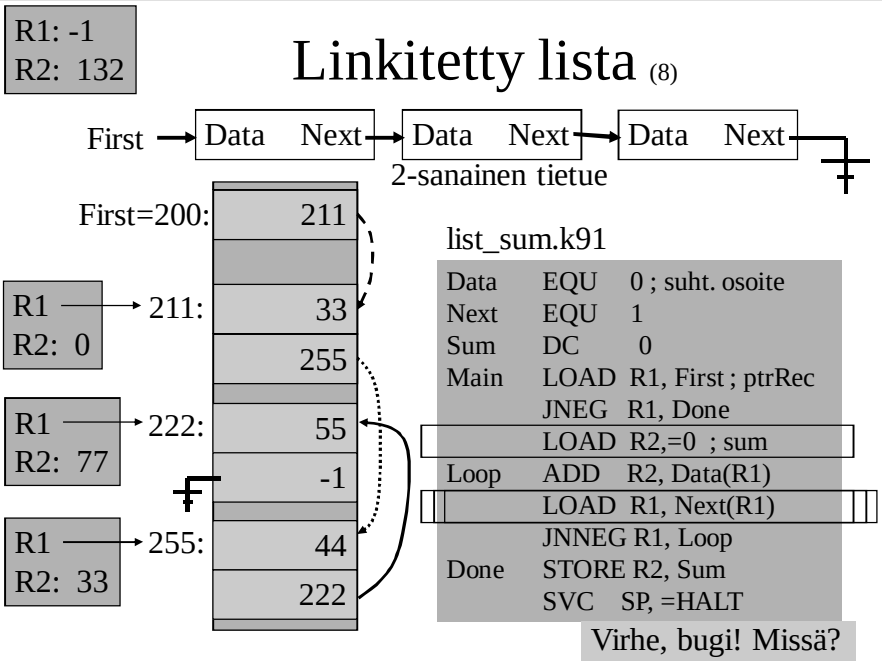
```
        LOAD  R1, =20
        STORE R1, I
Loop    LOAD  R2, =0
        LOAD  R1, I
        JNNEG R1, ok1
        SVC   SP,=BadIndex
ok1     COMP  R1, Tsize
        JLES  ok2
        SVC   SP, =BadIndex
ok2     STORE R2, T(R1)
        LOAD  R1, I
        ADD   R1, =1
        STORE R1, I ; 50 OK!
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

Moni-ulotteiset taulukot (3)

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



Linkitetty lista (8)



list_sum.k91

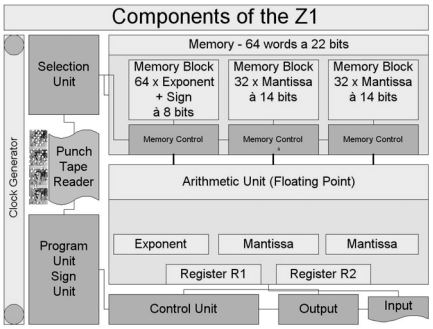
```
Data    EQU    0 ; suht. osoite
Next    EQU    1
Sum     DC      0
Main    LOAD   R1, First ; ptrRec
        JNEG   R1, Done
        LOAD   R2,=0  ; sum
Loop    ADD    R2, Data(R1)
        LOAD   R1, Next(R1)
        JNNEG  R1, Loop
Done    STORE  R2, Sum
        SVC    SP, =HALT
```

Virhe, bugi! Missä?

-- Luennon loppu --

Konrad Zuse: Z1 (1938)

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



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

22.3.2010 Copyright Teemu Kerola 2010 29