Suomalaiset ohjelmointikielet

Timo Lilja

TKK

6. 10. 2009

Sisältö

1. FAS
   - Kieli
   - Rakenteet
   - Makrot
   - Yhteenveto
   - Lähteet
2. Shines
   - Kieli
   - Esimerkki
3. Muita kieliä
4. Yhteenveto

FAS-kieli (1/2)

- Suunniteltu vuosina 1967–1969 Valtion tietokonekeskuksessa.
  - pääasialliset suunnittelijat: Tuomas Kotovirta ja Simo Töyrä
- Tarkoitettu kaupallis-hallinnollisiin ohjelmointitehtäviin.
- Laitteistoriippumaton
- Käännetty kieli
- Moduulijärjestelmä
- Makrojärjestelmä

FAS-kieli (2/2)

- Käännös yhdellä läpikäynnillä (single-pass).
- Kääntäjä generoi suoraan ajettavan binäärin (ei linkkeriä).
- Automaattinen ajonaikainen objektimoduulien segmentointi
  - ohjelmaa voidaan ajaa koneessa jossa vähemmän fyysista muistia kuin ohjelma koko (vrt. virtuaalimuisti)
  - koodin oltava relokatoituvaa
- FAS-ohjelmointijärjestelmä koostuu kolmesta loogisesta osasta:
  - kääntäjän ydin
  - lausekkedein käsittelymoduulit
  - ajonaikainen monitori (run-time monitor)
- Kääntäjällä pieni muistijälki (80 kt) OS/360-järjestelmässä.
- S/360 malli 50 (n. 1972)
  - pienillä työkuormilla 80% ajasta käyttöjärjestelmässä (overhead)
  - eräajokuorma: 3000 lauseketta/minuutti.

Suunnitteluperiaatteet

- Laitteistoriippumattomuus
  - tiedostojen merkkijonesitys
  - kokonaisluvut (ei liukulukuja)
  - lukujen pituus määritelty
- Soveltuvuus hallinnolliseen ohjelmointiin
  - jonomuuttujat ja -taulukot
  - standardoitu tiedostoformaatti: peräkkäis- ja hajatiedostot, lukujen/merkkijonojen porttautuva esitysmuoto
- Huonojen ohjelmointiratkaisujen välttäminen
  - moduulirakenne: viestinvälitys vain rajapinnan läpi
  - ei osoittimia
  - tiettyyn tiedostoon kohdistuvat operaatiot vain yhdestä moduulista
  - kaikki muuttujat alustettava
- Makrojärjestelmä
- Ei rekursiota
- Dynaaminen linkkeri: moduuleja voidaan ladata ajonaikaisesti

Kontrollirakenteet (1/2)

- Ehdoton hyppy: ⟨*nimi*⟩ GOTO ⟨*hyppyosoite*⟩
- Monihaarainen hyppy

```
⟨nimi⟩    FORK ⟨järjestysnumero⟩ ⟨hyppyosoite⟩,
               ⟨järjestysnumero⟩ ⟨hyppyosoite⟩
               ...
```

- Ehtolause:

⟨*nimi*⟩ IF ⟨*ehtolauseke*⟩ THEN ⟨*lause1*⟩ [ELSE ⟨*lause2*⟩]

```
IF A+B LT C+5/2   ..IF-LAUSE ON
THEN X =1         ..YKSI
ELSE GOTO OSA2    .LAUSE

IF JONO EQ 'A' THEN GOTO AJONO
```

Kontrollirakenteet (2/2)

- Silmukkarakenne: ⟨*nimi*⟩ LOOP ⟨*loppunimi*⟩ ⟨*kierrosmäärä*⟩

```
LOOP  E2 I+J                    .ULKOINEN SILMUKKA
.
IF    X EQ Y THEN GOTO YLI      .HYPPY POIS ON SALLITTU
.
LOOP  E3 10                     .SISÄINEN SILMUKKA
.
E3  END
E2  END
YLI A=B
```

- Ohjelman lopettaminen: ⟨*nimi*⟩ STOP ⟨*paluukoodi*⟩ [DUMP]
  - DUMP-määreellä muisti vedostetaan

Moduulit ja aliohjelmet

- Moduulit
  - jokaisessa ohjelmassa oltava `MAIN`-moduuli
  - moduulin määrittely: ⟨*tunniste*⟩ `SUBR` [⟨*arg*⟩, ⟨...⟩]
  - moduulien kutsu: ⟨*nimi*⟩ `CALL` ⟨*tunniste*⟩ [⟨*arg*⟩, ⟨...⟩]
  - moduulista paluu ⟨*nimi*⟩ `RETURN`
  - moduulin kääntämisen lopetus: `FIN`
  - moduulit käännettävissa erikseen
  - tiedonvälitys vain parameterien avulla (enkapsulointi)
  - moduulit dynaamisesta ladattavissa
  - relokatoituvia
- Sisäiset alirutiinit
  - määrittely: ⟨*tunniste*⟩ `BEGIN` ⟨...⟩ `BACK`
  - kutsu: `EXEC` ⟨*nimi*⟩
  - ei rekursiota

Muut lausekkeet

- Tiedostojen käsittely
  - primitiivisinä sarakepohjaiset tietueenkäsittelyrutiinit
  - tiedoston avaus, sulku, kirjoitus, päivitys (vrt. `SQL UPDATE`)
  - tiedostot siirrettäviä alustalta toiselle
- Aritmeettiset lausekkeet
  - evaluoidaan vasemmalta oikealle
  - lauseke `A + B * B/C` vastaa lauseketta $(A + B) * B/C$
- Merkkijonon tutkiminen
  - `STAB–SCAN` -rakenne merkkijonohakuihin
  - koodi joka tarkistaa että merkkijonossa `JONO` on vain numeroita:

```
NUMER   STAB  ALPHA, SPEC, ' ' .EI-NUMEROT
.
        SCAN  JONO IND=NUMER
        IF    IND NE 0 THEN GOTO EINUMER
```

Tietorakenteet

- Kokonaisluvaut
  - pituus 4/9/15 numeroa (SHORT/NORM/LONG)
  - positiivinen, negatiivinen tai nolla
  - ohjelmoijalle ei näy laitteiston lukuesitysmuoto
  - esitystarkkuuden ylitys aiheuttaa ajonaikaisen virheen
- Merkkijonot (CHAR)
  - jonon pituus 1 – 9999 merkkiä
  - merkkivalikoima rajoitettu ja standardoitu
  - kokonaislukujen ulkoinen esitysmuoto merkkijonoina
  - merkkijonon osia voidaan käsitellä osajonosyntaksilla
- Taulukot
  - voidaan muodostaa merkkijonoista tai kokonaisluvuista
  - indeksointi lähtee numerosta 1!
  - indeksimuuttuja voi olla SHORT tai NORM

Tietorakenteet: esimerkki

Esimerkkejä kokonaislukujen, merkkijonojen ja osajonojen määrittelystä:

```
A               NORM
B               SHORT   5                       .ALKUARVO

C(5)            LONG                            .TAULUKKO
D(4)            SHORT   1,2,3,4
E(100)          SHORT   1
M(2)            CHAR    3       'ABC','BCD'

SYÖTTÖ          FILE    80
STUNNUS         CHAR    SYÖTTÖ<1,3>             .OSAJONO
SHENK           CHAR    SYÖTTÖ<4,20>
```

Makrot

- CPP-tekstimakroja korkeampitasoisia
- Käsittelevät syntaksielementtejä ja pääsy syntaksitauluun
- Makron määrittely: `MACRO`
- Lausekkeen käsittely
  - lausekkeen syöttö takaisin ohjelmalle: `EMIT`
  - lausekkeen luku makrokutsusta:: `SEPAR`
- Symbolitaulun käsittely
  - arvon haku symbolitaulusta: `ASK`
  - arvon tallentaminen symbolitauluun: `TELL`

Makroesimerkki

Käyttöesimerkki:

```
FOO MAP SYÖTTÖ
    STNNUS  1,3
    SHENK   4,20
    SHENKNO 24,3
```

Vastaa osajonon nimeämisrakennetta:

```
SYÖTTÖ  FILE 80
STNNUS  CHAR SYÖTTÖ<1,3>
SHENK   CHAR SYÖTTÖ<4,20>
SHENKNO CHAR SYÖTTÖ<24,3>
```

Miksi FAS?

- VTKK:lla katsottiin COBOL hankalaksi käyttää
- Oma kieli
  - voitiin jättää standardikielen turhia ominaisuuksia pois
  - lisätä omia, paikallisesti tarpeellisena pidettyjä ominaisuuksia
- FAS-kielen käyttöä edelsi yritys rakentaa FORTRANin ympärille hallinnollista tieojenkäsittelyä helpottavia toimintoja
  - syntaksi FORTRANista, makrojärjestelmä symbolisista konekielistä
- Kielestä vain yksi versio, joka oli VTKK:n hallinnassa
- Kieli on yksinkertainen
  - kääntäjät yksivaiheisia (single pass)
  - kääntäjällä pienempi muistijälki ja selkeä rakenne
- FAS-kieli oli VTKK:n ratkaisu yhteensopivuusongelmaan – FAS-kieli porttattu:
  - "tavukone" IBM/360 (16/32-bittinen)
  - "sanakone" UNIVAC 1108 (18-bittinen)

FASin historiaa

- Perustui Suomen IBM:llä kehitettyyn PAS-järjestelmään (PL/I Assembly Subsystem), tuki laitteistoriippumattomaan datan siirtoon
- Laitteistoja: IBM S/360, DOS, Univac-1108, Honeyewell-66, Mini-6, Nova/Eclipse, PDP-11, VAX-perhe, Dec-System 20 (TKK)
- Mikrotietokoneita varten välikieli
  - ajomonitori sisälsi välikielisen tulkin nimeltä FASOS
  - moniajo-FASOS (diplomityö, Esa Tähtinen)
- FAStoC-kääntäjä Nokia Datalla
  - käytetty sairaalaympäristöissä
- Viimeisein FAS-totetutus OpenVMS:lle
- Kieli ei paljon muuttunut 1970-luvun jälkeen
  - paljon makroja käytössä
  - tuki ulkopuolisille funktioille (foreign function)
- Käytetty Suomen ulkopuolella ainakin Norjassa

Lähteet

- *FAS-Kielen käsikirja*. Valtion tietokonekeskus. 1972
- H. Jäppinen: *FAS-kielisten ohjelmien kääntäminen COBOL-kielelle*. Diplomityö. 1972
- T. Kotovirta: *FAS language compilers*. Valtion tietokonekeskus. Nord Data 1972.
- S. Töyrä: *Macro facility of FAS programming language*. Valtion tietokonekeskus. Nord Data 1972.

Sisältö

1. FAS
   - Kieli
   - Rakenteet
   - Makrot
   - Yhteenveto
   - Lähteet
2. Shines
   - Kieli
   - Esimerkki
3. Muita kieliä
4. Yhteenveto

Shines

- Shades
  - TKK:lla v. 1996 kehitetty tietokannan elvytysalgoritmi
  - pohjautuu versiointiin
  - persistentti muistinhallintajärjestelmä
- Shines
  - puhtaasti funktionaalinen, dynaamisesti tyypitetty
  - tavukoodikäännetty CPS:n avulla
  - tietorakenteet (myös sisäiset) säilytetään Shades-kannassa $\longrightarrow$ täysin persistentti laskennan tila
  - laskenta reaaliaikaista ja voidaan replikoida automaattisesti useammalle koneelle
- Hibase: A Persistent Functional Programming Environment
  - TKK:n ja Nokian yhteinen tutkimusprojetki n. 1999–2001
  - `http://hibase.cs.hut.fi/`

Shines-esimerkki

Esimerkki Mandelbrotin fraktaalin laskevasta ohjelmasta:

```
PRIVATE m_it(c::<INT, INT>,x::<INT, INT>,ctr::INT)::INT :=
  IF ctr >= MAX_CTR
  THEN MAX_CTR
  ELSE LET <cre, cim> := c,
           <xre, xim> := x,
           xre2 := xre * xre,
           xim2 := xim * xim
       IN IF xre2 + xim2 >= 33554432
          THEN ctr
          ELSE m_it(c,
                    <(xre2 - xim2) / 4096 + cre,
                     xre * xim / 2048 + cim>,
                    ctr + 1)
```

http://hibase.cs.hut.fi/shreds/shines/demos/mandel.shines.shtml

Sisältö

1. FAS
   - Kieli
   - Rakenteet
   - Makrot
   - Yhteenveto
   - Lähteet
2. Shines
   - Kieli
   - Esimerkki
3. Muita kieliä
4. Yhteenveto

Muita kieliä

- Survo 1966–
  - S. Mustonen, Helsingin yliopisto
  - alunperin tilastotieteen erikoiskieli, mutta kehittynyt yleisempään suuntaan. Ks. `http://www.survo.fi/`
- ZBasic 1978
  - Timo Koivunen, TKK:n opiskelija
  - pieni muistijälki (14kt), kehitetty Zilogin Z80-prosessorille
- EVE-kieli 1982–1990?
  - R. Linturi, J. Vuori
  - osa TEKESin SAA Dialogue -hanketta
  - Visual Basicin kaltainen, lomake-editori, perintä
- Paikallisia erityiskieliä
  - Mikko Lisp 1986, SSH Scheme 1999–, Hedgehog Lisp
  - Nokia TNSDL

Lähteet

- T. Alanko, S. Mustonen, M. Tienari: *A Statistical Programming Language SURVO 66*. 1968. BIT, 8, 69-85. `http://dx.doi.org/10.1007/BF01939330`
- O. Erling, O. Lassila: *Making Common Lisp Viable on Microcomputers*. 1986. STeP-86 Symposium, Volume 2.
- M. Mäkinen: *Nokia SAGA*. Gummerrus 1995.
- R. Linturi, M. Tala: *Mikrotietokone Suomessa 1973–1993*. Yritysmikrot Oy. 1993.
- P. Saarikoski: *The role of club activity in the early phases of microcomputing in Finland* (Tämä ja muuta historiaan liittyvää: `http://users.utu.fi/petsaari/tutkimukset/`)

Sisältö

1. FAS
   - Kieli
   - Rakenteet
   - Makrot
   - Yhteenveto
   - Lähteet
2. Shines
   - Kieli
   - Esimerkki
3. Muita kieliä
4. Yhteenveto

Yhteenveto

- Suomessa ei ole ollut paljon ohjelmointikielitutkimusta
- Yrityksissa ja yliopistoissa rakennettu paljon erilaisia erikoiskieliä
- Muutamia yleiskäyttöisiä kieliä, mutta ei mitään tunnetumpaa
- Miksi tehdä omia kieliä
  - oliko 30–40 vuotta sitten järkevää lähteä tekemään omaa kieltä?
  - olisiko nyt?
- Tulevaisuus
  - tutkitaanko tai tehdäänkö Suomessa seuraava C# tai Haskell?
  - ohjelmointikielten tutkiminen epätodennäköistä suomessa
  - onko ohjelmointikielissä mitään tutkittavaa?