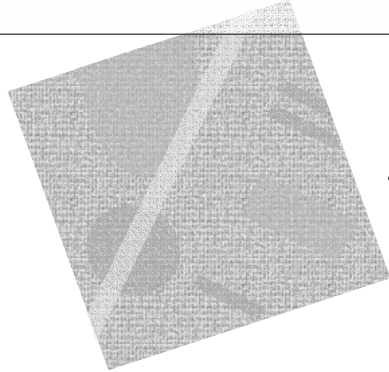


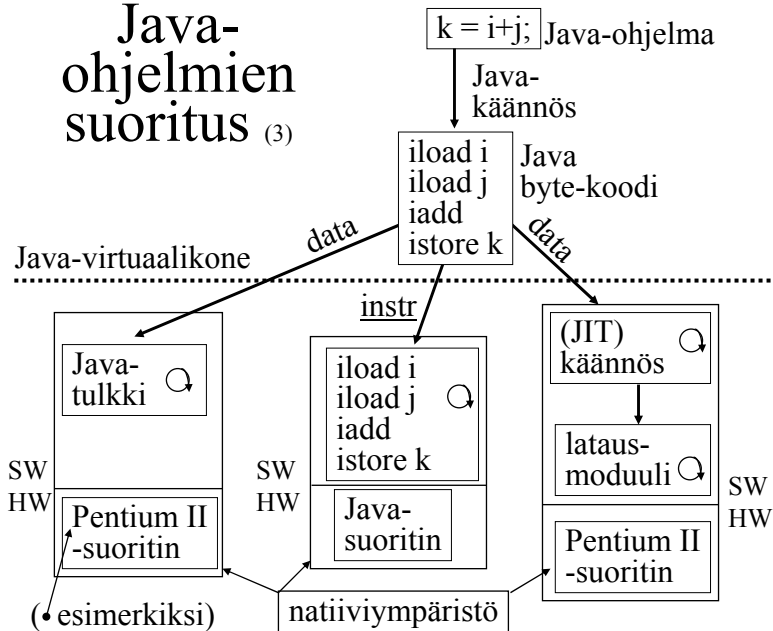
Jakso 11

Tulkinta ja emulointi



Tulkinta ja emulointi
Java-ohjelman suoritus,
tulkinta ja kääntäminen
Suorittimen emulointi:
ttk-91, Crusoe

Java-ohjelmien suoritus (3)



Java-virtuaalikone (JVM) ⁽⁵⁾

- Hypoteettinen suoritin, toteutus eri tavoilla
- Geneerinen, sitä on ”helppo” simuloida kaikilla todellisilla suorittimilla
- Useita säikeitä (thread) voi olla ’samanaikaisesti’ suorituksessa
 - suorittimella mikroaikaskaalassa vain yksi kerrallaan
- Tietorakenteet
 - mm. virtuaalikoneen suorittimen ”rekisterit”
 - luodaan JVM:n käynnistämisen yhteydessä
- Käskyt
 - virtuaalikoneen suorittimen konekäskyt
 - 226 käskyä á 32 bittiä

31/05/2004

Copyright Teemu Kerola, K2003

3

JVM:n tietorakenteet ⁽⁸⁾

- JVM-pino ks. Fig. 4-10 (Tane99)
 - vähän kuten tavallinen AT-pino
 - koostuu useista *kehyksistä* (frames) (vrt. aktivointitietue) ja operandipinosta
 - käyttö: kehyksille ainoastaan push/pop-operaatiot, operandipinon alkioille myös push/pop
 - ei tarvita yhtenäistä muistialuetta
 - allokoidaan keosta (heap)
 - toteutuksesta riippuen rajallinen tai dynaamisesti laajennettavissa
 - tila loppu \Rightarrow StackOverflowError, OutOfMemoryError

<http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html>

31/05/2004

Copyright Teemu Kerola, K2003

4

JVM:n tietorakenteet (jatkuu) ⁽⁶⁾

- JVM-keko (JVM heap)
 - yhteinen kaikille saman virtuaalikoneen säikeille
 - automaattinen roskienkeruu (garbage collector)
 - ei-käytössä oleva (eli vapautettu) muistialue palautetaan uusiokäyttöön (vapaaksi)
 - ei tarvita erikseen *free*-operaatiota Java-ohjelmassa
 - voi hidastaa suoritusta milloin vain
 - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
 - ei tarvitse muodostaa yhtenäistä muistialuetta natiivijärjestelmän kasassa
 - tila loppu ⇒ OutOfMemoryError

31/05/2004

Copyright Teemu Kerola, K2003

5

JVM:n tietorakenteet (jatkuu) ⁽⁶⁾

ks. Fig. 4-10 [Tane99]

- JVM-metodialue (JVM Method Area)
 - yhteinen kaikille JVM-säikeille
 - vastaa tavallista kääntäjän tuottamaa koodisegmenttiä
 - loogisesti osa JVM-kekoa
 - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
 - tila loppu ⇒ OutOfMemoryError

31/05/2004

Copyright Teemu Kerola, K2003

6

JVM:n tietorakenteet (jatkuu) (7)

ks. Fig. 4-10 [Tane99]

- Javan suoritusaikainen vakioallas (runtime constant pool)
 - joka luokalle (class) tai liittymälle (interface) omansa
 - suoritusaikainen esitystapa tiedoston *class constant_pool* -taulukolle
 - vastaa vähän tavallista symbolitaulua
 - useita erilaisia vakioita (käännösaikaiset literaalit, suoritusaikana ratkottavat attribuutit, ...)
 - talletetaan JVM-metodialueelle
 - tila loppu \Rightarrow OutOfMemoryError

JVM:n tietorakenteet (jatkuu) (6)

- Natiivimetodien pinot (Native Method Stacks)
 - toteutus voi käyttää tavallisia pinoja ("C stacks") sellaisten natiivimetodien tukena, jota ei ole kirjoitettu Javalla
 - käytetään myös Java-tulkin toteutuksessa
 - ei tarvita JVM-toteutuksissa, joissa ei ole natiivimetoodeja
 - toteutuksesta riippuen kiinteän kokoinen tai dynaamisesti laajennettavissa
 - tila loppu \Rightarrow StackOverflowError, OutOfMemoryError

JVM:n tietorakenteet (jatkuu) ⁽⁶⁾

ks. Fig. 4-10 [Tane99]

- JVM-rekisterit
 - PC osoittaa johonkin JVM-metodialueelle
 - CPP osoittaa vakioaltaaseen
 - LV on paikallisten muuttujien kantaosoite (vähän kuten FP ttk-91:ssä)
 - SP osoittaa JVM-operandipinon huipulle
 - kaikki rekisterit implisiittisiä; niitä ei erikseen nimetä JVM -konekäskyissä

JVM:n tietorakenteet (jatkuu) ⁽⁶⁾

ks. Fig. 4-10 [Tane99]

- JVM- kehys (frame, raami)
 - talletetaan JVM-pinoon, luodaan metodin kutsun yhteydessä, vapautetaan metodista poistuttaessa
 - paikalliset muuttujat
 - parametrit, paluarvon ja välitulokset
 - dynaamisen linkityksen toteutusväline
 - keskeytysten toteutusväline

JVM-kehiksen data ⁽⁸⁾

- Paikalliset muuttujat ks. Fig. 4-10 [Tane99]
sisältävä taulukko
 - viittaukset indeksoituna (0, 1, 2, ...) rekisterin LV suhteen
 - indeksit sanoina
 - kaksi sanaa vaativa muuttuja (long, double) sijoitetaan kahteen peräkkäiseen (32 bittiseen) sanaan
 - big-endian talletus
- Parametrit, paluuarvon ja välitulokset sisältävä operandipino
 - SP osoittaa pinon huipulle
 - pinoarkkitehtuuri (vs. rekisteriarkkitehtuuri)

JVM:n tiedon osoitusmoodit ⁽⁴⁾

- Välitön operandi iINC 2 (34)
- Indeksoitu iINC (2) 34 tehollinen muistiosoite (LV) + 2
- Pino-osoitus iADD korvaa pinon kaksi päällä olevaa kokonaislukua niiden summalla
ks. Fig. 4-10 [Tane99]
- Taulukko-osoitus pinon kautta iALOAD Korvaa pinon pinnalla olevat taulukon alkuosoite ja indeksi k.o. taulukon alkiolla

JVM-käskyt (7)

- Peruslaskutoimitukset
– add, sub, mul, div, rem, neg
- Boolean
– and, or, xor, shl, shr, ushr
- Pinon hallinta
– dup, pop, swap, tauluk. luonti, esitystavan muutokset
- Load/Store
– load, aload, store, astore, push-käskyt
- Vertailut
- Kontrollinsiirrot
- Muut

ks. Fig. 5-36 [Tane99]

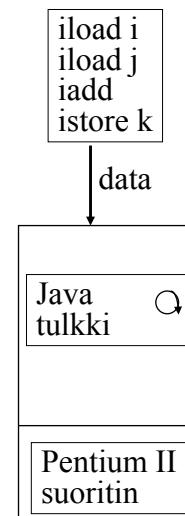
31/05/2004

Copyright Teemu Kerola, K2003

13

Java-tulkki (4)

- Emuloi JVM-konekielen käskyjä (byte-koodia)
- Yksi käsky kerrallaan
- JVM-rekisterit ja muistialueet emuloitu tulkin tietorakenteina
- Hidasta, mutta joustavaa



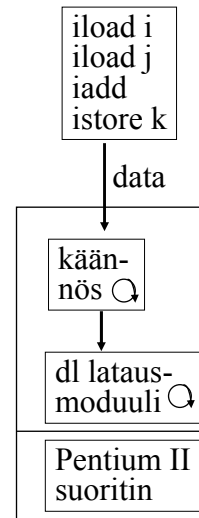
31/05/2004

Copyright Teemu Kerola, K2003

14

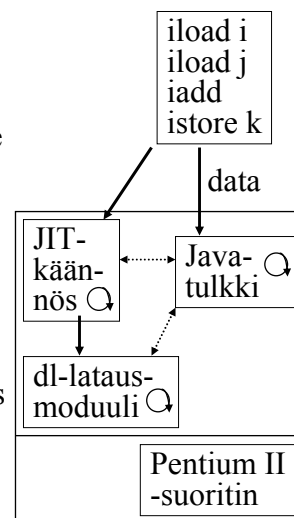
Käännös natiivikoneelle (3)

- (a) Käännetään tavukoodi natiivikoneen konekielelle, ja suoritetaan normaalin ohjelman tapaan
- (b) Käännetään tavukoodi korkean tason kielelle, joka sitten käännetään natiivikoneen konekielelle
 - alkuosa riittää tehdä kerran
 - loppuosa on jo valmiina yleensä
- ongelma: dynaaminen linkitys



Java JIT -käännös (6)

- JIT = Just-in-Time
- Emulointi ja/tai natiivikoneen suoritus tilanteesta riippuen
- Kääntää luokan natiivikonekielelle dynaamisesti linkitettäväksi moduuliksi, juuri ennen luokan metodin kutsua
- Tarvitsee paljon muistia
- Voi hidastaa suoritusta, jos käännökseen menee enemmän aikaa kuin tulkitsemiseen (käännös vasta 2. kutsukerralla?)
- JVM-rekisterit ja muistialueet emuloitu tulkin tietorakenteina, joita natiivikoodi myös käyttää



Java-suoritin: Sun PicoJAVA II ⁽⁴⁾

- Suorittimen määrittely, jonka mukaisessa koneessa byte-koodi -muodossa olevia ohjelmia voidaan sellaisenaan suorittaa
- Valinnainen välimuisti ja liukulukusuoritin
- Kaikki 226 JVM-konekäskyä
 - jotkut käskyt toteutettu aliohjelmilla, jotka aktivoidaan keskeytyskäsittelemekanismin avulla
- Myös 115 muuta konekäskyä käyttöjärjestelmän ja muiden ohjelmointikielten toteuttamiseksi

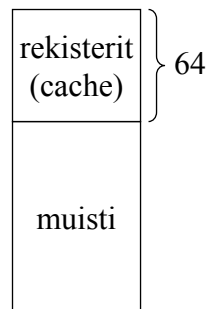
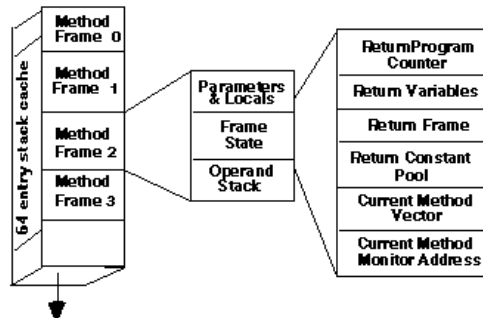
31/05/2004

Copyright Teemu Kerola, K2003

17

PicoJAVA II -pino

- 64 (välimuisti-) rekisteriä JVM-pinon huipun talletukseen
 - loput JVM-pinosta muistissa



Shawn Lauzon,
Survey of the JavaChip

31/05/2004

Copyright Teemu Kerola, K2003

18

PicoJAVA II -rekisterit ⁽¹¹⁾

- 25 rekisteriä á 32 bittiä
 - PC, LV, CPP, SP (pino kasvaa alaspäin)
 - OPLIM alaraja SP:lle; alitus aiheuttaa keskeytyksen
 - FRAME osoittaa paikallisten muuttujataulukon jälkeen talletettuun metodista paluu osoitteeseen
 - PSW
 - rekisteri, joka kertoo pinon välimuistirekistereiden tämänhetkisen käytön
 - 4 rekisteriä keskeytysten ja break-point'ien käsittelyyn
 - 4 rekisteriä säikeiden hallintaan
 - 4 rekisteriä C ja C++ ohjelmien toteutukseen
 - 2 rajarekisteriä sallitun muistialueen rajoittamiseen
 - suorittimen version numero ja konfiguraatiorekisterit

31/05/2004

Copyright Teemu Kerola, K2003

19

PicoJAVA:n ylim. käskyt ⁽⁵⁾

- Read/write ylimääräisille rekistereille
- Osoittimien manipulointikäskyt
 - mitä tahansa muistialuetta voidaan suoraan lukea/kirjoittaa
 - tarvitaan C/C++ varten
- C/C++ -aliohjelmien kutsu ja paluukäskyt
- Natiivi HW-manipulointi
 - tyhjännä välimuisti (osittain? kokonaan?), ...
- Muut käskyt
 - power on/off, ...

31/05/2004

Copyright Teemu Kerola, K2003

20

PicoJAVA-toteutuksia (2)

- Sun microJAVA 701
 - valinnainen välimuisti
 - oma muistiväylä
 - PCI-väylä muille laitteille
 - 16 ohjelmoitavaa I/O-johdinta
 - näppäimet, LEDit, ...
 - 3 ohjelmoitavaa ajastinta (⇒ kellolaitekeskeytykset)
 - suunnattu halpoihin kannettaviin laitteisiin (kämmenmikro, PDA - Personal Digital Assistant)
- Sun ultraJAVA
 - nopeampi, parempi, kalliimpi, ...
 - suunnattu grafiikka- ja multimediasovelluksiin

Muita Java-suorittimia

- JEM (Rockwell Collins)
- PSC1000 (Patriot Scientific)
 - dSys (Saksa), lääketieteellisiä laitteita
- MJ501 (LG Semicon)
 - TV, älykortit
- JSR001, Real-Time Specification for Java (Java Community Process, ”Sun Microsystems”)
 - aJile: aJ-80, aJ-100, älykkäät liikkuvat laitteet

Sun MAJC

- MAJC - Microprocessor Architecture for Java Computing
 - suoritinarkkitehtuurin määrittely
 - tavoitteena suuri nopeus Java-, C- ja C++-sovelluksille
 - suunnattu multimediasovelluksiin verkossa
 - tukee hyvin JIT-käännöstä

MAJC toteutus: MAJC 5200 ⁽⁴⁾

- 1-4 suoritinta (2 suorittimen lastu v.1999)
- Useiden (peräkkäin kutsuttavien) metodien samanaikainen suoritus eri suorittimilla
 - ennakoiva (speculative) suoritukselle oma kasa
 - peruutus (rollback), jos ennakoitu suoritus meni pieleen
- 4 säiettä suorituksessa per suoritin
 - säikeen vaihto nopeampaa kuin muistista luku!
 - laiterekisterit 4:lle säikeelle!
 - välimuistin hudin aikana suoritetaan muita säikeitä
- Suunnattu interaktiiviseen TV:hen, virtuaalitodellisuussovelluksiin, ...

TTK-91-emulointi (5)

- TTK-91-konekielen emulointi
- KOKSI-simulaattorin osa
- Yksi käsky kerrallaan
- TTK-91-koneen rekisterit ja muisti emuloitu tulkin tietorakenteina

ks. simulaattorin koodi

```
load R1, 234  
add R1, =5  
mul R1, R2
```

data

TTK-91
emulaat-
tori

Pentium II
suoritin

Transmetan Crusoe-suoritin (8)

- x86-konekielen emulointi, JIT-käännös
- Natiivikäskykanta ei ole julkistettu
- ”nopeampi, sama teknologia”?
- ”yhtä nopea, vähemmän virtaa”
- Monta x86-käskyä yhtäaikaan, sikin sokin emuloinnissa
- x86-rekisterit emuloitu natiivijärjestelmän laiterekistereillä
- x86-muisti emuloitu suojattuna tietorakenteina
- Tarkat keskeytykset:
 - suorituksen peruutus
 - uusi, hidas JIT-käännös
 - hidas mutta tarkka emulointi

```
movl %esp, %ebp  
subl $4, %esp  
pushl %eax
```

data

Pää-
ohjelma

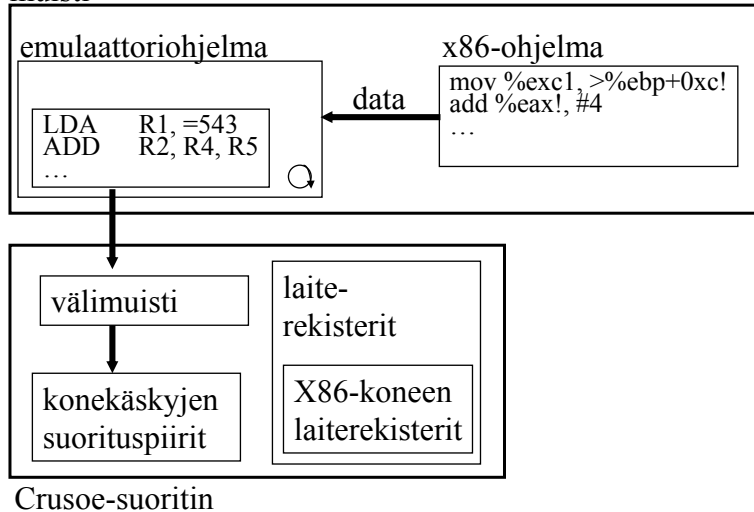
JIT-
kään-
nös

DI lataus-
moduuli

Crusoen
suoritin

Crusoe-emulaattorin suoritus

muisti

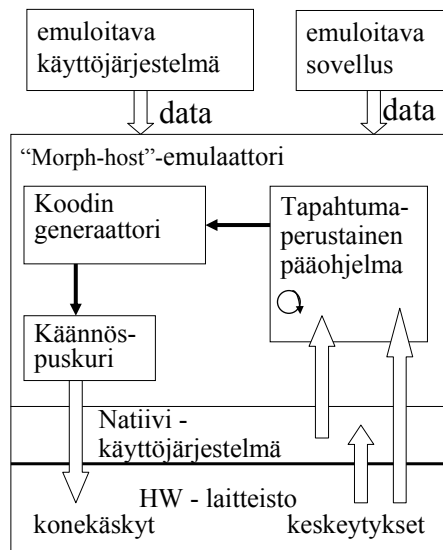


31/05/2004

Copyright Teemu Kerola, K2003

27

Crusoe-suorittimen looginen rakenne

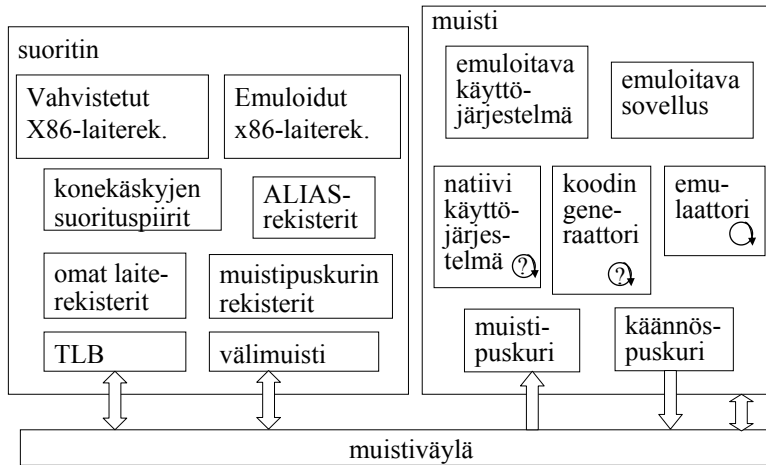


31/05/2004

Copyright Teemu Kerola, K2003

28

Crusoe-suorittimen fyysinen rakenne

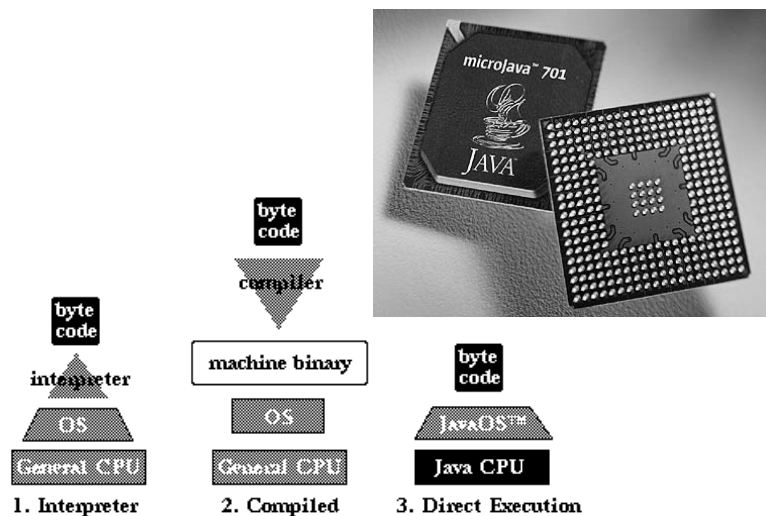


31/05/2004

Copyright Teemu Kerola, K2003

29

-- Jakson 11 loppu --



31/05/2004

Copyright Teemu Kerola, K2003

30