

jBACI Concurrency Simulator

Helsingin yliopisto

Tietojenkäsittelytieteen laitos

Rinnakkaisohjelmointi syksy 2008

Projekti C

Opintopiiri: Sanna Häkkinen, Mikko S. Niemelä, Antti Savolainen

1. Ohjelman tarkoitus

jBACI Concurrency Simulator on M. Ben-Arin kehittämä rinnakkaisuussimulaattori. jBACI eroaa BACI:sta (Ben-Ari Concurrent Interpreter) siinä, että se on toteutettu toimimaan Java-virtuaalikoneessa. Simulaattorin avulla on helppo havainnollistaa rinnakkaisessa laskennassa väistämättä eteen tulevia ongelmia. jBACI koostuu IDE:stä (Integrated Development Environment), jossa ohjelmia voidaan editoida, ja varten vasten luodusta suoritusympäristöstä, jossa ohjelmat voidaan suorittaa. Ohjelmien kirjoittamiseen käytetään C-kielestä kehitettyä C--:sta. Vaihtoehtoisesti koodaamiseen voi käyttää myös Pascalista jatkokehitettyä versiota.

2. Ohjelman asennus

Ohjelman asennus Windowsiin ja Linux-pohjaisiin käyttöjärjestelmiin eroaa hieman toisistaan, vaikka sovellus onkin toteutettu Javalla. Toimiakseen jBACI tarvitsee Java-version 1.4 (SDK tai JRE). Saatavilla Sunin sivuilta:

<http://java.sun.com/javase/downloads/index.jsp>

2.1 Asennus Linuxiin

Seuraavat asennusohjeet toimivat ainakin Helsingin yliopiston tietojenkäsittelytieteen laitoksen omassa Linux-distribuutiossa.

1. Lataa [jbaci1-4-5.zip](#) (esimerkiksi: täältä http://stwww.weizmann.ac.il/g-cs/benari/jbaci/jbaci1-4_5.zip).

2. Pura zip-tiedosto haluamaasi kansioon
(esimerkiksi /fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5).
3. Lataa balnxxe-2007Nov25.tar.gz -tiedosto (osoite:
http://www.mines.edu/fs_home/tcamp/baci/balnxxe-2007Nov25.tar.gz)
(paketti sisältää kääntäjät suoritusympäristöä varten).
4. Pura balnxxe-2007Nov25.tar.gz-tiedoston sisältö jbaci1-4-5/bin -kansioon
(esim. /fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/bin).
5. Avaa config.cfg -tiedosto
(sijainti: /fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/config.cfg) jollakin
tekstieditorilla.
6. Seuraavaksi config.cfg-tiedostoon täytyy muuttaa c- ja pascal -kääntäjien
polut oikeiksi.
Riville 9 lisätään oikea polku kansioon, joka sisältää lähdekoodiesimerkit
(esim. /fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/examples).
Riville 13 tulee polku pascal-kääntäjään
(esim. PASCAL_COMPILER=/fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/bin/bapas).
Riville 23 polku c-kääntäjään
(esim. C_COMPILER=/fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/bin/bacc).

Muutosten jälkeen **config.cfg**-tiedoston pitäisi näyttää tältä:

```
#jBACI configuration file
#Mon Mar 08 07:08:24 GMT 2004
SHOW_HISTORY_SOURCE=true
SUBWINDOW_Y=300
SUBWINDOW_X=300
INITIAL_CONSOLE=true
INITIAL_LINDA=true
G_YPOS=100
SOURCE_DIRECTORY=/fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/examples
G_WIDTH=600
GLOBALS_Y=350
GLOBALS_X=300
PASCAL_COMPILER=/fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/bin/bapas
SHOW_ACTIVE_DEFAULT=true
OPEN_MAIN_BY_DEFAULT=true
FRAME_Y=600
PTAB_Y=300
FRAME_X=800
G_HEIGHT=450
CONSOLE_Y=350
PTAB_X=150
```

CONSOLE_X=0

C_COMPILER=/fs-3/f/ktunnus/Kurssit/Rio/jbaci1-4-5/bin/bacc(ja niin edelleen)...

Nyt ohjelman voi ajaa kirjoittamalla komentoriville "java -jar jbaci.jar" siinä hakemistossa, johon jbaci1-4-5.zip on purettu

(esim. **/fs-3/f/awsavola/Kurssit/Rinnakkaisohjelmointi/jbaci1-4-5/**)

2.2 Asennus Windowsiin

Seuraavat ohjeet auttavat jBACI:n asentamisessa Windows XP Home Edition -käyttöjärjestelmään.

1. Lataa [jbaci1-4-5.zip](http://stwww.weizmann.ac.il/g-cs/benari/jbaci/jbaci1-4-5.zip) (esimerkiksi: täältä <http://stwww.weizmann.ac.il/g-cs/benari/jbaci/jbaci1-4-5.zip>).
2. Pura zip-tiedosto haluamaasi kansioon (esim. **C:\jbaci1-4-5**)
3. Avaa jbaci1-4-5 -kansiossa oleva **config.cfg**-tiedosto jollakin tekstieditorilla (esim. Notepad)
4. Seuraavaksi config.cfg-tiedostoon täytyy muuttaa c- ja pascal -kääntäjien polut oikeiksi.
Riville 9 lisätään oikea polku kansioon, joka sisältää lähdekoodiesimerkit (esim. **C:\jbaci1-4-5\examples**).
Riville 13 tulee polku pascal-kääntäjään (esim. **PASCAL_COMPILER=C:\jbaci1-4-5\bin\bapas.exe**).
Riville 23 polku c-kääntäjään (esim. **C_COMPILER=C:\jbaci1-4-5\bin\bacc.exe**).

Huomioi, että polkunimissä olevat "\"-merkit täytyy olla kahteen kertaan, jotta ohjelma osaa lukea polun oikein.

Muutosten jälkeen **config.cfg**-tiedoston pitäisi näyttää tältä:

```
#jBACI configuration file
#Mon Mar 08 07:08:24 GMT 2004
SHOW_HISTORY_SOURCE=true
SUBWINDOW_Y=300
SUBWINDOW_X=300
INITIAL_CONSOLE=true
INITIAL_LINDA=true
G_YPOS=100
```

```
SOURCE_DIRECTORY=C:\\jbaci1-4-5\\examples  
G_WIDTH=600  
GLOBALS_Y=350  
GLOBALS_X=300  
PASCAL_COMPILER=C:\\jbaci1-4-5\\bin\\bapas.exe  
SHOW_ACTIVE_DEFAULT=true  
OPEN_MAIN_BY_DEFAULT=true  
FRAME_Y=600  
PTAB_Y=300  
FRAME_X=800  
G_HEIGHT=450  
CONSOLE_Y=350  
PTAB_X=150  
CONSOLE_X=0  
C_COMPILER=C:\\jbaci1-4-5\\bin\\bacc.exe (ja niin edelleen)...
```

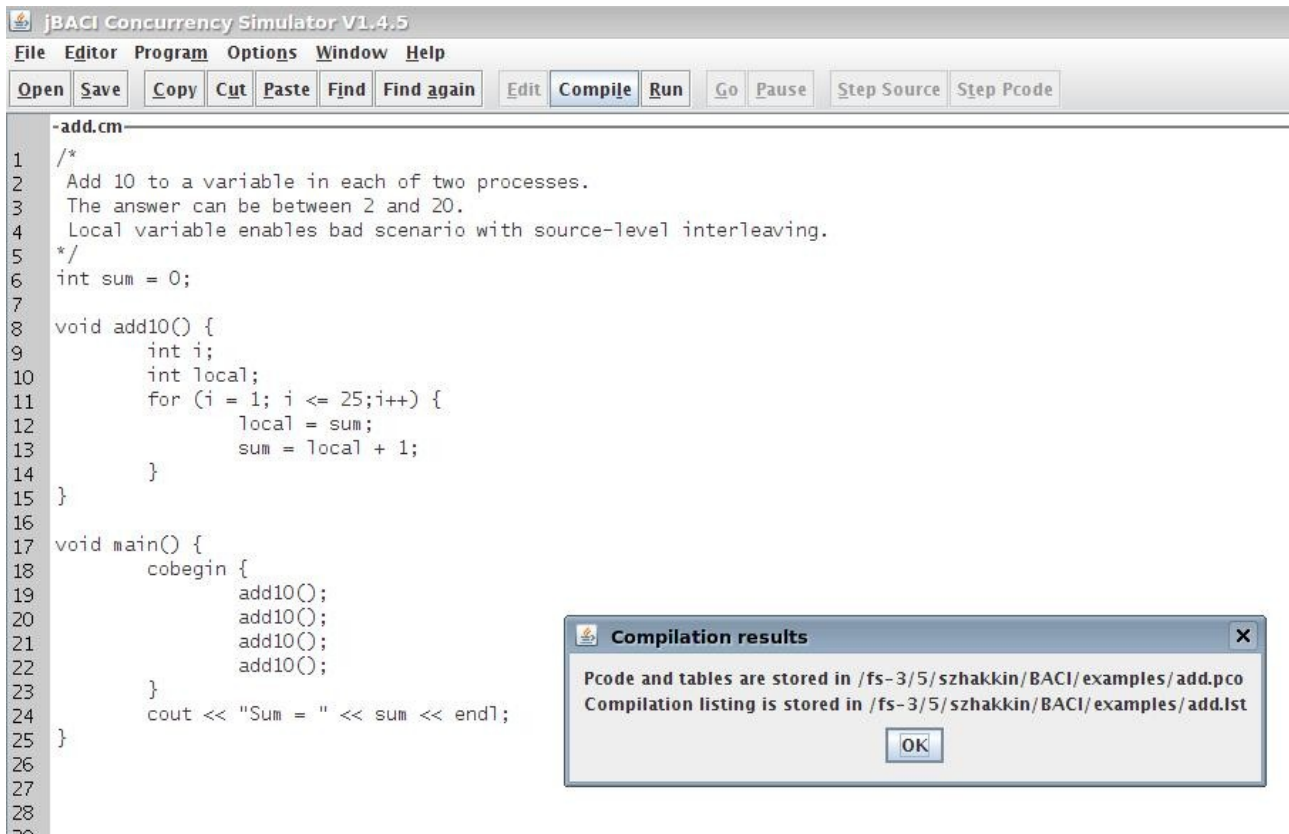
Nyt ohjelman voi käynnistää tuplaklikkaamalla **jbaci** jar-kuvaketta.

3 Ohjelman käyttö

3.1 Editointitila

Kun jBACI käynnistetään, se avautuu suoraan editointitilaan. Tässä tilassa voidaan muokata lähdekooditiedostoja, kuten missä tahansa tekstieditorissa. Ohjelman toimintaa voi helposti kokeilla avaamalla jonkin **/jbaci1-4-5/Examples** -kansiossa olevan esimerkkilähdekooditiedoston. Kansion pitäisi avautua suoraan näkyville klikkaamalla *Open*-painiketta (edellyttäen, että polku kansioon on oikein **config.cfg** -tiedostossa). On huomioita, että lähdekooditiedostojen täytyy olla .cm- (C--) tai .pm-loppuisia (Pascal), jotta ohjelma tunnistaa ne.

Jotta ohjelma voidaan ajaa, on se luonnollisesti käännettävä BACI:n ymmärtämäksi pcode-tavukoodiksi. Tämä tapahtuu painamalla editointitilassa *Compile*-painiketta. Mikäli käynnös onnistuu ilman virheitä, ohjelma tulostaa käyttäjälle suorituskoodin tallennuspaikan pop-up -ikkunassa.

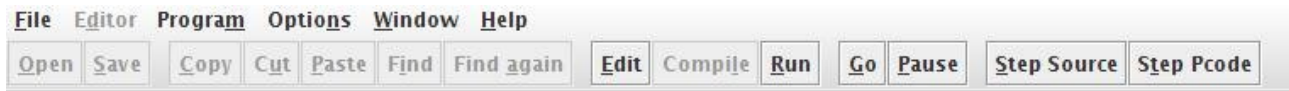


Kuva 1: jBaci tilassa, jossa lähdekoodi on juuri käännetty. Ohjelma ilmoittaa, että käännös on onnistunut.

Kun ohjelma on onnistuneesti käännetty, voidaan siirtyä suoritustilaan painamalla *Run*-nappia.

3.2 Suoritustila

Kun ohjelma on käännetty oikein, siirrytään suoritustilaan. Suoritustilassa toimivat seuraavat valikot.



Kuva 2: Suoritustilan valikko-näkymä.

File-valikko:

Sisältää toimintoja, joita voi käyttää editointitilassa kuten copy ja paste. Suoritustilassa toimii vain exit.

Program-valikko:

Tästä valikosta löytyvät toiminnot editointi- ja suoritustilojen käynnistämiseen. Samat toiminnot on kuitenkin tuotu toiminnan helpottamiseksi ikkunan yläreunaan.

Edit siirtyy takaisin lähdekoodin editointitilaan

Compile kääntää lähdekoodin.

Run siirtyy suoritustilaan.

Go aloittaa ohjelman suorituksen

Pause pysäyttää ohjelman suorituksen

Step source suorittaa lähdekoodia askel kerrallaan

Step pcode suorittaa konekäskyjä askel kerrallaan

Options-valikko

Options valikosta voi muokata ohjelman suoritusaikaisia asetuksia.

Pause on Process Swap:in ollessa valittuna, ohjelman suoritus pysähtyy aina automaattisesti, kun prosessia vaihdetaan.

Show Active Window:n ollessa valittuna, jokaiselle prosessille avautuu oma ikkuna kun prosessi tulee suoritukseen ensimmäistä kertaa. Aktiivisen prosessin ikkuna on päällimmäisenä koodia suorittaessa askel kerrallaan.

History of Source Steps:in ollessa valittuna, näytetään historia-ikkunassa vain lähdekoodin historia. Muulloin näytetään sekä lähdekoodin että konekäskyjen historia.

Write History File:in ollessa valittuna, talletetaan koko suoritettujen lähdekoodin tai kaikkien suoritettujen konekäskyjen historia tiedostoksi. *his* -päätteinen tiedosto talletetaan samaan kansioon kuin alkuperäinen lähdekoodi.

Window

Window valikkosta saadaan näkyviin eri ikkunat.

Console, johon ohjelma tulostaa tietoa.

Globals, jossa näkyy ohjelman globaalit muuttujat ja niiden muuttuvat arvot ohjelman suorituksen aikana.

History konsoli näyttää, riipuen options-valikon asetuksista, lähdekoodin ja konekäskyjen historian, tai pelkästään lähdekoodin historian.

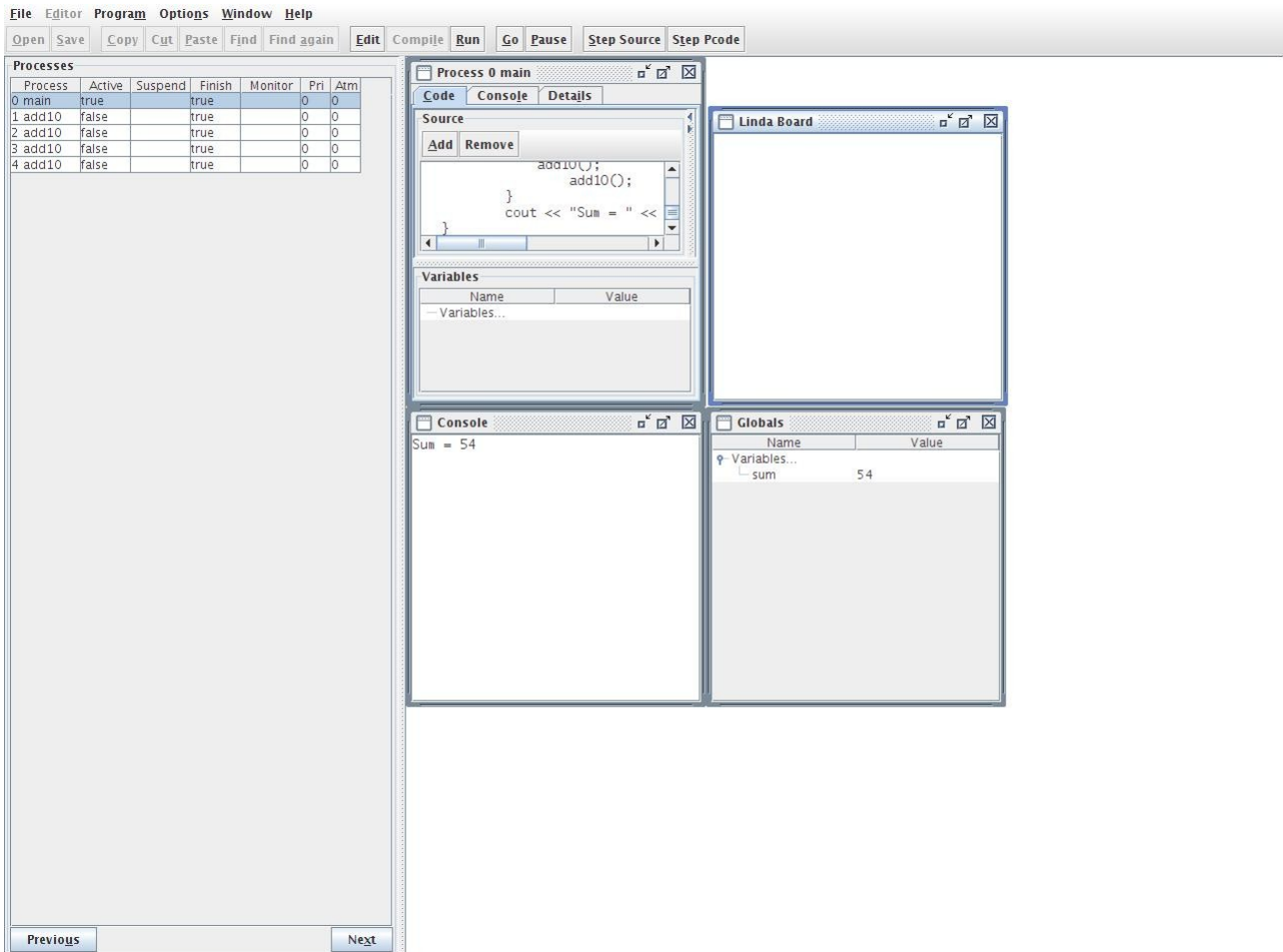
Linda Board on jBACIn laajennus, jonka on kehittänyt David Gelernter. Tätä ikkunaan ei tarvita Rinnakkaisohjelmointi -kurssilla, jota varten tämä ohje on tehty. Lisätietoja Lindasta löytyy seuraavasta linkistä, josta voit ladata jBACI viralliset ohjeet:

<http://stwww.weizmann.ac.il/g-cs/benari/jbaci/jbaci1-4-5docs.zip>

Suoritustilan ikkunat

Suoritustilan ikkuna on jaettu kahteen osaan. Vasemmalla puolella on tiedot kaikista prosesseista ja missä tilassa ne ovat. Oikealla puolella on ikkunat, jotka

kertovat ohjelman suorituksesta. Process-ikkunoita voi olla yksi, joka näyttää sillä hetkellä suorituksessa olevan prosessin lähdekoodia, tai jokaisella prosessilla voi olla oma ikkunansa. Päällimmäisenä näkyy sen prosessin ikkuna joka on suorituksessa. Ikkunoiden määrä säädetään Options-valikosta (ks. yllä).



Kuva 3: jBaci suoritustilassa

Prosess-ikkuna on jBACIn ydin. Se sisältää kolme alavalikkoa: Code, Console ja Details.

- *Code-valikko* näyttää ohjelman koodin.
- *Source* -osassa näytetään ohjelman koko lähdekoodi, ei vain yhden prosessin koodia.
- *Pcode* -osassa näytetään ohjelman konekäskyt, mutta vain valitulta lähdekoodin riviltä, eikä siis koko lähdekoodista. Pcode-osa näkyy ainoastaan, jos Prosess-ikkunan kokoa kasvattaa manuaalisesti.

- *Variables* -osassa näytetään prosessin paikallisten muuttujien arvot.
- *Console-valikko* näyttää, mitä juuri kyseinen prosessi tulostaa, toisin kuin erillinen Console-ikkuna (ks. kohta 3.2.2), jossa näkyy koko ohjelman tulostus.
- *Details-ikkuna* näyttää prosessipinojen sisällön sekä prosessien tilan.

Semaforien käyttö BACI:ssa

Semafori on määritelty BACI C-- kieleen positiiviseksi int-muuttujaksi.

Semaforin käyttö on rajoitettu. Semaforilla on myös aliluokka binary sem, joka käsittää vain arvot 0 ja 1.

Semaforin api-kuvaus sisältää seuraavia funktioita:

binary sem = 1 # alustaa binääri semaforin

semaphore sem = 4 #alustaa yleisen semaforin

P(sem) tai wait(sem) Jos binääri semaforin int-arvo on suurempi kuin 0, sitä vähennetään yhdellä ja kyseinen prosessi jatkaa.

Jos taas semaforin arvo on 0, prosessi lisätään nukkuvien prosessien joukkoon.

V(sem) tai signal(sem) Jos semafori on nolla ja prosessi/prosesseja on nukkumassa, herätetään yksi prosessi sattumanvaraisesti. Jos kukaan ei odota semaforia, lisätään int-arvoa yhdellä. Kyseinen suorittaja saa jatkaa.

Semaforien määrittelyt jBACI:ssa

```
1
2
3 semaphore count;    // a "general" semaphore
4 binarysem output;  // a binary (0 or 1) semaphore for unscrambling output
5
6 void increment()
7 {
8     p(output);      // obtain exclusive access to standard output
9     cout << "before v(count) value of count is " << count << endl;
10    v(output);
11    v(count);       // increment the semaphore
12 }
13
14 void decrement()
15 { p(output);      // obtain exclusive access to standard output
16
17     cout << "before p(count) value of count is " << count << endl;
18     v(output);
19     p(count);      // decrement the semaphore (or stop -- see manual text)
20 }
21
22 main()
23 {
24     initialisesem(count,0);
25     initialisesem(output,1);
26     cobegin {
27         decrement(); increment();
28     }
29 }
```

Esimerkki – SemPlusMinus.cm

```
int x1 = 0;
int x2 = 0;
int x3 = 0;
int x4 = 0;
int ctrl = 0;
int sum = 0;

binarysem sem;

void processP() {
    int i;
    for(i=0; i<50; i++) {
        /*Critical section: */
        wait(sem); /*Wait*/
        if(ctrl%2!=0) {
            x1=x1-1;
            x2=x2-1;
            x3=x3-1;
            x4=x4-1;
        } else {
            x1=x1+2;
            x2=x2+2;
            x3=x3+2;
            x4=x4+2;
        }
        ++ctrl;
        sum=x1+x2+x3+x4;
        /* Signal: */
        signal(sem); /*Signal*/
    }
}

void main() {
    initialisesem(sem, 1);
    cobegin {
        processP();
    }
    cout << "sum = " << sum << endl;
    cout << "ctrl = " << ctrl << endl;
    cout << "x1 = " << x1 << endl;
    cout << "x2 = " << x2 << endl;
    cout << "x3 = " << x3 << endl;
    cout << "x4 = " << x4 << endl;
}
```

Lähteet:

BACI – Käyttöohje (2007 - Joensuu, Hoppula, Kajander)

http://www.cs.helsinki.fi/u/kerola/rio/ohjeet/BACI/s07_joens_hopp_kaj.pdf

Rinnakkaisohjelmointi syksy 2008 aikataulusivu (2008 – Kerola)

<http://www.cs.helsinki.fi/u/kerola/rio/s2008/aikataulu.html>

jBACI Concurrency Simulator User's Guide (M. Ben-Ari)

<http://www.cs.helsinki.fi/u/kerola/rio/BACI/jbaci.pdf>

USER'S GUIDE BACI C-- Compiler and Concurrent PCODE Interpreter (2002 - Bill Bynum/Tracy Camp)

<http://www.cs.helsinki.fi/u/kerola/rio/BACI/baci-c.pdf>