Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №6**

**З предмету «Паралельні та розподілені обчислення»**

**Java**

Виконав:

Студент  
IІІ курсу ФІОТ  
групи ІО-12  
Бута С. О.

Залікова книжка №1205

Київ-2014

**Техническое задание**

1. Структура ПКС с ОП: 
2. Задача: .
3. Язык программирования: Java.
4. Средства взаимодействия задач: мониторы.

**Выполнение работы**

1. Разработка параллельного математического алгоритма.
2. ei = min(ZH);
3. e = min(e, ei);
4. AH = e\*BH – l\*C\*(MO\*MKH); OP: e, l, C, MO;
5. Разработка алгоритмов процессов.

**Задача Т1:**

1. Ввод Z, l
2. Сигнал Тj (j = 1..8, j ≠ 1) о завершении ввода Sj;1
3. Ждать завершение ввода в Т2, Т4 W2,4;1
4. Счёт1
   1. e1 = min(ZH)
5. Счет2 КУ
   1. e = min(e, e1);
6. Сигнал Тj (j = 1..8, j ≠ 1)о завершении Cчёт2 Sj;2
7. Ждать завершение Счёт2 в Тj (j = 1..8, j ≠ 1) Wj;1
8. Копии КУ
   1. l1 = l;
   2. e1 = e;
   3. C1 = C;
   4. MO1 = MO;
9. Счёт3
   1. AH = e\*BH – l\*C\*(MO\*MKH)
10. Ждать завершение Счёт3 в Тj (j = 1..8, j ≠ 1) Wj;2
11. Вывод А

**Задача Т2:**

1. Ввод MO, MK
2. Сигнал Тj (j = 1..8, j ≠ 2)о завершении ввода Sj;1
3. Ждать завершение ввода в Т1, Т4 W1,4;1
4. Счёт1
   1. e1 = min(ZH)
5. Счет2 КУ
   1. e = min(e, e1);
6. Сигнал Тj (j = 1..8, j ≠ 2)о завершении Cчёт2 Sj;2
7. Ждать завершение Счёт2 в Тj (j = 1..8, j ≠ 2) Wj;1
8. Копии КУ
   1. l2 = l;
   2. e2 = e;
   3. C2 = C;
   4. MO2 = MO;
9. Счёт3
   1. AH = e\*BH – l\*C\*(MO\*MKH)
10. Сигнал Т1 о завершении Cчёт3 S1;2

**Задача Т3:**

1. Ждать завершение ввода в Т1, Т2, Т4 W1,2,4;1
2. Счёт1
   1. e1 = min(ZH)
3. Счет2 КУ
   1. e = min(e, e1);
4. Сигнал Тj (j = 1..8, j ≠ 2)о завершении Cчёт2 Sj;2
5. Ждать завершение Счёт2 в Тj (j = 1..8, j ≠ 2) Wj;1
6. Копии КУ
   1. l3 = l;
   2. e3 = e;
   3. C3 = C;
   4. MO3 = MO;
7. Счёт3
   1. AH = e\*BH – l\*C\*(MO\*MKH)
8. Сигнал Т1 о завершении Cчёт3 S1;3

**Задача Т4:**

1. Ввод В, С
2. Сигнал Тj (j = 1..8, j ≠ 2)о завершении ввода Sj;1
3. Ждать завершение ввода в Т1, Т2 W1,2;1
4. Счёт1
   1. e1 = min(ZH)
5. Счет2 КУ
   1. e = min(e, e1);
6. Сигнал Тj (j = 1..8, j ≠ 2) о завершении Cчёт2 Sj;2
7. Ждать завершение Счёт2 в Тj (j = 1..8, j ≠ 2) Wj;1
8. Копии КУ
   1. l4 = l;
   2. e4 = e;
   3. C4 = C;
   4. MO4 = MO;
9. Счёт3
   1. AH = e\*BH – l\*C\*(MO\*MKH)
10. Сигнал Т1 о завершении Cчёт3 S1;4

**Задача Тi (i = 5..8):**

1. Ждать завершение ввода в Т1, Т2, Т4 W1,2,4;1
2. Счёт1
   1. e1 = min(ZH)
3. Счет2 КУ
   1. e = min(e, e1);
4. Сигнал Тj (j = 1..8, j ≠ i)о завершении Cчёт2 Sj;2
5. Ждать завершение Счёт2 в Тj (j = 1..8, j ≠ i) Wj;1
6. Копии КУ
   1. l3 = l;
   2. e3 = e;
   3. C3 = C;
   4. MO3 = MO;
7. Счёт3
   1. AH = e\*BH – l\*C\*(MO\*MKH)
8. Сигнал Т1 о завершении Cчёт3 S1;3

**Разработка схемы взаимодействия процессов:**



**Разработка программы.**

**Листинг**:

**Файл Runner.java:**

import java.util.Arrays;

/\*\*

\* -----------------------------------------

\* Лабораторна робота №6

\* Бута С.О.

\* Задача: A=min(Z)\*B-l\*C\*(MO\*MK)

\* Дата: 29.05.14

\* -----------------------------------------

\*/

public class Runner {

public static void main (String[] args){

final int N = 8;

final int P = 8;

final int H = N/P;

final int FILLER = 1;

final Vector A = new Vector(N),

B = new Vector(N),

C = new Vector(N),

Z = new Vector(N);

final Matrix MO = new Matrix(N),

MK = new Matrix(N);

final ResourceMonitor rm = new ResourceMonitor(N);

final SynchronizeMonitor sm = new SynchronizeMonitor();

class ThreadTask implements Runnable {

int threadNumber;

public ThreadTask(int tn){

threadNumber = tn;

}

@Override

public void run() {

System.out.println("Thread " + threadNumber + " started.");

switch(threadNumber){

case 0: {

//Ввод Z, l

Z.input();

int l = FILLER;

rm.writeL(l);

sm.signalInput();

break;

}

case 1: {

//Ввод MO, MK

MO.input();

rm.writeMO(MO);

MK.input();

sm.signalInput();

break;

}

case 3:{

//Ввод В, С

B.input();

C.input();

rm.writeC(C);

sm.signalInput();

break;

}

default:break;

}

//Ждать завершение ввода в Т1, Т2, Т4

sm.waitInput();

//Счёт1

int ei = Z.min(threadNumber\*H, threadNumber\*H + H);

//Счёт2

rm.calculateE(ei);

//Сигнал Тj о завершении Cчёт2

sm.signalCalculate2();

//Ждать завершение Счёт2

sm.waitCalculate2();

//Копии

int li = rm.copyL();

ei = rm.copyE();

Vector Ci = rm.copyC();

Matrix MOi = rm.copyMO();

//Счёт3

Matrix res = new Matrix(N);

int sum;

int from = threadNumber\*H;

int to = from+H;

for(int i = from; i < to; i++){

for(int j = 0; j < N; j++){

sum = 0;

for(int k = 0; k < N; k++){

sum += MOi.matrix[j][k]\*MK.matrix[k][i];

}

res.matrix[j][i] = sum;

}

}

for(int i =from; i < to; i++){

A.vector[i] = 0;

for(int j = 0; j < N; j++){

A.vector[i] += res.matrix[j][i]\*Ci.vector[i];

}

A.vector[i] = B.vector[i]\*ei - li\*A.vector[i];

}

if(threadNumber == 0){

//Ждать завершение Счёт3 в Тj (j = 1..8, j ≠ 1)

sm.waitCalculate3();

//Вывод А

if(N < 17){

System.out.println(Arrays.toString(A.vector));

}

} else{

//Сигнал Т1 о завершении Cчёт3

sm.signalCalculate3();

}

System.out.println("Thread " + threadNumber + " finished.");

}

}

Thread[] threads = new Thread[P];

for(int i = 0; i < P; i++){

threads[i] = new Thread(new ThreadTask(i));

threads[i].start();

}

try {

threads[0].join();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

**Файл ResourceMonitor.java:**

import java.util.Arrays;

public class ResourceMonitor {

private int N;

private int l;

private int e = Integer.MAX\_VALUE;

private Vector C;

private Matrix MO;

public ResourceMonitor(int n){

N = n;

}

public synchronized void writeC(Vector v){

C = v;

}

public synchronized void writeMO(Matrix m){

MO = m;

}

public synchronized void writeL(int l){

this.l = l;

}

public synchronized void calculateE(int e){

this.e = Math.min(this.e, e);

}

public synchronized Vector copyC(){

Vector res = new Vector(N);

res.vector = Arrays.copyOf(C.vector, N);

return res;

}

public synchronized Matrix copyMO(){

Matrix res = new Matrix(N);

for(int i = 0; i < N; i++)

{

res.matrix[i] = Arrays.copyOf(MO.matrix[i], N);

}

return res;

}

public synchronized int copyE(){

return e;

}

public synchronized int copyL(){

return l;

}

}

**Файл ResourceMonitor.java:**

public class SynchronizeMonitor {

private int flagInput = 0;

private int flagCalculate2 = 0;

private int flagCalculate3 = 0;

public synchronized void signalInput(){

flagInput++;

notifyAll();

}

public synchronized void signalCalculate2(){

flagCalculate2++;

notifyAll();

}

public synchronized void signalCalculate3(){

flagCalculate3++;

notifyAll();

}

public synchronized void waitInput(){

while(flagInput != 3){

try{

wait();

}catch (InterruptedException e){

e.printStackTrace();

}

}

}

public synchronized void waitCalculate2(){

while(flagCalculate2 != 8){

try{

wait();

}catch (InterruptedException e){

e.printStackTrace();

}

}

}

public synchronized void waitCalculate3(){

while(flagCalculate3 != 7){

try{

wait();

}catch (InterruptedException e){

e.printStackTrace();

}

}

}

}