Лабораторная работа №5

по курсу:

«Паралельные и распределённые вычисления»

Тема: «Java. Мониторы»

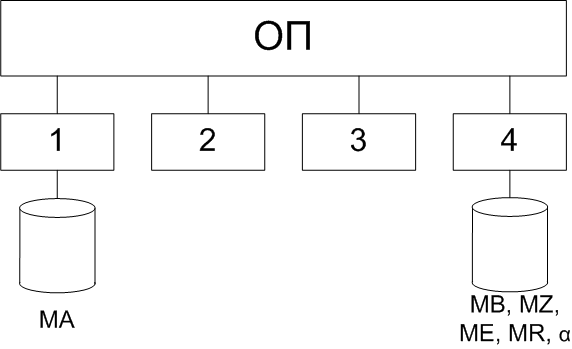
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

MA=(MB \* MZ \* α) + ME \* MR



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. MAH = (MBH \* MZ \* α) + MEH \* MR

**Общие ресурсы:** α, MZ, MR

**Этап 2. Разработка алгоритмов процессов (задач)**

**№ Т1**

1. Ожидание ввода в задаче Т4
2. Копирование α1:=α, MZ1:=MZ, MR1:= MR
3. Счёт MAH =(MBH\*MZ1\*α1) + MEH\*MR1
4. Ожидание окончания счёта от Т2, Т3, Т4
5. Вывод МА

**№ Т2**

1. Ожидание ввода в задаче Т4
2. Копирование α2:=α, MZ2:=MZ, MR2:=MR
3. Счёт MAH = (MBH\*MZ2\*α2) + MEH\*MR2
4. Сигнал задаче Т1 о конце счёта

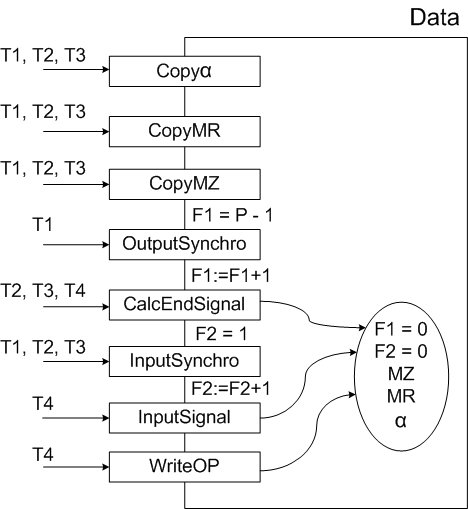
**№ Т3**

1. Ожидание ввода в задаче Т4
2. Копирование α3:=α, MZ3:=MZ, MR3:=MR
3. Счёт MAH = (MBH\*MZ3\*α3) + MEH\*MR3
4. Сигнал задаче Т1 о конце счёта

**№ Т4**

1. Ввод MB, MZ, ME, MR, α
2. Копирование α4:=α, MZ4:=MZ, MR4:=MR
3. Cигнал задачам Т1, Т2, Т3 о завершении ввода
4. Счёт MAH = (MBH\*MZ4\*α4) + MEH\*MR4
5. Сигнал задаче Т1 о конце счёта

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

//МОНИТОР//////////////////////////////////////////////////////////////////////////////

**public** **class** Data {

**private** **int** F1, F2;

**private** **int** [][] MZ, MR;

**private** **int** a;

**public** Data() {

F1 = 0;

F2 = 0;

}

//метод копирования Общих Ресурсов

**public** **int**[][] copyMZ() {

**return** Lab5.*clone*(MZ);

}

**public** **int**[][] copyMR() {

**return** Lab5.*clone*(MR);

}

**public** **int** copya() {

**return** a;

}

//метод сихронизации по вводу

**public** **synchronized** **void** InputSynchro() {

**try** {

**while** (F2 < 1) {

wait();

}

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

//Сигнал об окончании ввода

**public** **synchronized** **void** InputSignal() {

F2++;

notifyAll();

}

//метод сихронизации по счёту (выводу)

**public** **synchronized** **void** OutputSynchro() {

**try** {

**while** (F1 < Lab5.*P* - 1) {

wait();

}

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

//Сигнал об окончании счёта

**public** **synchronized** **void** CalcEndSignal() {

F1++;

notifyAll();

}

**public** **synchronized** **void** writeOP(**int** [][] MT1, **int** [][] MT2, **int** t) {

MR = Lab5.*clone*(MT1);

MZ = Lab5.*clone*(MT2);

a = t;

}

}

//Абстрактный класс потоков////////////////////////////////////////////////////////////

**public** **abstract** **class** Tx **extends** Thread {

**protected** **int** num;

**protected** Data dat;

**protected** **int** MRx[][];

**protected** **int** MZx[][];

**protected** **int** ax;

**public** Tx(Data d, **int** n) {

dat = d;

num = n;

}

**protected** **void** process() {

**int** sum1, sum2;

**for** (**int** i = 0; i < Lab5.*N*; i++) {

**for** (**int** j = (num-1)\*Lab5.*H*; j < num\*Lab5.*H*; j++) {

sum1 = 0; sum2 = 0;

**for** (**int** z = 0; z < Lab5.*N*; z++) {

sum1 = sum1 + Lab5.*MB*[z][j] \* MZx[i][z];

sum2 = sum2 + Lab5.*ME*[z][j] \* MRx[i][z];

}

Lab5.*MA*[i][j] = (sum1 \* ax) + sum2;

}

}

}

**protected** **int** [][] input() {

**int** [][] MT = **new** **int** [Lab5.*N*][Lab5.*N*];

**for** (**int** i = 0; i < Lab5.*N*; i++) {

**for** (**int** j = 0; j < Lab5.*N*; j++) {

MT[i][j] = 1;

}

}

**return** MT;

}

}

//ЗАДАЧА 1/////////////////////////////////////////////////////////////////////////////

**public** **class** T1 **extends** Tx {

**public** T1(Data d, **int** n) {

**super**(d, n);

}

**public** **void** run() {

System.*out*.println("Задача T" + num + " стартовала!");

//1. Ожидание ввода в задаче Т4

dat.InputSynchro();

//2. Копирование a1:=a, MZ1:=MZ, MR1:= MR

MRx = dat.copyMR();

MZx = dat.copyMZ();

ax = dat.copya();

//3. Счёт MAH =(MBH\*MZ1\*a1) + MEH\*MR1

process();

//4. Ожидание окончания счёта от Т2, Т3, Т4

dat.OutputSynchro();

//5. Вывод МА

**if** (Lab5.*N* <= 8) {

**for** (**int** i = 0; i < Lab5.N; i++) {

**for** (**int** j = 0; j < Lab5.N; j++) {

System.out.print(Lab5.MA[i][j] + " ");

}

System.out.println();

}

}

System.*out*.println("Задача T" + num + " финишировала!");

}

}

//ЗАДАЧА 2/////////////////////////////////////////////////////////////////////////////

**public** **class** T2 **extends** Tx {

**public** T2(Data d, **int** n) {

**super**(d, n);

}

**public** **void** run() {

System.*out*.println("Задача T" + num + " стартовала!");

//1. Ожидание ввода в задаче Т4

dat.InputSynchro();

//2. Копирование a2:=a, MZ2:=MZ, MR2:=MR

MRx = dat.copyMR();

MZx = dat.copyMZ();

ax = dat.copya();

//3. Счёт MAH =(MBH\*MZ2\*a2) + MEH\*MR2

process();

//4. Сигнал задаче Т1 о конце счёта

dat.CalcEndSignal();

System.*out*.println("Задача T" + num + " финишировала!");

}

}

//ЗАДАЧА 3/////////////////////////////////////////////////////////////////////////////

**public** **class** T3 **extends** Tx {

**public** T3(Data d, **int** n) {

**super**(d, n);

}

**public** **void** run() {

System.*out*.println("Задача T" + num + " стартовала!");

//1. Ожидание ввода в задаче Т4

dat.InputSynchro();

//2. Копирование a3:=a, MZ3:=MZ, MR3:=MR

MRx = dat.copyMR();

MZx = dat.copyMZ();

ax = dat.copya();

//3. Счёт MAH =(MBH\*MZ3\*a3) + MEH\*MR3

process();

//4. Сигнал задаче Т1 о конце счёта

dat.CalcEndSignal();

System.*out*.println("Задача T" + num + " финишировала!");

}

}

//ЗАДАЧА 4/////////////////////////////////////////////////////////////////////////////

**public** **class** T4 **extends** Tx {

**public** T4(Data d, **int** n) {

**super**(d, n);

}

**public** **void** run() {

System.*out*.println("Задача T" + num + " стартовала!");

//1. Ввод MB, MZ, ME, MR, a

//2. Копирование a4:=a, MZ4:=MZ, MR4:=MR

Lab5.*MB* = input();

Lab5.*ME* = input();

MZx = input();

MRx = input();

ax = 1;

dat.writeOP(MRx, MZx, ax);

//3. Cигнал задачам Т1, Т2, Т3 о завершении ввода

dat.InputSignal();

//4. Счёт MAH = (MBH\*MZ4\*a4) + MEH\*MR4

process();

//5. Сигнал задаче Т1 о конце счёта

dat.CalcEndSignal();

System.*out*.println("Задача T" + num + " финишировала!");

}

}

/\*--------------------------------------------------------------

-- Paralel and distributed computing --

-- Laboratory work #5. Java. Monitors --

-- Func: MA=(MB \* MZ \* a) + ME \* MR --

-- IO-83 Vorobyev Vitaliy --

-- 04.04.2010 --

--------------------------------------------------------------\*/

**public** **class** Lab5 {

**public** **static** **final** **int** *N* = 2500;

**public** **static** **final** **int** *P* = 4;

**public** **static** **final** **int** *H* = *N*/*P*;

**static** **int** *MB*[][] = **new** **int**[*N*][*N*];

**static** **int** *ME*[][] = **new** **int**[*N*][*N*];

**static** **int** *MA*[][] = **new** **int**[*N*][*N*];

**public** **static** **int**[][] clone(**int** [][] MM) {

**int** MT[][] = **new** **int** [MM.length][MM[0].length];

**for** (**int** i = 0; i < MM.length; i++) {

**for** (**int** j = 0; j < MM[0].length; j++) {

MT[i][j] = MM[i][j];

}

}

**return** MT;

}

**public** **static** **void** main(String[] args) {

Data dat = **new** Data();

T1 t1 = **new** T1(dat, 1);

T2 t2 = **new** T2(dat, 2);

T3 t3 = **new** T3(dat, 3);

T4 t4 = **new** T4(dat, 4);

t1.start();

t2.start();

t3.start();

t4.start();

System.*out*.println("КОНЕЦ");

}

}