Лабораторная работа №3

по курсу:

«Паралельные и распределённые вычисления»

Тема: «С#»

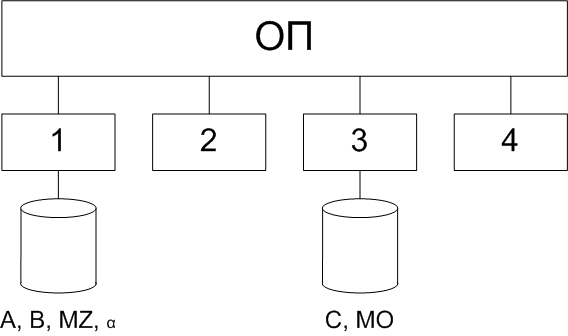
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Дроздовича Сергея

**Техническое задание**

A=sort(B- C\*(MO\*MZ\*α))



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. AH = sort(BH - C\*(MO\*MZH \*α)

**Общие ресурсы:** C, MO, α

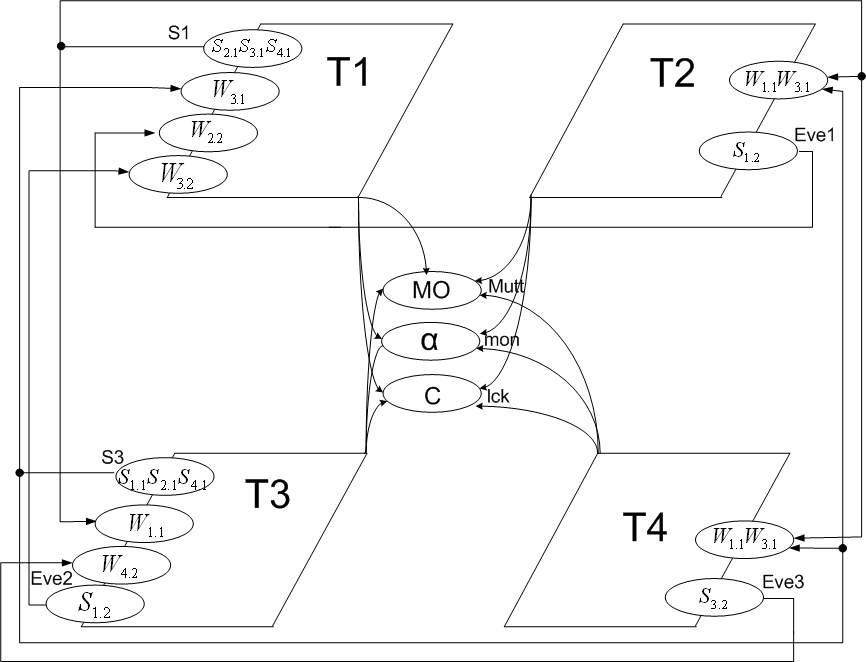
1. A2H = sort\*(AH, AH)
2. A = sort\* (A2H, A2H)

\* - сортировка слиянием

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

|  |  |  |
| --- | --- | --- |
| № | **Т1** | **ТС/КУ** |
| 1. | Ввод B, MZ, α |  |
| 2. | Сигнал задачам T2, T3, T4 о вводе B | S2-1, S3-1, S4-1 |
| 3. | Ждать ввода в T3 | W3-1 |
| 4. | Копирование C1 := C | КУ |
| 5. | Копирование МО1 := MO | КУ |
| 6. | Копирование α1 := α | КУ |
| 7. | Счет AH = sort(BH – C1 \* (MO1 \* MZH \* α1)) |  |
| 8. | Ждать окончания счета AH в T2 | W2-2 |
| 9. | Слияние A2H = sort(AH, AH) |  |
| 10. | Ждать окончания слияния A2H в T3 | W3-2 |
| 11. | Слияние A = sort(A2H, A2H) |  |
| 12. | Вывод А |  |
|  |  |  |
|  |  |  |
|  |  |  |
| № | **Т2** | **ТС/КУ** |
| 1. | Ждать ввода в T1, T3 | W1-1, W3-1 |
| 2. | Копирование C2 := C | КУ |
| 3. | Копирование МО2 := MO | КУ |
| 4. | Копирование α2 := α | КУ |
| 5. | Счет AH = sort(BH – C2 \* (MO2 \* MZH \* α2)) |  |
| 6. | Сигнал T1 об окончания счета AH | S1-2 |
| № | **Т3** | **ТС/КУ** |
| 1. | Ввод C, MO |  |
| 2. | Сигнал задачам T1, T2, T4 о вводе | S1-1, S2-1, S4-1 |
| 3. | Ждать ввода в T1 | W1-1 |
| 4. | Копирование C3 := C | КУ |
| 5. | Копирование МО3 := MO | КУ |
| 6. | Копирование α3 := α | КУ |
| 7. | Счет AH = sort(BH – C3 \* (MO3 \* MZH \* α3)) |  |
| 8. | Ждать окончания счета AH в T4 | W4-2 |
| 9. | Слияние A2H = sort(AH, AH) |  |
| 10. | Сигнал об окончания слияния A2H в T1 | S1-2 |
| № | **Т4** | **ТС/КУ** |
| 1. | Ждать ввода в T1, T3 | W1-1, W3-1 |
| 2. | Копирование C4 := C | КУ |
| 3. | Копирование МО4 := MO | КУ |
| 4. | Копирование α4 := α | КУ |
| 5. | Счет AH = sort(BH – C4 \* (MO4 \* MZH \* α4)) |  |
| 6. | Сигнал T3 об окончания счета AH | S3-2 |

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

namespace PROIILab3

{

class Program

{

static void T1()

{

Vector C1, T;

Matrix MO1;

int num = 1;

int alfa1;

//1. Ввод B, MZ, α

Console.WriteLine("T1 START");

B = new Vector(N);

B.inputVector();

MZ = new Matrix(N);

MZ.inputMatrix();

alfa = 1;

//B.setElement(5, 68);

//B.setElement(0, 77);

//B.setElement(2, 34);

//B.setElement(3, 535);

//2. Сигнал задачам T2, T3, T4 о вводе B S2-1, S3-1, S4-1

S1.Release(maxS);

//3. Ждать ввода в T3 W3-1

S3.WaitOne();

//4. Копирование C1 := C КУ

lock(lck) {

C1 = new Vector(C);

}

//5. Копирование МО1 := MO КУ

Mutt.WaitOne();

MO1 = new Matrix(MO);

Mutt.ReleaseMutex();

//6. Копирование α1 := α КУ

System.Threading.Monitor.Enter(mon);

alfa1 = alfa;

System.Threading.Monitor.Exit(mon);

//7. Счет AH = sort(BH – C1 \* (MO1 \* MZH \* α1))

T = new Vector(N);

int sum;

for (int z = (num - 1)\*H; z < num\*H; z++) {

// MO \* MZ

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

sum = 0;

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

sum = sum + MO1.getElement(i, j) \* MZ.getElement(j, z) \* alfa1;

}

T.setElement(i, sum);

}

//C\*(MO\*MZ)

sum = 0;

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

sum = sum + T.getElement(j) \* C1.getElement(j);

}

//B-C\*(MO\*MZ)

sum = B.getElement(z) - sum;

A.setElement(z, sum);

}

A.sort((num - 1) \* H, H);

Console.WriteLine("T1 - Ah");

//8. Ждать окончания счета AH в T2 W2-2

Eve1.WaitOne();

//9. Cлияние A2H = sort(AH, AH)

Vector tv = new Vector(2\*H);

int uk1 = (num - 1) \* H;

int uk2 = num \* H;

for (int z = 0; z < tv.getLength(); z++) {

if ((uk2 >= (num + 1) \* H) || ((A.getElement(uk1) <= A.getElement(uk2)) && (uk1 < num \* H)))

{

tv.setElement(z, A.getElement(uk1));

uk1++;

} else {

tv.setElement(z, A.getElement(uk2));

uk2++;

}

}

for (int i = 0; i < tv.getLength(); i++) {

A.setElement( (num - 1) \* H + i, tv.getElement(i));

}

Console.WriteLine("T1 - A2h");

//10. Ждать окончания слияния A2H в T3 W3-2

Eve2.WaitOne();

//11. Слияние A = sort(A2H, A2H)

tv = new Vector(N);

uk1 = 0;

uk2 = 2 \* H;

for (int z = 0; z < tv.getLength(); z++)

{

if ((uk2 >= N) || ((A.getElement(uk1) <= A.getElement(uk2)) && (uk1 < 2 \* H)))

{

tv.setElement(z, A.getElement(uk1));

uk1++;

}

else

{

tv.setElement(z, A.getElement(uk2));

uk2++;

}

}

for (int i = 0; i < tv.getLength(); i++)

{

A.setElement(i, tv.getElement(i));

}

Console.WriteLine("T1 - A");

//12. Вывод А

if (N <= 8) {

Console.WriteLine(A.outputVector());

}

Console.WriteLine("T1 END !!!!!!!!!!!!!!!!!!!!!!!!");

}

//=====================================================

static void T2()

{

Vector C2, T;

Matrix MO2;

int num = 2;

int alfa2;

Console.WriteLine("T2 START");

//1. Ждать ввода в T1, T3 W1-1, W3-1

S1.WaitOne();

S3.WaitOne();

//2. Копирование C2 := C КУ

lock (lck)

{

C2 = new Vector(C);

}

//3. Копирование МО2 := MO КУ

Mutt.WaitOne();

MO2 = new Matrix(MO);

Mutt.ReleaseMutex();

//4. Копирование α2 := α КУ

System.Threading.Monitor.Enter(mon);

alfa2 = alfa;

System.Threading.Monitor.Exit(mon);

//5. Счет AH = sort(BH – C2 \* (MO2 \* MZH \* α2))

T = new Vector(N);

int sum;

for (int z = (num - 1) \* H; z < num \* H; z++)

{

// MO \* MZ

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

{

sum = 0;

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

{

sum = sum + MO2.getElement(i, j) \* MZ.getElement(j, z) \* alfa2;

}

T.setElement(i, sum);

}

//C\*(MO\*MZ)

sum = 0;

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

{

sum = sum + T.getElement(j) \* C2.getElement(j);

}

//B-C\*(MO\*MZ)

sum = B.getElement(z) - sum;

A.setElement(z, sum);

}

A.sort((num - 1) \* H, H);

Console.WriteLine("T2 - Ah");

//6. Сигнал T1 об окончания счета AH S1-2

Eve1.Set();

Console.WriteLine("T2 END !!!!!!!!!!!!!!!!!!!!!!!!");

}

//=====================================================

static void T3()

{

Vector C3, T;

Matrix MO3;

int num = 3;

int alfa3;

Console.WriteLine("T3 START");

//1. Ввод C, MO

MO = new Matrix(N);

MO.inputMatrix();

C = new Vector(N);

C.inputVector();

//2. Сигнал задачам T1, T2, T4 о вводе S1-1, S2-1, S4-1

S3.Release(maxS);

//3. Ждать ввода в T1 W1-1

S1.WaitOne();

//4. Копирование C3 := C КУ

lock (lck)

{

C3 = new Vector(C);

}

//5. Копирование МО3 := MO КУ

Mutt.WaitOne();

MO3 = new Matrix(MO);

Mutt.ReleaseMutex();

//6. Копирование α3 := α КУ

System.Threading.Monitor.Enter(mon);

alfa3 = alfa;

System.Threading.Monitor.Exit(mon);

//7. Счет AH = sort(BH – C3 \* (MO3 \* MZH \* α3))

T = new Vector(N);

int sum;

for (int z = (num - 1) \* H; z < num \* H; z++)

{

// MO \* MZ

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

{

sum = 0;

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

{

sum = sum + MO3.getElement(i, j) \* MZ.getElement(j, z) \* alfa3;

}

T.setElement(i, sum);

}

//C\*(MO\*MZ)

sum = 0;

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

{

sum = sum + T.getElement(j) \* C3.getElement(j);

}

//B-C\*(MO\*MZ)

sum = B.getElement(z) - sum;

A.setElement(z, sum);

}

A.sort((num - 1) \* H, H);

Console.WriteLine("T3 - Ah");

//8. Ждать окончания счета AH в T4 W4-2

Eve3.WaitOne();

//9. Слияние A2H = sort(AH, AH)

Vector tv = new Vector(2 \* H);

int uk1 = (num - 1) \* H;

int uk2 = num \* H;

for (int z = 0; z < tv.getLength(); z++)

{

if ((uk2 >= (num + 1) \* H) || ((A.getElement(uk1) <= A.getElement(uk2)) && (uk1 < num \* H)))

{

tv.setElement(z, A.getElement(uk1));

uk1++;

}

else

{

tv.setElement(z, A.getElement(uk2));

uk2++;

}

}

for (int i = 0; i < tv.getLength(); i++)

{

A.setElement((num - 1) \* H + i, tv.getElement(i));

}

Console.WriteLine("T3 - A2h");

//10. Сигнал об окончания слияния A2H в T1 S1-2

Eve2.Set();

Console.WriteLine("T3 END !!!!!!!!!!!!!!!!!!!!!!!!");

}

//=====================================================

static void T4()

{

Vector C4, T;

Matrix MO4;

int num = 4;

int alfa4;

Console.WriteLine("T4 START");

//1. Ждать ввода в T1, T3 W1-1, W3-1

S1.WaitOne();

S3.WaitOne();

//2. Копирование C4 := C КУ

lock (lck)

{

C4 = new Vector(C);

}

//3. Копирование МО4 := MO КУ

Mutt.WaitOne();

MO4 = new Matrix(MO);

Mutt.ReleaseMutex();

//4. Копирование α4 := α КУ

System.Threading.Monitor.Enter(mon);

alfa4 = alfa;

System.Threading.Monitor.Exit(mon);

//5. Счет AH = sort(BH – C4 \* (MO4 \* MZH \* α4))

T = new Vector(N);

int sum;

for (int z = (num - 1) \* H; z < num \* H; z++)

{

// MO \* MZ \* alfa

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

{

sum = 0;

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

{

sum = sum + MO4.getElement(i, j) \* MZ.getElement(j, z) \* alfa4;

}

T.setElement(i, sum);

}

//C\*(MO\*MZ)

sum = 0;

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

{

sum = sum + T.getElement(j) \* C4.getElement(j);

}

//B-C\*(MO\*MZ)

sum = B.getElement(z) - sum;

A.setElement(z, sum);

}

A.sort((num - 1) \* H, H);

Console.WriteLine("T4 - Ah");

//6. Сигнал T3 об окончания счета AH S3-2

Eve3.Set();

Console.WriteLine("T4 END !!!!!!!!!!!!!!!!!!!!!!!!");

}

public static int N = 8;

public static int P = 4;

public static int H = N/P;

public static int maxS = 3;

public static Matrix MO, MZ;

public static Vector A, B, C;

public static int alfa;

public static Semaphore S1, S3;

public static Mutex Mutt = new Mutex(false);

public static object lck = new object();

public static object mon = new object();

public static EventWaitHandle Eve1, Eve2, Eve3;

static void Main(string[] args)

{

Console.WriteLine("Main thread started");

A = new Vector(N);

S1 = new Semaphore(0, maxS);

S3 = new Semaphore(0, maxS);

Eve1 = new AutoResetEvent(false);

Eve2 = new AutoResetEvent(false);

Eve3 = new AutoResetEvent(false);

Thread thr1 = new Thread(T1);

Thread thr2 = new Thread(T2);

Thread thr3 = new Thread(T3);

Thread thr4 = new Thread(T4);

thr1.Start();

thr2.Start();

thr3.Start();

thr4.Start();

Console.WriteLine("Main thread finished");

Console.ReadKey();

}

}