Лабораторная работа №3

по курсу:

«Паралельные и распределённые вычисления»

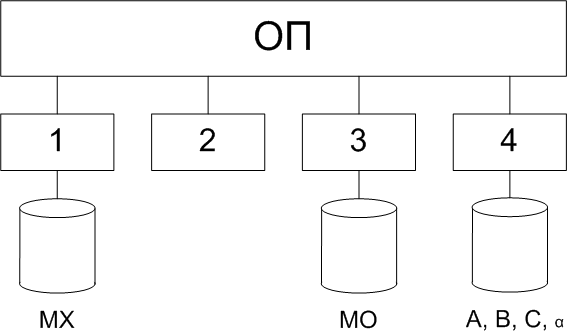
Тема: «С#»

Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

**Техническое задание**

A=sort(B\*MX+ α\*C\*MO)



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. AH = sort(B\*MXH+ α\*C\*MOH)

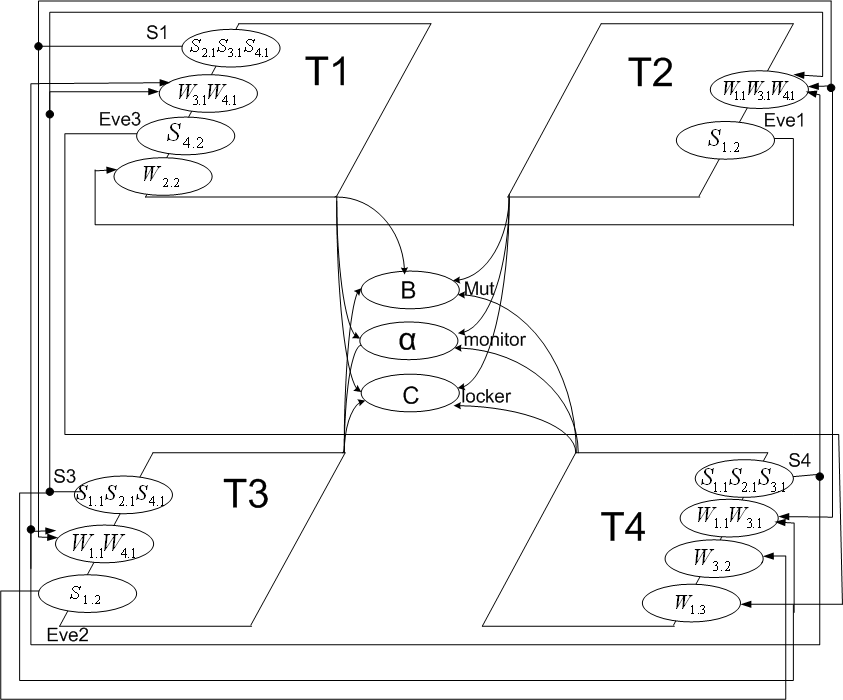
**Общие ресурсы:** B, C

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

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | | **Т1** | | | **ТС/КУ** |
| 1. | | Ввод MX | | |  |
| 2. | | Сигнал задачам T2, T3, T4 о вводе MX | | | S2-1, S3-1, S4-1 |
| 3. | | Ждать ввода в T3, T4 | | | W3-1, W4-1 |
| 4. | | Копирование C1 := C | | | КУ |
| 5. | | Копирование B1 := B | | | КУ |
| 6. | Копирование α1:= α | | | | КУ |
|  | | |
| 7. | | Счет AH = sort(B1\*MXH+ α1\*C1\*MOH) | | |  |
| 8. | | Ждать окончания счета AH в T2 | | | W2-2 |
| 9. | | Слияние A2H = sort(AH, AH) | | |  |
| 10. | | Сигнал об окончания слияния A2H в T4 | | | S4-2 |
|  | | | |
|  | | | |
|  | | | |
| № | **Т2** | | | | **ТС/КУ** |
| 1. | Ждать ввода в T1, T3, T4 | | | | W1-1, W3-1, W4-1 |
| 2. | Копирование C2 := C | | | | КУ |
| 3. | Копирование B2 := B | | | | КУ |
| 4. | Копирование α2:= α | | | | КУ |
| 5. | Счет AH = sort(B2\*MXH+ α2\*C2\*MOH) | | | |  |
| 6. | Сигнал T1 об окончания счета AH | | | | S1-2 |
|  |  | | | |  |
| № | **Т3** | | | | **ТС/КУ** |
| 1. | Ввод MO | | | |  |
| 2. | Сигнал задачам T1, T2, T4 о вводе MO | | | | S1-1, S2-1, S4-1 |
| 3. | Ждать ввода в T1, T4 | | | | W1-1, W4-1 |
| 4. | Копирование C3 := C | | | | КУ |
| 5. | Копирование B3:= B | | | | КУ |
| 6. | Копирование α3:= α | | | | КУ |
| 7. | Счет AH = sort(B3\*MXH+ α3\*C3\*MOH) | | | |  |
| 8. | Сигнал T4 об окончания счета AH | | | | S1-2 |
|  |
| № | **Т4** | | | | **ТС/КУ** |
| 1. | Ввод B, C | | | |  |
| 2. | Сигнал задачам T1, T2, T3 о вводе MR | | | | S1-1, S2-1, S3-1 |
| 3. | Ждать ввода в T1, T3 | | | | W1-1, W3-1 |
| 4. | Копирование C4 := C | | | | КУ |
| 5. | Копирование B4:= B | | | | КУ |
|  |  | | | |  |
| 6. | Копирование α4:= α | | | | КУ |
| 7. | Счет AH = sort(B4\*MXH+ α4\*C4\*MOH) | | | |  |
| 8.  9.  10.  11.  12. | Ждать окончания счета AH в T3  Слияние A2H = sort(AH, AH)  Ждать окончания счета A2H в T1  Слияние A = sort(A2H, A2H)  Вывод А | | | | W3-2  W1-3 |
|  |  | | | |  |
|  |  | | | |  |
|  |  | | | |  |

**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, B1;

int num = 1;

int alfa1;

Console.WriteLine("T1: started");

//1 Ввод MX

MX = new Matrix(N);

MX.inputMatrix();

//MX.setElement(0, 1, 2);

//MX.setElement(0, 3, -8);

//MX.setElement(0, 7, -3);

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

S1.Release(maxS);

//3 Ждать ввода в T3, T4 W3-1, W4-1

S3.WaitOne();

S4.WaitOne();

//4 Копирование C1 := C КУ

lock (locker)

{

C1 = new Vector(C);

}

//5 Копирование B1 := B КУ

Mut.WaitOne();

B1 = new Vector(B);

Mut.ReleaseMutex();

//6 Копирование alfa1:= alfa КУ

System.Threading.Monitor.Enter(monitor);

alfa1 = alfa;

System.Threading.Monitor.Exit(monitor);

//7 Счет AH = sort(B1\*MXH+C1\*MOH\*alfa)

int sum1 = 0;

int sum2 = 0;

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

{

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

{

sum1 = 0;

sum2 = 0;

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

{

sum1 = sum1 + C1.getElement(z) \* MO.getElement(j, z);

sum2 = sum2 + B1.getElement(z) \* MX.getElement(j, z) \* alfa1;

}

}

A.setElement(z, sum1 + sum2);

}

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

//7 Ждать окончания счета AH в T2

Eve1.WaitOne(); // W2-2

//8 Слияние A2H = sort(AH, AH)

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

int uk1 = 0;

int uk2 = H;

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

{

if ((uk2 >= 2 \* H) || ((A.getElement(uk1) <= A.getElement(uk2)) && (uk1 < 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: Calculated A2h");

//9 Сигнал об окончания слияния A2H в T4 S1-2

Eve3.Set();

Console.WriteLine("T1: \*\*FINISHED\*\*");

}

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

static void T2()

{

Vector C2, B2;

int num = 2;

int alfa2;

Console.WriteLine("T2: started");

//1 Ждать ввода в T1, T3, T4 W1-1, W3-1, W4-1

S1.WaitOne();

S3.WaitOne();

S4.WaitOne();

//2 Копирование C1 := C КУ

lock (locker)

{

C2 = new Vector(C);

}

//3 Копирование B1 := B КУ

Mut.WaitOne();

B2 = new Vector(B);

Mut.ReleaseMutex();

//4 Копирование alfa1:= alfa КУ

System.Threading.Monitor.Enter(monitor);

alfa2 = alfa;

System.Threading.Monitor.Exit(monitor);

//5 Счет AH = sort(B2\*MXH+ α2\*C2\*MOH)

int sum1 = 0;

int sum2 = 0;

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

{

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

{

sum1 = 0;

sum2 = 0;

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

{

sum1 = sum1 + C2.getElement(z) \* MO.getElement(j, z);

sum2 = sum2 + B2.getElement(z) \* MX.getElement(j, z) \* alfa2;

}

}

A.setElement(z, sum1 + sum2);

}

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

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

Eve1.Set();

Console.WriteLine("T2: \*\*FINISHED\*\*");

}

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

static void T3()

{

Vector C3, B3;

int num = 3;

int alfa3;

Console.WriteLine("T3: started");

//1. Ввод MO

MO = new Matrix(N);

MO.inputMatrix();

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

S3.Release(maxS);

//3. Ждать ввода в T1, T4 W1-1, W4-1

S1.WaitOne();

S4.WaitOne();

//4. Копирование C3 := C КУ

lock (locker)

{

C3 = new Vector(C);

}

//5 Копирование B1 := B КУ

Mut.WaitOne();

B3 = new Vector(B);

Mut.ReleaseMutex();

//6 Копирование alfa1:= alfa КУ

System.Threading.Monitor.Enter(monitor);

alfa3 = alfa;

System.Threading.Monitor.Exit(monitor);

//7. Счет AH = sort(B3\*MXH+ α3\*C3\*MOH)

int sum1 = 0;

int sum2 = 0;

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

{

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

{

sum1 = 0;

sum2 = 0;

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

{

sum1 = sum1 + C3.getElement(z) \* MO.getElement(j, z);

sum2 = sum2 + B3.getElement(z) \* MX.getElement(j, z) \* alfa3;

}

}

A.setElement(z, sum1 + sum2);

}

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

//8. Сигнал T4 об окончания счета AH S1-2

Eve2.Set();

Console.WriteLine("T3: \*\*FINISHED\*\*");

}

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

static void T4()

{

//1. Ввод B, C

Vector C4, B4;

int num = 4;

int alfa4;

Console.WriteLine("T4: started");

C = new Vector(N);

B = new Vector(N);

C.inputVector();

B.inputVector();

alfa = 1;

//2. Сигнал задачам T1, T2, T3 о вводе MR S1-1, S2-1, S3-1

S4.Release(maxS);

//3. Ждать ввода в T1, T3 W1-1, W3-1

S1.WaitOne();

S3.WaitOne();

//4. Копирование C3 := C КУ

lock (locker)

{

C4 = new Vector(C);

}

//5 Копирование B1 := B КУ

Mut.WaitOne();

B4 = new Vector(B);

Mut.ReleaseMutex();

//6 Копирование alfa1:= alfa КУ

System.Threading.Monitor.Enter(monitor);

alfa4 = alfa;

System.Threading.Monitor.Exit(monitor);

//7. Счет AH = sort(B3\*MXH+ α3\*C3\*MOH)

int sum1 = 0;

int sum2 = 0;

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

{

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

{

sum1 = 0;

sum2 = 0;

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

{

sum1 = sum1 + C4.getElement(z) \* MO.getElement(j, z);

sum2 = sum2 + B4.getElement(z) \* MX.getElement(j, z) \* alfa4;

}

}

A.setElement(z, sum1 + sum2);

}

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

//8.Ждать окончания счета AH в T3 W2-2

Eve2.WaitOne();

//9.Слияние A2H = sort(AH, AH)

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

int uk1 = 2 \* H;

int uk2 = 3 \* H;

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

{

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

{

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

uk1++;

}

else

{

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

uk2++;

}

}

Console.WriteLine("LOL");

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

{

A.setElement(2 \* H + i, tv.getElement(i));

}

Console.WriteLine("T4: Calculated A2h");

//10.Ждать окончания счета A2H в T1 W1-3

Eve3.WaitOne();

//11.Слияние A = sort(A2H, A2H)

A.sort(0, N);

//12.Вывод А

Console.WriteLine("T4: Calculated rezult");

if (N <= 8)

{

Console.WriteLine(A.outputVector());

}

Console.WriteLine("T4: \*\*FINISHED\*\*");

}

public static int N = 8;

public static int P = 4;

public static int H = N/P;

public static int maxS = 4;

public static Matrix MO, MX;

public static Vector A, B, C;

public static int alfa;

public static Semaphore S1, S3, S4;

public static Mutex Mut = new Mutex(false); //мютекс

public static object locker = new object(); //замок

public static object monitor = new object(); //монитор

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);//input sem

S3 = new Semaphore(0, maxS);

S4 = new Semaphore(0, maxS);

Eve1 = new AutoResetEvent(false);//окончания счета AH

Eve2 = new AutoResetEvent(false);//окончания слияния A2H k T4

Eve3 = new AutoResetEvent(false);//Сигнал T4 об окончания счета AH

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();

}

}