Лабораторная работа №3

по курсу:

«Паралельные и распределённые вычисления»

Тема: «С#»

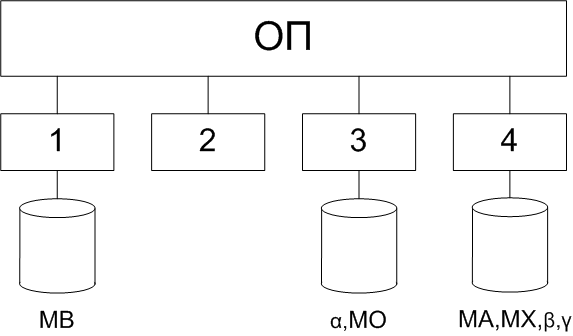
Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

MA=sort(β\*MB+α\*MO\*MX\*γ)



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. MAH = sort(β\*MBH+α\*MO\*MXH\*γ)

**Общие ресурсы:** β, α, MO, γ

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

**№ Т1 ТС/КУ**

1. Ввод МB
2. Cигнал задачам Т2, Т3, Т4 о завершении ввода S2.1, S3.1, S4.1
3. Ждать окончания ввода в задачах Т3, Т4 W3.1, W4.1
4. Копирование β1 := β КУ
5. Копирование α1 := α, MO1 := MO КУ
6. Копирование γ1:= γ КУ
7. Вычисление MAH = sort(β1\*MBH+α1\*MO1\*MXH\*γ1)
8. Сигнал задаче Т4 об окончании слияния MAH S4.2

**№ Т2 ТС/КУ**

1. Ждать окончания ввода в задачах T1, Т3, Т4 W1.1, W3.1, W4.1
2. Копирование β2 := β КУ
3. Копирование α2 := α, MO2 := MO КУ
4. Копирование γ2:= γ КУ
5. Вычисление MAH = sort(β2\*MBH+α2\*MO2\*MXH\*γ2)
6. Сигнал задаче Т4 об окончании слияния MAH S4.2

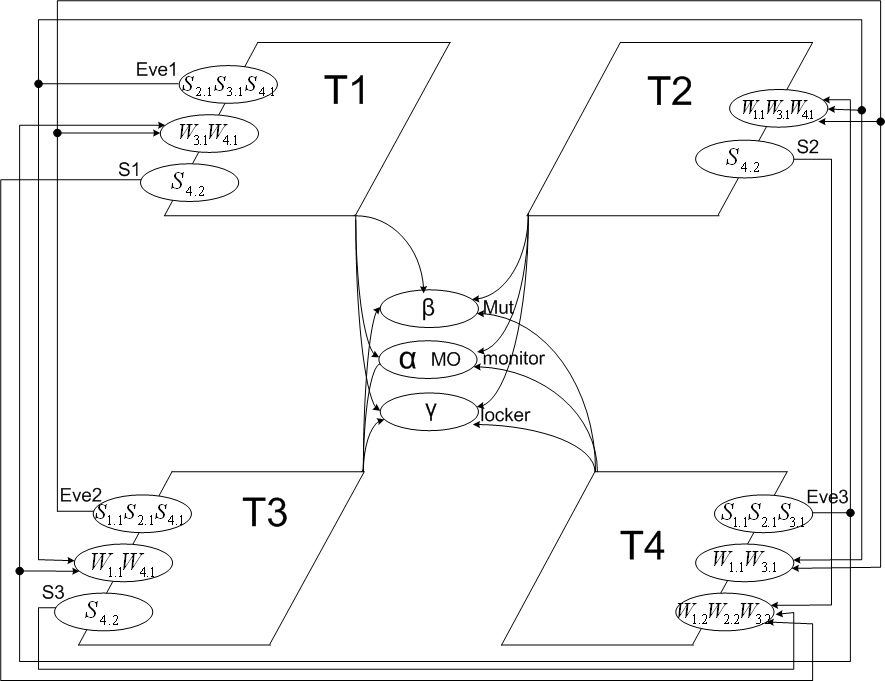
**№ Т3 ТС/КУ**

1. Ввод α, MO
2. Cигнал задачам Т1, Т2, Т4 о завершении ввода S1.1, S2.1, S4.1
3. Ждать окончания ввода в задачах Т1, Т4 W1.1, W4.1
4. Копирование β3 := β КУ
5. Копирование α3 := α, MO3 := MO КУ
6. Копирование γ3:= γ КУ
7. Вычисление MAH = sort(β3\*MBH+α3\*MO3\*MXH\*γ3)
8. Сигнал задаче Т4 об окончании счёта S4.2

**№ Т4 ТС/КУ**

1. Ввод γ, β, MX
2. Cигнал задачам Т1, Т2, Т3 о завершении ввода S1.1, S2.1, S3.1
3. Ждать окончания ввода в задачах Т1, Т3 W1.1, W3.1
4. Копирование β4 := β КУ
5. Копирование α4 := α, MO4 := MO КУ
6. Копирование γ4:= γ КУ
7. Вычисление MAH = sort(β4\*MBH+α4\*MO4\*MXH\*γ4)
8. Ждать окончания счёта в Т1, T2, T3 W1.2, W2.2, W3.2
9. Вывод МА

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

/\*\*

\* @author Vorobyev Vitaliy IO-83

\* Paralel and distributed computing

\* Laboratory work #3. C#

\* 21.02.2011

\*/

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

namespace PROIILab3

{

class Program

{

/////////////////ЗАДАЧА Т1//////////////////////////////

static void T1()

{

int num = 1;

int beta1, gamma1, alfa1;

Matrix MO1;

Console.WriteLine("T1 STARTED");

//1. Ввод МB

MB = new Matrix(N);

MB.inputMatrix();

//2. Cигнал задачам Т2, Т3, Т4 о завершении ввода

Eve1.Set(); //S2.1, S3.1, S4.1

//3. Ждать окончания ввода в задачах Т3, Т4

Eve2.WaitOne();//W3.1

Eve3.WaitOne();//W4.1

//4. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование β1 := β

Mut.WaitOne();

beta1 = beta;

Mut.ReleaseMutex();

//5. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование α1 := α, MO1 := MO

System.Threading.Monitor.Enter(monitor);

alfa1 = alfa;

MO1 = MO;

System.Threading.Monitor.Exit(monitor);

//6. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование γ1:= γ

lock (locker)

{

gamma1 = gamma;

}

//7. Вычисление MAH = sort(β1\*MBH+α1\*MO1\*MXH\*γ1)

int sum;

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

{

// MO1 \* MXh \* α1 \* γ1

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

{

sum = 0;

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

{

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

}

sum = sum \* alfa1 \* gamma1 + beta1 \* MB.getElement(i, z);

MA.setElement(i, z, sum);

}

}

MA.sort(num, H);

//8. Сигнал задаче Т4 об окончании слияния MAH

S1.Release(maxS);//S4.2

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

}

/////////////////ЗАДАЧА Т2//////////////////////////////

static void T2()

{

int num = 2;

int beta2, gamma2, alfa2;

Matrix MO2;

Console.WriteLine("T1 STARTED");

//1. Ждать окончания ввода в задачах T1, Т3, Т4

Eve1.WaitOne();//W1.1

Eve2.WaitOne();//W3.1

Eve3.WaitOne();//W4.1

//2. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование β2 := β

Mut.WaitOne();

beta2 = beta;

Mut.ReleaseMutex();

//3. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование α2 := α, MO2 := MO

System.Threading.Monitor.Enter(monitor);

alfa2 = alfa;

MO2 = MO;

System.Threading.Monitor.Exit(monitor);

//4. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование γ2:= γ

lock (locker)

{

gamma2 = gamma;

}

//5. Вычисление MAH = sort(β1\*MBH+α1\*MO1\*MXH\*γ1)

int sum;

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

{

// MO2 \* MXh \* α2 \* γ2

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

{

sum = 0;

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

{

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

}

sum = sum \* alfa2 \* gamma2 + beta2 \* MB.getElement(i, z);

MA.setElement(i, z, sum);

}

}

MA.sort(num, H);

//6. Сигнал задаче Т4 об окончании слияния MAH

S2.Release(maxS); //S4.2

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

}

/////////////////ЗАДАЧА Т3//////////////////////////////

static void T3()

{

int num = 3;

int beta3, gamma3, alfa3;

Matrix MO3;

Console.WriteLine("T1 STARTED");

//1. Ввод α, MO

alfa = 1;

MO = new Matrix(N);

MO.inputMatrix();

//2. Cигнал задачам Т1, Т2, Т4 о завершении ввода

Eve2.Set(); //S1.1, S2.1, S4.1

//3. Ждать окончания ввода в задачах Т1, Т4

Eve1.WaitOne();//W1.1

Eve3.WaitOne();//W4.1

//4. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование β3 := β

Mut.WaitOne();

beta3 = beta;

Mut.ReleaseMutex();

//5. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование α3 := α, MO3 := MO

System.Threading.Monitor.Enter(monitor);

alfa3 = alfa;

MO3 = MO;

System.Threading.Monitor.Exit(monitor);

//6. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование γ3:= γ

lock (locker)

{

gamma3 = gamma;

}

//7. Вычисление MAH = sort(β1\*MBH+α1\*MO1\*MXH\*γ1)

int sum;

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

{

// MO3 \* MXh \* α3 \* γ3

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

{

sum = 0;

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

{

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

}

sum = sum \* alfa3 \* gamma3 + beta3 \* MB.getElement(i, z);

MA.setElement(i, z, sum);

}

}

MA.sort(num, H);

//8. Сигнал задаче Т4 об окончании слияния MAH

S3.Release(maxS); //S4.2

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

}

/////////////////ЗАДАЧА Т4//////////////////////////////

static void T4()

{

int num = 4;

int beta4, gamma4, alfa4;

Matrix MO4;

Console.WriteLine("T1 STARTED");

//1. Ввод γ, β, MX

gamma = 2;

beta = 3;

MX = new Matrix(N);

MX.inputMatrix();

//MX.setElement(1, 4, 13);

//2. Cигнал задачам Т1, Т2, Т3 о завершении ввода S1.1, S2.1, S3.1

Eve3.Set(); //S1.1, S2.1, S3.1

//3. Ждать окончания ввода в задачах Т1, Т3

Eve1.WaitOne();//W1.1

Eve2.WaitOne();//W3.1

//4. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование β4 := β

Mut.WaitOne();

beta4 = beta;

Mut.ReleaseMutex();

MB.setElement(1, 1, 2);

//5. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование α4 := α, MO4 := MO

System.Threading.Monitor.Enter(monitor);

alfa4 = alfa;

MO4 = MO;

System.Threading.Monitor.Exit(monitor);

//6. (КРИТИЧЕСКАЯ СЕКЦИЯ) Копирование γ4:= γ

lock (locker)

{

gamma4 = gamma;

}

//7. Вычисление MAH = sort(β1\*MBH+α1\*MO1\*MXH\*γ1)

int sum;

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

{

// MO4 \* MXh \* α4 \* γ4

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

{

sum = 0;

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

{

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

}

sum = sum \* alfa4 \* gamma4 + beta4 \* MB.getElement(i, z);

MA.setElement(i, z, sum);

}

}

MA.sort(num, H);

//8. Ждать окончания счёта в Т1, T2, T3 W1.2, W2.2, W3.2

S1.WaitOne();

S2.WaitOne();

S3.WaitOne();

//9. Вывод МА

Console.WriteLine();

Console.WriteLine();

Console.WriteLine(MA.outputMatrix());

Console.WriteLine();

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

}

public static int N = 8;

public static int P = 4;

public static int H = N/P;

public static int maxS = 3;

public static int beta, alfa, gamma;

public static Matrix MO, MB, MX, MA;

public static Vector A, B, C;

public static Semaphore S1, S2, S3; //семафоры

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; //события по вводу

static void Main(string[] args)

{

Console.WriteLine("Main thread started");

MA = new Matrix(N);

S1 = new Semaphore(0, maxS);

S2 = new Semaphore(0, maxS);

S3 = new Semaphore(0, maxS);

Eve1 = new EventWaitHandle(false, EventResetMode.ManualReset); //установка в ручной режим?

Eve2 = new EventWaitHandle(false, EventResetMode.ManualReset);

Eve3 = new EventWaitHandle(false, EventResetMode.ManualReset);

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();

}

}