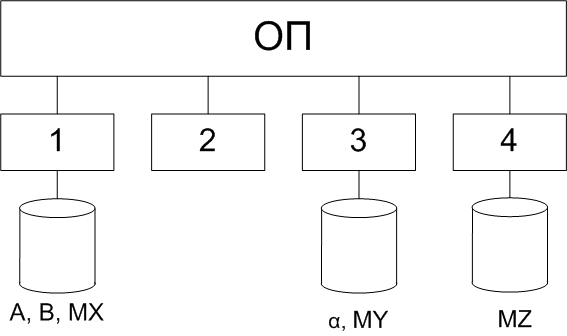
**Техническое задание**

A=B \* (MX \* MY – α \* MZ)



**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. AH = B \* (MX \* MYH – α \* MZH)

**Общие ресурсы:** B, MX, α

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

**№ Т1 ТС/КУ**

1. Ввод B, МX
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 B1 := B, MX1 := MX, α1 := α КУ
5. Вычисление AH = B1 \* (MX1 \* MYH – α1 \* MZH)
6. Ждать окончания счёта в Т2, T3, T4 W2.2, W3.2, W4.2
7. Вывод А

**№ Т2 ТС/КУ**

1. Ждать окончания ввода в задачах Т1, T3, Т4 W1.1, W3.1, W4.1
2. Копирование B2 := B, MX2 := MX, α2 := α КУ
3. Вычисление AH = B2 \* (MX2 \* MYH – α2 \* MZH)
4. Сигнал задаче Т1 об окончании счёта AH S4.2

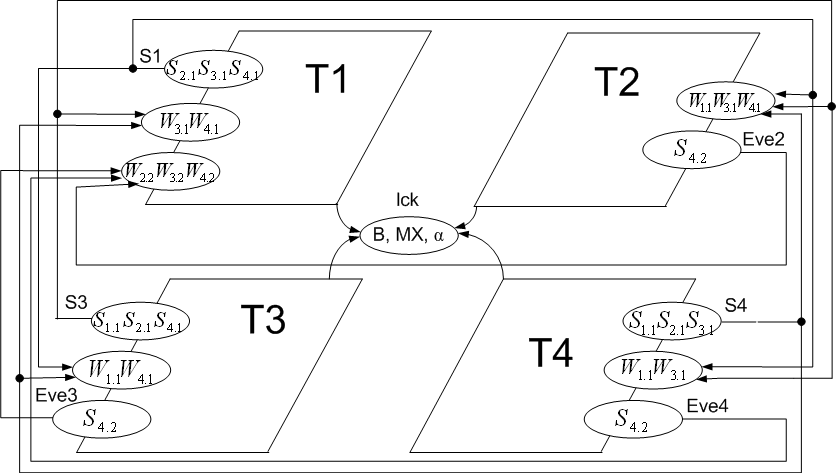
**№ Т3 ТС/КУ**

1. Ввод α, MY
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 B3 := B, MX3 := MX, α3 := α КУ
5. Вычисление AH = B3 \* (MX3 \* MYH – α3 \* MZH)
6. Сигнал задаче Т1 об окончании счёта AH S4.2

**№ Т4 ТС/КУ**

1. Ввод MZ
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 B4 := B, MX4 := MX, α4 := α КУ
5. Вычисление AH = B4 \* (MX4 \* MYH – α4 \* MZH)
6. Сигнал задаче Т1 об окончании счёта AH S4.2

**Этап 3. Разработка схемы взаимодействия задач**

****

**Этап 4. Разработка программы**

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

namespace PROCSharpCourse

{

class Program

{

static void T1()

{

Vector B1;

Matrix MX1;

int alfa1;

int num = 1;

Console.WriteLine("T1 START");

//1. Ввод B, МX

B = new Vector(N);

B.inputVector();

MX = new Matrix(N);

MX.inputMatrix();

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

S1.Release(maxS); //S2.1, S3.1, S4.1

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

S3.WaitOne();

S4.WaitOne();

//4. Копирование B1 := B, MX1 := MX, α1 := α

lock (lck)

{

B1 = new Vector(B);

MX1 = new Matrix(MX);

alfa1 = alfa;

}

//5. Вычисление AH = B1 \* (MX1 \* MYH – α1 \* MZH)

int sum;

Matrix T = new Matrix(N);

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

{

// MX1 \* MYH – α1 \* MZH

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

{

sum = 0;

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

{

sum = sum + MX1.getElement(i, j) \* MY.getElement(j, z);

}

sum = sum - alfa1 \* MZ.getElement(i, z);

T.setElement(i, z, sum);

}

}

//B1 \* (MX1 \* MYH – α1 \* MZH)

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

{

sum = 0;

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

{

sum = sum + B1.getElement(j) \* T.getElement(j, z);

}

A.setElement(z, sum);

}

//6. Ждать окончания счёта в Т2, T3, T4

Eve2.WaitOne(); //W2.2, W3.2, W4.2

Eve3.WaitOne();

Eve4.WaitOne();

//7. Вывод А

if (N <= 24) {

Console.WriteLine(A.outputVector());

}

sp = DateTime.Now - dold;

Console.WriteLine(sp);

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

}

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

static void T2()

{

Vector B2;

Matrix MX2;

int alfa2;

int num = 2;

Console.WriteLine("T2 START");

//1. Ждать окончания ввода в задачах Т1, T3, Т4

S1.WaitOne();

S3.WaitOne();

S4.WaitOne();//W1.1, W3.1, W4.1

//2. Копирование B2 := B, MX2 := MX, α2 := α

lock (lck)

{

B2 = new Vector(B);

MX2 = new Matrix(MX);

alfa2 = alfa;

}

//3. Вычисление AH = B2 \* (MX2 \* MYH – α2 \* MZH)

int sum;

Matrix T = new Matrix(N);

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

{

// MX1 \* MYH – α1 \* MZH

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

{

sum = 0;

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

{

sum = sum + MX2.getElement(i, j) \* MY.getElement(j, z);

}

sum = sum - alfa2 \* MZ.getElement(i, z);

T.setElement(i, z, sum);

}

}

//B2 \* (MX1 \* MYH – α2 \* MZH)

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

{

sum = 0;

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

{

sum = sum + B2.getElement(j) \* T.getElement(j, z);

}

A.setElement(z, sum);

}

//4. Сигнал задаче Т1 об окончании счёта AH

Eve2.Set(); //S4.2

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

}

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

static void T3()

{

Vector B3;

Matrix MX3;

int alfa3;

int num = 3;

Console.WriteLine("T3 START");

//1. Ввод α, MY

alfa = 1;

MY = new Matrix(N);

MY.inputMatrix();

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

S3.Release(maxS); //S1.1, S2.1, S4.1

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

S1.WaitOne();

S4.WaitOne(); //W1.1, W4.1

//4. Копирование B3 := B, MX3 := MX, α3 := α

lock (lck)

{

B3 = new Vector(B);

MX3 = new Matrix(MX);

alfa3 = alfa;

}

//5. Вычисление AH = B3 \* (MX3 \* MYH – α3 \* MZH)

int sum;

Matrix T = new Matrix(N);

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

{

// MX1 \* MYH – α1 \* MZH

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

{

sum = 0;

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

{

sum = sum + MX3.getElement(i, j) \* MY.getElement(j, z);

}

sum = sum - alfa3 \* MZ.getElement(i, z);

T.setElement(i, z, sum);

}

}

//B1 \* (MX1 \* MYH – α1 \* MZH)

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

{

sum = 0;

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

{

sum = sum + B3.getElement(j) \* T.getElement(j, z);

}

A.setElement(z, sum);

}

//6. Сигнал задаче Т1 об окончании счёта AH

Eve3.Set(); //S4.2

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

}

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

static void T4()

{

Vector B4;

Matrix MX4;

int alfa4;

int num = 4;

Console.WriteLine("T4 START");

//1. Ввод MZ

MZ = new Matrix(N);

MZ.inputMatrix();

//2. Cигнал задачам Т1, Т2, Т3 о завершении ввода

S4.Release(maxS); //S1.1, S2.1, S3.1

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

S1.WaitOne();

S3.WaitOne(); //W1.1, W3.1

//4. Копирование B4 := B, MX4 := MX, α4 := α

lock (lck)

{

B4 = new Vector(B);

MX4 = new Matrix(MX);

alfa4 = alfa;

}

//5. Вычисление AH = B4 \* (MX4 \* MYH – α4 \* MZH)

int sum;

Matrix T = new Matrix(N);

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

{

// MX1 \* MYH – α1 \* MZH

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

{

sum = 0;

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

{

sum = sum + MX4.getElement(i, j) \* MY.getElement(j, z);

}

sum = sum - alfa4 \* MZ.getElement(i, z);

T.setElement(i, z, sum);

}

}

//B1 \* (MX1 \* MYH – α1 \* MZH)

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

{

sum = 0;

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

{

sum = sum + B4.getElement(j) \* T.getElement(j, z);

}

A.setElement(z, sum);

}

//6. Сигнал задаче Т1 об окончании счёта AH

Eve4.Set(); //S4.2

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

}

public static int N = 4000;

public static int P = 4;

public static int H = N/P;

public static int maxS = 3;

public static Matrix MX, MY, MZ;

public static Vector A, B;

public static int alfa;

public static Semaphore S1, S3, S4;

public static object lck = new object();

public static DateTime dold;

public static TimeSpan sp;

public static EventWaitHandle Eve2, Eve3, Eve4;

static void Main(string[] args)

{

Console.WriteLine("Main thread started with N = " + N);

A = new Vector(N);

Console.ReadKey();

dold = DateTime.Now;

S1 = new Semaphore(0, maxS); //ввод с 1 задачи

S3 = new Semaphore(0, maxS); //ввод с 3 задачи

S4 = new Semaphore(0, maxS); //ввод с 4 задачи

Eve2 = new AutoResetEvent(false);

Eve3 = new AutoResetEvent(false);

Eve4 = 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();

}

}

class Vector

{

private int[] vect;

public Vector(int length)

{

vect = new int[length];

}

public Vector(Vector vec) {

vect = new int[vec.getLength()];

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

vect[i] = vec.getElement(i);

}

}

public int getLength()

{

return vect.Length;

}

public int getElement(int i)

{

return vect[i];

}

public void setElement(int i, int func)

{

vect[i] = func;

}

public int findMax()

{

int max = vect[0];

for (int i = 1; i < vect.Length; i++)

{

if (vect[i] > max)

{

max = vect[i];

}

}

return max;

}

public void sort(int a, int b)

{

Array.Sort(vect, a, b);

}

public void inputVector()

{

int i = 0;

while (i < vect.Length)

{

vect[i] = 1;

i++;

}

}

public String outputVector()

{

String s = "";

for (int i = 0; i < vect.Length; i++)

{

s = s + " " + vect[i];

}

return s;

}

}

class Matrix

{

private Vector[] matr;

public Matrix(int len)

{

matr = new Vector[len];

for (int i = 0; i < matr.Length; i++)

{

matr[i] = new Vector(matr.Length);

}

}

public Matrix(Matrix mat) {

matr = new Vector[mat.getLength()];

for (int i = 0; i < matr.Length; i++)

{

matr[i] = new Vector(mat.getString(i));

}

}

public Vector getString(int i)

{

return matr[i];

}

public int getElement(int i, int j)

{

return matr[i].getElement(j);

}

public void setElement(int i, int j, int func)

{

matr[i].setElement(j, func);

}

public void setString(int i, Vector func)

{

matr[i] = func;

}

public int getLength()

{

return matr.Length;

}

public void inputMatrix()

{

for (int i = 0; i < matr.Length; i++)

{

matr[i] = new Vector(matr.Length);

matr[i].inputVector();

}

}

public String outputMatrix()

{

String s = "";

for (int i = 0; i < matr.Length; i++)

{

s = s + matr[i].outputVector() + "\n";

}

return s;

}

}

}