1. Лістинг програми ПРГ1.

// kursova robota part 2

// MA=MB\*MC(Z\*T)

// Zaiats Yulia IO-01

// 13.04.2013

using System;

using System.Diagnostics;

using System.Threading;

using System.Collections.Generic;

using System.Text;

namespace k2\_y {

class ProgramA {

const int N = 900;

const int P = 4;

const int H = N/P;

static int[] T = new int[N];

static int[] Z = new int[N];

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

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

static int[,] MC = new int[N, N];

static int[,] MX = new int[N, N];

static int a = 0;

static EventWaitHandle Ev1 = new EventWaitHandle(false, EventResetMode.ManualReset);

static EventWaitHandle Ev2 = new EventWaitHandle(false, EventResetMode.ManualReset);

static EventWaitHandle Ev4 = new EventWaitHandle(false, EventResetMode.ManualReset);

static Mutex Mut = new Mutex(false);

private static readonly object lockobj = new object();

static Semaphore Sem1 = new Semaphore(0, 3);

static Semaphore Sem2 = new Semaphore(0, 3);

static Semaphore Sem3 = new Semaphore(0, 3);

static Semaphore Sem11 = new Semaphore(0, 3);

static Semaphore Sem21 = new Semaphore(0, 3);

static Semaphore Sem31 = new Semaphore(0, 3);

static Semaphore Sem41 = new Semaphore(0, 3);

public static void Main() {

Console.WriteLine("Main Started");

Thread Task1 = new Thread(T1);

Thread Task2 = new Thread(T2);

Thread Task3 = new Thread(T3);

Thread Task4 = new Thread(T4);

Stopwatch stopWatch = new Stopwatch();

Console.Write("Press enter to start");

Console.ReadLine();

stopWatch.Start();

Task1.Start();

Task2.Start();

Task3.Start();

Task4.Start();

Task1.Join();

Task2.Join();

Task3.Join();

Task4.Join();

stopWatch.Stop();

Console.Write(stopWatch.Elapsed+"\n");

Console.WriteLine("Main Finished");

}

public static void T1() {

int[,] MB1 = new int[N, N];

Console.WriteLine("T1 Started");

// input MT

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

{

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

MB[i,j] = 1;

}

// end of input

Sem1.Release(3);

// wait for other inputs

Sem2.WaitOne();

Sem3.WaitOne();

// copy

Mut.WaitOne();

MB1 = MB;

Mut.ReleaseMutex();

int tmp = 0;

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

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

MX[i,j] = 0;

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

MX[i,j] += MB1[i,k] \* MC[k,j];

}

tmp += Z[j] \* T[j];

}

lock (lockobj) {

a += tmp;

}

Sem11.Release(3);

Sem21.WaitOne();

Sem31.WaitOne();

Sem41.WaitOne();

Mut.WaitOne();

tmp = a;

Mut.ReleaseMutex();

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

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

MA[i,j] = MX[i,j]\*tmp;

// end calculation

Ev1.Set();

Console.WriteLine("T1 Finished");

}

public static void T2() {

int[,] MB2 = new int[N, N];

Console.WriteLine("T2 Started");

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

T[i] = 1;

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

MC[i,j] = 1;

}

//end of input

Sem2.Release(3);

// wait for other inputs

Sem1.WaitOne();

Sem3.WaitOne();

// copy

Mut.WaitOne();

MB2 = MB;

Mut.ReleaseMutex();

int tmp = 0;

for (int j = H; j < 2\*H; j++) {

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

MX[i,j] = 0;

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

MX[i,j] += MB2[i,k] \* MC[k,j];

}

tmp += Z[j] \* T[j];

}

lock (lockobj) {

a += tmp;

}

Sem21.Release(3);

Sem11.WaitOne();

Sem31.WaitOne();

Sem41.WaitOne();

Mut.WaitOne();

tmp = a;

Mut.ReleaseMutex();

for (int j = H; j < 2\*H; j++)

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

MA[i,j] = MX[i,j]\*tmp;

// end calculation

// wait for other threads

Ev2.Set();

Console.WriteLine("T2 Finished");

}

public static void T3() {

int[,] MB3 = new int[N, N];

Console.WriteLine("T3 Started");

// input MX, a

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

Z[i] = 1;

}

//end of input

Sem3.Release(3);

// wait for other inputs

Sem1.WaitOne();

Sem2.WaitOne();

// copy

Mut.WaitOne();

MB3 = MB;

Mut.ReleaseMutex();

int tmp = 0;

for (int j = 2\*H; j < 3\*H; j++) {

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

MX[i,j] = 0;

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

MX[i,j] += MB3[i,k] \* MC[k,j];

}

tmp += Z[j] \* T[j];

}

lock (lockobj) {

a += tmp;

}

Sem31.Release(3);

Sem11.WaitOne();

Sem21.WaitOne();

Sem41.WaitOne();

Mut.WaitOne();

tmp = a;

Mut.ReleaseMutex();

for (int j = 2\*H; j < 3\*H; j++)

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

MA[i,j] = MX[i,j]\*tmp;

// end calculation

Ev1.WaitOne();

Ev2.WaitOne();

Ev4.WaitOne();

// output

if (N <=8)

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

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

Console.Write(MA[i,j] + " ");

Console.Write("\n");

}

Console.WriteLine("T3 Finished");

}

public static void T4() {

int[,] MB4 = new int[N, N];

Console.WriteLine("T4 Started");

// wait for other inputs

Sem1.WaitOne();

Sem2.WaitOne();

Sem3.WaitOne();

// copy

Mut.WaitOne();

MB4 = MB;

Mut.ReleaseMutex();

int tmp = 0;

for (int j = 3\*H; j < N; j++) {

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

MX[i,j] = 0;

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

MX[i,j] += MB4[i,k] \* MC[k,j];

}

tmp += Z[j] \* T[j];

}

lock (lockobj) {

a += tmp;

}

Sem41.Release(3);

Sem11.WaitOne();

Sem21.WaitOne();

Sem31.WaitOne();

Mut.WaitOne();

tmp = a;

Mut.ReleaseMutex();

for (int j = 3\*H; j < N; j++)

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

MA[i,j] = MX[i,j]\*tmp;

// end calculation

Ev4.Set();

Console.WriteLine("T4 Finished");

}

}

}