НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №4

з дісципліни **«**Паралельні та розподілені обчислення**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Перевірив:

Корочкін О. В.

Київ – 2014 р.

ПОТОКИ В ЯЗЫКЕ С#

Цель работы: изучение средств языка C# для работы с потоками (процессами).

Выполнение работы: Разработать программу, содержащую п а р а л л е л ь н ы е потоки, каждый из которых реализует функцию F1, F2, F3 из лабораторной работы номер 1. Требования к созданию потоков и необходимые исследования программы описаны в лабораторной работе 2.

Необходимые теоретические сведения: язык C# обеспечивает возможность программиро-вания параллельных процессов с помощью потоков.

Теоретические сведения по программированию потоков и управлению ими в языке C#

можно найти в [21, 38, 43].

Лістинг програми

Файл ..\Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

/\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Laboratory work #4. \*

\* Threads in C# \*

\* \*

\* Task: \*

\* F1: 1.24 E = A + C \* (MA \* MZ) + B \*

\* F2: 2.14 MC = SORT(MA + MB \* MM) \*

\* F3: 3.18 p = MAX(SORT(MS) + MA\* MM) \*

\* \*

\* Autor: Kuzmenko Volodia \*

\* Group: IO-21 \*

\* Date: 1.10.2014 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

namespace PRO\_Lab4

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Input N");

String lineN = Console.ReadLine();

int n;

int.TryParse(lineN , out n);

Console.WriteLine("Input value");

String lineValue = Console.ReadLine();

int value;

int.TryParse(lineValue, out value);

Console.WriteLine();

Tasks tasks = new Tasks(n,value);

System.Int32 stackSizeT1 = 1000;

System.Int32 stackSizeT2 = 1000;

System.Int32 stackSizeT3 = 1000;

Thread t1 = new Thread(tasks.Task1, stackSizeT1);

Thread t2 = new Thread(tasks.Task2, stackSizeT2 );

Thread t3 = new Thread(tasks.Task3, stackSizeT3);

t1.Name = "Thread 1 to F1";

t1.Priority = ThreadPriority.Normal;

t2.Name = "Thread 2 to F2";

t2.Priority = ThreadPriority.Normal;

t3.Name = "Thread 3 to F3";

t3.Priority = ThreadPriority.Normal;

t1.Start();

t2.Start();

t3.Start();

Console.ReadLine();

}

}

}

Файл ..\Data.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

/\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Utils to work with vectors and matrix \*

\* \*

\* Autor: Kuzmenko Volodia \*

\* Group: IO-21 \*

\* Date: 1.10.2014 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

namespace PRO\_Lab4

{

class Data

{

private int N;

private int value;

protected Data(int n, int value)

{

this.N = n;

this.value = value;

}

public void setN(int n)

{

this.N = n;

}

public void setValue(int value)

{

this.value = value;

}

protected int[] vector\_input() {

int[] A = new int[N];

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

A[i] = value;

}

return A;

}

protected int[,] matrix\_input() {

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

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

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

MA[i,j] = value;

}

}

return MA;

}

protected void vector\_output(int[] vector) {

for (int j = 0; j < vector.Length; j++) {

Console.Write("|" + vector[j] + "|");

}

Console.WriteLine();

}

protected void matrix\_output(int[,] matrix) {

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

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

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

}

Console.WriteLine();

}

}

protected int[,] multMatr(int[,] MA, int[,] MZ) {

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

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

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

int s = 0;

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

s += MA[i,k] \* MZ[k,j];

MT[i,j] = s;

}

}

}

return MT;

}

protected int[] multingMatrixOnVector(int[,] MA,int[] A) {

int[] R = new int[N];

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

int s = 0;

for (int j = 0; j < A.Length; j++) {

s += MA[i,j] \* A[j];

}

R[i] = s;

}

return R;

}

protected int[] addVec(int[] B, int[] C) {

int[] K = new int[B.Length];

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

K[i] = B[i] + C[i];

}

return K;

}

protected int[,] addMatr(int[,] MA, int[,] MB) {

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

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

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

MM[i,j] = MA[i,j] + MB[i,j];

}

}

return MM;

}

protected int[] sortVector(int[] A) {

int[] B = A;

Array.Sort(B);

return B;

}

protected int[,] sortMatr(int[,] MA) {

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

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

{

MC[i] = new int[N];

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

{

MC[i][j] = MA[i,j];

}

}

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

MC[i] = sortVector(MC[i]);

}

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

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

{

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

{

MR[i, j] = MC[i][j];

}

}

return MR;

}

protected int maxElement(int[,] MD) {

int m = MD[0,0];

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

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

if (m < MD[i,j])

m = MD[i,j];

}

}

return m;

}

}

}

Файл ..\Tasls.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

/\*

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Tasks for calculating \*

\* \*

\* Autor: Kuzmenko Volodia \*

\* Group: IO-21 \*

\* Date: 1.10.2014 \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

namespace PRO\_Lab4

{

class Tasks : Data

{

public Tasks(int n, int value) : base(n, value)

{

}

public void Task1()

{

int []A,B,C,E;

int [,] MA,MZ;

Console.WriteLine("Input vector A in T1");

A = vector\_input();

Console.WriteLine("Output vector A from T1");

vector\_output(A);

Console.WriteLine("Input vector B in T1");

B = vector\_input();

Console.WriteLine("Output vector B from T1");

vector\_output(B);

Console.WriteLine("Input vector C in T1");

C = vector\_input();

Console.WriteLine("Output vector C from T1");

vector\_output(C);

Console.WriteLine("Input matrix MA in T1 ");

MA = matrix\_input();

Console.WriteLine("Output matrix MA from T1");

matrix\_output(MA);

Console.WriteLine("Input matrix MZ in T1");

MZ = matrix\_input();

Console.WriteLine("Output matrix MZ from T1");

matrix\_output(MZ);

Console.WriteLine("Calculation T1...");

Thread.Sleep(1000);

E = func1(A, B, C, MA, MZ);

Console.WriteLine("F1:");

vector\_output(E);

Console.WriteLine("Task 1 finished");

}

public void Task2()

{

Console.WriteLine("Task 2 started");

int[,] MA, MB, MC, MM;

Console.WriteLine("Input matrix MA in T2");

MA = matrix\_input();

Console.WriteLine("Output matrix MA from T2");

matrix\_output(MA);

Console.WriteLine("Input matrix MB in T2");

MB = matrix\_input();

Console.WriteLine("Output matrix MB from T2");

matrix\_output(MB);

Console.WriteLine("Input matrix MM in T2");

MM = matrix\_input();

Console.WriteLine("Output matrix MM from T2");

matrix\_output(MM);

Console.WriteLine("Calculation T2... ");

Thread.Sleep(5000);

MC = func2(MA, MB, MM);

Console.WriteLine("F2: ");

matrix\_output(MC);

Console.WriteLine("Task 2 finished");

}

public void Task3()

{

Console.WriteLine("Task 3 started");

int[,] MA, MS, MM;

int P;

Thread.Sleep(0);

Console.WriteLine("Input matrix MA in T3");

MA = matrix\_input();

Console.WriteLine("Output matrix MA from T3");

matrix\_output(MA);

Console.WriteLine("Input matrix MM in T3");

MM = matrix\_input();

Console.WriteLine("Output matrix MM from T3");

matrix\_output(MM);

Console.WriteLine("Input matrix MS in T3");

MS = matrix\_input();

Console.WriteLine("Output matrix MS from T3");

matrix\_output(MS);

Console.WriteLine("Calculation T3...");

Thread.Sleep(6000);

P = func3(MA, MM, MS);

Console.WriteLine("F3: ");

Console.WriteLine(P);

Console.WriteLine("Task 3 finished");

}

private int[] func1(int[] A, int[] B, int[] C, int[,] MA, int[,] MZ)

{

return addVec(addVec(multingMatrixOnVector(multMatr(MA, MZ), C), A), B);

}

private int[,] func2(int[,] MA, int[,] MB, int[,] MM)

{

return sortMatr(addMatr(multMatr(MB, MM), MA));

}

private int func3(int[,] MA, int[,] MM, int[,] MS)

{

return maxElement(addMatr(sortMatr(MS), multMatr(MA, MM)));

}

}

}