Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# Лабораторна робота № 4

з дисципліни «Паралельні і розподілені обчислення»

Виконав студент групи ІО-01 *Редько Олександр*

## Завдання



## Програма:

Source file: ..\Program.cs Sun Sep 30 15:51:58 2012

1 **using** System**;**

2 **using** System**.**Collections**.**Generic**;**

3 **using** System**.**Linq**;**

4 **using** System**.**Text**;**

5 **using** System**.**Threading**;**

6 **using** System**.**Threading**.**Tasks**;**

7

8 /\*\*

9 \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

10 \* \*

11 \* Laboratory work #4. Threads in C# \*

12 \* \*

13 \* Task: F1: E = A + B + C + D \* (MA \* MZ) \*

14 \* F2: MD = (MA \* MB) \* TRANS(MC) \*

15 \* F3: E = (MA \* MM) \* B + MB \* SORT(A) \*

16 \* \*

17 \* **@author** Red'ko Alexander \*

18 \* @group IO-01 \*

19 \* **@date** 30.09.12 \*

20 \* \*

21 \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

22 \*/

23

24 **namespace** pro\_lab4

25 **{**

26 class Program

27 **{**

28

29 static void Main**(**string**[]** args**)**

30 **{**

31 Console**.**WriteLine**(**"Enter N: "**);**

32 String line **=** Console**.**ReadLine**();**

33 int n**;**

34 int**.**TryParse**(**line**,** **out** n**);**

35 int value**;**

36 Console**.**WriteLine**(**"Enter value: "**);**

37 line **=** Console**.**ReadLine**();**

38 int**.**TryParse**(**line**,** **out** value**);**

39 Console**.**WriteLine**();**

40

41 Tasks tasks **=** **new** Tasks**(**n**,** value**);**

42

43 Thread t1 **=** **new** Thread**(**tasks**.**taskF1**);**

44 Thread t2 **=** **new** Thread**(**tasks**.**taskF2**);**

45 Thread t3 **=** **new** Thread**(**tasks**.**taskF3**);**

46

47 t1**.**Start**();**

48 t2**.**Start**();**

49 t3**.**Start**();**

50

51 Console**.**Read**();**

52 **}**

53 **}**

54 **}**

Source file: ..\Tasks.cs Sun Sep 30 15:51:58 2012

1 using System;

2 using System.Collections.Generic;

3 using System.Linq;

4 using System.Text;

5 using System.Threading;

6 using System.Threading.Tasks;

7

8 namespace pro\_lab4

9 {

10 class Tasks : Data

11 {

12

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

14 {

15 }

16

17 //-----------------------------------------------------------------

18 public void taskF1()

19 {

20 System.Console.WriteLine("Task F1 started");

21 Vector va, vb, vc, vd, ve;

22 Matrix ma, mz;

23 va = inputVector();

24 vb = inputVector();

25 vc = inputVector();

26 vd = inputVector();

27 ma = inputMatrix();

28 mz = inputMatrix();

29 ve = va + vb + vc + vd \* (ma \* mz);

30 outputVector(ve, "f1.txt");

31 System.Console.WriteLine("Task F1 ended");

32 }

33

34 //----------------------------------------------------------------

35 public void taskF2()

36 {

37 System.Console.WriteLine("Task F2 started");

38 Matrix ma, mb, mc, md;

39 ma = inputMatrix();

40 mb = inputMatrix();

41 mc = inputMatrix();

42 md = (ma \* mb) \* transpose(mc);

43 outputMatrix(md, "f2.txt");

44 System.Console.WriteLine("Task F2 ended");

45 }

46

47 //-----------------------------------------------------------------

48 public void taskF3()

49 {

50 System.Console.WriteLine("Task F3 started");

51 Vector va, vb, ve;

52 Matrix ma, mb, mm;

53 va = inputVector();

54 vb = inputVector();

55 ma = inputMatrix();

56 mb = inputMatrix();

57 mm = inputMatrix();

58 ve = vb \* (ma \* mb) + sort(va) \* mb;

59 outputVector(ve, "f3.txt");

60 System.Console.WriteLine("Task F3 ended");

61 }

62

63

64 private Matrix transpose(Matrix m)

65 {

66 for (int i = 0; i < m.size(); i++)

67 {

68 for (int j = i + 1; j < m.size(); j++)

69 {

70 int temp = m.get(i, j);

71 m.set(i, j, m.get(j, i));

72 m.set(j, i, temp);

73 }

74 }

75 return m;

76 }

77

78 private Vector sort(Vector vector)

79 {

80 int[] res = new int[vector.size()];

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

82 {

83 res[i] = vector.get(i);

84 }

85 Array.Sort(res);

86 Vector sortedVector = new Vector(res.Length);

87 for (int j = 0; j < sortedVector.size(); j++)

88 {

89 sortedVector.set(j, res[j]);

90 }

91 return sortedVector;

92 }

93

94

95 }

96 }

97

Source file: ..\Data.cs Sun Sep 30 15:51:58 2012

1 using System;

2 using System.Collections.Generic;

3 using System.Linq;

4 using System.Text;

5 using System.Threading.Tasks;

6

7 namespace pro\_lab4

8 {

9 class Data

10 {

11 private int n;

12 private int value;

13

14 protected Data (int n, int value)

15 {

16 this.setN(n);

17 this.setValue(value);

18 }

19

20 protected Vector inputVector()

21 {

22 Vector vector = new Vector(n);

23 for (int i = 0; i < n; i++)

24 {

25 vector.set(i, value);

26 }

27 return vector;

28 }

29

30 protected Matrix inputMatrix()

31 {

32 Matrix matrix = new Matrix(n);

33 for (int i = 0; i < n; i++)

34 {

35 for (int j = 0; j < n; j++)

36 {

37 matrix.set(i, j, value);

38 }

39 }

40 return matrix;

41 }

42

43 protected void outputVector(Vector vector, String nameOfFile) {

44 if (vector.size() < 5)

45 {

46 Console.WriteLine(vector.toString());

47 }

48 else

49 {

50 System.IO.File.WriteAllText(@nameOfFile, vector.toString());

51 }

52

53 }

54

55 protected void outputMatrix(Matrix matrix, String nameOfFile) {

56 if(matrix.size() < 5)

57 {

58 Console.WriteLine(matrix.toString());

59 }

60 else

61 {

62 System.IO.File.WriteAllText(@nameOfFile, matrix.toString());

63 }

64 }

65

66 protected void setN(int n)

67 {

68 if (n > 0)

69 {

70 this.n = n;

71 }

72 }

73

74 protected void setValue(int value)

75 {

76 this.value = value;

77 }

78

79 protected int getValue()

80 {

81 return value;

82 }

83 }

84 }

Source file: ..\Vector.cs Sun Sep 30 15:51:58 2012

1 using System;

2 using System.Collections.Generic;

3 using System.Linq;

4 using System.Text;

5 using System.Threading.Tasks;

6

7 namespace pro\_lab4

8 {

9 class Vector

10 {

11 private int[] array;

12

13 public Vector(int n)

14 {

15 array = new int[n];

16 }

17

18 public void set(int index, int value)

19 {

20 array[index] = value;

21 }

22

23 public int get(int index)

24 {

25 return array[index];

26 }

27

28 public int size()

29 {

30 return array.Length;

31 }

32

33 public String toString()

34 {

35 String res = "";

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

37 {

38 res += " " + array[i];

39 }

40 return res;

41 }

42

43 // Overloading '+' operator:

44 public static Vector operator +(Vector left, Vector right)

45 {

46 Vector result = new Vector(left.size());

47 for (int i = 0; i < result.size(); i++)

48 {

49 result.set(i, left.get(i) + right.get(i));

50 }

51 return result;

52 }

53

54 // Overloading '\*' operator:

55 public static Vector operator \*(Vector left, Matrix right)

56 {

57 Vector result = new Vector(left.size());

58 for (int i = 0; i < left.size(); i++)

59 {

60 result.set(i, 0);

61 for (int j = 0; j < left.size(); j++)

62 {

63 result.set(i, result.get(i) + left.get(j) \* right.get(j, i));

64 }

65 }

66 return result;

67 }

68

69 }

70 }

Source file: ..\Matrix.cs Sun Sep 30 15:51:58 2012

1 using System;

2 using System.Collections.Generic;

3 using System.Linq;

4 using System.Text;

5 using System.Threading.Tasks;

6

7 namespace pro\_lab4

8 {

9 class Matrix

10 {

11 private Vector[] vector;

12

13 public Matrix(int n)

14 {

15 vector = new Vector[n];

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

17 {

18 vector[i] = new Vector(n);

19 }

20 }

21

22 public void set(int n, int m, int val)

23 {

24 vector[n].set(m, val);

25 }

26

27 public int get(int n, int m)

28 {

29 return vector[n].get(m);

30 }

31

32 public Vector get(int index)

33 {

34 return vector[index];

35 }

36

37 public int size()

38 {

39 return vector.Length;

40 }

41

42 public String toString()

43 {

44 String res = "";

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

46 {

47 res += vector[i].toString();

48 if (i != vector.Length - 1)

49 {

50 res += "\n";

51 }

52 }

53 return res;

54 }

55

56 // Overloading '\*' operator:

57 public static Matrix operator\*(Matrix left, Matrix right)

58 {

59 Matrix result = new Matrix(left.size());

60 for (int i = 0; i < left.size(); i++)

61 {

62 for (int j = 0; j < left.size(); j++)

63 {

64 result.set(i, j, 0);

65 for (int y = 0; y < left.size(); y++)

66 {

67 result.set(i, j, result.get(i, j) + left.get(i, y)

68 \* right.get(y, j));

69 }

70 }

71 }

72 return result;

73 }

74 }

75 }