Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

# Лабораторна робота № 3

з дисципліни «Паралельні і розподілені обчислення»

Виконав студент групи ІО-01 *Редько Олександр*

## Завдання



## Програма:

Source file: ..\Main.java Tue Sep 25 20:51:58 2012

1 import java.util.Scanner;

2

3 /\*\*

4 \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

5 \* \*

6 \* Laboratory work #3. Threads in Java \*

7 \* \*

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

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

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

11 \* \*

12 \* @author Red'ko Alexander \*

13 \* @group IO-01 \*

14 \* @date 23.09.12 \*

15 \* \*

16 \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

17 \*/

18 public class Main {

19

20 public static void main(String[] args) {

21

22 Scanner sc = new Scanner(System.in);

23 System.out.print("Enter N: ");

24 int n = sc.nextInt();

25 System.out.print("Enter value: ");

26 int value = sc.nextInt();

27 System.out.println();

28

29 Thread t1 = new Thread(new RunnableF1(n, value));

30 t1.setPriority(Thread.NORM\_PRIORITY);

31 Thread t2 = new Thread(new RunnableF2(n, value));

32 t2.setPriority(Thread.NORM\_PRIORITY);

33 Thread t3 = new Thread(new RunnableF3(n, value));

34 t3.setPriority(Thread.NORM\_PRIORITY);

35

36 t1.start();

37 t2.start();

38 t3.start();

39 }

40 }

Source file: ..\RunnableF1.java Tue Sep 25 20:51:58 2012

1 import java.io.File;

2

3

4 /\*\*

5 \*

6 \* Task F1

7 \*

8 \*/

9 public class RunnableF1 extends Data implements Runnable {

10

11 public RunnableF1(int n, int value) {

12 super(n, value);

13 }

14

15 @Override

16 public void run() {

17 System.out.println("Task F1 started");

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

19 Matrix ma, mz;

20 va = inputVector();

21 vb = inputVector();

22 vc = inputVector();

23 vd = inputVector();

24 ma = inputMatrix();

25 mz = inputMatrix();

26 ve = f1(va, vb, vc, vd, ma, mz);

27 outputVector(ve, new File("f1.txt"));

28 System.out.println("Task F1 ended");

29 }

30

31 /\*\*

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

33 \*

34 \* @param va

35 \* @param vb

36 \* @param vc

37 \* @param vd

38 \* @param ma

39 \* @param mz

40 \* @return Vector E

41 \*/

42 private Vector f1(final Vector va, final Vector vb, final Vector vc,

43 final Vector vd, final Matrix ma, final Matrix mz) {

44 return add(add(add(va, vb), vc), mult(vd, mult(ma, mz)));

45

46 }

47

48 }

Source file: ..\RunnableF2.java Tue Sep 25 20:51:58 2012

1 import java.io.File;

2

3

4 /\*\*

5 \*

6 \* Task F2

7 \*

8 \*/

9 public class RunnableF2 extends Data implements Runnable {

10

11 public RunnableF2(int n, int value) {

12 super(n, value);

13 }

14

15 @Override

16 public void run() {

17 System.out.println("Task F2 started");

18 Matrix ma, mb, mc, md;

19 ma = inputMatrix();

20 mb = inputMatrix();

21 mc = inputMatrix();

22 md = inputMatrix();

23 md = f2(ma, mb, mc);

24 outputMatrix(md, new File("f2.txt"));

25 System.out.println("Task F2 ended");

26 }

27

28 /\*\*

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

30 \* @param ma

31 \* @param mb

32 \* @param mc

33 \* @return Matrix MD

34 \*/

35 private Matrix f2(final Matrix ma, final Matrix mb, final Matrix mc) {

36 return mult(mult(ma, mb), transpose(mc));

37 }

38

39 private Matrix transpose(Matrix m) {

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

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

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

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

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

45 }

46 }

47 return m;

48 }

49

50 }

Source file: ..\RunnableF3.java Tue Sep 25 20:51:58 2012

1 import java.io.File;

2 import java.util.Arrays;

3

4 /\*\*

5 \*

6 \* Task F3

7 \*

8 \*/

9 public class RunnableF3 extends Data implements Runnable {

10

11 public RunnableF3(int n, int value) {

12 super(n, value);

13 }

14

15 @Override

16 public void run() {

17 System.out.println("Task F3 started");

18 Vector va, vb, ve;

19 Matrix ma, mb, mm;

20 va = inputVector();

21 vb = inputVector();

22 ma = inputMatrix();

23 mb = inputMatrix();

24 mm = inputMatrix();

25 ve = f3(va, vb, ma, mb, mm);

26 outputVector(ve, new File("f3.txt"));

27 System.out.println("Task F3 ended");

28 }

29

30 /\*\*

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

32 \* @param va

33 \* @param vb

34 \* @param ma

35 \* @param mb

36 \* @param mm

37 \* @return Vector E

38 \*/

39 private Vector f3(final Vector va, final Vector vb,

40 final Matrix ma, final Matrix mb, final Matrix mm) {

41 return add(mult(vb, mult(ma, mm)), mult(sort(va), mb));

42 }

43

44 private Vector sort(final Vector va) {

45 int[] res = new int[va.size()];

46 for(int i = 0; i < res.length; i++) {

47 res[i] = va.get(i);

48 }

49 Arrays.sort(res);

50 Vector sortedVector = new Vector(res.length);

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

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

53 }

54 return sortedVector;

55 }

56

57 }

Source file: ..\Data.java Tue Sep 25 20:51:58 2012

1 import java.io.BufferedWriter;

2 import java.io.File;

3 import java.io.FileWriter;

4 import java.io.IOException;

5 import java.io.PrintWriter;

6

7

8 /\*\*

9 \*

10 \* Class with matrix and vector operations

11 \*

12 \*/

13 public class Data {

14

15 private int n;

16 private int value;

17

18 protected Data (int n, int value) {

19 this.setN(n);

20 this.setValue(value);

21 }

22

23 protected Vector inputVector() {

24 Vector vector = new Vector(n);

25 for(int i = 0; i < n; i++) {

26 vector.set(i, value);

27 }

28 return vector;

29 }

30

31 protected Matrix inputMatrix() {

32 Matrix matrix = new Matrix(n);

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

34 for(int j = 0; j < n; j++) {

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

36 }

37 }

38 return matrix;

39 }

40

41 protected void outputVector(Vector vector, File file) {

42 if(vector.size() < 5) {

43 System.out.print(vector.toString());

44 }

45 else {

46 try {

47 this.save(file, vector.toString());

48 } catch (IOException e) {

49 // TODO Auto-generated catch block

50 e.printStackTrace();

51 }

52 }

53 }

54

55 protected void outputMatrix(Matrix matrix, File file) {

56 if(matrix.size() < 5) {

57 System.out.print(matrix.toString());

58 }

59 else {

60 try {

61 this.save(file, matrix.toString());

62 } catch (IOException e) {

63 // TODO Auto-generated catch block

64 e.printStackTrace();

65 }

66 }

67 }

68

69 protected void setN(int n) {

70 try {

71 if(n < 0)

72 throw new Exception("Positive Number Required");

73 this.n = n;

74 }

75 catch(Exception e) {

76 System.out.println("Error: " + e.getMessage());

77 }

78 }

79

80 protected int getValue() {

81 return value;

82 }

83

84 protected void setValue(int value) {

85 this.value = value;

86 }

87

88

89 protected static Vector add(final Vector left, final Vector right) {

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

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

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

93 }

94 return result;

95 }

96

97

98 protected static Vector mult(final Vector left, final Matrix right) {

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

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

101 result.set(i, 0);

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

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

104 }

105 }

106 return result;

107 }

108

109 protected static Matrix mult(final Matrix left, final Matrix right) {

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

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

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

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

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

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

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

117 }

118 }

119 }

120 return result;

121 }

122

123

124 private void save(File file, final String data)

125 throws IOException {

126 PrintWriter pw = new PrintWriter(new BufferedWriter(

127 new FileWriter(file)));

128 pw.println(data);

129 pw.close();

130 }

132

131 }

Source file: ..\Vector.java Tue Sep 25 20:51:58 2012

1 public class Vector {

2

3 private int[] array;

4

5 public Vector(int n) {

6 array = new int[n];

7 }

8

9 public void set(int index, int value) {

10 array[index] = value;

11 }

12

13 public int get(int index) {

14 return array[index];

15 }

16

17 public int size() {

18 return array.length;

19 }

20

21 public String toString() {

22 String res = "";

23 for (int i = 0; i < array.length; i++){

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

25 }

26 res += "\n";

27 return res;

28 }

29

30 }

Source file: ..\Matrix.java Tue Sep 25 20:51:58 2012

1 public class Matrix {

2

3 private Vector[] array;

4

5 public Matrix(int n) {

6 array = new Vector[n];

7 for (int i = 0; i < array.length; i++){

8 array[i] = new Vector(n);

9 }

10 }

11

12 public void set(int n, int m, int val) {

13 array[n].set(m, val);

14 }

15

16 public int get(int n, int m) {

17 return array[n].get(m);

18 }

19

20 public Vector get(int index) {

21 return array[index];

22 }

23

24 public int size() {

25 return array.length;

26 }

27

28 public String toString() {

29 String res = "";

30 for (int i = 0; i < array.length; i++){

31 res += array[i];

32 }

33 return res;

34 }

35

36

37 }