Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №7**

*по курсу*

*«Паралельні та розподілені обчислення»*

Виконав:

студент III курсу

гр. IO-82

Куцовол В.В.

Київ 2010

1 /\*

2 \* @author Куцовол Віктор

3 \* ІІІ курс ФІОТ

4 \* гр. ІО-82

5 \* Лаборараторна робота №4

6 \* F1: 1.18 d=(A\*B)-(C\*B)

7 \* F2: 2.19 v=MAX(MA+MB\*MC)

8 \* F3: 3.20 R=SORT(B-C)\*SORT(MA\*MB)

9 \*/

10

11import mpi.\*;

12

13public class Lab7 {

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

15 MPI.Init(args);

16 int rank = MPI.COMM\_WORLD.Rank();

17 if (rank == 0) {

18 task1(60, 1, 1, 1);

19 }

20 if (rank == 1) {

21 task2(60, 1, 1, 1);

22 }

23 if (rank == 2) {

24 task3(60, 1, 1, 1, 1);

25 }

26 MPI.Finalize();

27 }

28

29 public static void task1(int n, int a, int b, int c) {

30 System.out.println("Task1 started");

31 int[] A = new int[n];

32 int[] B = new int[n];

33 int[] C = new int[n];

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

35 A[i] = a;

36 B[i] = b;

37 C[i] = c;

38 }

39 int res = Functions.f1(A, B, C);

40 System.out.println("F1: " + res);

41 System.out.println("Task1 finished");

42 }

43

44 public static void task2(int n, int ma, int mb, int mc) {

45 System.out.println("Task2 started");

46 int[][] MA = new int[n][n];

47 int[][] MB = new int[n][n];

48 int[][] MC = new int[n][n];

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

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

51 MA[i][j] = ma;

52 MB[i][j] = mb;

53 MC[i][j] = mc;

54 }

55 }

56 int res = Functions.f2(MA, MB, MC);

57 System.out.println("F2: " + res);

58 System.out.println("Task2 finished");

59 }

60

61 public static void task3(int n, int b, int c, int ma, int mb) {

62 System.out.println("Task3 started");

63 int[] B = new int[n];

64 int[] C = new int[n];

65 int[][] MA = new int[n][n];

66 int[][] MB = new int[n][n];

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

68 B[i] = b;

69 C[i] = c;

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

71 MA[i][j] = ma;

72 MB[i][j] = mb;

73 }

74 }

75 int[] res = Functions.f3(B, C, MA, MB);

76 System.out.println("F3:");

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

78 System.out.print(res[i] + " ");

79 System.out.println("\nTask3 finished");

80 }

1 import java.util.Arrays;

2

3 public class Functions {

4

5 public static int f1(int[] A, int[] B, int[] C) {

6 return mult(A, B) - mult(C, B);

7 }

8

9 public static int f2(int[][] MA, int[][] MB, int[][] MC) {

10 return maxElMatr(add(MA, multMatrix(MB, MC)));

11 }

12

13 public static int[] f3(int[] B, int[] C, int[][] MA, int[][] MB) {

14 int[] arr = sub(B, C);

15 Arrays.sort(arr);

16 return multMonV(sortMatrix(multMatrix(MA, MB)), arr);

17 }

18

19 private static int mult(int[] A, int[] B) {

20 int res = 0;

21 for (int i = 0; i < A.length; i++)

22 res += A[i] \* B[i];

23 return res;

24 }

25

26 private static int[][] add(int[][] MA, int[][] MB) {

27 int[][] res = new int[MA[0].length][MA[0].length];

28 for (int i = 0; i < MA[0].length; i++)

29 for (int j = 0; j < MA[0].length; j++) {

30 res[i][j] = MA[i][j] + MB[i][j];

31 }

32 return res;

33 }

34

35 private static int[][] multMatrix(int[][] MB, int[][] MC) {

36 int[][] res = new int[MB[0].length][MB[0].length];

37 int s;

38 for (int i = 0; i < MB[0].length; i++)

39 for (int j = 0; j < MB[0].length; j++) {

40 s = 0;

41 for (int k = 0; k < MB[0].length; k++)

42 s = s + MB[i][k] \* MC[k][j];

43 res[i][j] = s;

44 }

45 return res;

46 }

47

48 private static int maxElMatr(int[][] MA) {

49 int res = MA[0][0];

50 for (int i = 0; i < MA[0].length; i++)

51 for (int j = 0; i < MA[0].length; i++)

52 if (res < MA[i][j])

53 res = MA[i][j];

54 return res;

55 }

56

57 private static int[] sub(int[] B, int[] C) {

58 int[] res = new int[B.length];

59 for (int i = 0; i < B.length; i++)

60 res[i] = B[i] - C[i];

61 return res;

62 }

63

64 private static int[][] sortMatrix(int[][] MA) {

65 int[][] res = new int[MA[0].length][MA[0].length];

66 int buf, k;

67 for (int i = 0; i < MA[0].length; i++) {

68 for (int j = 0; j < MA[0].length; j++) {

69 k = j;

70 for (int l = 0; l < MA[0].length; l++)

71 if (MA[i][k] < MA[i][l])

72 k = l;

73 buf = MA[i][k];

74 MA[i][k] = MA[i][j];

75 MA[i][j] = buf;

76 }

77 }

78 return res;

79 }

80

81 private static int[] multMonV(int[][] MA, int[] A) {

82 int[] res = new int[A.length];

83 int s;

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

85 s = 0;

86 for (int j = 0; j < A.length; j++)

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

88 res[i] = s;

89 }

90 return res;

91 }