**Национальный технический университет Украины**

**«Киевский Политехнический Институт»**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

**Лабораторная работа №3**

по курсу: Параллельные и распределённые вычисления

Выполнила студентка

III курса ФИВТ

группы ИО-91

Косейкина Анна Сергеевна

Киев 2011

**Исполняющий класс**

lab3.java

1 /\*Косейкина Анна Сергеевна, группа ИО-91

2 Лаб.3. Потоки в JAVA

3 F1: 1.15 d = MAX(A + B + C)

4 F2: 2.27 MD = MA\*MB\*TRANS(MC)

5 F3: 3.21 W = SORT(B\*MD)\*(MA -MB)\*/

6 **package** lab3;

7 **public class** lab3 {

8 **public static void** main(String[] args) {

9 System.*out*.println("lab3 started");

10 F1\_TASK f1 = **new** F1\_TASK("T1", 4, 1000, 1);

11 F2\_TASK f2 = **new** F2\_TASK("T2", 1, 1000, 1);

12 F3\_TASK f3 = **new** F3\_TASK(1000, 1);

13 f1.start();

14 f2.start();

15 Thread t = **new** Thread(f3);

16 t.setName("T3");

17 t.setPriority(1);

18 t.start();

19 **try** {

20 f1.join();

21 f2.join();

22 t.join();

23 } **catch** (InterruptedException e) {

24 e.printStackTrace();

25 }

26 System.*out*.println("lab3 finished");

27 }

28 }

**Класс первой функции**

F1\_TASK.java

1 /\*Косейкина Анна Сергеевна, группа ИО-91

2 Лаб.3. Потоки в JAVA

3 F1: 1.15 d = MAX(A + B + C)

4 F2: 2.27 MD = MA\*MB\*TRANS(MC)

5 F3: 3.21 W = SORT(B\*MD)\*(MA -MB)\*/

6 **package** lab3;

7

8 **import** java.io.BufferedWriter;

9 **import** java.io.FileWriter;

10 **import** java.io.IOException;

11

12 **public class** F1\_TASK **extends** Thread{

13 **public** F1\_TASK(String NAME, **int** PRIORITY, **int** N, **int** VALUE){

14 **this**.setPriority(PRIORITY);

15 **this**.setName(NAME);

16 A = **new int**[N];

17 B = **new int**[N];

18 C = **new int**[N];

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

20 A[i] = VALUE;

21 B[i] = VALUE;

22 C[i] = VALUE;

23 }

24 }

25 **public** F1\_TASK(String NAME, **int** PRIORITY, **int**[] A, **int**[] B, **int**[] C){

26 **this**.setPriority(PRIORITY);

27 **this**.setName(NAME);

28 **this**.A = A;

29 **this**.B = B;

30 **this**.C = C;

31 }

32 **private int**[] PLUS(**int**[] ARG1, **int**[] ARG2){

33 **int**[] RESULT = **new int**[ARG1.length];

34 **for**(**int** i=0; i<ARG1.length; i++)

35 RESULT[i] = ARG1[i] + ARG2[i];

36 **return** RESULT;

37 }

38 **private int** MAX(**int** [] A){

39 **int** res=A[0];

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

41 **if** (A[i]>res){

42 res=A[i];

43 }

44 }

45 **return** res;

46 }

47 **public void** run(){

48 System.*out*.println(**this**.getName()+" started");

49 A=PLUS(A,PLUS(B,C));

50 d = MAX(A);

51 **try** {

52 BufferedWriter outfile = **new** BufferedWriter(**new** FileWriter("F1.txt"));

53 outfile.write("d = MAX(A + B + C)"+d);

54 outfile.close();

55 }

56 **catch** (IOException e) {

57 e.printStackTrace();

58 }

59 **try** {

60 *sleep*(1000);

61 } **catch** (InterruptedException e) {

62 e.printStackTrace();

63 }

64 System.*out*.println(**this**.getName()+" finished");

65 }

66 **public int** getD() {

67 **return** d;

68 }

69 **private int**[] A;

70 **private int**[] B;

71 **private int**[] C;

72 **private int** d;

73 }

74

**Класс второй функции**

1 /\*Косейкина Анна Сергеевна, группа ИО-91

2 Лаб.3. Потоки в JAVA

3 F1: 1.15 d = MAX(A + B + C)

4 F2: 2.27 MD = MA\*MB\*TRANS(MC)

5 F3: 3.21 W = SORT(B\*MD)\*(MA -MB)\*/

6 **package** lab3;

7

8 **import** java.io.BufferedWriter;

9 **import** java.io.FileWriter;

10 **import** java.io.IOException;

11

12 **public class** F2\_TASK **extends** Thread{

13 **public** F2\_TASK(String NAME, **int** PRIORITY, **int** N, **int** VALUE){

14 **this**.setPriority(PRIORITY);

15 **this**.setName(NAME);

16 MA = **new int**[N][N];

17 MB = **new int**[N][N];

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

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

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

21 MA[i][j] = VALUE;

22 MB[i][j] = VALUE;

23 MC[i][j] = VALUE;

24 }

25 }

26 **public** F2\_TASK(String NAME, **int** PRIORITY, **int**[][] MA, **int**[][] MB, **int**[][] MC){

27 **this**.setPriority(PRIORITY);

28 **this**.setName(NAME);

29 **this**.MA = MA;

30 **this**.MB = MB;

31 **this**.MC = MC;

32 }

33 **private int**[][] MUL(**int**[][] ARG1, **int**[][] ARG2){

34 **int**[][] RESULT = **new int**[ARG1.length][ARG1.length];

35 **for**(**int** i=0; i<ARG1.length; i++)

36 **for**(**int** j=0; j<ARG1.length; j++)

37 **for**(**int** k=0; k<ARG1.length; k++)

38 RESULT[i][j] += ARG1[i][k] \* ARG2[k][j];

39 **return** RESULT;

40 }

41 **private int**[][] TRANS(**int**[][] M){

42 **int** [][] Res = **new int**[M.length][M.length];

43 **for** (**int** i=0;i<M.length;i++){

44 **for** (**int** j=0;j<M.length;j++){

45 Res[i][j]=M[j][i];

46 }

47 }

48 **return** Res;

49 }

50 **public void** run(){

51 System.*out*.println(**this**.getName()+" started");

52 MD = MUL(MA,MUL(MB,TRANS(MC)));

53 **try** {

54 BufferedWriter outfile = **new** BufferedWriter(**new** FileWriter("F2.txt"));

55 outfile.write("MD = MA\*MB\*TRANS(MC)");

56 outfile.newLine();

57 **for**(**int** i=0; i<MD.length; i++){

58 **for**(**int** j=0; j<MD.length; j++)

59 outfile.write(MD[i][j]+" ");

60 outfile.newLine();

61 }

62 outfile.close();

63 }

64 **catch** (IOException e) {

65 e.printStackTrace();

66 }

67 **try** {

68 *sleep*(1000);

69 } **catch** (InterruptedException e) {

70 e.printStackTrace();

71 }

72 System.*out*.println(**this**.getName()+" finished");

73 }

74 **public int**[][] getMD() {

75 **return** MD;

76 }

77 **private int**[][] MA;

78 **private int**[][] MB;

79 **private int**[][] MC;

80 **private int**[][] MD;

81 }

**Класс третьей функции**

1 /\*Косейкина Анна Сергеевна, группа в ИО-91

2 Лаб.3. Потоки в JAVA

3 F1: 1.15 d = MAX(A + B + C)

4 F2: 2.27 MD = MA\*MB\*TRANS(MC)

5 F3: 3.21 W = SORT(B\*MD)\*(MA -MB)\*/

6 **package** lab3;

7

8 **import** java.io.BufferedWriter;

9 **import** java.io.FileWriter;

10 **import** java.io.IOException;

11

12 **public class** F3\_TASK **implements** Runnable{

13 **public** F3\_TASK(**int** N, **int** VALUE){

14 B = **new int**[N];

15 MA = **new int**[N][N];

16 MB = **new int**[N][N];

17 MD = **new int**[N][N];

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

19 B[i] = VALUE;

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

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

22 MA[i][j] = VALUE;

23 MB[i][j] = VALUE;

24 MD[i][j] = VALUE;

25 }

26 }

27 **public** F3\_TASK(**int**[][] MA, **int**[][] MB, **int**[][] MD, **int**[] B, **int**[] W){

28 **this**.MA = MA;

29 **this**.MB = MB;

30 **this**.MD = MD;

31 **this**.W = W;

32 **this**.B = B;

33 }

34 **private int**[][] MINUS(**int**[][] ARG1, **int**[][] ARG2){

35 **int**[][] RESULT = **new int**[ARG1.length][ARG1.length];

36 **for**(**int** i=0; i<ARG1.length; i++)

37 **for**(**int** j=0; j<ARG1.length; j++)

38 RESULT[i][j] = ARG1[i][j] - ARG2[i][j];

39 **return** RESULT;

40 }

41 **private int**[] MUL(**int**[][] ARG1, **int**[] ARG2){

42 **int**[] RESULT = **new int**[ARG2.length];

43 **for**(**int** i=0; i<ARG1.length; i++)

44 **for**(**int** j=0; j<ARG1.length; j++)

45 RESULT[i] += ARG1[i][j] \* ARG2[j];

46 **return** RESULT;

47 }

48 **private int**[] SORT(**int**[] ARG){

49 **for**(**int** i=0; i<ARG.length-1; i++)

50 **for**(**int** j=i+1; j<ARG.length; j++)

51 **if**(ARG[i] > ARG[j]){

52 **int** t = ARG[j];

53 ARG[j] = ARG[i];

54 ARG[i] = t;

55 }

56 **return** ARG;

57 }

58 **public void** run(){

59 System.*out*.println(Thread.*currentThread*().getName()+" started");

60 W = MUL(MINUS(MA,MB),SORT(MUL(MD,B)));

61 **try** {

62 BufferedWriter outfile = **new** BufferedWriter(**new** FileWriter("F3.txt"));

63 outfile.write("W = SORT(B\*MD)\*(MA -MB)");

64 outfile.newLine();

65 **for**(**int** i=0; i<W.length; i++)

66 outfile.write(W[i]+" ");

67 outfile.close();

68 }

69 **catch** (IOException e) {

70 e.printStackTrace();

71 }

72 **try** {

73 Thread.*sleep*(1000);

74 } **catch** (InterruptedException e) {

75 e.printStackTrace();

76 }

77 System.*out*.println(Thread.*currentThread*().getName()+" finished");

78 }

79 **public int**[] getW() {

80 **return** W;

81 }

82 **private int**[] W;

83 **private int**[] B;

84 **private int**[][] MA;

85 **private int**[][] MB;

86 **private int**[][] MD;

87 }