**Национальный технический университет Украины**

**«Киевский Политехнический Институт»**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

**Лабораторная работа №1**

по курсу: Параллельные и распределённые вычисления

Тема: Программирование для ПКС с ОП. Ада. Семафоры

Выполнил студент

III курса ФИВТ

группы ИО-91

Водопьянов Дмитрий Вадимович

Киев 2012

Задание на лабораторную работу:

Математическая задача:

Структура ПКС:

ОП

2

1

a, MC, MR MB, α

Средства программирования:

Язык АДА, семафоры

Выполнение работы.

Этап 1. Разработка параллельного математического алгоритма

1. ai=max(MB\*MCH – MRH\*α)
2. a=max(a, ai)

ОР: МB, a, α

Этап 2. Разработка алгоритмов задач

Т1

1. Ввод данных MC, MR
2. Сигнал Т2 о вводе S2,1
3. Ждать сигнал от Т2 о вводе данных W2,1
4. Копия MB1=MB КУ
5. Копия α1=α КУ
6. Счёт a1= max(MB\*MCH – MRH\*α1)
7. Счёт a=max(a, a1) КУ
8. Ждать завершения счёта а в Т2 W2,2
9. Вывод а

Т2

1. Ввод данных МB, α
2. Сигнал Т1 о вводе данных S1,1
3. Ждать сигнал от Т1 о вводе данных W1,1
4. Копия MB2=MB КУ
5. Копия α2=α КУ
6. Счёт a2= max(MB\*MCH – MRH\*α2)
7. Счёт a=max(a, a2) КУ
8. Сигнал Т1 о завершении счёта S1,2

Этап 3. Разработка схемы взаимодействия задач

S3

S1 S2

S4 S5

S6

Этап 4. Разработка программы

Source file: ..\lab1.ada Wed Feb 08 15:36:52 2012

1 -- Лаб1. Ада. Семафоры

2 -- a=max(MB\*MC-MR\*alpha)

3 -- Водопьянов Дмитрий Вадимович, группа IО-91

4 -- 08.02.2012

5 -----------------------------------------------

6 with Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control;

7 use Ada.Text\_IO, Ada.Integer\_Text\_IO, Ada.Synchronous\_Task\_Control;

8

9 procedure Lab1 is

10 N : integer := 3;

11 P : integer := 2;

12 H : integer := N/P;

13 type Vector is array (1..N) of integer;

14 type Matrix is array (1..N) of Vector;

15 a : integer := -32767;

16 alpha : integer;

17 MB, MC, MR : Matrix;

18 S0, S1, S2, S3, S4, S5, S6 : Suspension\_Object;

19

20 procedure Start is

21 task T1;

22 task body T1 is

23 a1 : integer;

24 alpha1 : integer;

25 MB1, MT1 : Matrix;

26 begin

27 put ("Task 1 started");

28 NEW\_LINE;

29 for i in 1..N LOOP

30 for j in 1..N loop

31 MC(i)(j) := 1;

32 MR(i)(j) := 1;

33 end loop;

34 end loop;

35 Set\_true (S4);

36 Suspend\_Until\_True (S5);

37 Suspend\_Until\_True (S2);

38 MB1 := MB;

39 Set\_True (S2);

40 Suspend\_Until\_True (S3);

41 alpha1 := alpha;

42 Set\_True (S3);

43 a1 := -32767;

44 for i in 1..H loop

45 for j in 1..N loop

46 MT1(i)(j) := 0;

47 for k in 1..N loop

48 MT1(i)(j) := MT1(i)(j) + MC(i)(j)\*MB1(k)(j);

49 end loop;

50 MT1(i)(j) := MT1(i)(j) - MR(i)(j)\*alpha1;

51 end loop;

52 end loop;

53 for i in 1..H loop

54 for j in 1..N loop

55 if (a1<MT1(i)(j)) then

56 a1 := MT1(i)(j);

57 end if;

58 end loop;

59 end loop;

60 Suspend\_Until\_True (S1);

61 if (a<a1) then

62 a := a1;

63 end if;

64 Set\_True (S1);

65 Suspend\_Until\_True (S6);

66 put (a);

67 new\_line;

68 put ("Task 1 finished");

69 end T1;

70 task T2;

71 task body T2 is

72 a2 : integer;

73 alpha2 : integer;

74 MB2, MT2 : Matrix;

75 begin

76 put ("Task 2 started");

77 NEW\_LINE;

78 for i in 1..N LOOP

79 for j in 1..N loop

80 MB(i)(j) := 1;

81 alpha := 1;

82 end loop;

83 end loop;

84 Set\_true (S5);

85 Suspend\_Until\_True (S4);

86 Suspend\_Until\_True (S3);

87 alpha2 := alpha;

88 Set\_True (S3);

89 Suspend\_Until\_True (S2);

90 MB2 := MB;

91 Set\_True (S2);

92 a2 := -32767;

93 for i in H..N loop

94 for j in 1..N loop

95 MT2(i)(j) := 0;

96 for k in 1..N loop

97 MT2(i)(j) := MT2(i)(j) + MC(i)(j)\*MB2(k)(j);

98 end loop;

99 MT2(i)(j):= MT2(i)(j) - MR(i)(j)\*alpha2;

100 end loop;

101 end loop;

102 for i in 1..H loop

103 for j in 1..N loop

104 if (a2<MT2(i)(j)) then

105 a2 := MT2(i)(j);

106 end if;

107 end loop;

108 end loop;

109 Suspend\_Until\_True (S1);

110 if (a<a2) then

111 a := a2;

112 end if;

113 Set\_True (S1);

114 Set\_True (S6);

115 new\_line;

116 put ("Task 2 finished");

117 end T2;

118 begin

119 null;

120 end Start;

121 begin

122 Set\_True (S1);

123 Set\_True (S2);

124 Set\_True (S3);

125 Start;

126 end Lab1;

127