Лабораторная работа №1

по курсу:

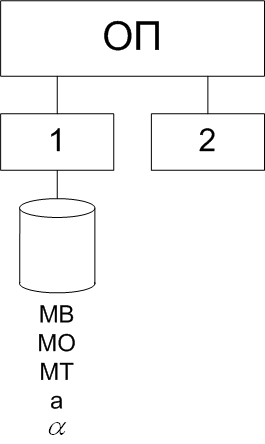
«Паралельные и распределённые вычисления»

Тема: «Ада. Семафоры»

Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

a = min(MB - α \* MO \* MT)

**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. ai = min(MBH – α \* MO \* MRH), i = , где Р – количество процессоров

**Общие ресурсы:** MO, α

1. a = min(a, ai)

**Общие ресурсы:** a

**Этап 2. Разработка алгоритмов процессов (задач)**

**№ Т1 ТС/КУ**

1. Ввод MB, MO, MT, α
2. Сигнал задаче Т2 о завершении ввода S2.1
3. Копирование α1 := α, МО1 := MO КУ
4. Счёт a1 = min(MBH – α1 \* MO1 \* MRH)
5. Счёт a = min(a, a1) KУ
6. Ждать завершения счёта в Т2 W2.1
7. Вывод а

**№ Т2 ТС/КУ**

1. Ждать введения MB, MO, MT, α в Т1 W1.1
2. Копирование α2 := α, МО2 := MO КУ
3. Счёт a2 = min(MBH – α2 \* MO2 \* MRH)
4. Счёт a = min(a, a2) KУ
5. Сигнал задаче Т2 о завершении счёта S1.1

**Этап 3. Разработка схемы взаимодействия задач**

Структурная схема взаимодействии процессов на рис. 3.1.



Рис. 3.1. Структурная схема взаимодействии процессов

S1 – синхронизация по вводу MB, MO, MT, α;

S2 – синхронизация по выводу а;

SKD1 – задача взаимного исключения по α и МО;

SKD2 – задача взаимного исключения по а;

**Этап 4. Разработка программы**

Source file: ..\lab1.adb Tue Feb 22 16:07:31 2011

1 ----------------------------------------------------------------

2 -- Paralel and distributed computing --

3 -- Laboratory work #1. Ada. Semaphores --

4 -- Func: a = MIN(MB-alfa\*MO\*MT) --

5 -- IO-83 --

6 ----------------------------------------------------------------

7

8 with Package1; use Package1;

9 with Ada.Text\_IO; use Ada.Text\_IO;

10 with Ada.Integer\_Text\_IO; use Ada.Integer\_Text\_IO;

11 with Ada.Synchronous\_Task\_Control; use Ada.Synchronous\_Task\_Control;

12

13 procedure Lab1 is

14

15

16 MB, MO, MT: Matr;

17 a: integer := 500000; --min - shared variable

18 Alfa: Integer;

19

20 --semaphores

21 S1, S2, SKD1, SKD2: Suspension\_Object;

22

23 procedure Task\_start is

24 task T1;

25 task body T1 is

26 alfa1: integer;

27 Sum: Integer := 0;

28 A1: Integer := 530000;

29 MO1: Matr;

30 begin

31 Put\_Line("T1 start");

32 --ENTERING MB, MO, MT, alfa (1)

33 Matr\_Input(MB);

34 Matr\_Input(MO);

35 Matr\_Input(MT);

36 alfa := 1;

37 --Signal to T2 about end of input (2)

38 Set\_True(S1); --S2-1

39

40 -- KY -- Copying of OP (3)

41 Suspend\_Until\_True(SKD1);

42 alfa1 := alfa;

43 MO1 := MO;

44 Set\_True(SKD1);

45 -- Calculation a1 = min(MBH - Alfa1 \* MO1 \* MRH (4)

46 for i in 1.. N loop

47 for j in 1 .. H loop

48 sum := 0;

49 for z in 1 .. N loop

50 sum := sum + MO1(i)(z) \* MT(Z)(J);

51 end loop;

52 Sum := MB(I)(J) - Alfa1 \* Sum;

53 if (A1 > Sum) then

54 A1 := Sum;

55 end if;

56 Sum := 0;

57 end loop;

58 end loop;

59 Sum := 0;

60 -- KY -- Calculation a = min(a, a1) (5)

61 Suspend\_until\_true(SKD2);

62 if (a > a1) then

63 a := a1;

64 end if;

65 Set\_True(SKD2);

66 --Waiting to end of calculation (6)

67 Suspend\_Until\_True(S2); --W2.1

68 --Out a. (7)

69 Constant\_Out(a);

70 Put\_Line("T1 stop");

71 end T1;

72

73 task T2;

74 task body T2 is

75 alfa2: integer;

76 A2: Integer := 540000;

77 Sum: Integer := 0;

78 MO2: Matr;

79 begin

80 Put\_Line("T2 start");

81

82 -- Waiting to the end of the input Data in T1 (1)

83 Suspend\_until\_true(S1); --W1-1

84

85 -- KY -- Copying of OP (2)

86 Suspend\_until\_true(SKD1);

87 alfa2 := alfa;

88 MO2 := MO;

89 Set\_True(SKD1);

90

91 -- Calculation a2 = min(MBH - Alfa2 \* MO2 \* MRH (3)

92 for i in 1.. N loop

93 for j in H+1 .. n loop

94 sum := 0;

95 for z in 1 .. N loop

96 sum := sum + MO2(i)(z) \* MT(Z)(J);

97 end loop;

98 Sum := MB(I)(J) - Alfa2 \* Sum;

99 if (A2 > Sum) then

100 A2 := Sum;

101 end if;

102 Sum := 0;

103 end loop;

104 end loop;

105 Sum := 0;

106

107 -- KY -- Calculation a = min(a, a2) (4)

108 Suspend\_until\_true(SKD2);

109 if (a > a2) then

110 a := a2;

111 end if;

112 Set\_True(SKD2);

113

114 --Waiting to end of calculation (6)

115 Set\_True(S2);

116

117 Put\_Line("T2 stop");

118 end T2;

119

120 begin

121 null;

122 end Task\_start;

123

124 begin

125 Set\_True(SKD1);

126 Set\_True(SKD2);

127 Task\_start;

128 end Lab1;