Лабораторная работа №1

по курсу:

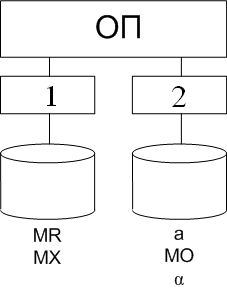
«Паралельные и распределённые вычисления»

Тема: «Ада. Семафоры»

Выполнил: студент группы ИВ-83

НТУУ «КПИ» ФИВТ

Воробйов Виталий

**Техническое задание**

a = min(MO \* MX + α \* MR)

**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. ai = min(MO \* MXH + α \* 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 MR, MX
2. Сигнал задаче Т2 о завершении ввода S2.1
3. Ждать введения α и MO в Т2 W2.1
4. Копирование α1 := α, МО1 := MO КУ
5. Счёт a1 = min(MO1 \* MXH + α1 \* MRH)
6. Счёт a = min(a, a1) KУ
7. Сигнал задаче Т2 о завершении счёта S2.3

**№ Т2 ТС/КУ**

1. Ввод MО, α
2. Ждать введения MR и MX в Т1 W1.1
3. Сигнал задаче Т1 о завершении ввода S1.1
4. Копирование α2 := α, МО2 := MO КУ
5. Счёт a2 = min(MO2 \* MXH + α2 \* MRH)
6. Счёт a = min(a, a2) KУ
7. Ждать завершения счёта в Т1 W1.3
8. Вывод а

**Этап 3. Разработка схемы взаимодействия задач**

S1 – синхронизация по вводу MR, MX;

S2 – синхронизация по вводу MО, α;

S3 – синхронизация по выводу а;

var1 – задача взаимного исключения по α и МО;

var2 – задача взаимного исключения по а;

Структурная схема взаимодействии процессов на рис. 3.1.

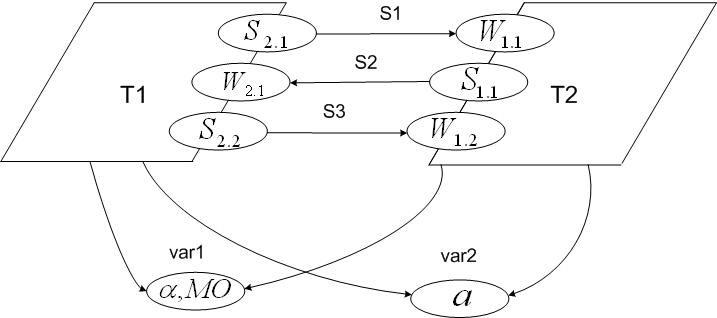


Рис. 3.1. Структурная схема взаимодействии процессов

**Этап 4. Разработка программы**

Source file: ..\lab1 Sat Feb 9 22:47:05 2011

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

2 -- Paralel and distributed computing --

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

4 -- Func: a = MIN(MO\*MX+alfa\*MR) --

5 -- IO-83 Vorobyev Vitaliy --

6 -- 9.02.2011 --

7 ----------------------------------------------------------------

8

9 with Package1; use Package1;

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

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

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

13

14 procedure Lab1 is

15

16

17 MO, MX, MR: Matr;

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

19 Alfa: Integer;

20

21 --СЕМАФОРЫ

22 S1, S2, S3, var1, var2: Suspension\_Object;

23

24 procedure Task\_start is

25 task T1;

26 task body T1 is

27 alfa1: integer;

28 Sum: Integer := 0;

29 A1: Integer := 530000;

30 MO1: Matr;

31 begin

32 Put\_Line("T1 start");

33 --1. ВВОД ДАННЫх

34 Matr\_Input(MR);

35 Matr\_Input(MX);

36 --2.3. Синхронизация задач

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

38 Suspend\_until\_true(S2); --W2-1

39

40 --4. Копирование Общих Ресурсов

41 Suspend\_Until\_True(Var1);

42 alfa1 := alfa;

43 MO1 := MO;

44 Set\_True(Var1);

45 --5. ВЫЧИСЛЕНИЕ ФУНКЦИИ

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) \* MX(Z)(J);

51 end loop;

52 Sum := Sum + Alfa1 \* MR(I)(J);

53 if (A1 > Sum) then

54 A1 := Sum;

55 end if;

56 Sum := 0;

57 end loop;

58 end loop;

59 Sum := 0;

60 --6. Доступ к общим ресурсам (КРИТИЧЕСКИЙ УЧАСТОК)

61 Suspend\_until\_true(Var2);

62 if (a > a1) then

63 a := a1;

64 end if;

65 Set\_True(Var2);

66 --7. Задача синхронизации (счёт окончен)

67 Set\_True(S3); --S2-2

68

69 Put\_Line("T1 stop");

70 end T1;

71 -----------------------------------------------------------

72 task T2;

73 task body T2 is

74 alfa2: integer;

75 A2: Integer := 540000;

76 Sum: Integer := 0;

77 MO2: Matr;

78 begin

79 Put\_Line("T2 start");

80 --1. ВВОД ДАННЫХ

81 Matr\_Input(MO);

82 alfa := 1;

83 --2.3. Синхронизация задач

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

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

86

87 --4. Копирование Общих Ресурсов

88 Suspend\_until\_true(Var1);

89 alfa2 := alfa;

90 MO2 := MO;

91 Set\_True(Var1);

92 --5. ВЫЧИСЛЕНИЕ ФУНКЦИИ

93 for i in 1.. N loop

94 for j in H+1 .. N loop

95 sum := 0;

96 for z in 1 .. N loop

97 sum := sum + MO2(i)(z) \* MX(Z)(J);

98 end loop;

99 Sum := Sum + Alfa2 \* MR(I)(J);

100 if (A2 > Sum) then

101 A2 := Sum;

102 end if;

103 Sum := 0;

104 end loop;

105 end loop;

106 Sum := 0;

107 --6. Доступ к общим ресурсам (КРИТИЧЕСКИЙ УЧАСТОК)

108 Suspend\_until\_true(Var2);

109 if (a > a2) then

110 A := A2;

111 end if;

112 Set\_True(Var2);

113 --7. Задача синхронизации (счёт окончен)

114 Suspend\_until\_true(S3); --W1-2

115 --8. ВЫВОД РЕЗУЛЬТАТА

116 Constant\_Out(A);

117

118 Put\_Line("T2 stop");

119 end T2;

120

121 begin

122 null;

123 end Task\_start;

124

125 begin

126 Set\_True(Var1);

127 Set\_True(Var2);

128 Task\_start;

129 end Lab1;