Лабораторная работа №1

по курсу:

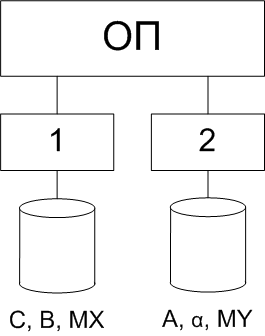
«Паралельные и распределённые вычисления»

Тема: «Ада. Семафоры»

Выполнил: студент группы ИВ-8

НТУУ «КПИ» ФИВТ

Аллахвердиев Тимур

**Техническое задание**

A = B \* α - C \* (MX \* MY)

**Этап 1. Построение параллельного алгоритма**

Вычисление данного матричного уравнения можно разбить на шаги:

1. AH = BH \* α – C \* (MX \* MYH)

**Общие ресурсы:** C, MX, α

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

**№ Т1 ТС/КУ**

1. Ввод C, B, MX
2. Сигнал задаче Т2 о завершении ввода S2.1
3. Ждать введения α и MY в Т2 W2.1
4. Копирование α1 := α, МX1 := MX, C1 := C КУ
5. Счёт AH = BH \* α1 – C1 \* (MX1 \* MYH)
6. Сигнал задаче Т2 о завершении счёта S2.2

**№ Т2 ТС/КУ**

1. Ввод MY, α
2. Ждать введения C, B, MX в Т1 W1.1
3. Сигнал задаче Т1 о завершении ввода S1.1
4. Копирование α2 := α, МX2 := MX, C2 := C КУ
5. Счёт AH = BH \* α2 – C2 \* (MX2 \* MYH)
6. Ждать завершения счёта в Т1 W1.2
7. Вывод A

**Этап 3. Разработка схемы взаимодействия задач**

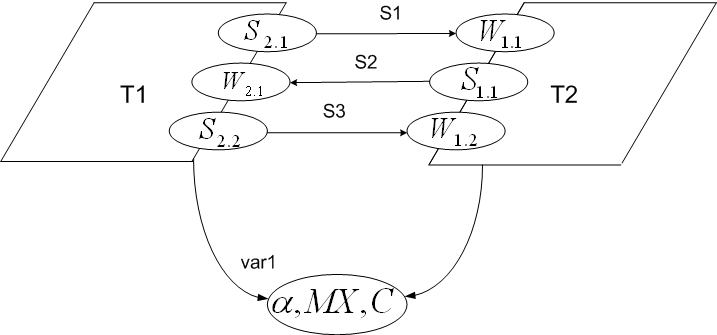


Рис. 3.1. Структурная схема взаимодействии процессов

**Этап 4. Разработка программы**

Source file: ..\lab1.adb Tue Mar 15 19:38:42 2011

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

2 -- Paralel and distributed computing --

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

4 -- Func: a = B\*Alfa - C(MX \* MY) --

5 -- IO-83 --

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 MX, MY: Matr;

18 A, B, C: Vec;

19 Alfa: Integer;

20

21 --semaphores

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

23

24 procedure Task\_start is

25 task T1;

26 task body T1 is

27 Sum: Integer := 0;

28 Alfa1: Integer := 0;

29 MX1, MBUF: Matr;

30 C1: Vec;

31 begin

32 Put\_Line("T1 start");

33 --1. Ввод C, B, MX

34 Matr\_Input(MX);

35 Vec\_Input(C);

36 Vec\_Input(B);

37 --2. Сигнал задаче Т2 о завершении ввода

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

39 --3. Ждать введения Alfa и MY в Т2

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

41

42 --4. Копирование Alfa1 := Alfa, МX1 := MX, C1 := C

43 --КРИТИЧЕСКАЯ СЕКЦИЯ

44 Suspend\_Until\_True(Var1);

45 alfa1 := alfa;

46 MX1 := MX;

47 C1 := C;

48 Set\_True(Var1);

49

50 --5. Счёт AH = BH \* Alfa1 - C1 \* (MX1 \* MYH)

51 for i in 1.. N loop

52 for j in 1 .. H loop

53 sum := 0;

54 for z in 1 .. N loop

55 sum := sum + MX1(i)(z) \* MY(Z)(J);

56 end loop;

57 MBUF(I)(J) := Sum; --(MX1 \* MYH)

58 Sum := 0;

59 end loop;

60 end loop;

61 --C1 \* (MX1 \* MYH)

62 for J in 1.. H loop

63 sum := 0;

64 for I in 1 .. N loop

65 sum := sum + C1(i) \* MBUF(I)(J);

66 end loop;

67 A(J) := B(J) \* Alfa1 - Sum;

68 end loop;

69 --6. Сигнал задаче Т2 о завершении счёта

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

71

72 Put\_Line("T1 stop");

73 end T1;

74

75 task T2;

76 task body T2 is

77 Sum: Integer := 0;

78 Alfa2: Integer := 0;

79 MX2, MBUF: Matr;

80 C2: Vec;

81 begin

82 Put\_Line("T2 start");

83 --1. Ввод MY, Alfa

84 Matr\_Input(MY);

85 alfa := 1;

86 --3. Сигнал задаче Т1 о завершении ввода

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

88 --2. Ждать введения C, B, MX в Т1

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

90

91 --4. Копирование Alfa2 := Alfa, МX2 := MX, C2 := C

92 --КРИТИЧЕСКАЯ СЕКЦИЯ

93 Suspend\_Until\_True(Var1);

94 alfa2 := alfa;

95 MX2 := MX;

96 C2 := C;

97 Set\_True(Var1);

98

99 --5. Счёт AH = BH \* Alfa2 - C2 \* (MX2 \* MYH)

100 for i in 1.. N loop

101 for j in H+1 .. N loop

102 sum := 0;

103 for z in 1 .. N loop

104 sum := sum + MX2(i)(z) \* MY(Z)(J);

105 end loop;

106 MBUF(I)(J) := Sum; --(MX2 \* MYH)

107 Sum := 0;

108 end loop;

109 end loop;

110 --C2 \* (MX2 \* MYH)

111 for J in H+1 .. N loop

112 sum := 0;

113 for I in 1 .. N loop

114 sum := sum + C2(i) \* MBUF(I)(J);

115 end loop;

116 A(J) := B(J) \* Alfa2 - Sum;

117 end loop;

118

119 --6. Ждать завершения счёта в Т1

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

121

122 --7. Вывод A

123 New\_Line;

124 Vec\_print(A);

125 New\_Line;

126 Put\_Line("T2 stop");

127 end T2;

128

129 begin

130 null;

131 end Task\_start;

132

133 begin

134 Set\_True(Var1);

135 Task\_start;

136 end Lab1;