НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА АВТОМАТИЗАЦІЇ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ТА УПРАВЛІННЯ

Лабораторна робота №1

з дісципліни **«**Паралельне програмування**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІП-31

Кахерський О.І.

Перевірив:

Корочкін О. В.

Київ – 2016 р.

**Тема:** Програмування для комп’ютерних систем зі спільною пам’яттю. Ада. Семафори

**Технічне завдання:** Розробити програму для розв’язання ПКС із СП (структура на рис. 1) математичної задачі: A = B\*(MO\*MZ)+\*C

**Мова програмування:** Ада.

**Засоби взаємодії:** семафори.

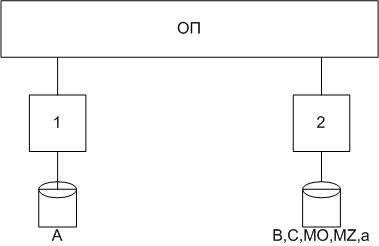


Рис.1 Структурна схема ПКС

**Виконання роботи**

**Етап 1.** Побудова паралельного алгоритму

Спільний ресурс: B, MZ,

**Етап 2. Розробка алогоритмів роботи кожного процесу**

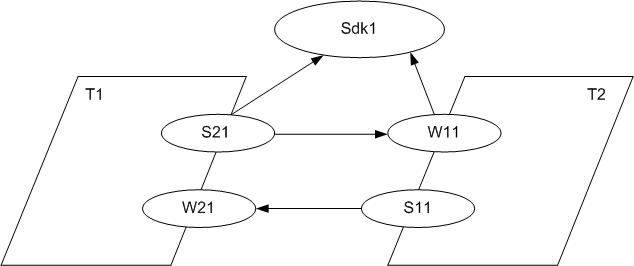
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Задача Т1 |  | Задача Т2 |  |
| 1 | Введення В,МО,MZ,, C |  | Чекати введення В,МО,MZ,, C | W11 |
| 2 | Надіслати сигнал про введення В,МО,MZ,, C задачі Т2 | S21 | Копіювати В, MZ, |  |
| 3 | Обчислення |  | Обчислення |  |
| 4 | Чекати сигнал від Т2 про завершення обчислень | W21 | Сигнал Т1 про завершення обчислень | S11 |
| 5 | Виведення результату А |  |  |  |

**Етап 3. Розробка структурної схеми взаємодії задач**

Skd1 – для керування доступом до спільних ресурсів B, MZ,

S1 – для синхронізації із завершенням введення в T1

S2 – для синхронізації із завершенням обчислень в Т2



**Етап 4. Лістинг коду**

GNAT 4.6.4

Copyright 1992-2010, Free Software Foundation, Inc.

Compiling: /home/oleg/Data/Документы/FICT/6\_семестр/ПП/lab1/lab1.adb (source file time stamp: 2016-03-03 00:28:07)

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

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

3.

4.

5. ------------------------------------------------------------------

6. -- --

7. -- Паралельні і розподілені обчислення --

8. -- Лабораторна робота №1. --

9. -- --

10. -- Файл: lab1.ada --

11. -- Завдання: --

12. -- А = B\*(MO\*MZ)+a\*C --

13. -- --

14. -- --

15. -- --

16. -- Автор: Кахерський Олег, група IП-31 --

17. -- Дата: 02.03.2015 --

18. -- --

19. ------------------------------------------------------------------

20. procedure Lab1 is

21. package DataN is new Data(3000);

22. use DataN;

23. MO,MZ: Matrix;

24. B,C,a:Vector;

25. alpha: Integer;

26. h,n:integer;

27. Skd1,S1,S2 : Suspension\_Object;

28.

29. procedure Start is

30. task T1;

31. task T2;

32.

33. task body T1 is

34. MO1, Buff: Matrix;

35. B1, cHBuff: Vector;

36. a1: integer;

37. begin

38. Put\_Line("T1 started\n");

39.

40. --init data

41. Input\_Matrix(MO);

42. Input\_Matrix(MZ);

43. Input\_Vector(B);

44. Input\_Vector(C);

45. Input\_Constant(alpha);

46. Put\_Line("T1 sent signal S1\n");

47. Set\_True(S1);

48.

49. Put\_Line("T1 waits for Skd1\n");

50. Suspend\_Until\_True(Skd1);

51. Put\_Line("T1 in crit section Skd1\n");

52. B1 := B;

53. a1 := alpha;

54. MO1 := MO;

55. Set\_True(Skd1);

56. Put\_Line("T1 cancelled Skd1 section\n");

57.

58. Matrix\_Matrix\_Multiply(MO1,MZ,1,h,Buff); --Buff = MOh\*MZ.

59. --Output\_Matrix(Buff);

60. Vector\_Matrix\_Multiply(B1,Buff,1,h,A); -- A = Bh\*Buff\_h

61. --Output\_Vector(A);

62. Constant\_Vector\_Multiply(a1,C,1,h,cHBuff); --a\*Ch

63. --Output\_Vector(CHBuff);

64. Vector\_Vector\_Add(CHBuff,A,1,h,A);

65.

66. Put\_Line("T1 waiting for signal S2 from T2\n");

67. Suspend\_Until\_True(S2);

68. if h\*2 <= 8 then

69. Output\_Vector(A);

70. end if;

71. Put\_Line("T1 finished\n");

72. end T1;

73.

74. task body T2 is

75. MO2, Buff: Matrix;

76. B2,cHBuff : Vector;

77. a2 : Integer;

78. begin

79. --Put\_Line("T2 started\n");

80. --Put\_Line("T2 waiting for signal S1 from T1");

81. Suspend\_Until\_True(S1);

82. --Put\_Line("T2 waits for Skd1\n");

83. Suspend\_Until\_True(Skd1);

84. --Put\_Line("T2 in crit section Skd1\n");

85. B2 := B;

86. a2 := alpha;

87. MO2:= MO;

88. Set\_True(Skd1);

89. Put\_Line("T2 cancelled Skd1 section\n");

90.

91. Matrix\_Matrix\_Multiply(MO2,MZ,H+1,n,Buff); --Buff = MOh\*MZ.

92. --Output\_Matrix(Buff);

93. Vector\_Matrix\_Multiply(B2,Buff,H+1,n,A); -- A = Bh\*Buff\_h

94. --Output\_Vector(A);

95. Constant\_Vector\_Multiply(a2,C,H+1,n,cHBuff); --a\*Ch

96. --Output\_Vector(CHBuff);

97. Vector\_Vector\_Add(cHBuff,A,H+1,n,A);

98. --Output\_Vector(A);

99.

100. --Put\_Line("T2 sent signal S2\n");

101. Set\_True(S2);

102. --Put\_Line("T2 finished\n");

103. end T2;

104. begin

105. null;

106. end Start;

107. begin

108. n:=3000;

109. h := n/2;

110. Set\_True(Skd1);

111. Put(H);

112. Start;

113. end Lab1;

113 lines: No errors