Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №1

Ада. Семафори

Виконав:

студент групи ІП-22

Сочка О. О.

Перевірив:

Корочкін О. В.

Київ - 2015

**Лабораторна робота №1. АДА. Семафори**

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** Ада

**Засоби організації взаємодії процесів:** семафори мови Ада з пакета

Ada.Synchronous\_Task\_Control.

**Завдання:**

**A = (B\*MC) \* (MO \* MK) + a \* E**

**Етап 1. Розробка паралельного математичного алгоритму**

1.Qh = B\*MCh

2. Ah = Q \* (MO \* MKh) + a \* Eh

OP: B, Q, MO

**Етап 2. Розробка алгоритмів задач**

|  |  |  |  |
| --- | --- | --- | --- |
| Т1 | ТС, КУ | Т2 | ТС, КУ |
| Ввід MK, B, E |  | Ввід a, MO, MC |  |
| Сигнал Т2 про ввід даних. | S₂₁ | Сигнал Т1 про ввід даних | S11 |
| Очікування сигналу від Т2 про ввід даних | W₂₁ | Очікування сигналу від Т2 про ввід даних | W12 |
| Копія B1 = B | КУ | Копія B2 = B | КУ |
| Підрахунок Qh = B1\*MCh |  | Підрахунок Qh = B2\*MCh |  |
| Сигнал Т2 про підрахунок Qh | S22 | Сигнал Т1 про підрахунок Qh | S12 |
| Очікування сигналу від Т2 про підрахунок Qh | W22 | Очікування сигналу від Т1 про підрахунок Qh | W12 |
| Копія Q1 = Q, a1 = a, MO1= MO | КУ | Копія Q2 = Q, a2 = a, MO2=MO | КУ |
| Підрахунок Ah = Q1 \* (MO1 \* MKh) + a1 \* Eh |  | Підрахунок Ah = Q2 \* (MO2 \* MKh) + a2 \* Eh |  |
| Очікування сигналу від T2 про підрахунок Ah | W23 | Надсилання сигналу Т1 про підрахунок Ah | S13 |
| Вивід A |  |  |  |

**Етап 3. Розробка схеми взаємодії задач**

**Етап 4. Розробка програми**

**Лістинг програми**

File: lab1.adb

1

2 ----------------------------------------------------------------

3 -- Paralel and distributed computing --

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

5 -- Func: A = (B\*MC)\*(MO\*MK) + a\*E --

6 -- IP-22 --

7 -- Sochka Oleksandr --

8 -- 12.02.2015 --

9 ----------------------------------------------------------------

10

11 with Package1; use Package1;

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

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

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

15

16 -- pragma Storage\_Size (3\_000\_000);

17

18 procedure Lab1 is

19 MC, MO, MK: Matr;

20 Q: Vec := (others => 0);

21 A, E, B: Vec;

22 alpha: Integer;

23

24 --semaphores

25 S21, S22, S11, S12, S13, SharedAccess: Suspension\_Object;

26

27 procedure Task\_start is

28 task T1;

29 task body T1 is

30 B1: Vec;

31 Q1: Vec;

32 alpha1: Integer;

33 MO1: Matr;

34 begin

35 Put\_Line("T1 start");

36 --1. Ввод MK, B, E

37 Matr\_Input(MK);

38 Vec\_Input(B);

39 Vec\_Input(E);

40 --2. Сигнал задаче Т2 о завершении ввода

41 Set\_True(S21); --S2-1

42 --3. Ждать введения

43 Suspend\_Until\_True(S11);

44 --4. Копія B1 = B

45 --КРИТИЧЕСКАЯ СЕКЦИЯ

46 Suspend\_Until\_True(SharedAccess);

47 B1 := B;

48 Set\_True(SharedAccess);

49 --5.Підрахунок Qh = B1\*MCh

50 for i in 1 .. H loop

51 for j in 1 .. N loop

52 Q(i) := Q(i) + B1(i) \* MC(j)(i);

53 end loop;

54 end loop;

55 --6.Сигнал Т2 про підрахунок Qh

56 Set\_True(S22);

57 --7.Очікування сигналу від Т2 про підрахунок Qh

58 Suspend\_Until\_True(S12);

59 --8.Копія Q1 = Q, a1 = a, MO1 = MO

60 Suspend\_Until\_True(SharedAccess);

61 Q1 := Q;

62 alpha1 := alpha;

63 MO1 := MO;

64 Set\_true(SharedAccess);

65 --9 Підрахунок Ah = Q1 \* (MO1 \* MKh) + a1 \* Eh

66 for k in 1 .. H loop

67 A(k) := alpha1 \* E(k);

68 for i in 1 .. N loop

69 for j in 1 .. N loop

70 A(k) := A(k) + Q1(k) \* (MO1(i)(j) \* MK(j)(k));

71 end loop;

72 end loop;

73 end loop;

74 --10. Очікування сигналу від T2 про підрахунок Ah

75 Suspend\_Until\_True(S13);

76 --11. Вивід Ah

77 Vec\_Print(A);

78 New\_Line;

79 Put\_Line("T1 stop");

80 end T1;

81

82 task T2;

83 task body T2 is

84 B2: Vec;

85 Q2: Vec;

86 alpha2: Integer;

87 MO2: Matr;

88 begin

89 Put\_Line("T2 start");

90 --1. Ввід a, MO, MC

91 alpha := 1;

92 Matr\_Input(MO);

93 Matr\_Input(MC);

94 --2. Сигнал задаче Т2 о завершении ввода

95 Set\_True(S11); --S2-1

96 --3. Ждать введения

97 Suspend\_Until\_True(S21);

98 --4. Копія B2 = B

99 --КРИТИЧЕСКАЯ СЕКЦИЯ

100 Suspend\_Until\_True(SharedAccess);

101 B2 := B;

102 Set\_True(SharedAccess);

103 --5.Підрахунок Qh = B2\*MCh

104 for i in H + 1 .. N loop

105 for j in 1 .. N loop

106 Q(i) := Q(i) + B2(i) \* MC(j)(i);

107 end loop;

108 end loop;

109 --6.Сигнал Т2 про підрахунок Qh

110 Set\_True(S12);

111 --7.Очікування сигналу від Т2 про підрахунок Qh

112 Suspend\_Until\_True(S22);

113 --8.Копія Q2 = Q, a2 = a

114 Suspend\_Until\_True(SharedAccess);

115 Q2 := Q;

116 alpha2 := alpha;

117 MO2 := MO;

118 Set\_true(SharedAccess);

119 --9 Підрахунок Ah = Q2 \* (MO1 \* MKh) + a2 \* Eh

120 for k in H + 1 .. N loop

121 A(k) := alpha2 \* E(k);

122 for i in 1 .. N loop

123 for j in 1 .. N loop

124 A(k) := A(k) + Q2(k) \* (MO2(i)(j) \* MK(j)(k));

125 end loop;

126 end loop;

127 end loop;

128 --10. Очікування сигналу від T2 про підрахунок Ah

129 Set\_True(S13);

130 Put\_Line("T2 stop");

131 end T2;

132

133 begin

134 null;

135 end Task\_start;

136 begin

137 Set\_True(SharedAccess);

138 Task\_start;

139 end Lab1;

File: package1.adb

1 package body Package1 is

2 procedure Matr\_Input(M: in out Matr) is

3 begin

4 for i in 1 .. N loop

5 for j in 1 .. N loop

6 M(i)(j) := 1;

7 end loop;

8 end loop;

9 end Matr\_Input;

10

11 procedure Vec\_Input(V: in out Vec) is

12 begin

13 for i in 1 .. N loop

14 V(i) := 1;

15 end loop;

16 end Vec\_Input;

17

18 procedure Constant\_Out(a: integer) is

19 begin

20 Put(a, 4);

21 New\_Line;

22 end Constant\_Out;

23

24 procedure Vec\_print(v: Vec) is

25 begin

26 for i in 1 .. N loop

27 Put(v(i), 4);

28 end loop;

29 end;

30

31 procedure Matr\_print(m: Matr) is

32 begin

33 for i in 1..n loop

34 Vec\_print(m(i));

35 New\_Line;

36 end loop;

37 end Matr\_print;

38

39 end Package1;

40

File: package1.ads

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

2 with Ada.INteger\_Text\_IO; use Ada.INteger\_Text\_IO;

3 with Ada.IO\_Exceptions; use Ada.IO\_Exceptions;

4

5 package Package1 is

6 N: integer := 2; --size of structures

7 H: integer := N/2; --size of halve

8

9 type Vec is array (1..N) of Integer;

10 type Matr is array (1..N) of Vec;

11

12 procedure Constant\_Out(a: integer);

13 procedure Vec\_print(v: Vec);

14 procedure Matr\_print(m: Matr);

15 procedure Vec\_Input(V: in out Vec);

16 procedure Matr\_Input(M: in out Matr);

17

18 end Package1;