Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №1**

**з дисципліни «Програмування паралельних комп’ютерних систем»**

**на тему «Ада. Семафори»**

Виконав:

студент ІІI курсу

групи ІО-43

Крут Владислав

Перевірив:

Доц. Корочкін О. В.

Київ – 2017

**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\*e + C\*(MO\*MX).

Мова програмування: Ада.

Засоби взаємодії: семафори.

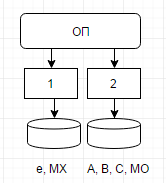


Рис. 1 Структурна схема ПКС

**Виконання роботи:**

**Етап 1. Побудова паралельного алгоритму**

AH = BH \* e + C \* (MO\* MXH)

Спільний ресурс: e, C, MO

**Етап 2. Розроблення алгоритмів роботи кожного процесу**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Задача Т1 |  | Задача Т2 |  |
| 1 | Ввод е, МХ |  | Ввод В, С, МО |  |
| 2 | Сообщить Т2 об вводе | S2,1 | Сообщить Т1 об вводе | S1,1 |
| 3 | Ждать Т2 | W2,1 | Ждать Т1 | W1,1 |
| 4 | Копирование е1=е, С1=С, МО1=МО | КУ | Копирование е2=е, С2=С, МО2=МО | КУ |
| 5 | Счёт AH=BH\*e+C\*(MO\*MXH) |  | Счёт AH=BH\*e+C\*(MO\*MXH) |  |
| 6 | Сообщить Т2 об завершении | S2,2 | Ждать Т1 | W1,2 |
| 7 |  |  | Вывод А |  |

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

На структурній схемі взаємодії задач уведено такі семафори

* Skd1 – для керування доступом до спільних ресурсів
* S1 – для синхронізації із завершенням введення в T1
* S2 – для синхронізації із завершенням введення в T2
* S3 – для синхронізації із завершенням обчислень в T1

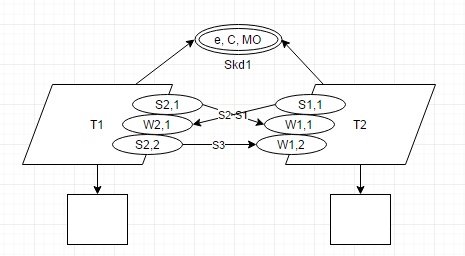


Рис. 2 Структурна схема взаємодії задач

**Етап 4. Розробка програми**

GNAT GPL 2014 (20140331)

Copyright 1992-2014, Free Software Foundation, Inc.

Compiling: lab1.adb (source file time stamp: 2017-02-24 00:19:12)

1. --------------------------------------------------

2. -- --

3. -- Parallel and Distributed Computing --

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

5. -- --

6. -- Task: A = B\*e + C\*(MO\*MX) --

7. -- --

8. -- Author: Krut Vladyslav, IO-43 --

9. -- Date: 23.02.2017 --

10. -- --

11. -------------------------------------------------

12.

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

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

15.

16. procedure Lab1 is

17. N: Integer := 500;

18. P: Integer := 2;

19. H: Integer := N/P;

20.

21. subtype Index is Positive range 1..N;

22. type Vector is array (Index) of Integer;

23. type Matrix is array (Index) of Vector;

24.

25. MO, MX: Matrix;

26. A, B, C: Vector;

27. e: Integer;

28. S1, S2, S3, Skd1: Suspension\_Object;

29.

30. procedure Input\_Matrix(MA: out Matrix);

31. procedure Input\_Vector(V: out Vector);

32. procedure Input\_Integer(a: out Integer);

33. procedure Vector\_Output(V: in Vector);

34.

35. procedure Start\_tasks is

36.

37. task T1 is

38. pragma Priority(3);

39. pragma Task\_Name ("T1");

40. pragma Storage\_Size(1500000000);

41. end;

42. task body T1 is

43. MO\_MX: Matrix;

44. MO1: Matrix;

45. C1: Vector;

46. e1: Integer;

47. s: Integer;

48. begin

49. Put\_Line("T1 started");

50. -- 1. Input

51. Input\_Matrix(MX);

52. Input\_Integer(e);

53. -- 2. Notify T2 about input

54. Set\_True(S1);

55. -- 3. Wait T2

56. Suspend\_Until\_True(S2);

57. -- 4. KY Copying e, C, MO

58. Suspend\_Until\_True(Skd1);

59. MO1 := MO;

60. C1 := C;

61. e1 := e;

62. Set\_True(Skd1);

63.

64.

65. -- 5. Computing

66. for i in 1..H loop

67. for j in Index loop

68. s := 0;

69. for k in Index loop

70. s := s + MX(i)(k) \* MO1(k)(j);

71. MO\_MX(i)(j) := s;

72. end loop;

73. end loop;

74. end loop;

75.

76. for i in 1..H loop

77. s := 0;

78. for j in Index loop

79. s := s + MO\_MX(i)(j) \* C1(j);

80. end loop;

81. A(i) := s + B(i)\*e1;

82. end loop;

83.

84. -- 6. Wait for computing

85. Set\_True(S3);

86.

87. Put\_Line("T1 finished");

88. end T1;

89.

90. task T2 is

91. pragma Priority(3);

92. pragma Task\_Name ("T2");

93. pragma Storage\_Size(1500000000);

94. end;

95.

96. task body T2 is

97. MO\_MX: Matrix;

98. MO2: Matrix;

99. C2: Vector;

100. e2: Integer;

101. s: Integer;

102. begin

103. Put\_Line("T2 started");

104. -- 1. Input

105. Input\_Matrix(MO);

106. Input\_Vector(B);

107. Input\_Vector(C);

108. -- 2. Notify T1 about input

109. Set\_True(S2);

110. -- 3. Wait T1

111. Suspend\_Until\_True(S1);

112. -- 4. KY Copying e, C, MO

113. Suspend\_Until\_True(Skd1);

114. MO2 := MO;

115. C2 := C;

116. e2 := e;

117. Set\_True(Skd1);

118.

119. -- 5. Compute

120. for i in H + 1..N loop

121. for j in Index loop

122. s := 0;

123. for k in Index loop

124. s := s + MX(i)(k) \* MO2(k)(j);

125. MO\_MX(i)(j) := s;

126. end loop;

127. end loop;

128. end loop;

129.

130. for i in H + 1..N loop

131. s := 0;

132. for j in Index loop

133. s := s + MO\_MX(i)(j) \* C2(j);

134. end loop;

135.

136. A(i) := s + B(i) \* e2;

137. end loop;

138. -- 6. Wait T1

139. Suspend\_Until\_True(S3);

140.

141. -- 7. Output A

142. Vector\_Output(A);

143.

144. Put\_Line("T2 finished");

145. end T2;

146.

147. begin

148. null;

149. end;

150.

151. procedure Input\_Vector(V: out Vector) is

152.

153. begin

154. for J in Index loop

155. V(J) := 1;

156. end loop;

157. end Input\_Vector;

158.

159. procedure Input\_Matrix(MA: out Matrix) is

160. begin

161. for I in Index loop

162. for J in Index loop

163. MA(I)(J) := 1;

164. end loop;

165. end loop;

166. end Input\_Matrix;

167.

168. procedure Input\_Integer(a: out Integer) is

169. begin

170. a := 1;

171. end Input\_Integer;

172.

173. procedure Vector\_Output(V: in Vector) is

174. begin

175. if N < 7 then

176. Put\_Line("");

177. for i in Index loop

178. Put(V(i));

179. end loop;

180. Put\_Line("");

181. end if;

182. end Vector\_Output;

183.

184. begin

185.

186. Set\_True(Skd1);

187. Start\_tasks;

188.

189. end Lab1;

190.

190 lines: No errors