



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА №5
З ДИСЦИПЛІНИ “ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ”
НА ТЕМУ: “Ада. Рандеву”

Виконав:

Студент III курсу ФІОТ
групи ІО-82
Шендріков Євгеній
Номер у списку - 24

Перевірив:

Доцент
Корочкін О. В.

Технічне завдання

1. Розробити паралельний алгоритм рішення математичної задачі
 $Z = (B * C) * D + E * (MA * MB) * x$ з використанням механізму рандеву на мові Ада;
2. Описати алгоритм кожного процесу;
3. Розробити структурну схему взаємодії задач, де застосувати всі вказані засоби взаємодії процесів;
4. Розробити програму (обов'язкові “шапка”, коментарі), виконати її налагодження та отримати правильні результати обчислень.

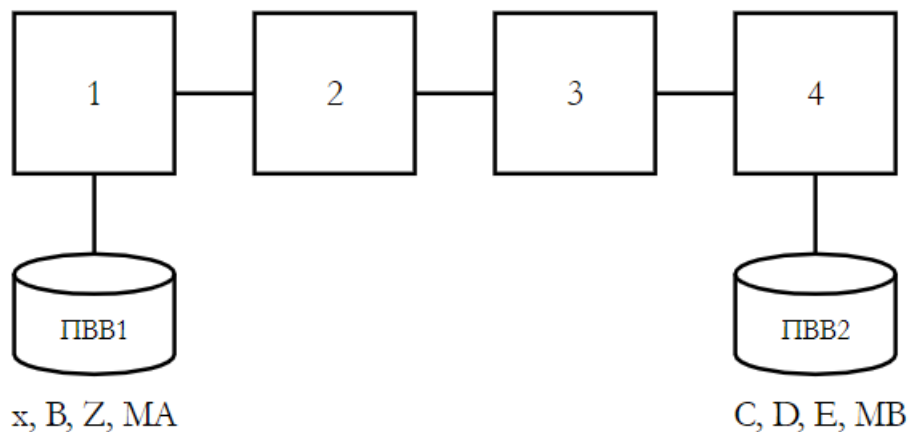
Задача: $Z = (B * C) * D + E * (MA * MB) * x$;

Мова програмування: Ада;

Засоби організації взаємодії: механізм рандеву;

Структура ПКС ЛП: лінійна.

Структурна схема ПКС ЛП



Етап 1. Побудова паралельного алгоритму

- 1) $b_i = B_H * C_H, i = \overline{1, P}$
- 2) $b = b + b_i, i = \overline{1, P}$
- 3) $Z_H = b * D_H + E * (MA * MB_H) * x$

Етап 2. Розроблення алгоритмів роботи кожного процесу

Задача T1	
1	Ввести x, V, MA
2	Передати x, V_{3H}, MA задачі T2
3	Отримати C_H, D_H, E, MB_H від задачі T2
4	Обчислити $b_1 = V_H * C_H$
5	Передати b_1 задачі T2
6	Отримати b від задачі T2
7	Обчислити $Z_H = b * D_H + E * (MA * MB_H) * x$
8	Отримати Z_{3H} від задачі T2
9	Вивести Z

Задача T2	
1	Отримати x, V_{3H}, MA від задачі T1
2	Передати x, V_{2H}, MA задачі T3
3	Отримати $C_{2H}, D_{2H}, E, MB_{2H}$ від задачі T3
4	Передати C_H, D_H, E, MB_H задачі T1
5	Обчислити $b_2 = V_H * C_H$
6	Отримати b_1 від задачі T1
7	Обчислити $b_2 = b_2 + b_1$
8	Отримати b_3 від задачі T3
9	Обчислити $b = b_2 + b_3$
10	Передати b задачам T1, T3
11	Обчислити $Z_H = b * D_H + E * (MA * MB_H) * x$
12	Отримати Z_{2H} від задачі T3
13	Передати Z_{3H} задачі T1

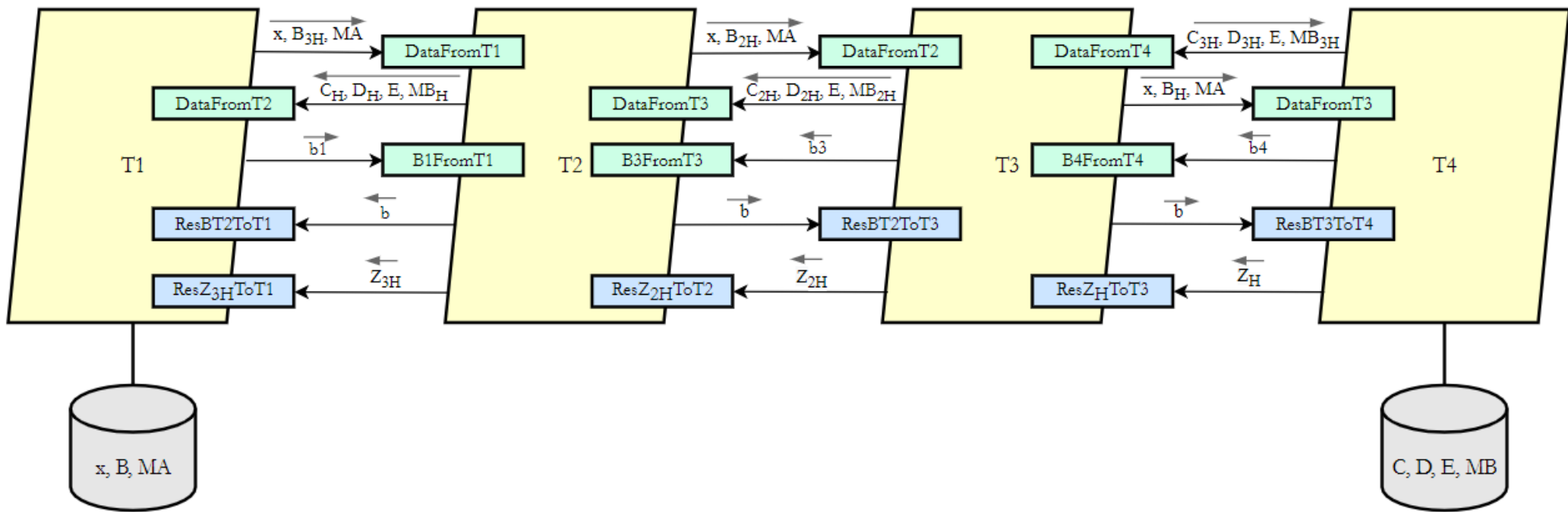
Задача T3	
1	Отримати $C_{3H}, D_{3H}, E, MB_{3H}$ від задачі T4
2	Отримати x, V_{2H}, MA від задачі T2
3	Передати $C_{2H}, D_{2H}, E, MB_{2H}$ задачі T2
4	Передати x, V_H, MA задачі T4
5	Обчислити $b_3 = V_H * C_H$
6	Отримати b_4 від задачі T4
7	Обчислити $b_3 = b_3 + b_4$
8	Передати b_3 задачі T2
9	Отримати b від задачі T2
10	Передати b задачі T4
11	Обчислити $Z_H = b * D_H + E * (MA * MB_H) * x$
12	Отримати Z_H від задачі T4
13	Передати Z_{2H} задачі T2

Задача Т4	
1	Ввести С, D, E, MB
2	Передати C_{3H} , D_{3H} , E, MB_{3H} задачі Т3
3	Отримати x, B_H , MA від задачі Т3
4	Обчислити $b_4 = B_H * C_H$
5	Передати b_4 задачі Т3
6	Отримати b від задачі Т3
7	Обчислити $Z_H = b * D_H + E * (MA * MB_H) * x$
8	Передати Z_H задачі Т3

Етап 3. Розроблення структурної схеми взаємодії задач

На структурній схемі взаємодії задач уведено такі умовні позначення:

- *DataFromT1* – для передачі даних, які вводяться у задачі Т1;
- *DataFromT4* – для передачі даних, які вводяться у задачі Т4;
- *ResBT2ToT1*, *ResBT2ToT3*, *ResBT3ToT4* – для передачі результату векторного добутку до всіх задач;
- *ResZ_{3H}ToT1*, *ResZ_{2H}ToT2*, *ResZ_HToT3* – для передачі частин результуючого вектора Z після закінчення обчислень;
- *B1FromT1*, *B3FromT3*, *B4FromT4* – для передачі частин векторного добутку для його подальшого розрахунку у Т2.



Етап 4. Розробка програми

Результат роботи

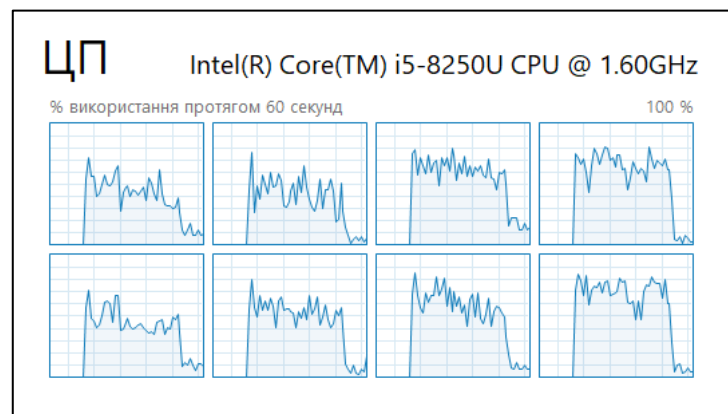
```
C:\Users\johnb\Desktop\lab5\obj\main.exe
Lab5 started!

T2 started.
T1 started.
T3 started.
T4 started.

T4 finished.
T3 finished.
T2 finished.
Vector Z: 20 20 20 20
T1 finished.

Lab5 finished!

[2021-04-06 20:16:55] process terminated successfully, elapsed time: 01.37s
```



Висновки

1. На основі механізму рандеву на мові Ада було розроблено програму та паралельний алгоритм для рішення математичної задачі заданої за варіантом.
2. Було описано алгоритм кожного потоку (T1 – T4);
3. Розроблено структурну схему взаємодії задач, де було застосовано вказані в завданні засоби взаємодії процесів. Засобом організації взаємодії слугував механізм рандеву.
4. Було написано програму згідно з завданням та перевірено її працездатність, а також проконтрольовано завантаження ядер процесору за допомогою Диспетчера задач. Програма забезпечує 80% завантаженості.

ЛІСТИНГ КОДУ

Lab5.adb

```
1. -----
2. --|                               Labwork #5                               |
3. --|                               Ada. Rendezvous                           |
4. -----
5. --|   Author   |   Jack (Yevhenii) Shendrikov   |
6. --|   Group    |   IO-82                        |
7. --|   Variant  |   #24                          |
8. --|   Date     |   06.04.2020                    |
9. -----
10. --| Function   |   Z = (B*C)*D + E*(MA*MB)*x           |
11. -----
12.
13. with Ada.Text_IO, Ada.Integer_Text_IO;
14. use Ada.Text_IO, Ada.Integer_Text_IO;
15.
16. procedure Main is
17.     N: Integer := 4;
18.     P: Integer := 4;
19.     H: Integer := N / P;
20.     FILL_NUM: Integer := 1;
21.
22.     type Vector_All is array(Integer range <>) of Integer;
23.     subtype Vector is Vector_All(1..N);
24.     subtype VectorH is Vector_All(1..1*H);
25.     subtype Vector2H is Vector_All(1..2*H);
26.     subtype Vector3H is Vector_All(1..3*H);
27.
28.
29.     type Matrix_All is array(Integer range <>) of Vector;
30.     subtype Matrix is Matrix_All(1..N);
31.     subtype MatrixH is Matrix_All(1..1*H);
32.     subtype Matrix2H is Matrix_All(1..2*H);
33.     subtype Matrix3H is Matrix_All(1..3*H);
34.
35.
36.     -- Input Procedures for Number, Vector, Matrix --
37.     procedure NumberInput(x: out Integer) is
38.     begin
39.         x := FILL_NUM;
40.     end NumberInput;
41.
42.     procedure VectorInput(V: out Vector) is
43.     begin
44.         for i in 1..N loop
45.             V(i) := FILL_NUM;
46.         end loop;
47.     end VectorInput;
48.
49.     procedure MatrixInput(M: out Matrix) is
50.     begin
51.         for i in 1..N loop
```

```

52.         for j in 1..N loop
53.             M(i)(j) := FILL_NUM;
54.         end loop;
55.     end loop;
56. end MatrixInput;
57.
58.
59. -- Output Procedures for Vector --
60. procedure VectorOutput(V: in Vector; str: in String) is
61. begin
62.     Put("Vector " & str & ":");
63.     for i in 1..N loop
64.         put(V(i));
65.         put(" ");
66.     end loop;
67.     Put_Line("");
68. end VectorOutput;
69.
70.
71. -- Calculation Procedure [Zh = b * Dh + E * (MA * MBh) * x] --
72. procedure Calculate(b : Integer; DH : VectorH; E : Vector; MA : Matrix;
73. MBH : MatrixH; x : Integer; ZH : out VectorH) is
74.     sum1, sum2 : Integer;
75. begin
76.     for i in 1..H loop
77.         sum1 := 0;
78.         for j in 1..N loop
79.             sum2 := 0;
80.             for k in 1..N loop
81.                 -- Calculate MA * MBh --
82.                 sum2 := sum2 + MA(k)(j) * MBH(i)(k);
83.             end loop;
84.             -- Calculate E * (MA * MBh) * x --
85.             sum1 := sum1 + E(j) * sum2 * x;
86.         end loop;
87.         -- Calculate Zh --
88.         ZH(i) := b * DH(i) + sum1;
89.     end loop;
90. end Calculate;
91.
92.
93. procedure tasksProcedure is
94.     task T1 is
95.         entry DataFromT4(VCH : in VectorH; VDH : in VectorH; V : in
96. Vector; M : in MatrixH);
97.         entry ResBT2ToT1(b : Integer);
98.         entry ResZ3HToT1(ResZ3H : Vector3H);
99.     end T1;
100.
101.     task T2 is
102.         entry DataFromT1(a : in Integer; V3H : in Vector3H; M : in
103. Matrix);

```



```

104.          entry DataFromT4(VC2H : in Vector2H; VD2H : in Vector2H; V : in
Vector; M : in Matrix2H);
105.          entry B1FromT1(b1 : Integer);
106.          entry B3FromT3(b3 : Integer);
107.          entry ResZ2HToT2(ResZ2H : Vector2H);
108.      end T2;
109.
110.      task T3 is
111.          entry DataFromT4(VC3H : in Vector3H; VD3H : in Vector3H; V : in
Vector; M : in Matrix3H);
112.          entry DataFromT1(a : in Integer; V2H : in Vector2H; M : in
Matrix);
113.          entry B4FromT4(b4 : Integer);
114.          entry ResBT2ToT3(b : Integer);
115.          entry ResZHToT3(ResZH : VectorH);
116.      end T3;
117.
118.
119.      task T4 is
120.          entry DataFromT1(a : in Integer; VH : in VectorH; M : in Matrix);
121.          entry ResBT3ToT4(b : in Integer);
122.      end T4;
123.
124.
125.      task body T1 is
126.          x, b1, res : Integer;
127.          B, E, Z: Vector;
128.          MA: Matrix;
129.
130.          CH, DH, ZH : VectorH;
131.          MBH : MatrixH;
132.
133.      begin
134.          Put_Line("T1 started.");
135.
136.          -- 1) Enter x, B, MA
137.          NumberInput(x);
138.          VectorInput(B);
139.          MatrixInput(MA);
140.
141.          -- 2) Pass x, B3H, MA to T2
142.          T2.DataFromT1(x, B(H+1..N), MA);
143.
144.          -- 3) Get CH, DH, E, MBH from T2
145.          accept DataFromT4(VCH : in VectorH; VDH : in VectorH; V : in
Vector; M : in MatrixH) do
146.              CH := VCH;
147.              DH := VDH;
148.              E := V;
149.              MBH := M;
150.          end DataFromT4;
151.
152.          -- 4) Calculate b1 = BH * CH
153.          b1 := 0;
154.          for i in 1..H loop
155.              b1 := b1 + B(i) * CH(i);

```

```

156.         end loop;
157.
158.         -- 5) Pass b1 to T2
159.         T2.B1FromT1(b1);
160.
161.         -- 6) Get b from T2
162.         accept ResBT2ToT1 (b : in Integer) do
163.             res := b;
164.         end ResBT2ToT1;
165.
166.         -- 7) Calculate  $ZH = b * DH + E * (MA * MBH) * x$ 
167.         Calculate(res, DH, E, MA, MBH, x, ZH);
168.
169.         -- 8) Get Z3H from T2
170.         accept ResZ3HToT1 (ResZ3H : in Vector3H) do
171.             Z(1..H) := ZH;
172.             Z(H+1..N) := ResZ3H;
173.         end ResZ3HToT1;
174.
175.         -- 9) Print Z
176.         if (N < 10) then
177.             delay(0.5);
178.             VectorOutput(Z, "Z");
179.         end if;
180.
181.         Put_Line("T1 finished.");
182.
183.     end T1;
184.
185.     task body T2 is
186.         x, b2, b : Integer;
187.         E : Vector;
188.         MA : Matrix;
189.
190.         ZH : VectorH;
191.         C2H, D2H : Vector2H;
192.         B3H, Z3H : Vector3H;
193.         MB2H : Matrix2H;
194.
195.     begin
196.         Put_Line("T2 started.");
197.
198.         -- 1) Get x, B3H, MA from T1
199.         accept DataFromT1(a : in Integer; V3H : in Vector3H; M : in
Matrix) do
200.             x := a;
201.             B3H := V3H;
202.             MA := M;
203.         end DataFromT1;
204.
205.         -- 2) Pass x, B2H, MA to T3
206.         T3.DataFromT1(x, B3H(H+1..3*H), MA);
207.
208.         -- 3) Get C2H, D2H, E, MB2H from T3

```

```

209.         accept DataFromT4(VC2H : in Vector2H; VD2H : in Vector2H; V : in
Vector; M : in Matrix2H) do
210.             C2H := VC2H;
211.             D2H := VD2H;
212.             E := V;
213.             MB2H := M;
214.         end DataFromT4;
215.
216.         -- 4) Pass CH, DH, E, MBH to T1
217.         T1.DataFromT4(C2H(H+1..2*H), D2H(H+1..2*H), E, MB2H(H+1..2*H));
218.
219.         -- 5) Calculate b2 = BH * CH
220.         b2 := 0;
221.         for i in 1..H loop
222.             b2 := b2 + B3H(i) * C2H(i);
223.         end loop;
224.
225.         -- 6) Get b1 from T1
226.         -- 7) Calculate b2 = b2 + b1
227.         accept B1FromT1 (b1 : in Integer) do
228.             b2 := b2 + b1;
229.         end B1FromT1;
230.
231.         -- 8) Get b3 from the problem T3
232.         accept B3FromT3 (b3 : in Integer) do
233.             b2 := b2 + b3;
234.         end B3FromT3;
235.
236.         -- 9) Calculate b = b2 + b3
237.         b:= b2;
238.
239.         -- 10) Pass b to T1, T3
240.         T1.ResBT2ToT1(b);
241.         T3.ResBT2ToT3(b);
242.
243.         -- 11) Calculate ZH = b * DH + E * (MA * MBH) * x
244.         Calculate(b2, D2H(1..H), E, MA, MB2H(1..H), x, ZH);
245.
246.         -- 12) Get Z2H from T3
247.         accept ResZ2HToT2 (ResZ2H : in Vector2H) do
248.             Z3H(1..H) := ZH;
249.             Z3H(H+1..3*H) := ResZ2H;
250.         end ResZ2HToT2;
251.
252.         -- 13) Pass Z3H to T1
253.         T1.ResZ3HToT1(Z3H);
254.
255.         Put_Line("T2 finished.");
256.     end T2;
257.
258.     task body T3 is
259.         x, b3, res : Integer;
260.         E : Vector;
261.         MA : Matrix;

```

```

262.
263.      ZH : VectorH;
264.      B2H, Z2H : Vector2H;
265.      C3H, D3H : Vector3H;
266.      MB3H : Matrix3H;
267.
268.      begin
269.          Put_Line("T3 started.");
270.
271.          -- 1) Get C3H, D3H, E, MB3H from T4
272.          accept DataFromT4(VC3H : in Vector3H; VD3H : in Vector3H; V : in
Vector; M : in Matrix3H) do
273.              C3H := VC3H;
274.              D3H := VD3H;
275.              E := V;
276.              MB3H := M;
277.          end DataFromT4;
278.
279.          -- 2) Get x, B2H, MA from T2
280.          accept DataFromT1(a : in Integer; V2H : in Vector2H; M : in
Matrix) do
281.              x := a;
282.              B2H := V2H;
283.              MA := M;
284.          end DataFromT1;
285.
286.          -- 3) Pass C2H, D2H, E, MB2H to T2
287.          T2.DataFromT4(C3H(H+1..3*H), D3H(H+1..3*H), E, MB3H(H+1..3*H));
288.
289.          -- 4) Pass x, BH, MA to T4
290.          T4.DataFromT1(x, B2H(H+1..2*H), MA);
291.
292.          -- 5) Calculate b3 = BH * CH
293.          b3 := 0;
294.          for i in 1..H loop
295.              b3 := b3 + B2H(i) * C3H(i);
296.          end loop;
297.
298.          -- 6) Get b4 from T4
299.          -- 7) Calculate b3 = b3 + b4
300.          accept B4FromT4 (b4 : in Integer) do
301.              b3 := b3 + b4;
302.          end B4FromT4;
303.
304.          -- 8) Pass b3 to T2
305.          T2.B3FromT3(b3);
306.
307.          -- 9) Get b from T2
308.          accept ResBT2ToT3 (b : in Integer) do
309.              res := b;
310.          end ResBT2ToT3;
311.
312.          -- 10) Pass b to T4
313.          T4.ResBT3ToT4(res);

```

```

314.
315.      -- 11) Calculate  $ZH = b * DH + E * (MA * MBH) * x$ 
316.      Calculate(res, D3H(1..H), E, MA, MB3H(1..H), x, ZH);
317.
318.      -- 12) Get ZH from T4
319.      accept ResZHToT3 (ResZH : in VectorH) do
320.          Z2H(1..H) := ZH;
321.          Z2H(H+1..2*H) := ResZH;
322.      end ResZHToT3;
323.
324.      -- 13) Pass Z2H to T2
325.      T2.ResZ2HToT2(Z2H);
326.
327.      Put_Line("T3 finished.");
328.  end T3;
329.
330.  task body T4 is
331.      x, b4, res : Integer;
332.      C, D, E : Vector;
333.      MB, MA : Matrix;
334.
335.      BH, ZH : VectorH;
336.  begin
337.      Put_Line("T4 started.");
338.
339.      -- 1) Enter C, D, E, MB
340.      VectorInput(C);
341.      VectorInput(D);
342.      VectorInput(E);
343.      MatrixInput(MB);
344.
345.      -- 2) Pass C3H, D3H, E, MB3H to T3
346.      T3.DataFromT4(C(H+1..N), D(H+1..N), E, MB(H+1..N));
347.
348.      -- 3) Get x, BH, MA from T3
349.      accept DataFromT1(a : in Integer; VH : in VectorH; M : in Matrix)
350.  do
351.          x := a;
352.          BH := VH;
353.          MA := M;
354.      end DataFromT1;
355.
356.      -- 4) Calculate  $b4 = BH * CH$ 
357.      b4 := 0;
358.      for i in 1..H loop
359.          b4 := b4 + BH(i) * C(i);
360.      end loop;
361.
362.      -- 5) Pass b4 to T3
363.      T3.B4FromT4(b4);
364.
365.      -- 6) Get b from T3
366.      accept ResBT3ToT4 (b : in Integer) do
367.          res := b;

```

```

367.         end ResBT3ToT4;
368.
369.         -- 7) Calculate  $ZH = b * DH + E * (MA * MBH) * x$ 
370.         Calculate(res, D(1..H), E, MA, MB(1..H), x, ZH);
371.
372.         -- 8) Pass ZH to T3
373.         T3.ResZHToT3(ZH);
374.
375.         Put_Line("T4 finished.");
376.     end T4;
377. begin
378.     Put_Line("");
379. end tasksProcedure;
380.
381. begin
382.     Put_Line("Lab5 started!"); New_Line;
383.     tasksProcedure;
384.     New_Line; Put_Line("Lab5 finished!"); New_Line;
385. end Main;

```