

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №7
з дисципліни «Програмування паралельних комп'ютерних систем»

Виконав:
студент 3 курсу
ФІОТ гр. ІО-43
Крут Владислав

Перевірів:
Корочкін О. В.

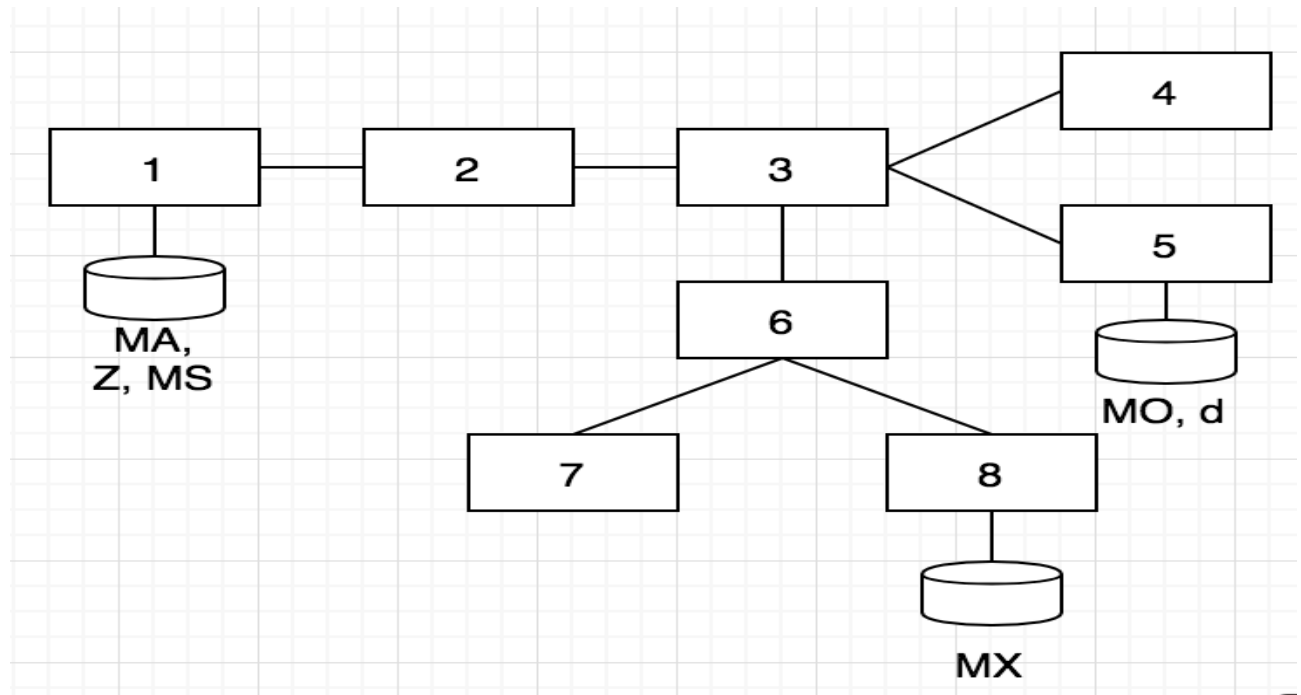
Київ – 2017 р.

ТЕМА: Програмування для комп'ютерних систем з локальною пам'яттю.
Ада. Рандеву

Розробити програму для розв'язання в ПКС зі ЛП математичної задачі: $MA = \max(Z) \cdot MO + d \cdot (MX \cdot MS)$

Мова програмування: Ада

Засоби організації взаємодії: рандеву



Структурна схема ПКС

Виконання роботи:

Етап 1. Побудова паралельного алгоритму

- 1) $\max Z_i = \max(Z_H), i = \overline{1, P}$
- 2) $\max Z = \max(\max Z, \max Z_i), i = \overline{1, P}$
- 3) $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$

Етап 2. Розроблення алгоритмів роботи кожного процесу

Задача T1

1. Введення MS, Z
2. **Передати** MS, Z задачі T2
3. **Прийняти** MX, MO, d від задачі T2
4. Обчислити $\max Z_1 = \max(Z_H)$
5. **Передати** $\max Z_1$ задачі T2
6. **Прийняти** $\max Z$ від задачі T2
7. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
8. **Прийняти** MA_{7H} задачі T2
9. **Вивести** MA

Задача T2

1. **Прийняти** MS, Z від задачі T1

2. **Передати** MS, Z задачі **T3**
3. **Прийняти** MX, MO, d від задачі **T3**
4. **Передати** MX, MO, d задачі **T1**
5. Обчислити $\max Z_2 = \max(Z_H)$
6. **Прийняти** $\max Z_1$ від задачі **T1**
7. Обчислити $\max Z_2 = \max(\max Z_2, \max Z_1)$
8. **Передати** $\max Z_2$ задачі **T3**
9. **Прийняти** $\max Z$ від задачі **T3**
10. **Передати** $\max Z$ задачі **T1**
11. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
12. **Прийняти** MA_{6H} від задачі **T3**
13. **Передати** MA_{2H} задачі **T1**

Задача T3

1. **Прийняти** MS, Z від задачі **T2**
2. **Прийняти** MO, d від задачі **T5**
3. **Прийняти** MX від задачі **T6**
4. **Передати** Z, MS, MO, d, MX задачі **T4**
5. **Передати** Z, MS, MX задачі **T5**
6. **Передати** MS, Z, MO, d задачі **T6**
7. **Передати** MX, MO, d задачі **T2**
8. Обчислити $\max Z_3 = \max(Z_H)$
9. **Прийняти** $\max Z_2$ від задачі **T2**
10. Обчислити $\max Z_3 = \max(\max Z_3, \max Z_2)$
11. **Прийняти** $\max Z_4$ від задачі **T4**
12. Обчислити $\max Z_3 = \max(\max Z_3, \max Z_4)$
13. **Прийняти** $\max Z_5$ від задачі **T5**
14. Обчислити $\max Z_3 = \max(\max Z_3, \max Z_5)$
15. **Прийняти** $\max Z_6$ від задачі **T6**
16. Обчислити $\max Z = \max(\max Z_3, \max Z_6)$
17. **Передати** $\max Z$ задачі **T4, T5, T2, T6**
18. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
19. **Прийняти** MA_{3H} від задачі **T6**
20. **Прийняти** MA_H від задачі **T4, T5**
21. **Передати** MA_{6H} задачі **T2**

Задача T4

1. **Прийняти** Z, MS, MO, d, MX від задачі **T3**
2. Обчислити $\max Z_4 = \max(Z_H)$
3. **Передати** $\max Z_4$ задачі **T3**
4. **Прийняти** $\max Z$ від задачі **T3**
5. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
6. **Передати** MA_H задачі **T3**

Задача T5

1. Введення MO, d
2. **Прийняти** Z, MS, MX від задачі **T3**
3. **Передати** MO, d задачі **T3**

4. Обчислити $\max Z_5 = \max (Z_H)$
5. **Передати** $\max Z_5$ задачі **T3**
6. **Прийняти** $\max Z$ від задачі **T3**
7. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
8. **Передати** MA_H задачі **T3**

Задача T6

1. **Прийняти** Z, MS, MO, d від задачі **T3**
2. **Прийняти** MX від задачі **T8**
3. **Передати** Z, MS, MO, d задачі **T8**
4. **Передати** Z, MS, MO, d, MX задачі **T7**
5. **Передати** MX задачі **T3**
6. Обчислити $\max Z_6 = \max (Z_H)$
7. **Прийняти** $\max Z_7$ від задачі **T7**
8. Обчислити $\max Z_6 = \max (\max Z_6, \max Z_7)$
9. **Прийняти** $\max Z_8$ від задачі **T8**
10. Обчислити $\max Z_6 = \max (\max Z_6, \max Z_8)$
11. **Передати** $\max Z_6$ задачі **T3**
12. **Прийняти** $\max Z$ від задачі **T3**
13. **Передати** $\max Z$ задачі **T7, T8**
14. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
15. **Прийняти** MA_H від задачі **T7, T8**
16. **Передати** MA_{3H} задачі **T3**

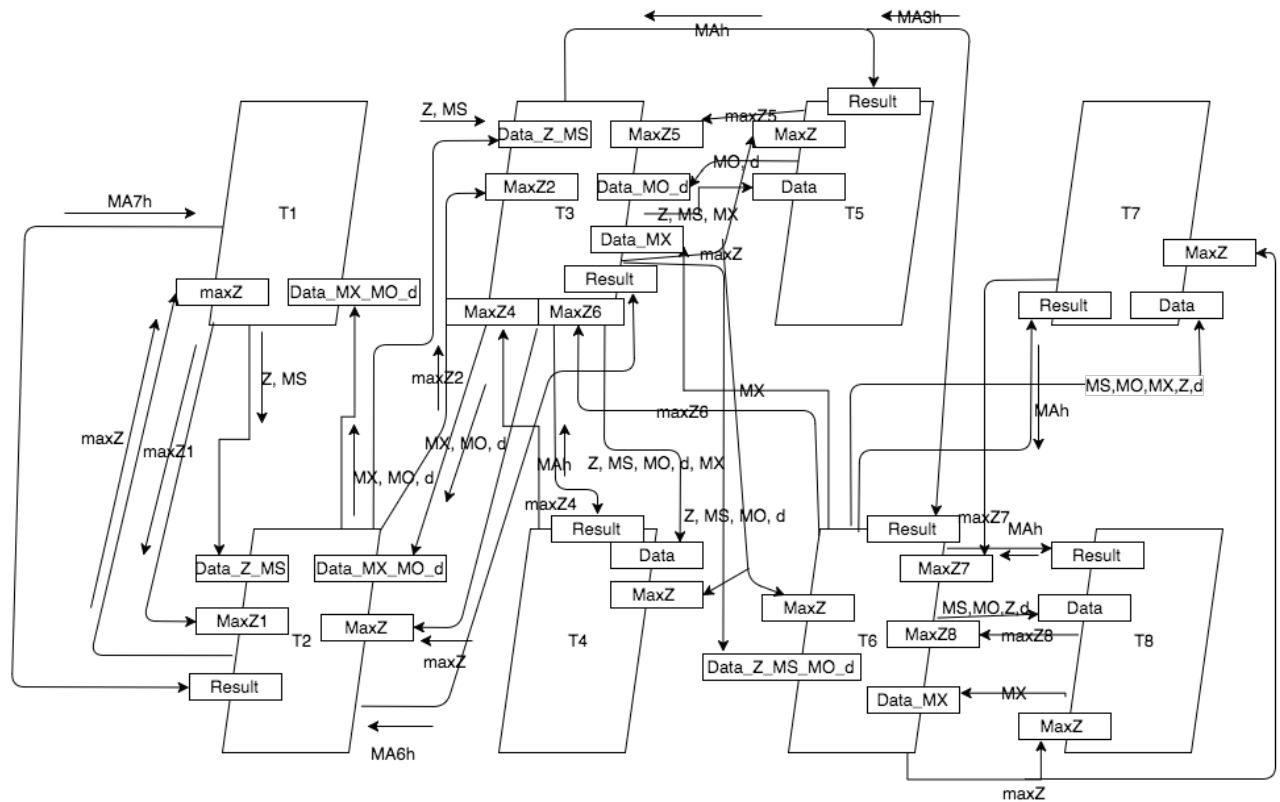
Задача T7

1. **Прийняти** Z, MS, MO, d , від задачі **T6**
2. Обчислити $\max Z_7 = \max (Z_H)$
3. **Передати** $\max Z_7$ задачі **T6**
4. **Прийняти** $\max Z$ від задачі **T6**
5. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
6. **Передати** MA_H задачі **T6**

Задача T8

1. Введення MX
2. **Передати** MX задачі **T6**
3. **Прийняти** Z, MS, MO, d від задачі **T6**
4. Обчислити $\max Z_8 = \max (Z_H)$
5. **Передати** $\max Z_8$ задачі **T6**
6. **Прийняти** $\max Z$ від задачі **T6**
7. Обчислити $MA_H = \max Z \cdot MO_H + d \cdot (MX_H \cdot MS)$
8. **Передати** MA_H задачі **T6**

Етап 3. Розроблення структурної схеми взаємодії задач



Етап 4. Розроблення програми

GNAT GPL 2017 (20170515-63)

Copyright 1992-2017, Free Software Foundation, Inc.

Compiling: main.adb

Source file time stamp: 2017-06-08 21:49:34

Compiled at: 2017-06-09 00:52:24

```

1.
2. -----
3. --
4. --          Parallel and Distributed Computing          --
5. --          Laboratory work #7. Ada. Rendezvous        --
6. --
7. -- File: main.adb
8. -- Task: MA = max(Z)*MO + d * (MX * MS)
9. --
10. -- Author: Krut Vladyslav, group IO-43
11. -- Date: 08.06.2017
12. --
13. -----
14.
15.
16. with Ada.Text_IO, Ada.Integer_text_io, Ada.Synchronous_Task_Control, Data;
17. use Ada.Text_IO, Ada.Integer_text_io, Ada.Synchronous_Task_Control;
18.
19. procedure Main is
20.
21.
22.   Value : Integer := 1;
23.   N : Integer := 8;
24.   P : Integer := 8;
25.   H : Integer := N/P;
26.
27.   package DataN is new Data(N, H);
28.   use DataN;
29.
30.   procedure StartTasks is
31.     ----- СПЕЦИФІКАЦІЯ ЗАДАЧ -----
32.
33.     --          Задача T1          --
34.     task T1 is
35.       entry Data_MX_MO_d(MX: in MatrixN;

```

```

36.          MO: in MatrixN;
37.          d: in Integer);
38.    entry MaxZ (maxZ : in Integer);
39.  end T1;
40.
41.  --          Задача T2          --
42.  task T2 is
43.    entry Data_MX_MO_d(MX: in MatrixN;
44.      MO: in MatrixN;
45.      d: in Integer);
46.    entry Data_Z_MS(Z: in VectorN;
47.      MS: in MatrixN);
48.
49.    entry MaxZ1 (maxZ1 : in Integer);
50.    entry MaxZ (maxZ : in Integer);
51.    entry Result(MA : out Matrix7H);
52.  end T2;
53.
54.  --          Задача T3          --
55.  task T3 is
56.    entry Data_MX(MX: in MatrixN);
57.    entry Data_MO_d(MO: in MatrixN;
58.      d: in Integer);
59.    entry Data_Z_MS(Z: in VectorN;
60.      MS: in MatrixN);
61.
62.    entry MaxZ2 (maxZi : in Integer);
63.    entry MaxZ4 (maxZi : in Integer);
64.    entry MaxZ5 (maxZi : in Integer);
65.    entry MaxZ6 (maxZi : in Integer);
66.    entry Result(MA : out Matrix6H);
67.
68.  end T3;
69.
70.
71.
72.  --          Задача T4          --
73.  task T4 is
74.    entry Data(MS: in MatrixN;
75.      MX: in MatrixN;
76.      MO: in MatrixN;
77.      Z: in VectorN;
78.      d: in Integer);
79.
80.    entry MaxZ (maxZ : in Integer);
81.    entry Result(MA : out Matrix1H);
82.  end T4;
83.
84.  --          Задача T5          --
85.  task T5 is
86.    entry Data(MS: in MatrixN;
87.      MX: in MatrixN;
88.      Z: in VectorN);
89.    entry MaxZ (maxZ : in Integer);
90.    entry Result(MA : out Matrix1H);
91.  end T5;
92.
93.  --          Задача T6          --
94.  task T6 is
95.    entry Data_Z_MS_MO_d(Z: in VectorN;
96.      MS: in MatrixN;
97.      MO: in MatrixN;
98.      d: in Integer);
99.    entry Data_MX(MX: in MatrixN);
100.
101.    entry MaxZ7 (maxZi : in Integer);
102.    entry MaxZ8 (maxZi : in Integer);
103.
104.    entry MaxZ (maxZ : in Integer);
105.
106.    entry Result(MA : out Matrix3H);
107.  end T6;
108.

```

```

109. -- Задача T7 --
110. task T7 is
111.   entry Data(MS: in MatrixN;
112.     MX: in MatrixN;
113.     MO: in MatrixN;
114.     Z: in VectorN;
115.     d: in Integer);
116.
117.   entry MaxZ (maxZ : in Integer);
118.   entry Result(MA : out Matrix1H);
119. end T7;
120.
121. -- Задача T8 --
122. task T8 is
123.   entry Data(MS: in MatrixN;
124.     MO: in MatrixN;
125.     Z: in VectorN;
126.     d: in Integer);
127.
128.   entry MaxZ (maxZ : in Integer);
129.   entry Result(MA : out Matrix1H);
130. end T8;
131.
132. -----
133. -- ТІЛА ЗАДАЧ --
134. -----
135.
136. -- Задача T1 --
137.
138. task body T1 is
139.   MA1 : MatrixN;
140.   Z1 : VectorN;
141.   MS1 : MatrixN;
142.   MO1 : MatrixN;
143.   MX1 : MatrixN;
144.   d1: Integer;
145.   maxZ1: Integer := -99999;
146. begin
147.   Put_Line("T1 started");
148.   --1
149.   Input(MS1,1);
150.   Input(Z1,1);
151.
152.   --2
153.   T2.Data_Z_MS(Z1, MS1);
154.
155.   --3
156.   accept Data_MX_MO_d (MX : in MatrixN; MO : in MatrixN; d : in Integer) do
157.     MX1 := MX;
158.     MO1 := MO;
159.     d1 := d;
160.   end Data_MX_MO_d;
161.
162.   --4
163.   FindMaxZ(Z1(1..H), maxZ1);
164.
165.   --5
166.   T2.MaxZ1(maxZ1);
167.
168.   --6
169.   accept MaxZ (maxZ : in Integer) do
170.     maxZ1:=maxZ;
171.   end MaxZ;
172.
173.   --7
174.   Calculation(d1, maxZ1, MO1, MX1, MS1, MA1);
175.
176.   --8
177.   T2.Result(MA1(H+1..N));
178.
179.   if N < 20 then
180.     Output(MA1);
181.   end if;

```

```

182.
183.     Put_Line("T1 finished");
184.
185. end T1;
186.
187. --                Задача T2                --
188.
189. task body T2 is
190.     MA2 : MatrixN;
191.     Z2 : VectorN;
192.     MS2 : MatrixN;
193.     MO2 : MatrixN;
194.     MX2 : MatrixN;
195.     d2: Integer;
196.     maxZ2: Integer := -99999;
197. begin
198.     Put_Line("T2 started");
199.
200.     --1
201.     accept Data_Z_MS (Z : in VectorN; MS : in MatrixN) do
202.         Z2 := Z;
203.         MS2 := MS;
204.     end Data_Z_MS;
205.
206.     --2
207.     T3.Data_Z_MS(Z2, MS2);
208.
209.     --3
210.     accept Data_MX_MO_d (MX : in MatrixN; MO : in MatrixN; d : in Integer) do
211.         MX2 := MX;
212.         MO2 := MO;
213.         d2 := d;
214.     end Data_MX_MO_d;
215.
216.     --4
217.     T1.Data_MX_MO_d(MX2, MO2, d2);
218.
219.     --5
220.     FindMaxZ(Z2(H+1..2*H), maxZ2);
221.
222.     --6
223.     accept MaxZ1 (maxZ1 : in Integer) do
224.         --7
225.         maxZ2 := Max(maxZ1, maxZ2);
226.     end MaxZ1;
227.
228.     --8
229.     T3.MaxZ2(maxZ2);
230.
231.     --9
232.     accept MaxZ (maxZ : in Integer) do
233.         maxZ2:=maxZ;
234.     end MaxZ;
235.
236.     --10
237.     T1.MaxZ(maxZ2);
238.
239.     --11
240.     Calculation(d2, maxZ2, MO2, MX2, MS2, MA2);
241.
242.     --12
243.     T3.Result(MA2(2*H+1..N));
244.
245.     accept Result (MA : out Matrix7H) do
246.         MA := MA2(H+1..N);
247.     end Result;
248.
249.     Put_Line("T2 finished");
250. end T2;
251.
252. --                Задача T3                --
253.
254. task body T3 is

```



```

255.    MA3 : MatrixN;
256.    Z3 : VectorN;
257.    MS3 : MatrixN;
258.    MO3 : MatrixN;
259.    MX3 : MatrixN;
260.    d3: Integer;
261.    maxZ3: Integer := -99999;
262.  begin
263.    Put_Line("T3 started");
264.
265.    --1
266.    accept Data_Z_MS (Z : in VectorN; MS : in MatrixN) do
267.      Z3 := Z;
268.      MS3 := MS;
269.    end Data_Z_MS;
270.
271.    --2
272.    accept Data_MO_d (MO : in MatrixN; d : in Integer) do
273.      MO3 := MO;
274.      d3 := d;
275.    end Data_MO_d;
276.
277.    --3
278.    accept Data_MX (MX : in MatrixN) do
279.      MX3 := MX;
280.    end Data_MX;
281.
282.    --4
283.    T4.Data(MS3, MX3, MO3, Z3, d3);
284.
285.    --5
286.    T5.Data(MS3, MX3, Z3);
287.
288.    --6
289.    T6.Data_Z_MS_MO_d(Z3, MS3, MO3, d3);
290.
291.    --7
292.    T2.Data_MX_MO_d(MX3, MO3, d3);
293.
294.    --8
295.    FindMaxZ(Z3(2*H+1..3*H), maxZ3);
296.
297.    --9
298.    accept MaxZ2 (maxZi : in Integer) do
299.      --10
300.      maxZ3 := Max(maxZ3, maxZi);
301.    end MaxZ2;
302.
303.    --11
304.    accept MaxZ4 (maxZi : in Integer) do
305.      --12
306.      maxZ3 := Max(maxZ3, maxZi);
307.    end MaxZ4;
308.
309.    --13
310.    accept MaxZ5 (maxZi : in Integer) do
311.      --14
312.      maxZ3 := Max(maxZ3, maxZi);
313.    end MaxZ5;
314.
315.    --15
316.    accept MaxZ6 (maxZi : in Integer) do
317.      --16
318.      maxZ3 := Max(maxZ3, maxZi);
319.    end MaxZ6;
320.
321.    --17
322.    T2.MaxZ(maxZ3);
323.    T4.MaxZ(maxZ3);
324.    T5.MaxZ(maxZ3);
325.    T6.MaxZ(maxZ3);
326.
327.    --18

```

```

328. Calculation(d3, maxZ3, MO3, MX3, MS3, MA3);
329.
330. --19
331. T6.Result(MA3(5*H+1..N));
332.
333. --20
334. T4.Result(MA3(3*H+1..4*H));
335. T5.Result(MA3(4*H+1..5*H));
336.
337. --21
338. accept Result (MA : out Matrix6H) do
339.   MA := MA3(2*H+1..N);
340. end Result;
341.
342. Put_Line("T3 finished");
343.
344. end T3;
345.
346. --          Задача T4          --
347.
348. task body T4 is
349.   MA4 : MatrixN;
350.   Z4 : VectorN;
351.   MS4 : MatrixN;
352.   MO4 : MatrixN;
353.   MX4 : MatrixN;
354.   d4: Integer;
355.   maxZ4: Integer := -99999;
356. begin
357.   Put_Line("T4 started");
358.
359.   --1
360.   accept Data (MS : in MatrixN; MX : in MatrixN; MO : in MatrixN; Z : in VectorN; d : in Integer) do
361.     MS4 := MS;
362.     MX4 := MX;
363.     MO4 := MO;
364.     Z4 := Z;
365.     d4 := d;
366.   end Data;
367.
368.   --2
369.   FindMaxZ(Z4(3*H+1..4*H), maxZ4);
370.
371.   --3
372.   T3.MaxZ4(maxZ4);
373.
374.   --4
375.   accept MaxZ (maxZ : in Integer) do
376.     maxZ4 := maxZ;
377.   end MaxZ;
378.
379.   --5
380.   Calculation(d4, maxZ4, MO4, MX4, MS4, MA4);
381.
382.   accept Result (MA : out Matrix1H) do
383.     MA := MA4(3*H+1..4*H);
384.   end Result;
385.
386.   Put_Line("T4 finished");
387. end T4;
388.
389. --          Задача T5          --
390.
391. task body T5 is
392.   MA5 : MatrixN;
393.   Z5 : VectorN;
394.   MS5 : MatrixN;
395.   MO5 : MatrixN;
396.   MX5 : MatrixN;
397.   d5: Integer;
398.   maxZ5: Integer := -99999;
399.
400.

```

```

401. begin
402.   Put_Line("T5 started");
403.
404.   --1
405.   Input(MO5, 1);
406.   d5 := 1;
407.
408.   --3
409.   T3.Data_MO_d(MO5, d5);
410.
411.   --2
412.   accept Data (MS : in MatrixN; MX : in MatrixN; Z : in VectorN) do
413.     MS5 := MS;
414.     MX5 := MX;
415.     Z5 := Z;
416.   end Data;
417.
418.   --4
419.   FindMaxZ(Z5(4*H+1..5), maxZ5);
420.
421.   --5
422.   T3.MaxZ5(maxZ5);
423.
424.   --6
425.   accept MaxZ (maxZ : in Integer) do
426.     maxZ5 := maxZ;
427.   end MaxZ;
428.
429.   --7
430.   Calculation(d5, maxZ5, MO5, MX5, MS5, MA5);
431.
432.   --8
433.   accept Result (MA : out Matrix1H) do
434.     MA := MA5(4*H+1..5*H);
435.   end Result;
436.
437.   Put_Line("T5 finished");
438. end T5;
439.
440. --              Задача T6              --
441.
442. task body T6 is
443.   MA6 : MatrixN;
444.   Z6 : VectorN;
445.   MS6 : MatrixN;
446.   MO6 : MatrixN;
447.   MX6 : MatrixN;
448.   d6: Integer;
449.   maxZ6: Integer := -99999;
450. begin
451.   Put_Line("T6 started");
452.
453.   --2
454.   accept Data_MX (MX : in MatrixN) do
455.     MX6 := MX;
456.   end Data_MX;
457.
458.   --5
459.   T3.Data_MX(MX6);
460.
461.   --1
462.   accept Data_Z_MS_MO_d (Z : in VectorN; MS : in MatrixN; MO : in MatrixN; d : in Integer) do
463.     Z6 := Z;
464.     MS6 := MS;
465.     MO6 := MO;
466.     d6 := d;
467.   end Data_Z_MS_MO_d;
468.
469.   --3
470.   T8.Data(MS6, MO6, Z6, d6);
471.
472.   --4
473.   T7.Data(MS6, MX6, MO6, Z6, d6);

```

```

474.
475.      --6
476.      FindMaxZ(Z6(5*H+1..6*H), maxZ6);
477.
478.      --7
479.      accept MaxZ7 (maxZi : in Integer) do
480.          --8
481.              maxZ6 := Max(maxZi, maxZ6);
482.          end MaxZ7;
483.
484.      --9
485.      accept MaxZ8 (maxZi : in Integer) do
486.          --10
487.              maxZ6 := Max(maxZi, maxZ6);
488.          end MaxZ8;
489.
490.      --11
491.      T3.MaxZ6(maxZ6);
492.
493.      --12
494.      accept MaxZ (maxZ : in Integer) do
495.          maxZ6 := maxZ;
496.      end MaxZ;
497.
498.      --13
499.      T7.MaxZ(maxZ6);
500.      T8.MaxZ(maxZ6);
501.
502.      --14
503.      Calculation(d6, maxZ6, MO6, MX6, MS6, MA6);
504.
505.      --15
506.      T7.Result(MA6(6*H+1..7*H));
507.
508.      T8.Result(MA6(7*H+1..N));
509.
510.      --16
511.      accept Result (MA : out Matrix3H) do
512.          MA := MA6(5*H+1..N);
513.      end Result;
514.
515.      Put_Line("T6 finished");
516.  end T6;
517.
518.      --                      Задача T7                      --
519.
520.  task body T7 is
521.      MA7 : MatrixN;
522.      Z7 : VectorN;
523.      MS7 : MatrixN;
524.      MO7 : MatrixN;
525.      MX7 : MatrixN;
526.      d7: Integer;
527.      maxZ7: Integer := -99999;
528.  begin
529.      Put_Line("T7 started");
530.
531.      --1
532.      accept Data (MS : in MatrixN; MX : in MatrixN; MO : in MatrixN; Z : in VectorN; d : in Integer) do
533.          MS7 := MS;
534.          MX7 := MX;
535.          MO7 := MO;
536.          Z7 := Z;
537.          d7 := d;
538.      end Data;
539.
540.      --2
541.      FindMaxZ(Z7(6*H+1..7*H), maxZ7);
542.
543.      --3
544.      T6.MaxZ7(maxZ7);
545.
546.      --4

```

```

547.    accept MaxZ (maxZ : in Integer) do
548.        maxZ7 := maxZ;
549.    end MaxZ;
550.
551.    --5
552.    Calculation(d7, maxZ7, MO7, MX7, MS7, MA7);
553.
554.    --6
555.    accept Result (MA : out Matrix1H) do
556.        MA := MA7(6*H+1..7*H);
557.    end Result;
558.
559.    Put_Line("T7 finished");
560. end T7;
561.
562.    --                      Задача T8                      --
563.
564. task body T8 is
565.     MA8 : MatrixN;
566.     Z8 : VectorN;
567.     MS8 : MatrixN;
568.     MO8 : MatrixN;
569.     MX8 : MatrixN;
570.     d8: Integer;
571.     maxZ8: Integer := -99999;
572. begin
573.     Put_Line("T8 started");
574.
575.     --1
576.     Input(MX8, 1);
577.
578.     --2
579.     T6.Data_MX(MX8);
580.
581.     --3
582.     accept Data (MS : in MatrixN; MO : in MatrixN; Z : in VectorN; d : in Integer) do
583.         MS8 := MS;
584.         MO8 := MO;
585.         Z8 := Z;
586.         d8 := d;
587.     end Data;
588.
589.     --4
590.     FindMaxZ(Z8(7*H+1..N), maxZ8);
591.
592.     --5
593.     T6.MaxZ8(maxZ8);
594.
595.     --6
596.     accept MaxZ (maxZ : in Integer) do
597.         maxZ8 := maxZ;
598.     end MaxZ;
599.
600.     --7
601.     Calculation(d8, maxZ8, MO8, MX8, MS8, MA8);
602.
603.     accept Result (MA : out Matrix1H) do
604.         MA := MA8(7*H+1..N);
605.     end Result;
606.
607.     Put_Line("T8 finished");
608. end T8;
609.
610. begin
611.    null;
612. end StartTasks;
613. begin
614.    Put_Line ("Lab7 started");
615.    StartTasks;
616.    Put_Line ("Lab7 finished");
617. end Main;
618.

```

618 lines: No errors