

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

Звіт до лабораторної роботи №6
«Програмування для комп'ютерних систем з локальною
пам'яттю. Мова Ада. Рандеву»
з дисципліни **«Програмне забезпечення високопродуктивних**
комп'ютерних систем»

Виконала:
студентка групи ІМ-13
Мартинюк Марія Павлівна
номер у списку групи 13

Перевірив:
Корочкін О. В.

Київ 2024

Мета роботи: розробка програми для ПКС з ЛП

Мова програмування: Ada

Засоби організації взаємодії процесів: механізм рандеву

Вхідні дані:

- комп'ютерна система з локальною пам'яттю включає 8 процесорів і три пристрої введення-виведення; структура 1 - лінійна (рис. 1.1);
- математичне завдання згідно **варіанту №11**:
$$A = \max(C) * Z + D * (MX * MS),$$

де C, Z, D - вектори розміру N ; MX, MS - матриці розміру N ;
- введення значень MX та виведення результату A виконується у процесорі 1;
введення значень Z, D виконується у процесорі 3;
введення значень C, MD виконується у процесорі 4.

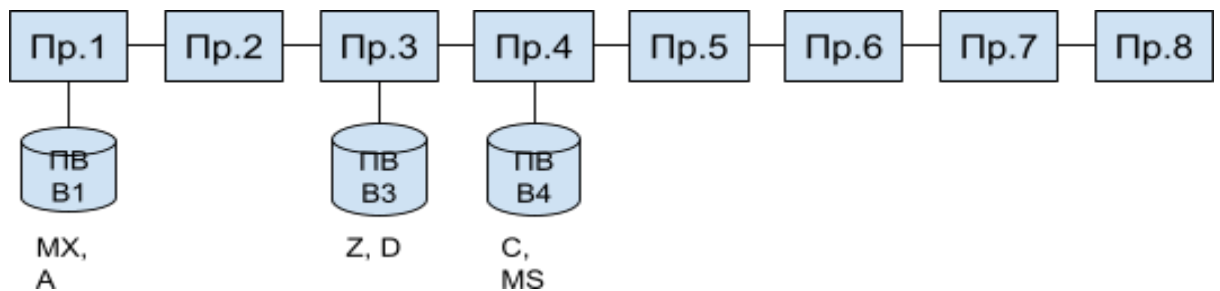


рис. 1.1 Структурна схема 8 процесорної системи

ВИКОНАННЯ РОБОТИ

Етап 1. Побудова паралельного алгоритму

$$A = \max(C) * Z + D * (MX * MS)$$

1. $a_i = \max(C)$, $i = 1 \dots P$

2. $a = \max(a, a_i)$

3. $A_n = a * Z_n + D * (MX * MS_n)$

, де $n = N/P$ частина вектору або матриці, N - розмір вектору/матриці, $P = 8$ - кількість процесорів (потоків).

Етап 2. Розробка алгоритмів потоків

Задача T1:

1. Введення MX
2. **Передати** задачі T2 дані MX
3. **Прийняти** від задачі T2 дані Z_n, D, C_n, MS_n
4. Обчислення $a_1 = \max(C_n)$
5. **Передати** задачі T2 дані a_1
6. **Прийняти** від задачі T2 дані a
7. Обчислення $A_n = a * Z_n + D * (MX * MS_n)$
8. **Прийняти** від задачі T2 результат - вектор A_{7n}
9. Виведення результату - вектор A

Задача T2:

1. **Прийняти** від задачі T1 дані MX
2. **Прийняти** від задачі T3 дані $Z_{2n}, D, C_{2n}, MS_{2n}$
3. **Передати** задачі T3 дані MX
4. **Передати** задачі T1 дані Z_n, D, C_n, MS_n
5. Обчислення $a_2 = \max(C_n)$
6. **Прийняти** від задачі T1 дані a_1
7. Обчислення $a_{12} = \max(a_1, a_2)$
8. **Передати** задачі T3 дані a_{12}
9. **Прийняти** від задачі T3 дані a
10. **Передати** задачі T1 дані a
11. Обчислення $A_n = a_2 * Z_n + D * (MX * MS_n)$
12. **Прийняти** від задачі T3 результат - вектор A_{6n}

13. **Передати** задачі T1 результат - вектор A7н

Задача T3:

1. Введення Z, D
2. **Прийняти** від задачі T4 дані C3н, MS3н
3. **Передати** задачі T2 дані Z2н, D, C2н, MS2н
4. **Прийняти** від задачі T2 дані MX
5. **Передати** задачі T4 дані Z5н, D, MX
6. Обчислення $a3 = \max(Cн)$
7. **Прийняти** від задачі T2 дані a12
8. Обчислення $a123 = \max(a12, a3)$
9. **Передати** задачі T4 дані a123
10. **Прийняти** від задачі T4 дані a
11. **Передати** задачі T2 дані a
12. Обчислення $Aн = a * Zн + D * (MX * MSн)$
13. **Прийняти** від задачі T4 результат - вектор A5н
14. **Передати** задачі T2 результат - вектор A6н

Задача T4:

1. Введення C, MS
2. **Передати** задачі T3 дані C3н, MS3н
3. **Прийняти** від задачі T3 дані Z5н, D, MX
4. **Передати** задачі T5 дані C4н, MS4н, Z4н, D, MX
5. Обчислення $a4 = \max(Cн)$
6. **Прийняти** від задачі T3 дані a123
7. Обчислення $a1234 = \max(a123, a4)$
8. **Передати** задачі T5 дані a1234
9. **Прийняти** від задачі T5 дані a
10. **Передати** задачі T3 дані a
11. Обчислення $Aн = a * Zн + D * (MX * MSн)$
12. **Прийняти** від задачі T5 результат - вектор A4н
13. **Передати** задачі T3 результат - вектор A5н

Задача T5:

1. **Прийняти** від задачі T4 дані C4н, MS4н, Z4н, D, MX
2. **Передати** задачі T6 дані C3н, MS3н, Z3н, D, MX
3. Обчислення $a5 = \max(Cн)$

4. **Прийняти** від задачі T4 дані a1234
5. Обчислення $a12345 = \max(a1234, a5)$
6. **Прийняти** від задачі T6 дані a678
7. Обчислення $a = \max(a12345, a678)$
8. **Передати** задачі T4 дані a
9. **Передати** задачі T6 дані a
10. Обчислення $A_n = a * Z_n + D * (MX * MS_n)$
11. **Прийняти** від задачі T6 результат - вектор A3n
12. **Передати** задачі T4 результат - вектор A4n

Задача T6:

1. **Прийняти** від задачі T5 дані C3n, MS3n, Z3n, D, MX
2. **Передати** задачі T7 дані C2n, MS2n, Z2n, D, MX
3. Обчислення $a6 = \max(C_n)$
4. **Прийняти** від задачі T7 дані a78
5. Обчислення $a678 = \max(a78, a6)$
6. **Передати** задачі T5 дані a678
7. **Прийняти** від задачі T5 дані a
8. **Передати** задачі T7 дані a
9. Обчислення $A_n = a * Z_n + D * (MX * MS_n)$
10. **Прийняти** від задачі T7 результат - вектор A2n
11. **Передати** задачі T5 результат - вектор A3n

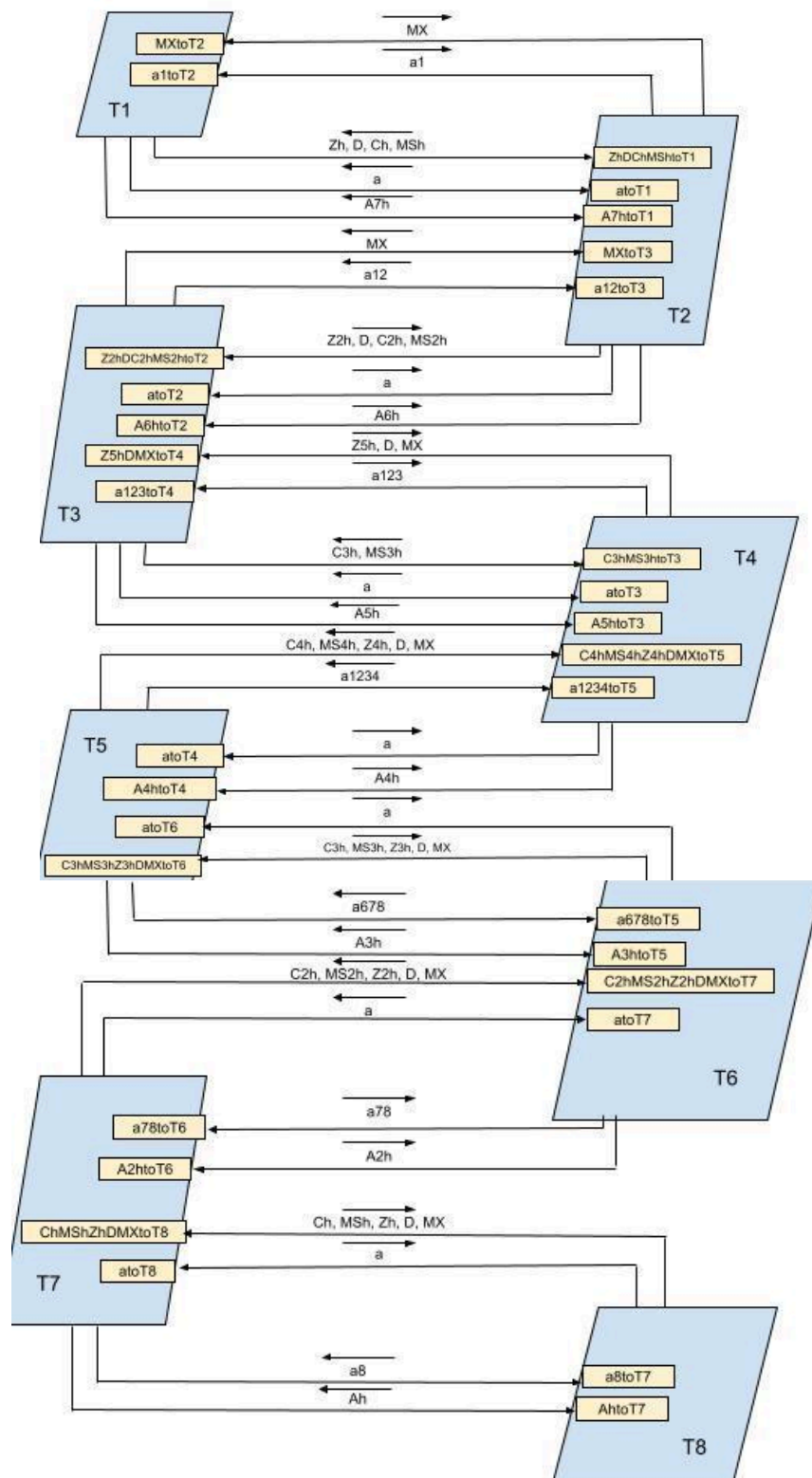
Задача T7:

1. **Прийняти** від задачі T6 дані C2n, MS2n, Z2n, D, MX
2. **Передати** задачі T8 дані Cn, MSn, Zn, D, MX
3. Обчислення $a7 = \max(C_n)$
4. **Прийняти** від задачі T8 дані a8
5. Обчислення $a78 = \max(a8, a7)$
6. **Передати** задачі T6 дані a78
7. **Прийняти** від задачі T6 дані a
8. **Передати** задачі T8 дані a
9. Обчислення $A_n = a * Z_n + D * (MX * MS_n)$
10. **Прийняти** від задачі T8 результат - вектор An
11. **Передати** задачі T6 результат - вектор A2n

Задача Т8:

1. **Прийняти** від задачі Т7 дані C_n , MS_n , Z_n , D , MX
2. Обчислення $a_8 = \max(C_n)$
3. **Передати** задачі Т7 дані a_8
4. **Прийняти** від задачі Т7 дані a
5. Обчислення $A_n = a * Z_n + D * (MX * MS_n)$
6. **Передати** задачі Т7 результат - вектор A_n

Етап 3. Розробка схеми взаємодії задач



На структурній схемі зображено задачі T1...T8, входи задач із їх розміщенням у відповідних задачах, а також протоколи взаємодії для них, які задають напрям і обсяг даних, що передаються між задачами.

Етап 4. Розробка програми

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;

procedure Lab6 is

    N: integer := 16;
    P: integer := 8;
    H: integer := N/P;

    subtype Index is Integer range 1 .. N;

    type Vector_General is array(Integer range <>) of Integer;
    subtype Vector is Vector_General (1..N);
    subtype Vector7H is Vector_General (1..7*H);
    subtype Vector6H is Vector_General (1..6*H);
    subtype Vector5H is Vector_General (1..5*H);
    subtype Vector4H is Vector_General (1..4*H);
    subtype Vector3H is Vector_General (1..3*H);
    subtype Vector2H is Vector_General (1..2*H);
    subtype VectorH is Vector_General (1..H);

    type Matrix_General is array(integer range <>) of Vector;
    subtype Matrix is Matrix_General (1..N);
    subtype Matrix7H is Matrix_General (1..7*H);
    subtype Matrix6H is Matrix_General (1..6*H);
    subtype Matrix5H is Matrix_General (1..5*H);
    subtype Matrix4H is Matrix_General (1..4*H);
    subtype Matrix3H is Matrix_General (1..3*H);
    subtype Matrix2H is Matrix_General (1..2*H);
    subtype MatrixH is Matrix_General (1..H);

    -- опис специфікації задач
    task T1 is
        entry MXtoT2 (MX_toT2 : out Matrix);
        entry altoT2 (a1_toT2 : out Integer);
    end T1;

    task T2 is
        entry MXtoT3 (MX_toT3 : out Matrix);
        entry ZhDChMShtoT1 (Zh_toT1 : out VectorH; D_toT1 : out Vector;
            Ch_toT1 : out VectorH; MSh_toT1 : out MatrixH);
```



```

    entry a12toT3 (a12_toT3 : out Integer);
    entry atoT1 (a_toT1 : out Integer);
    entry A7htoT1 (A7h_toT1 : out Vector7H);
end T2;

task T3 is
    entry Z2hDC2hMS2htoT2 (Z2h_toT2 : out Vector2H; D_toT2 : out
Vector; C2h_toT2 : out Vector2H; MS2h_toT2 : out Matrix2H);
    entry Z5hDMXtoT4 (Z5h_toT4 : out Vector5H; D_toT4 : out Vector;
MX_toT4 : out Matrix);
    entry a123toT4 (a123_toT4 : out Integer);
    entry atoT2 (a_toT2 : out Integer);
    entry A6htoT2 (A6h_toT2 : out Vector6H);
end T3;

task T4 is
    entry C3hMS3htoT3 (C3h_toT3 : out Vector3H; MS3h_toT3 : out
Matrix3H);
    entry C4hMS4hZ4hDMXtoT5 (C4h_toT5 : out Vector4H; MS4h_toT5 : out
Matrix4H; Z4h_toT5 : out Vector4H; D_toT5 : out Vector; MX_toT5 : out
Matrix);
    entry a1234toT5 (a1234_toT5 : out Integer);
    entry atoT3 (a_toT3 : out Integer);
    entry A5htoT3 (A5h_toT3 : out Vector5H);
end T4;

task T5 is
    entry C3hMS3hZ3hDMXtoT6 (C3h_toT6 : out Vector3H; MS3h_toT6 : out
Matrix3H; Z3h_toT6 : out Vector3H; D_toT6 : out Vector; MX_toT6 : out
Matrix);
    entry atoT4 (a_toT4 : out Integer);
    entry atoT6 (a_toT6 : out Integer);
    entry A4htoT4 (A4h_toT4 : out Vector4H);
end T5;

task T6 is
    entry C2hMS2hZ2hDMXtoT7 (C2h_toT7 : out Vector2H; MS2h_toT7 : out
Matrix2H; Z2h_toT7 : out Vector2H; D_toT7 : out Vector; MX_toT7 : out
Matrix);
    entry a678toT5 (a678_toT5 : out Integer);
    entry atoT7 (a_toT7 : out Integer);
    entry A3htoT5 (A3h_toT5 : out Vector3H);
end T6;

```

```

task T7 is
    entry ChMShZhDMXtoT8 (Ch_toT8 : out VectorH; MSh_toT8 : out
MatrixH; Zh_toT8 : out VectorH; D_toT8 : out Vector; MX_toT8 : out
Matrix);
    entry a78toT6 (a78_toT6 : out Integer);
    entry atoT8 (a_toT8 : out Integer);
    entry A2htoT6 (A2h_toT6 : out Vector2H);
end T7;

```

```

task T8 is
    entry a8toT7 (a8_toT7 : out Integer);
    entry AhtoT7 (Ah_toT7 : out VectorH);
end T8;

```

```

task body T1 is
    MX : Matrix;
    MSh : MatrixH;
    a, a1 : Integer;
    A_res : Vector;
    Zh, Ch, MXMShD : VectorH;
    D : Vector;
    A7h : Vector7H;
    MXMSh : MatrixH;

begin
    Put_Line("Task T1 started.");

    -- Введення MX
    for i in 1..N loop
        for j in 1..N loop
            MX(i)(j) := 1;
        end loop;
    end loop;

    -- Передати задачі T2 дані MX
    accept MXtoT2 (MX_toT2 : out Matrix) do
        MX_toT2 := MX;
    end MXtoT2;

    -- Прийняти від задачі T2 дані Zh, D, Ch, MSh
    T2.ZhDChMShtoT1(Zh, D, Ch, MSh);

```

```

-- Обчислення  $a_1 = \max(C_n)$ 
a1 := Ch(1);
for i in 1..H loop
    if Ch(i) > a1 then
        a1 := Ch(i);
    end if;
end loop;

-- Передати задачу T2 дані a1
accept altoT2 (a1_toT2 : out Integer) do
    a1_toT2 := a1;
end altoT2;

-- Прийняти від задачі T2 дані a
T2.atot1(a);

-- Обчислення  $A_n = a * Z_n + D * (MX * MSh)$ 
--  $MX * MSh$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MX(k)(j) * MSh(i)(k);
        end loop;
    end loop;
end loop;

--  $D * (MX * MSh)$ 
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

--  $A_n = a * Z_n + D * (MX * MSh)$ 
for i in 1..H loop
    A_res(i) := a * Zh(i) + MXMShD(i);
end loop;

-- Прийняти від задачі T2 результат - вектор  $A_7n$ 
T2.A7htoT1(A7h);

```

```

--Виведення результату - вектор A
for i in 1..H loop
    A_res(i) := A_res(i);
end loop;
for i in 1..H*7 loop
    A_res(H+i) := A7h(i);
end loop;

Put_Line("A = ");
for i in 1..N loop
    put(A_res(i), 1);
    put(" ");
end loop;

Put_Line("Task T1 finished");
end T1;

```

```

task body T2 is
    MX : Matrix;
    a1, a2, a12, a : Integer;
    D : Vector;
    Ah, MXMShD : VectorH;
    Z2h, C2h : Vector2H;
    MS2h : Matrix2H;
    A6h : Vector6H;
    MXMSh : MatrixH;

begin
    Put_Line("Task T2 started.");

    -- Прийняти від задачі T1 дані MX
    T1.MXtoT2(MX);

    -- Прийняти від задачі T3 дані Z2н, D, C2н, MS2н
    T3.Z2hDC2hMS2htoT2(Z2h, D, C2h, MS2h);

    -- Передати в задачу T3 дані MX
    accept MXtoT3 (MX_toT3 : out Matrix) do
        MX_toT3 := MX;
    end MXtoT3;

    -- Передати задачі T1 дані Zн, D, Cн, MSн

```

```

    accept ZhDChMShtoT1 (Zh_toT1 : out VectorH; D_toT1 : out Vector;
Ch_toT1 : out VectorH; MSh_toT1 : out MatrixH) do
        Zh_toT1 := Z2h(H+1..H*2);
        D_toT1 := D;
        Ch_toT1 := C2h(H+1..H*2);
        MSh_toT1 := MS2h(H+1..H*2);
    end ZhDChMShtoT1;

-- Обчислення a2 = max(Сн)
a2 := C2h(1);
for i in 1..H loop
    if C2h(i) > a2 then
        a2 := C2h(i);
    end if;
end loop;

-- Прийняти від задачі T1 дані a1
T1.a1toT2(a1);

-- Обчислення a12 = max(a1, a2)
a12 := a1;
if a2 > a12 then
    a12 := a2;
end if;

-- Передати задачі T3 дані a12
accept a12toT3 (a12_toT3 : out Integer) do
    a12_toT3 := a12;
end a12toT3;

-- Прийняти від задачі T3 дані a
T3.atot2(a);

-- Передати задачі T1 дані a
accept atot1 (a_toT1 : out Integer) do
    a_toT1 := a;
end atot1;

-- Обчислення  $A_n = a * Z_n + D * (M_x * M_{Sh})$ 
--  $M_x * M_{Sh}$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
    end loop;
end loop;

```

```

        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MX(k)(j) * MS2h(H+i)(k);
        end loop;
    end loop;
end loop;

-- D * (MX * MSh)
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

-- Ah = a * Zh + D * (MX * MSh)
for i in 1..H loop
    Ah(i) := a * Z2h(i) + MXMShD(i);
end loop;

-- Прийняти від задачі T3 результат - вектор A6h
T3.A6htoT2(A6h);

-- Передати задачі T1 результат - вектор A7h
accept A7htoT1 (A7h_toT1 : out Vector7H) do
    for i in 1..H loop
        A7h_toT1(i) := Ah(i);
    end loop;
    for i in 1..H*6 loop
        A7h_toT1(H+i) := A6h(i);
    end loop;
end A7htoT1;

Put_Line("Task T2 finished");
end T2;

-----

task body T3 is
    Z, D : Vector;
    MX : Matrix;
    a3, a12, a123, a : Integer;
    C3h : Vector3H;
    MS3h : Matrix3H;
    A5h : Vector5H;
    Ah, MXMShD : VectorH;

```

```

MXMSh : MatrixH;

begin
  Put_Line("Task T3 started.");

  -- Введення Z, D
  for i in 1..N loop
    Z(i) := 1;
    D(i) := 1;
  end loop;

  -- Прийняти від задачі T4 дані C3h, MS3h
  T4.C3hMS3htoT3(C3h, MS3h);

  -- Передати задачі T2 дані Z2h, D, C2h, MS2h
  accept Z2hDC2hMS2htoT2 (Z2h_toT2 : out Vector2H; D_toT2 : out
Vector; C2h_toT2 : out Vector2H; MS2h_toT2 : out Matrix2H) do
    Z2h_toT2 := Z(H+1..H*3);
    D_toT2 := D;
    C2h_toT2 := C3h(H+1..H*3);
    MS2h_toT2 := MS3h(H+1..H*3);
  end Z2hDC2hMS2htoT2;

  -- Прийняти від задачі T2 дані MX
  T2.MXtoT3(MX);

  -- Передати задачі T4 дані Z5h, D, MX
  accept Z5hDMXtoT4 (Z5h_toT4 : out Vector5H; D_toT4 : out Vector;
MX_toT4 : out Matrix) do
    Z5h_toT4 := Z(3*H+1..H*8);
    D_toT4 := D;
    MX_toT4 := MX;
  end Z5hDMXtoT4;

  -- Обчислення a3 = max(Cn)
  a3 := C3h(1);
  for i in 1..H loop
    if C3h(i) > a3 then
      a3 := C3h(i);
    end if;
  end loop;

  -- Прийняти від задачі T2 дані a12

```

```

T2.al2toT3(a12);

-- Обчислення a123 = max(a12, a3)
a123 := a3;
if a123 > a12 then
    a123 := a12;
end if;

-- Передати задачі T4 дані a123
accept al23toT4 (al23_toT4 : out Integer) do
    al23_toT4 := a123;
end al23toT4;

-- Прийняти від задачі T4 дані a
T4.atot3(a);

-- Передати задачі T2 дані a
accept atot2 (a_toT2 : out Integer) do
    a_toT2 := a;
end atot2;

-- Обчислення  $A_H = a * Z_H + D * (MX * MSh)$ 
--  $MX * MSh$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MX(k)(j) * MS3h(i)(k);
        end loop;
    end loop;
end loop;

--  $D * (MX * MSh)$ 
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

--  $A_H = a * Z_H + D * (MX * MSh)$ 
for i in 1..H loop
    Ah(i) := a * Z(i) + MXMShD(i);
end loop;

```



```

    end loop;

    -- Прийняти від задачі T4 результат - вектор A5н
    T4.A5htoT3(A5h);

    -- Передати задачі T2 результат - вектор A6н
    accept A6htoT2 (A6h_toT2 : out Vector6H) do
        for i in 1..H loop
            A6h_toT2(i) := Ah(i);
        end loop;
        for i in 1..H*5 loop
            A6h_toT2(H+i) := A5h(i);
        end loop;
    end A6htoT2;

    Put_Line("Task T3 finished");
end T3;
-----

task body T4 is
    C, D : Vector;
    MS, MX : Matrix;
    Z5h : Vector5H;
    a4, a123, a1234, a : Integer;
    A4h : Vector4H;
    Ah, MXMShD : VectorH;
    MXMSh : MatrixH;

begin
    Put_Line("Task T4 started.");

    -- Введення C, MS
    for i in 1..N loop
        for j in 1..N loop
            MS(i)(j) := 1;
        end loop;
        C(i) := 1;
    end loop;

    -- Передати задачі T3 дані C3н, MS3н
    accept C3hMS3htoT3 (C3h_toT3 : out Vector3H; MS3h_toT3 : out
Matrix3H) do
        C3h_toT3 := C(H+1..H*4);
        MS3h_toT3 := MS(H+1..H*4);
    end accept;
end T4;

```

```

end C3hMS3htoT3;

-- Прийняти від задачі T3 дані Z5h, D, MX
T3.Z5hDMXtoT4(Z5h, D, MX);

-- Передати задачі T5 дані C4h, MS4h, Z4h, D, MX
accept C4hMS4hZ4hDMXtoT5 (C4h_toT5 : out Vector4H; MS4h_toT5 : out
Matrix4H; Z4h_toT5 : out Vector4H; D_toT5 : out Vector; MX_toT5 : out
Matrix) do
    C4h_toT5 := C(H+1..H*5);
    MS4h_toT5 := MS(H+1..H*5);
    Z4h_toT5 := Z5h(H+1..H*5);
    D_toT5 := D;
    MX_toT5 := MX;
end C4hMS4hZ4hDMXtoT5;

-- Обчислення a4 = max(Cn)
a4 := C(1);
for i in 1..H loop
    if C(i) > a4 then
        a4 := C(i);
    end if;
end loop;

-- Прийняти від задачі T3 дані a123
T3.a123toT4(a123);

-- Обчислення a1234 = max(a123, a4)
a1234 := a4;
if a1234 > a123 then
    a1234 := a123;
end if;

-- Передати задачі T5 дані a1234
accept a1234toT5 (a1234_toT5 : out Integer) do
    a1234_toT5 := a1234;
end a1234toT5;

-- Прийняти від задачі T5 дані a
T5.atoT4(a);

-- Передати задачі T3 дані a
accept atoT3 (a_toT3 : out Integer) do

```

```

a_toT3 := a;
end atoT3;

-- Обчислення  $A_n = a * Z_n + D * (MX * MSh)$ 
--  $MX * MSh$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MX(k)(j) * MS(i)(k);
        end loop;
    end loop;
end loop;

--  $D * (MX * MSh)$ 
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

--  $A_n = a * Z_n + D * (MX * MSh)$ 
for i in 1..H loop
    Ah(i) := a * Z5h(i) + MXMShD(i);
end loop;

-- Прийняти від задачі T5 результат - вектор  $A_{4n}$ 
T5.A4htoT4(A4h);

-- Передати задачі T3 результат - вектор  $A_{5n}$ 
accept A5htoT3 (A5h_toT3 : out Vector5H) do
    for i in 1..H loop
        A5h_toT3(i) := Ah(i);
    end loop;
    for i in 1..H*4 loop
        A5h_toT3(H+i) := A4h(i);
    end loop;
end A5htoT3;

Put_Line("Task T4 finished");
end T4;
-----

```

```

task body T5 is
    a5, a1234, a12345, a678, a : Integer;
    MX : Matrix;
    D : Vector;
    Z4h, C4h : Vector4H;
    MS4h : Matrix4H;
    A3h : Vector3H;
    Ah, MXMShD : VectorH;
    MXMSh : MatrixH;

begin
    Put_Line("Task T5 started.");

    -- Прийняти від задачі T4 дані C4h, MS4h, Z4h, D, MX
    T4.C4hMS4hZ4hDMXtoT5(C4h, MS4h, Z4h, D, MX);

    -- Передати задачі T6 дані C3h, MS3h, Z3h, D, MX
    accept C3hMS3hZ3hDMXtoT6 (C3h_toT6 : out Vector3H; MS3h_toT6 : out
Matrix3H; Z3h_toT6 : out Vector3H; D_toT6 : out Vector; MX_toT6 : out
Matrix) do
        C3h_toT6 := C4h(H+1..H*4);
        MS3h_toT6 := MS4h(H+1..H*4);
        Z3h_toT6 := Z4h(H+1..H*4);
        D_toT6 := D;
        MX_toT6 := MX;
    end C3hMS3hZ3hDMXtoT6;

    -- Обчислення a5 = max(Cn)
    a5 := C4h(1);
    for i in 1..H loop
        if C4h(i) > a5 then
            a5 := C4h(i);
        end if;
    end loop;

    -- Прийняти від задачі T4 дані a1234
    T4.a1234toT5(a1234);

    -- Обчислення a12345 = max(a1234, a5)
    a12345 := a5;
    if a12345 > a1234 then
        a12345 := a1234;
    end if;

```

```

-- Прийняти від задачі T6 дані a678
T6.a678toT5(a678);

-- Обчислення a = max(a12345, a678)
a := a12345;
if a > a678 then
    a := a678;
end if;

-- Передати задачі T4 дані a
accept atoT4 (a_toT4 : out Integer) do
    a_toT4 := a;
end atoT4;

-- Передати задачі T6 дані a
accept atoT6 (a_toT6 : out Integer) do
    a_toT6 := a;
end atoT6;

-- Обчислення  $A_H = a * Z_H + D * (MX * MSh)$ 
--  $MX * MSh$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MS4h(i)(k) * MX(k)(j);
        end loop;
    end loop;
end loop;

--  $D * (MX * MSh)$ 
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

--  $A_H = a * Z_H + D * (MX * MSh)$ 
for i in 1..H loop
    Ah(i) := a * Z4h(i) + MXMShD(i);
end loop;

```

```

-- Прийняти від задачі T6 результат - вектор A3н
T6.A3htoT5 (A3h);

-- Передати задачі T4 результат - вектор A4н
accept A4htoT4 (A4h_toT4 : out Vector4H) do
  for i in 1..H loop
    A4h_toT4(i) := Ah(i);
  end loop;
  for i in 1..H*3 loop
    A4h_toT4(H+i) := A3h(i);
  end loop;
end A4htoT4;

Put_Line("Task T5 finished");
end T5;

-----

task body T6 is
  a6, a78, a678, a : Integer;
  MX : Matrix;
  D : Vector;
  Z3h, C3h : Vector3H;
  MS3h : Matrix3H;
  A2h : Vector2H;
  Ah, MXMShD : VectorH;
  MXMSh : MatrixH;

begin
  Put_Line("Task T6 started.");

  -- Прийняти від задачі T5 дані C3н, MS3н, Z3н, D, MX
  T5.C3hMS3hZ3hDMXtoT6(C3h, MS3h, Z3h, D, MX);

  -- Передати задачі T7 дані C2н, MS2н, Z2н, D, MX
  accept C2hMS2hZ2hDMXtoT7 (C2h_toT7 : out Vector2H; MS2h_toT7 : out
Matrix2H; Z2h_toT7 : out Vector2H; D_toT7 : out Vector; MX_toT7 : out
Matrix) do
    C2h_toT7 := C3h(H+1..H*3);
    MS2h_toT7 := MS3h(H+1..H*3);
    Z2h_toT7 := Z3h(H+1..H*3);
    D_toT7 := D;
    MX_toT7 := MX;
  end C2hMS2hZ2hDMXtoT7;

```

```

-- Обчислення a6 = max(Cн)
a6 := C3h(1);
for i in 1..H loop
    if C3h(i) > a6 then
        a6 := C3h(i);
    end if;
end loop;

-- Прийняти від задачі T7 дані a78
T7.a78toT6(a78);

-- Обчислення a678 = max(a78, a6)
a678 := a6;
if a78 > a678 then
    a678 := a78;
end if;

-- Передати задачі T5 дані a678
accept a678toT5 (a678_toT5 : out Integer) do
    a678_toT5 := a678;
end a678toT5;

-- Прийняти від задачі T5 дані a
T5.atoT6(a);

-- Передати задачі T7 дані a
accept atoT7 (a_toT7 : out Integer) do
    a_toT7 := a;
end atoT7;

-- Обчислення  $A_n = a * Z_n + D * (MX * MSh)$ 
--  $MX * MSh$ 
for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MS3h(i)(k) * MX(k)(j);
        end loop;
    end loop;
end loop;

--  $D * (MX * MSh)$ 

```

```

for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

--  $A_H = a * Z_H + D * (MX * MSh)$ 
for i in 1..H loop
    Ah(i) := a * Z3h(i) + MXMShD(i);
end loop;

-- Прийняти від задачі T7 результат - вектор A2h
T7.A2htoT6(A2h);

-- Передати задачі T5 результат - вектор A3h
accept A3htoT5 (A3h_toT5 : out Vector3H) do
    for i in 1..H loop
        A3h_toT5(i) := Ah(i);
    end loop;
    for i in 1..H*2 loop
        A3h_toT5(H+i) := A2h(i);
    end loop;
end A3htoT5;

Put_Line("Task T6 finished");
end T6;

-----

task body T7 is
    a7, a78, a8, a : Integer;
    MX : Matrix;
    D : Vector;
    Z2h, C2h : Vector2H;
    MS2h : Matrix2H;
    MXMSh : MatrixH;
    Ah, Ah_T8, MXMShD : VectorH;

begin
    Put_Line("Task T7 started.");

    -- Прийняти від задачі T6 дані C2h, MS2h, Z2h, D, MX
    T6.C2hMS2hZ2hDMXtoT7(C2h, MS2h, Z2h, D, MX);

```



```

-- Передати задачі T8 дані Cн, MSh, Zh, D, MX
accept ChMShZhDMXtoT8 (Ch_toT8 : out VectorH; MSh_toT8 : out
MatrixH; Zh_toT8 : out VectorH; D_toT8 : out Vector; MX_toT8 : out
Matrix) do
    Ch_toT8 := C2h(H+1..H*2);
    MSh_toT8 := MS2h(H+1..H*2);
    Zh_toT8 := Z2h(H+1..H*2);
    D_toT8 := D;
    MX_toT8 := MX;
end ChMShZhDMXtoT8;

-- Обчислення a7 = max(Cн)
a7 := C2h(1);
for i in 1..H loop
    if C2h(i) > a7 then
        a7 := C2h(i);
    end if;
end loop;

-- Прийняти від задачі T8 дані a8
T8.a8toT7(a8);

-- Обчислення a78 = max(a8, a7)
a78 := a7;
if a8 > a78 then
    a78 := a8;
end if;

-- Передати задачі T6 дані a78
accept a78toT6 (a78_toT6 : out Integer) do
    a78_toT6 := a78;
end a78toT6;

-- Прийняти від задачі T6 дані a
T6.atoT7(a);

-- Передати задачі T8 дані a
accept atoT8 (a_toT8 : out Integer) do
    a_toT8 := a;
end atoT8;

-- Обчислення  $A_n = a * Z_n + D * (MX * MSh)$ 
--  $MX * MSh$ 

```

```

for i in 1..H loop
    for j in 1..N loop
        MXMSh(i)(j) := 0;
        for k in 1..N loop
            MXMSh(i)(j) := MXMSh(i)(j) + MS2h(H+i)(k) * MX(k)(j);
        end loop;
    end loop;
end loop;

-- D * (MX * MSh)
for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
        MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
end loop;

-- AH = a * ZH + D * (MX * MSh)
for i in 1..H loop
    Ah(i) := a * Z2h(i) + MXMShD(i);
end loop;

-- Прийняти від задачі T8 результат - вектор AH
T8.AhtoT7(Ah_T8);

-- Передати задачі T6 результат - вектор A2H
accept A2htoT6 (A2h_toT6 : out Vector2H) do
    for i in 1..H loop
        A2h_toT6(i) := Ah(i);
    end loop;
    for i in 1..H loop
        A2h_toT6(H+i) := Ah_T8(i);
    end loop;
end A2htoT6;

Put_Line("Task T7 finished");
end T7;

-----

task body T8 is
    a8, a : Integer;
    MX : Matrix;
    D : Vector;

```

```

Zh, Ch : VectorH;
MXMSh, MSh : MatrixH;
Ah, MXMShD : VectorH;

begin
  Put_Line("Task T8 started.");

  -- Прийняти від задачі T7 дані Cн, MSн, Zн, D, MX
  T7.ChMShZhDMXtoT8(Ch, MSh, Zh, D, MX);

  -- Обчислення a8 = max(Cн)
  a8 := Ch(1);
  for i in 1..H loop
    if Ch(i) > a8 then
      a8 := Ch(i);
    end if;
  end loop;

  -- Передати задачі T7 дані a8
  accept a8toT7 (a8_toT7 : out Integer) do
    a8_toT7 := a8; -- Pass the correct value of a8
  end a8toT7;

  -- Прийняти від задачі T7 дані a
  T7.atot8(a);

  -- Обчислення An = a * Zн + D * (MX * MSн)
  -- MX * MSн
  for i in 1..H loop
    for j in 1..N loop
      MXMSh(i)(j) := 0;
      for k in 1..N loop
        MXMSh(i)(j) := MXMSh(i)(j) + MSh(i)(k) * MX(k)(j);
      end loop;
    end loop;
  end loop;

  -- D * (MX * MSн)
  for i in 1..H loop
    MXMShD(i) := 0;
    for j in 1..N loop
      MXMShD(i) := MXMShD(i) + D(i) * MXMSh(i)(j);
    end loop;
  end loop;
end

```


Дослідження завантаженості ядер процесору:

Варто зазначити, що конкретно мій комп'ютер містить 8 логічних ядер та 4 фізичних. Нижченаведений рис.1.2 демонструє, що при використанні 8 ядер процесору розподілення обчислювальних можливостей відбувається рівномірно між кожним з них. Поєднуючи цю інформацію із раніше отриманими остаточними значеннями обрахунків, можна зробити висновок, що восьми потокова програма написана коректно.

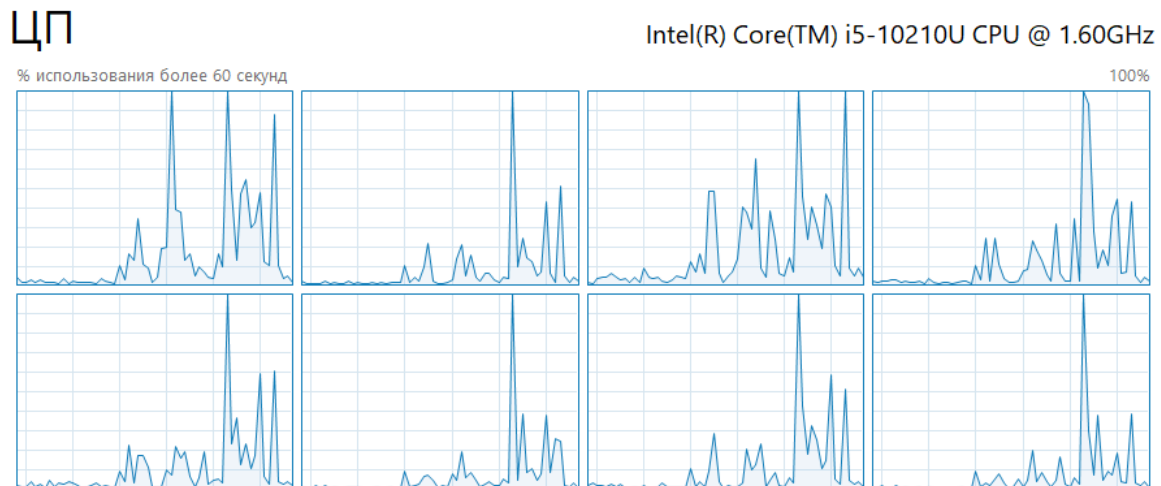


рис.1.2 Завантаженість системи при використанні 8 ядер

Додатково для порівняння було досліджено завантаженість системи при використанні 1 ядра (рис. 1.3).

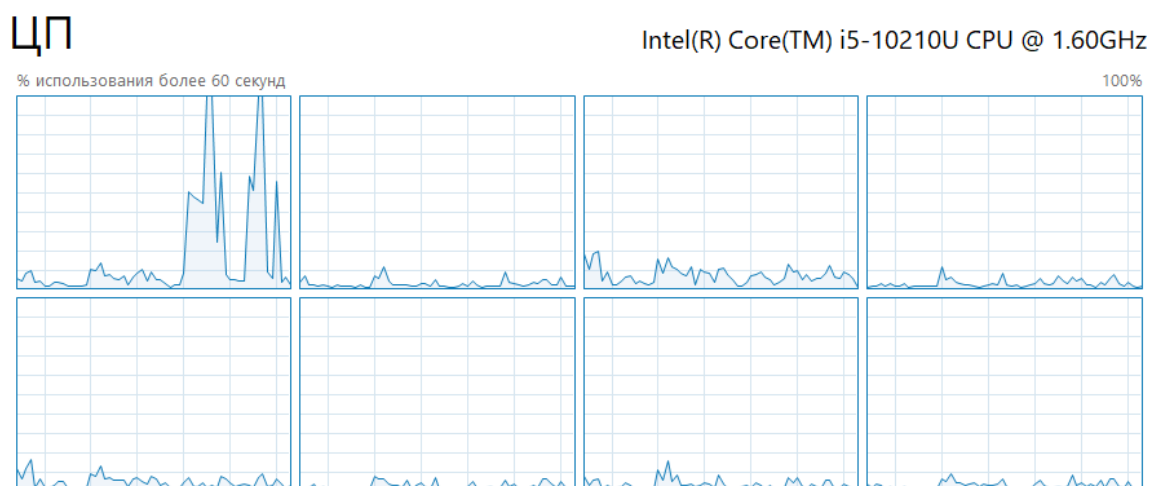


рис.1.3 Завантаженість системи при використанні 1 ядра

Також було обраховано коефіцієнт прискорення при запуску програми на 8 ядрах та на 1 ядрі при $N = 2400$: $K_p = 17.6 \text{ s} / 8.7 \text{ s} = 2,02$. Тобто при використанні 8 ядер замість 1, програма завершується у 2,02 рази швидше.

Висновок:

1. Було розроблено паралельний математичний алгоритм, який дозволяє отримати розв'язок для математичного завдання за варіантом. За отриманим алгоритмом було визначено які ресурси у ПКС ЛП потрібно передавати повністю, а які частково.
2. Були розроблені алгоритми для кожного з потоків T1-T8, що включають обчислення, ввід та вивід даних, а також взаємодію між задачами через механізм передачі даних - рандеву. Отримані алгоритми дозволили зрозуміти та пізніше практично реалізувати коректну передачу ресурсів потрібних розмірів.
3. Було розроблено структурну схему взаємодії задач, завдяки якій було визначено та наочно продемонстровано потрібні засоби організації взаємодії потоків.
4. Виконано практичну реалізацію програми мовою програмування Ada, де було застосовано визначені у попередніх етапах алгоритми коректної взаємодії потоків. Отримані наприкінці правильні результати обчислень підтверджують коректність усіх розроблених етапів організації взаємодії у восьми поточній системі.
5. Додатково було проведено дослідження завантаженості ядер, що підтвердило коректність розподілення обчислювальних ресурсів. Також було обчислено коефіцієнт прискорення при запуску програми на 8 ядрах та на 1 ядрі при $N = 2400$: $K_p = 2,02$.