

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-34
Мозговий Іван

Перевірів:
Корочкін О. В.

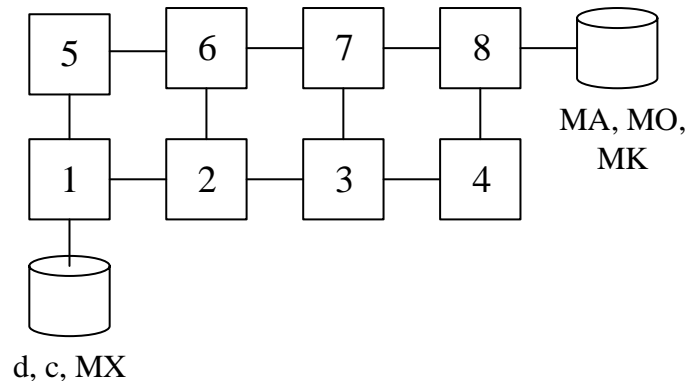
Київ – 2016 р.

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

Розробити програму для розв'язання в ПКС з ЛП математичної задачі: $MA = d * MO + c * (MX * MK)$

Мова програмування: Ада

Засоби організації взаємодії: рандеву



Структурна схема ПКС

Виконання роботи:

Етап 1. Побудова паралельного алгоритму

1. $MA_H = dMO_H + c(MX_H * MK)$

Етап 2. Розроблення алгоритмів роботи кожного процесу

Задача T1

1. Введення c, d, MX
2. Передати c, d, MX задачі T2, T5
3. Прийняти MO, MK від задачі T2
4. Обчислення $MA_H = dMO_H + c(MX_H * MK)$
5. Передати MA_H задачі T5

Задача T2

1. Прийняти c, d, MX від задачі T1
2. Передати c, d, MX задачі T3
3. Прийняти MO, MK від задачі T3
4. Передати MO, MK задачі T1
5. Обчислення $MA_H = dMO_H + c(MX_H * MK)$
6. Передати MA_H задачі T6

Задача T3

1. Прийняти c, d, MX від задачі T2
2. Передати c, d, MX задачі T4
3. Прийняти MO, MK від задачі T4
4. Передати MO, MK задачі T2
5. Обчислення $MA_H = dMO_H + c(MX_H * MK)$
6. Передати MA_H задачі T7

Задача Т4

1. **Прийняти** c, d, MX від задачі Т3
2. **Прийняти** MA, MO, MK від задачі Т8
3. **Передати** MA, MO, MK задачі Т3
4. **Обчислення** $MA_H = dMO_H + c(MX_H * MK)$
5. **Передати** MA_H задачі Т8

Задача Т5

1. **Прийняти** c, d, MX від задачі Т1
2. **Передати** c, d, MX задачі Т6
3. **Прийняти** MO, MK від задачі Т6
4. **Обчислення** $MA_H = dMO_H + c(MX_H * MK)$
5. **Прийняти** MA_H від задачі Т1
6. **Передати** MA_{2H} задачі Т6

Задача Т6

1. **Прийняти** c, d, MX від задачі Т5
2. **Передати** c, d, MX задачі Т7
3. **Прийняти** MO, MK від задачі Т7
4. **Передати** MO, MK задачі Т5
5. **Обчислення** $MA_H = dMO_H + c(MX_H * MK)$
6. **Прийняти** MA_H від задачі Т2
7. **Прийняти** MA_{2H} від задачі Т5
8. **Передати** MA_{4H} задачі Т7

Задача Т7

1. **Прийняти** c, d, MX від задачі Т6
2. **Передати** c, d, MX задачі Т8
3. **Прийняти** MO, MK від задачі Т8
4. **Передати** MO, MK задачі Т6
5. **Обчислення** $MA_H = dMO_H + c(MX_H * MK)$
6. **Прийняти** MA_H від задачі Т3
7. **Прийняти** MA_{4H} від задачі Т6
8. **Передати** MA_{6H} задачі Т8

Задача Т8

1. **Введення** MO, MK
2. **Передати** MO, MK задачі Т4, Т7
3. **Прийняти** c, d, MX від задачі Т7
4. **Передати** MO, MK задачі Т6
5. **Обчислення** $MA_H = dMO_H + c(MX_H * MK)$
6. **Прийняти** MA_H від задачі Т4
7. **Прийняти** MA_{6H} від задачі Т7
8. **Виведення** MA

Етап 4. Розроблення програми

Lab7.adb

```
-----
--                                     --
--               Parallel and Distributed Computing               --
--               Lab7. Ada. Rendezvous                           --
--                                     --
-- File: lab7.adb                                               --
-- Task:  $MA = d * MO + c * (MX * MK)$                                --
--                                     --
-- Author: Mozghoviy Ivan, group IO-34                         --
-- Date: 07.06.2016                                           --
--                                     --
-----

with Ada.Text_IO, Ada.Integer_text_io, Ada.Synchronous_Task_Control,
     Data, Ada.Real_Time;
use Ada.Text_IO, Ada.Integer_text_io, Ada.Synchronous_Task_Control,
     Ada.Real_Time;

procedure Lab7 is

    N : Natural := 8;
    P : Natural := 8;
    H : Natural := N/P;

    package Data_N is new Data(N);
    use Data_N;

    procedure Start_Tasks is
        -----
        --
        -----

        --                                     Task T1                                     --
        task T1 is
            entry dataT2(
                MO: in Matrix;
                MK: in Matrix);
        end T1;

        --                                     Task T2                                     --
        task T2 is
            entry DataT1(
                Alpha: in Integer;
                c : in Integer;
                MX : in Matrix);
            entry DataT3(
                MO: in Matrix;
                MK: in Matrix);
        end T2;

        --                                     Task T3                                     --
        task T3 is
            entry DataT2(
                Alpha: in Integer;
                c : in Integer;
                MX : in Matrix);
            entry DataT4(
                MO: in Matrix;
                MK: in Matrix);
        end T3;

        --                                     Task T4                                     --
        task T4 is
            entry DataT3(
                Alpha: in Integer;
                c : in Integer;
                MX : in Matrix);
```

```

        entry DataT8(
            MO: in Matrix;
            MK: in Matrix);
    end T4;

--                                     Task T5                                     --
    task T5 is
        entry DataT1(
            Alpha: in Integer;
            c : in Integer;
            MX : in Matrix);
            entry DataT6(
                MO: in Matrix;
                MK: in Matrix);
            entry ResultT1(
                MA: in Matrix);
    end T5;

--                                     Task T6                                     --
    task T6 is
        entry DataT5(
            Alpha: in Integer;
            c : in Integer;
            MX : in Matrix);
            entry DataT7(
                MO: in Matrix;
                MK: in Matrix);
            entry ResultT2(MA: in Matrix);
            entry ResultT5(MA: in Matrix);
    end T6;

--                                     Task T7                                     --
    task T7 is
        entry DataT6(
            Alpha: in Integer;
            c : in Integer;
            MX : in Matrix);
            entry DataT8(
                MO: in Matrix;
                MK: in Matrix);

            entry ResultT3(
                MA: in Matrix);
            entry ResultT6(
                MA: in Matrix);
    end T7;

--                                     Task T8                                     --
    task T8 is
        entry DataT7(
            Alpha: in Integer;
            c : in Integer;
            MX : in Matrix);

            entry ResultT4(
                MA: in Matrix);
            entry ResultT7(
                MA: in Matrix);
    end T8;

-----
--                                     --
-----

--                                     Task T1                                     --
    task body T1 is
        Alpha1, C1 : Integer;
        MK1 : Matrix_N;
        MX1 : Matrix_N;

```

```

        M01: Matrix(1..H);
        MA1: Matrix(1..H);
begin
    Put_Line("T1 started");

    -- 1. Input data
        Alpha1 := 1;
        C1:=1;
    Input(MX1, 1);

    T5.DataT1(Alpha1, C1, MX1(4*H+1..8*H));
    T2.DataT1(Alpha1, C1, MX1(H+1..4*H));
    accept DataT2(M0: in Matrix; MK: in Matrix)do
        M01 := M0;
        MK1:=MK;
    end DataT2;
    Calculation(Alpha1, M01, c1, MX1, MK1, 1, H, 0, MA1);
    T5.ResultT1(MA1);
    Put_Line("T1 finished");
end T1;

--                                     Задача T2                                     --
task body T2 is
    Alpha2, C2 : Integer;
    MK2: Matrix_N;
    MX2 : Matrix(1..3*H);
    M02: Matrix(1..2*H);
    M02_H: Matrix(1..H);
    MA2: Matrix(1..H);
begin
    Put_Line("T2 started");

    accept DataT1(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha2:= Alpha;
        C2:=C;
        MX2:=MX;
    end DataT1;
    T3.DataT2(Alpha2, C2, MX2(H+1..3*H));
    accept DataT3(M0: in Matrix; MK: in Matrix) do
        MK2:=MK;
        M02:=M0;
    end DataT3;

    for i in 1..H loop
        for j in 1..N loop
            M02_H(i)(j):=M02(i+H)(j);
        end loop;
    end loop;

    T1.DataT2(M02(1..H), MK2);

    Calculation(Alpha2, M02_H, c2, MX2, MK2, 1, H, 0, MA2);

    T6.ResultT2(MA2);
    Put_Line("T2 finished");
end T2;

--                                     Задача T3                                     --
task body T3 is
    Alpha3, C3 : Integer;
    MK3: Matrix_N;
    MX3 : Matrix(1..2*H);
    M03: Matrix(1..3*H);
    M03_H: Matrix(1..H);
    MA3: Matrix(1..H);
begin
    Put_Line("T3 started");

    accept DataT2(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha3:= Alpha;

```

```

        C3:=C;
        MX3:=MX;
    end DataT2;
    T4.DataT3(Alpha3, C3, MX3(H+1..2*H));
    accept DataT4(M0: in Matrix; MK: in Matrix) do
        MK3:=MK;
        M03:=M0;
    end DataT4;
    T2.DataT3(M03(1..2*H), MK3);
    for i in 1..H loop
        for j in 1..N loop
            M03_H(i)(j):=M03(i+2*H)(j);
        end loop;
    end loop;

Calculation(Alpha3, M03_H, c3, MX3, MK3, 1, H, 0, MA3);

    T7.ResultT3(MA3);
    Put_Line("T3 finished");
end T3;

--                                     Задача T4                                     --
task body T4 is
    Alpha4, C4 : Integer;
    MK4: Matrix_N;
    MX4 : Matrix(1..H);
    M04: Matrix(1..4*H);
    M04_H: Matrix(1..H);
    MA4: Matrix(1..H);
begin
    Put_Line("T4 started");

    accept DataT3(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha4:= Alpha;
        C4:=C;
        MX4:=MX;
    end DataT3;
    accept DataT8(M0: in Matrix; MK: in Matrix) do
        MK4:=MK;
        M04:=M0;
    end DataT8;
    T3.DataT4(M04(1..3*H), MK4);
    for i in 1..H loop
        for j in 1..N loop
            M04_H(i)(j):=M04(i+3*H)(j);
        end loop;
    end loop;

    Calculation(Alpha4, M04_H, c4, MX4, MK4, 1, H, 0, MA4);

    T8.ResultT4(MA4);
    Put_Line("T4 finished");
end T4;

--                                     Задача T5                                     --
task body T5 is
    Alpha5, C5 : Integer;
    MK5: Matrix_N;
    MX5: Matrix(1..4*H);
    M05: Matrix(1..H);
    MA5: Matrix(1..2*H);
begin
    Put_Line("T5 started");
    accept DataT1(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha5:= Alpha;
        C5:=C;
        MX5:=MX;
    end DataT1;

```

```

        T6.DataT5(Alpha5, C5, MX5(H+1..4*H));
        accept DataT6(M0: in Matrix; MK: in Matrix) do
            MK5:=MK;
            M05:=M0;
        end DataT6;
        Calculation(Alpha5, M05, c5, MX5, MK5, 1, H, H, MA5);
        accept ResultT1(MA: in Matrix) do
            MA5(1..H):=MA;
        end ResultT1;

        T6.ResultT5(MA5);

    Put_Line("T5 finished");

end T5;

--                                     Задача T6                                     --
task body T6 is
    Alpha6, C6 : Integer;
    MK6: Matrix_N;
    MX6: Matrix(1..3*H);
    M06: Matrix(1..2*H);
    M06_H: Matrix(1..H);
    MA6: Matrix(1..4*H);
begin
    Put_Line("T6 started");

    accept DataT5(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha6:= Alpha;
        C6:=C;
        MX6:=MX;
    end DataT5;
    T7.DataT6(Alpha6, C6, MX6(H+1..3*H));
    accept DataT7(M0: in Matrix; MK: in Matrix) do
        MK6:=MK;
        M06:=M0;
    end DataT7;
    T5.DataT6(M06(1..H), MK6);
    for i in 1..H loop
        for j in 1..N loop
            M06_H(i)(j):=M06(i+H)(j);
        end loop;
    end loop;

    Calculation(Alpha6, M06_H, c6, MX6, MK6, 1, H, 3*H, MA6);
    accept ResultT2(MA: in Matrix)do
        MA6(H+1..2*H):=MA;
    end ResultT2;

    accept ResultT5(MA: in Matrix)do
        MA6(1..H):=MA(1..H);
        MA6(2*H+1..3*h):=MA(H+1..2*H);
    end ResultT5;

    T7.ResultT6(MA6);
    Put_Line("T6 finished");
end T6;

task body T7 is
    Alpha7, C7 : Integer;
    MK7: Matrix_N;
    MX7: Matrix(1..2*H);
    M07: Matrix(1..3*H);
    M07_H: Matrix(1..H);
    MA7: Matrix(1..6*H);
    MA7_H: Matrix(1..H);
begin
    Put_Line("T7 started");

    accept DataT6(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha7:= Alpha;

```



```

        C7:=C;
        MX7:=MX;
    end DataT6;
    T8.DataT7(Alpha7, C7, MX7(H+1..2*H));
    accept DataT8(M0: in Matrix; MK: in Matrix) do
        MK7:=MK;
        M07:=M0;
    end DataT8;
    T6.DataT7(M07(1..2*H), MK7);
    for i in 1..H loop
        for j in 1..N loop
            M07_H(i)(j):=M07(i+2*H)(j);
        end loop;
    end loop;
    Calculation(Alpha7, M07_H, c7, MX7, MK7, 1, H, 5*H, MA7);

    accept ResultT3(MA: in Matrix) do
        MA7(2*H+1..3*H):=MA;
    end ResultT3;

    accept ResultT6(MA: in Matrix) do
        MA7(1..2*H):=MA(1..2*H);
        MA7(3*H+1..5*H):=MA(2*H+1..4*H);
    end ResultT6;

    T8.ResultT7(MA7);
    Put_Line("T7 finished");
end T7;

task body T8 is
    Alpha8, C8 : Integer;
    MK8 : Matrix_N;
    MX8: Matrix(1..H);
    M08: Matrix_N;
    M08_H: Matrix(1..H);
    MA8: Matrix_N;
    MA8_H: Matrix(1..H);
begin
    Put_Line("T8 started");

    Input(M08, 1);
    M08(4)(4):=10;
    Input(MK8, 1);
    accept DataT7(Alpha: in Integer; C: in Integer; MX: in Matrix) do
        Alpha8:= Alpha;
        C8:=C;
        MX8:=MX;
    end DataT7;
    T7.DataT8(M08(4*H+1..7*H), MK8);
    T4.DataT8(M08(1..4*H), MK8);
    for i in 1..H loop
        for j in 1..N loop
            M08_H(i)(j):=M08(i+7*H)(j);
        end loop;
    end loop;
    Calculation(Alpha8, M08_H, c8, MX8, MK8, 1, H, 7*h, MA8);

    accept ResultT4(MA: in Matrix) do
        MA8(3*H+1..4*H):=MA;
    end ResultT4;

    accept ResultT7(MA: in Matrix) do
        MA8(1..3*H):=MA(1..3*H);
        MA8(4*H+1..7*H):=MA(3*H+1..6*H);
    end ResultT7;

    Output(MA8);

```

```

        Put_Line("T8 finished");
    end T8;

begin
    null;
end Start_Tasks;

begin
    Put_Line ("Lab7 started");
    Start_Tasks;
    Put_Line ("Lab7 finished");
end Lab7;

```

Data.adb

```

with Ada.Text_IO, Ada.Integer_Text_IO;
use Ada.Text_IO, Ada.Integer_Text_IO;

package body Data is

    procedure Input (V : out Vector; Value : in Integer) is
    begin
        for I in V'Range loop
            V(I) := Value;
        end loop;
    end Input;

    procedure Input (MA : out Matrix; Value : in Integer) is
    begin
        for I in MA'Range loop
            for J in 1..N loop
                MA(I)(J) := Value;
            end loop;
        end loop;
    end Input;

    procedure Output (V : in Vector) is
    begin
        New_Line;
        for I in V'Range loop
            Put(Item => V(I));
        end loop;
        New_Line;
    end Output;

    procedure Output (MA : in Matrix) is
    begin
        New_Line;
        for I in MA'Range loop
            for J in 1..N loop
                Put(Item => MA(i)(j), Width => 7);
            end loop;
            New_line;
        end loop;
        New_Line;
    end Output;

    function Maximum_Of_Matrix (MO : in Matrix) return Integer is
        Result : Integer := Integer'First;
        X      : Integer;
    begin
        for I in MO'Range loop
            for J in 1..N loop
                X := MO(I)(J);
            end loop;
        end loop;
    end Maximum_Of_Matrix;

```

```

        if X > Result then
            Result := X;
        end if;
    end loop;
end loop;
return Result;
end Maximum_Of_Matrix;

-- AH = m*BH + alpha*C*(MOH*MK)
--Ah = dB + C(MO*MK)
procedure Calculation (d : in Integer; MO : in Matrix; c: in integer; MX : in Matrix;
MK : in Matrix_N; First : in Integer; Last : in Integer; Start: in integer; MA : out Matrix) is

    Sum1 : Integer;
begin
    for I in First..Last loop
        for J in 1..N loop
            Sum1 := 0;
            for K in 1..N loop
                Sum1 := Sum1 + MX(I)(K)*MK(J)(K);
            end loop;
            MA(I+Start)(J) := d*MO(I)(J) + C * Sum1;
        end loop;
    end loop;
end Calculation;

procedure Transpose (MO : in out Matrix) is
    MT : Matrix := MO;
begin
    for I in 1..N loop
        for J in 1..N loop
            MO(J)(I) := MT(I)(J);
        end loop;
    end loop;
end Transpose;

end Data;
```