

Міністерство освіти і науки України

Національний технічний університет України “Київський політехнічний інститут”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота № 1

з курсу “ Паралельне програмування-2. Програмування для паралельних систем”

на тему:

“ Ada. Семафори”

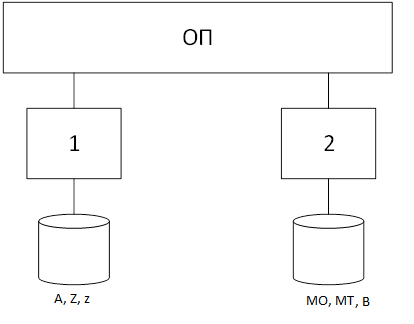
Виконав: студент ІІI курсу групи ІП – 42

Водотієць Денис Ігорович

Київ – 2017

**Технічне завдання**

Структура ПКС:



Математична задача: A = max(Z) \* B + z \* (MO \* MT)

**Виконання**

Етап 1. Розробка паралельного математичного алгоритму

1. ai = max(ZH);
2. a = max(a, ai), де i = 1..2;
3. AH = a \* BH + z \* (MO \* MTH) ,

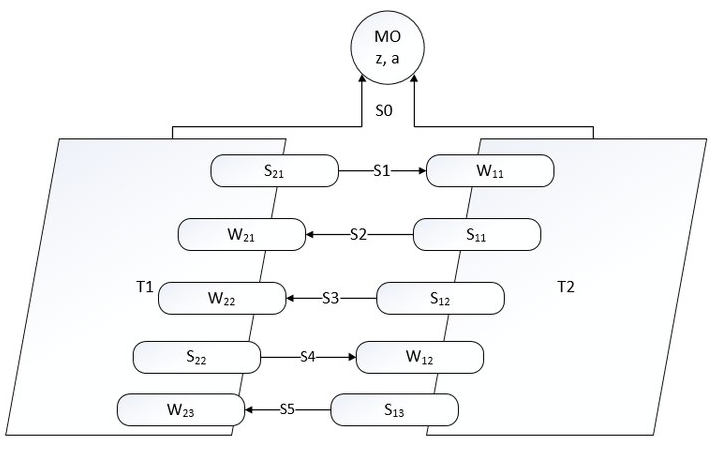
де H = N / P, де P – кількість процесів.

ЗР: МO, z, a.

Етап 2. Розробка алгоритму потоків

|  |  |  |  |
| --- | --- | --- | --- |
| Т1 |  | Т2 |  |
| 1. Введення Z, z |  | 1. Введення B, MO, MT |  |
| 2. Сигнал Т2 про завершення введення | S21 | 2. Сигнал Т1 про завершення введення | S11 |
| 3. Очікувати введення Т2 | W21 | 3. Очікувати введення Т1 | W11 |
| 4. Копіювати z1 := z, MO1 := MO | КУ | 4. Копіювати z2 := z, MO2 := MO | КУ |
| 5. Рахувати: a1 = max(ZH) |  | 5. Рахувати: a2 = max(ZH) |  |
| 6. Очікувати сигнал про завершення обрахування Т2 | W22 | 6. Сигнал Т1 про завершення обрахування | S12 |
| 7. Рахувати: a = max(a1, a2)­ |  | 7. Очікувати знаходження максимуму | W12 |
| 8. Сигнал Т2 про завершення знаходження максимуму | S22 | 8. Копіювати: a2 := a | КУ |
| 9. Копіювати: a1 := a | КУ | 9. Рахувати: AH = a2 \* BH + z \* (MO2 \* MTH) |  |
| 9. Рахувати: AH = a1 \* BH + z \* (MO1 \* MTH) |  | 10. Сигнал Т1 про завершення рахунку | S13 |
| 10. Очікувати сигнал про завершення рахунку Т2 | W23 |  |  |
| 11. Вивід А |  |  |  |

Етап 3. Розробка схеми взаємодії потоків



Етап 4. Розробка програми

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

----- Lab 1. Ada. Semaphore -----

----- A = max(Z) \* B + z \* (MO \* MT) -----

----- Denys Vodotiiets -----

----- IP-42 -----

----- 22.03.17 -----

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

with Ada.Text\_IO; use Ada.Text\_IO;

with Ada.Integer\_Text\_IO; use Ada.Integer\_Text\_IO;

with Ada.Synchronous\_Task\_Control; use Ada.Synchronous\_Task\_Control;

procedure Main is

N: Integer := 2;

P: Integer := 2;

H: Integer := N/P;

s0, s1, s2, s3, s4, s5: Suspension\_Object;

type Vector is array(1..N) of Integer;

type Matrix is array(1..N) of Vector;

z, a : Integer;

a1 : Integer := Integer'First;

a2 : Integer := Integer'First;

Result, B, Z\_Vector: Vector;

MO, MT: Matrix;

procedure inputVector (v : out vector) is

begin

for i in 1..N loop

v(i) := 1;

end loop;

end inputVector;

procedure outputVector (v : in vector) is

begin

if N > 6 then

Put\_Line("Vector is too large");

else

for i in 1..N loop

Put(v(i));

Put(" ");

end loop;

Put\_Line("");

end if;

end outputVector;

procedure inputMatrix (m : out Matrix) is

begin

for i in 1..N loop

for j in 1..N loop

m(i)(j) := 1;

end loop;

end loop;

end inputMatrix;

procedure run is

task T1;

task body T1 is

MO1: Matrix;

z1: Integer;

sum: Integer;

begin

Put\_Line(">>T1 started");

-- entering data

inputVector(Z\_Vector);

z := 1;

-- send signal to T2 about end of input

Set\_True(s1);

-- wait until T2 ends entering

Suspend\_Until\_True(s2);

-- copy mutual resources

Suspend\_Until\_True(s0);

z1 := z;

for i in 1..N loop

for j in 1..N loop

MO1(i)(j) := MO(i)(j);

end loop;

end loop;

Set\_True(s0);

-- calculating

for i in 1..H loop

if Z\_Vector(i) > a1 then

a1 := Z\_Vector(i);

end if;

end loop;

-- wait while T2 is calculating

Suspend\_Until\_True(s3);

if a1 > a2 then

a := a1;

else

a := a2;

end if;

-- send signal to T2 about end of calculating

Set\_True(s4);

-- copy mutual resources

Suspend\_Until\_True(s0);

a1 := a;

Set\_True(s0);

-- calculating

for j in 1..H loop

for i in 1..N loop

sum := 0;

for k in 1..N loop

sum := sum + MO1(i)(k) \* MT(j)(k);

end loop;

Result(j) := sum \* z1;

end loop;

Result(j) := Result(j) + a1 \* B(j);

end loop;

-- wait while T2 is calculating

Suspend\_Until\_True(s5);

-- output result

Put\_Line("Result A is ");

outputVector(Result);

Put\_Line(">>T1 finished");

end T1;

task T2;

task body T2 is

MO2: Matrix;

z2: Integer;

sum: Integer;

begin

Put\_Line(">>T2 started");

-- entering data

inputVector(B);

inputMatrix(MO);

inputMatrix(MT);

-- send signal to T1 about end of input

Set\_True(s2);

-- wait until T1 ends entering

Suspend\_Until\_True(s1);

-- copy mutual resources

Suspend\_Until\_True(s0);

z2 := z;

for i in 1..N loop

for j in 1..N loop

MO2(i)(j) := MO(i)(j);

end loop;

end loop;

Set\_True(s0);

-- calculating

for i in H + 1..N loop

if Z\_Vector(i) > a2 then

a2 := Z\_Vector(i);

end if;

end loop;

-- send signal to T1 about end of calculating

Set\_True(s3);

-- wait while T1 is calculating

Suspend\_Until\_True(s4);

-- copy mutual resources

Suspend\_Until\_True(s0);

a1 := a;

Set\_True(s1);

-- calculating

for j in H+1..N loop

for i in 1..N loop

sum := 0;

for k in 1..N loop

sum := sum + MO2(i)(k) \* MT(j)(k);

end loop;

Result(j) := sum \* z2;

end loop;

Result(j) := Result(j) + a2 \* B(j);

end loop;

-- send signal to T1 about end of calculating

Set\_True(s5);

Put\_Line(">>T2 finished");

end T2;

begin

null;

end run;

begin

Set\_True(s0);

run;

end Main;