

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

Кафтанатій Богдан Сергійович

Київ – 2017

**Технічне завдання**

Структура ПКС:



Математична задача: MA = MB \* MC + e \* MZ

**Виконання**

Етап 1. Розробка паралельного математичного алгоритму

MAh = MB \* MCh + e \* MZh

h = N / P, де P – кількість процесів.

ОР: МВ, е.

Етап 2. Розробка алгоритму потоків

|  |  |  |  |
| --- | --- | --- | --- |
| Т1 |  | Т2 |  |
| 1. Введення МВ, MZ |  | 1. Введення МС, е |  |
| 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 е1 := е, МВ1 := МВ | КУ | 4. Копіювати е2 := е, МВ2 := МВ | КУ |
| 5. Рахувати: МАh = MB1 \* MCh + e1 \* MZh |  | 5. Рахувати: МАh = MB2 \* MCh + e2 \* MZh |  |
| 6. Очікувати сигнал про завершення рахунку Т2 | W22 | 6. Сигнал Т1 про завершення рахунку | S12 |
| 7. Вивід МА |  |  |  |

Етап 3. Розробка схеми взаємодії потоків



Етап 4. Розробка програми

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

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

----- MA = MB \* MC + e \* MZ -----

----- Kaftanatiy Bogdan -----

----- 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 := 4;

P: Integer := 2;

H: Integer := N/P;

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

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

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

e: Integer;

MA, MB, MC, MZ: Matrix;

procedure inputMatrix (MA : out Matrix) is

begin

for i in 1..N loop

for j in 1..N loop

MA(i)(j):=1;

end loop;

end loop;

end inputMatrix;

procedure outputMatrix (MA : in Matrix) is

begin

if N > 6 then

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

else

for i in 1..N loop

for j in 1..N loop

Put(MA(i)(j));

Put(" ");

end loop;

Put\_Line("");

end loop;

end if;

end outputMatrix;

procedure startTasks is

task T1;

task body T1 is

MB1: Matrix;

e1: Integer;

tempSum: Integer;

begin

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

-- entering data

inputMatrix(MB);

inputMatrix(MZ);

-- 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);

e1 := e;

for i in 1..N loop

for j in 1..N loop

MB1(i)(j) := MB(i)(j);

end loop;

end loop;

Set\_True(s0);

-- calculating

for j in 1..H loop

for i in 1..N loop

tempSum := 0;

for k in 1..N loop

tempSum := tempSum + MB1(i)(k) \* MC(j)(k);

end loop;

MA(i)(j) := tempSum + (MZ(i)(j) \* e1);

end loop;

end loop;

-- wait while T2 is calculating

Suspend\_Until\_True(s3);

-- output result

Put\_Line("Result MA is ");

outputMatrix(MA);

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

end T1;

task T2;

task body T2 is

MB2: Matrix;

e2: Integer;

tempSum: Integer;

begin

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

-- entering data

inputMatrix(MC);

e := 1;

-- send signal to T2 about end of input

Set\_True(s2);

-- wait until T2 ends entering

Suspend\_Until\_True(s1);

-- copy mutual resources

Suspend\_Until\_True(s0);

e2 := e;

for i in 1..N loop

for j in 1..N loop

MB2(i)(j) := MB(i)(j);

end loop;

end loop;

Set\_True(s0);

-- calculating

for j in H+1..N loop

for i in 1..N loop

tempSum := 0;

for k in 1..N loop

tempSum := tempSum + MB2(i)(k) \* MC(j)(k);

end loop;

MA(i)(j) := tempSum + (MZ(i)(j) \* e2);

end loop;

end loop;

-- send signal about and of calculating

Set\_True(s3);

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

end T2;

begin

null;

end startTasks;

begin

Set\_True(s0);

startTasks;

end Main;