Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №1**

**З предмету «Паралельні та розподілені обчислення»**

**Ада. Семафори**

Виконав:

Студент  
IІІ курсу ФІОТ  
групи ИП-21  
Кутовий А.С.

Залікова книжка №2107

Київ-2014

**Техническое задание**

1. Структура ПКС с ОП:

ОП

2

Z, MO

1

A, a, B, MK

1. Задача: A.
2. Язык программирования: Ада.
3. Средства взаимодействия задач: семафоры.

**Выполнение работы**

1. Разработка параллельного математического алгоритма.
   1. *.*

OP: a, MK, B.

1. Разработка алгоритмов процессов.

**Задача T1**:

* 1. Ввод *a, MK, B*.
  2. Сигнал T2 о завершении ввода.
  3. Ждать завершения ввода в T2.
  4. Копии
* B1 = B
  1. Счет1
  2. Сигнал о завершении счета 1
  3. Счет3
  4. Сигнал о завершении счета 3
  5. Ждать завершения счета в T2.
  6. Ждать завершения счета в T2.
  7. Счет 4
  8. Ждать завершения счета в T2.
  9. Вывод A.

**Задача T2**:

1. Ввод *Z, MO*.
2. Ждать завершения ввода в T1.
3. Сигнал T1 о завершении ввода.
4. Копии

* B2 = B

1. Счет1
2. Ждать завершение счета 1 в Т1
3. Счет2
4. Сигнал T1 о завершении счета.
5. Счет 3
6. Ожидаем завершения счета 3
7. Сигнал T1 о завершении счета.
8. Счет 4
9. Сигнал T1 о завершении счета.

**Разработка схемы взаимодействия процессов:**

T1

T2

S1

S2

S3

S0

S4

S5

S6

S7

**Разработка программы.**

**Листинг**:

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

-- Лабораторна робота №1. Ада. Семафори

-- Кутовий А.С.

-- ИП-21

-- A.

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

with Ada.Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Integer\_Text\_IO;

use Ada.Text\_IO, Ada.Synchronous\_Task\_Control, Ada.Integer\_Text\_IO;

procedure Main is

N: integer := 4;

P: integer := 2;

H: integer := N/P;

tmp: integer := 1;

alfa: integer;

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

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

A, Z, B: Vector;

MO, MK, MT: Matrix;

S0, S1, S2, S3, S4, S5, S6, S7: Suspension\_Object;

procedure inputMatrix(m : out Matrix) is

begin

for i in 1..N loop

for j in 1..N loop

m(i)(j) := tmp;

end loop;

end loop;

end inputMatrix;

procedure inputVector(v : out Vector) is

begin

for i in 1..N loop

v(i) := tmp;

end loop;

end inputVector;

procedure copyMatrix (mOut : out Matrix; mIn : in Matrix) is

begin

for i in 1..N loop

for j in 1..N loop

mOut(i)(j) := mIn(i)(j);

end loop;

end loop;

end copyMatrix;

procedure copyVector (v1 : out Vector; v2 : in Vector) is

begin

for i in 1..N loop

v1(i) := v2(i);

end loop;

end copyVector;

procedure StartTasks is

task T1 is

pragma Storage\_Size(1024\*1024\*700);

end T1;

task body T1 is

a1, tmp1 : integer;

B1, t1:Vector;

MK1: Matrix;

begin

--Ввод

Put\_Line("Task T1 is started.");

alfa := tmp;

inputMatrix(MK);

inputVector(B);

--Сигнал о завершении ввода

Set\_True(S1);

--ожидаем завершение ввода в другом потоке

Suspend\_Until\_True(S2);

--копируем

Suspend\_Until\_True(S0);

copyMatrix(MK1, MK);

a1 := alfa;

copyVector(B1,B);

Set\_True(S0);

--Счет1

for i in 1..H-1 loop

for j in 1..H-1 loop

if(Z(i)>Z(i+1)) then

tmp1 := Z(i);

Z(i) := Z(i+1);

Z(i+1) := tmp1;

end if;

end loop;

end loop;

for i in 1..H loop

Z(i) := Z(i)\*a1;

end loop;

--сигнал о окончании счета 1

Set\_True(S3);

--счет 3

for i in 1..H loop

for j in 1..N loop

MT(i)(j):=0;

for k in 1..N loop

MT(i)(j) := MT(i)(j)+MO(i)(k)\*MK1(k)(j);

end loop;

end loop;

end loop;

Set\_True(S4);

--ожидаем завершения счета 2 и 3

Suspend\_Until\_True(S5);

Suspend\_Until\_True(S6);

--счет 4

for i in 1..H loop

t1(i) := 0;

for j in 1..N loop

t1(i):= t1(i)+B1(j)\*MT(i)(j);

end loop;

end loop;

for i in 1..H loop

A(i) := t1(i) - Z(i);

end loop;

--ожидаем завершения счета 4

Suspend\_Until\_True(S7);

--вывод

if(N<10) then

for i in 1..N loop

Put(A(i));

Put(" ");

end loop;

end if;

put\_line(" ");

put\_line("T1 finish");

end T1;

task T2 is

pragma Storage\_Size(1024\*1024\*700);

end T2;

task body T2 is

a2, tmp2, i1,i2 : integer;

MK2: Matrix;

B2, t2: Vector;

begin

Put\_Line("Task T2 is started.");

--ввод

inputMatrix(MO);

inputVector(Z);

--сигнал о завершении ввода

Suspend\_Until\_True(S1);

--ожидаем завершения ввода

Set\_True(S2);

--копируем

Suspend\_Until\_True(S0);

copyMatrix(MK2, MK);

a2 := alfa;

copyVector(B2,B);

Set\_True(S0);

--счет1

for i in H+1..N-1 loop

for j in H+1..N-1 loop

if(Z(i)>Z(i+1)) then

tmp2 := Z(i);

Z(i) := Z(i+1);

Z(i+1) := tmp2;

end if;

end loop;

end loop;

for i in H+1..N loop

Z(i) := Z(i)\*a2;

end loop;

--ожидаем сигнала о завершеи счета 1

Suspend\_Until\_True(S3);

--счет2

copyVector(t2,Z);

i1 := 1;

i2 := H+1;

for i in 1..N loop

if t2(i1)<t2(i2) then

Z(i):=t2(i1);

t2(i1):= 999999;

i1 := i1+1;

else

Z(i):=t2(i2);

t2(i2):= 999999;

if(i2 < N) then

i2 := i2+1;

end if;

end if;

end loop;

--сигнал завершнии счета2

Set\_True(S5);

--счет 3

for i in H+1..N loop

for j in 1..N loop

MT(i)(j):=0;

for k in 1..N loop

MT(i)(j) := MT(i)(j)+MO(i)(k)\*MK2(k)(j);

end loop;

end loop;

end loop;

--сигнал о завершении счета 3

Set\_True(S6);

--ожидаем завершения счета 3

Suspend\_Until\_True(S4);

--счет4

for i in H+1..N loop

t2(i) := 0;

for j in 1..N loop

t2(i):= t2(i)+B2(j)\*MT(i)(j);

end loop;

end loop;

for i in H+1..N loop

A(i) := t2(i) - Z(i);

end loop;

--сигнал о завершении счета4

Set\_True(S7);

put\_line("T2 finish");

end T2;

begin

null;

end StartTasks;

begin

put\_line("Main procedure started");

Set\_True(S0);

StartTasks;

put\_line("Main procedure finished");

end Main;