**Национальный технический университет Украины**

**«Киевский Политехнический Институт»**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

**Лабораторная работа №1**

по курсу: Параллельные и распределённые вычисления

Тема: Программирование для ПКС с ОП. Ада. Семафоры

Выполнила студентка

III курса ФИВТ

группы ИО-91

Косейкина Анна Сергеевна

Киев 2012

Задание на лабораторную работу:

Математическая задача:

Структура ПКС:

ОП

2

1

MZ, ME MA, ME

Средства программирования:

Язык АДА, семафоры

Выполнение работы.

Этап 1. Разработка параллельного математического алгоритма

1. a=min(a, ai)

ОР: а

1. MAH=a\*(MO\*MEH)

ОР: МО, а

Этап 2. Разработка алгоритмов задач

Т1

1. Ввод данных MZ, ME
2. Сигнал Т2 о вводе S2,1
3. Счёт a1= min(MZH)
4. Счёт a=min(a, a1) КУ
5. Сигнал Т2 о завершении счёта а S2,2
6. Ждать завершения счёта а в Т2 W2,1
7. Копия а1=а КУ
8. Копия МТ1=МО КУ
9. Счёт MAH=a\*(MТ1\*MEH)
10. Сигнал Т2 о завершении счёта S2,3

Т2

1. Ввод данных МО
2. Ждать сигнал от Т1 о вводе данных W1,1
3. Счёт a2= min(MZH)
4. Счёт a=min(a, a2) КУ
5. Сигнал Т1 о завершении счёта а S1,1
6. Ждать завершения счёта а в Т1 W1,2
7. Копия а2=а КУ
8. Копия МТ2=МО КУ
9. Счёт MAH=a\*(MТ2\*MEH)
10. Ждать сигнала Т1 о завершении счёта W1,3
11. Вывод МА

Этап 3. Разработка схемы взаимодействия задач

Этап 4. Разработка программы -- Лаб1. Ада. Семафоры

-- MA=min(MZ)(MO\*ME)

-- Косейкина Анна Сергеевна, группа IО-91

-- 01.02.2012

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

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

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

procedure Lab1 is

N : integer := 5;

P : integer := 2;

H : integer := N/P;

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

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

a : integer := 32767;

MA, MZ, MO, ME : Matrix;

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

procedure Start is

task T1;

task body T1 is

a1 : integer;

i : integer := 0;

j : integer := 0;

k : integer := 0;

MO1 : Matrix;

begin

put ("T1 started");

NEW\_LINE;

for i in 1..N LOOP

for j in 1..N loop

MZ(i)(j) := 1;

ME(i)(j) := 1;

end loop;

end loop;

Set\_true (S3);

a1 := 32767;

for i in 1..H loop

for j in 1..N loop

if (a1>MZ(i)(j)) then

a1 := MZ(i)(j);

end if;

end loop;

end loop;

Suspend\_Until\_True (S1);

if (a>a1) then

a := a1;

end if;

Set\_True(S1);

Set\_True(S4);

Suspend\_Until\_True (S5);

Suspend\_Until\_True (S1);

a1 := a;

Set\_True (S1);

Suspend\_Until\_True (S2);

MO1 := MO;

Set\_True (S2);

for i in 1..H loop

for j in 1..N loop

MA(i)(j) := 0;

for k in 1..N loop

MA(i)(j) := MA(i)(j) + ME(i)(j)\*MO1(k)(j);

end loop;

MA(i)(j) := a1\*MA(i)(j);

end loop;

end loop;

Set\_True (S6);

put ("T1 finished");

new\_line;

end T1;

task T2;

task body T2 is

a2 : integer;

i : integer := 0;

j : integer := 0;

k : integer := 0;

MO2 : Matrix;

begin

put ("T2 started");

NEW\_LINE;

for i in 1..N loop

for j in 1..N loop

MO(i)(j) := 1;

end loop;

end loop;

Suspend\_Until\_True (S3);

a2 := 32767;

for i in H..N loop

for j in 1..N loop

if (a2>MZ(i)(j)) then

a2 := MZ(i)(j);

end if;

end loop;

end loop;

Suspend\_Until\_True (S1);

if (a>a2) then

a := a2;

end if;

Set\_True(S1);

Set\_True(S5);

Suspend\_Until\_True (S4);

Suspend\_Until\_True (S1);

a2 := a;

Set\_True (S1);

Suspend\_Until\_True (S2);

MO2 := MO;

Set\_True (S2);

for i in H..N loop

for j in 1..N loop

MA(i)(j) := 0;

for k in 1..N loop

MA(i)(j) := MA(i)(j) + ME(i)(j)\*MO2(k)(j);

end loop;

MA(i)(j) := a2\*MA(i)(j);

end loop;

end loop;

Suspend\_Until\_True(S6);

put (MA(1)(1));

new\_line;

put ("T2 finished");

end T2;

begin

null;

end Start;

begin

Set\_True (S1);

Set\_True (S2);

Start;

end Lab1;