Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Основи паралельного програмування

Лабораторна робота №1

**«Потоки в мові АДА»**

Виконала:

студентка групи ІО-64

Бровченко А. В.

Перевірив:

Корочкін О. В.

Київ

2018 р.

**Завдання:**

F1: C = A – B\*(MA\*MС)\*е

F2: MF = MF\*MG\*k

F3: O = SORT(P)\*(MR\*MT)

**Лістинг програми:**

main\_lab.adb

----------------Main programm------------------------

--Parallel and distributed computing.

--Labwork 1. Ada. Subprograms and packages

--Anastasiya Brovchenko

--IO-64

--16.09.2018

--Func1: C = A-B\*(MA\*MC)\*e

--Func2: MF = MF\*MG\*k

--Func3: O = sort(P)\*(MR\*MT)

with Data, Text\_IO, Ada.Integer\_Text\_IO, System.Multiprocessors;

use Text\_IO, Ada.Integer\_Text\_IO, System.Multiprocessors;

procedure Main\_lab1 is

n: Integer := 3 ;

package data1 is new data (n);

use data1;

procedure tasks is

task T1 is

pragma Priority(1);

pragma Storage\_Size(100000);

pragma CPU(1);

end;

task body T1 is

A, B: Vector;

MA, MC: Matrix;

e: Integer;

f1: Vector;

begin

Put\_Line("T1 started");

Vector\_Filling\_Ones(A);

Vector\_Filling\_Ones(B);

Matrix\_Filling\_Ones(MA);

Matrix\_Filling\_Ones(MC);

e:=1;

f1:=Func1(A, B, MA, MC, e);

delay(2.0);

if n < 7 then

Put\_Line("---Func1 = sort(B)\*(MB\*MC)---");

Vector\_Output(f1);

New\_Line;

New\_Line;

end if;

Put\_Line("T1 finished");

end T1;

task T2 is

pragma Priority(3);

pragma Storage\_Size(100000);

pragma CPU(2);

end;

task body T2 is

MF, MG, f2: Matrix;

k: Integer;

begin

Put\_Line("T2 started");

Matrix\_Filling\_Ones(MF);

Matrix\_Filling\_Ones(MG);

k:=1;

f2:=Func2(MF, MG, k);

delay(2.0);

if n < 7 then

Put\_Line("---Func2 = sort(R\*MT)\*(MX\*MS)---");

Matrix\_Output(f2);

New\_Line;

New\_Line;

end if;

Put\_Line("T2 finished");

end T2;

task T3 is

pragma Priority(5);

pragma Storage\_Size(100000);

pragma CPU(3);

end;

task body T3 is

MT, MR : Matrix;

P: Vector;

f3: Vector;

begin

Put\_Line("T3 started");

Vector\_Filling\_Ones(P);

Matrix\_Filling\_Ones(MR);

Matrix\_Filling\_Ones(MT);

f3:=Func3(P, MR, MT);

delay(2.0);

if n < 7 then

Put\_Line("---Func3 = sort(S+T)\*trans(MS\*MR)---");

Vector\_Output(f3);

New\_Line;

New\_Line;

end if;

Put\_Line("T3 finished");

end T3;

begin

null;

end tasks;

Begin

tasks;

End Main\_lab1;

data.adb

-----------Package Data, body-----------

with Text\_IO, Ada.Integer\_Text\_IO;

use Text\_IO, Ada.Integer\_Text\_IO;

package body Data is

--Read Vector

procedure Vector\_Input(A: out Vector) is

begin

for i in 1..n loop

Get(A(i));

end loop;

end Vector\_Input;

--Write vector on screen

procedure Vector\_Output(A: in Vector) is

begin

for i in 1..n loop

Put(A(i));

Put(" ");

end loop;

end Vector\_Output;

--Read matrix

procedure Matrix\_Input(A: out Matrix) is

begin

for i in 1..n loop

for j in 1..n loop

Get(A(i)(j));

end loop;

end loop;

end Matrix\_Input;

--Write matrix on screen

procedure Matrix\_Output (A: in Matrix) is

begin

for i in 1..n loop

for j in 1..n loop

Put(A(i)(j));

Put(" ");

end loop;

Put\_Line(" ");

end loop;

end Matrix\_Output;

--Multiplication of matrices

function Matrix\_Multiplication(A, B: in Matrix) return Matrix is

P: Matrix;

S: Integer;

begin

for k in 1..n loop

for i in 1..n loop

s := 0;

for j in 1..n loop

S := S + A(k)(j)\*B(j)(i);

P(k)(i) := s;

end loop;

end loop;

end loop;

return P;

end Matrix\_Multiplication;

--Multiplication of matrix and integer

function Matrix\_Integer\_Multiplication(A: in Matrix; k: in Integer) return Matrix is

P: Matrix;

begin

for i in 1..n loop

for j in 1..n loop

P(i)(j):=A(i)(j)\*k;

end loop;

end loop;

return P;

end Matrix\_Integer\_Multiplication;

--Multiplication of vector and matrix

function Vector\_Matrix\_Multiplication(A: in Vector; B: in Matrix) return Vector is

P: Vector;

s: Integer;

begin

for i in 1..n loop

s := 0;

for j in 1..n loop

S := s + A(i)\*B(j)(i);

end loop;

P(i) := S;

end loop;

return P;

end Vector\_Matrix\_Multiplication;

--Multiplication of Vector and Integer

function Vector\_Integer\_Multiplication (A: in Vector; e: in Integer) return Vector is

B: Vector;

begin

for i in 1..n loop

B(i) := A(i)\*e;

end loop;

return B;

end Vector\_Integer\_Multiplication;

--Sum of vectors

function Vector\_Sum(A, B: in Vector) return Vector is

S: Vector;

begin

for i in 1..n loop

S(i) := A(i)+B(i);

end loop;

return S;

end Vector\_Sum;

--Difference of Vectors

function Vector\_Difference(A, B: in Vector) return Vector is

S: Vector;

begin

for i in 1..n loop

S(i) := A(i)-B(i);

end loop;

return S;

end Vector\_Difference;

--Sorting of vector

procedure Vector\_Sorting(A: in out Vector) is

S: Integer;

begin

for i in 1..n loop

for j in i..n loop

if A(i)>A(j) then

S:=A(j);

A(j):=A(i);

A(i):=S;

end if;

end loop;

end loop;

end Vector\_Sorting;

--Transposition of Matrix

procedure Matrix\_Transposition(A: in out Matrix) is

S: Integer;

begin

for i in 1..n loop

for j in i..n loop

S:=A(j)(i);

A(j)(i):=A(i)(j);

A(i)(j):=S;

end loop;

end loop;

end Matrix\_Transposition;

--Filling matrix with ones

procedure Matrix\_Filling\_Ones(A: out Matrix) is

begin

for i in 1..n loop

for j in 1..n loop

A(i)(j) := 1;

end loop;

end loop;

end Matrix\_Filling\_Ones;

--Filling vector with ones

procedure Vector\_Filling\_Ones (A: out vector) is

begin

for i in 1..n loop

A(i) := 1;

end loop;

end Vector\_Filling\_Ones;

procedure Matrix\_Filling\_Number(A: out Matrix; i,j,number:Integer) is

begin

A(i)(j) := number;

end Matrix\_Filling\_Number;

procedure Vector\_Filling\_Number(A: out Vector; i,number:integer) is

begin

A(i) := number;

end Vector\_Filling\_Number;

--Calculation function 1

function Func1 (A, B: in Vector; MA, MC : in Matrix; e: in Integer) return Vector is

MD:Matrix;

C,D,F:Vector;

begin

MD:=Matrix\_Multiplication(MA,MC);

D:=Vector\_Matrix\_Multiplication(B,MD);

F:=Vector\_Integer\_Multiplication(D,e);

C:=Vector\_Difference(A,F);

return C;

end Func1;

--Calculation function 2

function Func2 (MF, MG: in Matrix; k: in Integer) return Matrix is

MD, MN: Matrix;

begin

MD:=Matrix\_Multiplication(MF,MG);

MN:=Matrix\_Integer\_Multiplication(MD, k);

return MN;

end Func2;

--Calculation function 3

function Func3 (P: out Vector; MR, MT : in Matrix) return Vector is

MD:Matrix;

O:Vector;

begin

Vector\_Sorting(P);

MD:=Matrix\_Multiplication(MR,MT);

O:=Vector\_Matrix\_Multiplication(P,MD);

return O;

end Func3;

end Data;

data.ads

generic

n: Integer;

package Data is

---Declaration of private types

type Vector is private;

type Matrix is private;

--Read Vector

procedure Vector\_Input(A: out Vector);

--Write vector on screen

procedure Vector\_Output(A: in Vector);

--Read matrix

procedure Matrix\_Input(A: out Matrix);

--Write matrix on screen

procedure Matrix\_Output (A: in Matrix);

--Calculation function 1

function Func1 (A, B: in Vector; MA, MC : in Matrix; e: in Integer) return Vector;

--Calculation function 2

function Func2 (MF, MG: in Matrix; k: in Integer) return Matrix;

--Calculation function 3

function Func3 (P: out Vector; MR, MT : in Matrix) return Vector;

--Filling matrix with ones

procedure Matrix\_Filling\_Ones(A: out Matrix);

--Filling vector with ones

procedure Vector\_Filling\_Ones (A: out vector);

procedure Matrix\_Filling\_Number(A: out Matrix; i,j,number:Integer);

procedure Vector\_Filling\_Number(A: out Vector; i,number:integer);

--Determination private types

private

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

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

end Data;

**Результат роботи:**

D:\GNAT\2018\bin\obj\main\_lab1.exe

T1 started

T2 started

T3 started

---Func1 = sort(B)\*(MB\*MC)---

-8 -8 -8

T1 finished

---Func2 = sort(R\*MT)\*(MX\*MS)---

3 3 3

3 3 3

3 3 3

T2 finished

---Func3 = sort(S+T)\*trans(MS\*MR)---

9 9 9

T3 finished

[2018-09-20 12:51:10] process terminated successfully, elapsed time: 02.24s