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

--

-- Parallel and Distributed Computing --

-- Laboratory work #5. Ada. Protected unit --

-- --

-- Task: MA = a\*MO+ (B\*C)(MT\*MR) --

-- Kobylynskyi D.A. IP-31 --

-- --

-- Date: 13.04.2016 --

-- --

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

with Ada.Text\_IO, Ada.Integer\_text\_iO, Ada.Synchronous\_Task\_Control, Data;

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

procedure Lab5 is

Value : Integer :=1;

N: Natural :=2000;

package DataN is new Data(N);

use DataN;

P: Natural :=4;

H: Natural:= N/P;

B, C : Vector;

MT,MO, MR, MA: Matrix;

buf : Matrix;

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

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

protected Synchronization is

entry WaitForInput;

entry WaitForCalcV;

entry WaitForCalcMA;

procedure InputSignal;

procedure CalcVSignal;

procedure CalcMASignal;

private

inputFlag:Natural:=0;

vFlag:Natural:=0;

MAflag:Natural:=0;

end Synchronization;

protected GeneralResourse is

procedure addV(data : in Integer);

procedure setMR(data : in Matrix);

procedure setAlfa(data : in Integer);

function CopyAlfa return Integer;

function CopyV return Integer;

function CopyMR return Matrix;

private

alfa: Integer;

v: Integer:=0;

MR:Matrix;

end GeneralResourse;

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

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

protected body Synchronization is

procedure InputSignal is

begin

inputFlag := inputFlag + 1;

end InputSignal;

procedure CalcVSignal is

begin

vFlag := vFlag + 1;

end CalcVSignal;

procedure CalcMASignal is

begin

MAflag := MAflag + 1;

end CalcMASignal;

entry WaitForInput

when inputFlag = 3 is

begin

null;

end WaitForInput;

entry WaitForCalcV

when vFlag = 4 is

begin

null;

end WaitForCalcV;

entry WaitForCalcMA

when MAflag = 3 is

begin

null;

end WaitForCalcMA;

end Synchronization;

protected body GeneralResourse is

procedure addV(data : in Integer) is

begin

v := v+data;

end addV;

procedure setAlfa(data : in Integer) is

begin

alfa := data;

end setAlfa;

procedure setMR(data : in Matrix) is

begin

MR:=data;

end setMR;

function CopyAlfa return Integer is

begin

return alfa;

end CopyAlfa;

function CopyV return Integer is

begin

return v;

end CopyV;

function CopyMR return Matrix is

begin

return MR;

end CopyMR;

end GeneralResourse;

procedure StartTasks is

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

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

task T1;

TMP1:Integer;

MR1:Matrix;

v1:Integer;

alfa1:Integer;

task body T1 is

begin

Put\_Line ("T1 started");

Input(Value, MR);

Synchronization.InputSignal;

Synchronization.WaitForInput;

v1 :=0;

for i in 1..H loop

v1:=v1+B(i)\*C(I);

end loop;

GeneralResourse.addV(v1);

Synchronization.CalcVSignal;

Synchronization.WaitForCalcV;

alfa1 := GeneralResourse.CopyAlfa;

MR1 := GeneralResourse.CopyMR;

v1 := GeneralResourse.CopyV;

for i in 1..H loop

for j in 1..N loop

TMP1:=0;

for k in 1..N loop

TMP1:=TMP1+(MT(i)(k)\*MR(k)(j));

Buf(i)(j):=TMP1;

end loop;

end loop;

end loop;

for i in 1..H loop

for j in 1..N loop

MA(i)(j) := Buf(i)(j) \* V1 + alfa1 \* MO(i)(j);

end loop;

end loop;

Synchronization.CalcMASignal;

Put\_Line ("T1 finished");

end T1;

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

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

task T2;

TMP2:Integer;

v2: Integer;

alfa2:Integer;

MR2: Matrix;

task body T2 is

begin

Put\_Line ("T2 started");

Synchronization.WaitForInput;

v2 :=0;

for i in H+1..2\*H loop

v2:=v2+B(i)\*C(I);

end loop;

GeneralResourse.addV(v2);

Synchronization.CalcVSignal;

Synchronization.WaitForCalcV;

alfa2 := GeneralResourse.CopyAlfa;

MR2 := GeneralResourse.CopyMR;

v2 := GeneralResourse.CopyV;

for i in H+1..2\*H loop

for j in 1..N loop

TMP2:=0;

for k in 1..N loop

TMP2:=TMP2+(MT(i)(k)\*MR(k)(j));

Buf(i)(j):=TMP2;

end loop;

end loop;

end loop;

for i in H+1..2\*H loop

for j in 1..N loop

MA(i)(j) := Buf(i)(j) \* V2 + alfa2 \* MO(i)(j);

end loop;

end loop;

Synchronization.CalcMASignal;

Put\_Line ("T2 finished");

end T2;

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

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

task T3;

TMP3:Integer;

v3: Integer;

alfa3:Integer;

MR3: Matrix;

task body T3 is

begin

Put\_Line ("T3 started");

Input(Value, B);

Input(Value, MO);

Input(Value, MT);

Synchronization.InputSignal;

Synchronization.WaitForInput;

v3 :=0;

for i in 2\*H+1..3\*H loop

v3:=v3+B(i)\*C(I);

end loop;

GeneralResourse.addV(v3);

Synchronization.CalcVSignal;

Synchronization.WaitForCalcV;

alfa3 := GeneralResourse.CopyAlfa;

MR3 := GeneralResourse.CopyMR;

v3 := GeneralResourse.CopyV;

for i in 2\*H+1..3\*H loop

for j in 1..N loop

TMP3:=0;

for k in 1..N loop

TMP3:=TMP3+(MT(i)(k)\*MR(k)(j));

Buf(i)(j):=TMP3;

end loop;

end loop;

end loop;

for i in 2\*H+1..3\*H loop

for j in 1..N loop

MA(i)(j) := Buf(i)(j) \* V3 + alfa3 \* MO(i)(j);

end loop;

end loop;

Synchronization.CalcMASignal;

Put\_Line ("T3 finished");

end T3;

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

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

task T4;

TMP:Integer;

v4: Integer;

alfa4:Integer;

MR4: Matrix;

task body T4 is

begin

Put\_Line ("T4 started");

GeneralResourse.setAlfa(Value);

Input(Value, C);

Synchronization.InputSignal;

Synchronization.WaitForInput;

v4 :=0;

for i in 3\*H+1..4\*H loop

v4:=v4+B(i)\*C(I);

end loop;

GeneralResourse.addV(v4);

Synchronization.CalcVSignal;

Synchronization.WaitForCalcV;

alfa4 := GeneralResourse.CopyAlfa;

MR4 := GeneralResourse.CopyMR;

v4 := GeneralResourse.CopyV;

for i in 3\*H+1..4\*H loop

for j in 1..N loop

TMP:=0;

for k in 1..N loop

TMP:=TMP+(MT(i)(k)\*MR(k)(j));

Buf(i)(j):=TMP;

end loop;

end loop;

end loop;

for i in 3\*H+1..4\*H loop

for j in 1..N loop

MA(i)(j) := Buf(i)(j) \* V4 + alfa4 \* MO(i)(j);

end loop;

end loop;

Synchronization.WaitForCalcMA;

Output(MA);

Put\_Line ("T4 finished");

end T4;

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

begin

null;

end StartTasks;

begin

Put\_Line ("Lab5 started");

StartTasks;

Put\_Line ("Lab5 finished");

end Lab5;