Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №5**

**з дисципліни «Програмування паралельних комп’ютерних систем»**

**на тему «Ада. Захищені модулі»**

Виконав:

студент ІІI курсу

групи ІО-53

Лісовий В. О.

Перевірив:

доц. Корочкін О. В.

Київ – 2018

**Тема**

«Програмування для комп’ютерних систем зі спільною пам’яттю. Ада. Захищенні модулі»

**Технічне завдання**

Розробити програму для розв’язання ПКС із СП (структура на рис. 1) математичної задачі: A = (B \* C) \* S + min(T) \* Z \* (MO \* MH).

Мова програмування: C/C++, бібліотека OpenMP.

Засоби взаємодії: бар’єри, критичні секції, замки.



Рис. 1 Структурна схема ПКС

**Виконання роботи**

**Етап 1. Побудова паралельного алгоритму**

1. ai = (BH \* CH), i = 1..4
2. a = a + ai
3. bi = min(TH)
4. b = min(b, bi)
5. AH = a \* SH + b \* Z \* MO \* MHH;

Спільний ресурс: a, b, Z, MO.

**Етап 2. Розроблення алгоритмів роботи кожного процесу**

|  |  |  |
| --- | --- | --- |
| № | Задача Т1 |  |
| 1 | Ввід даних (B, C, MH) |  |
| 2 | Сигнал усім потокам про завершення вводу даних | S\*,1 |
| 3 | Чекати завершення вводу даних в Т3, Т4 | W3,1 , W4,1 |
| 4 | Обчислення a1 = (BH \* CH) |  |
| 5 | Обчислення a = a + a1 | КД |
| 6 | Обчислення b1 = min (TH) |  |
| 7 | Обчислення b = min (b, b1) | КД |
| 8 | Копіювання al = a, b1 = b, Z1 = Z, MO1 = MO | КД |
| 9 | Сигнал усім потокам про завершення обчислення скаляру та пошуку мінімуму | S\*,2 |
| 10 | Чекати завершення завершення обчислення скаляру та пошуку мінімуму в Т2, Т3, Т4 | W\*,2 |
| 11 | Обчислення AH = a1 \* SH + b1 \* Z1 \* MO1 \* MHH |  |
| 12 | Чекати завершення завершення обчислення в Т2, Т3, Т4 | W\*,3 |
| 13 | Вивід А |  |

|  |  |  |
| --- | --- | --- |
| № | Задача Т2 |  |
| 1 | Чекати завершення вводу даних в Т1, Т3, Т4 | W\*,1 |
| 2 | Обчислення a2 = (BH \* CH) |  |
| 3 | Обчислення a = a + a2 | КД |
| 4 | Обчислення b2 = min (TH) |  |
| 5 | Обчислення b = min (b, b2) | КД |
| 6 | Копіювання a2 = a, b2 = b, Z2 = Z, MO2 = MO | КД |
| 7 | Сигнал усім потокам про завершення обчислення скаляру та пошуку мінімуму | S\*,2 |
| 8 | Чекати завершення завершення обчислення скаляру та пошуку мінімуму в Т1, Т3, Т4 | W\*,2 |
| 9 | Обчислення AH = a2 \* SH + b2 \* Z2 \* MO2 \* MHH |  |
| 10 | Сигнал T1 про завершення обчислення | S1,3 |

|  |  |  |
| --- | --- | --- |
| № | Задача Т3 |  |
| 1 | Ввід даних (S, MO) |  |
| 2 | Сигнал усім потокам про завершення вводу даних | S\*,1 |
| 3 | Чекати завершення вводу даних в Т1, Т4 | W1,1 , W4,1 |
| 4 | Обчислення a3 = (BH \* CH) |  |
| 5 | Обчислення a = a + a3 | КД |
| 6 | Обчислення b3 = min (TH) |  |
| 7 | Обчислення b = min (b, b3) | КД |
| 8 | Копіювання a3 = a, b3 = b, Z3 = Z, MO3 = MO | КД |
| 9 | Сигнал усім потокам про завершення обчислення скаляру та пошуку мінімуму | S\*,2 |
| 10 | Чекати завершення завершення обчислення скаляру та пошуку мінімуму в Т1, Т2, Т4 | W\*,2 |
| 11 | Обчислення AH = a3 \* SH + b3 \* Z3 \* MO3 \* MHH |  |
| 12 | Сигнал T1 про завершення обчислення | S1,3 |

|  |  |  |
| --- | --- | --- |
| № | Задача Т4 |  |
| 1 | Ввід даних (T, Z) |  |
| 2 | Сигнал усім потокам про завершення вводу даних | S\*,1 |
| 3 | Чекати завершення вводу даних в Т1, Т3 | W1,1 , W3,1 |
| 4 | Обчислення a4 = (BH \* CH) |  |
| 5 | Обчислення a = a + a4 | КД |
| 6 | Обчислення b4 = min (TH) |  |
| 7 | Обчислення b = min (b, b4) | КД |
| 8 | Копіювання a4 = a, b4 = b, Z4 = Z, MO4 = MO | КД |
| 9 | Сигнал усім потокам про завершення обчислення скаляру та пошуку мінімуму | S\*,2 |
| 10 | Чекати завершення завершення обчислення скаляру та пошуку мінімуму в Т1, Т2, Т3 | W\*,2 |
| 11 | Обчислення AH = a4 \* SH + b4 \* Z4 \* MO4 \* MHH |  |
| 12 | Сигнал T1 про завершення обчислення | S1,3 |

**Етап 3. Розробка програми**

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

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

with Data;

procedure Main is

N : Integer := 4;

package new\_data is new data(Size => N);

use new\_data;

A, B, C, S, T : Vector;

MH: Matrix;

protected **ResourceMonitor** is

procedure SetScalar(scalar: in Integer);

procedure SetMin(min: in Integer);

procedure SetZ(Z: in Vector);

procedure SetMO(MO: in Matrix);

function GetScalar return Integer;

function GetMin return Integer;

function GetZ return Vector;

function GetMO return Matrix;

private

scalar\_g: Integer := 0;

min\_g: Integer := 200000000;

Z\_g: Vector;

MO\_g: Matrix;

end **ResourceMonitor**;

protected **SignalMonitor** is

procedure InputSignal;

procedure MinScalarSignal;

procedure CalcSignal;

entry WaitInput;

entry WaitMinScalar;

entry WaitCalcSignal;

private

F1: Integer := 0;

F2: Integer := 0;

F3: Integer := 0;

end **SignalMonitor**;

protected body **ResourceMonitor** is

procedure SetScalar(scalar: in Integer) is

begin

scalar\_g := scalar\_g + scalar;

end SetScalar;

procedure SetMin(min: in Integer) is

begin

if (min < min\_g) then

min\_g := min;

end if;

end SetMin;

procedure SetZ(Z: in Vector) is

begin

Z\_g := Z;

end SetZ;

procedure SetMO(MO: in Matrix) is

begin

MO\_g := MO;

end SetMO;

function GetScalar return Integer is

begin

return scalar\_g;

end GetScalar;

function GetMin return Integer is

begin

return min\_g;

end GetMin;

function GetZ return Vector is

begin

return Z\_g;

end GetZ;

function GetMO return Matrix is

begin

return MO\_g;

end GetMO;

end **ResourceMonitor**;

protected body **SignalMonitor** is

procedure InputSignal is

begin

F1 := F1 + 1;

end InputSignal;

procedure MinScalarSignal is

begin

F2 := F2 + 1;

end MinScalarSignal;

procedure CalcSignal is

begin

F3 := F3 + 1;

end CalcSignal;

entry WaitInput

when F1 = 3 is

begin

null;

end WaitInput;

entry WaitMinScalar

when F2 = 4 is

begin

null;

end WaitMinScalar;

entry WaitCalcSignal

when F3 = 3 is

begin

null;

end WaitCalcSignal;

end **SignalMonitor**;

task Task1 is

pragma Storage\_Size(65535);

end Task1;

task Task2 is

pragma Storage\_Size(65535);

end Task2;

task Task3 is

pragma Storage\_Size(65535);

end Task3;

task Task4 is

pragma Storage\_Size(65535);

end Task4;

task body Task1 is

scalar1, min1 : Integer;

Z1: Vector;

MO1: Matrix;

begin

Put\_Line("Task 1 start");

Fill\_Vector\_With\_Num(B, 1);

Fill\_Vector\_With\_Num(C, 1);

Fill\_Matrix\_With\_Num(MH, 1);

SignalMonitor.InputSignal;

SignalMonitor.WaitInput;

scalar1 := Scalar(B, C, 1, n / 4);

ResourceMonitor.SetScalar(scalar1);

min1 := Vector\_Min(T, 1, n / 4);

ResourceMonitor.SetMin(min1);

SignalMonitor.MinScalarSignal;

SignalMonitor.WaitMinScalar;

scalar1 := ResourceMonitor.GetScalar;

min1 := ResourceMonitor.GetMin;

Z1 := ResourceMonitor.GetZ;

MO1 := ResourceMonitor.GetMO;

F(A, scalar1, S, min1, Z1, MO1, MH, 1, n/4);

SignalMonitor.WaitCalcSignal;

if n < 7 then

Put("A = ");

Vector\_Output(A);

end if;

Put\_Line("Task 1 finish");

end Task1;

task body Task2 is

scalar2, min2 : Integer;

Z2: Vector;

MO2: Matrix;

begin

Put\_Line("Task 2 start");

SignalMonitor.WaitInput;

scalar2 := Scalar(B, C, n/4 + 1, n / 2);

ResourceMonitor.SetScalar(scalar2);

min2 := Vector\_Min(T, n/ 4 + 1, n / 2);

ResourceMonitor.SetMin(min2);

SignalMonitor.MinScalarSignal;

SignalMonitor.WaitMinScalar;

scalar2 := ResourceMonitor.GetScalar;

min2 := ResourceMonitor.GetMin;

Z2 := ResourceMonitor.GetZ;

MO2 := ResourceMonitor.GetMO;

F(A, scalar2, S, min2, Z2, MO2, MH, n/4 + 1, n/2);

SignalMonitor.CalcSignal;

Put\_Line("Task 2 finish");

end Task2;

task body Task3 is

scalar3, min3 : Integer;

Z3: Vector;

MO3: Matrix;

begin

Put\_Line("Task 3 start");

Fill\_Vector\_With\_Num(S, 1);

Fill\_Matrix\_With\_Num(MO3, 1);

ResourceMonitor.SetMO(MO3);

SignalMonitor.InputSignal;

SignalMonitor.WaitInput;

scalar3 := Scalar(B, C, n / 2 + 1, 3 \* n / 4);

ResourceMonitor.SetScalar(scalar3);

min3 := Vector\_Min(T, n / 2 + 1, 3 \* n / 4);

ResourceMonitor.SetMin(min3);

SignalMonitor.MinScalarSignal;

SignalMonitor.WaitMinScalar;

scalar3 := ResourceMonitor.GetScalar;

min3 := ResourceMonitor.GetMin;

Z3 := ResourceMonitor.GetZ;

MO3 := ResourceMonitor.GetMO;

F(A, scalar3, S, min3, Z3, MO3, MH, n/2 + 1, 3\*n/4);

SignalMonitor.CalcSignal;

Put\_Line("Task 3 finish");

end Task3;

task body Task4 is

scalar4, min4 : Integer;

Z4: Vector;

MO4: Matrix;

begin

Put\_Line("Task 4 start");

Fill\_Vector\_With\_Num(T, 1);

--Num\_Vector(T, -1, 2);

Fill\_Vector\_With\_Num(Z4, 1);

ResourceMonitor.SetZ(Z4);

SignalMonitor.InputSignal;

SignalMonitor.WaitInput;

scalar4 := Scalar(B, C, 3 \* n / 4 + 1, n);

ResourceMonitor.SetScalar(scalar4);

min4 := Vector\_Min(T, 3 \* n / 4 + 1, n);

ResourceMonitor.SetMin(min4);

SignalMonitor.MinScalarSignal;

SignalMonitor.WaitMinScalar;

scalar4 := ResourceMonitor.GetScalar;

min4 := ResourceMonitor.GetMin;

Z4 := ResourceMonitor.GetZ;

MO4 := ResourceMonitor.GetMO;

F(A, scalar4, S, min4, Z4, MO4, MH, 3 \* n / 4 + 1, n);

SignalMonitor.CalcSignal;

Put\_Line("Task 4 finish");

end Task4;

begin

Put\_Line("Main start");

Put\_Line("Main finish");

end Main

;