# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп’ютеризованих систем управління

ЛАБОРАТОРНА РОБОТА №1

з дисципліни “Паралельні та розподілені обчислення”

на тему «Програмування для комп’ютерних систем зі спільною пам’яттю»

Виконала:

студентка групи СП-425

Гурман К.Д.

Перевірив:

Корочкін О.В.

Київ 2019

ЗАВДАННЯ НА РОБОТУ

1. Структура паралельної комп’ютерної системи

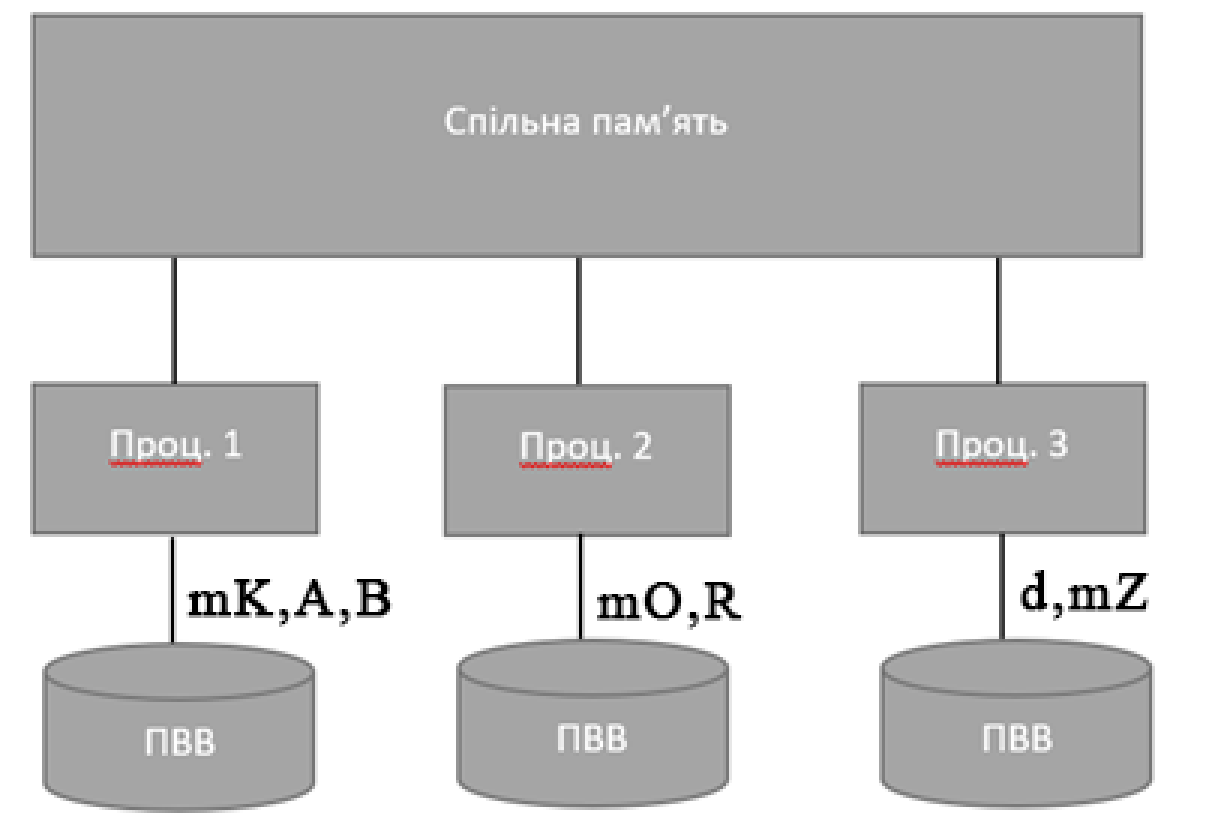


Рис. 1 – Структура ПКС СП

1. Математична задача
2. Мова програмування: С#
3. Засоби програмування взаємодії потоків (задач): class Semaphore

ВИКОНАННЯ РОБОТИ

1. Етап 1. Побудова паралельного алгоритму

Спільні ресурси: d

1. Етап 2. Розроблення алгоритмів потоків (задач)

|  |  |  |
| --- | --- | --- |
| № | Задача Т1 | Точки синхронізації |
| 1 | Введення mK,B |  |
| 2 | Сигнал задачі Т2 про введення | S2,1 |
| 3 | Чекати на завершення обчислень в Т2 | W1,2 |
| 4 | Копіювати d1:=d | Критична секція |
| 5 | Обчислення |  |
| 6 | Виведення результату А |  |

|  |  |  |
| --- | --- | --- |
| № | Задача Т2 | Точки синхронізації |
| 1 | Чекати на сигнал про введення з Т1 | W2,1 |
| 2 | Копіювати d2:=d | Критична секція |
| 3 | Введення mO,R |  |
| 4 | Сигнал задачі Т3 про введення | S3,2 |
| 5 | Чекати на введення даних в задачі Т3 | W2,3 |
| 7 | Обчислення |  |
| 8 | Сигнал Т1 про завершення обчислень | S1,2 |

|  |  |  |
| --- | --- | --- |
| № | Задача Т3 | Точки синхронізації |
| 1 | Чекати сигнал про введення з Т2 | W3,2 |
| 2 | Введення d, mZ |  |
| 3 | Копіювати d3:=d | Критична секція |
| 4 | Обчислення |  |
| 5 | Сигнал Т2 про завершення обчислень | S2,3 |

1. Етап 3. Розроблення структурної схеми взаємодії задач

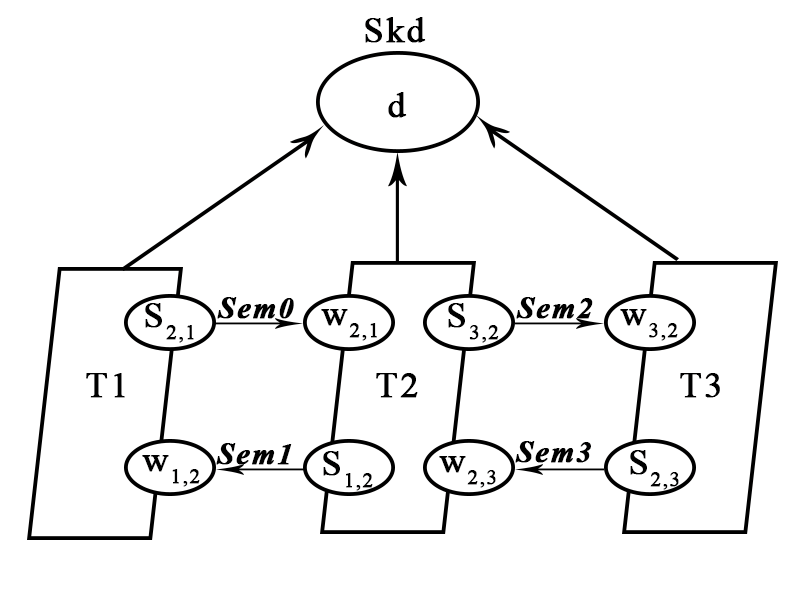


Рис. 2 – Структурна схема взаємодії задач

Призначення семафорів:

* Skd – для керування доступом до спільного ресурсу
* Sem0 – для синхронізації про завершення введення в Т1
* Sem1 – для синхронізації про завершення обчислення в Т2
* Sem2 – для синхронізації про завершення введення в Т2
* Sem3 – для синхронізації про завершення обчислень в Т3

1. Етап 4. Розроблення програми

using System;

using System.Threading;

namespace Lab1\_Semaphore

{

class Program

{

public static int N = 4;

public static int[] B, A, R = new int[N];

public static int[,] MK, MO, MZ;

public static int d, d1, d2, d3;

public static Semaphore S0 = new Semaphore(0, 1);

public static Semaphore S1 = new Semaphore(0, 1);

public static Semaphore S2 = new Semaphore(0, 1);

public static Semaphore S3 = new Semaphore(0, 1);

public static Thread T1 = new Thread(T1Part), T2 = new Thread(T2Part), T3 = new Thread(T3Part);

public static void T1Part()

{

Console.WriteLine("T1 Started");

A = new int[N];

for (int i = 0; i < A.Length; i++)

{

A[i] = 0;

}

MK = new int[N, N];

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

MK[i, j] = 1;

}

}

B = new int[N];

for (int i = 0; i < B.Length; i++)

{

B[i] = 1;

}

Console.WriteLine("T1 END INPUT");

S0.Release();

S1.WaitOne();

Console.WriteLine("T1 Calculate");

Thread.BeginCriticalRegion();

d1 = d;

Thread.EndCriticalRegion();

int[] mul\_MO\_B = Mul(MO, B);

int[] mul\_MK\_R = Mul(MK, R);

int[] res1 = Mul(MZ, mul\_MO\_B);

int[] res2 = new int[mul\_MK\_R.Length];

for (int i = 0; i < mul\_MK\_R.Length / 2; i++)

{

res2[i] = mul\_MK\_R[i] \* d1;

}

int[] result = Sub(res1, res2);

for (int i = 0; i < N / 2; i++)

{

A[i] = result[i];

}

Console.WriteLine("T1 FINISHED");

Console.WriteLine("RESULT:");

for (int i = 0; i < A.Length; i++)

{

Console.WriteLine(A[i]);

}

Console.WriteLine("END");

}

public static void T2Part()

{

Console.WriteLine("T2 Started");

S0.WaitOne();

Thread.BeginCriticalRegion();

d2 = d;

Thread.EndCriticalRegion();

MO = new int[N, N];

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

MO[i, j] = 1;

}

}

R = new int[N];

for (int i = 0; i < R.Length; i++)

{

R[i] = 1;

}

Console.WriteLine("T2 END INPUT");

S2.Release();

S3.WaitOne();

Console.WriteLine("T2 Calculate");

int[] mul\_MO\_B = Mul(MO, B);

int[] mul\_MK\_R = Mul(MK, R);

int[] res1 = Mul(MZ, mul\_MO\_B);

int[] res2 = new int[mul\_MK\_R.Length];

for (int i = 0; i < mul\_MK\_R.Length / 2; i++)

{

res2[i] = mul\_MK\_R[i] \* d2;

}

int[] result = Sub(res1, res2);

for (int i = 0; i < A.Length / 2; i++)

{

A[i] = result[i];

}

Console.WriteLine("T2 FINISHED");

S1.Release();

}

public static void T3Part()

{

Console.WriteLine("T3 Started");

S2.WaitOne();

MZ = new int[N, N];

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

MZ[i, j] = 1;

}

}

d = 1;

Console.WriteLine("T3 END INPUT");

Thread.BeginCriticalRegion();

d3 = d;

Thread.EndCriticalRegion();

Console.WriteLine("T3 Calculate");

int[] mul\_MO\_B = Mul(MO, B);

int[] mul\_MK\_R = Mul(MK, R);

int[] res1 = Mul(MZ, mul\_MO\_B);

int[] res2 = new int[mul\_MK\_R.Length];

for (int i = 0; i < mul\_MK\_R.Length / 2; i++)

{

res2[i] = mul\_MK\_R[i] \* d3;

}

int[] result = Sub(res1, res2);

for (int i = 0; i < A.Length / 2; i++)

{

A[i] = result[i];

}

Console.WriteLine("T3 FINISHED");

S3.Release();

}

static int[] Mul(int[,] a, int[] b)

{

int[] r = new int[a.GetLength(0)];

for (int i = 0; i < a.GetLength(0); i++)

{

for (int j = 0; j < b.Length; j++)

{

r[i] += a[i, j] \* b[j];

}

}

return r;

}

static int[] Sub(int[] a, int[] b)

{

int[] res = new int[a.Length];

for (int i = 0; i < a.Length; i++)

{

res[i] = a[i] - b[i];

}

return res;

}

static void Main(string[] args)

{

Console.WriteLine("PROGRAM STARTED");

T1.Start();

T2.Start();

T3.Start();

Console.ReadKey();

}

}

}