



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

ЛАБОРАТОРНА РОБОТА №3
з дисципліни «Паралельні та розподілені обчислення
Паралельне програмування-2»
Тема: «С#. Семафори, мютекси, події, критичні секції»

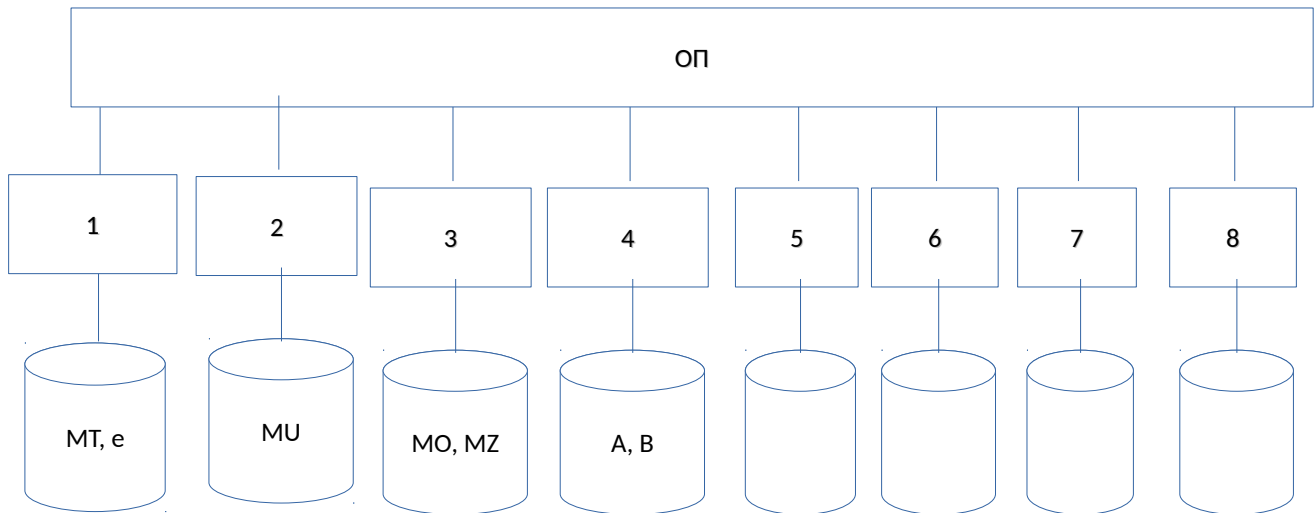
Виконав:
студент 3-го курсу
групи ІІІ-42
з номер заліковки 4206
Дзюба Влад

Завдання

Мета роботи: розробка програми для ПКС зі СП

Мова програмування: C#

Засоби організації взаємодії процесів: семафори, мютекси, події, критичні секції мови C#



Варіант

$$A = (B \cdot MO) (MZ \cdot MT + e \cdot MU)$$

Математичний паралельний алгоритм:

1. $D_h = B \cdot MO_h$
 2. $A_h = D \cdot (MZ \cdot MT_h + e \cdot MU_h)$
- CP: B, MZ, e, D.

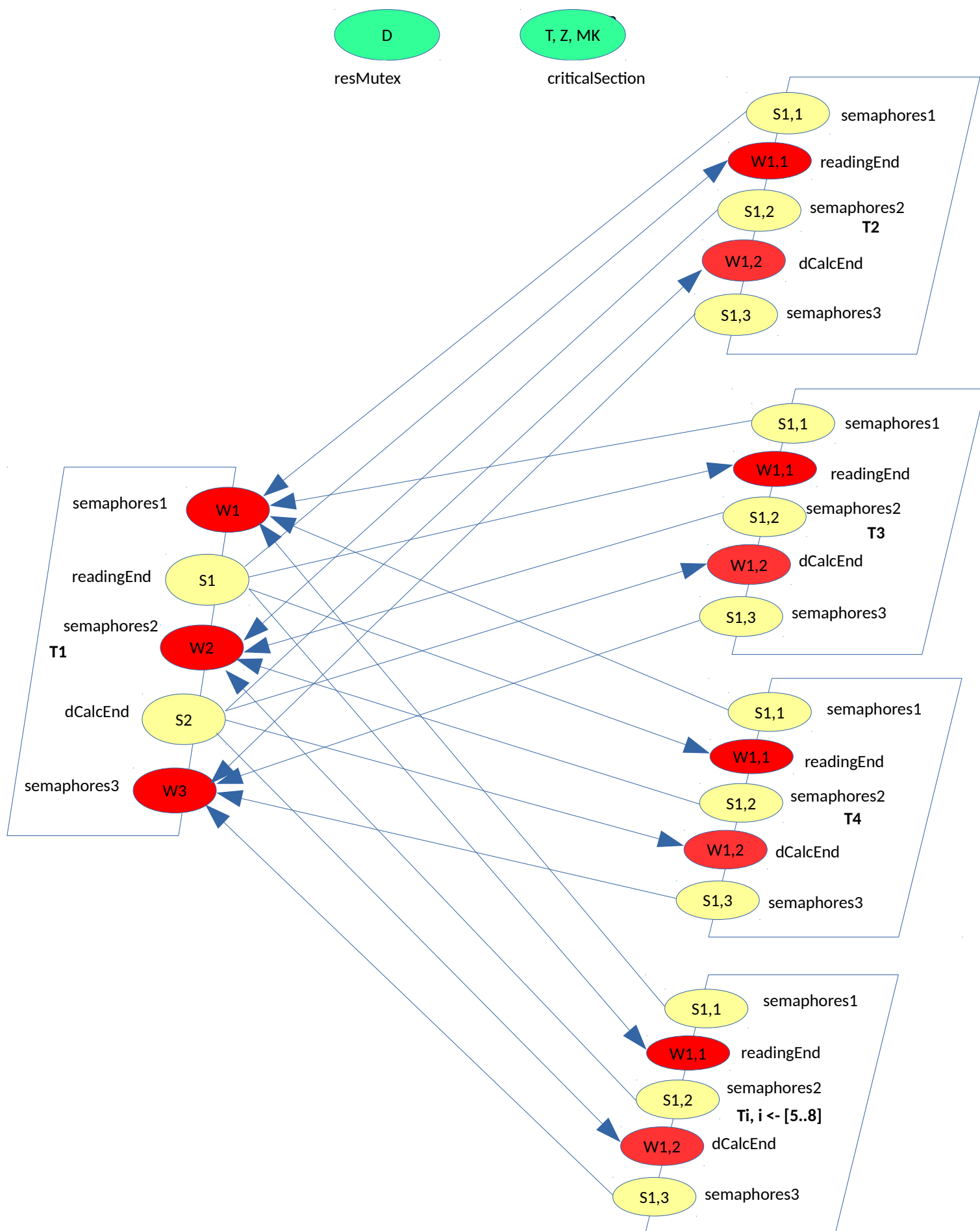
Алгоритм для кожного процесу:

T_1 : 1. Ввести MT, e . 2. Чекати сигнал про завершення вводу з T_2 . 3. Чекати сигнал про завершення вводу з T_3 . 4. Чекати сигнал про завершення вводу з T_4 . 5. Сигнал про початок обрахунку D до T_2 . 6. Сигнал про початок обрахунку D до T_3 . 7. Сигнал про початок обрахунку D до T_4 . 8. Копіювати $B_1 = B$. 9. Обрахувати $D_h = B_1 \cdot MO_h$. 10. Чекати сигнал про завершення обрахунку D_h з T_2 . 11. Чекати сигнал про завершення обрахунку D_h з T_3 . 12. Чекати сигнал про завершення обрахунку D_h з T_4 . 13. Сигнал про початок обрахунку A до T_2 . 14. Сигнал про початок обрахунку A до T_3 . 15. Сигнал про початок обрахунку A до T_4 . 16. Копіювати $D_1 = D, MZ_1 = MZ, e_1 = e$. 17. Обрахувати $A_h = D_1 \cdot (MZ_1 \cdot MT_h + e_1 \cdot MU_h)$. 18. Чекати сигнал про завершення обрахунку A_h з T_2 . 19. Чекати сигнал про завершення обрахунку з A_h T_3 . 20. Чекати сигнал про завершення обрахунку з A_h T_4 . 21. Вивести A .	 $W_{2,1}$ $W_{3,1}$ $W_{4,1}$ $S_{2,1}$ $S_{3,1}$ $S_{4,1}$ КУ1 $W_{2,2}$ $W_{3,2}$ $W_{4,2}$ $S_{2,2}$ $S_{3,2}$ $S_{4,2}$ КУ2 $W_{2,3}$ $W_{3,3}$ $W_{4,3}$	T_2 : 1. Ввести MU . 2. Сигнал про завершення вводу до T_1 . 3. Чекати сигнал про початок обрахунку D від T_1 . 4. Копіювати $B_2 = B$. 5. Обрахувати $D_h = B_2 \cdot MO_h$. 6. Сигнал про завершення обрахунку D_h до T_1 . 7. Чекати сигнал про початок обрахунку A від T_1 . 8. Копіювати $D_2 = D, MZ_2 = MZ, e_2 = e$. 9. Обрахувати $A_h = D_2 \cdot (MZ_2 \cdot MT_h + e_2 \cdot MU_h)$. 10. Сигнал про завершення обрахунку A_h до T_1 .	 $S_{1,1}$ $W_{1,1}$ КУ1 $S_{1,2}$ $W_{1,2}$ КУ2 $S_{1,3}$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------

T_3 : <ol style="list-style-type: none"> 1. Ввести MU. 2. Сигнал про завершення вводу до T_1 . 3. Чекати сигнал про початок обрахунку D від T_1 . 4. Копіювати $B_3=B$. 5. Обрахувати $D_h=B_3 \cdot MO_h$. 6. Сигнал про завершення обрахунку D_h до T_1 . 7. Чекати сигнал про початок обрахунку A від T_1 . 8. Копіювати $D_3=D, MZ_3=MZ, e_3=e$. 9. Обрахувати $A_h=D_3 \cdot (MZ_3 \cdot MT_h + e_3 \cdot MU_h)$ 10. Сигнал про завершення обрахунку A_h до T_1 . $T_i, i \in [5..8]$: <ol style="list-style-type: none"> 1. Чекати сигнал про початок обрахунку D від T_1 . 2. Копіювати $B_i=B$. 3. Обрахувати $D_h=B_i \cdot MO_h$. 4. Сигнал про завершення обрахунку D_h до T_1 . 5. Чекати сигнал про початок обрахунку A від T_1 . 6. Копіювати $D_i=D, MZ_i=MZ, e_i=e$. 7. Обрахувати $A_h=D_i \cdot (MZ_i \cdot MT_h + e_i \cdot MU_h)$ 8. Сигнал про завершення обрахунку A_h до T_1 . 	$S_{1,1}$ $W_{1,1}$ КУ1 $S_{1,2}$ $W_{1,2}$ КУ2 $S_{1,3}$	T_4 : <ol style="list-style-type: none"> 1. Ввести MU. 2. Сигнал про завершення вводу до T_1 . 3. Чекати сигнал про початок обрахунку D від T_1 . 4. Копіювати $B_4=B$. 5. Обрахувати $D_h=B_4 \cdot MO_h$. 6. Сигнал про завершення обрахунку D_h до T_1 . 7. Чекати сигнал про початок обрахунку A від T_1 . 8. Копіювати $D_4=D, MZ_4=MZ, e_4=e$. 9. Обрахувати $A_h=D_4 \cdot (MZ_4 \cdot MT_h + e_4 \cdot MU_h)$. 10. Сигнал про завершення обрахунку A_h до T_1 . 	$S_{1,1}$ $W_{1,1}$ КУ1 $S_{1,2}$ $W_{1,2}$ КУ2 $S_{1,3}$
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------

Структурна схема взаємодії процесів

Запис « $T_i, i \leftarrow [5..8]$ » для позначення того, що даний блок та його зв'язки є ідентичними для потоків T_5, T_6, T_7 та T_8 .



Лістинг програми:

```
main.cs
/*
// main
// Author:
//     Dzyuba Vlad, IP-42
*/
using System;
using System.Threading;

namespace Main {
    public class MainClass {
        // parameters
        const int N = 8;
        const int AllP = 8;
        const int ReadP = 4;
        const int H = N / AllP;

        // calc data
        static int[,] mt, mu, mo, mz;
        static int[] b, a, d;
        static int e;

        // control objects
        static Semaphore[] semaphores1 = new Semaphore[ReadP];
        static EventWaitHandle readingEnd =
            new EventWaitHandle(false,
EventResetMode.ManualReset);
        static Mutex resMutex = new Mutex();
        static Semaphore[] semaphores2 = new Semaphore[AllP];
        static EventWaitHandle dCalcEnd =
            new EventWaitHandle(false,
EventResetMode.ManualReset);
        static Object criticalSection = new Object();
        static Semaphore[] semaphores3 = new Semaphore[AllP];

        public static void Main(string[] args) {
            Thread[] threads = new Thread[AllP];
            ReadFunc[] readFuncs = new ReadFunc[ReadP] {
                new T1Read(), new T2Read(), new T3Read(), new
T4Read()
            };
            for (int i = 0; i < ReadP; i++) {
                semaphores1[i] = new Semaphore(0, 1);
                threads[i] = new Thread(new
ThreadCalcWithRead(readFuncs[i], i).Run);
            }
            for (int i = ReadP; i < AllP; i++) {
                threads[i] = new Thread(new ThreadCalc(i).Run);
            }
            for (int i = 0; i < AllP; i++) {
                semaphores2[i] = new Semaphore(0, 1);
                semaphores3[i] = new Semaphore(0, 1);
            }
        }
    }
}
```

```

        threads[i].Start();
    }
    for (int i = 0; i < ReadP; i++) {
        semaphores1[i].WaitOne();
        semaphores1[i] = null;
    }
    d = new int[N];
    readingEnd.Set();
    for (int i = 0; i < AllP; i++) {
        semaphores2[i].WaitOne();
        semaphores2[i] = null;
    }
    a = new int[N];
    dCalcEnd.Set();
    for (int i = 0; i < AllP; i++) {
        semaphores3[i].WaitOne();
        semaphores3[i] = null;
    }
    if (N < 20) {
        for (int i = 0; i < N; i++) {
            Console.Write(a[i]);
            Console.Write(" ");
        }
        Console.WriteLine();
    }
}

class ThreadCalc {
    protected int partIndex;

    public ThreadCalc(int partIndex) {
        this.partIndex = partIndex;
    }

    public virtual void Run() {
        readingEnd.WaitOne();

        int[] bi;
        resMutex.WaitOne();
        bi = b;
        resMutex.ReleaseMutex();

        for (int i = H * partIndex; i < H * (partIndex
+ 1); i++) {
            d[i] = 0;
            for (int j = 0; j < N; j++) {
                d[i] += bi[j] * mo[j, i];
            }
        }

        semaphores2[partIndex].Release();
        dCalcEnd.WaitOne();

        int[,] mzi;

```

```

        int[] di;
        int ei;
        lock (criticalSection) {
            ei = e;
            di = d;
            mzi = mz;
        }

        for (int i = H * partIndex; i < H * (partIndex
+ 1); i++) {
            a[i] = 0;
            for (int j = 0; j < N; j++) {
                int zt = ei * mu[j, i];
                for (int k = 0; k < N; k++) {
                    zt += mzi[j, k] * mt[k, i];
                }
                a[i] += di[i] * zt;
            }
        }

        semaphores3[partIndex].Release();
    }
}

class ThreadCalcWithRead : ThreadCalc {
    ReadFunc readFunc;
    public ThreadCalcWithRead( ReadFunc readFunc
, int partIndex) : base(partIndex) {
        this.readFunc = readFunc;
    }

    public override void Run() {
        readFunc.Read();
        semaphores1[partIndex].Release();
        base.Run();
    }
}

interface ReadFunc {
    void Read();
}

class T1Read : ReadFunc {
    public void Read() {
        mt = new int[N, N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                mt[i, j] = 1;
            }
        }
        e = 1;
    }
}

```



```

class T2Read : ReadFunc {
    public void Read() {
        mu = new int[N, N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                mu[i, j] = 1;
            }
        }
    }
}

class T3Read : ReadFunc {
    public void Read() {
        mo = new int[N, N];
        mz = new int[N, N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                mo[i, j] = 1;
                mz[i, j] = 1;
            }
        }
    }
}

class T4Read : ReadFunc {
    public void Read() {
        b = new int[N];
        for (int i = 0; i < N; i++) {
            b[i] = 1;
        }
    }
}

}

```