



Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

**ЛАБОРАТОРНА РОБОТА №3**  
**З ДИСЦИПЛІНИ “ОСНОВИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ ”**  
**НА ТЕМУ: “ПОТОКИ В МОВІ C#”**

**Виконав:**

Студент III курсу ФІОТ  
групи ІО-82  
Шендріков Євгеній  
Номер у списку - 25

**Перевірив:**

Доцент  
Корочкін О. В.

## Мета

Вивчення засобів мови C# для роботи з потоками.

## Завдання

Розробити програму яка містить *паралельні потоки* (задачі), кожна із яких реалізовує функцію F1, F2, F3 із Додатку Б згідно отриманому варіанту.

Вимоги що до створення потоків і завдання дослідження особливості виконання паралельної програми визначені в лабораторній роботі 1.

## Варіант завдання

		<i>F1</i>	<i>F2</i>	<i>F3</i>	
25	Шендріков Євгеній Олександрович	1.21	2.27	3.30	

$$1.21 \quad D = \text{SORT}(A) + \text{SORT}(B) + \text{SORT}(C) * (MA * ME)$$

$$2.27 \quad MF = (MG * MH) * \text{TRANS}(MK)$$

$$3.30 \quad S = (MO * MP) * V + t * MR * (O + P)$$

## Висновки

Під час виконання лабораторної роботи проблемою при тестуванні програмної частини, як і в перших лабораторних, стало введення початкових даних з консолі.

Для вирішення цієї проблеми необхідно блокувати процес, який звертається до спільного ресурсу, який в цей момент вже використовується іншим процесом. А розблокування процесу відбувається одразу ж після звільнення спільного ресурсу.

Для реалізації цього підходу в даній лабораторній роботі було використано семафори, щоб організувати взаємодію потоків для рішення задачі взаємного виключення до спільного ресурсу.

Семафори в C# – це конструкція синхронізації потоків, яка контролює доступ до загального ресурсу за допомогою лічильників. Загалом, щоб використовувати семафор, потік, якому потрібен доступ до загального ресурсу, намагається отримати дозвіл. Якщо значення лічильника більше нуля, то потік отримує доступ до ресурсу, при цьому лічильник зменшується на одиницю. Після закінчення роботи з ресурсом потік звільняє семафор, і лічильник збільшується на одиницю. Якщо ж лічильник дорівнює нулю, то потік блокується і чекає, поки не отримає дозвіл від семафора.

При  $N > 10$  можливість вводити дані з клавіатури не передбачена.

Отже, в даній лабораторній роботі було розглянуто принцип побудови паралельних програм на основі мови C#. Було використано семафори для організації взаємодії потоків задля рішення задачі взаємного виключення до спільного ресурсу, яким є клавіатура.

Було зроблено аналіз проблем, які виникли під час виконання, а також запропоновано шляхи їх рішення, перевірено працездатність програми і програмним шляхом оброблено виключні ситуації.

Кінцева мета роботи досягнута.

+

```

    "!!! Note that if the value of N > 10 -> the result will not be displayed
    "!!! If you enter N <= 0 - execution will be terminated !!!\n\n" + "Enter N:
    ");

    // check for int value of N, else N = 3
    try
    {
        N = Convert.ToInt32(Console.ReadLine());
    }
    catch (FormatException)
    {
        Console.WriteLine("\n!!! You should enter data of type int, N will be
taken as 3 !!!\n");
        N = 3;
    }

    // check for positive value of N
    if (N <= 0) throw new ArithmeticException("Restart the program and enter N >
0.");

    // if N > 10 - input from the keyboard denied
    if (N > 10)
    {
        throw new ArgumentException("If you want to enter a value from the
keyboard - enter N <= 10.");
    }

    Console.WriteLine("\n!!! Enter All Values From The Keyboard !!!");

    //----- Main Body -----
    Console.WriteLine("\nLab3 started!\n");

    Thread T1 = new Thread(new F1().Run);
    Thread T2 = new Thread(new F2().Run);
    Thread T3 = new Thread(new F3().Run);

    T1.Priority = ThreadPriority.Normal;
    T2.Priority = ThreadPriority.Lowest;
    T3.Priority = ThreadPriority.Highest;

    T1.Start();
    T2.Start();
    T3.Start();

    T1.Join();
    T2.Join();
    T3.Join();

    Thread.Sleep(100);
    Console.WriteLine("Lab 3 finished.\n");
    Console.Write("Press any key to end the program...");
    Console.ReadKey();
}
}
}
}
}

```

### F1.cs

```

using System;
using System.Threading;
using static Lab3.Lab3;

namespace Lab3
{
    public class F1

```

```

{
    // D = SORT(A)+SORT(B)+SORT(C)*(MA*ME)
    public void Run()
    {
        Console.WriteLine("T1 started.");
        Data data = new Data(N);

        int[] A, B, C;
        int[ , ] MA, ME;

        // Generate Input Values
        Thread.Sleep(50);
        Console.WriteLine("T1 is waiting for a permit.");

        Lab3.sem.WaitOne();
        Thread.Sleep(100);
        Console.WriteLine("\nT1 gets a permit.\n");

        A = data.VectorInput('A');
        B = data.VectorInput('B');
        C = data.VectorInput('C');
        MA = data.MatrixInput("MA");
        ME = data.MatrixInput("ME");

        Console.WriteLine("\nT1 releases the permit.");
        Lab3.sem.Release();
        Console.WriteLine("\nT1 is waiting for a permit.");

        // Calculate The Result
        int[] result = data.Func1(A, B, C, MA, ME);
        Thread.Sleep(100);

        // Output
        Lab3.sem.WaitOne();

        Console.WriteLine("T1 gets a permit.\n");
        Console.WriteLine("T1 result:\n");
        data.VectorOutput(result, 'D');

        Console.WriteLine("T1 releases the permit.");
        Lab3.sem.Release();

        Console.WriteLine("T1 finished.\n");
    }
}

```

## F2.cs

```

using System;
using System.Threading;
using static Lab3.Lab3;

namespace Lab3
{
    public class F2
    {
        // MF = (MG*MH)*TRANS(MK)
        public void Run()
        {
            Console.WriteLine("T2 started.");
            Data data = new Data(N);

            int[ , ] MG, MH, MK;

```

```

        // Generate Input Values
        Thread.Sleep(50);
        Console.WriteLine("T2 is waiting for a permit.");

        Lab3.sem.WaitOne();
        Thread.Sleep(100);
        Console.WriteLine("\nT2 gets a permit.\n");

        MG = data.MatrixInput("MG");
        MH = data.MatrixInput("MH");
        MK = data.MatrixInput("MK");

        Console.WriteLine("\nT2 releases the permit.");
        Lab3.sem.Release();
        Console.WriteLine("\nT2 is waiting for a permit.");

        // Calculate The Result
        int[ , ] result = data.Func2(MG, MH, MK);
        Thread.Sleep(100);

        // Output
        Lab3.sem.WaitOne();

        Console.WriteLine("T2 gets a permit.\n");
        Console.WriteLine("T2 result:\n");
        data.MatrixOutput(result, "MF");

        Console.WriteLine("T2 releases the permit.");
        Lab3.sem.Release();

        Console.WriteLine("T2 finished.\n");
    }
}
}

```

### F3.cs

```

using System;
using System.Threading;
using static Lab3.Lab3;

namespace Lab3
{
    public class F3
    {
        //  $S = (M_0 * M_P) * V + t * M_R * (O + P)$ 
        public void Run()
        {
            Console.WriteLine("T3 started.");
            Data data = new Data(N);

            int t;
            int[] V, O, P;
            int[ , ] M0, MP, MR;

            // Generate Input Values
            Thread.Sleep(50);
            Console.WriteLine("T3 is waiting for a permit.");

            Lab3.sem.WaitOne();
            Thread.Sleep(100);
            Console.WriteLine("\nT3 gets a permit.\n");

            t = data.NumInput('t');
            V = data.VectorInput('V'); O = data.VectorInput('O'); P = data.VectorInput('P');

```

```

        MO = data.MatrixInput("MO"); MP = data.MatrixInput("MP"); MR =
data.MatrixInput("MR");

        Console.WriteLine("\nT3 releases the permit.");
        Lab3.sem.Release();
        Console.WriteLine("\nT3 is waiting for a permit.");

        // Calculate The Result
        int[] result = data.Func3(t, V, O, P, MO, MP, MR);
        Thread.Sleep(100);

        // Output
        Lab3.sem.WaitOne();

        Console.WriteLine("T3 gets a permit.\n");
        Console.WriteLine("T3 result:\n");
        data.VectorOutput(result, 'S');

        Console.WriteLine("T3 releases the permit.");
        Lab3.sem.Release();

        Console.WriteLine("T3 finished.\n");
    }
}
}

```

## Data.cs

```

using System;
using static Lab3.Lab3;

namespace Lab3
{
    class Data
    {
        private int N;

        public Data(int N)
        {
            this.N = N;
        }

        // ----- Fill Matrix/Vector With Specific Number -----
        public int [ , ] FillMatrixWithNumber(int number)
        {
            int [ , ] MA = new int[N, N];
            for (int i = 0; i < N; i++)
            {
                for (int j = 0; j < N; j++)
                {
                    MA[i, j] = number;
                }
            }
            return MA;
        }

        public int[] FillVectorWithNumber(int number)
        {
            int[] A = new int[N];
            for (int i = 0; i < N; i++)
            {
                A[i] = number;
            }
            return A;
        }
    }
}

```

```

    }

    // ----- Data Entry Handler For Matrices, Vectors And Numbers -----
    public int [ , ] MatrixInput(String name)
    {
        Console.WriteLine("Enter the " + N * N + " elements of the Matrix " + name +
":");

        int[ , ] MA = new int[N, N];

        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                Console.Write(name + "[" + i + "]" + "[" + j + "] = ");
                MA[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        return MA;
    }

    public int[] VectorInput(char name)
    {
        Console.WriteLine("Enter the " + N + " elements of the Vector " + name + ":");
        int[] input = new int[N];

        for (int i = 0; i < N; i++)
        {
            Console.Write(name + "[" + i + "] = ");
            input[i] = Convert.ToInt32(Console.ReadLine());
        }
        return input;
    }

    public int NumInput(char name)
    {
        Console.Write("Enter number " + name + " = ");
        return Convert.ToInt32(Console.ReadLine());
    }

    // ----- Print Matrix, Vector And Number Into Console -----
    public void MatrixOutput(int[ , ] MA, String name)
    {
        Console.WriteLine("\tMatrix " + name + ":");
        for (int i = 0; i < N; i++)
        {
            Console.Write("\t\t");
            for (int j = 0; j < N; j++)
            {
                Console.Write(string.Format("{0} ", MA[i, j]));
            }
            Console.WriteLine();
        }
    }

    public void VectorOutput(int[] A, char name)
    {
        Console.Write("\tVector " + name + ": ");
        for (int i = 0; i < N; i++)
        {
            Console.Write(string.Format("{0} ", A[i]));
        }
        Console.WriteLine();
    }

    public void NumOutput(int a, char name)
    {

```



```

        Console.WriteLine("\tNumber " + name + ": " + a + "\n");
    }

    // Sort Vector
    public int[] SortVector(int[] A)
    {
        Array.Sort(A);
        return A;
    }

    // Calculates Sum Of 2 Vectors
    public int[] SumVectors(int[] A, int[] B)
    {
        int[] C = new int[N];
        for (int i = 0; i < N; i++)
        {
            C[i] = A[i] + B[i];
        }
        return C;
    }

    // Transposing Matrix
    public int[,] MatrixTransp(int[,] MA)
    {
        int buf;
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j <= i; j++)
            {
                buf = MA[i, j];
                MA[i, j] = MA[j, i];
                MA[j, i] = buf;
            }
        }
        return MA;
    }

    // Multiply 2 Matrices
    public int[,] MatrixMult(int[,] MA, int[,] MB)
    {
        int[,] MC = new int[N, N];
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {
                for (int k = 0; k < N; k++)
                {
                    MC[i, j] += MA[i, k] * MB[k, j];
                }
            }
        }
        return MC;
    }

    // Multiply Matrix And Vector
    public int[] VectorMatrixMult(int[] A, int[,] MA)
    {
        int[] B = new int[N];
        for (int i = 0; i < N; i++)
        {
            for (int j = 0; j < N; j++)
            {

```

```

        B[i] += A[j] * MA[i , j];
    }
    }
    return B;
}

// Multiply Integer And Matrix
public int[] IntVectorMult(int a, int[] A)
{
    int[] B = new int[N];
    for (int i = 0; i < N; i++)
    {
        B[i] = a * A[i];
    }
    return B;
}

// F1 -> D = SORT(A)+SORT(B)+SORT(C)*(MA*ME)
public int[] Func1(int[] A, int[] B, int[] C, int[ , ] MA, int[ , ] ME)
{
    return SumVectors(SumVectors(SortVector(A), SortVector(B)),
        VectorMatrixMult(SortVector(C), MatrixMult(MA, ME)));
}

// F2 -> MF = (MG*MH)*TRANS(MK)
public int[ , ] Func2(int[ , ] MG, int[ , ] MH, int[ , ] MK)
{
    return MatrixMult(MatrixMult(MG, MH), MatrixTransp(MK));
}

// F3 -> S = (MO*MP)*V+t*MR*(O+P)
public int[] Func3(int t, int[] V, int[] O, int[] P, int[ , ] MO, int[ , ] MP, int[ , ] MR)
{
    return SumVectors((VectorMatrixMult(V, MatrixMult(MO, MP))),
        IntVectorMult(t, VectorMatrixMult(SumVectors(O, P), MR)));
}
}

```

### Приклад роботи програми

```

-----
| Function 1 | D = SORT(A)+SORT(B)+SORT(C)*(MA*ME) |
| Function 2 |      MF = (MG*MH)*TRANS(MK)           |
| Function 3 |      S = (MO*MP)*V+t*MR*(O+P)         |
-----

```

!!! Note that if the value of N > 10 -> the result will not be displayed !!!  
 !!! If you enter N <= 0 - execution will be terminated !!!

Enter N: 2

!!! Enter All Values From The Keyboard !!!

Lab3 started!

T1 started.

T3 started.

T2 started.

T1 is waiting for a permit.

T3 is waiting for a permit.

T2 is waiting for a permit.

T1 gets a permit.

Enter the 2 elements of the Vector A:

A[0] = 1

A[1] = 1

Enter the 2 elements of the Vector B:

B[0] = 1

B[1] = 1

Enter the 2 elements of the Vector C:

C[0] = 1

C[1] = 1

Enter the 4 elements of the Matrix MA:

MA[0][0] = 1

MA[0][1] = 1

MA[1][0] = 1

MA[1][1] = 1

Enter the 4 elements of the Matrix ME:

ME[0][0] = 1

ME[0][1] = 1

ME[1][0] = 1

ME[1][1] = 1

T1 releases the permit.

T1 is waiting for a permit.

T3 gets a permit.

Enter number t = 3

Enter the 2 elements of the Vector V:

V[0] = 3

V[1] = 3

Enter the 2 elements of the Vector O:

O[0] = 3

O[1] = 3

Enter the 2 elements of the Vector P:

P[0] = 3

P[1] = 3

Enter the 4 elements of the Matrix MO:

MO[0][0] = 3

MO[0][1] = 3

MO[1][0] = 3

MO[1][1] = 3

Enter the 4 elements of the Matrix MP:

MP[0][0] = 3

MP[0][1] = 3

MP[1][0] = 3

MP[1][1] = 3

Enter the 4 elements of the Matrix MR:

MR[0][0] = 3

MR[0][1] = 3  
MR[1][0] = 3  
MR[1][1] = 3

T3 releases the permit.

T3 is waiting for a permit.

T2 gets a permit.

Enter the 4 elements of the Matrix MG:

MG[0][0] = 2  
MG[0][1] = 2  
MG[1][0] = 2  
MG[1][1] = 2

Enter the 4 elements of the Matrix MH:

MH[0][0] = 2  
MH[0][1] = 2  
MH[1][0] = 2  
MH[1][1] = 2

Enter the 4 elements of the Matrix MK:

MK[0][0] = 2  
MK[0][1] = 2  
MK[1][0] = 2  
MK[1][1] = 2

T2 releases the permit.

T2 is waiting for a permit.

T1 gets a permit.

T1 result:

Vector D: 6 6

T1 releases the permit.

T1 finished.

T3 gets a permit.

T3 result:

Vector S: 216 216

T3 releases the permit.

T3 finished.

T2 gets a permit.

T2 result:

Matrix MF:

32 32

32 32

T2 releases the permit.

T2 finished.

Lab 3 finished.