

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«Чувашский государственный университет И.Н. Ульянова»
Факультет информатики и вычислительной техники
Кафедра вычислительной техники

Параллельное программирование
Лабораторная работа 5
Выполнение заданий с 31 - 32

Выполнил:

Студент группы ИВТ-41-20
Галкин Д.С.

Проверил:

Ковалев С.В.

Цель работы (Технология программирования в MPI + OpenMP):

Задание 31. Программа "I am":

Задание для выполнения

Напишите программу, в которой в каждом процессе создается n нитей. Каждая нить должна выводить на экран свой номер, номер процесса родителя и общее количество нитей во всех процессах в следующем формате:

I am <Номер нити> thread from <Номер родительского процесса> process. Number of hybrid threads = <Количество нитей * Количество процессов>

Входные данные: целое число n – количество нитей, которые должны быть запущены

Выходные данные: строка вида «I am <Номер нити> thread from <Номер родительского процесса> process. Number of hybrid threads = <Количество нитей * Количество процессов>»

Входные данные	Выходные данные
3	I am 0 thread from 0 process. Number of hybrid threads = 6
	I am 1 thread from 0 process. Number of hybrid threads = 6
	I am 2 thread from 0 process. Number of hybrid threads = 6
	I am 0 thread from 1 process. Number of hybrid threads = 6
	I am 1 thread from 1 process. Number of hybrid threads = 6
	I am 2 thread from 1 process. Number of hybrid threads = 6

Полный текст программы:

1. Класс Main

```
var lab11 = new Lab11("Lab11", args);  
lab11.Start();
```

2. Класс Lab11

```
using MPI;
using ProgramMPI.Interfaces;

namespace ProgramMPI.Labs
{
    public class Lab13 : ILab
    {
        public string Name { get; set; }
        public string[] _args { get; set; }

        public Lab13(string name, string[] args)
        {
            Name = name;
            _args = args;
        }

        public void ProcessStart(string[] args)
        {
            using (new MPI.Environment(ref args))
            {
                Intracommunicator comm = Communicator.world;

                //Console.WriteLine("Введите количество нитей n:");
                //int n = int.Parse(Console.ReadLine() ?? "1");

                int n = 3;
                int numProcesses = comm.Size;
                int processRank = comm.Rank;
                int totalThreads = n * numProcesses;
                var options = new ParallelOptions { MaxDegreeOfParallelism = 1 };

                // Ограничиваем степень параллелизма до 1
                Parallel.For(0, n, options, i =>
                {
                    Print(i, processRank, totalThreads);
                });
            }
        }

        public void Start()
        {
            Console.WriteLine(this);

            ProcessStart(_args);

            Console.WriteLine("\n");
        }

        public override string ToString() => $"{Name} started classes:";

        private void Print(int index, int processRank, int totalThreads) =>
        Console.WriteLine($"I am {index} thread from {processRank} process. Number of hybrid threads = {totalThreads}.");
    }
}
```

```
}  
}
```

Результат:

```
Lab13 started classes:
```

```
I am 0 thread from 1 process. Number of hybrid threads = 6.
```

```
I am 0 thread from 0 process. Number of hybrid threads = 6.
```

```
I am 1 thread from 0 process. Number of hybrid threads = 6.
```

```
I am 1 thread from 1 process. Number of hybrid threads = 6.
```

```
I am 2 thread from 0 process. Number of hybrid threads = 6.
```

```
I am 2 thread from 1 process. Number of hybrid threads = 6.
```

Задание 14. Программа "Число π ":

Задание для выполнения

Реализуйте программу вычисления числа π (см. задание 8) с использованием MPI+OpenMP

Входные данные: одно целое число N (точность вычисления)

Выходные данные: одно вещественное число π

Входные данные	Выходные данные
1000000000	3.14159265

Полный текст программы:

1. Класс Main

```
var lab12 = new Lab12("Lab12");  
lab14.Start();
```

2. Класс Lab12

```
using MPI;
using ProgramMPI.Interfaces;

namespace ProgramMPI.Labs
{
    public class Lab14 : ILab
    {
        public string Name { get; set; }
        public string[] _args { get; set; }

        public Lab14(string name, string[] args)
        {
            Name = name;
            _args = args;
        }

        public void ProcessStart(string[] args)
        {
            using (new MPI.Environment(ref args))
            {
                Intracommunicator comm = Communicator.world;

                int N;
                if (comm.Rank == 0)
                {
                    //Console.WriteLine("Введите целое число N: ");
                    //N = int.Parse(Console.ReadLine() ?? "1000000000"); // Значение
по умолчанию, если ввод некорректен
                    N = 1000000000;
                }
                else
                {
                    N = 0;
                }

                // Рассылка значения N всем процессам
                comm.Broadcast(ref N, 0);

                double pi = CalculatePi(N, comm.Rank, comm.Size);

                // Сбор результатов вычисления  $\pi$  со всех процессов
                double[] allPi = comm.Gather(pi, 0);

                if (comm.Rank == 0)
                {
                    double finalPi = 0;
                    foreach (var p in allPi)
                    {
                        finalPi += p;
                    }
                    Print(finalPi);
                }
            }
        }
    }
}
```

```

    }

    public void Start()
    {
        Console.WriteLine(this);

        ProccessStart(_args);

        Console.WriteLine("\n");
    }

    private double CalculatePi(int n, int rank, int size)
    {
        double sum = 0.0;
        double step = 1.0 / n;
        int localN = n / size;
        int start = rank * localN;
        int end = start + localN;

        object locker = new object();

        // Использование Parallel.For для имитации OpenMP
        Parallel.For(start, end, i =>
        {
            double x = (i + 0.5) * step;
            double term = 4.0 / (1.0 + x * x);

            lock (locker)
            {
                sum += term;
            }
        });

        return sum * step;
    }

    public override string ToString() => $"{Name} started classes:";

    private void Print(double pi) => Console.WriteLine($"Result PI = {pi}");

}

```

Результат:

```

Lab14 started classes:
Result PI = 3,141592653590079

```