ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет ФИСП

Кафедра ПИ

Лабораторная работа №6

предмет: «Операционные системы»

по теме: «Межпроцессное взаимодействие в ОС UNIX»

Выполнил:

ст. гр. ПИ-19а

Саевский О.В.

Проверил:

Чернышова А.В.

Московченко А.В.

ДОНЕЦК – 2021

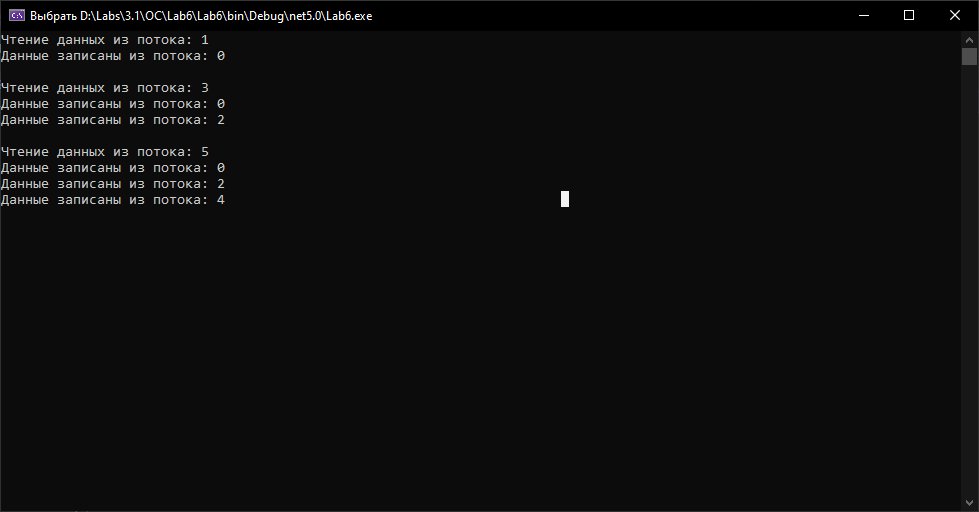
Задание:

11. Синхронизация процессов с помощью семафоров

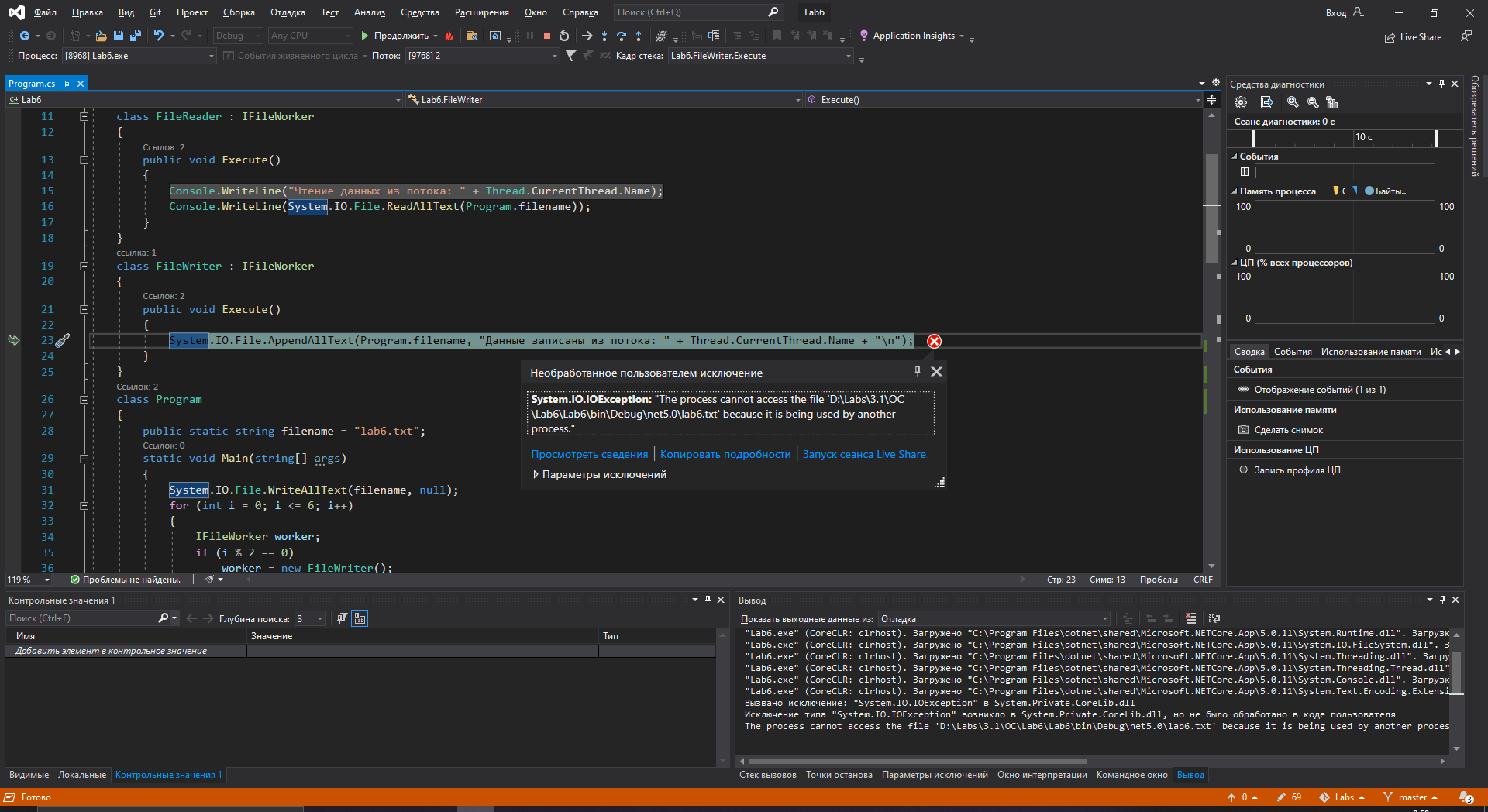
В данном примере я синхронизирую процессы чтение и записи с файла. Запускаем на 6 потоках с чередованием чтения и записи (первый поток записывает, второй считывает, третий записывает и тд.).

Результаты работы с семафорами и без:

С семафорами:



Без семафоров:



(Ошибка: файл занят другим процессом)

Таким образом процессы синхронизируется, и доступ к файлу имеется только для одного потока на время его выполнения.

Код программы (написано на C#):

using System;

using System.Threading;

namespace Lab6

{

interface IFileWorker

{

void Execute();

}

class FileReader : IFileWorker

{

public void Execute()

{

Console.WriteLine("Чтение данных из потока: " + Thread.CurrentThread.Name);

Console.WriteLine(System.IO.File.ReadAllText(Program.filename));

}

}

class FileWriter : IFileWorker

{

public void Execute()

{

System.IO.File.AppendAllText(Program.filename, "Данные записаны из потока: " + Thread.CurrentThread.Name + "\n");

}

}

class Program

{

public static string filename = "lab6.txt";

static void Main(string[] args)

{

System.IO.File.WriteAllText(filename, null);

for (int i = 0; i <= 6; i++)

{

IFileWorker worker;

if (i % 2 == 0)

worker = new FileWriter();

else worker = new FileReader();

var reader = new FileWorker(worker, i);

}

FileWorker.sem.Release(1);

}

}

class FileWorker

{

// создаем семафор

public static Semaphore sem = new Semaphore(0, 4);

Thread myThread;

public FileWorker(IFileWorker worker, int thread)

{

myThread = new Thread(Run);

myThread.Name = thread.ToString();

//Запускаем новый поток с методом Run

myThread.Start(worker);

}

public void Run(object worker)

{

//Ждем пока освободится семафор

sem.WaitOne();

((IFileWorker)worker).Execute();

Thread.Sleep(1000);

//Освобождаем семафор

sem.Release();

Thread.Sleep(1000);

}

}

}