Белорусско-Российский университет

Кафедра «ПОИТ»

Отчёт по лабораторной работе №4 по дисциплине «Операционные системы»

**Синхронизация потоков в ОС Windows**

Выполнил Проверил

Ст. гр. АСОИР-181 Преподаватель

Ковалевский Т В Зайченко Е. А.

Могилев 2021

Цель: научиться использовать функции Win32 API предназначенные для синхронизации потоков в ОС Windows.

Проанализировать работу программы:

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading;**

**namespace ThreadingWithLock**

**{**

**class Program**

**{**

**private static object syncToken = new object();**

**static void Main(string[] args)**

**{**

**int[] array = CreateArray();**

**Console.WriteLine("Исходный массив:");**

**DisplayArray(array);**

**Thread sortingThread = new Thread(SortingThreadMain);**

**Thread displayingThread = new Thread(DisplayingThreadMain);**

**sortingThread.Start(array);**

**Thread.Sleep(200);**

**displayingThread.Start(array);**

**}**

**static void SortingThreadMain(object data)**

**{**

**int[] array = data as int[];**

**if (array == null)**

**{**

**return;**

**}**

**lock (syncToken)**

**{**

**Console.WriteLine("Сортировка....");**

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

**{**

**for (int j = 0; j < array.Length - i - 1; j++)**

**{**

**int tmp;**

**if (array[j] > array[j + 1])**

**{**

**tmp = array[j];**

**array[j] = array[j + 1];**

**array[j + 1] = tmp;**

**Thread.Sleep(200);**

**}**

**}**

**}**

**}**

**}**

**private static void DisplayingThreadMain(object data)**

**{**

**int[] array = data as int[];**

**if (array == null)**

**{**

**return;**

**}**

**lock (syncToken)**

**{**

**DisplayArray(array);**

**Console.ReadLine();**

**}**

**}**

**private static int[] CreateArray()**

**{**

**Console.WriteLine("Введите размерность массива");**

**string arraySizeString = Console.ReadLine();**

**int arraySize;**

**if (!int.TryParse(arraySizeString, out arraySize))**

**{**

**arraySize = 0;**

**}**

**int[] array = new int[arraySize];**

**Random rand = new Random();**

**for (int i = 0; i < arraySize; i++)**

**{**

**array[i] = -1 \* rand.Next(99) + rand.Next(99);**

**}**

**return array;**

**}**

**private static void DisplayArray(int[] array)**

**{**

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

**{**

**Console.Write(array[i] + " ");**

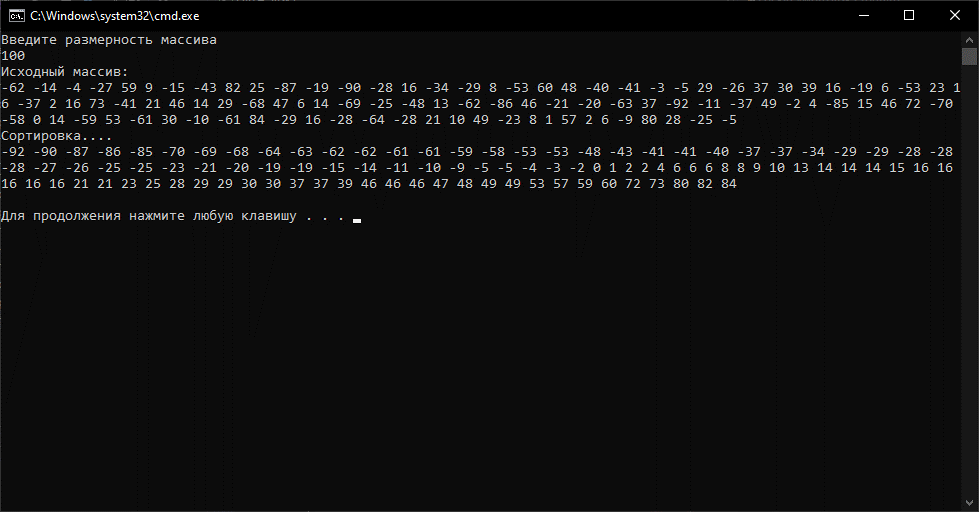
**}**

**Console.WriteLine();**

**}**

**}**

**}**



**Задание 1.**

Написать программу для консольного процесса, который состоит из потока main и двух потоков: worker1 и worker2 (см. варианты). Синхронизацию потоков выполнить с помощью оператора lock. Протестировать программу без использования lock. Объяснить результаты.

* Поток main должен выполнить следующие действия:

1. Создать массив случайных целых чисел (как положительных, так и

отрицательных), размерность которого вводится с консоли.

2. Вывести на консоль исходный массив.

3. Создать поток worker1.

4. Создать поток worker2.

5. Дождаться завершения потоков worker1 и worker2.

6. Вывести на консоль массив, полученный после обработки потоком worker1.

7. Вывести на консоль результаты работы потока worker2.

8. Завершить работу.

* Поток worker1 должен выполнить следующие действия:

1. Выполнить обработку массива (согласно варианту): Поток worker1 должен уменьшить значения четных элементов массива на 5.

2. Завершить свою работу.

* Поток worker2 должен выполнить следующие действия

1. Найти индексы минимального и максимального элементов обработанного

массива.

**using System;**

**using System.Threading;**

**namespace Lab\_4**

**{**

**class Program**

**{**

**private static object syncToken = new object();**

**static void Main(string[] args)**

**{**

**int[] array = CreateArray();**

**Console.WriteLine("Исходный массив:");**

**DisplayArray(array);**

**// Создание потоков для сортировки и отображения массива**

**Thread worker1 = new Thread(ReduceEvenNumberedElementOfArrayByFive);**

**Thread worker2 = new Thread(GetMinAndMaxIndexesOfArray);**

**worker1.Start(array);**

**Thread.Sleep(200);**

**DisplayArray(array);**

**worker2.Start(array);**

**worker1.Join();**

**worker2.Join();**

**}**

**// Точка входа потока worker1**

**static void ReduceEvenNumberedElementOfArrayByFive(object data)**

**{**

**// Проверка передачи корректных данных**

**if (!(data is int[] array))**

**{**

**return;**

**}**

**// Вход в критическую секцию**

**lock (syncToken)**

**{**

**Console.WriteLine("\nУменьшение четных элементов массива на 5...\n");**

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

**{**

**if ((i+1) % 2 == 0)**

**{**

**array[i] -= 5;**

**Thread.Sleep(200);**

**}**

**}**

**}**

**}**

**// Точка входа потока worker2**

**private static void GetMinAndMaxIndexesOfArray(object data)**

**{**

**if (!(data is int[] array))**

**{**

**return;**

**}**

**// Вход в критическую секцию**

**lock (syncToken)**

**{**

**int indexOfMax = 0, indexOfMin = 0;**

**Console.WriteLine("\nПоиск индексов минимального и максимального элементов массива...");**

**for (int i = 1; i < array.Length; i++)**

**{**

**if (array[i] > array[indexOfMax])**

**{**

**indexOfMax = i;**

**}**

**else if (array[i] < array[indexOfMin])**

**{**

**indexOfMin = i;**

**}**

**}**

**Console.WriteLine($"Индекс минимального элемента - {indexOfMin}, максимального - {indexOfMax}.");**

**}**

**}**

**private static int[] CreateArray()**

**{**

**Console.WriteLine("Введите размерность массива");**

**string arraySizeString = Console.ReadLine();**

**int arraySize;**

**if (!int.TryParse(arraySizeString, out arraySize))**

**{**

**arraySize = 0;**

**}**

**int[] array = new int[arraySize];**

**Random rand = new Random();**

**for (int i = 0; i < arraySize; i++)**

**{**

**array[i] = -1 \* rand.Next(99) + rand.Next(99);**

**}**

**return array;**

**}**

**private static void DisplayArray(int[] array)**

**{**

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

**{**

**Console.Write(array[i] + " ");**

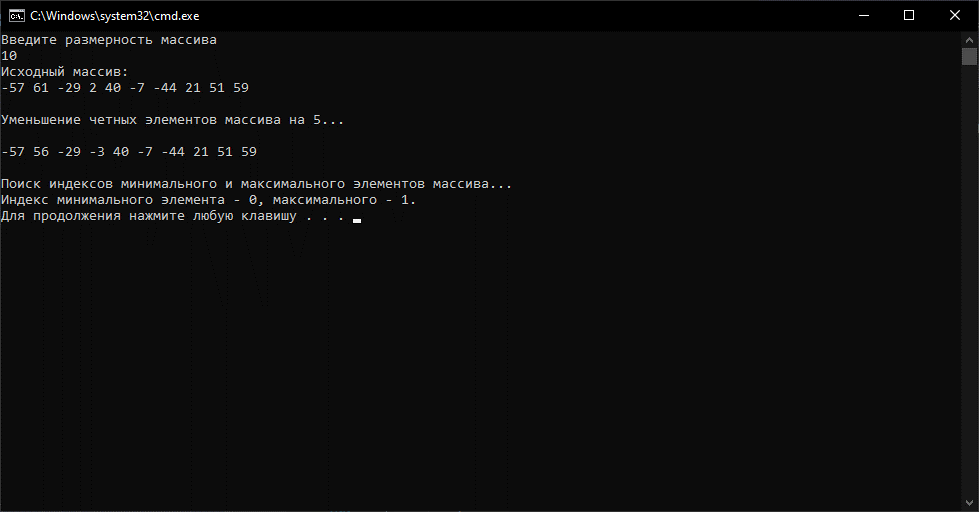
**}**

**Console.WriteLine();**

**}**

**}**

**}**



**Задание 2.**

Написать программу для консольного процесса, который состоит из потока main и двух потоков: worker1 и worker2.

* Поток main должен выполнить следующие действия:

1. Создать 2 массива случайных целых чисел (как положительных, так и

отрицательных), размерность которых вводится с консоли.

2. Создать поток worker1.

3. Создать поток worker2.

4. Вывести на консоль исходные массивы.

5. Дождаться завершения потоков worker1 и worker2.

6. Вывести на консоль значение, полученное потоком worker1.

7. Вывести на консоль результаты работы потока worker2.

8. Завершить работу.

* Поток worker1 должен выполнить следующие действия:

1. Найти сумму положительных элементов первого массива.

2. Найти сумму положительных элементов второго массива.

3. Определить, какое из найденных значений больше.

* Поток worker2 должен выполнить следующие действия

1. Найти количество положительных элементов второго массива.

2. Найти количество отрицательных элементов первого массива.

3. Определить, какое из найденных значений больше.

При разработке программы проанализируйте возможность возникновения тупиков.

**using System;**

**using System.Threading;**

**namespace Lab\_4**

**{**

**class Program**

**{**

**static readonly Arrays arrays = new Arrays();**

**static void Main(string[] args)**

**{**

**arrays.array1 = CreateArray();**

**arrays.array2 = CreateArray();**

**Thread worker1 = new Thread(Worker1);**

**Thread worker2 = new Thread(Worker2);**

**Console.WriteLine("Исходные массивы:");**

**DisplayArray(arrays.array1);**

**DisplayArray(arrays.array2);**

**worker1.Start(arrays);**

**worker2.Start(arrays);**

**worker1.Join();**

**worker2.Join();**

**}**

**static int GetSumOfPositiveElements(int[] array)**

**{**

**int sum = 0;**

**foreach (var item in array)**

**{**

**if (item > 0) sum += item;**

**}**

**return sum;**

**}**

**static int GetSumOfNegativeElements(int[] array)**

**{**

**int sum = 0;**

**foreach (var item in array)**

**{**

**if (item < 0) sum += item;**

**}**

**return sum;**

**}**

**static void Worker1(object data)**

**{**

**if (!(data is Arrays arrays))**

**{**

**return;**

**}**

**int sum1 = GetSumOfPositiveElements(arrays.array1);**

**int sum2 = GetSumOfPositiveElements(arrays.array2);**

**Console.WriteLine((sum1 > sum2) ? $"Сумма положительных элементов первого массива больше, чем второго: {sum1} > {sum2}"**

**: $"Сумма положительных элементов второго массива больше, чем первого: {sum1} < {sum2}");**

**}**

**static void Worker2(object data)**

**{**

**if (!(data is Arrays arrays))**

**{**

**return;**

**}**

**int sum1 = GetSumOfNegativeElements(arrays.array1);**

**int sum2 = GetSumOfNegativeElements(arrays.array2);**

**Console.WriteLine((sum1 > sum2) ? $"Сумма отрицательных элементов первого массива больше, чем второго: {sum1} > {sum2}"**

**: $"Сумма отрицательных элементов второго массива больше, чем первого: {sum1} < {sum2}");**

**}**

**private static int[] CreateArray()**

**{**

**Console.WriteLine("Введите размерность массива");**

**string arraySizeString = Console.ReadLine();**

**if (!int.TryParse(arraySizeString, out int arraySize))**

**{**

**arraySize = 0;**

**}**

**int[] array = new int[arraySize];**

**Random rand = new Random();**

**for (int i = 0; i < arraySize; i++)**

**{**

**array[i] = -1 \* rand.Next(99) + rand.Next(99);**

**}**

**return array;**

**}**

**private static void DisplayArray(int[] array)**

**{**

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

**{**

**Console.Write(array[i] + " ");**

**}**

**Console.WriteLine();**

**}**

**}**

**public class Arrays**

**{**

**public int[] array1;**

**public int[] array2;**

**}**

**}**

