

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

Выполнила
Радиончик С.С.,
студентка группы ПО-5

Проверил
Крощенко А.А.,
ст. преп. Кафедры ИИТ,
«__» _____ 2021 г.

Брест, 2021

Цель работы: приобрести базовые навыки работы с файловой системой в C#.

Вариант 6.

Задание 1. Напишите программу сравнения двух файлов, которая будет печатать первую строку и позицию символа, где они различаются. В противном случае должно выводиться сообщение об эквивалентности содержимого файлов.

Задание 2. Написать bat-файл для запуска.

Утилита `split` копирует и разбивает файл на отдельные файлы заданной длины. В качестве аргументов ей надо указать имя исходного файла и префикс имен выходных файлов. Если файл не задан или задан как `-`, программа читает стандартный ввод. По умолчанию размер части разбиения равен 10 строк, а префикс равен `x`. Имена выходных файлов будут состояться из этого префикса и двух дополнительных букв `aa`, `ab`, `ac` и т. д. (без пробелов и точек между префиксом и буквами). Если префикс имен файлов не задан, то по умолчанию используется `x`, так что выходные файлы будут называться `хаа`, `хаб` и т. д. Формат использования: `split [-b | -l] [-d] [входной_файл [префикс_выходных_файлов]]` где ключи имеют следующее значение:

- `-b` , `--bytes=num` Записывать в каждый выходной файл заданное число `num` байт. При задании числа байт можно использовать суффиксы: `b` означает байты, `k` – `1kb` , `m` – `1Mb`.
- `-l` , `--lines=num` Записывать в каждый выходной файл `num` строк.
- `-d` , `--numericuffixes` Использовать числовые, а не алфавитные суффиксы, начинающиеся с `00`. Суффиксы файлов будут иметь вид: `00`, `01`, `02` и т. д.

Выполнение:

Код программы.

1)

```
using System;
using System.IO;

namespace Lab2_1
{
    class Program
    {
        public void fileComparison()
        {
            try
            {
                StreamReader sr1 = new StreamReader("D:\\spp\\file1.txt");
                var lineFile1 = sr1.ReadLine();

                StreamReader sr2 = new StreamReader("D:\\spp\\file2.txt");
                var lineFile2 = sr2.ReadLine();

                bool filesEquals = true;

                while (lineFile1 != null || lineFile2 != null)
                {
                    bool nullLineEquals;
```

```

        if (lineFile1 != null)
            nullLineEquals = lineFile1.Equals(lineFile2);
        else
            nullLineEquals = lineFile2.Equals(lineFile1);

        if (!nullLineEquals)
        {
            filesEquals = false;
            Console.WriteLine("Различающиеся строки: ");
            Console.WriteLine($"{lineFile1}");
            Console.WriteLine($"{lineFile2}");
            int N = 0;

            if (lineFile1 != null && lineFile2 != null)
            {
                if (lineFile1.ToCharArray().Length >
lineFile2.ToCharArray().Length)
                    N = lineFile1.ToCharArray().Length;
                else
                    N = lineFile2.ToCharArray().Length;
            }
            else if (lineFile1 != null)
                N = lineFile1.ToCharArray().Length;
            else
                N = lineFile2.ToCharArray().Length;

            for (int i = 0; i < N; i++)
            {
                if (lineFile1 != null && lineFile2 != null)
                {
                    if (i >= lineFile1.ToCharArray().Length || i
>= lineFile2.ToCharArray().Length)
                    {
                        Console.WriteLine($"Позиция
различающегося символа в строке: {++i}");
                        break;
                    }
                    else if (lineFile1.ToCharArray()[i] !=
lineFile2.ToCharArray()[i])
                    {
                        Console.WriteLine($"Позиция
различающегося символа в строке: {++i}");
                        break;
                    }
                }
                else
                {
                    Console.WriteLine($"Позиция различающегося
символа в строке: {i}");
                    break;
                }
            }

            lineFile1 = sr1.ReadLine();
            lineFile2 = sr2.ReadLine();
        }

        sr1.Close();
        sr2.Close();
        if (filesEquals)
            Console.WriteLine("Файлы идентичны");
    }
    catch (Exception e)

```

```

        {
            Console.WriteLine("Exception: " + e.Message);
        }
    }

    static void Main(string[] args)
    {
        Program lab = new Program();
        lab.fileComparison();
    }
}

```

2)

```

using System;
using System.IO;
using System.Threading.Tasks;

namespace Lab2_2
{
    class Program
    {
        public async Task Split(string[] args)
        {
            if(args[0].Equals("split"))
            {
                int numLines = 10;
                int numBytes = -1;
                string prefix = "x";
                string fileName = "";
                string postfixValue = "abcdefghijklmnopqrstuvwxyz";

                for(int i = 1; i < args.Length; i++)
                {
                    if (args[i].Contains("-b") || args[i].Contains("--bytes"))
                    {
                        string size;
                        int result = 0;

                        if (args[i].Contains("-b"))
                            size = args[i].Replace("-b=", "");
                        else
                            size = args[i].Replace("--bytes=", "");

                        if (size.Contains("b"))
                            int.TryParse(size.Replace("b", ""), out numBytes);
                        else if (size.Contains("k"))
                        {
                            int.TryParse(size.Replace("k", ""), out result);
                            numBytes = (int)Math.Pow(10, 3) * result;
                        }
                        else if (size.Contains("m"))
                        {
                            int.TryParse(size.Replace("m", ""), out result);
                            numBytes = (int)Math.Pow(10, 6) * result;
                        }
                        else
                            int.TryParse(size, out numBytes);
                    }
                    else if(args[i].Contains("-l") || args[i].Contains("--lines"))
                    {
                        if (args[i].Contains("-l"))
                            int.TryParse(args[i].Replace("-l=", ""), out
numLines);
                    }
                }
            }
        }
    }
}

```

```

        else
            int.TryParse(args[i].Replace("--lines=", ""), out
numLines);
    }
    else if(args[i].Contains("-d") || args[i].Contains("--
numericssuffixes"))
    {
        postfixValue = "0123456789";
    }
    else
    {
        if (args[i].Contains(".txt"))
            fileName = args[i];
        else
            prefix = args[i];
    }
}

string arr = "";

if(string.IsNullOrEmpty(fileName))
{
    fileName = "x.txt";

    string str = Console.ReadLine();
    while (!str.Contains("#"))
    {
        arr += str;
        arr += "\n";
        str = Console.ReadLine();
    }
}

try
{
    if (arr.Length != 0)
    {
        var strArr = arr.Split("\n");

        using (StreamWriter sw = new StreamWriter(fileName, false,
System.Text.Encoding.Default))
        {
            for (int i = 0; i < strArr.Length; i++)
            {
                await sw.WriteLineAsync(strArr[i]);
            }
        }
    }

    if (numBytes == -1)
    {
        string line = null;
        int numFiles = -1;

        using (StreamReader sr = new StreamReader(fileName))
        {
            while (!sr.EndOfStream)
            {
                numFiles++;

                string newFileName = setFileName(prefix,
postfixValue, numFiles);

                bool isText = true;

```

```

        using (StreamWriter sw = new
StreamWriter(newFileName, false, System.Text.Encoding.Default))
        {
            for (int i = 0; i < numLines; i++)
            {
                line = sr.ReadLine();
                sw.WriteLine(line);
            }
        }
    }
}
else
{
    int numFiles = -1;
    int offset = 0;
    bool isText = true;

    using (FileStream fread = File.OpenRead(fileName))
    {
        while (isText)
        {
            numFiles++;
            string newFileName = setFileName(prefix,
postfixValue, numFiles);

            using (StreamWriter sw = new
StreamWriter(newFileName, false, System.Text.Encoding.Default))
            {
                byte[] array = new byte[numBytes];
                fread.Read(array, offset, numBytes);
                string textFromFile =

                if(textFromFile.StartsWith('\0'))
                {
                    isText = false;
                    break;
                }

                sw.Write(textFromFile);
            }
        }
    }
}

}

}
catch (FileNotFoundException e)
{
    Console.WriteLine(e.Message);
}
catch (IOException e)
{
    Console.WriteLine(e.Message);
}
}

}

public string setFileName(string prefix, string postfixValue, int numFiles)
{
    string fileName = prefix;

    int i = numFiles / postfixValue.Length;
    int j = numFiles % postfixValue.Length;

```

```

        fileName += postfixValue[i];
        fileName += postfixValue[j];

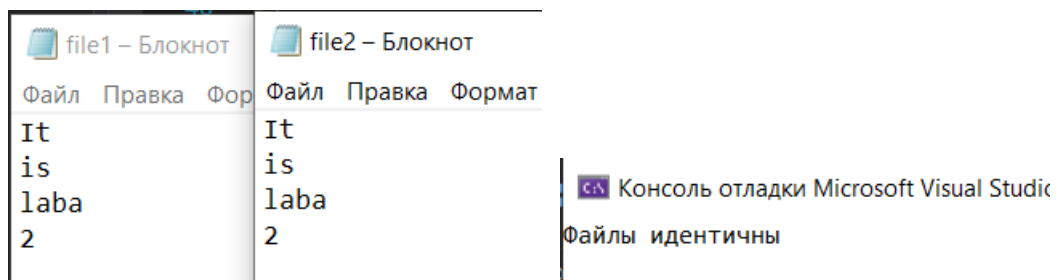
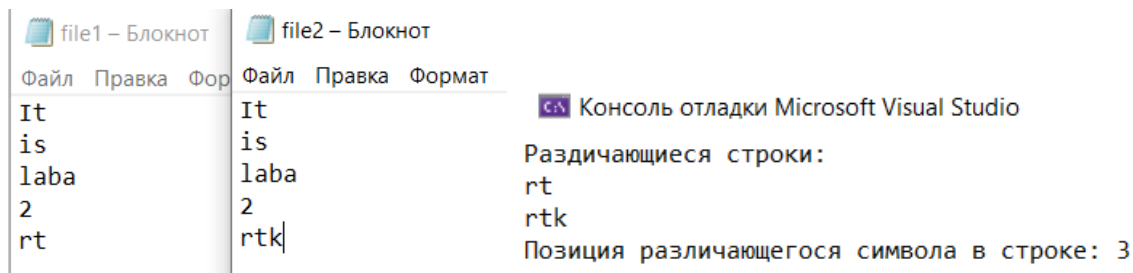
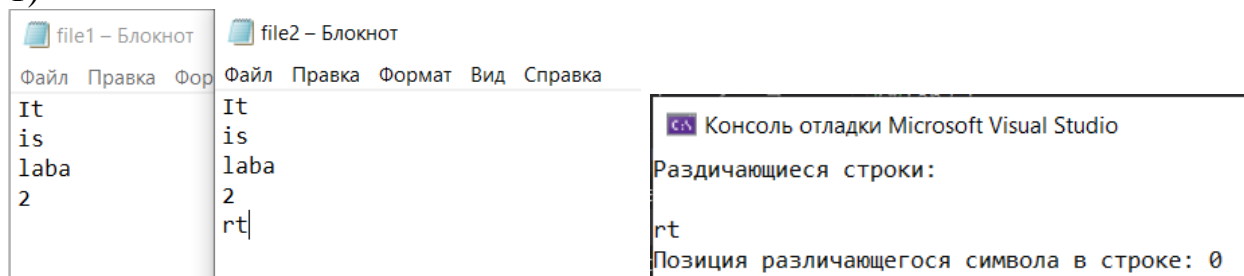
        return fileName;
    }

    static void Main(string[] args)
    {
        Program lab = new Program();
        lab.Split(args);
    }
}

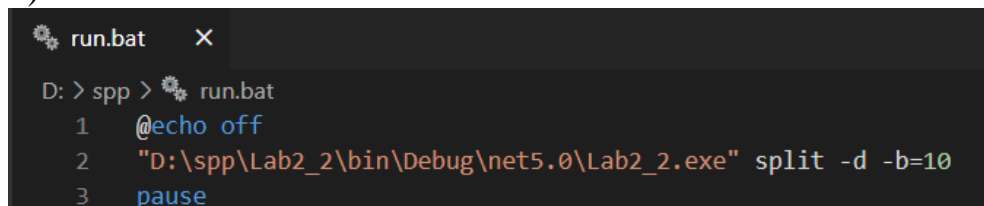
```

Результаты работы программы:

1)

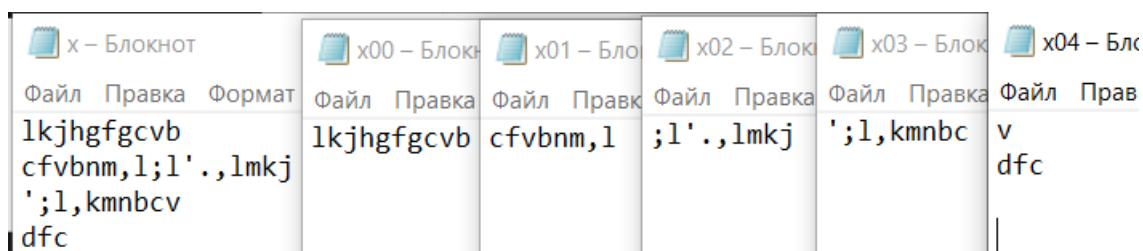


2)



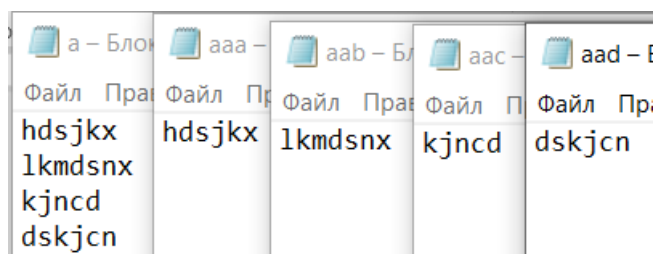
```
C:\WINDOWS\system32\cmd.exe
lkjhfgcvb
cfvbnm,1;1'. ,lmkj
';1,kmnbcv
dfc
#
```

- x
- x00
- x01
- x02
- x03
- x04



```
run.bat
D: > spp > run.bat
1 @echo off
2 "D:\spp\Lab2_2\bin\Debug\net5.0\Lab2_2.exe" split a.txt a -l=1
3 pause
4
```

- a
- aaa
- aab
- aac
- aad



Вывод: приобрела базовые навыки работы с файловой системой в C#.