

Лабораторная работа

Работа с двоичными файлами

1 Цель работы

1.1 Изучить процесс обработки двоичных файлов средствами языка C++ с использованием потоков.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва: Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.24

2.2 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.19.8-19.11.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

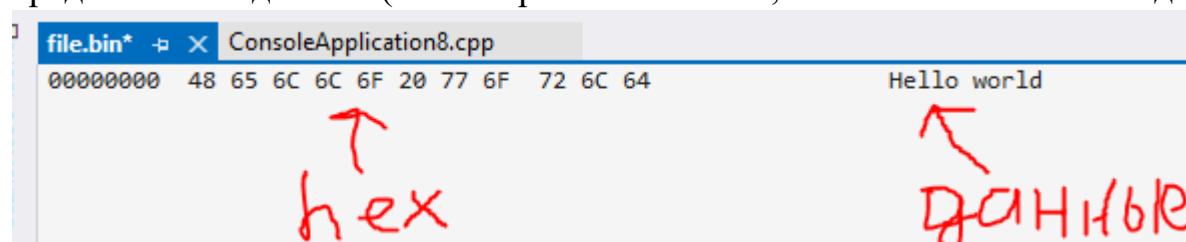
4.1 Персональный компьютер.

5 Задание

Работа производится с двоичными файлами (*.bin).

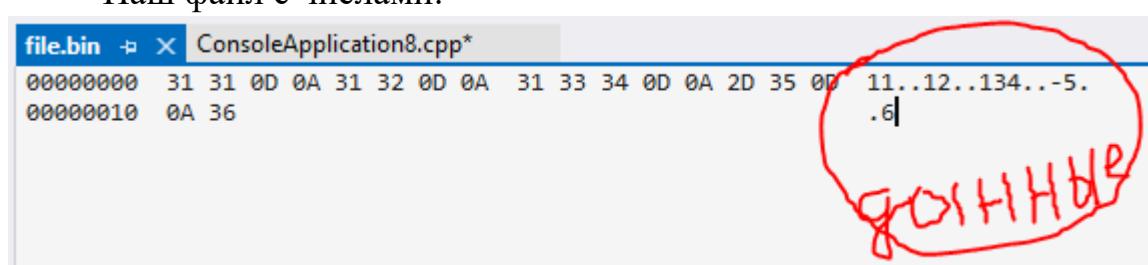
Например:

Был создан бинарный файл file.bin, его содержимое можно посмотреть в Visual Studio. Можно отредактировать нажав справа, слева находится hex-представление данных (на ПК хранятся в 0 и 1, но в hex-системе нагляднее).



Пример 1:

Наш файл с числами:



Листинг с примером кода для чтения бинарного файла:

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    //открытие на чтение file.bin в двоичном режиме
    ifstream fin("file.bin", ios::binary);
    int num;//переменная, куда будем считывать число из файла,
    //до этого считывали в строку (string)
```

```

//если файл открылся
if (fin.is_open())
{
    //считываем в переменную num числа из файла
    //на самом деле просто считываем данные до пробела или enter
    //раньше использовали getline, он считывал до конца строки (enter)
    while (fin >> num)
    {
        //выводим считанное число на консоль
        cout << num << endl;
    }
    fin.close(); //обязательно закрыть файл
}

```

Результат выполнения:

The screenshot shows two windows from Microsoft Visual Studio. The top window is the 'Output' window titled 'file.bin' with the tab 'ConsoleApplication8.cpp'. It displays binary data: address 00000000 shows 31 31 0D 0A 31 32 0D 0A followed by 31 33 34 0D 0A 2D 35 0D 11..12..134...-5.; address 00000010 shows 0A 36 followed by .6. The bottom window is the 'Консоль отладки Microsoft Visual Studio' (Debug Console) window, titled 'Immediate'. It shows the following output:
11
12
134
-5
6

5.1 Создать в проекте бинарный файл. При открытии потока на чтение или запись добавлять режим работы `ios::binary`.

5.2 Написать программу, запрашивающую у пользователя имя файла и записывающую в него n случайных целых чисел от 1 до 100 (n вводится пользователем).

5.3 Написать программу, запрашивающую у пользователя имя файла,читывающую из него целые числа до конца файла и выводящую их на экран.

5.5 Написать программу, запрашивающую у пользователя логин (хранится в файле), если в файле с логинами такой логин есть – вывести приветствие, иначе – завершить работу программы.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 В чем преимущества использования двоичных файлов?

8.2 С помощью каких функций можно записывать информацию в двоичные файлы?

8.3 С помощью каких функций можно считывать информацию из двоичных файлов?

8.4 Как считать переменные стандартных типов данных из двоичного файла?

8.5 Как считать переменные структурированных типов данных из двоичного файла?