**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

по дисциплине «Технические средства информатизации»

на тему:

«Получение информации и управление оборудованием»

Вариант №1

г. Санкт-Петербург

2018г.

Выполнил студент группы Y2338

ФИО Матвеенко Д.В.

Проверил:

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_201\_г

**Министерство образования и науки Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ»**

Факультет среднего профессионального образования

Цель работы – Получение информации и управления оборудованием через API функции операционной системы.

Текст задания: разработка программы с использованием Windows API (Linux API) определения разрешения и глубины цвета экран и расчета размера одного кадра, исходя из полученных параметров в байтах, килобайтах и мегабайтах. Информацию вывести на экран.

Порядок выполнения работы.

1. Установлена IDE Visual Studio 2017 с предустановленными расширениями, для компиляции C++ кода в консольном приложении Windows.
2. Чтение документации Windows API, про возможности функций GetSystemMetrics и GetDeviceCaps для выполнения лабораторной работы.
3. Написание C++ кода для вывода текста о разрешении монитора и глубине цвета экрана в консольном приложении.
4. Компилирование и затем запуск программы, проверка корректности выводимых данных.

Текст исходной программы

#include "pch.h"

#include <iostream>

#include <windows.h>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int width = GetSystemMetrics(SM\_CXSCREEN);

int height = GetSystemMetrics(SM\_CYSCREEN);

HDC dc = GetDC(NULL);

int bitsPerPixel = GetDeviceCaps(dc, BITSPIXEL);

cout << "Разрешение экрана: " << width << "/" << height << "\n";

cout << "Глубина цвета: " << bitsPerPixel << "\n";

cout << "\n";

cout << "Размер кадра: " << "\n";

cout << width \* height \* bitsPerPixel / 8 << " Байт" << "\n";

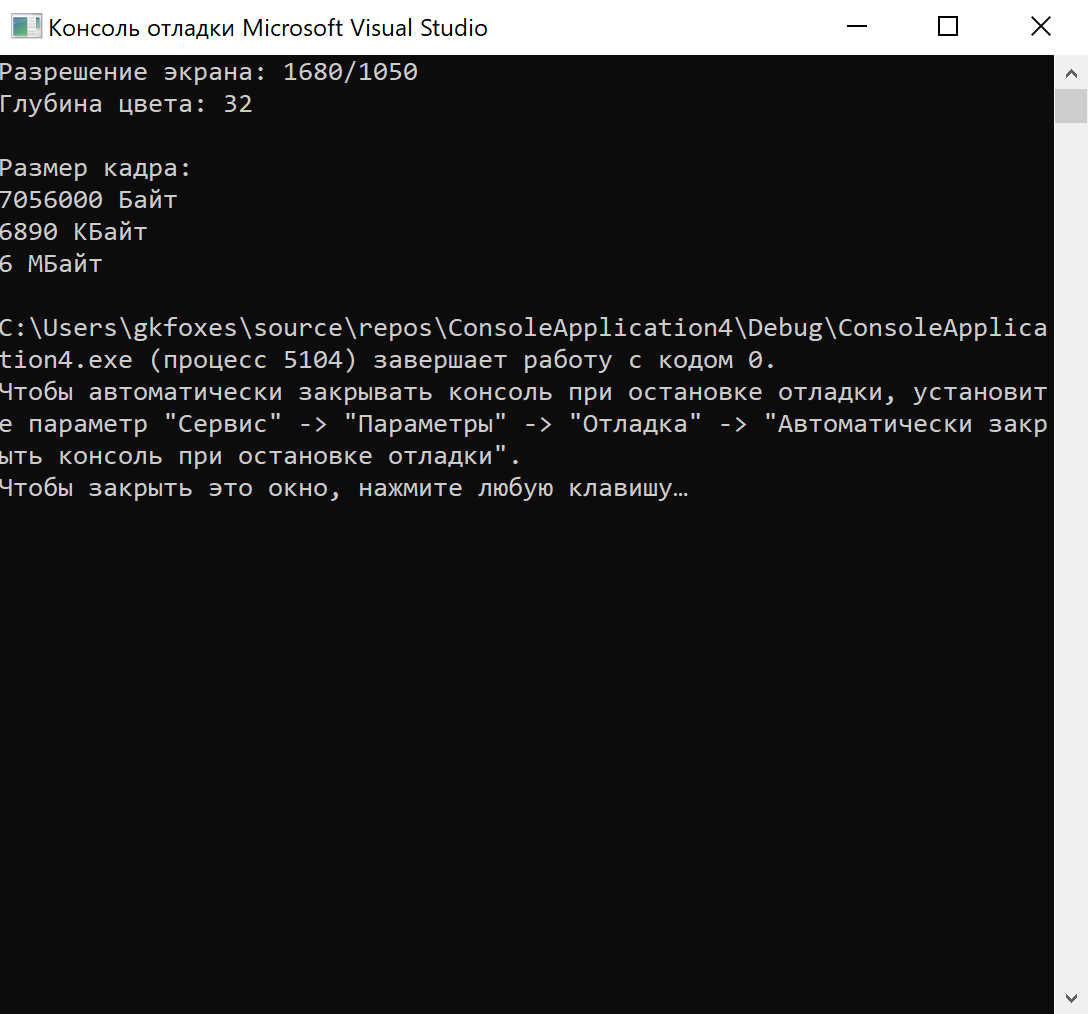
cout << width \* height \* bitsPerPixel / 8 / 1024 << " KБайт" << "\n";

cout << width \* height \* bitsPerPixel / 8 / 1024 / 1024 << " МБайт" << "\n";

}

Вывод по работе

Программа полностью выполняет требуемые условия задания. Функция из Windows API GetSystemMetrics получает разрешение экрана по ширине и высоте, а GetDeviceCaps записывает информацию о глубине цвета монитора. Порядок загрузки программы: препроцессинг добавляет хэдеры в код (#include), затем идет компиляция и ассемблирование. Процесс компоновки связывает все объектные файлы, статические библиотеки в единый исполняемый файл и при запуске программы, ЦП начинает выполнение кода с верхней части функции main(). Пример работы программы можно посмотреть ниже на рисунке 1.



*Рисунок 1 Пример работы программы*