**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР.**

отчет

**по лабораторной работе №2**

**по дисциплине «Организация ЭВМ»**

Тема: «Исследование видеосистемы (текстовый режим)».

|  |  |  |
| --- | --- | --- |
| Студенты гр. 1301 |  | Ищенко Д. О.  Носков. Е. C. |
| Преподаватель |  | Гречухин М. Н. |

Санкт-Петербург

2022

Оглавление

[Теоретический аспект. 3](#_Toc117894229)

[Формулировка задания. 4](#_Toc117894230)

[Описание работы алгоритма. 5](#_Toc117894231)

[Текст программы. 5](#_Toc117894232)

[Пример работы программы. 5](#_Toc117894233)

[О программной среде. 5](#_Toc117894234)

# Теоретический аспект.

Основным техническим средством для оперативного формирования и отображения как текстовой, так и графической информации в компьютере является видеосистема.

Видеосистема компьютера состоит из трех основных компонентов:

* видеоадаптер
* монитор
* ПО (драйвера)

Режимы работы видеоадаптера можно объединить в две группы: текстовые и графические.

CGA - Color Graphics Adapter

VGA - Video Graphics Array

EGA - Enhanced Graphics Adapter

В текстовом режиме пиксели представлены двумя байтами: байтом кода символа и байтом атрибута.

*Что-то про кодирование байта атрибута*

Видеопамять адаптера при работе в текстовых режимах доступна непосредственно из программы.

При работе в графическом режиме появляется возможность управлять цветом любой телевизионной точки экрана или пиксела. Число строк пикселов и число пикселов в каждой строке зависит от режима работы видеоадаптера. Таким образом, экран в графическом режиме представляет собой матрицу пикселов.

Функции консольного ввода-вывода С++ помещены в файле <conio.h>:

window([int](http://code-reference.com/c/datatypes/int) left, [int](http://code-reference.com/c/datatypes/int) top, [int](http://code-reference.com/c/datatypes/int) right, [int](http://code-reference.com/c/datatypes/int) bottom);

// Установка параметров активного текстового окна

Функции textcolor(), textbackground(), textattr() и другие управляют цветом отображаемых символов в окне.

Видеоадаптеры всех типов аппаратно поддерживают курсор.

Прерывание — сигнал к процессору, испускаемый аппаратными средствами или программным обеспечением, и указывающий на событие, которое требует немедленного внимания. Прерывание предупреждает процессор о высокоприоритетном состоянии, требующем прерывания текущего кода, выполняемого процессором. Процессор отвечает, приостанавливая свои текущие действия, сохраняя своё состояние и выполняя функцию, называемую обработчиком прерываний (или подпрограммой обработки прерываний, ISR) для обработки события. Это прерывание является временным, и после завершения обработки обработчика прерывания процессор возобновляет обычную работу. Существует два типа прерываний: аппаратные прерывания и программные прерывания.

Каждое прерывание имеет свой собственный обработчик прерываний. Количество аппаратных прерываний ограничено числом строк запроса прерывания (IRQ) для процессора, но могут быть сотни различных программных прерываний. Прерывания — это широко используемая техника многозадачности компьютеров, в первую очередь в реальном времени. Такая система называется управляемой прерываниями.

# Формулировка задания.

Вариант 5.

Написать программу~~,~~ чтобы в окно с координатами (x1,у1,х2,у2) с шагами Т (секунд) и S (строк) выводилась ~~надпись~~ строка при всех возможных комбинациях цвета фона и цвета символов. Строка содержит обозначение цвета фона и символа. Для каждой комбинации цветов в окне должны выводиться номера или символьные обозначения цветов фона и символов (варианты приведены в табл. 2.4). Цвет окна должен соответствовать цвету фона.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варианта | Координаты окна | | | | Обозначение цвета | | Шаг | | Направление |
| X1 | Y1 | X2 | Y2 | Фона | Символа | T | S |
| 5 | 10 | 8 | 70 | 18 | Англ. | Англ | 0.7 | 2 | Вверх |

Дополнить программу из скроллингом окна с направлением в соответствии с вариантом, используя функции прерывания 10h BIOS. Пример скроллинга приведен в мет.указаниях.

# Описание работы алгоритма.

Реализовано 2 функции, первая, print, отвечает за отрисовку надписей, их цвета, и цвета заднего фона. В ней задается размер окна, а далее реализуется перебор из двух циклов всех пар цветов, который может быть. Эти цвета поочередно присваиваются фону и тексту и выводятся на экран.  
 Вторая функция, scroll, отвечает за скроллинг вверх. В ней заданы все атрибуты для скроллинга, и эта функция вставлена в первую. Включается скроллинг, если задан нужный параметр (int with\_scroll) для первой функции.  
 В функции main запускается функция print.

# Текст программы.

[Ссылка на гитхаб](https://github.com/Nekttuman/EVM_labs_v5)

# Пример работы программы.

# О программной среде.

Ищенко Д. Clion 2022, Cmake 3.10

Носков Е. Visual Studio 2019, Microsoft Visual c++