**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

**отчет**

**по лабораторной работе №2**

**по дисциплине «Организация ЭВМ и Систем»**

**Тема: «Исследование текстового режима видеосистемы»**

**Вариант 4**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0302 |  | Чёрный Я.И. |
| Преподаватель |  | Жандаров В.В. |

Санкт-Петербург

2021

**Оглавление**

[1 Постановка задачи 3](#_Toc90499466)

[1.1 Параметры вывода надписей 3](#_Toc90499467)

[1.2 Дополнительная задача: 3](#_Toc90499468)

[2 Краткие сведения о видеосистемах ПЭВМ, текстовом режиме их работы и функциях обслуживания текстового режима. 3](#_Toc90499469)

[3 Блок-схема программы 5](#_Toc90499470)

[4 Пример работы 6](#_Toc90499471)

[5 Текст программы 7](#_Toc90499472)

[6 Вывод 8](#_Toc90499473)

# Постановка задачи

Целью данной задачи является разработка программы, выводящую надпись в окно с координатами (x1,y1,x2,y2) с шагами T (секунд) и S (строк) при всех возможных комбинациях цвета фона и цвета символов. Для каждой комбинации цветов в окне должны выводиться номера или символьные обозначения цветов фона и символов.

## Параметры вывода надписей

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варианта | Координаты окна | | | | Обозначение цвета | | Шаг | | Направление |
| Xl | Y1 | X2 | Y2 | Фона | Символа | T | S |
| 4 | 25 | 5 | 55 | 15 | Англ. | Англ. | 0.6 | 1 | Вниз |

## Дополнительная задача:

Дополнить программу скроллингом окна с направлением в соответствии с вариантом, используя функции прерывания 10h BIOS.

# Краткие сведения о видеосистемах ПЭВМ, текстовом режиме их работы и функциях обслуживания текстового режима.

|  |
| --- |
| Enum COLORS {  /\* Цвета нормальной яркости: \*/  BLACK, /\* черный, 0 \*/  BLUE, /\* синий, 1 \*/  GREEN, /\* зеленый. 2 \*/  CYAN, /\* сине-зеленый, 3 \*/  RED, /\* красный, 4 \*/  MAGENTA, /\* красно-синий, 5 \*/  BROWN, /\* коричневый, 6 \*/  LIGHTGRAY, /\* светло-серый. 7 \*/  /\* Цвета повышенной яркости: \*/  DARKGRAY, /\* темно-серый, 8 \*/  LIGHTBLUE, /\* ярко-синий, 9 \*/  LIGHTGREEN, /\* ярко-зеленый, 10 \*/  LIGHTCYAN, /\* яркий сине-зеленый, 11 \*/  LIGHTRED, /\* ярко-красный, 12 \*/  LIGHTMAGENTA, /\* яркий красно-синий, 13 \*/  YELLOW, /\* желтый, 14 \*/  WHITE /\* белый. 15 \*/} |

Функция textcolor(int newcolor) задает цвет символов, не затрагивая установленный цвет фона. Цвет может быть или числом, или формироваться из символических констант, значения которых определяет перечислимый тип COLORS.

Функция textbackground(int newcolor) задает цвет фона символов, не затрагивая установленный цвет символа. Цвет может быть или числом, или формироваться из символических констант, значения которых определяет перечислимый тип COLORS. Для цвета фона выбор ограничен значениями цветов 0-7. Если для цвета фона выбирается значение 8 - 15, то символы будут мерцать, так как бит мерцания установится в единицу, но цвет фона будет соответствовать значениям 0-7.

**Функции АН = 06 и 07 прерывания 10h BIOS** осуществляют так называемый скроллинг (прокрутку) окна экрана. При выполнении скроллинга на одну строку вверх вся информа­ция в окне перемещается на строку вверх. Внизу окна появляется чистая строка. При выполнении скроллинга на одну строку вниз содержимое окна сдвигается на строку вниз и вверху окна добавляется чистая строка. Значение регистра AL задает число строк, на которое выполняется скроллинг. Если AL=0, выполняется очистка окна. Значения в СН и CL определяют строку и столбец левого верхнего угла окна, в DH и DL -строку и столбец правого нижнего угла. Строки и столбцы нумеруются от 0. Значение в регистре ВН задает атрибут добавляемой чистой строки.

Приведем пример Си-функции, выполняющей вертикальный скроллинг окна экрана, заданного строкой и столбцом левого верхнего (l\_row, l\_col) и строкой и столбцом правого нижнего (r\_row, r\_col) углов окна. Если переменная direction равна UP, происходит скроллинг на одну строку вверх, если она равна DOWN - скроллинг на одну строку вниз, если ENTIRE -выполняется очистка окна. Добавляется строка с атрибутом attr.

|  |
| --- |
| #include < dos.h>  void scroll (int direction, char l\_row, char l\_col, char r\_row, char r\_col, char attr)  {  union REGS r;  if (direction == DOWN)  r.h.ah = 7;  else  r.h.ah = 6;  r.h.al = lines;  r.h.ch = l\_row; r.h.cl= l\_col; r.h.dh = r\_row; r.h.dl= r\_col;  r.h.bh=attr;  int86(0x10,&r,&r);  } |
|  |

Если окно занимает весь экран и задается direction = ENTIRE, происходит фактическая очистка всего экрана и его "заливка" цветом, задаваемым атрибутом чистой строки attr.

Вывод информации на экран персонального компьютера может выполняться на трех уровнях:

на уровне MS-DOS с использованием функций прерывания 21h

на уровне BIOS с использованием функций прерывания 10h

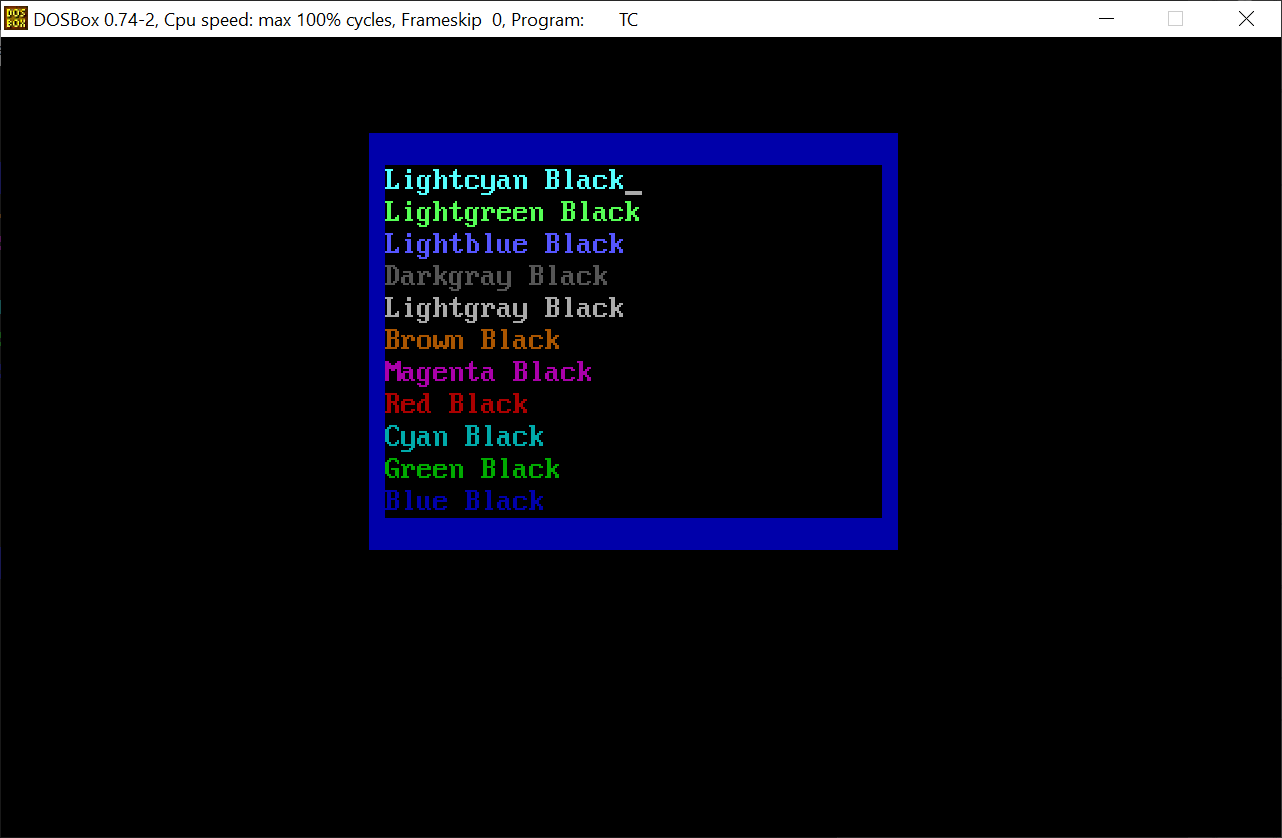
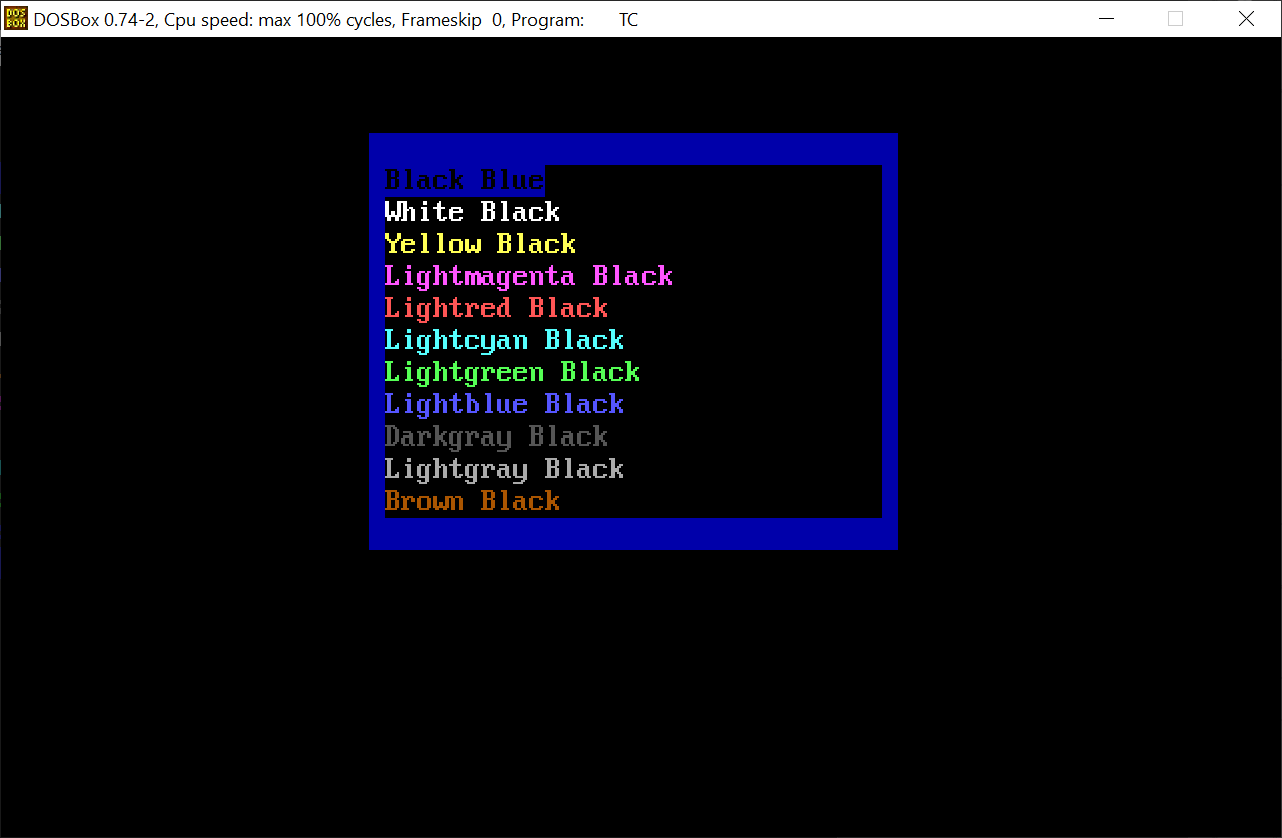
непосредственным доступом к аппаратным средствам.

Функция clreol() стирает в текстовом окне строку, на которую установлен кур-сор, начиная с текущей позиции курсора и до конца строки (до правой вертикаль-ной границы окна).

# Блок-схема программы



# Пример работы



# Текст программы

MAIN.CPP

|  |
| --- |
| #include <conio.h>  #include <dos.h>  #define UP 0x06  #define DOWN 0x07  #define CLEAR 0x00  class SuperWindow  {  private:  int x1, x2, y1, y2;  void scroll(int dir, char l\_row, char l\_col, char r\_row, char r\_col, char attr)  {  union REGS regs;  if (dir)  {  //scroll by direction  regs.h.al = 0x01;  regs.h.ah = dir;  }  else //clear window  {  regs.h.al = 0x00;  regs.h.ah = 0x06;  }  regs.h.ch = l\_row;  regs.h.cl = l\_col;  regs.h.dh = r\_row;  regs.h.dl = r\_col;  regs.h.bh = attr;  int86(0x10, &regs, &regs);  }  public:  SuperWindow(int x1, int x2, int y1, int y2)  {  this->x1 = x1;  this->x2 = x2;  this->y1 = y1;  this->y2 = y2;  clrscr();  window(x1, y1, x2, y2);  DrawBorder(0x10);  }  void DrawBorder(char colorAttr)  {  char borderSize = 1;  //draw border --> blue foreground black text  scroll(CLEAR, y1 - borderSize - 1, x1 - borderSize - 1, y2, x2, colorAttr);  //draw window black foreground white text  scroll(CLEAR, y1 - 1, x1 - 1, y2 - 1, x2 - 1, 0x07);  }  void ScrollUp()  {  scroll(UP, y1 - 1, x1 - 1, y2 - 1, x2 - 1, 0x07);  }  void ScrollDown()  {  scroll(DOWN, y1 - 1, x1 - 1, y2 - 1, x2 - 1, 0x07);  }  void Clear()  {  textbackground(0);  scroll(CLEAR, 0, 0, 24, 79, 0x07);  }  };  int main()  {  int x1 = 25, x2 = 55, y1 = 5, y2 = 15;  float delayTime = 0.6f;  int sCount = 1;  SuperWindow w(x1, x2, y1, y2);  int colorsCount = 16;  for (int i = 0; i < colorsCount; i++)  {  for (int j = 0; j < colorsCount; j++)  {  textcolor(j);  textbackground(i);  char\* eng[16] = { "Black","Blue","Green","Cyan", "Red", "Magenta",  "Brown","Lightgray","Darkgray","Lightblue","Lightgreen","Lightcyan",  "Lightred","Lightmagenta","Yellow","White" };  for (int k = 0; k < sCount; k++)  {  w.ScrollDown();  gotoxy(1, 1);  cprintf("%s %s", eng[j], eng[i]);  }  delay(delayTime \* 1000);  }  }  getch();  w.Clear();  return 1;  } |

# Вывод

В результате выполнения данной работы были получены навыки управления текстовой видеосистемой ПЭВМ, получен опыт использования системных прерываний.