

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«Вятский государственный университет»**  
**(«ВятГУ»)**

Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2  
по дисциплине «Интерфейсы периферийных устройств»

Выполнил студент группы ИВТ-42 \_\_\_\_\_/Рзаев А. Э./

Проверил старший преподаватель кафедры ЭВМ \_\_\_\_\_/Гагарский К. Н./

Киров 2018

## 1 Задание на лабораторную работу

Составить на языке assembler программу вывода на экран монитора графическое изображение в 4-м режиме, состоящее из всех 4-х цветов и палитр.

## 2 Теория

Вывод изображения на экран осуществляется в видеорежиме №4 (CGA). Экран рассматривается как совокупность точек (пикселей)- наименьших элементов изображения. Вывод изображения на экран осуществляется путем управления цветом каждой точки экрана.

В этом режиме экран монитора представляет из себя:

-320 точек по горизонтали

-200 точек по вертикали

2 бита на точку, 1 байт видеопамати – 4 точки.

4 цвета, 2 палитры.

B800-базовый адрес – верхний левый угол экрана.

Координата выводимого пикселя определяется адресом видеопамати

## 3 Исходный код программы

Программа вывода изображения на экран представлена на рисунке 1

```
.286
.model small
.data
    studentName db 'Ali Rzaev', 0
    groupName db "IVT-42", 0
.stack 256h
.code

drawPoint proc near
    push es

    mov di, 0
    mov dx, cx
    and dx, 1
    jz skipFixY
    add di, 2000h
skipFixY:
    shr cx, 1
```

```

; расчет положения точки в байте
    push cx
    and ax, 3
    shl ax, 8
    mov cx, bx
    and cx, 3
    add cx, 1
    fixPointY:
        shr ax, 2
        loop fixPointY
    pop cx

; расчет адреса
    push ax

    mov ax, 80
    mul cx
    shr bx, 2
    add ax, bx
    add di, ax

    pop ax

; рисование точки
    or byte ptr es:[di], al

    pop es
    ret
drawPoint endp

drawLine proc near
    push bp
        mov bp, sp
        sub sp, 12
        ;[bp - 2] = e
        ;[bp - 4] = i
        ;[bp - 6] = s1
        ;[bp - 8] = s2
        ;[bp - 10] = f
        ;[bp - 12] = tmp

        ; расчет Dx, Dy
        ; cx = Dx
        mov word ptr [bp-6], 1 ; s1 = 1
        sub cx, ax ; Dx = x2 - x1
        cmp cx, 0
        jge skipAbsDx ; Dx >= 0?
        neg cx ; Dx = -Dx
        mov word ptr [bp-6], -1 ; s1 = -1
    skipAbsDx:
        ; dx = Dy
        mov word ptr [bp-8], 1 ; s2 = 1
        sub dx, bx ; Dy = y2 - y1
        cmp dx, 0
        jge skipAbsDy ; Dy >= 0?

```

```

        neg cx          ; Dy = -Dy
        mov word ptr [bp-8], -1 ; s2 = -1
skipAbsDy:

; расчет флага f
mov word ptr [bp-10], 0 ; f = 0
cmp dx, cx
jng skipSwapDxDy      ; Dy > Dx?
        xchg dx, cx    ; Dy <=> Dx
        mov word ptr [bp-10], 1 ; f = 1
skipSwapDxDy:

mov word ptr [bp-2], dx ; e = Dy
mov word ptr [bp-4], 1 ; i = 1

loopCond:
        mov di, cx     ; di = Dx
        add di, dx     ; di += Dy
        cmp word ptr [bp-4], di
        jg loopEnd     ; i <= Dy + Dx?

loopBegin:
        ; putpixel(x, y, c)
        pusha
        mov cx, bx ; y
        mov bx, ax ; x
        mov ax, si ; c
        call drawPoint
        popa

        ; поправка e
        mov di, word ptr [bp-2] ; di = e
        mov word ptr [bp-12], di ; tmp = di

        shl word ptr [bp-12], 1 ; tmp = tmp * 2
        cmp word ptr [bp-12], cx
        jl ifLess      ; tmp(2e) < Dx?
        jnl ifNotLess

ifLess:
        add word ptr [bp-2], dx ; e += dy

        mov di, word ptr [bp-10] ; di = f
        mov word ptr [bp-12], di ; tmp = f
        cmp word ptr [bp-12], 0
        jnz inclY      ; f == 1?
        jz inclX

inclY:
        add bx, word ptr [bp-8] ; y += s2
        jmp endInc1

inclX:
        add ax, word ptr [bp-6] ; x += s1
        jmp endInc1

```

```

        endInc1:
            jmp endIf1

    ifNotLess:
        sub word ptr [bp-2], cx ; e -= dx

        mov di, word ptr [bp-10] ; di = f
        mov word ptr [bp-12], di ; tmp = f
        cmp word ptr [bp-12], 0
        jnz inc2X            ; f == 1?
        jz inc2Y
    inc2Y:
        add bx, word ptr [bp-8] ; y += s2
        jmp endInc2
    inc2X:
        add ax, word ptr [bp-6] ; x += s1
        jmp endInc2
    endInc2:
        jmp endIf1
    endIf1:

        add word ptr [bp-4], 1 ; i += 1
        jmp loopCond
    loopEnd:

    add sp, 12
    pop bp

    ret
drawLine endp

; рисуем фигуру
drawFigure proc near
    pusha
    mov ax, 45
    mov bx, 15
    mov cx, 105
    mov dx, 15
    mov si, 1h
    call drawLine
    popa

    pusha
    mov ax, 45
    mov bx, 15
    mov cx, 45
    mov dx, 75
    mov si, 2h
    call drawLine
    popa

    pusha
    mov ax, 105
    mov bx, 15
    mov cx, 105

```

```

        mov dx, 75
        mov si, 3h
        call drawLine
        popa

        ret
drawFigure endp

main:
        mov ax, @data
        mov ds, ax

        ; переход в графический режим
        mov al, 04h
        mov ah, 00h
        int 10h

        ; установка базового адреса
        mov ax, 0B800h
        mov es, ax

        ; установка палитры
        mov ah, 0Bh
        mov bx, 0100h ; первая палитра
        ;mov bx, 0101h ; вторая палитра
        int 10h

        pusha
        call drawFigure
        popa

        ; ожидание клавиши
        mov ah, 00h
        mov al, 13h
        int 16h

        ; переход в текстовый режим
        mov al, 03h
        mov ah, 00h
        int 10h

        ; выход из программы
        mov ax, 4c00h
        int 21h
end main

```

Рисунок 1 – Программа вывода изображения на экран

## 4 Экранные формы

Экранная форма разработанной программы представлена на рисунках 2 и 3.

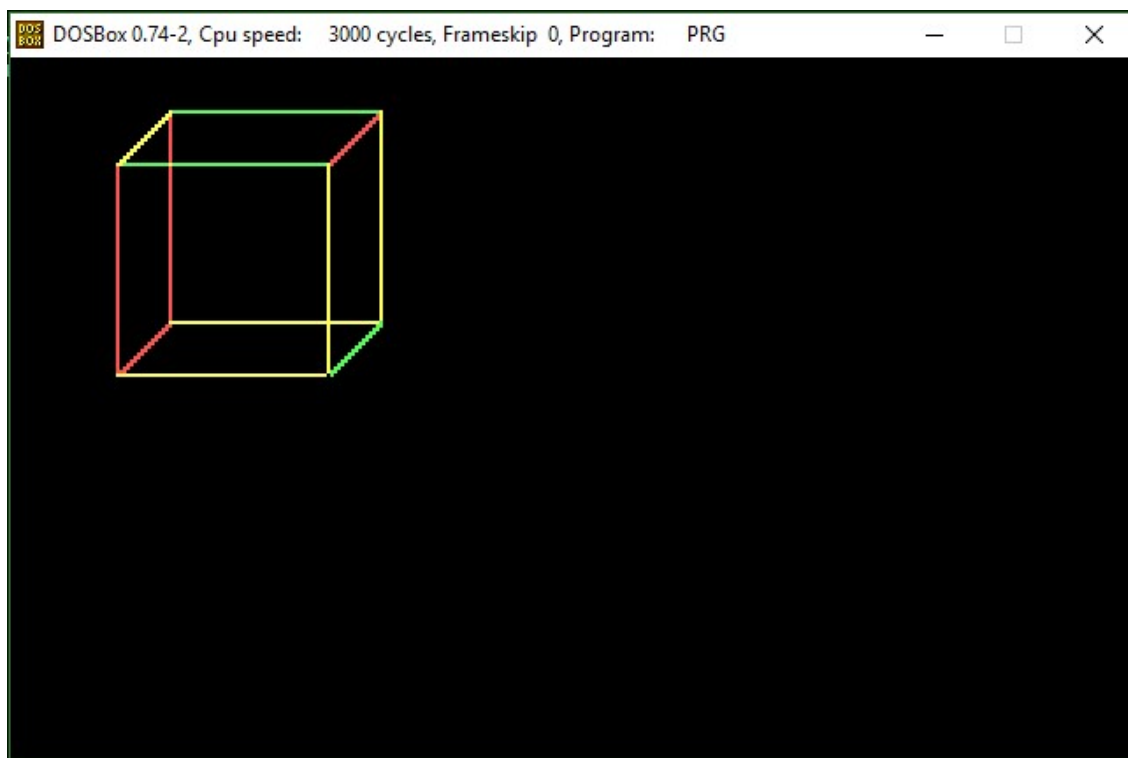


Рисунок 2 – Экранная форма программы

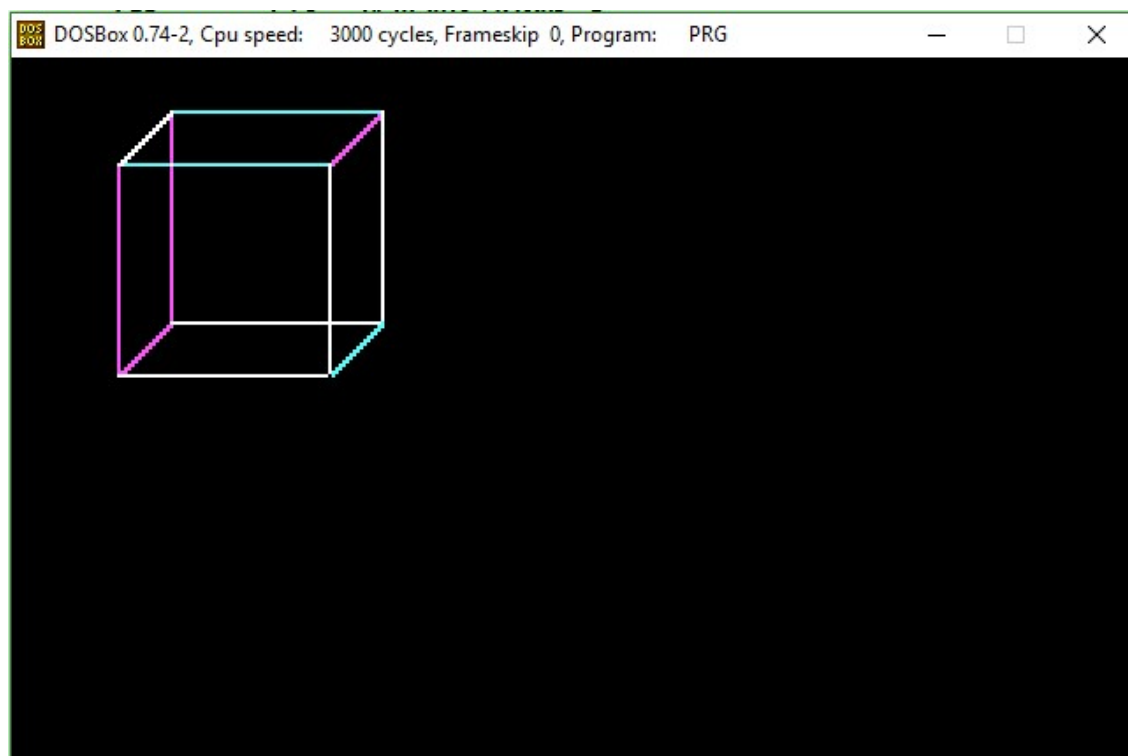


Рисунок 3 – Экранная форма программы

## **5 Вывод**

В ходе выполнения лабораторной работы была разработана программа, демонстрирующая работу с видеопамятью. Был подробно изучен 4-й видеорежим и различные виды цветов и палитры. Каждый пиксель задается 2 битами, соответственно в одном байте закодировано 4 пикселя. Для обеспечения чересстрочной развертки четные и нечетные строки изображения разнесены в видеопамяти на 2000h