Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Лабораторная работа №1 (вариант 29)**

по дисциплине: «Автоматизация проектирования

микропроцессорных систем».

Выполнил:

студент 4 курса, гр. ИВТАПбд-41

Кондратьев Павел Сергеевич.

Проверил:

преподаватель кафедры ВТ

Игонин Андрей Геннадьевич.

г. Ульяновск, 2019

**1) Задание:**



Реализовать на Assembler подключение к советующему устройству, считать и передать на него данные. На экране должно отображаться текущее состояние получаемых/передаваемых данных.

**2) Краткие теоретические сведения:**

Видеоадаптер является, пожалуй, одним из самых интересных устройств, входящих в состав современного компьютера. В тандеме с монитором он позволяет проникнуть в совершенно другой, виртуальный мир, расцвеченный невообразимыми красками и эффектами. Он служит своеобразным мостом между бездушными железяками и ожившими образами.

Стандарты видеоадаптеров имеют достаточно большую историю. Первые монохромные адаптеры появились в начале 80-х годов прошлого века. Они поддерживали только 2 цвета и работали исключительно в текстовом режиме. Далее появился новый стандарт – CGA (Color Graphics Adapter), позволяющий работать и в текстовом и в графическом режиме. Сегодня применяются современные VGA-совместимые (Video Graphics Array) видеоадаптеры, которые поддерживают большое количество текстовых и графических режимов: 4 текстовых, 256-цветных, HiColor (65 536 цветов) и TrueColor (минимум 16,7 млн цветов).

**3) Порядок выполнения работы:**

Для лабораторной работы был поставлен на PC эмулятор, который создавал DOS-окружение (DosBox). Также был установлен язык программирования TASM, на котором и собственной производилось выполнение лабораторной работы.

Исходный код программы смотрите в Приложении 1.

Также для удобства было создано 2 bat файла:

1. Debug.bat – запуск пошагового решения программы

mount d e:\DOSBox\TASM

d:\tasm mlab1

if errorlevel 1 goto exit

d:\tlink mlab1

d:\td MLAB1.exe

:exit

pause

exit

1. Start.bat – запуск приложения

mount d e:\DOSBox\TASM

d:\tasm mlab1

if errorlevel 1 goto exit

d:\tlink mlab1

MLAB1

:exit

pause

exit

Bat файлы были создана для упрощения работы (сборки решения) и проверок на наличие ошибок.

**Подфункция 00h(функции 0bh)** - предназначена для установки цвета обрамляющей экран рамки в текстовом режиме и цвета фона в графическом.

mov ah, 0 ; установим новый режим

mov al, 04h ; графический режим 04h, 320 x 200

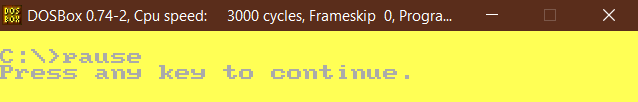
int 10h

mov ah,0bh

mov bh, 00h ; код конфигурации для устанвки цвета

mov bl, 14 ; устанавливаем желтый цвет фона [от 0 до 15]

int 10h



**Функция 0ch** - позволяет вывести на экран дисплея один пиксель.

mov ah, 0

mov al, 04h

int 10h

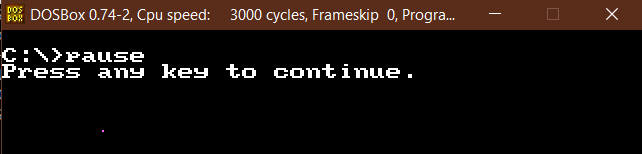
mov ah, 0ch

mov al, 14 ;устанавливаем желтый цвет

mov cx, 50 ;столбец 50

mov dx, 50 ;строка 50

int 10h



**Функция 0eh** - позволяет выводить на экран дисплея в режиме телетайпа. Эту функцию удобно использовать для вывода строки символов вместо функции 09h.

mov ah, 0eh

lea bx, OutPutText ; получаем адресс строки

@repeat:

mov al, [bx] ;записываем первый символ

cmp al, '$' ;проверям конец строки

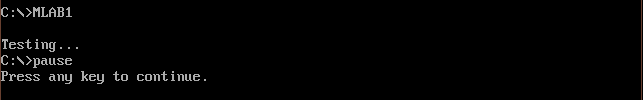
je EXIT\_CODE ;выход из цикла

int 10h ; выводим символ на экран

inc bx

jmp short @repeat

EXIT\_CODE:



Пример для получения информации об установленном видеоконтроллере.

mov bx, ds ; адресуем сегмент данных

mov es, bx ; передаем адрес

mov di, offset Video\_Info

mov ax, 4f00h

int 10h

cmp al, 4fh

jne No\_Sup ;функция не поддерживается

cmp ah, 0

jnz Error\_HND ; произошла ошибка

сохранение полный размер видеопамяти в килобайтах

xor eax, eax

mov ax, [word ptr Video\_Info + 12h]

shl ax, 6

mov Total\_Memory, eax

mov edx, eax

call print

mov ah,9

mov dx,offset MB

int 21h

jmp TOEND

No\_Sup:

mov ah,9

mov dx,offset OutPutSup

int 21h

jmp TOEND

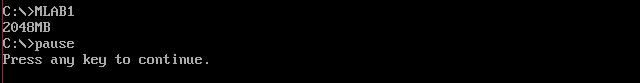
Error\_HND:

mov ah,9

mov dx,offset OutPutError

int 21h

jmp TOEND



Для вывода больших чисел была написана процедура print, которая через остаток от деления на 10 выводила получившиеся цифры.

print:

mov eax, edx

xor ecx, ecx

mov ebx, 10

oi2:

xor edx,edx

div ebx

push edx

inc ecx

test eax, eax

jnz oi2

mov ah, 02h

oi3:

pop edx

add dl, '0'

int 21h

loop oi3

ret

Установка текстового или графического режима.

mov ah, 00h

mov al, 11h ; 11h - графический режим 640 x 480 (2 цвета) 01h - текстовый режим 40 x 25 (16 цвета)

int 10h

cmp al, 30h

je Error\_Mode

mov ah,9

mov dx,offset OutPutWork

int 21h

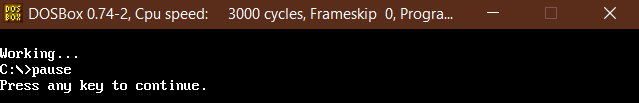
jmp TOEND

Error\_Mode:

mov ah,9

mov dx,offset OutPutError

int 21h



**01h** - позволяет установить размер курсора для текстовых режимов работы видеоадаптеров.

mov ah, 01h

mov ch, 0 ;номер начальной линии, начиная сверху экрана

mov cl, 0ah ; номер конечной линии равен 10

int 10h



**03h** - позволяет получить размер и позицию курсора для указанной страницы видеопамяти.

mov ah, 03h

mov bh, 0 ; номер страницы

int 10h получаем позицию курсора

mov CUR\_Y, dh ; номер строки

mov CUR\_X, dl ; номер столбца

**09h** - предназначена для записи символа с указанными атрибутами в текущуб позицию курсора.

mov ah, 02h

mov bh, 00h ; номер страницы

mov dh, 1 ; номер строки

mov dl, 10 ; номер столбца

int 10h

mov ah, 9h

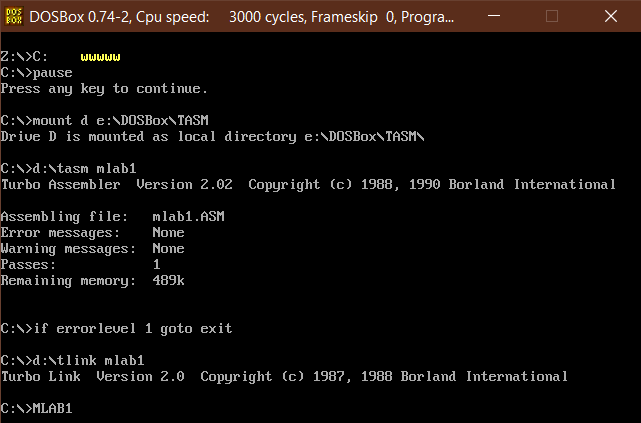
mov al, 'w' ; указываем символ, который нужно записать

mov bh, 00h ; номер страницы

mov bl, 14 ; желтый цвет сивола

mov cx, 5 ; повторить 5 раз

int 10h



**1015h** - позволяет получить информацию о цвете для указанного регистра ЦАП (Цифро-аналоговый преобразователь) номер регистра от 00h до ffh.

mov ax, 1015h

mov bl, 03h

int 10h

mov GREEN, ch ; читаем насыщенность зеленого цвета

mov BLUE, cl ; читаем насыщенность синего цвета

mov RED, dh ; читаем насыщенность красного цвета

mov ah,9

mov dx,offset GREEN\_INTENSITY

int 21h

mov dl, GREEN

call print call Newline

mov ah,9

mov dx,offset BLUE\_INTENSITY

int 21h

mov dl, BLUE

call print call Newline

mov ah,9

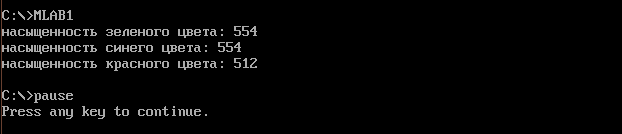
mov dx,offset RED\_INTENSITY

int 21h

mov dl, RED

call print call Newline

jmp TOEND



**1a00h** - позволяет определить тип подключения дисплея.

mov ax, 1a00h

int 10h

cmp bl, 08h

jne ER

mov ah,9

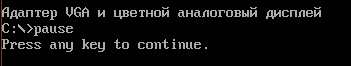
mov dx,offset OutPut01

int 21h

jmp TOEND

ER:

jmp TOEND



**Вывод:**

В ходе выполнения данной лабораторной работы, мне удалось освоить базовую разработку программ на assembler. Я научился создавать bat-файлы, с помощью которых можно сократить время работы при ручном написании команд в DosBox. Получил базовые навыки работы со строками и числами. Разобрался с отличиями макросов от процедур. Научился реализовывать подключение к советующему устройству, считать и передать на него данные, делая так, чтобы на экране отображалось текущее состояние получаемых/передаваемых данных.

**Список литературы:**

* Всеволод Невский Программирование аппаратных средств в windows СПБ.: БХВ-Петербург, 2004. – 880с.: ил.
* Калашников О. А. Ассемблер? Это просто! Учимся программировать. — [БХВ-Петербург](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3&action=edit&redlink=1), 2011.
* Юров В., Хорошенко С. Assembler: учебный курс. — СПб.: [Питер](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE))

**Приложение 1**

.MODEL SMALL

.STACK 200h

.386

; Декларации данных

.DATA

OutPutText db 13, 10, "Testing...$"

OutPutError db 13, 10, "произошла ошибка...$"

OutPutSup db 13, 10, "функция не поддерживается...$"

OutPutWork db 13, 10, "Working...$"

OutPut01 db 13, 10, "Адаптер VGA и цветной аналоговый дисплей$"

NL db 13,10,'$'

Video\_Info db 512 dup (?)

Total\_Memory dd ?

CUR\_Y db ?

CUR\_X db ?

GREEN db ?

BLUE db ?

RED db ?

MB db "MB$"

GREEN\_INTENSITY db "насыщенность зеленого цвета: $"

BLUE\_INTENSITY db "насыщенность синего цвета: $"

RED\_INTENSITY db "насыщенность красного цвета: $"

;========================= Программа =========================

.CODE

BEGIN LABEL NEAR

; инициализация сегментного регистра

mov AX, @DATA

mov DS, AX

; Подфункция 00h(0bh) предназначена для установки цвета обрамляющей экран рамки в текстовом режиме и цвета фона в графическом

;mov ah, 0 ; установим новый режим

;mov al, 04h ; графический режим 04h, 320 x 200

;int 10h

;mov ah,0bh

;mov bh, 00h ; код конфигурации для устанвки цвета

;mov bl, 14 ; устанавливаем желтый цвет фона [от 0 до 15]

;int 10h

; Функция 0ch позволяет вывести на экран дисплея один пиксель

;mov ah, 0

;mov al, 04h

;int 10h

;mov ah, 0ch

;mov al, 14 ;устанавливаем желтый цвет

;mov cx, 50 ;столбец 50

;mov dx, 50 ;строка 50

;int 10h

; Функция 0eh позволяет выводить на экран дисплея в режиме телетайпа. Эту функцию удобно использовать для вывода строки символов вместо функции 09h

;mov ah, 0eh

;lea bx, OutPutText ; получаем адресс строки

;@repeat:

;mov al, [bx] ;записываем первый символ

;cmp al, '$' ;проверям конец строки

;je EXIT\_CODE ;выход из цикла

;int 10h ; выводим символ на экран

;inc bx

;jmp short @repeat

;EXIT\_CODE:

; Пример для получения информации об установленном видеоконтроллере

;mov bx, ds ; адресуем сегмент данных

;mov es, bx ; передаем адрес

;mov di, offset Video\_Info

;mov ax, 4f00h

;int 10h

;cmp al, 4fh

;jne No\_Sup ;функция не поддерживается

;cmp ah, 0

;jnz Error\_HND ; произошла ошибка

;сохранение полный размер видеопамяти в килобайтах

;xor eax, eax

;mov ax, [word ptr Video\_Info + 12h]

;shl ax, 6

;mov Total\_Memory, eax

;mov edx, eax

;call print

;mov ah,9

;mov dx,offset MB

;int 21h

;jmp TOEND

;No\_Sup:

; mov ah,9

; mov dx,offset OutPutSup

; int 21h

; jmp TOEND

;Error\_HND:

; mov ah,9

; mov dx,offset OutPutError

; int 21h

; jmp TOEND

; Удаление курсора с экрана

;xor al, al

;mov al, 0ah

;mov dx, 03d4h ;

;out dx, al ;

;xor bl, bl

;mov dx, 03d5h ;

;in bl, dx ;

;test bl, 20h ;

;je END\_PROC ;

;and bl, 20h ;

;mov al, 0ah ;

;mov dx, 03d4h ;

;out dx, al ;

;mov dx, 03d5h ;

;mov al, bl ;

;out dx, al ;

;END\_PROC:

; Установка текстового или графического режима

;mov ah, 00h

;mov al, 11h ; 11h - графический режим 640 x 480 (2 цвета) 01h - текстовый режим 40 x 25 (16 цвета)

;int 10h

;cmp al, 30h

;je Error\_Mode

;mov ah,9

;mov dx,offset OutPutWork

;int 21h

;jmp TOEND

;Error\_Mode:

; mov ah,9

; mov dx,offset OutPutError

; int 21h

; 01h позволяет установить разер курсора для текстовых режимов работы видеоадаптеров

;mov ah, 01h

;mov ch, 0 ;номер начальной линии, начиная сверху экрана

;mov cl, 0ah ; номер конечной линии равен 10

;int 10h

; 02h позволяет установить текущую позицию курсорана экране дисплея

;mov ah, 02h

;mov bh, 0 ; номер страницы

;mov dh, 6 ; номер строки

;mov dl, 27 ; номер столбца

;int 10h

;mov ah,9

;mov dx,offset OutPutWork

;int 21h

; 03h позволяет получить размер и позицию курсора для указанной страницы видеопамяти

;mov ah, 03h

;mov bh, 0 ; номер страницы

;int 10h

; получаем позицию курсора

;mov CUR\_Y, dh ; номер строки

;mov CUR\_X, dl ; номер столбца

; 05h позволяет выбрать текущую страницу видеопамяти для текстового режима диапазаон от 00h до 07h

;mov ah, 05h

;mov bh, 01h ; вторая станица видеопамяти

;int 10h

; 05h позволяет прокрутить активную страницу вверх

;mov ah, 05h

;mov bh, 00h ; выбираем первую страницу

;int 10h

;mov ah, 06h

;mov al, 00h ; число строк

;mov bh, 7 ; атрибут очистки

;mov ch, 2 ; ноер строки вверхнего угла

;mov cl, 15 ; номер столбца вверхнего угла

;mov dh, 10 ; номер строки нижнего угла

;mov cl, 15 ; номер столбца нижнего угла

;int 10h

; 09h предназначена для записи символа с указанными атрибутами в текущуб позицию курсора

;mov ah, 02h

;mov bh, 00h ; номер страницы

;mov dh, 1 ; номер строки

;mov dl, 10 ; номер столбца

;int 10h

;mov ah, 9h

;mov al, 'w' ; указываем символ, который нужно записать

;mov bh, 00h ; номер страницы

;mov bl, 14 ; желтый цвет сивола

;mov cx, 5 ; повторить 5 раз

;int 10h

; 36h позволяет управлять регенерацией (обновлением) экрана дисплея

;mov ah, 12h

;mov bl, 36h

;mov al, 00h ; разрешить регенерацию (01h - запретить)

;int 10h

;cmp al, 12h

;jne ERR

;jmp TOEND

;ERR:

; mov ah,9

; mov dx,offset OutPutWork

; int 21h

; 1015h позволяет получить информацию о цвете для указанного регистра ЦАП(Цифро-аналоговый преобразователь) номер регистра от 00h до ffh

;mov ax, 1015h

;mov bl, 03h

;int 10h

;mov GREEN, ch ; читаем насыщенность зеленого цвета

;mov BLUE, cl ; читаем насыщенность синего цвета

;mov RED, dh ; читаем насыщенность красного цвета

;mov ah,9

;mov dx,offset GREEN\_INTENSITY

;int 21h

;mov dl, GREEN

;call print

;call Newline

;mov ah,9

;mov dx,offset BLUE\_INTENSITY

;int 21h

;mov dl, BLUE

;call print

;call Newline

;mov ah,9

;mov dx,offset RED\_INTENSITY

;int 21h

;mov dl, RED

;call print

;call Newline

;jmp TOEND

; 1a00h позволяет определить тип подключения дисплея

;mov ax, 1a00h

;int 10h

;cmp bl, 08h

;jne ER

;mov ah,9

;mov dx,offset OutPut01

;int 21h

;jmp TOEND

;ER:

;jmp TOEND

jmp TOEND

;;;;;;;; Вывод числа ;;;;;;;;;;;;;;;;;;;;;

print:

mov eax, edx

xor ecx, ecx

mov ebx, 10

oi2:

xor edx,edx

div ebx

push edx

inc ecx

test eax, eax

jnz oi2

mov ah, 02h

oi3:

pop edx

add dl, '0'

int 21h

loop oi3

ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;; перенос троки ;;;;;;;;;;;;;;;;

Newline:

push ax

push dx

mov ah,9

lea dx,NL

int 21h

pop dx

pop ax

ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

TOEND:

mov ax, 4c00h

int 21h

END BEGIN