**Лабораторная работа. Работа с экраном и видеоадаптерами**Цель работы: Вывод текста на экран путём непосредственного программирования видеобуфера. Введение задержки для программных операций.

**2.3. Методические рекомендации**

**2.3.1. Введение**

Все возможности видеосистемы компьютера можно реализовать с помощью видеофункций BIOS прерывания int 10h. Прерывание int 10h обеспечивает: смену видеорежима (текстовый или графический); вывод символьной и текстовой информации; смену шрифтов, настройку цветовой палитры, работу с графическим изображением. Программирование видеосистемы с помощью средств BIOS более громоздко, однако большие возможности и высокая скорость вывода обуславливают широкое использование этого метода в прикладных программах.

В данной работе рассматриваются функции BIOS для обслуживания видеосистемы компьютера, а также функции для работы с клавиатурой. Перечислим функции, являющиеся предметом рассмотрения в лабораторной работе.

**Int 10h:**

* функция 00h - установка видеорежима;
* функция 02h - установка позиции курсора;
* функция 03h - считывание позиции и размера курсора;
* функция 05h - установка видеостраницы;
* функция 06h (07h) - инициализация или прокрутка окна вверх (вниз);
* функция 08h - чтение символа и атрибута в позиции курсора;
* функция 09h - запись символа и атрибута в позицию курсора;
* функция 0Ah - запись символа в позицию курсора с текущим атрибутом;
* функция 0Eh - запись символа в режиме телетайпа с текущим атрибутом;
* функция 0Fh - получить режим дисплея;
* функция 1003h - переключение назначения старшего бита байта атрибута: мерцание/яркость,
* функция 13h - запись строки с заданным атрибутом в режиме телетайпа.

**Int 16h:**

* функция 00h (10h) - чтение символа с клавиатуры с ожиданием;
* функция 01h(11h)- проверка буфера клавиатуры на наличие в нём символа;
* функция 02h (12h) - получение флагов (расширенной) клавиатуры.

**2.3.2. Прямое программирование видеобуфера в текстовом режиме**

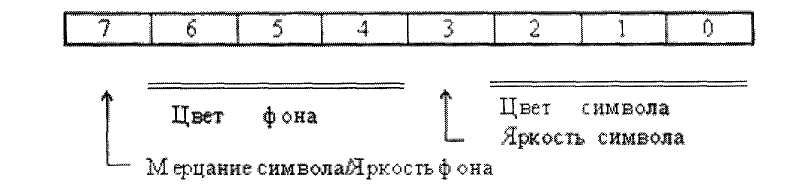
Современные видеоконтроллеры поддерживают разнообразные текстовые и графические режимы. Текстовые режимы различаются по разрешению (число отображаемых символов по горизонтали и вертикали) и цветовой палитре (монохромный или 16-цветный режим). Для графических режимов основным признаком классификации является количество одновременно отображаемых цветов и, соответственно, количество бит видеопамяти, отводимое на каждую точку (пиксел) изображения. Различают следующие типы графических режимов:

* монохромный (1-битное кодирование);
* 16-цветный *EGA/VGA* (4-битное кодирование);
* 256-цветный *SVGA* (8-битное кодирование);
* *HiColor* (16-битное кодирование);

- *True Color* (24-битное / 32-битное кодирование).

Всё, что изображено на мониторе - графика, текст - одновременно присутствует в памяти, встроенной в видеоадаптер. Для того чтобы изображение появилось на мониторе, оно должно быть записано в память видеоадаптера.

На экране отображается видеобуфер, соответствующий активной странице. В текстовых режимах для изображения каждого символа отводится 2 байта: байт с ASCII-кодом символа и байт с его атрибутом. Вообще при формировании изображения непосредственно в видеобуфере, в обход программ DOS и BIOS, все управляющие коды ASCII теряют свои управляющие функции и отображаются в виде соответствующих символов. Структура байта атрибутов приведена на рис. 2.1.

  
  
Рис. 2.1.-Структура байта атрибутов

Из рис. 2.1 следует, что каждый символ может принимать любой из 16 возможных цветов, определяемых сочетанием младших 4-х битов. Биты 4-6 байта атрибутов задают цвет фона под данным символом. Последний бит 7, в зависимости от режима видеоадаптера, определяет либо яркость фона под данным символом (тогда фон также может принимать 16 разных цветов), либо мерцание символа (устанавливается DOS no умолчанию).

При загрузке машины устанавливается стандартная палитра, коды цветов которой приведены в табл. 2.1. Рассмотрим некоторые примеры. Так, в режиме мерцания значение старшего полубайта атрибута 8h обозначает не серый фон, а чёрный при мерцающем символе, цвет которого по-прежнему определяется младшим полубайтом; значение старшего полубайта 0Ch - красный фон при мерцающем символе. Переключение назначения бита 7 осуществляется подфункцией 03h функции 10h прерывания int 10h.   
Таблица 2.1 – Коды цветов стандартной палитры

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Цвет | Код | Цвет |
| 0h | Чёрный | 8h | Серый |
| lh | Синий | 9h | Голубой |
| 2h | Зелёный | 0Ah | Салатовый |
| 3h | Бирюзовый | 0Bh | Светло-бирюзовый |
| 4h | Красный | 0Ch | Розовый |
| 5h | Фиолетовый | 0Dh | Светло-фиолетовый |
| 6h | Коричневый | 0Eh | Жёлтый |
| 7h | Белый | 0Fh | Ярко- белый |

Двухбайтовые коды символов записываются в видеобуфер в том порядке, в каком они должны появиться на экране: первые 80\*2 байт соответствуют первой строке экрана, вторые 80\*2 байт - второй и т.д. При этом переход на следующую строку экрана определяется не управляющими кодами возврата каретки и перевода строки, а размещением кода в другом месте видеобуфера. Вычислить смещение ячейки в координатах "строка-столбец" (row, clm) можно так:

VidAddr= (row\* 160) + (clm\* 2)

При большом объёме выводимых данных, информационный кадр формируется заранее в буфере пользователя, располагающегося в сегменте данных программы.

**2.3.3. Прерывание int 10h. Видеофункции BIOS**

**Функция 00h.** Установка видеорежима (табл. 2.2) текущей видеостраницы с очисткой экрана *(быстрая очистка экрана реализуется функцией 06h и 07h, полная очистка экрана 00h).*

Вызов: *АН = 00h,*

*Al =* видеорежим (код режима задаётся в младших 7 битах, установка в 1 старшего бита запрещает очистку экрана).

Регистры *АХ, ВР, SI, и DI* – не используются в данной функции.

**Функция 02h.** Установка позиции курсора.

Задаёт положение курсора на экране в текстовых координатах, с которых в дальнейшем будет выводиться текст. Отсчёт номера строки и столбца ведётся от верхнего левого угла. Курсор можно установить как в текстовом, так и в графическом режиме, однако, в графическом режиме курсор не виден. BIOS поддерживает до восьми независимых курсоров - по одному на каждую страницу (см. табл. 2.2) независимо от того, какая страница является активной. Функцию *02h* BIOS можно использовать в комбинации с функциями DOS для организации вывода на экран.

Вызов: *АН* = *02h; ВН =* номер страницы (0,1,...7), **обычно 0;** *DH =* строка; *DL =* столбец.

Регистры *АХ, ВР, SI* и *DI* – не используются в данной функции.

**Функция 03h.** Считывание позиции и размера курсора.

Возвращает текущие координаты состояния курсора на выбранной странице. Это даёт возможность временно перейти для работы на другое место экрана, а затем вернуться на старое место. Функцию *03h* BIOS можно использовать в комбинации с функциями DOS для организации вывода на экран.

Вызов: *АН = 03h, ВН* = номер страницы (0,1,...7), **обычно 0.**

Возврат: *DH, DL* = строка и столбец текущей позиции курсора, *СH, CL =* первая и последняя строки развёртки курсора – размер курсора.

Регистры АХ, ВР, SI и DI – не используются в данной функции.

**Функция 08h.** Чтение символа и атрибута в текущей позиции курсора на выбранной странице.

Вызов: *АН = 08h, (цвет) ВН* = номер страницы (0,...,7), **обычно 0.**

Возврат: *АН =* атрибут символа, *AL* = *ASCII-код* символа.

Регистры *ВР, SI* и *D* – не используются в данной функции.

**Функция 09h.** Запись символа с заданным атрибутом на экран в позицию курсора. Действует как в графическом, так и в текстовом режимах. В графическом режиме символы не должны переходить на следующую строку. *Все коды в AL* *рассматриваются как символьные и не управляют положением курсора. После вывода символа курсор смещается к следующей позиции функцией 02h.* Коэффициент повторения позволяет выводить строки одинаковых символов. В текстовом режиме символ выводится с указанным в *BL* атрибутом. В графическом - содержимое *BL* влияет только на цвет символа, но не на фон под ним. Графическое изображение под знакоместом затирается.

Вызов: *АН =09h, AL = ASCII-код* символа,

*BL —* атрибут символа (текстовый режим) или только цвет символа (графический режим),

*ВН =* номер страницы (0,1,...7), *СХ=* коэффициент повторения.

Регистры *АХ, ВР, SI* и *DI* – не используются в данной функции.

**Функция 0Ah.** Запись символа с текущим атрибутом на экран в позицию курсора. Функция действует как в графическом, так и в текстовом режимах. Символ принимает атрибут, установленный ранее для этой позиции. *Все ASCII-коды в AL рассматриваются как символьные и не управляют положением курсора (также как и в функции 09h). После вывода символа курсор смещается к следующей позиции функцией 02h.*

Вызов: *АН = 0Ah, AL = ASCII-код* символа,

*ВН =* номер страницы (0,1,...7), *СХ =* коэффициент повторения.

Регистры *АХ, ВР, SI* и *DI* – не используются в данной функции.

**Функция 0Fh.** Получить режим дисплея и номер текущей страницы.

Вызов: *АН = 0Fh.*

Возврат: *AL =* режим дисплея, *АН* = ширина экрана в текстовом формате

*ВН* =номер активной страницы.

Регистры *ВР, SI* и *DI* – не используются в данной функции.

**Пример.** Процедура установки позиции курсора на текущей странице.

Вход: *dh* = строка (0 - 25), *dl =* столбец (0 - 79)

Proc SetCursor

;Сохранить регистры (по необходимости)

Mov ah,0Fh

Int 10h

Mov ah,02h

Int l0h

;восстановить регистры

Endp SetCursor

**Функция 10h.** Подфункция *03h.* Переключение бита "мерцание/яркость".

Определяет назначение старшего бита 7 атрибута символа: мерцание символа или повышенная яркость фона.

Вызов: *АХ=* 1003h, *BL* = назначение 7-го бита атрибута:

0 - повышенная яркость, 1 - мерцание *(устанавливается по умолчанию).*

Функция воздействует сразу на все символы экрана, у которых установлен старший бит атрибута фона.

**Функция 13h.** Запись строки символов с заданными атрибутами.

Записывает строку в текущую страницу видеобуфера, начиная с указанной позиции. Коды *ASCII: 07h -* звонок, *08h* - шаг назад, *0Ah -* перевод строки, *0Dh -* возврат каретки, рассматриваются как управляющие, остальные - как символьные.

Вызов: *АН = I3h, AL* = режим записи:

1. — атрибут символа в *BL,* строка содержит только коды символов, после записи курсор принимает исходное положение (т.е. вывод следующей строки, если не изменить позицию курсора, начинается с изначально установленной позиции);
2. - отличается от режима 0 тем, что после записи курсор остаётся в конце строки;
3. - строка содержит попеременно коды символов и атрибутов (т.е. каждый символ описывается 2 байтами — *ASCII-кодом* и атрибутом), после записи курсор принимает исходное положение;
4. - отличается от режима 2 тем, что по окончании вывода курсор остаётся в конце строки.

ВН = номер страницы (0,1,.. .7), BL = атрибут для режимов 0 и 1,

СХ - длина символьной строки (в длину входят только коды символов, но не байты атрибутов),

DX = DH.DL — координаты курсора (строка, столбец) в исходной точке вывода строки на экране,

ES:BP = адрес начала строки в памяти.

**Пример программы индивидуального задания.**

Нарисовать прямоугольник зелёного цвета в любом месте экрана

.model small

.stack

100h

VGA\_mode equ 13h ; 320x200 256 цветный графический режим

color equ 2 ; цвет линий

x\_sise equ 300 ; ширина прямоугольника в пикселах

y\_sise equ 100 ; высота прямоугольника в пикселах

x\_pos equ 10 ; положение нижнего левого угла прямоуг-ка

y\_pos equ 50

.code

start:

set\_mode:

mov ah,00h ; вызов нулевой функции BIOS

mov al,VGA\_mode ; и инициализация графического режима

int 10h

set\_proc:

mov ah,0Ch ; настройка параметров для вызова функции 0Ch

mov al,color

mov cx,x\_pos

mov dx,y\_pos

http://gendocs.ru/docs/28/27893/conv_1/file1_html_m51a09a60.gifline\_1: ;

int 10h

inc cx

cmp cx,(x\_pos + x\_sise)

http://gendocs.ru/docs/28/27893/conv_1/file1_html_m6e7253b0.gifjne line\_1

line\_2: ;

int

int10h

inc dx

cmp dx,(y\_pos + y\_sise)

jne line\_2;

http://gendocs.ru/docs/28/27893/conv_1/file1_html_m53705967.gifline\_3: ;

int 10h

dec cx

cmp cx,x\_pos

http://gendocs.ru/docs/28/27893/conv_1/file1_html_m2f29c9f7.gifjne line\_3;

line\_4: ;

int 10h

dec dx

cmp dx,y\_pos

jne line\_4

anykey: ; блок отвечающий за завершение приложения

mov ah,1 ; при нажатии любой клавиши

int 16h ; вызов 16h прерывания BIOS, определения

jz anykey ; наличия введенного символа

int 21h

end start ; завершене программы

**2.4. Варианты индивидуального задания:**

1. Разработать программу рисования зеленого треугольника в любом месте экрана.
2. Разработать программу рисования красного прямоугольного треугольника  
   в нижнем правом углу экрана.
3. Разработать программу рисования синего равнобедренного треугольника в верхнем правом углу экрана.
4. Разработать программу рисования белого ромба в любом месте экрана.
5. Разработать программу рисования желтого квадрата в центре экрана.
6. Разработать программу рисования белой трапеции в любом месте экрана.
7. Разработать программу рисования синего параллелограмма в любом месте экрана.
8. Разработать программу рисования двух любых букв русского алфавита в любом месте экрана.
9. Разработать программу рисования красного закрашенного квадрата в центре экрана.