

# Лабораторная работа №1. Текстовый режим работы дисплея

Большинство программ требуют вывод данных в удобном формате на экран. Все необходимые экранные операции можно выполнить используя команду INT 10H, которая передает управление непосредственно в BIOS и затем возвращает управление в прерванную программу для продолжения работы. Функция, которую следует выполнить, и другие параметры необходимые для её выполнения, передается в BIOS через регистры. Таким образом, перед вызовом прерывания необходимо в соответствующие регистры загрузить необходимые данные. Например, чтобы вывести на экран символ, необходимо указать номер функции, которая выводит символ, номер видео страницы, на которую будет выведен символ, и ASCII код этого символа. Каждая команда требует свои параметры, все они будут описаны ниже.

## Установка видеорежима

Установка видеорежима для выполняемой в текущий момент программы осуществляется с помощью функции 00h программного прерывания BIOS INT 10H. Данная функция позволяет переключать цветной монитор в текстовый или графический режим. Содержимое регистра AL в момент вызова прерывания определяет видеорежим, который будет установлен после выполнения операции, и может принимать следующие значения:

- 00h – 40 x 25 черно-белый текстовый режим;
- 01h – 40 x 25 стандартный 16-цветовой текстовый режим;
- 02h – 80 x 25 черно-белый текстовый режим;
- 03h – 80 x 25 стандартный 16-цветовой текстовый режим;
- 04h – 320 x 200 стандартный 4-цветовой графический режим;
- 05h – 320 x 200 черно-белый графический режим;
- 06h – 640 x 200 черно-белый графический режим;
- 07h – 80 x 25 черно-белый стандартный монохромный;
- 0Dh – 320 x 200 16-цветовой графический режим (EGA);
- 0Eh – 640 x 200 16-цветовой графический режим (EGA);
- 0Fh – 640 x 350 черно-белый графический режим (EGA);
- 10h – 640 x 350 64-цветовой графический режим (EGA).

*Пример 1.* Установить графический режим 320 x 200, 4 цвета.

```
mov ah, 00h ;Указываем номер функции
; выбора графического Режима
mov al, 04h ;Номер режима 320x200, 4 цвета
int 10h ;Вызов прерывания
```

## Установка курсора

Экран можно представить в виде двумерного пространства с адресуемыми позициями, в любую из которых может быть установлен курсор. Обычный видеомонитор, например, имеет 25 строк (нумеруемых от 0 до 24) и 80 столбцов (нумеруемых от 0 до 79).

*Пример 2.* Установить курсор на 5-ю строку и 12-й столбец.

```
mov ah, 02h ;Функция установки курсора
mov bh, 00h ;Видео страница 0
mov dh, 05h ;Номер строки 5
mov dl, 0Ch ;Номер столбца 12
int 10h ;Вызов прерывания
```

Значение 02h в регистре AH, указывает на выполнение функции установки курсора. Значение строки и столбца должны быть, соответственно, в регистрах DH и DL. Номер видео страницы – в регистре BH (обычно 0, для графического режима). Для установки строки и столбца можно также использовать одну команду MOV, с непосредственным двухбайтовым значением:

```
mov dx, 050Ch
```

Для того чтобы сделать курсор невидимым, установите его на 25-ю строку.

## Роллирование окна вверх/вниз

Для роллирования окна вверх и вниз используются функции с номерами, соответственно, 06h и 07h. Также эти функции можно использовать для очистки всего экрана или определенной его области. В регистре AH указывается номер функции (06h или 07h, в зависимости от того куда роллировать), в AL – количество строк, которое будет скроллировано (если 0, то окно будет очищено). В CH,CL – строка, колонка верхнего левого угла окна (считая от 0). В DH,DL – строка, колонка нижнего правого угла окна. В BH – видеоатрибут, используемый для пустых строк (07h – нормальный атрибут черно-белый).

*Пример 3.* Выполнить очистку всего экрана

```
mov ah, 06h ; Функция роллирования
mov al, 00h ; Очистка
mov bh, 07h ; Черно-белый
mov cx, 0000h ; Загрузка левой верхней
; позиции экрана
mov dx, 184Fh ; Правая нижняя позиция
int 10h ; Вызов прерывания
```

В этом примере использован указанный выше способ загрузки двухбайтового значения.

## Вывод символа

Для вывода одного символа на экран могут использоваться две функции: 0Ah и 0Eh. Обе операции выводят указанный символ от позиции курсора, отличие в том, что первая при этом не изменяет положение курсора, а вторая перемещает его вслед за выводом символов. Если выводить 5 символов функцией 0Ah, они будут отображаться на одном и том же месте, затирая собой предыдущий, а если использовать функцию 0Eh, то при каждом выводе курсор будет перемещаться, и в результате символы отобразятся друг за другом, слева направо.

При использовании функции 0Ah, перед вызовом прерывания, необходимо в регистре AH указать номер функции (0Ah), в AL – ASCII код символа, в BH – номер видео страницы, в CX – количество раз. Вывод на экран последовательности различных символов требует организации цикла.

При использовании функции 0Eh, содержимое регистров AL и BH такое же, как и в случае использования функции 0Ah. В регистр AH загружаем 0Eh, а в BL – цвет символа в графическом режиме.

*Пример 4.* Вывести произвольный символ в центр экрана.

```
; Перед выводом необходимо установить курсор в центр
; экрана по примеру 2
mov ah, 0Ah ; Номер функции
mov al, 03h ; В AL код символа «Черви»
mov bh, 00h ; Видеостраница 0
```

```
mov cx, 01h ;Один раз
int 10h ;Вызов прерывания
```

## Байт атрибутов

Байт атрибутов как для монохромного, так и для графического дисплея в текстовом (не графическом) режиме определяет характеристики каждого отображаемого символа. Байт-атрибут имеет следующие 8 бит:

| Атрибут | BL | R | G | B | I | R | G | B |
|---------|----|---|---|---|---|---|---|---|
| № бита  | 7  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Буквы RGB представляют битовые позиции, управляющие красным (red), зеленым (green) и синим (blue) лучем в цветном мониторе. Бит 7 (BL) устанавливает мигание, а бит 3 (I) – уровень яркости.

Для модификации атрибутов можно комбинировать биты следующим образом:

| Двоичный код | Шестнадцатеричный код | Эффект                                      |
|--------------|-----------------------|---------------------------------------------|
| 0000 0000    | 00                    | Неотображаемый (для паролей)                |
| 0000 0111    | 07                    | Белый по черному (обычный)                  |
| 1000 0111    | 87                    | Белый по черному (мигание)                  |
| 0000 1111    | 0F                    | Белый по черному (яркий)                    |
| 0111 0000    | 70                    | Черный по белому (инвертированный)          |
| 1111 0000    | F0                    | Черный по белому (инвертированный мигающий) |

Как видно, меняя атрибуты можно выделять фрагменты выводимого текста, делая его более наглядным и удобным для прочтения, можно сделать вводимый текст невидимым – для запроса ввода паролей, и т.д.

## Вывод символа с атрибутом

Функция 09h. Данная функция аналогична функции 0Ah, описанной выше, с тем отличием, что при использовании этой функции существует возможность указать атрибут выводимых символов.

Атрибут указывается в регистре BL. Содержимое остальных регистров аналогично операции 0Ah.

**Пример 5.** Вывести на экран 5 мигающих «сердечек» в инвертированном виде.

```
mov ah, 09h ;Указываем номер функции
mov al, 03h ;В AL код символа «Черви»
mov bh, 00h ;Номер страницы
mov bl, 0F0h ;Атрибут (мерцание, инверсия)
mov cx, 05h ;5 раз
int 10h ;Вызов прерывания
```

## Методические указания

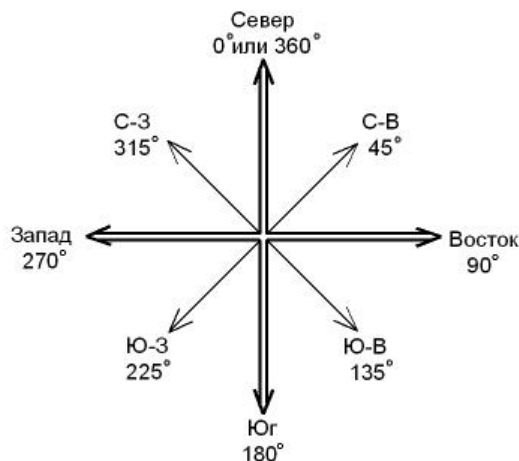
Решение задачи выполняется в FASMW.exe

Вариант определяется по списку преподавателя. Для данного варианта задание определяется по таблицам: цифра определяет направление текста, первая буква – цвет фона, вторая – цвет текста.

Установите стандартный 16-цветовой текстовый режим и выведите ваши имя и фамилию в указанном направлении с заданными цветами текста и фона.

| Направление   | № |
|---------------|---|
| Север         | 1 |
| Юг            | 2 |
| Запад         | 3 |
| Восток        | 4 |
| Северо-восток | 5 |
| Юго-восток    | 6 |
| Северо-запад  | 7 |
| Юго-запад     | 8 |

| Цвет    | № |
|---------|---|
| Красный | a |
| Зеленый | b |
| Синий   | c |
| Желтый  | d |
| Голубой | e |
| Лиловый | f |
| Белый   | g |
| Черный  | h |



## Варианты заданий

- 1) 3fb
- 2) 2ce
- 3) 1db
- 4) 2ge
- 5) 1bc
- 6) 4he
- 7) 7gb
- 8) 8fb
- 9) 3eg
- 10) 4ag
- 11) 6eh
- 12) 5hb
- 13) 7ce
- 14) 8bg
- 15) 2db
- 16) 6bg
- 17) 6gc
- 18) 7eh
- 19) 4ah
- 20) 5ae
- 21) 8fe
- 22) 8ag
- 23) 6ef
- 24) 7ab
- 25) 3db
- 26) 1ef
- 27) 5gd
- 28) 7hb
- 29) 8cb
- 30) 1ca