

ЛАБОРАТОРНАЯ РАБОТА №16

ПЛАН

По дисциплине:	<u>Программирование встроенных систем</u>
Тема занятия:	Работа с изображением
Цель занятия:	научиться выводить графическое изображение на экран, обучение приемам внешнего воздействия на изображение
Количество часов:	2

Содержание работы

Студент самостоятельно выбирает один из вариантов задания

- Используя функции BIOS вывести на экран (VGA-режим) произвольное изображение. Выполнить плавный сдвиг экрана, перемещая начало изображения на одну строку за один раз.
- Используя функции BIOS вывода на экран (SVGA-режим), выполнить изображение в разрешении 1024x768x64K (окрашенная фигура), которое плавно перемещается по экрану стрелками вверх и вниз.
- Воспользовавшись функцией организации задержки написать интерактивную игру «Питон» («Змея», «Червяк»):
 1. Управление осуществлять курсорными клавишами.
 2. Питон погибает, если он выходит за верхнюю или нижнюю границу экрана либо самопересекается.

Дополнительная информация

Графические видеорежимы

Работа с VGA-режимами

Функция 00 прерывания BIOS 10h позволяет переключаться не только в текстовые режимы, но и в некоторые графические. Эти видеорежимы стандартны и поддерживаются всеми видеоадаптерами (начиная с VGA).

Таблица. Основные графические режимы VGA

Номер	Разрешен	Число
11h	640X480	2
12h	640X480	16
13h	320x200	256

Существуют еще несколько видеорежимов, использовавшихся более старыми видеоадаптерами CGA и EGA (с номерами от 4 до 10h).

BIOS также предоставляет видеофункции чтения и записи точки на экране в графических режимах, но эти функции настолько медленно исполняются, что никогда не применяются в реальных программах.

INT 10h AH = 0Ch: Вывести точку на экран

Вход: AH = 0Ch

BH = номер видеостраницы (игнорируется для режима 13h, поддерживающего только одну страницу)

DX = номер строки

CX = номер столбца

AL = номер цвета (для режимов 10h и 11h, если старший бит 1, номер цвета точки на экране будет результатом операции «исключающее ИЛИ»)

Выход: Никакого

INT 10h AH = 0Dh: Считать точку с экрана

Вход: AH = 0Dh

BH = номер видеостраницы (игнорируется для режима 13h, поддерживающего только одну страницу)

DX = номер строки

CX = номер столбца

Выход: AL = номер цвета

Работа с SVGA-режимами

В режиме VGA 320x200 с 256 цветами для отображения видеопамати на основное адресное пространство используется 64000 байт, располагающихся с адреса 0A000h:0000h. Дальнейшее увеличение разрешения или числа цветов приводит к тому, что объем видеопамати превышает максимальные границы сегмента в реальном режиме (65535 байт), а затем и размер участка адресного пространства, отводимого для видеопамати (160 Кб, от 0A000h:0000h до 0B800h:0FFFFh. С адреса 0C800h:000h начинается область ROM BIOS). Чтобы вывести изображение, используются два механизма - переключение банков видеопамати для реального режима и LFB (линейный кадровый буфер) для защищенного.

Во втором случае видеопамать отображается на непрерывный кусок адресного пространства, но начинающегося не с 0A000h, а с какого-нибудь другого адреса, так чтобы весь массив видеопамати, который может занимать несколько мегабайтов, отобразился в одну непрерывную область. В защищенном режиме максимальный размер сегмента составляет 4 Гб, поэтому никаких сложностей с адресацией этого буфера не возникает.

В реальном режиме вывод на экран осуществляется по-прежнему копированием данных в 64-килобайтный сегмент, обычно начинающийся с адреса 0A000h:0000h, но эта область памяти соответствует только части экрана. Чтобы вывести изображение в другую часть экрана, нужно вызвать функцию перемещения окна (или, что то же самое, переключения банка видеопамати), изменяющую область видеопамати, которой соответствует сегмент 0A000h. Например, в режиме 640x480 с 256 цветами требуется 309200 байт для хранения всего видеоизображения. Заполнение сегмента 0A000h:0000h - 0A000h:0FFFFh приводит к закраске приблизительно 1/5 экрана, перемещение окна А на позицию 1 (или переключение на банк 1) и повторное заполнение этой же области - к закраске следующей 1/5 экрана и т. д. Перемещение окна осуществляется подфункцией 05 видеофункции 4Fh или передачей управления прямо на процедуру, адрес которой можно получить, активизировав подфункцию 01. Некоторые видеорежимы позволяют использовать сразу два таких 64-килобайт-ных окна, окно А и окно В, так что можно записать 128 Кб данных, не вызывая прерывания.

Стандартные графические режимы SVGA могут быть 4-, 8-, 15-, 16-, 24- и 32-битными.

4-битные режимы (16 цветов)

VGA

012h:640x480 (64 Кб)

VESA VBE 1.0

102h: 800x600 (256 Кб)

104h: 1024x768 (384 Кб)

106h: 1280x1024 (768 Кб)

Каждый пиксель описывается одним битом, для вывода цветного изображения требуется программирование видеоадаптера на уровне портов ввода-вывода.

8-битные режимы (256 цветов)

VGA

013h: 320x200 (64 Кб)

VBE 1.0

100h: 640x400 (256 Кб)

101h: 640x480 (320 Кб)

103h: 800x600 (512 Кб)

105h: 1024x768 (768 Кб)

107h: 1280x1024 (1,3 Мб)

VBE 2.0

120h: 1600x1200 (1.9 Мб)

Каждый пиксель описывается ровно одним байтом. Значение байта - номер цвета из палитры, значения цветов которой можно изменять, например вызывая подфункцию 09h видеофункции 4Fh.

15-битные режимы (32 К цветов)

VBE 1.2

10Dh: 320x200 (128 Кб)

110h: 640x480 (768 Кб)

113h: 800x600 (1 Мб)

116h: 1024x768 (1,5 Мб)

119h: 1280x1024 (2,5 Мб)

VBE 2.0

121h: 1600x1200 (3,8 Мб)

Каждый пиксель описывается ровно одним словом (16 бит), в котором биты 0-4 содержат значение синей компоненты цвета, биты 5-9 - зеленой, а биты 10-14 - красной. Бит 15 не используется.

16-битные режимы (64 К цветов)

VBE 1.2

10Eh: 320x200 (128 Кб)

111h: 640x480 (768 Кб)

114h: 800x600 (1 Мб)

117h: 1024x768 (1,5 Мб)

11Ah: 1280x1024 (2,5 Мб)

VBE 2.0

122h: 1600x1200 (3,8 Мб)

Так же как и в 15-битных режимах, каждый пиксель описывается ровно одним словом. Обычно биты 0-4 (5 бит) содержат значение синей компоненты, биты 5-10 (6 бит) - зеленой, а биты 11-15 (5 бит) - красной. В нестандартных режимах число битов, отводимое для каждого цвета, может отличаться, так что при их использовании следует вызвать подфункцию 01 видеофункции 4Fh и получить информацию о видеорежиме, включающую битовые маски и битовые смещения для цветов.

24-битные и 32-битные режимы (16 М цветов)

VBE 1.2

10Fh: 320x200 (192 Кб)
112h: 640x480 (1 Мб)
115h: 800x600 (1,4 Мб)
118h: 1024x768 (2,3 Мб)
11Bh: 1280x1024 (3,7 Мб)

В режимах с 24-битным и 32-битным цветом каждому пикселю на экране соответствуют три байта и одно двойное слово (4 байта). Если видеорежим использует модель памяти 6 (Direct Color), то младший байт (байт 0) содержит значение синего цвета, байт 1 содержит значение зеленого, байт 2 - значение красного, а байт 3 - в 32-битных режимах резервный и используется либо для выравнивания, либо содержит значение для альфа-канала. Некоторые видеорежимы могут использовать не Direct Color, а YUV (модель памяти 7) - здесь младший байт соответствует насыщенности красного, байт 1 - насыщенности синего, а байт 2 - яркости.

Видеоадаптер может поддерживать и собственные нестандартные видеорежимы. Список их номеров можно получить, вызвав подфункцию 00h, а получить информацию о режиме по его номеру - вызвав подфункцию 01h видеофункции 4Fh. Более того, для стандартных режимов также следует вызывать подфункцию 01h, чтобы проверить реальную доступность режима (например, режим может быть в списке, но не поддерживаться из-за нехватки памяти). VBE 2.0 разрешает видеоадаптерам не поддерживать никаких стандартных режимов вообще.

INT 10h AH = 4Fh, AL = 00: Получить общую SVGA-информацию
Вход: AX = 4FO0h
ES:DI = адрес буфера (512 байт)
Выход: AL = 4Fh, если функция поддерживается AH = 01, если произошла ошибка AH=00, если данные получены и записаны в буфер

Буфер для общей SVGA-информации:
+00h: 4 байта - будет содержать VESA после вызова прерывания, чтобы получить поля, начиная с 14h, здесь надо предварительно записать строку VBE2 +04h: слово - номер версии VBE в двоично-десятичном формате (0102h - для 1.2, 0200h - для 2.0)
+06h: 4 байта - адрес строки-идентификатора производителя +0Ah: 4 байта - флаги:
бит 0: АЦП поддерживает 8-битные цветовые компоненты (см. подфункцию 08h)
бит 1: видеоадаптер несовместим с VGA
бит 2: АЦП можно программировать только при обратном ходе луча бит 3: поддерживается спецификация аппаратного ускорения графики VBE/AF 1.0
бит 4: требуется вызов EnableDirectAccess перед использованием LFB бит 5: поддерживается аппаратный указатель мыши бит 6: поддерживается аппаратный clipping бит 7: поддерживается аппаратный BitBlt биты 8-31 зарезервированы
+0Eh: 4 байта - адрес списка номеров поддерживаемых видеорежимов (массив слов, последнее слово = 0FFFFh, после которого обычно следует список нестандартных режимов, также заканчивающийся словом 0FFFFh)
+12h: слово - объем видеопамати в 64-килобайтных блоках + 14h: слово - внутренняя версия данной реализации VBE
+16h: 4 байта - адрес строки с названием производителя
+1Ah: 4 байта - адрес строки с названием видеоадаптера
+1Eh: 4 байта - адрес строки с версией видеоадаптера
+22h: слово - версия VBE/AF (BCD, то есть 0100h для 1.0)

+24h: 4 байта - адрес списка номеров режимов, поддерживающих аппаратное ускорение (если бит поддержки VBE/AF установлен в 1)
+28h: 216 байт- зарезервировано VESA
+100h:256 байт- зарезервировано для внутренних данных VBE. Так, например, в эту область копируются строки с названиями производителя, видеоадаптера, версии и т. д.

INT 10h AH = 4Fh, AL = 01: Получить информацию о режиме
Вход: AX = 4F01h
CX = номер SVGA-режима (бит 14 соответствует использованию LFB, бит 13 - аппаратному ускорению)

ES:DI = адрес буфера для информации о режиме (256 байт)

Выход: AL = 4Fh, если функция поддерживается

AH = 01h, если произошла ошибка

AH = 00h, если данные получены и записаны в буфер

Буфер для информации о SVGA-режиме:

+00h: слово - атрибуты режима:

бит 0: режим присутствует

бит 1: дополнительная информация (смещения 12h - 1Eh) присутствует (для VBE 2.0 эта информация обязательна и бит всегда установлен)

бит 2: поддерживается вывод текста на экран средствами BIOS

бит 3: режим цветной

бит 4: режим графический

бит 5: режим несовместим с VGA

бит 6: переключение банков не поддерживается

бит 7: LFB не поддерживается

бит 8: не определен

бит 9: (для VBE/AF) приложения должны вызвать EnableDirectAccess, прежде чем переключать банки

+02h: байт - атрибуты окна A:

бит 1: окно существует

бит 2: чтение из окна разрешено

бит 3: запись в окно разрешена

+03h: байт - атрибуты окна B

+04h: слово - гранулярность окна - число килобайтов, которому всегда кратен адрес начала окна в видеопамяти (обычно 64)

+06h: слово - размер окна в килобайтах (обычно 64)

+08h: слово - сегментный адрес окна A (обычно 0A000h)

+0Ah: слово - сегментный адрес окна B

+0Ch: 4 байта - адрес процедуры перемещения окна (аналог подфункции 05h, но выполняется быстрее)

+10h: слово - число целых байтов в логической строке

+12h: слово - ширина в пикселах (для графики) или символах (для текста)

+14h: слово - высота в пикселах (для графики) или символах (для текста)

+16h: байт - высота символов в пикселах

+17h: байт - ширина символов в пикселях

+18h: байт - число плоскостей памяти (4 - для 16-цветных режимов, 1 - для обычных, число переключений банков, требуемое для доступа ко всем битам (4 или 8), - для модели памяти 5)

+19h: байт - число битов на пиксел

+1Ah: байт - число банков для режимов, в которых строки группируются в банки (2 - для CGA, 4 - для HGC)

+1Bh: байт - модель памяти:

00h - текст
 01h - CGA-графика
 02h - HGC-графика
 03h - EGA-графика (16 цветов)
 04h - VGA-графика (256 цветов в одной плоскости)
 05h - Режим X (256 цветов в разных плоскостях)
 06h - RGB (15-битные и выше)
 07h - YUV
 08h - OFh - зарезервированы VESA
 10h - FFh - нестандартные модели
 +1Ch: байт - размер банка в килобайтах (8 - для CGA и HGC, 0 - для остальных)
 +1Dh: байт - число видеостраниц
 +1Eh: байт-зарезервирован
 +1Fh: байт - битовая маска красной компоненты
 +20h: байт - первый бит красной компоненты
 +21h: байт - битовая маска зеленой компоненты
 +22h: байт - первый бит зеленой компоненты
 +23h: байт - битовая маска синей компоненты
 +24h: байт - первый бит синей компоненты
 +25h: байт - битовая маска зарезервированной компоненты
 +26h: байт - первый бит зарезервированной компоненты
 +27h: байт - бит 0: поддерживается перепрограммирование цветов (подфункция
 09h) бит 1: приложение может использовать биты в зарезервированной компоненте
 +28h: 4 байта - физический адрес начала LFB
 +2Ch: 4 байта - смещение от начала LFB, указывающее на первый байт после
 конца участка памяти, отображающейся на экране
 +30h: слово - размер памяти в LFB, не отображающейся на экране, в килобайтах
 +32h: 206 байт - зарезервировано

INT 10h AH = 4Fh, AL = 02: Установить режим

Вход: AX = 4F02h

BX = номер режима:

биты 0-6: собственно номер режима

бит 7: видеопамять не очищается при установке режима, если все следующие биты - нули

бит 8: стандартный VBE SVGA-режим

бит 9: нестандартный SVGA-режим

биты 10-12: зарезервированы

бит 13: режим использует аппаратное ускорение

бит 14: режим использует LFB

бит 15: видеопамять не очищается при установке режима

Кроме того, специальный номер режима 81 FFh соответствует доступу ко всей видеопамети и может использоваться для сохранения ее содержимого.

Выход: AL = 4Fh, если функция поддерживается

АН = 00, если режим установлен

АН = 01 или 02, если произошла ошибка

INT 10h AH = 4Fh, AL = 03: Узнать номер текущего видеорежима

Вход: AX = 4F03h

Выход: AL = 4Fh, если функция поддерживается

BX = номер режима

INT 10h AH = 4Fh AL = 05: Перемещение окна (переключение банка видеопамати)

Вход: AX = 4F05H

BH = 00 - установить окно

BH = 01 - считать окно

BL = 00 - окно A

BL = 01 -окно B

DX = адрес окна в единицах гранулярности (номер банка), если BH = 0

Выход: AL = 4Fh, если функция поддерживается

DX = адрес окна в единицах гранулярности (номер банка), если BH = 1

AH = 03, если функция была вызвана в режиме, использующем LFB

Всегда предпочтительнее переключать банки прямым дальним вызовом процедуры, адрес которой возвращается подфункцией 01h в блоке информации о видеорежиме. Все параметры передаются в процедуру точно так же, как и в подфункцию 05h, но содержимое регистров AX и DX по возвращении не определено.

INT 10h AH = 4Fh AL = 07: Установка начала изображения

Вход: AX = 4F07h BH = 00

BL = 00 - считать начало изображения

BL = 80h - установить начало изображения (в VBE 2.0 автоматически выполняется при следующем обратном ходе луча)

CX = первый изображаемый пиксель в строке (для BL = 80h)

DX = первая изображаемая строка (для BL = 80h)

Выход: AL = 4Fh, если функция поддерживается

AH = 01, если произошла ошибка

AH = 00, если функция выполнялась успешно

BH = 00 (для BL = 00)

CX = первый изображаемый пиксель в строке (для BL = 00)

DX = первая изображаемая строка (для BL = 00)

С помощью этой функции можно выполнять как плавный сдвиг экрана, перемещая начало изображения на одну строку за один раз, так и быстрый показ двух разных изображений, изменяя одно, пока на экране находится другое, — своего рода эффект плавной анимации.