ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
(РОСАВИАЦИЯ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра вычислительных машин, комплексов, систем и сетей

Н.И. Черкасова

ПРОГРАММИРОВАНИЕ НА МАШИННО-ОРИЕНТИРОВАННОМ ЯЗЫКЕ

Учебно-методическое пособие
по выполнению лабораторной работы № 2

для студентов II курса
направления 09.03.01
очной формы обучения

Москва
ИД Академии Жуковского
2021

УДК 004.43
ББК 6Ф7.3
Ч-48

Рецензент:
Феоктистова О.Г. – д-р техн. наук, доцент

Черкасова Н.И.

Ч-48 Программирование на машинно-ориентированном языке [Текст] : учебно-методическое пособие по выполнению лабораторной работы № 2 / Н.И. Черкасова. – М.: ИД Академии Жуковского, 2021. – 32 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины «Программирование на машинно-ориентированном языке» по учебному плану для студентов II курса направления 09.03.01 очной формы обучения.

Рассмотрено и одобрено на заседаниях кафедры 30.08.2021 г. и методического совета 30.08.2021 г.

УДК 004.43
ББК 6Ф7.3

В авторской редакции

Подписано в печать 27.10.2021 г.
Формат 60x84/16 Печ. л. 2 Усл. печ. л. 1,86
Заказ № 855/1004-УМП18 Тираж 30 экз.

Московский государственный технический университет ГА
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского
125167, Москва, 8-го Марта 4-я ул., д. 6А
Тел.: (495) 973-45-68
E-mail: zakaz@itsbook.ru

Содержание

1. Цель лабораторной работы 4
2. Содержание отчета 4
3. Краткие теоретические сведения 4
3.1. Видеосистема компьютера 4
3.2. Типы видеоадаптера 7
3.3. Режимы работы монитора 9
3 3.1.Текстовый видеорежим 9
3.3.2. Графический режим 13
3.3.3.Функции BIOS для обслуживания видеосистемы компьютера 14
3.3.4.Прерывание int 10h. Видеофункции BIOS 17
3.3.5. Прерывание int 16h 23
3.3.6.Задержка программных операций 25
Задание на выполнение 27
Выполнение лабораторной работы 29
Список вопросов 29
Литература 29
Приложение1 30

ЛАБОРАТОРНАЯ РАБОТА№2 «Графика и расширенные средства ввода/вывода в Ассемблере»

1. Цель лабораторной работы

Целью лабораторной работы является освоение:

1. Способов осуществления ввода с клавиатуры и вывода на экран символьной информации с использованием функций DOS. Графический режим работы процессора.
2. Перехода в графический режим.
3. Особенности прерывания int 21.
4. Особенности прерывания int 10.
5. Особенности прерывания int 16.

2. Содержание отчета

Отчет по лабораторной работе должен включать:

1) цель лабораторной работы;
2) конкретный вариант задания на выполнение;
3) тексты программ;
4) схемы алгоритмов;
5) результаты выполнения программ.

3. Краткие теоретические сведения

3.1. Видеосистема компьютера

Персональный компьютер смог стать привлекательным вычислительным средством благодаря интерактивности взаимодействия с пользователем. Основной поток исходной информации ПК визуальный, причем информация представляется как в текстовом, так и в графическом виде. Рассмотрим механизмы этого представления.

Видеосистема современного компьютера служит для вывода текстовой и графической информации на монитор. В общем случае видеосистема состоит из трех компонент:

1. монитор (дисплей);
2. видеоадаптер;
3. программное обеспечение (драйверы видеосистемы).

Видеоадаптер посылает в монитор сигналы управления яркостью лучей и синхросигналы строчной и кадровой развёрток. Монитор преобразует эти сигналы в зрительные образы. А программные средства обрабатывают видеоизображения - выполняют кодирование и декодирование сигналов, координатные преобразования, сжатие изображений и др.

Видеосистема компьютера формирует изображение программно, но при этом видеоадаптер (видеокарта) отвечает за формирование изображения и монитора, на который это изображение выводится.

Рассмотрим основной элемент видеоподсистемы-видеоадаптер.

Видеокарта (также видеоадаптер, графический адаптер, графическая плата, графическая карта, графический ускоритель) - устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера (или самого адаптера), в форму, пригодную для дальнейшего вывода на экран монитора.

Видеоадаптер отвечает за вывод изображения из видеопамяти, обновление ее содержимого, формирование сигналов для монитора и обработку запросов центрального процессора, который задает необходимый поток информации для вывода.

Видеоадаптер является важнейшим элементом видеосистемы, поскольку определяет следующие ее характеристики:

1. максимальное разрешение и частоту разверток (также зависит от возможностей монитора);
2. максимальное количество отображаемых цветов и оттенков (палитра);
3. скорость обработки и передачи видеоданных.

Для работы видеоадаптера необходимы следующие основные компоненты:

1. BIOS (Basic Input/Output System - базовая система ввода вывода);
2. графический процессор;
3. видеоконтроллер;
4. видеопамять;
5. цифроаналоговый преобразователь, он же DAC (Digital to Analog Converter);
6. видео-ПЗУ;
7. разъем;
8. система охлаждения;
9. видеодрайвер.

Видеоадаптеры имеют свою BIOS, которая подобна системной BIOS, но полностью независима от нее. Другие устройства в компьютере, такие, как адаптеры SCSI, могут также иметь собственную BIOS.

Графический процессор, или набор микросхем, характеризует быстродействие адаптера и его функциональные возможности.

В основном графический процессор (Graphics processing unit - графическое процессорное устройство) - занимается расчётами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчёты для обработки команд трёхмерной графики.

Видеоконтроллер - устройство, отвечающее за формирование изображения в видеопамяти, подаёт команды системе RAMDAC (Random Access Memory Digital-to-Analog Converter) на формирование сигналов развёртки для монитора и осуществляет обработку запросов центрального процессора.

Видеопамять - устройство, которое выполняет роль кадрового буфера, в котором хранится изображение, генерируемое и постоянно изменяемое графическим процессором и выводимое на экран монитора (или нескольких мониторов). В видеопамяти хранятся также промежуточные невидимые на экране элементы изображения и другие данные.

Цифро-аналоговый преобразователь (ЦАП, RAMDAC - Random Access Memory Digital-toAnalog Converter) - служит для преобразования изображения, формируемого видеоконтроллером, в уровни интенсивности цвета, подаваемые на аналоговый монитор. Цифроаналоговый преобразователь DAC (ранее используемый в качестве отдельной микросхемы) зачастую встраивается в графический процессор новых наборов микросхем, поэтому необходимость в подобном преобразователе в полностью цифровых системах - цифровая видеокарта плюс цифровой монитор - отпадает, однако, для аналогового интерфейса VGA и аналогового монитора, DAC еще некоторое время используется.

Видео-ПЗУ (Video ROM) - постоянное запоминающее устройство, в которое записаны видео BIOS, экранные шрифты, служебные таблицы и т.п. ПЗУ не используется видеоконтроллером напрямую - к нему обращается только центральный процессор.

Система охлаждения - набор средств для отвода тепла от нагревающихся в процессе работы компьютерных компонентов.

Видеодрайвер выполняет функции интерфейса между системой с запущенными в ней приложениями и видеоадаптером. Так же, как и видео-BIOS, видеодрайвер организует и программно контролирует работу всех частей видеоадаптера через специальные регистры управления, доступ к которым происходит через соответствующую шину.

3.2. Типы видеоадаптера

Видеоадаптер, является важнейшим элементом видеосистемы, поскольку определяет ее характеристики. В таблице 1 представлены некоторые характеристики различных типов видеоадаптеров.

Таблица 1- Характеристики видеоадаптера

Адаптер	Характеристики
MDA (монохромный дисплейный адаптер)	Разработан фирмой IBM, является одним из самых ранних адаптеров, может воспроизводить лишь алфавитно-цифровую информацию и небольшое количество служебных символов. В нем отсутствуют графические возможности. Обеспечивает разрешающую способность 25 80 символов, размер точечной матрицы символа 9 14 пикселей
CGA (цветной графический адаптер)	Обеспечивает воспроизведение информации только со средним разрешением и ограниченным количеством цветов (был предназначен для работы с цифровыми RGB-мониторами). Обеспечивает разрешающую способность 25½80 символов, размер точечной матрицы символа 8 8 пикселей
CGA (цветной графический адаптер	Обеспечивает воспроизведение информации только со средним разрешением и ограниченным количеством цветов (был предназначен для работы с цифровыми RGB-мониторами). Обеспечивает разрешающую способность 25½80 символов, размер точечной матрицы символа 8 8 пикселей
MGA (монохромный графический адаптер)	Обладает характеристиками, аналогичными MDA
EGA (улучшенный графический адаптер)	Разработан в 1984 г., оснащен видеопамятью емкостью 64, 128 или 256 Кбайт.

	Адаптер разрабатывался для монитора RGBrgb, способного воспроизводить 64 цвета. Но малый объем видеопамяти позволял работать с четырьмя палитрами по 16 цветов
VGA (видеографическая матрица	Разработан в 1988 г., позволял реализовать 480 640 точек в графическом режиме при 64–256 одновременно отображаемых цветах (в зависимости от объема видеопамяти) из 262 144 возможных. В текстовом режиме адаптер VGA позволяет отображать на экране 25 80 или 50 80 символов. Количество цветов, отображаемых в этом режиме, ограничивалось 16 цветами из 256 возможных. Количество воспроизводимых цветов ограничивает архитектура адаптера с целью сделать его совместимым с адаптером EGA
SVGA (супервидео графический массив)	Практически все SVGA-видеоадаптеры следуют стандарту VESA. Наиболее распространенные видеорежимы: 600 800, 768 1024, 1024 1280, 1200 1600

Дальнейшее развитие подтипов видеоадаптеров основано на стандарте VGA, например, SVGA, XGA и т. д.

Современный видеоадаптер-это сложное почти самостоятельное устройство, представляющее собой мини-компьютер. Помимо своей основной задачи он способен выполнять ряд дополнительных функций: аппаратное ускорение 2D и 3D-графики, обработку видеоданных, прием теле- и видеосигналов и многое другое. Раньше все эти дополнительные функции реализовывались на отдельных платах и подсоединялись к видеоадаптеру как дочерние карты или с помощью локальных интерфейсных шин. Сейчас используется метод интеграции all-in-one, когда все эти функции реализуются в одном графическом чипе видеоадаптера. Современный видеоадаптер значительно отличается по своему функциональному составу от видеоадаптера VGA (более старые не используются), но его основное назначение осталось прежним: сканирование и цифро-аналоговое преобразование содержимого кадрового буфера с последующим формированием непрерывного трехканального RGB-сигнала.

Видеоадаптер VGA был пассивным устройством, не принимавшем участие в формировании содержимого кадрового буфера и не обрабатывавшем микрокоманды преобразования цифровых данных. Современный интегрированный видеоадаптер также использует:

1. Графические акселераторы обработки двумерной и трехмерно графики большой разрядности.
2. Быстродействующую видеопамять.
3. Высокоскоростные шины интерфейса.

Такой видеоадаптер называют видеокартой, хотя это название не совсем правильно и неточно.

Большинство из перечисленных элементов видеоадаптера содержат специальные регистры (8 разрядов и более), доступные центральному процессору (CPU) для чтения и записи данных. Эти регистры содержат конфигурационную и статусную информацию и предназначены для управления работой соответствующих элементов видеоадаптера. Модифицируя их содержимое, CPU может управлять работой видеоадаптера.

Помимо этих регистров, в состав элементов видеоадаптера входят несколько специальных регистров. Выходной регистр предназначен для задания адресов портов ввода/вывода, а также начальных адресов кадрового буфера и выбора тактового генератора. Регистр состояния используется для синхронизации процесса обновления кадрового буфера с сигналами обратного хода кадровой развертки.

Все элементы, за исключением видеопамяти, Video BIOS, тактовых генераторов и шин интерфейса реализованы в одной микросхеме.

3.3.Режимы работы монитора

3 3.1.Текстовый видеорежим

Различие между текстовым и графическим режимами работы монитора заключается в возможностях управления выводом визуальной информации. В текстовом режиме минимальным объектом, отображаемым на экране, является символ, алфавитно-цифровой или какой-либо иной. В обычных условиях экран монитора, работающего в текстовом режиме (алфавитно-цифрового дисплея), может содержать не более 80 символов по горизонтали и 25 символов по вертикали, то есть всего 2000 визуальных объектов. При этом имеются ограниченные возможности по управлению цветом символов. Конечно, в таком режиме можно выводить на экран не только обычный текст, но и некие графические изображения (например, таблицы), однако понятно,

что качество таких изображений будет посредственным. Для полноценной работы с изображениями текстовый режим дисплея абсолютно не подходит.

Текстовый видеорежим - режим компьютерного видеоадаптера, в котором экран представлен в виде решётки знакомест (а не пикселей, в отличие от графических режимов). В каждом из знакомест может находиться один символ из ограниченного набора.

Текстовые видеотерминалы начали заменять телетайпы в начале 1970-х годов и изменили способ ведения диалога оператора с компьютером. Вместо командной строки появился текстовый интерфейс пользователя; в шрифты начали вводить псевдографические символы для рисования рамок и имитации элементов графического интерфейса. Одна из кодировок с псевдографикой - CP437 (IBM).

Текстовый режим даёт превосходство над графическим в скорости и простоте программирования. К тому же в те времена (1970-е годы) считалось расточительством ставить в терминал столько видеопамяти, чтобы хранить каждый пиксель экрана. В текстовом режиме изображение генерируется динамически из матрицы знакомест и изображений символов с помощью особой схемы - знакогенератора. На более ранних ЭВМ использовались также знакопечатающие кинескопы, которые генерировали символы без сложных схем знакогенератора, используя трафарет. Вторым преимуществом текстовых интерфейсов, связанным с терминалами, стали низкие требования к скорости связи терминала и ЭВМ.

Поскольку изображение представляет собой матрицу символов, шрифт в текстовом режиме, естественно, может быть только моноширинным - примерно таким же, как и в пишущих машинках. Таким же образом работают АЦПУ барабанного типа - так что изображение с экрана можно без проблем отправлять на печать. Более новые устройства печати (матричные принтеры) имитировали эту черту АЦПУ.

Многие ОС позволяют не только эмулировать телетайп, но и писать в любое возможное знакоместо. Для этого есть два стандарта: ANSI-графика и команды, совместимые с VT100.

Грань между текстовыми и графическими режимами размыта: например, некоторые программы (Norton Utilities) динамически переопределяют шрифт, чтобы отображать графические знаки или графический курсор мыши. Компьютер «Корвет» мог одновременно выводить текст поверх графического изображения. Иногда текстовый режим из-за его скорости применялся и в играх. Недокументированный 16-цветный графический режим CGA 160×100 с точки зрения программирования являлся текстовым режимом.

Интерфейс командной строки и эмуляторы терминала имитируют поведение компьютера в текстовом режиме.

Таблица 2 - Текстовые режимы на IBM-совместимом компьютере

Разре-шение	Кол-воцветов	Размер символа	Графическое разрешение	Адаптеры
80×25	Чёрно-белый	9×14	720×350	MDA, Hercules
		CGA/EGA/VGA также поддерживают этот режим, качество эквивалентно 80×25, 16 цветов		
40×25	16 цветов	8×8	320×200	CGA и выше
80×25	16 цветов	8×8	640×200	CGA и выше
		8×14	640×350	EGA и выше
		9×16	720×400	VGA
80×43	16 цветов	8×8	640×350	EGA и выше
80×30	16 цветов	8×16	640×480	VGA
80×34	16 цветов	8×14	640×480	VGA
80×50	16 цветов	9×8	720×400	VGA
80×60	16 цветов	8×8	640×480	VESA-совместимые Super VGA
132×25	16 цветов			VESA-совместимые Super VGA

132×43	16 цветов			VESA-совместимые Super VGA
132×50	16 цветов			VESA-совместимые Super VGA
132×60	16 цветов			VESA-совместимые Super VGA

Видеоадаптер, способный работать в текстовом режиме, имеет две особых области видеопамяти - текстовый буфер и шрифт. Шрифт - изображения всех возможных символов (как правило, битовые). Текстовый буфер - массив по количеству знакомест. Для каждого из знакомест в текстовом буфере хранятся код символа и дополнительная информация - атрибут. В зависимости от модели адаптера, атрибут может хранить цвета символа и фона, флаги инверсии, яркости, подчёркивания, мигания, девятый бит кода символа.

Работой текстового режима управляет схема видеоадаптера, именуемая знакогенератор.. В видеоадаптере есть два счётчика: строк (Y) и пикселей в строке (X). Эти координаты делятся с остатком на размер знакоместа. Частные - координаты в текстовом буфере; остатки - координаты в шрифте. Если размеры знакоместа - степени двойки[en], то деление с остатком представляет собой просто отсечение верхних и нижних битов.

Координаты в текстовом буфере направляются в текстовый буфер. Тот возвращает код символа и атрибут. Код символа, X и Y в шрифте проходят через шрифтовую память, которая возвращает один бит - 0, если в этой позиции фон, и 1, если изображение. Схема применения атрибута (на рисунке справа не указана) превращает линии атрибута и этот бит в окончательный сигнал, пригодный к прогонке через ЦАП. В простейшем случае эта схема - мультиплексор на два входа по 4 бита, переключающий между цветом изображения и цветом фона. Эта же схема рисует текстовый курсор.

Шрифт хранится, в зависимости от модели видеоадаптера, в ПЗУ или ОЗУ. В последнем случае шрифт можно переопределить - это позволяет русифицировать компьютер или, изменяя шрифт синхронно с развёрткой, делать пиксельную графику.

В некоторых текстовых режимах (например, на том же VT100) существуют и атрибуты строк. Строка может иметь двойную ширину.

Режим, ранее применяемый как в DOS, так и в консольных программах Windows - 80×25 символов, 16 цветов. 40-символьные режимы использовались в играх и на телевизорах. Размеры символов в SVGA-режимах зависят от производителя. Также SVGA позволяют уменьшить количество цветов с 16 до 8, зато выводить целых 512 разных символов. Некоторые платы (например, S3) поддерживают огромные текстовые режимы (до 160×120). Чтобы работать с такими режимами в консолях Linux, применялась программа SVGATextMode.

Некоторые из современных графических программ моделируют те или иные дизайнерские ходы текстового ПО. Существует ПО, эмулирующее текстовые режимы: эмуляторы терминала или консоль ОС. Иногда (например, в Windows) эмулированную консоль можно переключить в настоящий текстовый режим (нажатием Alt+ Enter).

В Приложении 1 приведен разбор примера программы осуществления ввода с клавиатуры и вывода на экран символьной информации в реальном режиме.

Виртуальная консоль Linux работает в текстовом режиме. Большинство вариантов Linux поддерживают несколько консолей, между которыми можно переключаться нажатием Ctrl+Alt+F1, F2 и т. д.

3.3.2. Графический режим

В графическом режиме минимальным объектом, выводом которого может управлять программист, является так называемый пиксель (от английского pixel, возникающего в результате объединения слов «рисунок» (picture) и «элемент» (element). Пиксель представляет собой точку с тремя цветами. Его геометрические размеры определяются разрешением монитора. Разрешение монитора обычно задается в виде rx * ry, где rx-количество пикселей на экране по горизонтали, а ry-количество пикселей по вертикали. На практике используются не произвольные, а некоторые определенные значения разрешения. Такими разрешениями являются, например, 320x200, 640x480, 800x600, 1024x768, 1280x1024 и т.д. Даже в случае самого грубого разрешения изображение в графическом режиме формируется с помощью 64000 графических элементов.

Размер экрана-величина фиксированная. Если величина диагонали экрана 14 дюймов, его геометрические размеры составляют примерно 28x20 см. Размер пикселя можно приблизительно получить, разделив размер экрана на разрешение. Геометрические размеры пикселя определяют степень детализации изображения, его качество. Имеется минимально допустимое

значение размера пикселя, определяемое техническими параметрами монитора.

Монитор - устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и др.).

Графический режим монитора предназначен для вывода на экран графиков, рисунков и других графических элементов. Очевидно, что в этом режиме можно выводить также и текстовую информацию в виде различных надписей, которые могут иметь произвольный шрифт, размер букв и т.д.

В лабораторной работе № 1 рассматривались различные системные функции DOS вывода на экран символьной информации. Однако возможности DOS весьма ограничены: она не имеет функций для изменения цвета выводимых символов и позиционирования курсора. Кроме того, в DOS отсутствуют средства формирования графических изображений

Итак, выводить строки на экран можно двумя путями:

1. Посимвольно (то есть в цикле выводить каждый символ строки).
2. Строку целиком.

В текстовом режиме вывод на экран можно выполнить одним из трёх способов:

1. С помощью функций DOS.
2. С помощью функций BIOS.
3. Путём прямой записи в видеопамять.

Третий способ отличается тем, что он сразу записывает данные в видеопамять, что позволяет выполнять вывод быстро. Однако он практически применим, так как современные операционные системы не позволяют напрямую обращаться к аппаратуре (реальный режим).

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.

3.3.3.Функции BIOS для обслуживания видеосистемы компьютера

Рассмотрим функции BIOS для обслуживания видеосистемы компьютера, а также функции для работы с клавиатурой.

Функции BIOS для работы в графическом режиме.

Прерывание Int 10h:

функция 00h-установка видеорежима (см. ПриложениеТабл.1);

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.

Функции BIOS для работы с клавиатурой.

Прерывание 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.

Прерывание Int 15h, функция 86h-задержка.

Прерывание Int 1Ah, функция 00h-получение системного времени.

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);
- TrueColor (24-битное / 32-битное кодирование).

Графические режимы VGA (SVGA) устарели, а текстовые продолжают применяться. В таблице 3 представлены коды цветов стандартной палитры.

Таблица 3 - Коды цветов стандартной палитры

Код	Цвет		Код	Цвет
0h	Чёрный		8h	Серый
1h	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

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. В текстовом режиме для VGA-совместимых систем для видеопамяти отводится адресное пространство (исключая 7-й видеорежим с монохромным адаптером), начинающееся с логического адреса B800h:0000h и заканчивающееся адресом BF00h:0FFFh. Данная область разбивается на 8 секторов по числу видеостраниц (4 Кбайта на страницу). Таким образом, постраничное деление адресного пространства видеопамяти в текстовом режиме имеет следующий вид:

- B800h:0000h-страница 0, смещение в диапазоне 0000h-0FFFh
- B900h:0000h-страница 1, смещение в диапазоне 0000h-0FFFh
- ……
- BF00h:0000h-страница 7, смещение в диапазоне 0000h-0FFFh

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. При этом по адресу B800h:0000h находится байт с кодом символа (левый верхний угол экрана), а в B800h:0001h-атрибут этого символа; B800h:0002h-код второго символа, а в B800h:0003h-атрибут второго символа и т.д. 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 файла атрибутов приведена на рис. 1.

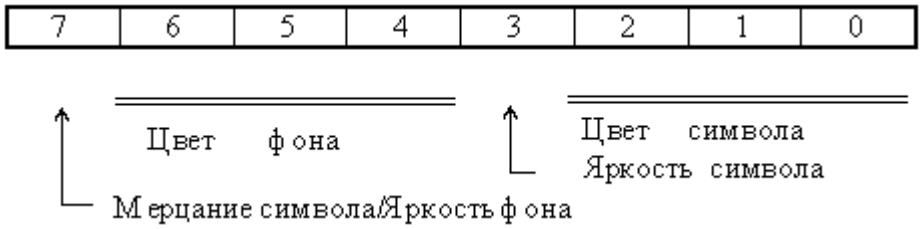

Рис.1. Структура байта атрибутов

Из рисунка 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 ОС DOS по умолчанию).

При загрузке устанавливается стандартная палитра, коды цветов которой приведены в табл. 3.

Рассмотрим пример. 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.

3.3.4.Прерывание int 10h. Видеофункции BIOS

В таблице 4 представлены видеорежимы и страницы в стандарте VGA, поддерживаемые видеоконтроллерами

Таблица 4 - видеорежимы и страницы в стандарте VGA

Режим	Тип	Разрешение	Цвет	Размер знака	Адрес	Страницы
0	text	40x25	16 полутонов	9x16	B8000	0-7
1	text	40x25	16/8	9x16	B8000	0-7
2	text	80x25	16 полутонов	9x16	B8000	0-7
3	**text**	**80x25**	**16/8**	**9x16**	**B8000**	**0-7**
6	graphic	640x200 / 80x25	2	8x8	B8000	0

7	text	80x25	3 (Mono)	9x16	B0000	0
10h	graphic	640x350 / 80x25	4 или 16	8x14	A0000	0-1
11h	graphic	640x480 / 80x30	2 (Mono)	8x16	A0000	0
12h	graphic	640x480 / 80x30	16	8x16	A0000	0
13h	graphic	640x480 / 80x30	256	8x8	A0000	0

Для прерывания int 10h устанавливаются следующие функции.

Функция 00h - Установка видеорежима (табл.4) текущей видеостраницы с очисткой экрана (быстрая очистка экрана реализуется функцией 06h и 07h).

Вызов: AH = 00h,

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

Вызов замещает регистры AX, BP, SI, и DI.

По умолчанию в DOS используется режим 3 (следует отметить, что корректно оформленная программа должна выполнять проверку или установку требуемого текстового режима с последующим восстановлением прежнего).

Пример: Установка цветного графического режима

```
MOV AH, 0
MOV AL, 12H
INT 10H
```

Все графические образы состоят из точек (пикселей). Любой графический объект

можно нарисовать с помощью точек, каждый раз меняя координаты вывода точки на экран

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 (табл. 4) независимо от того, какая страница является активной. Функцию 02h BIOS можно использовать в комбинации с функциями DOS для организации вывода на экран.

Вызов: AH = 02h; BH = номер страницы (0,1,...7), обычно 0;

DH = строка; DL = столбец.

Вызов замещает регистры AX, BP, SI и DI.

Пример: Установка курсора в заданную позицию.

MOV AH, 2 ;устанавливаем номер функции

MOV BH, 0 ;номер активной страницы

MOV DH, 13 ;строка установки курсора

MOV DL, 20 ;столбец установки курсора

INT 10H ;позиционируем курсор

AH=9: Вывод символа с атрибутами на экра

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: AH = 03h, BH = номер страницы (0,1,...7), обычно 0.

Возврат: DH, DL = строка и столбец текущей позиции курсора,

CH, CL = первая и последняя строки развёртки курсора.

Вызов разрушает регистры AX, BP, SI и **DI.**

Функция 05h. Установка видеостраницы**.**

Устанавливает активную видеостраницу (как текстовую, так и графическую).

Вызов: AH= 05h, AL= номер страницы (0,...,7).

Вызов разрушает регистры AX, BP, SI и DI.

Программа, устанавливающая страницу, отличную от текущей, обязана по окончании работы восстанавливать исходную.

Функция 06h (07h). Инициализация или прокрутка окна вверх (вниз).

Инициализирует окно с указанными координатами, пробелами ASCII с заданным атрибутом (AL = 0), или прокручивает содержимое окна вверх (вниз) на заданное число строк (AL = число строк). При прокрутке появляющиеся снизу (сверху) строки заполняются пробелами ASCII с заданным атрибутом. Функцию удобно использовать для быстрой очистки экрана или некоторого прямоугольного окна.

Вызов: AH = 06h(07h);

AL = 0-очистка, AL = N (N >0)-прокрутка на N строк;

BH = атрибут символов в окне;

CH, CL = координаты строки и столбца (Y,X) левого верхнего угла;

DH, DL = координаты строки и столбца (Y,X) правого нижнего угла.

Вызов разрушает регистры AX, BP, SI, и DI.

Пример: Очистка окна (скроллинг вверх)

```
MOV AH, 6 ;задаем процедуру скроллинга вверх
MOV AL, 0 ;очищаем все окно
MOV BH, 7 ;байт атрибутов для заполнения
MOV CH, 3 ;строка верхнего левого угла
MOV CL, 4 ;столбец верхнего левого угла
MOV DH, 13 ;строка нижнего правого угла
MOV DL, 15 ;столбец нижнего правого угла
INT 10H
```

AH=2: Установка курсора в заданную позици

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

Вызов: AH = 08h, BH = номер страницы (0,...,7), обычно 0.

Возврат: AH = атрибут символа, AL = ASCII-код символа.

Вызов разрушает регистры BP, SI и DI.

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: AH =09h, AL = ASCII-код символа,

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

BH = номер страницы (0,1,...7), CX = коэффициент повторения.

Вызов разрушает регистры AX, BP, SI и DI.

Пример: Вывод символа с атрибутами на экран

Например:

```
MOV AH, 9 ;задаем функцию записи с атрибутами
MOV AL, CHAR ;символ в AL
MOV BL, 112 ;атрибуты в BL
MOV AH, 6 ;задаем процедуру скроллинга вверх
MOV BH, 0 ;активная страница
MOV CX, 1 ;вывести один раз
INT 10H
```

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: AH = 0Ah, AL = ASCII-код символа,

BH = номер страницы (0,1,...7), CX = коэффициент повторения.

Вызов разрушает регистры AX, BP, SI и DI.

Функция 0Eh. Запись символа с текущим атрибутом в режиме телетайпа.

Записывает символ ASCII в позицию курсора (предварительно установленную функцией 02h) на активной странице и смещает курсор к следующей позиции. Коды ASCII: 07h-звонок (BEL), 08h-шаг назад(BS), 0Dh-возврат каретки (CR), 0Ah-перевод строки (LF), рассматриваются как управляющие и выполняются соответствующие им действия. Остальные управляющие коды рассматриваются как символы и выводятся на экран. Действует автоматический перевод курсора на следующую строку после завершения предыдущей, а также прокрутка экрана вверх на 1 строку после заполнения самой нижней.

Вызов: AH = 0Eh, AL = ASCII-код символа,

BL = цвет символа (только для графического режима),

BH = номер страницы (0,1,...7), по умолчанию действует активная страница.

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

Вызов: AH = 0Fh.

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

BH =номер активной страницы.

Вызов разрушает регистры BP, SI и DI.

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 10h
;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: AX = 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: AH = 13h, AL = режим записи:

0-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);

1-отличается от режима 0 тем, что после записи курсор остаётся в конце строки;

2-строка содержит попеременно коды символов и атрибутов (т.е. каждый символ описывается 2 байтами-ASCII-кодом и атрибутом), после записи курсор принимает исходное положение;

3-отличается от режима 2 тем, что по окончании вывода курсор остаётся в конце строки.

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

CX = 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.

Отметим, что программы, выполняемые в операционной среде DOS, используют по умолчанию текстовый режим 3, страницу 0. Программы более широкого назначения должны запрашивать текущий видеорежим и страницу (функция 0Fh, int 10h) с последующим их применением в используемых функциях BIOS.

Пример:

```
Mov ah,0Fh                ;Запрос текущего режима
```

```
Int 10h
Mov v_mode, al          ;Сохранение режима
Mov current_page, bh    ;Сохранение строки.
```

Если программа выводит изображение на разные страницы, то последовательность действий с каждой страницей может быть следующей (предполагается режим по умолчанию с «0»-страницей):

– установка страницы функцией 05h;
– установка позиции курсора функцией 02h;
– построчное форматирование текста BIOS или DOS.

В дальнейшем может быть организован циклический просмотр содержания страниц путём их переключения функцией 05h, int 10h. При выходе из программы необходимо восстанавливать искомую «0» - страницу. Пример:

```
continue:
;Анализ буфера клавиатуры функцией DOS 06h int 21h с целью её завершения нажатием ;произвольной клавиши
mov ah,06h              ;Функция ввода без ожидания
mov dl,0FFh             ;Ввод
int 21h
jnz out_program          ;zf=0, есть символ, выход
jmp continue             ;zf=1, символа нет, продолжение работы
out_program:            ;Восстановление страницы функцией 05h, int 10h
exit:
mov ax,4C00h             ;Вызов функции завершения программы
int 21h
End start
```

Страницы видеобуфера могут быть последовательно отформатированы и способом непосредственного программирования памяти. Выбор страниц при этом осуществляется соответствующей инициализацией сегментного регистра ES/ Просмотр содержимого страниц также может быть выполнен путём их последовательного переключения с помощью функции 05h, int 10h.

3.3.5. Прерывание int 16h. Ввод с клавиатуры

Рассмотрим функции прерывания int 16h.

Функция 00h (10h). Чтение символа клавиатуры с ожиданием.

Читает из кольцевого буфера ввода символ и скан-код. После считывания они удаляются из буфера и возвращаются в регистре AX. Если

буфер пуст, ожидает ввода. Каждой клавише на клавиатуре соответствует так называемый скан-код, соответствующий только этой клавише. Этот код посылается клавиатурой при каждом нажатии и отпускании клавиши и обрабатывается в BIOS обработчиком прерывания Int 09h. Функция 00h даёт возможность получить код нажатия, не перехватывая этот обработчик. Если нажатой клавише соответствует ASCII-символ, то:

AL-ASCII-код символа, AH-скан-код клавиши.

Если нажатой клавише соответствует расширенный ASCII-код, то:

AL-00h, AH-расширенный ASCII-код.

Вызов: AH = 00h (83/84-key).

Возврат: AL = ASCII-код символа, изображённый на клавише/00h,

AH = скан-код/расширенный ASCII-код клавиши.

Функция 10h (AH = 10)-усовершенствованный вариант функции 00h для расширенной клавиатуры (101/102-key). Позволяет получить расширенные ASCII-коды для клавиш F11, F12, а также для ряда других комбинаций. В качества признака управляющих клавиш или их комбинаций, помимо значения 00h, используются 0Ah, 0Dh и E0h.

Функция 01h (11h). Поверка буфера клавиатуры на наличие в нём символа.

Определяет, имеются ли в кольцевом буфере ожидающие ввода символы; возвращает флаг ожидания и сам символ при его наличии. Однако символ и его скан-код не извлекаются из буфера и могут быть снова получены при повторном вызове функции 00h Int 16h. Данная функция относится к числу асинхронных: определив состояние буфера ввода, она возвращает управление программе.

Вызов: AH = 01h (83/84-key), 11h(101/102-key).

Возврат: ZF = 1, если буфер пуст и ZF = 0, если в буфере имеется ожидающий считывания символ. В этом случае:

AL = ASCII-код символа/00h, AH = скан-код клавиши/расширенный ASCII-код.

Функция 11h (AH = 11h)-усовершенствованный вариант функции 01h для расширенной клавиатуры (101/102-key). Позволяет получить расширенные ASCII-коды для клавиш F11, F12, а также для ряда других комбинаций. В качестве признака управляющих клавиш или их комбинаций, помимо значения 00h, используются 0Ah, 0Dh и E0h.

Функция 02h (12h). Получение флагов клавиатуры.

Возвращает байт флагов клавиатуры, описывающих состояние управляющих клавиш, записанное в байте (слове) области данных BIOS по адресу 0000h:0417h.

Вызов: AH = 02h

Возврат: A L=1-ый байт флагов клавиатуры.

Биты байта имеют следующие значения:

0: 1-правая Shift нажата
1: 1-левая Shift нажата
2: 1-Ctrl (любая) нажата
3: 1-Alt (любая) нажата
4: 1-режим Scroll Lock
5: 1-режим Num Lock
6: 1-режим Caps Lock
7: 1-режим Insert активен

Функция 12h (AH = 12h)-усовершенствованный вариант функции 02h для расширенной клавиатуры (101/102-key). Выводит такое же значение байта, как и функция 02h, по адресу 0000h:0417h, и, дополнительно, второй байт статуса клавиатуры (адрес 0000h:0418h) со следующими значениями:

0: 1-левая Ctrl нажата
1: 1-левая Alt нажата
2: 1-правая Ctrl нажата
3: 1-правая Alt нажата
4: 1-нажата Scroll Lock
4: 1-нажата Scroll Lock
6: 1-нажата Caps Lock
7: 1-нажата SysReg

3.3.6.Задержка программных операций

Программные задержки используются в тех случаях, когда в какой-либо точке программы надо приостановить её выполнение на некоторое время. По виду исполнения программные задержки делятся на два типа: задержки, реализуемые на основе выполнения программой «пустых» вложенных циклов, и задержки, реализуемые на основе системного таймера компьютера. Рассмотрим пример реализации задержки первого типа.

*Пример:*Программная задержка на основе выполнения вложенных циклов с командой Loop.

```
Proc        delay          ;Подпрограмма задержки
Mov cx, N       ;N-счётчик внешнего цикла
```

```
Outer:
push cx              ;Сохранение содержания счётчика внешнего цикла
Mov cx,0         ;Обеспечение максимального числа повторений (64К раз)
;внутреннего цикла
Inner:
loop Inner           ;Внутренний цикл
Pop cx               ;Восстановление содержания счётчика внешнего цикла
Loop Outer           ;Повторение внешнего цикла N раз
Endp
delay
```

В примере параметр N выполняет роль масштабного множителя времени задержки tзад = N*t исполнения внутреннего цикла.

При этом наименьшей единицей времени (т.е. «тиком») является время выполнения внутреннего цикла, состоящего, в свою очередь, из времени исполнения 65535 раз команды Loop. Параметр N подбирается экспериментально для получения tзад (в мсек или сек) с учётом быстродействия конкретного компьютера.

Из рассмотрения данного примера очевидны недостатки данного подхода, когда требуется обеспечить выполнение временной задержки в программе, независимо от типа используемого компьютера. Поэтому разумно определять время программной задержки непосредственно по таймеру. Выходные сигналы таймера с частотой 18,2 раза в секунду не зависят от производительности компьютера и играют роль счетчика суточного времени. Реализация данного способа использует функцию 00h прерывания BIOS Int 1Ah.

Int 1Ah, функция 00h. Чтение счетчика циклов таймера.

Обработчик прерывания BIOS от системного таймера (Int 8) подсчитывает количество прерываний (каждые 55 мсек или 18,2 раза в секунду) в двойном слове памяти с адресом 0040h:006Ch. Данная функция возвращает накопленное значение (двоичный код) и сбрасывает его в 0. В регистре AL возвращается 0, если содержимое счетчика не превысило значения, соответствующего 24 часам (при достижении этого значения счетчик сбрасывается), иначе возвращается AL = 1.

Вызов: AH = 00h.

Возврат: CX:DX-число тактов системного времени от полуночи,

AL-флаг перехода через сутки.

Примеры возвращаемых значений в CX:DX:

1 сек - 12h или 18,

1 минута - 04 44h или 1092,

1 час - 1 00 07h или 65543,

24 часа - 18 00 B0h или 1 573 040.

Для задержек меньших 14 секунд можно пользоваться только младшим байтом регистра DX

Пример: Задержка, реализуемая на основе системного таймера компьютера, задержка на 5 секунд, что соответствует 91 отсчету таймера

```
mov ah,0            ;Функция «чтения» циклов таймера
int 1Ah             ;Получение значения счетчика циклов в cx:dx
add dx,91           ;Добавление 5 сек. к младшему слову в dx
mov bx,dx           ;Запоминание требуемое значение в bx и выполнения
;постоянной проверки значений счетчика времени суток
repeat:     int 1Ah         ;Вновь получение значения счетчика
cmp dx,bx           ;Сравнение с искомым
jne repeat          ;Если не равно, то повторение,
;иначе задержка окончена
```

Если требуется введение задержки с высокой точностью, то необходимо использовать функцию 86h прерывания BIOS Int 15h. Она позволяет определить время задержки в микросекундах. Во время выполнения задержки разрешены прерывания. Управление программе возвращается после истечения заданного времени.

Int 15h, функция 86h

Вызов: AH = 86h, CX:DX = время задержки в мксек.

Возврат: CF = 0-нормальное исполнение, CF = 1-функция не поддерживается.

Пример: CX:DX = 0098h:9680h = 10 000 000 мксек = 10 сек.

Задание на выполнение

Варианты заданий

Разработать программу вывода текста на экран путём непосредственного программирования видеобуфера с использованием элементов форматирования (отступ от левой границы, перенос текста на следующую строку после пересечения словом правой границы).

Вход: DS:SI-адрес ASCI-строки, AH-атрибуты;

CX-число выводимых символов;

DH/DL-строка (row)/столбец (clm);

Indent_L, Indent_R-поля отступа (в столбцах) слева и справа.

Необходимо оптимизировать расчёт адреса видеобуфера ES:DI. Процедура должна возвращать исходное значение регистра ES.

Используя прямое программирование видеопамяти, заполнить несколько страниц видеобуфера с последующим просмотром их (выводом на экран) в циклическом режиме. При выходе из программы обеспечить восстановление текущей страницы.

Разработать программу рисования прямоугольника, используя графические символы в кодировке ASCII. Координаты верхнего левого угла (строка, столбец) и нижнего правого должны вводиться с клавиатуры после соответствующего приглашения.

В цикле ввести символ с клавиатуры и вывести его двоичное представление на экран. Если введен символ *, закончить работу программы. Для ввода данных на экран «Введите символ:», при выводе двоичного представления на экран выведите «Двоичное представление:»

В цикле ввести десятичное число с клавиатуры (Функция AH=2 INT 21H). Число десятичных разрядов от 1 до 5. Признак конца ввода - нажатие клавиши [Ввод] (код 13). Преобразовать число в двоичное и вывести его двоичное представление на экран. Для ввода данных на экран выведите строку с сообщением: «Введите цифру:»

В цикле ввести десятичное число с клавиатуры (Функция AH=0AH INT 21H). Число десятичных разрядов от 1 до 5. Признак конца ввода - нажатие клавиши [Ввод] (код 13). Преобразовать число в двоичное и вывести его двоичное представление на экран. Для ввода данных на экран выведите строку с сообщением: «Введите цифру:» (Функция AH=0AH INT 21H).

Вывести в текстовом режиме прямоугольную рамку на экран. Координаты левого верхнего и правого нижнего углов (0-24 и 0-79) ввести с клавиатуры. Символы для вывода рамки - коды ASCII. Для ввода данных на экран выведите сообщение: «Введите координаты левого верхнего угла:» и «Введите координаты правого нижнего угла:».

Введите координаты начала и конца отрезка прямой. В графическом режиме

выведите на экран прямую (горизонтальную, вертикальную, диагональную). Для ввода данных на экран выведите строку с сообщением: «Введите координаты начала отрезка:», «Введите координаты конца отрезка:.»

В графическом режиме выведите на экран квадрат. Координаты верхнего левого угла и длину стороны введите с клавиатуры. Для ввода данных на экран выведите строку с сообщением: «Введите координаты:», «Введите длину стороны:».

В графическом режиме выведите на экран прямоугольник. Координаты верхнего левого угла и длины сторон введите с клавиатуры. Для ввода данных на экран выведите строку с сообщением: «Введите координаты:», «Введите длину стороны:».

Выполнение лабораторной работы

1. Напишите программу на Ассемблере в соответствии с заданием.
2. Выполните ассемблирование.
3. Распечатайте листинг программы.
4. Распечатайте результаты работы программы.
5. Подготовьте отчет по лабораторной работе.

Контрольные вопросы

1. Что такое прерывание?
2. Графический режим работы процессора.
3. Переход в графический режим.
4. Особенности прерывания int 21.
5. Особенности прерывания int 10.
6. Особенности прерывания int 16.
7. Куда вводятся символы при нажатии клавиши?

Литература

1.В.И.Юров .- Ассемблер, 2 издание- СПб: Питер, 2003.
2.Пирогов В. Ю.- Ассемблер для Windows. Изд. 4-е перераб. и доп. - СПб.: БХВ- Петербург, 2015.
3. И.А. Калашников. Ассемблер-это просто. Программирование на Ассемблере. «Бином», 200х г.

Приложение 1

Для осуществления ввода с клавиатуры и вывода на экран символьной информации в реальном режиме используются функции DOS. Однако DOS не поддерживает ни позиционирование курсора, ни смену цвета выводимых символов. В текстовом режиме расширить возможности DOS можно с помощью драйвера ANSI.SYS. С графическими изображениями дело обстоит хуже, так как в DOS нет никаких графических функций. Нет их также и в драйвере ANSI.SYS, за исключением возможности перевода видеоадаптера в графический режим (с помощью Esc-последовательности Esc[=режимh). Для того, чтобы вывести на экран графическое изображение необходимо воспользоваться нижним уровнем операционной системы - базовой системы ввода-вывода (Basic In-Out System, BIOS). Программы BIOS находятся в постоянном запоминающем устройстве (ПЗУ) BIOS. В отличие от DOS, ко всем функциям которой можно обратиться с помощью прерывания 21h, в BIOS за каждым устройством компьютера закреплено свое прерывание. Так, программирование диска осуществляется с помощью прерывания int13h, клавиатуры - int16h, экрана – int10h. Прерывание int10h обеспечивает все функции видеоадаптера: смену видеорежима, вывод символьной и текстовой информации, смену шрифтов, настройку цветовой палитры, работу с графическим изображением и т.д. Воспользуемся прерыванием int10h для перехода в графический режим и вывода простейшего графического изображения.

Пример . Вывод на экран горизонтальной прямой.

```
        ;Установим графический режим EGA
mov AH,00h      ;(1)Функция задания режима
mov AL,10h      ;(2)Графический режим EGA
int 10h         ;(3)Вызов BIOS
        ;Нарисуем прямую линию в цикле по X
mov SI,150 ;(4)Начальная X-координата
mov CX,300      ;(5)Число точек по горизонтали
line:   push CX ;(6)Сохраним его в стеке
mov AX,0Ch      ;(7)Функция вывода пиксела
mov AL,4   ;(8)Цвет красный
mov BH,0   ;(9)Видеостраница
mov CX,SI  ;(10)X-координата (переменная)
mov DX,175      ;(11)Y-координата (константа)
```

```
int 10h             ;(12)Вызов BIOS
inc SI          ;(13)Инкремент X-координаты
pop CX              ;(14)Восстановим счетчик шагов
loop line       ;(15)Цикл из CX шагов
            ;Остановим программу для наблюдения результата ее работы
mov AH,08h          ;(16)Функция ввода с клавиатуры без эха
int 21h             ;(17)Вызов DOS
            ;Переключим видеоадаптер назад в текстовый режим
mov AH,00h          ;(18)Функция задания режима
mov AL,03h          ;(19)Текстовый режим
int 10h             ;(20)Вызов BIOS
```

В предложениях 1-3 с помощью функции 00h прерывания BIOS 10h осуществляется переключение видеоадаптера в графический режим. Поскольку номер режима заносится в байтовый регистр AL, всего может существовать 256 различных текстовых и графических режимов, из которых на сегодняшний день используются (аппаратурой различных фирм) около ста. Режим 10h обеспечивает вывод графического изображения 16 цветами с разрешением 640x350 точек и широко используется с видеоадаптерами EGA и VGA.

Изображение рисуется по точкам (в BIOS не предусмотрено программных средств вывода каких-либо геометрических фигур или хотя бы линий, как нет и средств закрашивания областей экрана). Для вывода на экран цветной точки (пиксела) используется функция 0Ch прерывания 10h. Эта функция требует занесения в регистр AL кода цвета, в BH - номера видеостраницы, в CX - X-координаты выводимой точки в диапазоне 0-349, а в DX - Y-координаты точки в диапазоне 0-639. Поскольку регистр CX используется, как счетчик шагов в цикле, для хранения X - координаты зарезервирован регистр SI.

Прямая горизонтальная линия в примере 3.1 рисуется путем вызова функции 0Ch в цикле, в каждом шаге которого значение Y-координаты остается неизменным (175 в примере), а значение X-координаты увеличивается на 1 (предложение 13). После завершения цикла формирования изображения в программе предусмотрена остановка (предложения 16-17) для того, чтобы пользователь мог, оставаясь в графическом режиме, проанализировать результаты работы программы. Для остановки программы используется функция DOS 08h ввода одного символа с клавиатуры, функция 08h, как уже отмечалось, не отображает введенный символ на экране и, тем самым, не искажает графическое изображение. Нажатие любой клавиши (кроме управляющих - Ctrl, Alt, Shift и др.) возобновляет выполнение программы.

В конце рассматриваемого фрагмента предусмотрено переключение видеоадаптера в стандартный текстовый режим с номером 03h (предложения

18...20). Если такое переключение не выполнить, видеоадаптер останется в графическом режиме, что может помешать правильному выполнению прикладных программ.

Рассмотрим кратко параметры вызова функции 0Ch прерывания 10h. В регистр BH заносится номер видеостраницы, на которую выводится данная точка. Графический адаптер EGA обеспечивает хранение и отображение двух графических страниц. По умолчанию видимой (активной) делается страница 0, однако рисовать изображение можно как на видимой, так и на невидимой странице. Для переключения страниц предусмотрена функция 05h прерывания 10h.

В регистр AL заносится код цвета точки. Адаптер поддерживает 64 цвета, хотя в каждый момент времени изображение на экране может содержать только 16 цветов. Этот набор из 16 цветов, выводимых на экран (цветовая палитра), задается программно и может легко изменяться. При загрузке машины устанавливается стандартная палитра, коды цветов которой приведены в табл.