Министерство общего и профессионального образования

Российской Федерации

САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО

Кафедра теоретических основ компьютерной безопасности

и криптографии

 \_\_\_\_РАБОТА С ДИСКАМИ

наименование курсовой работы ПРОПИСНЫМИ буквами.

КУРСОВАЯ РАБОТА

студента 1 курса факультета компьютерных наук

и информационных технологий

Томникова Вадима Анатольевича

фамилия, имя, отчество (полностью)

Научный руководитель

профессор В. В. Ридель

должность, уч.степень, уч.звание подпись, дата инициалы, фамилия

Зав. кафедрой

профессор В. В. Салий

должность, уч.степень, уч.звание подпись, дата инициалы, фамилия

Саратов 2003

**Оглавление.**

Введение.

1. Глава 1. Обзор дисковых устройств.
2. Глава 2. Физические аспекты работы с дисками.
3. Глава 3. Работа с дисками на физическом уровне.
	1. Определение типа компьютера.
	2. Прерывание INT 11h.
	3. Анализ содержимого CMOS-памяти.
	4. Таблицы параметров НМД и НГМД.
4. Глава 4. Программирование контроллера НГМД.
	1. Порты контроллера НГМД.
	2. Команды для контроллера НГМД.
	3. Использование команд контроллера НГМД.
	4. Немного о контроллере прямого доступа к памяти.
5. Глава 5. Функции BIOS для работы с дисками.
6. Глава 6. Работа с дисками по средствам MS-DOS.
	1. Таблица разделов и логические диски.
	2. Главная загрузочная запись.
	3. Таблица разделов диска.
	4. Загрузочная запись.
	5. Загрузка операционной системы.
	6. Поля элемента таблицы раздела диска.
	7. Первичный и расширенный раздел.
	8. Загрузочная запись.
	9. Формат загрузочной записи.
	10. Расширенный блок параметров BIOS.
	11. Логический номер сектора.
	12. Прерывания INT 25h и INT 26h.
	13. Таблица размещения файлов.
	14. Последовательный и прямой доступ.
	15. Кластеры.
	16. Содержимое таблицы FAT.
	17. Два формата таблицы FAT.
	18. Определение формата таблицы FAT.
	19. Идентификация кластеров.
	20. Чтение таблицы FAT.

**Введение.**

В распоряжении программиста имеются средства разного уровня, предназначенные для работы с дисковой системой в среде MS-DOS.

Во-первых, любая программа может обращаться непосредственно к аппаратуре контроллера жесткого диска или контроллера накопителя на гибких магнитных дисках. Это самый низкий уровень, применение которого оправдано далеко не всегда. Более того, во многих источниках настоятельно рекомендуется не работать с контроллером на уровне портов, если точно не известно, зачем это нужно. Большинство задач решается с применением функций MS-DOS или BIOS (даже такие нетрадиционные задачи, как защита от несанкционированного копирования). Однако иногда программист бывает вынужден использовать самый низкий уровень, рискуя потерять совместимость с многочисленными типами дисковых контроллеров.

Во-вторых, программа MS-DOS может работать с дисками при помощи прерывания BIOS. В работе будут рассмотрены различные функции прерывания INT 13h, предназначенного для выполнения операций чтения, записи и форматирования. Использование функций прерывания INT 13h намного предпочтительнее, чем непосредственное программирование контроллера, так как BIOS скрывает особенности аппаратной реализации контроллера.

В-третьих, программа может обращаться к прерываниям MS-DOS, специально предназначенным для доступа к дисковой системе. Сервис, предоставляемый этими прерываниями также можно разделить на низко- и высокоуровневый, поэтому у программиста всегда есть выбор.

Выбирая средства обращения к дисковой системе, программист должен вначале рассмотреть возможность использования наиболее высокоуровневых документированных средств, таких как прерывания MS-DOS. И лишь затем имеет смысл обратиться к функциям BIOS или к программированию портов контроллера.

Несмотря на все сказанное выше, повествование о средствах работы с дисковой системой будет вестись в обратном порядке. Вначале будет рассказано об использовании наиболее низкоуровневых средств, и лишь затем будет рассказано о прерываниях BIOS и MS-DOS. Это позволит подойти к изучению высокоуровневых средств, имея полное представление о том, как устроена дисковая система.

**Глава 1. Обзор дисковых устройств.**

Первые персональные компьютеры типа IBM PC не имели жесткого диска ("винчестера", или, по отечественной терминологии, накопителя на жестком магнитном диске - НМД). Они были оборудованы двумя флоппи-дисками (накопителями на гибком магнитном диске - НГМД), которые и представляли собой дисковую систему. В таких компьютерах установлены, как правило, два накопителя для флоппи-дисков (дискет).

Эти накопители подключены к контроллеру - специальному устройству, выполняющему функции управления. Контроллер обычно выполнен в виде платы и вставлен в разъем системной шины, который находится на основной плате компьютера.

В компьютере IBM PC использовали дискеты диаметром 5,25". Сейчас пользователи практически отказались от громоздких дискет диаметром 5,25", отдавая предпочтение более компактным дискетам диаметром 3,5". Компьютер IBM XT имел один или два НГМД для дискет диаметром 5,25" и, как правило, один НМД емкостью 20 Мбайт. Все дисководы подключались к одному общему контроллеру. Модель IBM AT и компьютеры более высокого класса могут иметь несколько дисковых контроллеров, два НГМД с различным диаметром (3,5" и 5,25") и несколько НМД.

Пожалуй, самая интересная разновидность современных дисковых накопителей - оптические, или лазерные. В настоящее время существует несколько типов оптических дисковых накопителей. Это устройство чтения компакт-дисков CD-ROM , устройство записи CD Recordable , накопители WORM и магнитооптические накопители.

Диски CD-ROM (Compact-Disk, Read-Only Memory) - это диски, которые по своему формату и технологии записи информации напоминают звуковые компакт-диски. Они имеют диаметр 120 мм и могут содержать порядка 800 Мбайт информации. Эта информация записывается один раз и впоследствии может только читаться, как из постоянного запоминающего устройства.

Диски CD Recordable внешне очень похожи на диски CD-ROM , однако с помощью специального устройства пользователь может записать на него свою информацию. При необходимости можно на один и тот же диск дозаписывать новые данные. Диск CD Recordable можно прочитать с помощью обычного устройства чтения компакт-дисков CD-ROM.

WORM -диски (Write Once, Read Many) предназначены для однократной записи и многократного считывания данных. Эти диски, как и CD Recordable , наилучшим образом подходят для архивного хранения информации, например, содержимого обширных баз данных.

Магнитооптические накопители могут многократно записывать информацию на один и тот же носитель (как обычные магнитные диски). Это одни из самых дорогостоящих дисковых накопителей. Их производительность сравнима с производительностью обычных жестких дисков.

Основной недостаток лазерных накопителей (кроме магнитооптических) - относительно невысокое быстродействие по сравнению с традиционными накопителями на жестких дисках. Однако этот недостаток постепенно преодолевается.

**Глава 2. Физические аспекты работы с дисками.**

Что же, собственно, представляет из себя диск?

Дискета (флоппи-диск) - это круглая пластинка в квадратном конверте, покрытая с двух сторон магнитным материалом. Этот материал похож на тот, что используется в магнитных лентах обычных бытовых магнитофонов, но отличается по некоторым характеристикам (например, по форме и ширине петли гистерезиза). Ближе к центру в диске находится маленькое отверстие, предназначенное для синхронизации. Когда дискета вставляется в дисковод, с обеих сторон (сверху и снизу) к ней прижимаются магнитные головки. При этом нет никакого зазора между головками и поверхностью дискеты. С помощью специального шагового двигателя головки могут перемещаться скачкообразно вдоль радиуса диска, как бы прочерчивая при вращении диска концентрические окружности. Эти окружности называются дорожками, треками или цилиндрами - в литературе можно встретить различные названия.

Жесткий диск состоит из нескольких жестких круглых пластинок, покрытых магнитным материалом. Пластинки вращаются с огромной скоростью (порядка 7200 - 14400 оборотов в минуту) в герметичном корпусе. Около каждой стороны пластинки располагается по одной магнитной головке, но эти головки не соприкасаются с диском, а плавают на воздушной подушке в непосредственной близости от его поверхности. Подавая команды дисковому контроллеру, программа может перемещать блок головок вдоль радиуса диска, переходя таким образом от одного цилиндра к другому. Перемещаясь вдоль окружности дорожки, магнитная головка может записывать или считывать информацию примерно так, как это происходит в бытовом магнитофоне. Запись выполняется по битам, при этом добавляется различная служебная информация и информация для контроля данных.

Данные записываются не сплошным потоком, а блоками определенного размера. Эти блоки называются секторами. Сектор представляет собой наименьший объем данных, который записывается или прочитывается контроллером. Для каждого сектора выполняется контроль записи или чтения. При записи сектора вычисляется контрольная сумма всех байтов, находящихся в секторе, и эта контрольная сумма записывается на диск в служебную область, расположенную после сектора. При чтении эта контрольная сумма вычисляется заново и сравнивается с контрольной суммой, считанной из служебной области. При несовпадении контроллер сообщает программе об ошибке. Дорожки нумеруются начиная от нулевой, головки тоже начиная от нулевой, а вот секторы - начиная с первого. Почему так было сделано - сказать трудно, но именно такая нумерация используется при работе с контроллером диска и функциями прерывания BIOS, обслуживающими дисковую подсистему. В операциях чтения или записи на физическом уровне необходимо указывать номер дорожки (0, 1, ...), головки (0, 1, ...), номер сектора (1, 2, ...).

Для правильной работы с дисками на физическом уровне программа должна располагать существенно большей информацией о дисках, чем просто номер нужной дорожки или головки. Например, она должна знать, сколько головок и сколько дорожек имеет то или иное дисковое устройство, сколько байт содержится в одном секторе и многое другое.

**Глава 3. Работа с дисками на физическом уровне.**

Прежде чем начать работу с дисками на физическом уровне, необходимо выяснить конфигурацию дисковой системы - сколько дисководов и какого типа подключено к компьютеру, сколько дорожек и головок имеется на каждом из дисководов и т. д. Способ, которым определяется конфигурация дисковой системы, зависит от модели компьютера (PC, XT, AT), поэтому вначале займемся определением типа персонального компьютера.

**3.1 Определение типа компьютера.**

ПЗУ базовой системы ввода/вывода BIOS содержит по адресу FFFFh:FFFEh байт конфигурации, значение которого можно использовать для идентификации типа компьютера:

|  |  |
| --- | --- |
| Значение | Тип компьютера |
| FFh | Оригинальный IBM PC |
| FEh | IBM PC/XT, Portable PC  |
| FDh | PCjr |
| FCh | IBM PC/AT |
| FBh | IBM PC/XT с памятью 640 Кбайт на материнской плате  |
| F9h | Convertible PC |

Для компьютеров IBM PC и IBM PC/XT конфигурация дисковой системы определяется установкой переключателей на основной плате, в частности, переключателями устанавливается количество подключенных к системе НГМД.

Компьютеры IBM PC/AT (и более высокого класса) имеют на основной плате CMOS-память с малым энергопотреблением, которая питается от аккумулятора. В CMOS-памяти хранится информация о конфигурации дисковой системы. В процессе инициализации BIOS считывает эту информацию и записывает ее в свою внутреннюю область данных.

Проанализировав значение байта конфигурации, можно сделать предварительное заключение о составе дисковой системы компьютера. Если оно равно FFh, FDh, F9h, то наш компьютер не имеет НМД - это одна из разновидностей IBM PC. Значения FEh, FBh могут соответствовать IBM PC/XT и совместимым с ним компьютерам. Такие компьютеры могут быть оборудованы НМД. И, наконец, значение FCh соответствует IBM PC/AT. Для этого компьютера конфигурация дисковой системы должна определяться исходя из содержимого CMOS-памяти.

**3.2 Прерывание INT 11h.**

Прерывание базовой системы ввода/вывода INT 11h возвращает в регистре AX байт конфигурации системы, который можно использовать для определения количества НГМД и наличия НМД. Самый младший бит байта конфигурации (бит 0) - признак наличия в системе НМД. Если этот бит установлен в 1, то компьютер оборудован НМД, иначе дисковая система состоит только из накопителей на гибких магнитных дисках.

Биты 7 и 6 содержат информацию о количестве НГМД:

|  |  |
| --- | --- |
| Содержимое бит 7 и 6 | Количество установленных НГМД  |
| 00 | 1 |
| 01 | 2 |
| 10 | 3 |
| 11 | 4 |

Это прерывание лучше всего использовать для IBM PC/XT и IBM PC. Для IBM PC/AT необходимо исследовать содержимое CMOS-памяти. Займемся этим.

**3.3 Анализ содержимого CMOS-памяти.**

Программа не может непосредственно адресовать CMOS-память, как обычную оперативную память. Для работы с CMOS-памятью необходимо использовать порты ввода/вывода с адресами 70h и 71h, причем процедура записи или чтения состоит из двух шагов.

На первом шаге операции чтения или записи программа должна записать в порт 70h номер нужной ячейки CMOS-памяти (0...3Fh). На втором шаге программа должна обратиться к порту 71h для выполнения записи в указанную ячейку памяти или чтения из нее.

Приведем фрагмент программы, составленной на языке ассемблера, который считывает байт из CMOS-памяти с адресом 12h:

*mov al,12h*

*out 70h,al ; задаем адрес в CMOS-памяти*

*jmp $+2 ; небольшая задержка*

*in al,71h ; записываем в AL считанное значение*

Запись в CMOS-память выполняется аналогично.

При анализе конфигурации дисковой системы для нас представляют наибольший интерес ячейки CMOS-памяти со следующими адресами:

• 14h - байт конфигурации

Биты 7, 6 этого байта имеют такое же значение, что и в младшем байте слова конфигурации, возвращаемого прерыванием INT 11h - они содержат информацию о количестве установленных в компьютере НГМД.

Значение бита 0, равное нулю, говорит о том, что в системе нет ни одного НГМД.

• 10h - тип НГМД

Младшая и старшая тетрады этого байта описывают, соответственно, второй и первый НГМД:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Значение | Емкость, Кбайт | Диаметр | Количество секторов на одну дорожку | Количество дорожек |
| 0000 | НГМД не установлен | - | - | - |
| 0001 | 360 | 5,25" | 9 | 40 |
| 0010 | 1200 | 5,25" | 15 | 80 |
| 0011 | 720 | 3,5" | 9 | 40 |
| 0100 | 1440 | 3,5" | 18 | 80 |

• 12h - тип НМД C: и D:

Этот байт разделен на две тетрады аналогично байту, который описывает НГМД. Однако в тетраде можно закодировать только 16 значений, а различных типов НМД значительно больше. Поэтому тип 15 используется специальным образом - если тип НМД в младшей тетраде (диск C:) равен 15, то правильное значение типа находится в CMOS-памяти по адресу 19h. Аналогично для диска D: этот тип можно взять из байта по адресу 1Ah (если содержимое старшей тетрады байта с адресом 12h равно 15).

Если в вашем компьютере установлен НМД с интерфейсом ESDI , SCSI или другим специализированным интерфейсом, то, как правило, для работы с ними используется специальная "дисковая" базовая система ввода/вывода. Соответствующая микросхема ПЗУ может быть расположена непосредственно в контроллере. При этом в CMOS-памяти в ячейке 12h для типа диска может быть указано нулевое значение, несмотря на то, что диск установлен. Прерывание INT 11h , тем не менее, скажет вам, что в системе имеется НМД.

Если используется "дисковая" базовая система ввода/вывода, то она сама инициализирует таблицу параметров диска (будет описана позже) и выполняет обработку прерывания INT 13h . Так как MS-DOS при обращении к дискам использует именно это прерывание, то не возникает никаких проблем, связанных с отсутствием типа диска в CMOS-памяти. Другие операционные системы, такие как Windows NT и OS/2 , используют для работы с дисками специальные драйверы.

**3.4 Таблицы параметров НМД и НГМД.**

Для работы с диском на физическом уровне необходимо знать такие его характеристики, как количество головок, секторов и др. Эти характеристики можно определить из таблиц параметров НГМД и НМД, заполняемых BIOS в процессе инициализации системы.

Анализируя содержимое CMOS-памяти в компьютерах IBM PC/AT или установку переключателей конфигурации на основной плате в компьютерах IBM PC и IBM PC/XT, BIOS в процессе инициализации создает таблицу параметров дискеты DPT (Diskette Parameter Table), а также одну или две таблицы параметров жесткого диска HDPT (Hard Disk Parameter Table). Если имеется специальная "дисковая" система ввода/вывода, то она сама создает таблицы HDPT.

Таблица параметров дискеты DPT имеет длину 10 байт, ее адрес располагается в области данных BIOS по адресу 0000h:0078h, что соответствует вектору прерывания INT 1Eh . Таблица содержит следующие параметры:

|  |  |  |  |
| --- | --- | --- | --- |
| Смещение, байт | Размер, байт  | Имя поля | Описание |
| 0 | 1 | srt\_hut  | Биты 0...3:SRT (Step Rate Time) - задержка для переключения головок, лежит в пределах 1 - 16 мс и задается с интервалом 1 мс (0Fh - 1 мс, 0Eh - 2 мс, 0Dh - 3 мс, ...).Биты 4...7:Задержка разгрузки головки, лежит в пределах 16 - 240 мс и задается с интервалом 16 мс (1 - 16 мс, 2 - 32 мс, ..., 0Fh - 240 mc)  |
| 1 | 1 | dma\_hlt  | Бит 0:Значение этого бита, равное 1, говорит о том, что используется прямой доступ к памяти; Биты 2...7:Время загрузки головок HLT - интервал между сигналом загрузки головок и началом операции чтения или записи, лежит в пределах 2 - 254 мс и задается с интервалом 2 мс (1 - 2 мс, 2 - 4 мс, ..., 0FFh - 254 мс)  |
| 2 | 1 | motor\_w  | Задержка перед выключением двигателя |
| 3 | 1 | sec\_size  | Код размера сектора в байтах:0 - 128;1 - 256;2 - 512;3 - 1024  |
| 4 | 1 | eot | Номер последнего сектора на дорожке |
| 5 | 1 | gap\_rw  | Длина межсекторного промежутка для чтения или записи  |
| 6 | 1 | dtl | Максимальная длина передаваемых данных. Используется, когда не задана длина сектора  |
| 7 | 1 | gap\_f | Длина межсекторного промежутка для операции форматирования  |
| 8 | 1 | fill\_char  | Байт-заполнитель для форматирования, обычно используется F6h  |
| 9 | 1 | hst | Время установки головки в мс |
| 10 | 1 | mot\_start  | Время запуска двигателя в 1/8 долях секунды  |

Все времена зависят от частоты тактового генератора контроллера НГМД, приведенные значения соответствуют частоте 8 МГц.

Адреса таблиц параметров жестких дисков HDPT расположены по адресам, соответствующим векторам прерываний INT 41h (для первого физического диска) и INT 46h (для второго физического диска). Эти таблицы имеют следующий формат:

|  |  |  |  |
| --- | --- | --- | --- |
| Смещение, байт | Размер, байт  | Имя поля | Описание |
| 0 | 2 | max\_cyl  | Максимальное количество дорожек на диске |
| 2 | 1 | max\_head  | Максимальное количество магнитных головок |
| 3 | 2 | srwcc | Начальная дорожка для предварительной записи (Starting reduced-write current cylinder)  |
| 5 | 2 | swpc | Начальная дорожка для предварительной компенсации при записи (Starting write precompensation cylinder)  |
| 7 | 1 | max\_ecc  | Максимальная длина блока коррекции ошибок ECC (Maximum ECC data burst length)  |
| 8 | 1 | dstopt  | Параметры устройства:бит 7 - запрет восстановления; бит 6 - запрет восстановления по блоку коррекции ошибок ECC (Error Correction Code);биты 2-0 - дополнительные параметры устройства  |
| 9 | 1 | st\_del  | Стандартная величина задержки |
| 10 | 1 | fm\_del  | Величина задержки для форматирования диска  |
| 11 | 1 | chk\_del  | Величина задержки для проверки диска |
| 12 | 4 | reserve  | Зарезервировано |

Таблица параметров жесткого диска содержит такие важнейшие значения, как максимальное количество дорожек и максимальное количество головок. Если вам не удалось определить тип диска, то таблица HDPT - единственное надежное место, откуда можно получить информацию о количестве дорожек и головок.

**Глава 4. Программирование контроллера НГМД.**

Большинство дисковых операций можно выполнить на уровне функций BIOS. Это самый простой и надежный способ работы с диском на физическом уровне. Однако в отдельных случаях вам может потребоваться непосредственный доступ к контроллеру НГМД - например, если вы разрабатываете систему защиты данных от несанкционированного копирования.

Информация, приведенная в этой главе, ориентирована прежде всего не на выполнение операций чтения или записи (которые лучше выполнять с помощью функций BIOS), а на управление контроллером и получение состояния контроллера. Именно эти операции требуются для организации защиты данных от несанкционированного копирования.

**4.1 Порты контроллера НГМД.**

Программа обращается к контроллеру для выполнения различных операций с помощью команд ввода/вывода. Для IBM PC и IBM PC/XT используются три порта с адресами 3F2h, 3F4h и 3F5h. В компьютерах IBM PC/AT дополнительно используются два порта с адресами 3F6h и 3F7h.

Порт 3F2h работает только на запись, это порт вывода. С его помощью можно выбирать для работы один из НГМД (одновременно можно работать только с одним НГМД), сбрасывать контроллер в исходное состояние, разрешать или запрещать прерывания от контроллера и работу схем прямого доступа к памяти, включать или выключать двигатели НГМД.

Приведем назначение отдельных бит этого порта:

|  |  |
| --- | --- |
| Биты | Назначение |
| 0-1 | Выбор НГМД. Компьютеры IBM PC/AT не используют бит 1, так как в них установлены только два НГМД  |
| 2 | 0 - сброс контроллера; 1 - разрешение работы контроллера  |
| 3 | 1 - разрешение прерываний и прямого доступа к памяти  |
| 4-7 | Значение 1 в каждом разряде вызывает включение соответствующего двигателя НГМД. Для компьютеров IBM PC/AT биты 6-7 не используются  |

Порт 3F4h предназначен только для чтения. С его помощью можно получить байт основного состояния контроллера. Назначение отдельных бит приведено ниже:

|  |  |
| --- | --- |
| Биты | Назначение |
| 0-3 | Значение 1 говорит о том, что соответствующий НГМД занят, он выполняет операцию поиска. Для IBM PC/AT биты 2-3 не используются  |
| 4 | Контроллер занят выполнением команды чтения или записи  |
| 5 | 0 - используется режим прямого доступа к памяти; 1 - режим прямого доступа к памяти не используется  |
| 6 | Направление передачи данных:0 - от процессора к контроллеру; 1 - от контроллера к процессору  |
| 7 | Запрос на передачу данных - контроллер готов к записи или чтению данных  |

Порт 3F5h предназначен для записи или чтения данных. Он используется для всех операций.

Выполнение любой операции начинается с того, что программа посылает в этот порт байт кода операции, за которым следует один или несколько байт параметров. Количество байт параметров и их назначение зависит от кода операции (т. е. от первого байта). После выполнения операции программа считывает несколько байт результата для анализа результата выполнения операции.

Порт 3F7h работает на запись и чтение, он используется только в IBM PC/AT.

При записи в этот порт биты 0-1 определяют скорость передачи данных:

|  |  |
| --- | --- |
| Биты | Скорость передачи данных, Кбайт/с  |
| 00 | 500 (высокая плотность HD)  |
| 01 | 300 (двойная плотность DD)  |
| 10 | 250 (одинарная плотность SD)  |
| 11 | Зарезервировано |

Приведем назначение отдельных бит порта 3F7h при чтении из него:

|  |  |
| --- | --- |
| Биты | Назначение |
| 0 | 1 - выбран НГМД 0 |
| 1 | 1 - выбран НГМД 1 |
| 2-5 | Выбраны головки : бит 2 соответствует головке 0, бит 3 - головке 1 и т. д.  |
| 6 | Переключатель записи |
| 7 | 1 - признак замены дискеты  |

**4.2 Команды для контроллера НГМД.**

Контроллер НГМД может выполнять 15 операций, или команд. Команда разделяется на три фазы - командная фаза, фаза выполнения, фаза результата. В командной фазе программа должна передать контроллеру всю информацию, необходимую для команды. В фазе выполнения команда выполняется, и в фазе результата программа получает от контроллера информацию о состоянии контроллера.

Информация, необходимая для команды, передается контроллеру через порт данных 3F5h. В соответствии с форматом команды программа должна последовательно вывести в этот порт код команды и все параметры.

Прежде чем программа начнет командную фазу, она должна убедиться в том, что контроллер завершил выполнение предыдущей операции и готов к приему команды. Для этого программа должна прочитать байт основного состояния контроллера из порта с адресом 3F4h и проверить биты 6 и 7. Бит 6 должен быть установлен в 0. Это означает, что данные будут передаваться от процессора к контроллеру. Бит 7 должен быть установлен в 1 - это готовность контроллера к приему команды.

Фаза выполнения начинается после установки битов 6 и 7 байта основного состояния в 1. После завершения команды контроллер формирует сигнал запроса прерывания.

В фазе результата процессор считывает состояние контроллера. Это состояние хранится в нескольких внутренних регистрах контроллера:

• RS - регистр основного состояния;

• ST0, ST1, ST2, ST3 - регистры дополнительного состояния.

Регистр основного состояния доступен через порт 3F4h, содержимое остальных регистров процессор считывает после выполнения контроллером команды через порт данных 3F5h.

В форматах команд и таблицах, приведенных ниже, используются следующие обозначения:

|  |  |
| --- | --- |
| Обозначение | Описание |
| MT | Двухсторонняя операция |
| MFM | Двойная/одинарная плотность записи  |
| SK | Пропуск удаленных данных  |
| HDS | Номер головки для двухстороннего НГМД  |
| DS1, DS0 | Номер выбираемого НГМД  |
| C | Номер цилиндра |
| H | Номер головки для двухстороннего НГМД  |
| R | Номер сектора |
| N | Число байт в секторе |
| EOT | Номер последнего сектора на дорожке  |
| GPL | Размер промежутка |
| DTL | Количество считываемых или записываемых байт  |
| SC | Количество секторов в цилиндре  |
| D | Данные |
| PCN | Номер цилиндра после выполнения команды чтения состояния прерывания  |
| SRT | Время шага, мс |
| HUT | Время разгрузки головки  |
| HLT | Время загрузки головки  |
| ND | Режим прерывания |
| NCN | Номер цилиндра после поиска  |

Приведем форматы всех команд контроллера НГМД.

* Чтение данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 0  | 0 | 1  | 1 | 0  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Чтение удаленных данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 0  | 1 | 1  | 0 | 0  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Запись данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | 0 | 0  | 0 | 1  | 0 | 0  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Запись удаленных данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | 0 | 0  | 1 | 0  | 0 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Чтение данных с дорожки

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 0  | 0 | 0  | 1 | 0  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Сканирование до "равно"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 1  | 0 | 0  | 0 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Сканирование до "меньше" или "равно"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 1  | 1 | 0  | 0 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Сканирование до "больше" или "равно"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MT | MFM  | SK | 1  | 1 | 1  | 0 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Форматирование дорожки

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | MFM  | 0 | 0  | 1 | 1  | 0 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Считывание индексных данных

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | MFM  | 0 | 0  | 1 | 0  | 1 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Инициализация

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0  | 0 | 0  | 0 | 1  | 1 | 1  |
| 0 | 0  | 0 | 0  | 0 | 0  | 1 | 0  |

* Чтение состояния прерывания

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0  | 0 | 0  | 1 | 0  | 0 | 0  |

* Определение параметров

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0  | 0 | 0  | 0 | 0  | 1 | 1  |

* Чтение состояния накопителя

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0  | 0 | 0  | 0 | 1  | 0 | 0  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

* Поиск

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0  | 0 | 0  | 1 | 1  | 1 | 1  |
| 0 | 0  | 0 | 0  | 0 | HDS  | DS1 | DS0  |

Первые несколько команд имеют одинаковый формат параметров и одинаковые байты результата.

Приведем байты параметров, которые должны следовать за командами и байты результата, которые процессор должен считать после выполнения команды.

|  |  |  |
| --- | --- | --- |
| Команда | Байты параметров | Байты результата |
| Чтение данныхЧтение удаленных данныхЗапись данныхЗапись удаленных данныхЧтение данных с дорожкиСканирование до "равно"Сканирование до "меньше" или "равно"Сканирование до "больше" или "равно"  | C, H, R, N, EOT,EOT, GPL, DTL | ST0, ST1, ST2,C, H, R, N  |
| Форматирование дорожки | N, SC, GPL, D  | ST0, ST1, ST2,C, H, R, N |
| Чтение индексных данных | Отсутствуют  | ST0, ST1, ST2,C, H, R, N |
| Инициализация | Отсутствуют  | Отсутствуют |
| Чтение состояния прерывания | Отсутствуют  | ST0, PCN |
| Определить параметры | 1 байт:мл. тетрада - HUTст. тетрада - SRT 2 байт:бит 0 - NDбиты 1-7 - HLT  | Отсутствуют |
| Чтение состояния накопителя | Отсутствуют  | ST3 |
| Поиск | C | Отсутствуют  |

После выполнения команды центральный процессор должен получить от контроллера байты результата. Среди них - содержимое внутренних регистров состояния контроллера ST0, ST1, ST2, ST3. Опишем назначение отдельных бит этих регистров.

Формат регистра ST0:

|  |  |  |
| --- | --- | --- |
| Биты | Название | Назначение  |
| 1, 0 | US1, US2 | Код НГМД  |
| 2 | HD | Номер головки  |
| 3 | NC | Устанавливается, если НГМД не готов выполнить команду чтения или записи  |
| 4 | EC | Сбой оборудования  |
| 5 | SE | Завершена команда "Поиск"  |
| 7, 6 | I, C | Код прерывания:00 - нормальное завершение;01 - аварийное завершение;10 - неправильная команда;11 - нет готовности НГМД  |

Формат регистра ST1:

|  |  |  |
| --- | --- | --- |
| Биты | Название | Назначение  |
| 0 | MA | Пропуск адресной метки. Этот бит устанавливается в 1, если контроллер не может найти адресную метку  |
| 1 | NN | Защита записи. Устанавливается, если при выполнении операции контроллер получает от НГМД сигнал защиты записи  |
| 2 | ND | Сектор не найден  |
| 3 | - | Зарезервировано  |
| 4 | OR | Переполнение. Процессор не успевает выполнять обмен данными с контроллером  |
| 5 | DE | Ошибка в данных при проверке контрольной суммы  |
| 6 | - | Зарезервировано  |
| 7 | EN | Несуществующий сектор, устанавливается, когда контроллер пытается прочесть сектор со слишком большим адресом  |

Формат регистра ST2:

|  |  |  |
| --- | --- | --- |
| Биты | Название | Назначение  |
| 0 | MD | Пропущен адресный маркер в поле данных  |
| 1 | BC | Дорожка не читается  |
| 2 | SN | Ошибка сканирования. Устанавливается, если при выполнении команды сканирования контроллер не может найти требуемую дорожку  |
| 3 | SH | Сканирование выполнено, дорожка найдена  |
| 4 | WC | Ошибка адреса дорожки  |
| 5 | DD | Ошибка в поле данных  |
| 6 | CM | Во время операции чтения или сканирования не обнаружен сектор с маркером удаленных данных  |
| 7 | - | Зарезервировано  |

Формат регистра ST3:

|  |  |  |
| --- | --- | --- |
| Биты | Название | Назначение  |
| 1, 0 | US1, US2 | Код выбранного НГМД  |
| 2 | HD | Номер выбранной головки  |
| 3 | TS | Используется режим двухсторонней записи  |
| 4 | T0 | Головка установлена на дорожку 0  |
| 5 | RDY | НГМД готов к работе  |
| 6 | WP | Защита записи на диске  |
| 7 | FT | Неисправность НГМД  |

Дополнительно перед выполнением операции и после ее завершения надо проанализировать содержимое описанного выше регистра основного состояния контроллера RS.

Команда "Определить параметры" задает времена задержки для трех внутренних таймеров контроллера. Первый байт параметров состоит из двух полей - SRT и HUT. Поле SRT задает временной интервал между шаговыми импульсами двигателя перемещения головки. Это поле имеет ширину 4 бита. Поле HUT определяет время разгрузки головки и тоже имеет ширину 4 бита.

Второй байт параметров состоит из полей HLT и ND. Поле HLT имеет ширину 7 бит и определяет время загрузки головки. Бит ND предназначен для использования канала прямого доступа ПДП - если этот бит установлен в 0, то ПДП используется, иначе обмен данными идет через центральный процессор.

Параметры для команды "Определить параметры" лучше всего взять из таблицы параметров дискеты, которая заполняется базовой системой ввода/вывода BIOS во время инициализации системы. Конечно, если вам нужны нестандартные параметры, вы можете использовать свои, ориентируясь на оригинальные значения из таблицы параметров дискеты.

Команда "Инициализация" может выполняться одновременно для всех накопителей. По этой команде головки перемещаются на нулевую дорожку.

Команда "Поиск" используется для установки головки на нужную дорожку. Поиск может выполняться одновременно для нескольких накопителей.

Команда "Чтение состояния прерывания" может вырабатываться после завершения других команд для выяснения состояния контроллера после прерывания. Эту команду удобно использовать после команд "Поиск" или "Инициализация".

После поступления команды "Чтение данных" загружается головка, контроллер считывает метки адреса идентификатора ID и поля ID. Контроллер последовательно считывает номера секторов, и как только считанный номер совпадет с запрошенным, контроллер байт за байтом считывает данные, расположенные в секторе, и передает их либо центральному процессору, либо каналу прямого доступа к памяти. При передаче данных контроллер должен обслуживаться каждые 27 мкс в режиме одинарной плотности и 13 мкс в режиме двойной плотности, иначе в регистре состояния ST3 устанавливается флаг переполнения OR.

Если контроллер не может найти нужный сектор, то в регистре ST1 устанавливается флаг отсутствия данных ND. При ошибке чтения данных, обнаруженной схемами избыточного циклического контроля CRC , устанавливается флаг ошибки данных DE.

При считывании адресной метки удаленных данных в регистре ST2 и сброшенном в 0 бите SK команды флаг CM устанавливается в 1, читаются все данные из этого сектора, затем выполнение команды прекращается.

Поле команды MT позволяет задать выполнение многодорожечной операции, при которой контроллер считывает данные с обеих сторон дискеты. Поле MFM определяет плотность обрабатываемой информации: значение 0 соответствует одинарной плотности, 1 - двойной.

Если поле команды N содержит 0, то поле DTL определяет объем передаваемых данных. Если поле N содержит отличное от нуля значение, поле DTL игнорируется и должно содержать значение 0FFh.

Выполнение команды "Запись" аналогично. В режиме записи обмен данными процессора с контроллером должен происходить каждые 31 мкс в режиме одинарной плотности и каждые 15 мкс в режиме двойной плотности.

По команде "Запись удаленных данных" в начале поля данных записывается адресная метка удаленных данных вместо обычной адресной метки данных.

По команде "Чтение данных дорожки" считываются все поля данных с каждого сектора дорожки как непрерывные блоки данных. С помощью этой команды можно выполнять многодорожечные операции и пропуски.

Команда "Чтение индексных данных" позволяет определить положение головки.

Команда "Форматирование дорожки" форматирует всю дорожку - на нее записываются интервалы, адресные метки, поля индексных данных и поля данных. Вам не обязательно располагать секторы в порядке увеличения номеров, так как при форматировании контроллер запрашивает параметры C, H, R и N.

Группа команд "Сканирование" позволяет сравнивать данные, поступающие от контроллера и от центрального процессора. Контроллер выполняет побайтное сравнение и ищет сектор, удовлетворяющий заданному условию. При выполнении условия сканирования в регистре состояния ST2 устанавливается флаг SH, в противном случае - флаг SN.

**4.3 Использование команд контроллера НГМД.**

Выполнив сброс контроллера, вам надо его проинициализировать, указав все рабочие параметры. Затем можно выдавать контроллеру команды, каждый раз проверяя регистр основного состояния ST и анализируя байты результата ST0...ST3. Можно предложить следующую последовательность действий:

• сброс контроллера выдачей в порт 3F2h байта с битом 2, установленным в 0;

• разрешение работы контроллера выдачей в этот же порт байта с битом 2, установленным в 1;

• выдача контроллеру команды "Инициализация";

• выдача контроллеру команды "Определить параметры";

• включение двигателя и ожидание в течение примерно 0,5 с (время разгона двигателя);

• установка головки в нужное положение командой "Поиск";

• проверка результатов установки командой "Чтение состояния прерывания";

• для IBM PC/AT установка нужной скорости передачи данных выдачей в порт 3F7h байт с соответствующим значением: 0 для дискет с высокой плотностью записи (HD), 1 для двойной плотности (DD) и 2 для одинарной (SD);

• если установка головок выполнена правильно, можно выдавать команды чтения или записи данных (перед этим надо правильно запрограммировать контроллер прямого доступа к памяти, если вы собираетесь использовать режим ПДП).

**4.4 Немного о контроллере прямого доступа к памяти.**

Приведем основные сведения, необходимые для того чтобы разобраться в программе, демонстрирующей использование команд контроллера НГМД.

Контроллер прямого доступа к памяти (ПДП) имеет несколько каналов и для IBM PC/AT состоит из двух микросхем Intel 8237A (или аналогичных). Контроллер НГМД использует канал с номером 2.

Перед началом инициализации контроллера ПДП программа должна послать в порты 0Bh и 0Ch код операции, которая будет выполняться - 46h для операции чтения и 4Ah для операции записи.

В процессе инициализации программа должна сообщить контроллеру ПДП адрес буфера, куда ему следует поместить данные или откуда надо взять данные, и длину передаваемых данных в байтах.

Адрес необходимо представить в виде номера страницы и смещения. Для контроллера ПДП компьютера IBM PC/AT используется 8-битовый номер страницы и 16-битовое смещение. Например, для адреса 23456h номер страницы будет равен 2h, а смещение - 3456h.

Для программирования канала 2 контроллера ПДП программа должна сначала вывести младший байт смещения в порт с адресом 04h, затем вывести в этот же порт старший байт смещения и, наконец, вывести байт номера страницы в порт с адресом 81h.

Длина передаваемых данных выводится аналогичным образом в порт с адресом 05h - сначала младший байт длины, затем старший.

После определения режима работы канала, адреса буфера и длины передаваемых данных, программа должна разрешить работу контроллера ПДП, выдав в порт с адресом 0Ch байт 2. Теперь канал прямого доступа готов к работе и будет ждать данные от контроллера НГМД.

**Глава 5. Функции BIOS для работы с дисками.**

Наилучший и самый безопасный способ работы с дисками на физическом уровне в среде MS-DOS - использование функций базовой системы ввода/вывода BIOS. Эти функции учитывают все особенности аппаратуры и предоставляют достаточно широкий набор средств доступа к дискам на физическом уровне.

Вся дисковая подсистема обслуживается прерыванием INT 13h . Это прерывание выполняет множество функций. Для вызова определенной функции программа должна занести ее код в регистр AH. При этом в другие регистры следует записать параметры - номер НМД или НГМД, номер цилиндра, головки, адрес таблиц параметров НМД или НГМД и т. д.

Приведем краткую таблицу функций прерывания INT 13h , после чего займемся детальным описанием этих функций. В примечании к описанию функций мы будем указывать типы компьютеров, на которых данная функция работоспособна.

|  |  |
| --- | --- |
| Номер функции | Описание |
| 00h | Сброс дисковой системы  |
| 01h | Определение состояния дисковой системы  |
| 02h | Чтение сектора |
| 03h | Запись сектора |
| 04h | Проверка сектора |
| 05h | Форматирование дорожки  |
| 06h | Форматирование дорожки НМД  |
| 07h | Форматирование НМД |
| 08h | Получить текущие параметры НГМД или НМД  |
| 09h | Инициализация таблиц параметров НМД  |
| 0Ah | Чтение длинное (только для НМД)  |
| 0Bh | Запись длинная (только для НМД)  |
| 0Ch | Поиск цилиндра (только для НМД)  |
| 0Dh | Альтернативный сброс НМД  |
| 0Eh | Чтение буфера сектора (только для НМД)  |
| 0Fh | Запись буфера сектора (только для НМД)  |
| 10h | Проверка готовности НМД  |
| 11h | Рекалибровка НМД |
| 12h | Проверка памяти контроллера НМД  |
| 13h | Проверка НМД |
| 14h | Проверка контроллера НМД  |
| 15h | Получить тип НМД или НГМД  |
| 16h | Проверка замены диска |
| 17h | Установка типа дискеты  |
| 18h | Установка среды носителя данных для форматирования  |
| 19h | Парковка головок (только для НМД)  |
| 1Ah | Форматирование НМД с интерфейсом ESDI  |

**5.1 Сброс дисковой системы.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 00h  |
|  | DL | Адрес устройства НМД или НГМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | - |  |
| Примечание: |  | PC, XT, AT, PS/2 |

Эта функция вызывает сброс и рекалибровку дискового контроллера (в ходе этой процедуры головки устанавливаются на нулевую дорожку). Если в байте адреса старший бит (бит 7) установлен в 1, выполняется сброс контроллера НМД.

Сброс рекомендуется выполнять после того, как произошла ошибка при выполнении других операций, таких как чтение или запись. После сброса можно попытаться повторить операцию.

Адрес устройства, равный 0, соответствует первому НГМД (A:), 1 - второму (B:) и т. д. Адреса 80h, 81h соответствуют первому и второму НМД, соответственно.

**5.2 Определение состояния дисковой системы.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 01  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | AL | Состояние устройства после завершения последней операции  |
| Примечание: |  | PC, XT, AT, PS/2  |

Эта функция может быть использована для анализа результата выполнения операции и получения кода ошибки. Функция берет передаваемый в регистре AL код ошибки из области данных BIOS (байт с адресом 0000h:0441h).

Код ошибки может принимать следующие значения:

|  |  |
| --- | --- |
| Значение | Описание |
| 00h | Успешное завершение операции  |
| 01h | Неправильная команда |
| 02h | Не найдена адресная метка  |
| 03h | Попытка записи на диск, защищенный от записи  |
| 04h | Сектор не найден |
| 05h | Ошибка при сбросе НМД |
| 06h | Произошла замена дискеты  |
| 07h | Неправильные параметры НМД  |
| 08h | Переполнение канала прямого доступа ПДП при работе с НГМД  |
| 09h | Переход за границу 64 Кбайт при работе с ПДП  |
| 0Ah | Обнаружен плохой сектор (только для НМД)  |
| 0Bh | Обнаружена плохая дорожка (только для НМД)  |
| 0Ch | Неправильный номер дорожки  |
| 0Dh | Неправильный номер сектора при форматировании (только для НМД)  |
| 0Eh | Обнаружена адресная метка управляющих данных (только для НМД)  |
| 0Fh | Ошибка ПДП при работе с НМД  |
| 10h | Обнаружена ошибка в контрольной сумме CRC /ECC  |
| 11h | Данные скорректированы с использованием ECC (только для НМД)  |
| 20h | Сбой контроллера |
| 40h | Сбой при поиске дорожки  |
| 80h | Таймаут - программа не успевает обрабатывать данные  |
| AAh | НМД не готов |
| BBh | Неизвестная ошибка (только для НМД)  |
| CCh | Сбой при записи (только для НМД)  |
| E0h | Ошибка регистра состояния НМД  |
| FFh | Ошибка операции чтения (только для НМД)  |

**5.3 Чтение сектора.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 02h  |
|  | AL | Количество секторов, которые нужно прочитать  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
|  | ES:BX | Адрес буфера для данных  |
| На выходе: | AH | Состояние дисковода после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Эта функция позволяет прочитать один или несколько секторов диска в буфер, находящийся в оперативной памяти. Для НМД номер дорожки и номер сектора задаются следующим образом: биты 5...0 регистра CX задают номер сектора, а биты 15...6 - номер дорожки. Перед чтением необходимо подготовить таблицу параметров НГМД или НМД.

**5.4 Запись сектора.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 03h  |
|  | AL | Количество секторов, которые нужно прочитать  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
|  | ES:BX | Адрес буфера для данных  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Функция записи секторов аналогична предыдущей, за исключением направления перемещения данных - данные записываются из буфера в один или несколько секторов диска. Необходимо отметить, что при работе с НГМД не для всякой BIOS будет выполняться ожидание перед выполнением операции записи, пока двигатель раскрутится до рабочей скорости. В результате программа может получить признак ошибки. Прежде чем делать вывод о причинах ошибки, следует сбросить контроллер НГМД функцией 00H и повторить операцию записи три раза.

**5.5 Проверка сектора.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 04h  |
|  | AL | Количество секторов, которые нужно проверить  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | AL | Количество проверенных секторов  |
|  | CF | 1, если произошла ошибка,0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

С помощью этой функции программа может убедиться, что указанные секторы существуют и их можно прочесть. Данные проверяются по методу циклического избыточного контроля (CRC). Адрес буфера не нужен, так как при проверке секторов чтение данных в оперативную память не выполняется.

**5.6 Форматирование дорожки.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 05h  |
|  | AL | Количество секторов, которые нужно создать на дорожке, или фактор чередования для НМД IBM PC/XT  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
|  | ES:BX | Адрес буфера формата, используется для НГМД и НМД, установленных в IBM PC/XT  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Функция форматирования предназначена для начального формирования структуры дорожки диска. Эта функция разрушает все данные, записанные на дорожке. С помощью функции 05h вы можете за один раз отформатировать только одну дорожку с указанным в регистре CH номером.

Для функции форматирования необходимо задать два параметра, на которых мы остановимся подробнее - фактор чередования и адрес буфера формата.

Что такое фактор чередования? Этот фактор определяет последовательность расположения секторов на дорожке. Секторы могут располагаться в порядке своих номеров, через один, через два и т. д. Способ размещения секторов определяется значением фактора чередования.

Некоторые программы, предназначенные для подготовки НМД к работе, требуют указания величины фактора чередования при выполнении низкоуровневого форматирования.

Для чего может понадобиться несмежное расположение секторов с последовательными номерами на дорожке диска? При последовательном расположении секторов может получиться так, что процессор не будет успевать обрабатывать смежные секторы за один проход дорожки.

Например, программа считывает последовательно второй и третий сектор. В момент времени, когда второй сектор уже считан, при быстром вращении диска к моменту начала чтения третьего сектора головки могут оказаться в середине третьего сектора. В результате диск совершит еще один оборот, прежде чем головки окажутся в начале третьего сектора. Поэтому если программа последовательно обращается к смежным секторам, может получиться так, что для чтения каждого сектора потребуется один оборот. Если же секторы будут расположены через один или через два, количество оборотов диска, нужных для обработки последовательности смежных секторов, будет значительно меньше. Для подбора оптимального фактора чередования при форматировании НМД можно использовать специальные программы или делать это методом проб и ошибок, задавая каждый раз новое значение фактора и проверяя быстродействие диска.

Займемся теперь буфером формата. Перед вызовом функции форматирования в регистры ES:BX следует записать полный адрес буфера формата. Перед форматированием дискеты этот буфер должен представлять из себя заполненный массив четырехбайтовых элементов - номер дорожки, головки, сектора и код размера сектора. Код размера сектора может иметь следующие значения:

|  |  |
| --- | --- |
| Код | Размер сектора в байтах  |
| 0 | 128 |
| 1 | 256 |
| 2 | 512 |
| 3 | 1024 |

Количество элементов в массиве должно быть равно количеству секторов, создаваемых на дорожке. То есть для каждого сектора буфер формата должен содержать один четырехбайтовый элемент. Для НМД буфер формата должен представлять из себя массив размером 512 байт. В начале этого массива для каждого сектора необходимо подготовить двухбайтовые элементы. Первый байт содержит признак - хороший это сектор (00) или плохой (80h), второй байт - номер сектора. Задавая последовательность номеров в буфере формата соответствующим образом, программа определяет фактор чередования.

**5.7 Форматирование дорожки НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 06h  |
|  | AL | Фактор чередования  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
|  | ES:BX | Адрес буфера формата  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT  |

Функция форматирования дорожки с кодом 6 предназначена только для НМД. При необходимости она устанавливает флаг плохого сектора. Буфер формата подготавливается таким же образом, что и для функции 05h.

**5.8 Определение текущих параметров НГМД или НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 08h  |
|  | DL | Адрес НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
|  | BL | Тип устройства (только для IBM PC/AT и PS/2)  |
|  | DL | Кол-во устройств, подключенных к первому контроллеру диска  |
|  | DH | Максимальный номер головки  |
|  | CL | Максимальный номер сектора  |
|  | CH | Максимальный номер цилиндра  |
|  | ES:DI | Адрес таблицы параметров НГМД  |
| Примечание:  |  | PC, XT, AT, PS/2  |

С помощью этой функции программа может определить тип НГМД или НМД, количество устройств, подключенных к первому контроллеру НМД и другие параметры, которые нужны программе для организации доступа к диску на физическом уровне.

**5.9 Инициализация контроллера НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 09h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Функцию инициализации контроллера НМД применяют после изменения таблиц параметров жесткого диска. Функции BIOS "узнают" о внесенных в таблицы изменениях и инициализируют соответствующим образом контроллер НМД.

**5.10 Чтение секторов длинное (НМД).**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 0Ah  |
|  | AL | Количество секторов, которые нужно прочитать  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
|  | ES:BX | Адрес буфера для данных  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Функция "Чтение секторов длинное" отличается от обычной функции чтения (с кодом 02h) тем, что она дополнительно считывает в буфер данных 4 байта кода коррекции ошибки (ECC).

**5.11 Поиск дорожки (НМД).**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 0Ch  |
|  | CH | Номер дорожки  |
|  | CL | Номер сектора  |
|  | DH | Номер головки  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

С помощью функции 0Ch программа может подвести головки к дорожке с заданным номером. Функции чтения и записи секторов не требуют предварительного поиска дорожки, они выполняют поиск самостоятельно.

**5.12 Альтернативный сброс НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 0Dh  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Вы можете использовать эту функцию для сброса контроллера вместо функции с кодом 00h. В отличие от функции сброса дисковой системы с кодом 00h, эта функция не влияет на контроллер НГМД, она сбрасывает только контроллер накопителя на жестком магнитном диске.

**5.13 Чтение буфера сектора (НМД).**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 0Eh  |
|  | ES:BX | Адрес буфера для данных  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT  |

Контроллеры НМД, как правило, имеют внутренний буфер данных. С помощью функции 0Eh программа может прочитать содержимое этого буфера в оперативную память. Обращение к диску при этом не происходит. Функция чтения буфера используется в основном для диагностики контроллера.

**5.14 Запись буфера сектора (НМД).**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 0Fh  |
|  | ES:BX | Адрес буфера для данных  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT  |

Функция полностью аналогична предыдущей, за исключением того, что происходит не чтение, а запись данных из оперативной памяти в буфер контроллера. Она может быть использована для инициализации содержимого буфера сектора перед форматированием диска функцией 05h прерывания INT 13h.

**5.15 Определение состояния НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 10h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

О готовности НМД можно судить по байту состояния, передаваемому функцией в регистре AH. Аналогичный байт возвращается в регистре AH функцией 01h.

**5.16 Рекалибровка НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 11h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Вызов функции приводит к позиционированию головок выбранного НМД на нулевую дорожку. Дополнительно в регистре AH возвращается байт состояния устройства.

**5.17 Проверка памяти контроллера НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 12h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT  |

Эта функция предназначена для запуска встроенной диагностики дискового контроллера, она проверяет внутренний буфер сектора и возвращает байт состояния.

**5.18 Проверка НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 13h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT  |

Функция используется для запуска внутренней диагностики контроллера.

**5.19 Проверка контроллера НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 14h  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PC, XT, AT, PS/2  |

Функция запускает внутреннюю диагностику контроллера.

**5.20 Определение типа НГМД или НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 15h  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | AH | Тип НГМД или НМД  |
|  | CX:DX | Количество секторов размером 512 байт  |
| Примечание: |  | AT, PS/2  |

С помощью этой функции программа может определить тип дискового устройства и возможность обнаружения замены магнитного носителя данных (как правило, дискеты).

**5.21 Проверка замены диска.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 16h  |
|  | DL | Адрес устройства НГМД или НМД (0, 1, ..., 80h, 81h, ...)  |
| На выходе: | AH | Признак замены носителя данных: 0 - замены не было; 6 - носитель данных был заменен  |
| Примечание: |  | AT, PS/2  |

В некоторых случаях замена носителя данных (дискеты или сменного магнитного диска) нежелательна до выполнения определенных действий (мы говорили об этом при обсуждении драйверов дисковых устройств). С помощью этой функции программа может убедиться в том, что в дисковом устройстве установлен все тот же носитель данных, что и в начале цикла операций. Если носитель данных был по ошибке заменен раньше времени, программа может потребовать установить старый носитель для завершения работы с ним.

**5.22 Установка типа дискеты.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 17h  |
|  | AL | Тип дискеты  |
|  | DL | Адрес НГМД (0, 1, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | AT, PS/2  |

Если перед вызовом этой функции был установлен флаг замены дискеты, то он сбрасывается. Дополнительно BIOS устанавливает скорость передачи данных через контроллер НГМД в зависимости от типа дискеты.

**5.23 Установка среды для форматирования НГМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 18h  |
|  | DL | Адрес НГМД (0, 1, ...)  |
|  | CH | Младшие 8 бит количества дорожек  |
|  | CL | Количество секторов на дорожку (биты 0-5)  |
| На выходе: | AH | 00h - указанная комбинация количества дорожек и количества секторов на дорожку поддерживается операцией форматирования  |
|  |  | 01h - функция недоступна  |
|  |  | 0Ch - функция не поддерживается или неизвестен тип дисковода  |
|  |  | 80h - дискета не установлена в НГМД  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | AT, PS/2  |

Эта функция должна быть вызвана перед использованием функции 05h форматирования дискеты для установки правильной скорости передачи данных через контроллер. Дополнительно функция сбрасывает флаг замены дискеты (если этот флаг был установлен).

**5.24 Парковка головок НМД.**

|  |  |  |
| --- | --- | --- |
| На входе: | AH | 19h  |
|  | DL | Адрес НМД (80h, 81h, ...)  |
| На выходе: | AH | Состояние устройства после завершения последней операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |
| Примечание: |  | PS/2  |

Парковка головок - это их установка в нерабочую область, т. е. на нерабочую дорожку. Такую операцию обычно выполняют перед транспортировкой компьютера для исключения повреждения НМД. Современные НМД выполняют парковку головок автоматически.

**Глава 6. Работа с дисками по средствам MS-DOS.**

До сих пор при работе с дисками мы не обращались за помощью к MS-DOS, выполняя все дисковые операции либо на уровне команд контроллера НГМД, либо на уровне функций BIOS. Операционная система предоставляет намного более удобные средства для работы с диском, чем обращение к отдельным секторам по их номеру, а также по номеру дорожки и головки.

Если возможности MS-DOS по обслуживанию диска вас устраивают, то лучше пользоваться именно функциями MS-DOS. В этом случае вам не придется заботиться о многих мелочах и вы будете застрахованы от некоторых ошибок. Кроме того, ваша программа будет меньше зависеть от типа и конфигурации компьютера, так как дисковый драйвер MS-DOS скроет от вас многие детали и особенности реализации системы.

**6.1 Таблица разделов и логические диски.**

Персональный компьютер обычно комплектуется одним или двумя НМД. Однако операционная система позволяет вам разбивать НМД на части, причем каждая часть будет рассматриваться MS-DOS как отдельный, "логический" диск.

Существует несколько причин, по которым может быть полезно разделение большого диска на части:

* в случае повреждения логического диска пропадает только та информация, которая находилась на этом логическом диске;
* реорганизация и выгрузка диска маленького размера выполняется быстрее, чем большого;
* на одном диске может находиться несколько различных операционных систем, расположенных в разных разделах. В ходе начальной загрузки вы можете указать раздел диска, из которого должна загружаться нужная вам в данный момент операционная система.

По своей внутренней структуре логический диск полностью соответствует дискете, поэтому сначала мы изучим логическую структуру жесткого диска, затем сделаем некоторые замечания, касающиеся дискет.

**6.2 Главная загрузочная запись.**

Самый первый сектор жесткого диска (сектор 1, дорожка 0, головка 0) содержит так называемую главную загрузочную запись (Master Boot Record). Эта запись занимает не весь сектор, а только его начальную часть.

Сама по себе главная загрузочная запись является программой. Эта программа во время начальной загрузки операционной системы с НМД помещается по адресу 7C00h:0000h, после чего ей передается управление. Загрузочная запись продолжает процесс загрузки операционной системы.

**6.3 Таблица разделов диска.**

В конце самого первого сектора жесткого диска располагается таблица разделов диска (Partition Table). Эта таблица содержит четыре элемента, описывающих до четырех разделов диска. В последних двух байтах сектора находится значение 55AAh. Это признак таблицы разделов (сигнатура таблицы разделов).

Элемент таблицы разделов диска представляет собой структуру размером 16 байт, соответствующую части диска, называемой разделом. В структуре располагается информация о расположении и размере раздела в секторах, а также о назначении раздела.

Разделы диска бывают активными или неактивными. Активный раздел может использоваться для загрузки операционной системы. Заметим, что диск может содержать одновременно несколько активных разделов, которые могут принадлежать разным операционным системам.

Приведем формат первого сектора жесткого диска:

|  |  |  |
| --- | --- | --- |
| Смещение, байт | Размер, байт  | Описание |
| 0 | 1BEh | Загрузочная запись  |
| 1BEh | 10H | Элемент таблицы разделов диска  |
| 1CEh | 10H | Элемент таблицы разделов диска  |
| 1DEh | 10H | Элемент таблицы разделов диска  |
| 1EEh | 10H | Элемент таблицы разделов диска  |
| 1FEh | 2 | Признак таблицы разделов - значение 55AAh  |

Все элементы таблицы разделов диска имеют одинаковый формат:

|  |  |  |
| --- | --- | --- |
| Смещение, байт | Размер, байт  | Описание |
| 0 | 1 | Признак активного раздела: 0 - раздел неактивный; 80h - раздел активный  |
| 1 | 1 | Номер головки для начального сектора раздела  |
| 2 | 2 | Номер сектора и дорожки для начального сектора раздела в формате функции чтения сектора INT 13h  |
| 4 | 1 | Код системы: 0 - неизвестная система; 1, 4 - MS-DOS; 5 - расширенный раздел MS-DOS  |
| 5 | 1 | Номер головки для последнего сектора раздела  |
| 6 | 2 | Номер сектора и дорожки для последнего сектора раздела в формате функции чтения сектора INT 13h  |
| 8 | 4 | Относительный номер сектора начала раздела  |
| 12 | 4 | Размер раздела в секторах  |

**6.4 Загрузочная запись.**

В самом первом секторе активного раздела расположена загрузочная запись (Boot Record), которую не следует путать с главной загрузочной записью (Master Boot Record). Загрузочная запись считывается в оперативную память главной загрузочной записью, после чего ей передается управление. Загрузочная запись и выполняет загрузку операционной системы.

**6.5 Загрузка операционной системы.**

Загрузка операционной системы с жесткого диска - двухступенчатый процесс. Вначале модули инициализации BIOS считывают главную загрузочную запись в память по адресу 7C00h:0000h и передают ей управление. Главная загрузочная запись просматривает таблицу разделов и находит активный раздел. Если активных разделов несколько, на консоль выводится сообщение о необходимости выбора активного раздела для продолжения загрузки.

После того как активный раздел найден, главная загрузочная запись считывает самый первый сектор раздела в оперативную память. Этот сектор содержит загрузочную запись, которой главная загрузочная запись и передает управление.

Загрузочная запись активного раздела выполняет загрузку операционной системы, находящейся в активном разделе.

Такой двухступенчатый метод загрузки операционной системы необходим по той причине, что способ загрузки зависит от самой операционной системы. Поэтому каждая операционная система имеет свой собственный загрузчик. Фиксированным является только расположение загрузочной записи - самый первый сектор активного раздела.

**6.6 Поля элемента таблицы раздела диска.**

Расскажем подробнее о некоторых полях элемента таблицы раздела диска. Байт со смещением 0, как мы уже говорили, является флагом активного раздела и может принимать одно из двух значений - 0 или 80h, соответственно, для неактивного и активного разделов диска. Слово, имеющее размер 2 байта и расположенное со смещением 8, содержит относительный номер первого сектора раздела. Как он вычисляется? Значение 0 соответствует дорожке 0, головке 0, сектору 1. При увеличении относительного номера сектора вначале увеличивается номер сектора на дорожке, затем номер головки, и, наконец, номер дорожки. Для вычисления относительного номера сектора можно использовать следующую формулу:

*RelSect = (Cyl \* Sect \* Head) + (Head \* Sect) + (Sect -1)*

В этой формуле Cyl - номер дорожки, Sect - номер сектора на дорожке, Head - номер головки.

Замечание относительно границ разделов диска: обычно разделы начинаются с четных номеров дорожек, за исключением самого первого раздела. Этот раздел может начинаться с сектора 2 нулевой дорожки (головка 0), так как самый первый сектор диска занят главной загрузочной записью.

Байт со смещением 4 - это код системы, использующей раздел диска. Для MS-DOS зарезервированы значения 0, 1, 4, 5. Значение 0 соответствует свободному разделу диска.

**6.7 Первичный и расширенный раздел.**

Если код системы в элементе таблицы раздела равен 1 или 4, это означает, что раздел используется MS-DOS в качестве первичного раздела (Primary Partition). Этот раздел обычно является активным и из него выполняется загрузка операционной системы. В зависимости от того, какой код системы указан для первичного раздела (1 или 4) меняется одна из характеристик логического диска - размер элемента таблицы размещения файлов (FAT). Код 1 используется для обозначения 12-битовой FAT, 4 - для 16-битовой FAT. Таблица размещения файлов будет описана ниже в этой главе. Значение кода системы, равное 5, обозначает расширенный раздел MS-DOS (Extended DOS Partition).

Нетрудно заметить, что, даже используя все элементы таблицы разделов для создания логических дисков, невозможно создать более четырех дисков. В расширенном разделе MS-DOS вы можете создать любое количество логических дисков.

Программа fdisk.exe позволяет вам создать один первичный раздел MS-DOS и один расширенный раздел . Первичный раздел должен быть активным, он используется как диск С: и из него выполняется загрузка операционной системы. Расширенный раздел разбивается программой fdisk.exe на логические диски D:, E: и т. д. Расширенный раздел не может быть активным, следовательно, невозможно выполнить загрузку операционной системы с логических дисков, расположенных в этом разделе.

Если байт кода системы имеет значение 5, то в начале соответствующего раздела располагается сектор, содержащий таблицу логических дисков. Фактически эта таблица является расширением таблицы разделов диска, расположенной в самом первом секторе физического диска.

Таблица логических дисков имеет формат, аналогичный таблице разделов диска, но содержит только два элемента. Один из них указывает на первый сектор логического диска MS-DOS, он имеет код системы 1 или 4. Второй элемент может иметь код системы 5 или 0. Если этот код равен 5, то элемент указывает на следующую таблицу логических дисков. Если код системы равен 0, то соответствующий элемент не используется.

Из сказанного выше следует, что таблицы логических дисков связаны в список, на начало этого списка указывает элемент таблицы разделов диска с кодом системы, равным 5.

Для таблицы логических дисков имеется отличие в использовании полей границ логических дисков. Если код системы равен 1 или 4, эти границы вычисляются относительно начала расширенного раздела. Для элемента с кодом системы 5 используется абсолютная адресация (относительно физического начала диска).

**6.8 Загрузочная запись.**

Самый первый сектор логического диска (и самый первый сектор на системной дискете) занимает загрузочная запись (Boot Record). Эта запись считывается из активного раздела диска программой главной загрузочной записи (Master Boot Record) и запускается на выполнение. Задача загрузочной записи - выполнить загрузку операционной системы. Каждый тип операционной системы имеет свою загрузочную запись. Даже для разных версий одной и той же операционной системы программа загрузки может выполнять различные действия.

Кроме программы начальной загрузки операционной системы в загрузочной записи находятся параметры, описывающие характеристики данного логического диска. Все эти параметры располагаются в самом начале сектора, в его так называемой форматированной области. Формат этой области изменился в версии 4.0 операционной системы MS-DOS.

**6.9 Формат загрузочной записи.**

Сначала приведем формат загрузочной записи для версий MS-DOS, более ранних, чем 4.0.

|  |  |  |
| --- | --- | --- |
| **Смещение** | **Размер** | **Содержимое**  |
| 0 | 3 | Команда JMP xxxx - ближний переход на программу начальной загрузки  |
| 3 | 8 | Название фирмы-изготовителя операционной системы и версия, например: "IBM 4.0"  |
| 11 | 13 | Блок параметров BIOS (BPB)  |
| 24 | 2 | Количество секторов на дорожке  |
| 26 | 2 | Количество головок (поверхностей диска)  |
| 28 | 2 | Количество скрытых секторов, эти секторы могут использоваться для схемы разделения физического диска на разделы и логические диски  |

В самом начале загрузочного сектора располагается команда внутрисегментного перехода JMP. Она нужна для обхода форматированной зоны сектора и передачи управления загрузочной программе, располагающейся со смещением 30. Название фирмы-изготовителя не используется операционной системой.

Со смещением 11 располагается BPB - блок параметров BIOS , о котором мы уже говорили в разделах книги, посвященных драйверам. Этот блок содержит некоторые характеристики логического диска, о которых мы будем говорить немного позже. Он активно используется дисковыми драйверами. Для MS-DOS версий до 4.0 блок BPB имеет следующий формат:

|  |  |  |  |
| --- | --- | --- | --- |
| **Смещение, байт** | **Размер, байт**  | **Имя поля** | **Описание** |
| 0 | 2 | sect\_siz  | Количество байт в одном секторе диска |
| 2 | 1 | clustsiz  | Количество секторов в одном кластере |
| 3 | 2 | res\_sect  | Количество зарезервированных секторов |
| 5 | 1 | fat\_cnt  | Количество таблиц FAT  |
| 6 | 2 | root\_siz  | Максимальное количество дескрипторов файлов в корневом каталоге диска  |
| 8 | 2 | tot\_sect  | Общее количество секторов на носителе данных (в разделе MS-DOS)  |
| 10  | 1 | media  | Байт-описатель среды носителя данных |
| 11  | 2 | fat\_size  | Количество секторов, занимаемых одной копией FAT  |

Поля загрузочного сектора со смещениями 24 и 26 содержат, соответственно, количество секторов на дорожке и количество головок в НМД. Поле со смещением 28 содержит количество "скрытых" секторов, которые не принадлежат ни одному логическому диску. Эти секторы могут содержать основную или вторичные таблицы разделов диска.

Для современных версий MS-DOS загрузочный сектор имеет другой формат:

|  |  |  |
| --- | --- | --- |
| **Смещение** | **Размер** | **Содержимое**  |
| 0 | 3 | Команда JMP xxxx - ближний переход на программу начальной загрузки  |
| 3 | 8 | Название фирмы-изготовителя операционной системы и версия  |
| 11 | 25 | Extended BPB - расширенный блок параметров BIOS  |
| 36 | 1 | Физический номер устройства (0 -НГМД, 80h -НМД)  |
| 37 | 1 | Зарезервировано  |
| 38 | 1 | Символ ')' - признак расширенной загрузочной записи  |
| 39 | 4 | Серийный номер диска (Volume Serial Number), создается во время форматирования диска  |
| 43 | 11 | Метка диска (Volume Label)  |
| 54 | 8 | Зарезервировано, обычно содержит запись типа 'FAT12 ', которая идентифицирует формат таблицы размещения файлов FAT  |

Первые два поля в загрузочном секторе аналогичны описанным раньше. Поле со смещением 38 всегда содержит символ ')'. Этот символ означает, что используется формат расширенной загрузочной записи.

Серийный номер диска формируется во время форматирования диска на основе даты и времени форматирования. Это поле может быть использовано для определения факта замены дискеты.

Метка диска формируется при форматировании и может быть изменена командой LABEL операционной системы MS-DOS. Одновременно метка диска помещается в корневой каталог.

**6.10 Расширенный блок параметров BIOS.**

Поле загрузочного сектора со смещением 11 содержит расширенный блок параметров BIOS . Он состоит из обычного блока BPB и дополнительного расширения:

|  |  |  |  |
| --- | --- | --- | --- |
| **Смещение, байт** | **Размер, байт**  | **Имя поля** | **Описание** |
| 0 | 2 | sect\_siz  | Количество байт в одном секторе диска |
| 2 | 1 | clustsiz  | Количество секторов в одном кластере |
| 3 | 2 | res\_sect  | Количество зарезервированных секторов |
| 5 | 1 | fat\_cnt  | Количество таблиц FAT  |
| 6 | 2 | root\_siz  | Максимальное количество дескрипторов файлов в корневом каталоге диска  |
| 8 | 2 | tot\_sect  | Общее количество секторов на носителе данных (в разделе MS-DOS)  |
| 10  | 1 | media  | Байт-описатель среды носителя данных |
| 11  | 2 | fat\_size  | Количество секторов, занимаемых одной копией FAT  |
| 13  | 2 | sectors  | Количество секторов на дорожке |
| 15  | 2 | heads  | Количество магнитных головок |
| 17  | 2 | hidden\_l  | Количество скрытых секторов для раздела, который по размеру меньше 32 Мбайт  |
| 19  | 2 | hidden\_h  | Количество скрытых секторов для раздела, превышающего по размеру 32 Мбайт  |
| 21  | 4 | tot\_secs  | Общее количество секторов на логическом диске для раздела, превышающего по размеру 32 Мбайт  |

Как обычный, так и расширенный блок параметров BIOS содержит байт-описатель среды media. Этот байт может служить для идентификации носителя данных и может содержать следующие величины, характеризующие носитель данных по количеству сторон диска и количеству секторов на дорожке:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Значение** | **Количество сторон** | **Количество секторов** | **Диаметр, дюймы**  | **Емкость, Кбайт** |
| F0h | 2 | 18 | 3,5 | 1440 |
| - " - | 2 | 36  | 3,5 | 2880 |
| - " - | 2 | 15  | 5,25 | 1200 |
| F8h | - | - |   | Жесткий диск любой емкости |
| F9h | 2 | 9 | 3,5 | 720 |
| - " - | 2 | 15  | 5,25 | 1200 |
| FAh | 1 | 8 | 5,25 | 320 |
| FBh | 2 | 8 | 3,5 | 640 |
| FCh | 1 | 9 | 5,25 | 180 |
| FDh | 2 | 9 | 5,25 | 360 |
| FEh | 1 | 8 | 5,25, 8 | 160 |
| FFh | 2 | 8 | 5,25, 8 | 320 |

Прежде чем мы продолжим изучение логической структуры диска, покажем, как программа может получить содержимое загрузочного сектора.

**6.11 Логический номер сектора.**

MS-DOS предоставляет программе возможность работы с так называемыми логическими номерами секторов. Это номера секторов внутри логического диска.

Вы знаете, что для адресации сектора при помощи функций BIOS необходимо указывать номер дорожки, номер головки и номер сектора на дорожке. MS-DOS организует "сквозную" нумерацию секторов, при которой каждому сектору логического диска присваивается свой номер. Порядок нумерации выбран таким, что при последовательном увеличении номера сектора вначале увеличивается номер головки, затем номер дорожки. Это сделано для сокращения перемещений блока головок при обращении к последовательным логическим номерам секторов.

Пусть, например, у нас есть дискета с девятью секторами на дорожке. Сектор с логическим номером, равным 1, расположен на нулевой дорожке и для обращения к нему используется нулевая головка. Это самый первый сектор на дорожке, он имеет номер 1. Следующий сектор на нулевой дорожке имеет логический номер 2, последний сектор на нулевой дорожке имеет логический номер 9. Сектор с логическим номером 10 расположен также на нулевой дорожке. Это тоже самый первый сектор на дорожке, но теперь для доступа к нему используется головка с номером 1. И так далее, по мере увеличения логического номера сектора изменяются номера головок и дорожек.

**6.12 Прерывания INT 25h и INT 26h.**

Для работы с логическим диском (или дискетой) на уровне логических номеров секторов MS-DOS предоставляет программам два прерывания - INT 25h (чтение сектора по его логическому номеру) и INT 26h (запись сектора по его логическому номеру). Вызов этих прерываний имеет различный формат для разных версий MS-DOS. Для тех версий, которые не поддерживают размер логических дисков более 32 Мбайт (MS-DOS 3.10, 3.20, 3.30) используется следующий формат:

INT 25h - Чтение сектора по его логическому номеру

|  |  |  |
| --- | --- | --- |
| На входе: | AL | Адрес НГМД или НМД (0 - A:, 1 - B:, ...)  |
|  | CX | Количество секторов, которые нужно прочитать  |
|  | DX | Логический номер начального сектора  |
|  | DS:BX | Адрес буфера для чтения  |
| На выходе: | AH | Код ошибки при неуспешном завершении операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |

INT 26h - Запись сектора по его логическому номеру

|  |  |  |
| --- | --- | --- |
| На входе: | AL | Адрес НГМД или НМД (0 - A:, 1 - B:, ...)  |
|  | CX | Количество секторов, которые нужно записать  |
|  | DX | Логический номер начального сектора  |
|  | DS:BX | Адрес буфера, содержащего данные  |
| На выходе: | AH | Код ошибки при неуспешном завершении операции  |
|  | CF | 1, если произошла ошибка, 0, если ошибки нет  |

Для более поздних версий MS-DOS и для COMPAQ DOS версии 3.31 используется другой способ указания номера логического сектора.

Так как шестнадцати разрядов недостаточно для адресации диска размером более 32 Мбайт, то при работе с расширенным разделом диска, занимающим более 32 Мбайт, регистры используются по-другому.

Регистр CX содержит FFFFh - признак того, что программа работает с логическим диском, имеющим размер более 32 Мбайт. Регистры DS:BX содержат адрес следующей структуры:

|  |  |  |
| --- | --- | --- |
| **Смещение** | **Размер** | **Содержимое**  |
| 0 | 4 | Начальный номер логического сектора  |
| 4 | 2 | Количество секторов для чтения или записи  |
| 6 | 4 | Дальний адрес буфера для передачи данных  |

Так как для указания начального номера логического сектора в этом управляющем блоке отводится 4 байта, то снимается указанное ранее ограничение на размер логического диска.

Сделаем очень важное замечание, касающееся только что рассмотренных прерываний MS-DOS.

Эти прерывания оставляют в стеке одно слово - старое значение регистра флагов. Поэтому после вызова прерывания должна следовать, например, такая команда:

*pop ax*

Содержимое загрузочного сектора может быть использовано для определения общего количества секторов на логическом диске, для работы с таблицей размещения файлов FAT, о которой мы будем говорить ниже, для определения других характеристик логического диска.

**6.13 Таблица размещения файлов.**

Сразу вслед за загрузочным сектором на логическом диске находятся секторы, содержащие таблицу размещения файлов FAT (File Allocation Table). Для того, чтобы назначение этой таблицы стало более понятным, вспомним, как организовано хранение информации на различных носителях данных.

**6.14 Последовательный и прямой доступ.**

Начнем с магнитных лент. При использовании магнитных лент информация записывается в виде файлов с последовательным доступом. Последовательный доступ означает, что для чтения какого-либо файла требуется вначале прочитать (или просмотреть) все предыдущие файлы. При записи информация может добавляться в конец ленты, после той информации, которая была записана в последний раз.

Для такого устройства, как магнитный диск, возможна запись информации либо последовательным, либо прямым методом доступа. Использование прямого метода доступа позволяет программе позиционировать головки сразу на тот файл, который вам нужен (или на нужную запись файла). Например, при чтении записи вы можете задать номер сектора на определенной дорожке и номер головки, где она расположена, либо смещение записи относительно начала файла в байтах.

Как правило, прямой метод доступа более эффективен, однако важное значение имеет также способ распределения места на диске для файлов.

**6.15 Кластеры.**

Операционная система ОС ЕС для ЭВМ ряда ЕС позволяла задать начальное количество цилиндров диска для размещения набора данных и размер области диска, которая может быть использована для этого набора дополнительно. Если при записи в файл все распределенное для файла место на диске окажется исчерпанным, программа завершится аварийно, даже если на диске еще есть свободные цилиндры.

Операционная система MS-DOS использует дисковое пространство другим способом.

При создании файла для него не задается начальное распределение памяти в дорожках или секторах. По мере того как файл увеличивается в размерах, операционная система распределяет этому файлу секторы из числа свободных, не используемых другими файлами. При этом файл не обязательно располагается в смежных областях диска, он может быть разбросан по разным дорожкам и секторам.

Очевидно, что в этом случае операционная система должна вести учет используемых секторов диска. Для каждого файла она должна хранить где-то информацию о распределении файлам секторов диска. В операционной системе MS-DOS для хранения этой информации используется таблица размещения файлов.

Весь диск разбивается операционной системой на участки одинакового размера, называемые кластерами. Кластер может содержать несколько секторов. Для каждого кластера в таблице FAT есть своя индивидуальная ячейка, в которой хранится информация об использовании данного кластера. Другими словами, таблица размещения файлов - это массив, содержащий информацию о кластерах. Размер этого массива определяется общим количеством кластеров на логическом диске.

**6.16 Содержимое таблицы FAT.**

Что же хранится в таблице размещения файлов?

Если ответить на этот вопрос коротко, то в FAT находятся списки кластеров, распределенных файлам. Все свободные кластеры отмечены нулями.

Таким образом, если файл занимает несколько кластеров, то эти кластеры связаны в список. При этом элементы таблицы FAT содержат номера следующих используемых данным файлом кластеров. Конец списка отмечен в таблице специальным значением. Номер первого кластера, распределенного файлу, хранится в элементе каталога, описывающего данный файл.

Программа format.com, предназначенная для форматирования диска и некоторые специальные программы аналогичного назначения проверяют диск на предмет наличия дефектных областей. Кластеры, которые находятся в этих дефектных областях, отмечаются в FAT как плохие и не используются операционной системой.

Итак, FAT - массив информации об использовании кластеров диска, содержит списки кластеров, распределенных файлам. Номера начальных кластеров файлов хранятся в каталогах, о которых мы будем говорить в разделе "Файлы и каталоги".

В каталоге кроме всего прочего указаны номера первых кластеров, распределенных этим файлам (соответственно 11 и 12). В своей одиннадцатой ячейке таблица FAT содержит число 17 - номер второго кластера, распределенного файлу autoexec.bat. Ячейка с номером 17 содержит число 18. Это номер третьего кластера, принадлежащего файлу autoexec.bat. Последняя ячейка, которая соответствует последнему кластеру, распределенному этому файлу, содержит специальное значение - FFFF.

Таким образом, файл autoexec.bat занимает три несмежных кластера с номерами 11, 17 и 18. Что же касается файла config.sys , то в нашем примере для него отведено два смежных кластера с номерами 12 и 13.

* 1. **Два формата таблицы FAT.**

Таблица FAT может иметь 12- или 16-битовый формат. При этом в таблице для хранения информации об одном кластере диска используется, соответственно, 12 и 16 бит.

12-битовый формат удобен для дискет с небольшим количеством секторов - вся таблица размещения файлов помещается в одном секторе.

Если размер диска такой, что для представления всех секторов недостаточно двенадцати разрядов, можно увеличить размер кластера, например до восьми секторов.

Однако большой размер кластера приводит к неэффективному использованию дискового пространства. Это происходит из-за того что минимальный фрагмент дисковой памяти, выделяемый файлу, имеет слишком большой размер. Даже для файла размером 1 байт выделяется целый кластер. Значит, если размер кластера составляет 8 секторов, то для хранения одного байта будет использовано 4 Кбайт дисковой памяти (размер кластера составляет 512 байт).

При использовании 16-битового формата таблицы FAT операционная система MS-DOS может работать с диском, который имеет размер более 32 Мбайт.

* 1. **Определение формата таблицы FAT.**

Сектор загрузочной записи диска, отформатированного в MS-DOS версий 4.0 - 6.22 в поле со смещением 36h содержит строку длиной 8 байт, идентифицирующую формат FAT. Она имеет вид "FAT12" или "FAT16".

Если разделы на жестком диске создавались программой fdisk.exe , формат FAT можно определить, анализируя содержимое поля sys главной загрузочной записи (Master Boot Record). Если это поле содержит значение 1, используется 12-битовый формат, если 4, то 16-битовый.

* 1. **Идентификация кластеров.**

Первый байт таблицы FAT называется "Описатель среды" (Media Descriptor). Он имеет такое же значение, как и байт-описатель среды, находящийся в загрузочном секторе логического диска.

Следующие 5 байт для 12-битового формата или 7 байт для 16-битового формат всегда содержат значение 0FFh.

Остальная часть таблицы FAT состоит из 12- или 16-битовых ячеек. Каждая ячейка соответствует одному кластеру диска. Эти ячейки могут содержать следующие значения:

|  |  |  |
| --- | --- | --- |
| **FAT12** | **FAT16** | **Что означает**  |
| 000h | 0000h | Свободный кластер  |
| FF0h - FF6h | FFF0h - FFF6h | Зарезервированный кластер |
| FF7h | FFF7h | Плохой кластер  |
| FF8h - FFFh | FFF8h - FFFFh | Последний кластер в списке |
| 002h - FEFh | 0002h - FFEFh | Номер следующего кластера в списке |

* 1. **Чтение таблицы FAT.**

Непосредственный доступ к FAT может потребоваться вам для организации сканирования каталогов при поиске файлов, для чтения каталогов как файлов, для организации защиты информации от несанкционированного копирования. Общая схема использования FAT такая:

* Читаем таблицу FAT в память

Обычно FAT располагается сразу после загрузочного сектора (логический сектор с номером 1). Для точного определения начального сектора FAT следует прочитать в память загрузочный сектор и проанализировать содержимое блока параметров BIOS. В поле ressecs записано количество зарезервированных секторов, которые располагаются перед FAT. Поле fatsize содержит размер FAT в секторах. Кроме того, следует учитывать, что на диске может находиться несколько копий FAT. Операционная система использует только первую копию, но обновляет вторую. Другие копии FAT нужны для утилит восстановления содержимого диска, таких как scandisk.exe . Количество копий FAT находится в поле fatcnt загрузочного сектора.

* Получаем номер первого кластера файла, для которого необходимо определить его расположение на диске
* Используем номер первого кластера как индекс в таблице FAT для извлечения номера следующего кластера
* Повторяем предыдущую процедуру до тех пор, пока извлеченное из FAT значение не будет соответствовать концу файла

Процедура извлечения номера кластера из FAT зависит от формата таблицы размещения файлов.

16-битовую таблицу FAT можно представить как массив 16-битовых чисел. Для определения номера следующего кластера вам надо просто извлечь 16-битовое значение из FAT, использовав в качестве индекса номер предыдущего кластера.

Для 12-битовой таблицы FAT процедура значительно сложнее. Необходимо выполнить следующие действия:

* умножить номер начального кластера на 3;
* разделить результат на 2 (так как каждый элемент таблицы имеет длину 1,5 байта);
* прочитать 16-битовое слово из FAT , используя в качестве смещения значение, полученное после деления на 2;
* если номер начального кластера четный, на выбранное из FAT слово надо наложить маску 0FFFh, оставив младшие 12 бит, если же номер начального кластера нечетный, выбранное из FAT значение необходимо сдвинуть вправо на 4 бита, оставив старшие 12 бит;
* полученный результат - это номер следующего кластера в цепочке, при этом значение 0FFFh (или другое в диапазоне от 0FF8h до 0FFFh) соответствует концу цепочки кластеров.

Используя описанные выше процедуры просмотра FAT , вы сможете для каждого файла определить цепочку занимаемых им кластеров. Для чтения файла при помощи прерывания INT 25h вам будет нужно установить соответствие между номерами кластеров и номерами секторов, в которых располагаются эти кластеры. Для того чтобы это сделать, необходимо определить расположение и размер корневого каталога. Поэтому следующий раздел книги будет посвящен каталогам и файлам. Там же будут приведены примеры программ для работы с FAT.

* 1. **Файлы и каталоги.**

Вы, конечно, знаете, что файловая система MS-DOS имеет древовидную структуру. В корневом каталоге располагаются 32-байтовые элементы, которые содержат информацию о файлах и других каталогах. Для чтения корневого каталога необходимо определить его расположение и размер.

**6.22 Расположение и размер корневого каталога.**

Корневой каталог находится сразу за последней копией FAT . Количество секторов, занимаемых одной копией FAT, находится в блоке параметров BIOS в загрузочном секторе (поле fatsize), а количество копий FAT - в поле fatcnt блока BPB . Следовательно, перед корневым каталогом находится один загрузочный сектор и fatcnt \* fatsize секторов таблицы размещения файлов FAT.

Размер корневого каталога можно определить исходя из значения поля rootsize. При форматировании диска в это поле записывается максимальное количество файлов и каталогов, которые могут находиться в корневом каталоге. Для каждого элемента в каталоге отводится 32 байта, поэтому корневой каталог имеет длину 32 \* rootsize байт.

Корневой каталог занимает непрерывную область фиксированного размера. Размер корневого каталога задается при форматировании и определяет максимальное количество файлов и каталогов, которые могут быть в нем описаны.

Для определения количества секторов, занимаемых корневым каталогом, можно воспользоваться следующей формулой:

RootSecs = sectsize / (32 \* rootsize)

В этой формуле sectsize - размер сектора в байтах, он может быть получен из соответствующего поля загрузочного сектора.

* 1. **Область файлов и подкаталогов.**

Вслед за корневым каталогом на логическом диске находится область файлов и подкаталогов корневого каталога.

Область данных разбита на кластеры, причем нумерация кластеров начинается с числа 2. Кластеру с номером 2 соответствуют первые секторы области данных.

Теперь мы можем привести формулу, которая позволит нам связать номер кластера с номерами секторов, занимаемых им на логическом диске:

SectNu = DataStart + ((ClustNu - 2) \* clustsize)

В этой формуле использованы следующие обозначения:

|  |  |
| --- | --- |
| SectNu | номер первого сектора, распределенного кластеру с номером ClustNu;  |
| DataStart | начало области данных, вычисляется по формуле: ressecs + (fatsize \* fatcnt) + (32 \* rootsize/ sectsize);  |
| ClustNu | номер кластера, для которого необходимо определить номер первого сектора;  |
| clustsize | количество секторов, занимаемых кластером; находится в блоке параметров BIOS.  |

* 1. **Дескрипторы файлов.**

Как мы уже говорили, любой каталог содержит 32-байтовые элементы - дескрипторы, описывающие файлы и другие каталоги. Приведем формат дескриптора:

|  |  |  |
| --- | --- | --- |
| **Смещение** | **Размер** | **Содержимое**  |
| 0 | 8 | Имя файла или каталога, выровненное на левую границу и дополненное пробелами  |
| 8 | 3 | Расширение имени файла, выровненное на левую границу и дополненное пробелами  |
| 11 | 1 | Байт атрибутов файла  |
| 12 | 10 | Зарезервировано  |
| 22 | 2 | Время создания файла или время его последней модификации  |
| 24 | 2 | Дата создания файла или дата его последней модификации  |
| 26 | 2 | Номер первого кластера, распределенного файлу  |
| 28 | 4 | Размер файла в байтах  |

В любом каталоге, кроме корневого, два первых дескриптора имеют специальное назначение. Первый дескриптор содержит в поле имени строку: ". ". Этот дескриптор указывает на содержащий его каталог. То есть каталог имеет ссылку сам на себя.

Второй специальный дескриптор содержит в поле имени строку: ".. ". Этот дескриптор указывает на каталог более высокого уровня. Если в поле номера первого занимаемого кластера для дескриптора с именем ".. " находится нулевое значение, это означает, что данный каталог содержится в корневом каталоге.

Таким образом, в древовидной структуре каталогов имеются ссылки как в прямом, так и в обратном направлении. Эти ссылки можно использовать для проверки сохранности структуры каталогов файловой системы.

**6.24 Атрибуты файлов.**

Байт атрибутов является принадлежностью каждого файла. Биты этого байта имеют следующие значения:

|  |  |
| --- | --- |
| **Бит** | **Описание** |
| 0 | Файл предназначен только для чтения.В этот файл нельзя писать и его нельзя стирать  |
| 1 | Скрытый файл.Этот файл не будет появляться в списке файлов, создаваемом командой DIR  |
| 2 | Системный файл. Этот бит обычно установлен в файлах, являющихся составной частью операционной системы  |
| 3 | Данный дескриптор описывает метку диска.Для этого дескриптора поле имени файла и поле расширения имени файла должны рассматриваться как одно поле длиной 11 байт. Это поле содержит метку диска  |
| 4 | Дескриптор описывает файл, являющийся подкаталогом данного каталога  |
| 5 | Флаг архивации.Если этот бит установлен в 1, то данный файл не был выгружен утилитой архивации  |
| 6-7 | Зарезервированы |

Обычно файлы имеют следующие атрибуты:

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| 0 | Обычные файлы (тексты программ, загрузочные модули, пакетные файлы)  |
| 7 | Только читаемые, скрытые, системные файлы. Такая комбинация битов байта атрибутов используется для файлов операционной системы io.sys , msdos.sys  |
| 8 | Метка тома. Дескриптор метки тома может находиться только в корневом каталоге логического диска  |
| 10h | Дескриптор, описывающий каталог  |
| 20h | Обычный файл, который не был выгружен программами backup.exe или xcopy.exe  |

**6.25 Дескрипторы удаленных файлов.**

При удалении файла первый байт его имени заменяется на байт E5h (символ "х"). Все кластеры, распределенные файлу, отмечаются в FAT как свободные. Если вы только что удалили файл, его еще можно восстановить, так как в дескрипторе сохранились все поля, кроме первого байта имени файла. Но если на диск записать новые файлы, то содержимое кластеров удаленного файла будет изменено и восстановление станет невозможным.

**6.26 Время создания или изменения файла.**

Остановимся подробнее на полях времени и даты создания или последней модификации файла. MS-DOS обновляет содержимое этих полей после любой операции, изменяющей содержимое файла - создания файла, перезаписи содержимого файла, добавления данных в файл или обновления содержимого файла. После обновления файла MS-DOS устанавливает бит архивации 5 байта атрибутов в 1.

Старшие пять бит содержат значение часа модификации файла, шесть бит с номерами 5 - 10 содержат значение минут модификации файла, и, наконец, в младших 5 битах хранится значение секунд, деленное на 2. Для того, чтобы время обновления файла уместилось в шестнадцати битах, пришлось пойти на снижение точности времени до двух секунд.

**6.27 Дата создания или изменения файла.**

Формат даты обновления файла напоминает формат времени. Для того чтобы получить значение года обновления файла, необходимо прибавить к величине, хранимой в старших семи битах, значение 1980. Поля месяца и дня каких-либо особенностей не имеют, они полностью соответствуют календарной дате.

**6.28 Длина файла.**

Поле длины в дескрипторе содержит точную длину файла в байтах. Для каталогов в поле длины записано нулевое значение. Вы не можете работать с каталогом средствами MS-DOS, как с обычным файлом. Единственный способ прочитать каталог как файл - использование таблицы FAT для определения цепочки занимаемых каталогом кластеров и чтение секторов, соответствующих этим кластерам при помощи прерывания INT 25h.

**Список литературы.**

1. Фролов А.В., Фролов Г.В. Библиотека системного программиста. Т. 1. Часть 1, 2, 3. Операционная система MS-DOS. М: ДИАЛОГ-МИФИ, 1991, 1993.
2. Фролов А.В., Фролов Г.В. Библиотека системного программиста. Т. 2. Аппаратное обеспечение IBM PC. Часть 1, 2. М: ДИАЛОГ-МИФИ, 1992.
3. Фролов А.В., Фролов Г.В. Библиотека системного программиста. Т. 19, MS-DOS для программиста. Часть 2, М.: ДИАЛОГ-МИФИ, 1995.