\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_@Содержание :

\_ Введение ...................................................3

\_ Физическая организация хранения информации на дискете ......3

\_ Методы\_\_ \_ кодирования информации на дискете \_\_...................\_ 4

\_ Физическая структура диска ...............................5

\_ Интерфейс адаптера НГМД ....................................7

\_ Логическая организация диска ...............................7

\_ Работа BIOS с НГМД .........................................9

\_ Некоторые форматы дисков, принятые в MS-DOS ...............10

\_ Как\_\_ \_ увеличить скорость чтения дискет ......................11

\_ О восстановлении дискет ...................................12

\_ Методы защиты от копирования ..............................12

\_ Использованная литература .................................14

\_@Введение

В настоящее время во всех вычислительных системах имеются ус-

тройства внешней памяти, использующие для накопления информации

гибкие и жесткие диски. Независимо от типа и емкости, они ис-

пользуют один и тот же принцип долговременного хранения информа-

ции в виде намагниченных участков поверхности накопителя. При

движени мимо них считывающего устройства, в нем возбуждаются им-

пульсы тока.

Сначала ( в 1981 году ) IBM PC имели один-единственный тип

внешней памяти - пятидюймовые односторонние гибкие магнитные дис-

ки двойной плотности с програмной разбивкой секторов емкостью

150K. С тех пор IBM значительно увеличила емкость дисков, был

добавлен новый стандартный размер дисков (три с половиной дюйма),

однако физическая и логическая структура диска не притерпела зна-

чительных изменений.

\_@ Физическая организация хранения информации на дискете

Гибкий диск имеет пластиковую основу с нанесенным на нее маг-

нитным покрытием. В центре находится шпиндельное отверстие , а на

некотором смещении от центра имеется одно индексное отверстие.

Назначение индексного отверстия - обеспечить накопителю точку от-

счета при счытывании или записи данных. Гибкий диск помещен в

квадратный чехол, в котором также имеются шпиндельное и индек-

сное отверстие. Находящееся на внутренней окружности магнитного

диска метализированное кольцо предназначено для усиления места

посадки диска на шпиндель электродвигателя дисковода, чтобы пре-

дотвратить нежелательную деформацию тонкого магнитного диска.Кро-

ме того, в чехле есть прорезь для контакта головки считывания/за-

писи с поверхностью диска и вырез защиты от записи.

При записи информации на магнитный диск используется потен-

циальный метод записи без возвращения к нулю. Запись по этому ме-

тоду осуществляется путем изменения направления тока записи в

магнитной головке в соответсвии с обрабатываемыми данными. Изме-

нение направления тока записи вызывает перемену магнитного потока

в магнитной головке, что приводит к изменению намагниченности

учаска носителя информаци, проходящего в это время под головкой.

В зависимости от направления вектора намагниченности рабочего

слоя магнитного носителя по отношению к направлению вектора ско-

рости перемещения носителя, различают продольную, поперечную и

перпендикулярную намагниченность. В накопителях IBM формата

( имеются в виду накопители 34 и 3740 фирмы IBM ) используется

только продольная намагниченность, поэтому в дальнейшем речь пой-

дет именно о ней.

Процесс записи может быть представлен в виде следующих переда-

точных звеньев:

\_\_i(t) -> F(t) -> H(x,y,z,t) -> M(l,y,z)

Процесс воспроизведения:

\_\_M(l,y,z) -> Ф(t) -> e(t)

Здесь

\_\_i(t)\_\_ - ток в обмотке записи магнитной головки;

\_\_F(t)\_\_ - магнитодвижущая сила магнитной головки записи;

\_\_H(x,y,z,t)\_\_ - поле записи;

\_\_M(l,y,z)\_\_ - остаточная намагниченность после воздействия поля

записи;

\_\_Ф(t)\_\_ - магнитный поток в сердечнике магнитной головки вос-

произведения;

\_\_e(t)\_\_ - электродвижущая сила, наводимая в обмотке магнитной го-

ловки воспроизведения;

\_\_x,y,z\_\_ - пространственные координаты, связанные с головкой;

\_\_l=vt\_\_ - координата, связанная с носителем записи;

\_\_v\_\_ - скорость записи.

Во время считывания на выходной обмотке головки воспроизведе-

ния наводятся разнополярные сигналы в те моменты времени, когда

под головкой проходят участки поверхности с изменением направле-

ния намагниченности. Эти сигналы воспринимаются усилителем-форми-

рователем, который преобразует их в выходные униполярные им-

пульсы считывания "единиц". Записи "нуля" соответствует отсут-

ствие импульсов в некоторые определенные моменты времени.

\_@ Методы кодирования информации на дискете

Для записи информации на магнитные носители применяют спе-

циально разработанные модуляционные коды записи. Данные коды раз-

рабатываются специалистами и должны обладать высокой информатив-

ностью и способностью с самосинхронизации. Под информативностью

способа записи понимают количество записанной информации, прихо-

дящийся на один период намагниченности.

В накопителях 3740 ( IBM ) используется метод частотной моду-

ляции, а в накопителях 34 - метод модифицированной частотной мо-

дуляции.

\_"Метод частотной модуляции

Начало каждого элемента отмечается тактовым импульсом в виде

смены направления намагниченности. Если элемент должен представ-

лять 1, то в его центральной части записывается еще один такто-

вый импульс ( что бы создать изменение магнитного потока ), а ес-

ли 0, то смены напрвления намагниченности не происходит вплоть до

начала следующего элемента. Таким образм, если тактовая частота

равна F, то поток двоичных единиц дает частоту 2F.

┌┐ ┌┐ ┌┐ ┌┐ ┌┐ ┌┐ ┌┐ ┌┐

─┘└──┘└──┘└──┘└──┘└──┘└──┘└──┘└─── \_\_тактовые импульсы

┌┐ ┌┐ ┌┐ ┌┐ ┌┐

───┘└──────────┘└──────┘└──┘└──┘└─ \_\_данные

\_\_ 1 0 0 1 0 1 1 1

┌┐┌┐┌┐ ┌┐ ┌┐┌┐┌┐ ┌┐┌┐┌┐┌┐┌┐┌┐

─┘└┘└┘└──┘└──┘└┘└┘└──┘└┘└┘└┘└┘└┘└── \_\_сигналы записи

\_"Метод модифицированной частотной модуляции

В этом методы 1 всегда представляется переходом намагниченности

в центре элемента. Переход вводиться в начале элемента, если это

0, а за ним НЕ следует 1. При том же разнесении переходов этот

метод позволяет записывать на единицу длины в два раза больше

символов, чем метод частотной модуляции.

┌┐┌┐┌┐┌┐┌┐┌┐┌┐┌┐┌┐

┘└┘└┘└┘└┘└┘└┘└┘└┘└─ \_\_тактовые импульсы

┌┐ ┌┐ ┌┐┌┐┌┐

─┘└────┘└──┘└┘└┘└── \_\_данные

\_\_1 0 0 1 0 1 1 1

┌┐ ┌┐ ┌┐ ┌┐┌┐┌┐

─┘└─┘└─┘└──┘└┘└┘└── \_\_сигналы записи

При записи информации по методу модифицированной частотной моду-

ляции возникает так называемое смещение синхронизации. Это возни-

кает потому, что в общем случае при считывании информации с дис-

кеты невозможно отличить тактовые сигналы от сигналов данных.

Поэтому в зависимости от точки отчета одна и таже последова-

тельность импульсов может трактоваться по-разному. Для устране-

ния этой в высшей мере неприятной неоднозначности на каждой до-

рожке вводят специальные поля, заполненные нулями, размером каж-

дого поля 12 байт. При считывании информации контроллер НГМД

знает, что в них находятся нули, поэтому трактует поступающие

сигналы как тактовые импульсы, одновременно соответствующим об-

раз подстраивая схему сепаратора данных.

Помимо рассмотренных выше методов частотной и модифицированной

частотной модуляции используется \_ кодирование с ограниченным рас-

\_ стоянием между периодами намагниченности\_\_ ( RLL - кодирование ).

По сравнению с методом модифицированной частотной модуляции об'ем

хранимой на диске информации увеличивается на 50%. Метод RLL ос-

нован на записи с групповым кодированием. В этом методе каждый

байт поступающих данных разбивается на две тетрады, а затем тет-

рада шифруется специальным 5-ти разрадным кодом, характерным тем,

что каждое число в нем содержит, по крайней мере, одну перемену

направлении потока. При считывании две 5-ти разрядные тетрады

снова сливаются в байты.

\_@ Физическая структура диска

Емкость диска зависит от характеристики дисковода и особеннос-

тей операционной системы; однако структура диска, в сущноcти,

всегда одна и та же. Данные всегда записываются на магнитной по-

верхности в виде концентрических окружностей, называемых дорож-

ками.Каждая дорожка, в свою очередь, состоит из нескольких секто-

ров, количество которых определяется при операции форматирования.

Сектор является единицей хранения информации на дискете.Количес-

тво информации на диске, таким образом, зависит от числа дорожек

( от плотности записи ) и общего размера секторов на каждой до-

рожке. Старые модели дисководов работали с 40 дорожками, нынешние

модели - с 80, большинство современных дисководов позволяют фор-

матировать дискеты плотностью до 85 дорожек.

Для стандартных дискет IBM расположение каждой дорожки не мо-

жет быть изменен, потому что это прежде всего зависит не от ОС и

не от дискеты, а от конструктивных особенностей дисковода.Однако,

число, размер и расположение секторов задаются программно при

первоначальной разметке ( форматировании ) дискеты. Разметка осу-

ществляется либо ОС, либо используются функции BIOS. Хотя MS-DOS

поддерживает размеры сектора дискет 128, 256, 512 и 1024 байта,

однако используется сектор размером 512 байт и, по-видимому, это

в ближайшее время не изменится ( если и измениться, то только в

сторону увеличения ).

Структура формата дорожки зависит от типа контроллера, но, как

правило, включает в себя байты синхронизации, указывающие на на-

чало каждого сектора, идентификационные заголовки, cостоящие из

номера цилиндра, головки, сектора и размера сектора, и поля, хра-

нящего байты циклического контроля, предназначеные для обнаруже-

ния ошибок при считывании данных и служебной информации. На сле-

дующем рисунке представлен формат дорожки для стандарта IMB 34.

индекс

───┐ ┌───────────────────────────────────────────────────────────

│ │

└─┘

──┬────────┬───────┬──────┬──────────┬──∙∙──┬──────────┬─────────

│ gap4a │ index │ gap1 │ sector 1 │ │ sector n │ gap4b

──┴────────┼───────┼──────┼──────────┼──∙∙──┴──────────┴─────────

┌────┘ └───┐ │ │

├───────┬────────┤ │ │

│ sync │ IAM │ │ │

└───────┴────────┘ │ │

┌──────┘ └───┐

├────────┬────────────┤

│ header │ data field │

├────────┴────────────┤

┌────────────────┘ └──────────────────────┐

├─────┬─────┬─┬─┬─┬─┬────┬──────┬─────┬─────┬──────┬────┬─────┤

│ sync│ SAM │c│h│r│n│ crc│ gap2 │ sync│ DAM │ data │ crc│ gap3│

└─────┴─────┴─┴─┴─┴─┴────┴──────┴─────┴─────┴──────┴────┴─────┘

На приведенной выше схеме :

┌────────┬──────────────────────────────────┬───────────┬──────┐

│ Обозн. │ Назначение поля │ Содержимое│ Длина│

├────────┼──────────────────────────────────┼───────────┼──────┤

│ GAP4A │ Предындексный зазор дорожки │ 4E │ 50 │

│ SYNC │ Поле синхронизации │ 00 │ 0C │

│ IAM │ Адресный маркер начала дорожки─┬─┼── C2\* │ 3 │

│ │ └─┼── FC │ 1 │

│ SAM │ Маркер начала сектора──────────┬─┼── A1 │ 3 │

│ │ └─┼── FE │ 1 │

│ C │ Номер цилиндра │ -- │ 1 │

│ Н │ Номер головки │ -- │ 1 │

│ R │ Номер сектора │ -- │ 1 │

│ N │ Код размера сектора │ -- │ 1 │

│ CRC │ Контрольный код │ -- │ 2 │

│ GAP2 │ Зазор заголовка сектора │ 4E │ 16 │

│ DAM │ Маркер начала данных───────────┬─┼── A1 │ 3 │

│ │ └─┼── FB │ 1 │

│ GAP3 │ Зазоp области данных │ 4E │ 50 │

│ data │ Данные │ -- │ \*\* │

│ GAP4B │ Зазор дорожки │ 4E │ \*\*\* │

└────────┴──────────────────────────────────┴───────────┴──────┘

\* - \_^^&Данные поля записываются со специально нарушенными битами

\_^^&синхронизации.

\*\* - \_^^&Длина данных определяется по формуле 128\*2^N, где N - код

\_^^&длины из заголовка сектора: от 0 (128б) до 7 (16384б).

\*\*\* - \_^^&Длина определяется оставшимся расстоянием до индексного от-

\_^^&верстия и зависит только от скорости вращения дисковода в

\_^^&момент форматирования.

Поля GAP1..GAP4 служат прежде всего для организации задержки

при выдачи порций данных с дискеты, а также для компенсации раз-

бросов физической длины различных полей, возникающих из-за несо-

вершенства механизма дисковода ( конкретнее, из-за нестабильнос-

ти вращения ). Маркеры служат для выделения определенных облас-

тей на диске: идентификатора дорожки, заголовка сектора или об-

ласти данных. Для того что бы маркеры можно было отличить от дан-

нных, их записывают со специально нарушенным кодом синхронизации.

Четвертый байт маркера обозначает тип выделяемой им области. Кон-

кретно в маркере области данных значение fb соответствует обыч-

ным данным, а f8 - удаленным.

Целостность информации в областях данных контролируется с по-

мощью циклического контрольного кода, контрольные числа которого

записываются после определенных областей. При считывании с диске-

ты контроллер самостоятельно высчитывает контрольную сумму, а за-

тем сравнивает ее со считанной с диска. Эта контрольная сумма,

называемая \_ кодом циклического контороля \_\_( CRC - Cyrcle Redundency

Contol ), подсчитывается с помощью полинома следующего вида :

X\_\_16\_\_ + X\_\_12\_\_ + X\_\_5\_\_ + X + 1

В случае несовпадении этих двух чисел выставляется флаг ошибки.

\_@ Интерфейс адаптера НГМД

Функции сигналов адаптера НГМД следующие:

┌─────────────┬─────────────────────────────────────────────────┐

│Наименование │ Выполняемая функция │

├─────────────┼─────────────────────────────────────────────────┤

│High/Normal │ Используется для переключения схем записи/чтения│

│ │ для работы с нормальной и высокой плотностью │

│Index │ Перепад напряжения с высокого уровня на низкий │

│ │ при обнаружение индексного отверстия │

│Select │ Выбор накопителя ( присутствует постоянно ) │

│Mot │ Включение мотора дисковода │

│Direction │ Направление перемешения головок при сигнале Step│

│ │ низкий уровень - к центру дискеты; │

│ │ высокий уровень - к краю дискеты; │

│Step │ Перемещеие головок на одну дорожку │

│Write data │ Запись информации, каждый импульс вызывает │

│ │ изменение магнитного потока на обратный │

│Write gate │ Разрешение записи информации │

│Track 0 │ Перемещене головок к началу дискеты │

│Write │ Запрет записи на дискеты ( сигнал появляется при│

│ protected │ присутствии дискеты с защитной наклейкой ) │

│Read data │ Перепад напряжения с высокого уровня на низкий │

│ │ соответствуют обнаружению на дискете перехода │

│ │ намагниченности. │

│Select head │ Выбор головки для чтения/записи │

│ │ низкий уровень - головка 1; │

│ │ высокий уровень - головка 0; │

│Ready │ Дискета готова к работе │

│Disk change │ Вставлена новая дискета │

└─────────────┴─────────────────────────────────────────────────┘

Полярность всех сигналов интерфейса (кроме тех, где указано иное)

отрицательная. Сигналы Disk change и High/Normal в IBM PC/XT не

используются.

\_@ Логическая организация диска

Первая операция, которую необходимо выполнить перед тем, как

дискета будет готова к использованию - это форматирование. Этот

процесс позволяет придать диску его окончательную структуру. В

ходе форматирования определяется количество дорожек и число сек-

торов на дорожке.

MS-DOS предусматривает четыре логических области дискеты:

- \_ загрузочный сектор ( boot record )

- \_ таблица размещения файлов ( file allocation table )

- \_ корневой каталог

- \_ область данных

\_"Загрузочный сектор

Содержит короткую ( менее 512 байт ) программу начальной загруз-

ки ОС в память компьютера. Независимо от типа ОС и способа форма-

тирования дискеты, эта программа \_ всегда\_\_ занимает самый первый

сектор на самом первой дорожке диска. Следует различать \_ Boot

\_ record \_\_и \_ Master\_\_ \_ Boot record. \_\_Первый находится на дискеты в слу-

чае если эта дискета не системная. Второй же находится исключи-

тельно на системных дисках. Также этот сектор содержит всю важ-

ную информацию о характеристиках диска. Структура этой информации

следующая :

┌────────┬─────────────────┬────┬─────────────────────────────────┐

│Cмещение│ Название │байт│ Описание │

├────────┼─────────────────┼────┼─────────────────────────────────┤

│ 00 │ Команда JMP │ 3 │ Длинный или короткий переход │

│ │ │ │ на программу загрузчика │

│ 03 │ Идентификатор │ 8 │ Идентификатор версии DOS, │

│ │ │ │ с помощью которой отформа- │

│ │ │ │ тирован диск │

│ 0B │ Байт/Сектор │ 2 │ Размер сектора в байтах │

│ 0D │ Сектор/Кластер │ 1 │ Число секторов в кластере │

│ 0E (\*)│ Резерв │ 2 │ Количество резервных секторов │

│ 10 │ Число FAT │ 1 │ Число копий FAT на диске │

│ 11 │ Число корневых │ 2 │ Max количество записей в │

│ │ записей │ │ корневом каталоге диска │

│ 13(\*\*)│ Всего секторов │ 2 │ Общее число секторов на диске │

│ 15 │ Носитель │ 1 │ Тип магнитного носителя │

│ 16 │ Сектор/FAT │ 2 │ Число секторов в каждой из FAT │

│ 18 │ Сектор/дорожка │ 2 │ Число секторов на одну дорожку │

│ 1A │ Число головок │ 2 │ Число головок на диске │

│ 1С │ Скрытых секторов│ 4 │ Число скрытых секторов на диске│

│ 20 │ Боьших секторов │ 4 │ Общее число секторов │

│ 22 │ Номер накопителя│ 1 │ Номер накопителя по BIOS'у │

│ 23 │ │ 1 │ Резерв │

│ 24 │ Сигнатура │ 1 │ Содержит число 29 │

│ 25 │ ID тома │ 4 │ Идентификатор тома диска │

│ 29 │ Метка тома │ 11 │ Копия метки тома │

│ 34 │ ID типа FAT │ 8 │ FAT12 или FAT16 -байт │

└────────┴─────────────────┴────┴─────────────────────────────────┘

(\*) - \_^^&для формата IBM PC длина зарезервированной области всегда

\_\_составляет 1 сектор ( сектор загрузчика )

(\*\*)- \_^^&если более 65535 то 0

\_"Таблица размещения файлов ( FAT )

Содержит информацию о месторасположении записанных на дискету

файлов. Системa MS-DOS выделяет для хранения файла, в зависимос-

ти от его длины, один или более кластеров ( кластер - единица

хранения данных на диске, обычно один кластер равен нескольким

секторам ), однако MS-DOS не заботится, чтобы запись файла проис-

ходила последовательно ( скорее наоборот : логика работы MS-DOS

такова, что она всячески способствует фрагментации файлов ), поэ-

тому необходимо хранить информацию, по каким именно кластерам

раскидан данный файл. В силу особой важности этой информации FAT

существует на диске в двух копиях. FAT дискеты состоит из 12-би-

товых элементов. Структура таблицы размещения файлов - следующая:

┌─────────────┬───────────────────────────────────────────┐

│ Элемент FAT │ Выполняемая функция │

├─────────────┼───────────────────────────────────────────┤

│ 0 │ Идентификатор формата │

│ 1 │ Зарезервирован ( должен быть FFF ) │

│ 2 │ Статус кластера 2 │

│ 3 │ Статус кластера 3 │

│ ... │ ... │

└─────────────┴───────────────────────────────────────────┘

Идентификатор формата может принимать

следующие значения :

F0 - 3'5/2/18 ( дискета 3'5, 2 стороны, 18 секторов )

F8 - жесткий диск

F9 - 5'25/2/15 или 3'5/2/9

FC - 5'25/1/9

FD - 5'25/2/9

FE - 5'25/1/8

FF - 5'25/2/8

Статус кластера может быть следующим :

000 - кластер свободен

FF0..FF6 - зарезервирован

FF7 - испорчен

FF8...FFF - последний кластер

иначе - номер следующего кластера файла

Интересно заметить, что, по-видимому, значение статуса 001 яв-

ляется запрещенной комбинацей.

\_"Корневой каталог

Следует непосредственно за FAT. Содержит информацию об основ-

ных параметрах файлов ( длине, дате создания,...). Каждая запись

в каталоге, соответствуeт тому или иному файлу ( кроме, разумеет-

ся, метки диска ), включает номер первого кластера, назначенного

MS-DOS этому файлу, который используется как точка входа в FAT.

Начиная с этой точки, каждая запись в FAT содержит номер следую-

щего кластера, распеределенного под этот файл, или метку послед-

него кластера. Структура записи в корневом каталоге следующая :

┌─────────┬───────────┬───────────────────────────────────┐

│Cмещение │ Название │ Описание │

├─────────┼───────────┼───────────────────────────────────┤

│ 00 │ Имя │ Имя файла │

│ │ │ 0 если запись свободна │

│ │ │ E5 если файл удален │

│ │ │ 2E запись родительского каталога │

│ 08 │ Расширение│ │

│ 0B │ Атрибуты │ Системный атрибут файла │

│ 0C │ │ Резерв │

│ 16 │ Время │ Время последней перезаписи файла │

│ 18 │ Дата │ Дата последней перезаписи файла │

│ 1A │ Кластер │ Номер первого отведенного кластера│

│ 1C │ Размер │ Размер файла ( 4 байта ) │

└─────────┴───────────┴───────────────────────────────────┘

\_"Область данных

Именно то место, ради которого и используется дискета - здесь

храниться информация пользователя. MS-DOS рассматривает эту об-

ласть как совокупность кластеров, каждый из которых содержит один

или несколько секторов. Из-за того что первые два поля FAT заре-

зервированы, первому кластеру в области данных присвоен номер 2.

Все каталоги, кроме корневого, также раасматриваются MS-DOS как

файлы особого вида, и поэтому помещаются в область данных.

\_@Работа BIOS с НГМД

Програмное управление дискетой ( точнее говоря, адаптером НГМД)

осуществляется с помощью драйвера BIOS, вызов которого осущес-

твляется через прерывание int 13. Методика вызова конкретных фун-

кций стандартна, то есть номер функции загружается в ah, ос-

тальные параметры в другие регистры общего назначения, для адре-

совки буферов так же используется регистровая пара es:bx.

Всего стандартный драйвер поддерживает 6 функций работы с НГМД

с номерами от 0 до 5. Перечислим их в порядке возрастания :

\_ 0\_\_ \_ -\_\_ \_ Сброс системы НГМД \_\_ \_\_ \_ 3\_\_ \_ -\_\_ \_ Записать сектор

\_ 1\_\_ \_ -\_\_ \_ Прочитать состояние \_\_ \_\_ \_ 4\_\_ \_ -\_\_ \_ Проверить сектор

\_ 2\_\_ \_ -\_\_ \_ Прочитать сектор \_\_ \_\_ \_ 5\_\_ \_ -\_\_ \_ Разметка дорожки

Все функции выполняются, согласовываясь с базовой дисковой

таблицей, на которую указывает вектор 1e. Разумеется пользова-

тель может модифицировать этот вектор и создать свою таблицу. При

загрузке ОС BIOS инициализирует ее, а DOS модифицирует, чтобы

улучшить производительность дискет. Структура этой таблицы сле-

дующая ( везде, где не сказано иное, время указывается в квантах

системного таймера ).

┌────┬───────────────────────────────────────────────────────────┐

│Смещ│ Содержимое │

├────┼───────────────────────────────────────────────────────────┤

│ 0 │ биты 4-7: время перехода с дорожки на дорожку ( в млс ) │

│ │ биты 0-3: время под'ема головки после чтения/записи │

│ 1 │ бит 0: 1= исп.DMA , 0= DMA не используется │

│ │ биты 2-7: время опускания головок │

│ 2 │ время не остановки двигателя после окончания чтения/записи│

│ 3 │ размер сектора (0=128, 1=256, 2=512, 3=1024) │

│ 4 │ номер последнего сектора на дорожке │

│ 5 │ длина межсекторного промежутка для операций чтения/записи │

│ 6 │ Максимальная длина передаваемых данных │

│ 7 │ длина межсекторного промежутка для операции форматирования│

│ 8 │ символ-заполнитель для форматирования ( обычно f6 ) │

│ 9 │ время позиционирования головок │

│ a │ время разгона мотора (в 1/8-секундных интервалах) │

└────┴───────────────────────────────────────────────────────────┘

Также в BIOS Data Segment содержатся следующие параметры НГМД:

\_ 0:043e\_\_ требуется рекалибрация флоппи-дисковода

\_ \_\_ (бит 0=A, бит 1=B, etc.)

\_ 0:043f\_\_ мотор дискеты включен (бит 0=drive A, бит 1=B, etc.)

\_ 0:0440\_\_ время до выкл. мотора. Если 0, то мотор выключается

\_ 0:0441\_\_ код ошибки дискеты (то же, что возвращается по int 13 )

\_ 0:0442\_\_ область информации состояния контроллера дискет (7байт)

\_@ Некоторые форматы дисков, принятые в MS-DOS

Начиная с MS-DOS v2.0 возможна запись/чтение практически лю-

бых физических форматов дискет. Эта возможность осуществляется

использованием механизмом загружаемых драйверов устройств. Появ-

ление в последнее время расширенных версий BIOS'a практически уп-

разнило понятие "стандартный формат", теперь стандартным можно

считать практически любой формат, который сответствует специфика-

ции MS-DOS.

Вообще, появление новых форматов тесно связано с историей раз-

вития DOS. Первоначальная версия MS-DOS v1.0 поддерживала только

формат, обозначенный ниже как (1.0) следующая версия 1.1 добави-

ла (1.1), а версия 2.0 - (2.0). Короче говоря, почти каждая вер-

сия DOS приносила что-то новое. Все, что из этого вышло, пред-

ставлено ниже.

Возможные варианты форматов в зависимости от типа дисковода:

\_ 360 Кб дисковод

(1) 160 Kb 40 дорожек 8 секторов 1 сторона

180 Kb 40 дорожек 9 секторов 1 сторона

200 Kb 40 дорожек 10 секторов 1 сторона

(1.1) 320 Kb 40 дорожек 8 секторов

(2.0) 360 Kb 40 дорожек 9 секторов

400 Kb 40 дорожек 10 секторов

420 Kb 42 дорожки 10 секторов

430 Kb 43 дорожки 10 секторов

\_ 720 Кб дисковод (3'5" и 5'25")

720 Kb 80 дорожек 9 секторов

747 Kb 83 дорожки 9 секторов

800 Kb 80 дорожек 10 секторов

820 Kb 82 дорожки 10 секторов

830 Kb 83 дорожки 10 секторов

\_ 1.2 Мб дисковод

180 Kb 40 дорожек 9 секторов 1 сторона

200 Kb 40 дорожек 10 секторов 1 сторона

320 Kb 40 дорожек 8 секторов

360 Kb 40 дорожек 9 секторов

400 Kb 40 дорожек 10 секторов

420 Kb 42 дорожки 10 секторов

430 Kb 43 дорожки 10 секторов

720 Kb 80 дорожек 9 секторов

747 Kb 83 дорожки 9 секторов

800 Kb 80 дорожек 10 секторов

820 Kb 82 дорожки 10 секторов

830 Kb 83 дорожки 10 секторов

1.2 Mb 80 дорожек 15 секторов

1.36 Mb 80 дорожек 17 секторов

1.39 Mb 82 дорожки 17 секторов

1.41 Mb 83 дорожки 17 секторов

1.44 Mb 80 дорожек 18 секторов

1.47 Mb 82 дорожки 18 секторов

1.49 Mb 83 дорожки 18 секторов

\_ 1.44 Мб дисковод

720 Kb 80 дорожек 9 секторов

800 Kb 80 дорожек 10 секторов

820 Kb 82 дорожки 10 секторов

830 Kb 83 дорожки 10 секторов

1.44 Mb 80 дорожек 18 секторов

1.52 Mb 80 дорожек 19 секторов

1.60 Mb 80 дорожек 20 секторов

1.64 Mb 82 дорожки 20 секторов

1.68 Mb 83 дорожки 20 секторов

1.68 Mb 80 дорожек 21 сектор

1.72 Mb 82 дорожки 21 сектор

1.74 Mb 83 дорожки 21 сектор

\_ Скорости передачи данных :

Дисковод

360-720 5'25" - 300 Кбит

720 3'5" - 250 Кбит

1.2 5'25" - 300 Кбит (DD) 500 Кбит (HD)

1.44 3'5" - 250 Кбит (DD) 500 Кбит (HD)

2.88 3'5" - ??? (вероятно 1000 Кбит)

Дисковод 1.44 Мб вообще интересен тем, что при той же скорости

передачи данных обеспечивает гораздо более высокую плотность за-

писи, чем дисковод 1.2 Мб. По этой причине при форматировании на

720-800 Кб скорость передачи ниже.

\_` \_@Как увеличить скорость чтения дискет

Оказывается возможно форматировать диски так, что скорость об-

ращения к дискете увеличивается в полтора раза (а в ряде случаев

и больше). Суть состоит в следующем: когда дисковод перемещает

головку с дорожки на дорожку после чтения/записи при обычном рас-

положении секторов, первый сектор успевает "ускользнуть" от го-

ловки и приходится ждать еще целый оборот диска, чтобы прочитать

его. Замечено, что если на каждой последующей дорожке "сдвинуть"

первый сектор на три сектора, то при перемещении головки он счи-

тывается сразу - что и является причиной увеличения производи-

тельности.

\_@О восстановлении дискет

\_@ \_\_Продолжитиельность жизни гибких дисков обычно около трех лет.

Хотя чисто теоретически правильно эксплуатируемый диск выдержи-

вает 70 миллионов проходов по одной дорожке, что составляет бо-

лее 20 лет непрерывной работы. Однако, все это относится к

идеальным условиям эксплуатации, но где вы их видели ?! Дискеты

зачастую лежат без конвертов на пыльной поверхности, их сгибают,

на них пишут, их обкуривают "Беломором", наконец. Кто же это вы-

держит ? И в итоге даже самые хорошие импортные диски начинают

сыпаться.

При обнаружении повреждения 0 дорожки на дискете ни одна прог-

рамма не форматирует такую дискету. На западе такие дискеты, воз-

можно, просто выбрасывают. Для нас такой подход не приемлем. Дис-

кеты стоят достаточно дорого и выбрасывать деньги на ветер не в

моих правилах.

Один из способов получения работоспособных дискет был предло-

жен Панковым (автором PU\_1700): нулевая дорожка перемещалась в

середину дискеты. Достаточно оригинальный способ, однако имеет

недостатки:

1) Необходимость постоянно держать в памяти PU\_1700

2) Невозможно прочитать обычную дискету без переустановки

PU\_1700 - что крайне неудобно на машине с 1 дисководом

Достоинством является то, что этот способ работает с любым

форматом даже при полном отсутствии 0 дорожки дискеты.

Однако, возможен и другой метод.

Обычно на 0 дорожке, а равно как и на других дорожках

дискеты, в силу различных причин (в основном механическое

повреждение), пропадает читабельность одного или двух сектора.

Дискета форматируется так, что поврежденная часть поверхности

просто не используется. У метода есть недостаток: невозможно

восстановить более одного сектора на дорожке (360-720 Кб) или

двух (1.2-1.44 Мб), однако выбирайте: дискета с поврежденной 0

дорожкой на 800 Кб или абсолютно нормальная на 720 Кб ?

\_@ Методы защиты от копирования

По сути дела, проблема защиты от копирования - это прежде все-

го проблема идентификации дистрибутивного носителя. Поэтому зна-

ние некоторых особенностей организации хранения информации на

дискете делает возможным указание некоторых методов идентифика-

ции, которые программист может использовать для защиты своего ПО.

Изложим их в порядке возрастания сложности.

\_"Использование собственного формата

Возможны три варианта использования этого метода :

1) Часть дорожек на дискете, кроме тех, где размещаются систем-

ные области ОС, форматируется нестандартным способом. Доста-

точно написать собственный драйвер работы с этим нестандар-

тным форматом и сделать так, что бы он заменял стандартный

обработчик int 13 после загрузки с данной дискеты и дискета

становиться нечитаемой с помощью ОС. И, следовательно, скопи-

ровать ее стандартным образом также невозможно.

2) Возможно также отформатировать только одну дорожку на диске-

те и разместить там некоторую ключевую информацию, а затем

после запуска программы, проверять наличие этой информации.

3) Как вариант возможно вообще не форматировать одну из дорожек

где-нибудь в середине дискеты,за которой располагается неко-

торая информация ( записанная в стандартном формате ). При

копировании стандартной утилитой ОС эта дискета также не бу-

дет скопирована полностью.

\_"Введение дополнительных секторов

Открытое использование собственного формата равноценно установ-

ке железной двери в ранее неприметном доме - видно, что его хо-

зяевам есть что прятать и видно, где это спрятано. Но можно впол-

не успешно имитировать стандартный формат, с вынесением меток за

стандартные поля копирования. Самый очевидный способ - введение

дополнительного сектора на дорожке, в котором храниться ключевая

информация. В данном случае сам факт наличия этого сектора яв-

ляется достаточным основанем, что бы считать диск дистрибутивным.

\_"Использование дополнительных (инженерных) дорожек

На любой дискете, отформатированной стандартным образом за пос-

ледней дорожкой \_ всегда \_\_остается некоторое пространство, которое

можно использовать для хранения ключевой информации. Разумеется

никто не мешает отформатировать эти дорожки нестандартным образом.

\_"Использование промежуточных цилиндров

Широко известен тот факт, что при форматировании дискеты на 360К

на дисководе на 1.2М головки дисковода перемещаются не на одну, а

на две дорожки, поэтому нечетные дорожки остаются не использован-

ными. На этих дорожках вполне можно разместить весь код програм-

мы, оставив "видимым" только небольшой загрузчик.

\_"Нестандартное чередование секторов

Система MS-DOS ориентирована исключительно на стандартные форма-

ты, которые характеризуются, в частности, строго последова-

тельным возрастанием номеров секторов на дорожке. Поэтому если

изменить порядок следования секторов, то при создании копии DOS

изменит их номера на "правильный порядок". Проверку же легко ор-

ганизовать, замеряя временные интервалы между чтениями секторов с

определенными номерами.

\_"Создание псевдосбойных секторов

Когда MS-DOS встречает сбойный сектор ( а с точки зрения MS-DOS,

сбойным является сектор с неправильной контрольной суммой ), то

она игнорирует его содержимое и просто не копирует его, таким об-

разом данные, которые находились в сбойном секторе не попадут на

копию, хотя сигнал несовпадения контрольных сумм вовсе не запре-

щает доступа к данным, а лишь предупреждает об ошибке. Следова-

тельно, достаточно записать ключевую информацию в сектор, заста-

вить MS-DOS считать его сбойным, а затем при запуске программы

проводить чтение этого сектора, игнорируя сообщение об ошибке и

проверять его содержимое. Создать же псевдосбойный сектор не

очень сложно, для этого необходимо провести операцию сброса кон-

троллера НГМД после после того, как на дискету записано необходи-

мое количество данных.

\_"Разрушение поверхности дискеты

В отличие от предыдушего метода предлагается создавать в задан-

ном количестве секторов "самые настоящие" сбойные сектора, напри-

мер, путем протыкания поверхности дискеты иглой или лазером. При

запуске программа проверяет наличие сбойных секторов на диске не

просто попыткой чтения, а попыткой записи в них какой-либо инфор-

мации, что бы не оказаться жертвой предыдушего способа.

\_"Нестандартная плотность записи

Плотность записи зависит от скорости приема-выдачи информации

контроллером и скоростью вращения дискеты. Скорость вращения дис-

ковода на стандартном IBM PC изменить невозможно, но если немно-

го изменить электронную схему дисковода, то теоретически вполне

возможно создать экземпляр, который бы мог записывать дискеты с

нестандартной плотностью информации, записанной на каждой дорожке.

Вполне возможно также изменять метод записи информации, ис-

пользуя попеременно частотную и модиф. частотную модуляцию.

\_"Измерение межсекторных промежутков

Размер поля GAP3 определяется при операции форматирования и мо-

жет изменяться в значительных пределах без изменения количества и

размеров сектора на дорожке. Вычисления можно производить на ос-

нове измерения интервалов между последовательно выполняемыми ко-

мандами контроллера НГМД "Чтение идентификатора сектора". Однако,

так как эти результаты в большой степени будут искажаться неста-

бильным вращением дискеты, то очень сложно будет получить одноз-

начно трактуемые результаты.

\_@ Использованная литература

\_@ \_\_1. P.Norton "Programmer's guide to the IBM PC"

- Microsoft Press 1985

2. С.Х.Гореликов "IBM PC.Дисковая подсистема: контроллеры, нако-

пители и их обслуживание" - М, Звезды и С, 1992

3. Л.В.Букчин, Ю.Л.Безрукий "Дисковая подсистема IBM-совмести-

мых персональных компьютеров" - М, Press-Media, 1993

4. TECH Help!, Flambeaux Software, Dan Rollins

5. Описание FFORMAT v2.97

6. Р.Данкан "Профессиональная работа в MS-DOS"

- М, Мир, 1993

7. Толковый словарь по вычислительным системам / под редакцией

В.Иллингуорта и др. - М, Машиностроение, 1989

8. А.Щербаков "Защита от копирования", М, Эдель, 1992

\_\_(c) Copyright by (cs) BREDcorp. 1995 v1.3

\_\_ (c) Used text editor Word&Deed v7.0 by A.Gutnikov

\_\_ (c) Printed by Epson(tm) LQ-100 style 'Presti\_\_ge'

\_\_ (c) Corrected by (Dreago)

\_\_\_\_\_\_\_\_2\_

\_;\_

\_\_\_\_\_\_\_\_\_\_\_\_\_A\_\_\_\_\*.FRM\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\*.MAC\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_E\_\_\_А═╠╠╠L\_\_                                        \_