

# Файловая система FAT

## Физическая структура магнитного диска

Магнитный диск представляет собой кольцо из алюминиевого сплава (*жесткий* диск) или из полимерного материала (*гибкий* диск), на поверхность которого нанесено магнитное покрытие (рисунок 1). За внутреннее отверстие магнитный диск крепится на устройстве вращения.

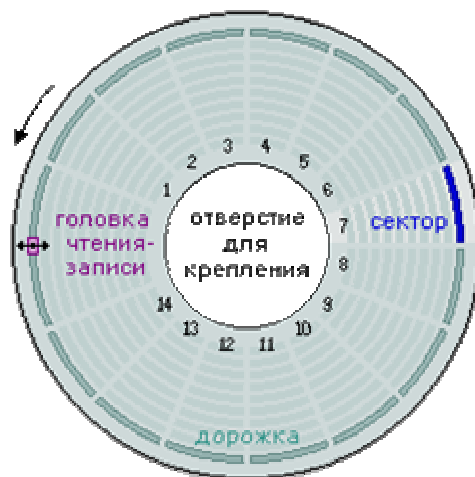


Рисунок 1 — Физическая структура магнитного диска

Запись и считывание информации осуществляет магнитная **головка** (*Head*), которая может занимать одно из фиксированных положений на радиусе диска или на кривой, близкой к радиусу. Ширина хода головки составляет примерно 1 дюйм (2,54 см) для диска диаметром 3,5 дюйма. При вращении диска и фиксированном положении головки на поверхности диска остается *воображаемый* след движения головки, называемый **дорожкой** (*Track*).

Так как головка может занимать множество фиксированных положений, на диске существует множество дорожек, образующих концентрические круги небольшой ширины. Так, для гибкого диска 3,5 дюйма число дорожек составляет 80-83, для жесткого диска число дорожек имеет порядок 1000. Каждой дорожке присвоен порядковый номер, начиная от **нулевой** дорожки, расположенной у внешнего края диска. Для дискеты 3,5 дюйма, например, рабочие дорожки имеют номера от 0 до 79.

При относительном перемещении головки и диска вследствие намагниченности поверхности диска в головке индуцируется электрический ток, который может быть усилен и использован для считывания информации с дорожки. Наоборот, если по катушке головки пропустить электрический ток, то на поверхности диска, на дорожке, поверхность диска приобретает некоторую намагниченность, и записываемая информация запоминается.

За время одного полного оборота диска с дорожки может быть считан (или записан на нее) поток информационных элементов (битов), который можно использовать непосредственно для считывания/записи информации, однако вся длина дорожки делится на равные части (за исключением последней, которая чуть длиннее), называемые **секторами** (*Sector*).

Начало и конец каждого сектора определяется углом поворота диска относительно магнитной головки, так что все дорожки поделены на секторы одинаковым образом (радиальные линии на рисунке 1). Из-за того, что длина сектора на внутренних дорожках меньше, чем на наружных, плотность записи информации на них выше, потому что все секторы имеют одинаковое количество информационных элементов.

Для уменьшения влияния номера дорожки на длину сектора число секторов на дорожках может уменьшаться по мере приближения к центру диска. Для пользователя диска (для файловой системы) это происходит незаметно, так как внутренний *контроллер* диска выполняет пересчет секторов с тем, чтобы внешне структура диска выглядела как равномерная.

Кроме того, секторы каждой последующей дорожки сдвигают относительно предыдущей для увеличения скорости доступа. Так как при непрерывном чтении секторов требуется переходить с дорожки на дорожку, то после перехода, на которое требуется некоторое время, под головкой оказывается не первый сектор следующей дорожки, а какой-то другой, из-за чего требуется сделать почти полный оборот диска для того, чтобы продолжить непрерывную передачу данных. В целом структуру, показанную на рисунке 1, имеют, пожалуй, только дискеты.

Сектором называют как сегмент круга, так и часть дорожки, попадающую в этот сегмент. На рисунке 1 синим цветом выделен сектор как часть дорожки.

Секторы также пронумерованы. Номер первого сектора равен 1. На рисунке номера секторов проставлены у отверстия для крепления диска.

Начало и конец каждого сектора на дорожке отмечаются на самой дорожке специальными маркерами (последовательностями бит), которые записываются во время операции **низкоуровневого форматирования**, которая обязательно должна быть выполнена перед первым использованием магнитного диска. В эти маркеры записывается адрес сектора и служебная информация для настройки скорости и синхронизации читающего устройства. При помощи адреса сектора управляющее диском устройство проверяет правильность установки головки на дорожку и определяет, что под головкой находится нужный сектор.

**Сектор дорожки** является самой важной частью структуры диска, потому что чтение и запись информации производится секторами. Нельзя прочитать или записать информацию объемом меньше сектора. Файловая система с этой точки зрения является ориентированной на блоки информации, так как низкоуровневые функции для чтения и записи информации на диск используют обращение к секторам.

Для увеличения емкости дискового устройства используют обе стороны диска, а в жестких дисках часто используют несколько дисков, посаженных на одно устройство вращения (*шпиндель*). У каждой рабочей поверхности каждого диска есть своя головка чтения-записи, и все головки составляют один блок. Поэтому перемещение головок происходит одновременно, хотя считывание или запись информации в один момент времени обычно производит только одна головка.

Для выбора той или иной головки они пронумерованы от нуля. На жестком диске самая верхняя головка нулевая. На гибком диске нулевая головка нижняя. Иногда вместо термина *головка* используют термин *поверхность*.

Для обращения к конкретному сектору диска нужно выбрать номер дорожки, номер головки и номер сектора. Эти три величины называют **физическим адресом** сектора.

На практике вместо номера дорожки используют номер цилиндра. **Цилиндр** — это совокупность всех дорожек (на всех поверхностях всех дисков) с одинаковым номером (перемещение блока головок в определенную позицию выбирает множество дорожек с одинаковым радиусом).

Секторы, на которые поделена вся поверхность магнитного диска, называют **абсолютными физическими секторами**. Файловая система обращается к абсолютным физическим секторам по их порядковому (*линейному*) номеру, который формируется следующим образом:

```
Все секторы цилиндра 0 :
    Все секторы головки 0
    Все секторы головки 1
    . . .
    Все секторы последней головки
Все секторы цилиндра 1 :
    Все секторы головки 0
    Все секторы головки 1
    . . .
    Все секторы последней головки
. . .
Все секторы последнего цилиндра :
    Все секторы головки 0
    Все секторы головки 1
    . . .
    Все секторы последней головки
```

Номер первого абсолютного физического сектора — 0. При обращении к диску файловая система пересчитывает номер сектора в его физический адрес. Порядковый номер сектора является логическим адресом.

Абсолютный сектор 0 имеет большое значение в структуре диска — его существование на диске предполагается *однозначно*. Поэтому нулевой сектор диска предназначен для хранения *информации о самом диске*.

Эта информация имеет отношение не к физической, а к логической организации диска. Если нельзя прочитать сектор 0 диска, можно сказать, что диска для файловой системы не существует, хотя при помощи специализированных программ его содержимое можно прочитать. Например, порча нулевой дорожки дискета приводит к невозможности прочитать дискету средствами файловой системы. Однако при помощи утилиты, например, *Disk Editor* пакета *Norton Utilities* можно прочитать секторы диска и скопировать их в виде файлов.

## Логическая структура магнитного диска

Логическая структура магнитного диска описывает распределение абсолютных физических секторов. Это распределение зависит от того, является ли диск жестким или гибким, а также от количества логических дисков, размещенных на поверхности физического диска.

Нулевой сектор жесткого магнитного диска содержит главную загрузочную запись (MBR, *Master Boot Record*), предназначенную для старта компьютера. Она содержит первичный загрузчик ОС и **таблицу разделов** жесткого диска (*Partition Table*).

Предполагается, что диск может быть поделен на части, называемые **разделами** (*partition*) для лучшей организации файлов и для обеспечения возможности

установки нескольких операционных систем. Разделы можно рассматривать как **логические диски**, разделяющие пространство физического диска (рисунок 2).

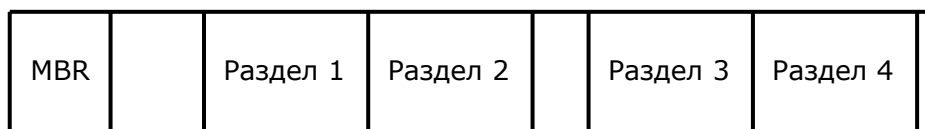


Рисунок 2 — Разделы жесткого диска

Каждый из 4-х возможных разделов диска может иметь свою независимую файловую систему. В зависимости от типа раздела, логический диск имеет операционную систему, которая может стартовать компьютер, или является просто логическим диском с данными.

На жесткий диск может быть установлено множество операционных систем. Выбор той или иной ОС при старте компьютера происходит по признаку активного раздела, установленному в таблице разделов для соответствующей ОС.

При старте компьютера загрузчик BIOS читает MBR и исполняет ее. Просматривая записи таблицы разделов, MBR определяет, какой из разделов активный, берет из таблицы адрес нулевого сектора этого раздела, загружает его в память и передает ему исполнение. В секторе 0 логического диска выбранного раздела должен находиться начальный загрузчик ОС, расположенной на этом диске.

Таким образом, возникает иерархия загрузчиков: загрузчик MBR — загрузчик ОС активного раздела. Если на диске один раздел, можно исключить загрузчик MBR, разместив логический диск на всей поверхности диска, а в секторе 0 — начальный загрузчик ОС. Такая организация диска присуща, например, дискетам.

Таблица разделов состоит из 4-х записей, называемых **дескрипторами разделов**. Дескриптор содержит физические адреса первого и последнего сектора раздела, и она определяет, таким образом, распределение дискового пространства. Эта информация является чрезвычайно важной для работы с физическим диском.

Структура физического сектора 0 (MBR):

Адрес	Назначение
000-1BDh	первичный загрузчик MBR
1BE-1CDh	Дескриптор раздела 1
1CE-1DDh	Дескриптор раздела 2
1DE-1EDh	Дескриптор раздела 3
1EE-1FDh	Дескриптор раздела 4
1FE-1FFh	Признак загрузочной записи AA55h

Структура дескриптора раздела:

Адрес	Назначение
00h	Признак активности раздела
01h	Начальная головка
02h-03h	Начальный цилиндр/сектор
04h	Тип раздела
05h	Конечная головка
06h-07h	Конечные цилиндр/сектор
08h-0Bh	Начальный сектор раздела относительно начала диска
0Ch-0Fh	Размер раздела в секторах

Расположение разделов на диске указывается в таблице разделов одновременно двумя способами: в формате CHS и в формате LBA.

Формат CHS (*Cylinder, Head, Sector*) является *координатным* способом адресации и задает физический адрес сектора: *цилиндр, поверхность* (головку) и *сектор* начала или конца раздела.

Формат LBA (*Logical Block Addressing*) является *линейным* способом адресации и задает логический адрес сектора (*порядковый номер*).

Недостаток формата CHS — ограничения на максимальные значения цилиндра, головки и сектора.

Для номера головки предусмотрено 8 бит, следовательно, описываемый диск не может иметь более 256 головок. Для номера сектора отводится 6 младших бит двухбайтного слова, следовательно, максимальное число секторов на дорожке не может превышать 63 (а не 64, так как секторы нумеруются с 1). Для номера цилиндра отводятся 10 бит, поэтому максимальное число цилиндров в формате CHS составляет 1024.

Ограничение на число цилиндров в формате CHS имеет неприятные последствия: так как при чтении сектора 0 (загрузочной записи) используется формат CHS, загрузочная запись должна располагаться не далее 1024 цилиндра, следовательно, не каждый раздел можно сделать загрузаемым.

Формат CHS работает только с дисками небольшого размера:  $1024 \times 256 \times 63 = 16\,515\,072$  сектора = 7,875 Гбайт. Для дисков большего размера используется формат LBA.

Количество цилиндров, головок и секторов, используемое в таблице разделов и репортируемое жестким диском, не совпадает с фактическими количествами цилиндров, головок и секторов. Попробуйте представить себе набор из 128 пластин и сравните его получаемую высоту с полудюймовой высотой винчестера. На самом деле внутри диска находится 1-3 пластины. Этот пример наглядно показывает, как решения, принятые на заре развития компьютерной индустрии, оказывают влияние на ее современное состояние.

Разделы, представляющие собой логические устройства, делятся на загрузаемые и незагрузаемые. Загрузаемый раздел (называемый также *primary* — основным) может иметь свою загрузочную запись и, соответственно, свою копию ОС. Незагрузаемые разделы предназначены для хранения данных. Так как объем дисков постоянно возрастает, необходимость в разделении всего дискового пространства на множество отдельных устройств является очевидной. Это связано также с необходимостью ограничить размер кластера.

Для того, чтобы поделить физический диск на большое число логических дисков, незагрузаемым разделам назначают тип «расширенный» (*Extended*). Расширенный раздел можно представить как новый физический диск со своей главной загрузочной записью типа MBR.

Запись расширенного раздела называют SMBR (*Secondary Master Boot Record*). Отличие этой записи заключается в том, что она не имеет загрузчика, а таблица разделов состоит из двух записей: основной раздел и расширенный раздел.

Таким образом, одна запись SMBR определяет один логический диск и, возможно, еще логические диски, которые образуют иерархию (все последующие логические диски оказываются вложенными друг в друга). Рисунок 3 поясняет использование расширенного раздела. На диске 2 раздела: основной (например, *Windows*, обозначен как DOS) и расширенный. Расширенный раздел определяет два дополнительных логических диска.

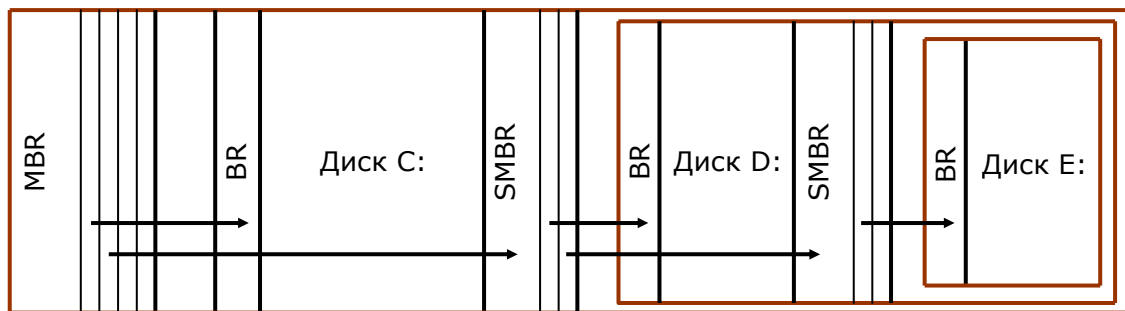


Рисунок 3 — Расширенный раздел

### Дополнительные сведения

Типы разделов:

- 00h - раздел не используется
- 01h - раздел FAT12 (16M)
- 02h - раздел ОС CP/M
- 03h - раздел ОС Xenix
- 04h - основной раздел FAT16 (32M)
- 05h - расширенный раздел FAT16 (2Г)
- 06h - основной раздел FAT16 (2Г)
- 07h - раздел ОС OS/2
- 0Bh - основной раздел FAT32 (2Т)
- 0Ch - расширенный раздел FAT32 (2Т)

### Принцип размещения файлов

Файловая система, основанная на FAT (*File Allocation Table* — Таблица Размещения Файлов), впервые появилась в ОС MS-DOS, но до сих пор используется в ОС *Windows*, а также для гибких магнитных дисков. Считается, что она происходит от файловой системы ОС CP/M, так как обе файловых системы обладают схожими элементами.

С помощью таблицы размещения файлов для каждого файла формируется односвязный список (рисунок 4, показан стрелками).



Рисунок 4 — Соотношение кластеров и ячеек FAT

Все пространство логического диска, отведенное под файлы, разбито на блоки одинакового размера, называемые кластерами (*Cluster*). В FAT каждому кластеру соответствует одна ячейка. Кластеры и соответствующие им ячейки одинаково пронумерованы.

Файл записывается на диск в один или несколько кластеров. В качестве примера рассмотрим размещение файла, занимающего 4 кластера: 4, 5, 8, 10.

Номер первого кластера файла 4 записывается в каталог.

Номер второго кластера файла 5 записывается в ячейку FAT, соответствующую номеру первого кластера 4. Номер третьего кластера файла 8 записывается в ячейку FAT, соответствующую номеру второго кластера 5 и так далее до последнего кластера 10, занимаемого файлом.

В ячейку FAT, соответствующую последнему кластеру, записывается специальный номер, обозначающий, что данный кластер последний. На рисунке 4 он обозначен EOF (*End Of File* — конец файла).

Таким образом, каждая ячейка FAT содержит номер следующего кластера файла.

Размер ячейки FAT накладывает ограничение на количество кластеров, так как в ячейку из  $n$  бит может быть записано максимум  $2^n$  чисел. В действительности максимальное количество кластеров на диске определяется по формуле  $2^n - 11$ . Во-первых, первые две ячейки FAT (с номерами 0 и 1) не используются для учета файлов — в них хранится байт-описатель диска. Во-вторых, 9 последних номеров используются в качестве специальных. Следующие номера кластеров имеют специальное значение:

(0..)000h - свободный кластер;  
(F..)FF7h - дефектный кластер;  
(F..)FF8h - (F..)FFFh - последний кластер файла (EOF);

Здесь (0..) обозначает одну или несколько дополнительных цифр. Для FAT12 дополнительных цифр нет, для FAT16 — 1 дополнительная цифра, для FAT32 — 5 дополнительных цифр, причем первая цифра равна нулю.

Если ячейка FAT содержит 0, кластер не занят никаким файлом. Если ячейка FAT содержит (F..)FF7h, кластер является дефектным. В качестве признака последнего кластера на практике используется значение (F..)FFFh.

**Кластер**, иначе называемый единицей выделения дискового пространства (*Allocation Unit*), — это минимальный объем диска, который может быть выделен файлу. Кластер — это логический блок диска (сектор — физический блок диска). Файл, независимо от размера, всегда занимает **целое** число кластеров. Кластер состоит из **смежных** (последовательных) секторов диска.

Кластер может иметь размер 1/2, 1, 2, 4, 8, 16 или 32 Кбайт. Соответственно, размер файла не может быть меньше, чем размер кластера. Учитывая, что фактический размер файла обычно число произвольное, последний кластер файла практически всегда занимается файлом лишь частично. Размер файла обычно меньше, чем выделяемое для его размещения на диске место. Это один из недостатков файловой системы FAT. Неиспользованные остатки кластеров, кстати, могут быть незаметно (для файловой системы) заняты вирусом.

Мы знаем, что минимальным объемом дискового пространства, который позволяет считывать или записывать аппаратура, равен одному сектору. Сектор имеет размер 512 байт (это не обязательное условие, сектор может иметь размер от 128

байт до 1 Кбайт). Почему разработчики MS-DOS использовали кластеры, ухудшая ситуацию с остатками кластеров?

Ответ заключается в том, что во время разработки файловой системы диски имели небольшой размер, и первоначально размер кластера был не больше, а *меньше* размера сектора. Системы с 12 битными ячейками FAT хватало для размещения файлов на диске емкостью 2042 Кбайт при размере кластера в 512 байт. Это может показаться смешным, но в то время для компьютеров были доступны только гибкие диски, а емкость одностороннего диска диаметром 5,25 дюйма первоначально составляла всего 160-180 Кбайт, а двустороннего — 320-360 Кбайт.

С появлением первого жесткого диска объемом более 5 Мбайт перед операционной системой MS-DOS возникла проблема. Число кластеров размером 1 Кбайт на диске составляет более 5000, а максимальное количество 12-битных ячеек FAT всего 4085. Поэтому фирма *Microsoft* разрешила использовать кластеры размером 1, 2 и 4 Кбайт. При 12-битных ячейках FAT это дает возможность использовать логические диски размером до 16 Мбайт. С учетом того, что жесткий диск может содержать 4 раздела, общая емкость диска могла составлять 64 Мбайт. Это предел для 12-битной системы FAT.

Дальнейшее увеличение емкости логического диска было возможно за счет увеличения размера ячейки FAT, что и было сделано. Систему с 12-битными ячейками назвали FAT12, а новую систему с 16-битными ячейками — FAT16. Максимальное число кластеров в ней составляет 65525. Так как объем дисковых устройств постоянно возрастал, то было разрешено использовать кластеры размером 8, 16 и 32 Кбайт и это окончательное решение. В результате максимальный объем логического диска, поддерживаемый системой FAT16 и, соответственно, MS-DOS, составляет 2 Гбайт (на самом деле чуть меньше — 1,999).

С появлением системы *Windows 95* и дисков большего размера появилась новая версия FAT, которую назвали FAT32, хотя на самом деле ячейки имели размер 28 бит. В операционной системе *Windows 95 OSR2* появилась новая версия FAT32 с 32-битными ячейками (из которых действительны только 28 бит), которая применяется в *Windows 98* и *Windows Me*.

Максимальный размер логического диска в FAT32 составляет 2 Тбайт, что не соответствует расчетам, и связано с особенностью файловой системы — FAT целиком размещается в основной (оперативной) памяти. Так, FAT16 требует 128 Кбайт памяти независимо от размера диска. FAT32 слишком велика для постоянного размещения в основной памяти, поэтому она хранится частями.

## Структура логического диска

### Введение

Логический диск FAT состоит из *системной* области и области файлов. Каждая часть логического диска занимает целое число логических секторов.

Секторы области файлов объединены в кластеры, размер кластера 1, 2, 4, 8, 16, 32 или 64 сектора по 512 байт. Номер первого кластера равен 2, так как первые две ячейки FAT зарезервированы (содержат байт-описатель).



## Логический диск FAT12/FAT16

На рисунке 5 приведена структура логического диска для файловых систем FAT12 и FAT16:

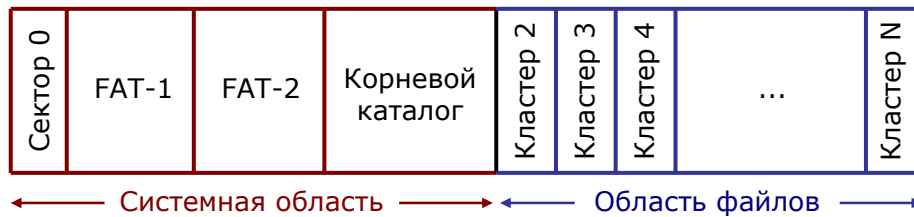


Рисунок 5 — Логический диск FAT16

Сектор 0, называемый также *начальным загрузчиком*, содержит таблицу параметров диска и начальный *загрузчик* операционной системы. Первые 3 байта сектора 0 содержат команду перехода JMP на начало загрузчика: либо байт 0E9h и 1 байт короткого смещения, за которым следует команда NOP (код 90h), либо байт 0EBh и два байта длинного смещения. Длинное смещение используется, если загрузчик располагается в зарезервированных секторах.

Далее расположено поле из 8 байт, в которое при форматировании заносится идентификатор версии ОС. Это текстовая строка, имеющая значение, например, MSDOS5.0 или MSWIN4.1.

Далее располагается таблица BPB (BIOS Parameter Block), описывающая физические характеристики диска и позволяющая вычислить правильный физический адрес на диске по данному номеру логического сектора. За таблицей располагаются дополнительные данные.

За сектором 0 могут быть расположены дополнительные секторы загрузчика. Общее число зарезервированных секторов, включая сектор 0, указывается в BPB.

За зарезервированными секторами (или непосредственно за сектором 0) располагается таблица размещения файлов (на рисунке FAT-1), после которой могут располагаться дополнительные копии FAT. Обычно используется 2 копии. Число копий указано в BPB.

В конце системной области расположен корневой каталог, который имеет фиксированный размер. Размер корневого каталога в каталожных записях также указан в таблице BPB.

Таблица BPB для FAT12, FAT16 содержит следующие характеристики диска:

Смещение	Назначение	Размер, байт	Для дискеты 1.44
00Bh	Размер сектора	2	512
00Dh	Размер кластера	1	1
00Eh	Число секторов загрузчика	2	1
010h	Число FAT	1	2
011h	Число записей корневого каталога	2	224
013h	Число секторов на диске	2	2880
015h	Байт-описатель носителя	1	F0 (hex)
016h	Размер FAT в секторах	2	9
018h	Число секторов на дорожке	2	18
01Ah	Число головок чтения/записи	2	2
01Ch	Число скрытых секторов	4	0
020h	Число секторов на диске (если диск > 32 Мб)	4	(нет)

За таблицей BPB расположена дополнительная информация для операционной системы:

Смещение	Назначение	Размер, байт	Для дискеты 1.44
024h	Физический номер диска (номер дисководов)	2	0
026h	Признак расширенного загрузчика	1	29h
027h	Серийный номер тома	4	
02Bh	Метка тома	11	
036h	Описатель FAT	8	FAT12

Физический номер диска для жестких дисков начинается с 80h.

Серийный номер тома — это шестнадцатеричное псевдослучайное число, предназначенное для того, чтобы ОС могла проверять смену пользователем дискета, когда он еще используется, но на самом деле такой проверки не делается.

Метка тома — это строка, предназначенная для идентификации тома пользователем.

Описатель FAT нельзя рассматривать как идентификатор файловой системы.

Далее с адреса 03Eh расположен начальный загрузчик.

Последние два байта сектора 0 содержат **признак загрузочной записи** — байт 55h по адресу 1FEh и байт AAh по адресу 1FFh (его можно рассматривать как слово AA55h по адресу 1FEh).

На рисунке 6 приведено начало дампа загрузочного сектора дискеты.

```

0000: EB 3C 90 2B 34 5F 52 3F 49 48 43 00 02 01 01 00 ы<P+4 R?INC.....
0010: 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 .p.@.E.....
0020: 00 00 00 00 00 00 29 03 08 42 27 4E 4F 20 4E 41 .....).B'NO NA
0030: 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 FA 33 ME FAT12 ·3

```

Рисунок 6 — Дамп загрузочного сектора дискеты

### Логический диск FAT32

Система FAT32 предназначена для дисков большого объема (более 2 Гбайт). В системе FAT16 диск большого объема имеет кластеры максимального размера, что приводит к потере до 30-50% дискового пространства в остатках кластеров. В FAT32 размер кластера может быть небольшим (4 Кбайт для диска 16 Гбайт), но число кластеров при этом значительно возрастает. Так как таблица FAT имеет слишком большой размер (16 Мбайт для диска 16 Гбайт), в FAT32 используются дополнительные структуры для ускорения работы.

На рисунке 7 приведена структура логического диска для файловой системы FAT32:

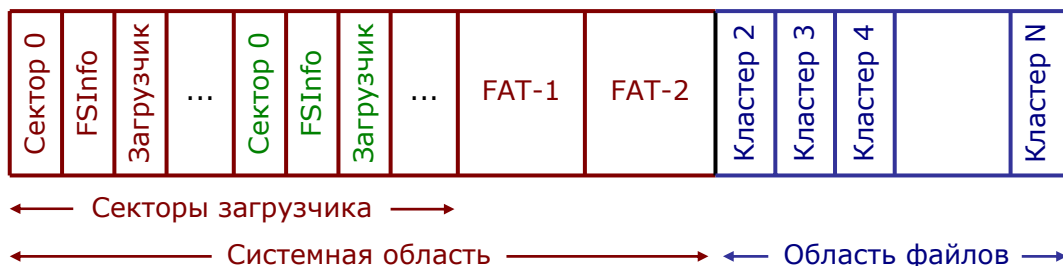


Рисунок 7 — Логический диск FAT32

Как видно из рисунка, FAT32 имеет расширенный загрузчик. Кроме того, сектор 1 содержит структуру FSInfo, предназначенную для управления FAT. Загрузочный сектор, структура FSInfo и загрузчик дублируются в зарезервированных секторах (выделено зеленым цветом).

Корневой каталог не имеет фиксированного положения и размера, как в FAT16. Вместо этого он является файлом специального вида, как и все другие каталоги. На практике корневой каталог обычно начинается с первого кластера области файлов (кластер 2).

Несмотря на то, что FAT32 использует ячейки размером 32 бита, старшие 4 бита не используются и всегда равны нулю.

Первые две ячейки таблицы FAT в системе FAT32 используются следующим образом:

- ячейка 0 содержит байт-описатель носителя, дополненный слева двоичными единицами;
- ячейка 1 содержит код признака конца цепочки кластеров файла (для FAT32 это обычно код 0F FF FF FFh).

Бит 27 ячейки 1 является признаком завершения работы с диском: 1 — работа завершена нормальным образом, 0 — нет. Бит 26 ячейки 1 является аналогичным признаком нормального завершения операции ввода-вывода.

Таблица BPB для FAT32 содержит следующие характеристики диска:

Смещение	Назначение	Размер, байт
00Bh	Размер сектора	2
00Dh	Размер кластера	1
00Eh	Число секторов загрузчика	2
010h	Число FAT	1
011h	Число записей корневого каталога	0 (не используется)
013h	Число секторов на диске	2 (если секторов меньше 65536)
015h	Байт-описатель носителя	1
016h	Размер FAT в секторах	0 (не используется)
018h	Число секторов на дорожке	2
01Ah	Число головок чтения/записи	2
01Ch	Число скрытых секторов	4
020h	Число секторов на диске	4 (если секторов больше 65535)

За таблицей BPB расположена дополнительная информация для операционной системы:

Смещение	Назначение	Размер, байт
024h	Размер FAT в секторах	4
028h	Номер активной FAT	2
02Ah	Номер версии FAT	2
02Ch	Первый кластер корневого каталога	4
030h	Номер сектора структуры FSInfo	2 (обычно 1)
032h	Номер первого сектора копии	2 (обычно 6)
034h	Зарезервировано	12
041h	Физический номер диска	1
042h	Зарезервировано	1
043h	Серийный номер тома	4
047h	Метка тома	11
052h	Описатель FAT	8

Как и в случае FAT16, последние два байта сектора 0 содержат признак загрузочной записи AA55h.

Структура FSInfo предназначена для ускорения выполнения операций с FAT. В ней содержится количество свободных секторов и номер первого свободного кластера. Формат структуры приведен в следующей таблице:

Смещение	Назначение	Размер, байт
000h	Признак структуры 41615252h	4
004h	Резерв	480
1E4h	Признак структуры 61417272h	4
1E8h	Текущее число свободных кластеров	4
1ECh	Номер первого свободного кластера	4
1F0h	Резерв	12
1FCh	Признак структуры AA550000h	4

Если текущее число свободных кластеров содержит FFFFFFFFh, это означает, что оно неизвестно и его следует вычислять. Если такое же число находится в поле первого свободного кластера, это означает, что искать свободный кластер необходимо с кластера 2. Другое значение номера первого свободного кластера указывает не на первый свободный кластер, а на кластер, с которого нужно начинать поиск свободных кластеров.

### Дополнительные сведения

Значения байта-описателя носителя:

0F0h	- дискет 3.5" (2 стороны, 18 секторов - 1.44 Мб)
0F8h	- жесткий диск
0F9h	- дискет 5.25" (2 стороны, 15 секторов)
0FCh	- дискет 5.25" (2 стороны, 15 секторов) или 3.5" (2 стороны, 9 секторов)
0FDh	- дискет 5.25" (1 сторона, 9 секторов) или 8" (1 сторона, 1x плотность)
0FEh	- дискет 5.25" (1 сторона, 8 секторов) или 8" (1/2 стороны, 1x/2x плотность)
0FFh	- дискет 5.25" (2 стороны, 8 секторов) или 8" (2 стороны, 2x плотность)

## Каталоги и каталожные записи

### Введение

Каталог (*Directory*) — это, логически, список файлов. Физически каталог является файлом специального вида. Исключение составляет корневой каталог FAT12/FAT16 — он не является файлом, хотя имеет структуру, полностью аналогичную структуре каталога-файла.

Каталоги в файловой системе FAT образуют иерархию: одни каталоги «вложены» в другие. На самом деле никакой вложенности одних файлов в другие не существует. Файлы приписываются к тем или иным каталогам, но располагаются в пространстве логического диска там, где есть место. Иллюзия вложенности возникает из-за того, что увидеть файл, расположенный в каком-либо каталоге, можно только прочитав этот каталог, «открыв» его.

Единственное назначение каталогов — структурировать большой объем файлов, объединив их в логические группы.

Файл каталога состоит из записей размером ровно 32 байта, называемых **каталожными** записями.

## Каталожная запись MS-DOS

На рисунке 8 приведен формат каталожной записи, применяющейся в MS-DOS (FAT12/FAT16).

Полное имя файла - 11 байт				A	Резерв
Резерв	Time	Date	Cluster	Size	

Рисунок 8 — Каталогная запись MS-DOS

Поля каталожной записи имеют следующее значение:

Смещение	Назначение	Размер, байт
00h	Полное имя файла	11
0Bh	Байт атрибутов файла (поле A)	1
0Ch	Зарезервировано	10
16h	Время изменения файла (Time)	2
18h	Дата изменения файла (Date)	2
1Ah	Номер первого кластера файла (Cluster)	2
1Ch	фактический размер файла, байт (Size)	4

**Полное имя файла** указывается в формате 8+3:

- 8 символов для имени,
- 3 символа для расширения.

Поля выравниваются влево, неиспользуемые символы заполняются символом «пробел» (код 20h), например:

```
12345678123
MSDOS   SYS (для файла с именем MSDOS.SYS)
FILE    1  (для файла с именем FILE.1)
```

Такой формат принят в системных функциях MS-DOS, предназначенных для операций с файлами и каталогами. Буквы в имени и расширении файла в каталожной записи приводятся к регистру **ПРОПИСНЫЕ**.

**Атрибут** файла — это признак, относящий файл (запись) к той или иной [логической] группе. Байт атрибутов имеет формат, представленный на рисунке 2. Биты 6 и 7 зарезервированы.

7	6	5	4	3	2	1	0
0	0	A	D	V	S	H	R
		20h	10h	8h	4h	2h	1h

Рисунок 9 — Байт атрибутов MS-DOS

В MS-DOS предусмотрены следующие атрибуты для файлов (и каталогов), которые видны пользователю:

Бит	Маска	Назначение
0	01h	Read only - файл только для чтения
1	02h	Hidden - Скрытый файл
2	04h	System - Системный файл
5	20h	Archive - Архивный файл

**Файл только для чтения** дополнительно защищается от случайного удаления или модификации пользователем.

**Скрытые и системные** файлы исключаются из списка отображаемых в каталоге (исключаются из нормальной процедуры поиска записи в каталоге), системный файл дополнительно защищается от случайного удаления.

**Архивный** файл подлежит операции резервного копирования. После выполнения этой операции этот атрибут очищается, а операция модификации файла устанавливает его снова. Таким образом становится известно, что файл был изменен и его следует заменить в архиве.

Следующие атрибуты не видны пользователю, и используются операционной системой:

Бит	Маска	Назначение
3	08h	Volume Label - Метка тома
4	10h	Directory - Каталог

Записи о каталогах всегда имеют поле «размер файла», равное 0, независимо от фактического размера каталога.

Значение 0 в поле «первый кластер» записи о каталоге означает, что каталог находится в корневом (родительский каталог — корневой).

**Время создания файла** записывается в следующем формате:

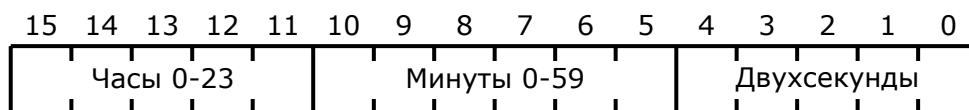


Рисунок 10 — Формат записи времени

Так как ширина поля для записи секунд составляет 5 бит (32 возможных значения), секунды записываются при помощи двухсекундных единиц, от 0 до 29 (от 0 до 58 секунд).

**Дата создания файла** записывается в формате:

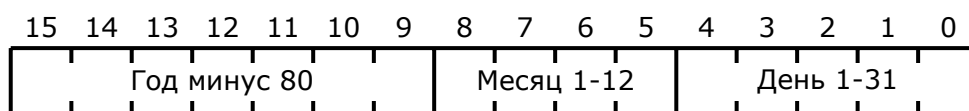


Рисунок 11 — Формат записи даты

Год записывается от 1980, например, 2003 год будет записан как 23 (2003-1980), а 1960 и 2060 год будут записаны одинаково как 80 (2060-1980).

### Специальные записи

Все каталоги, кроме корневого, в качестве первых двух записей содержат специальные записи текущего и родительского каталогов. В этих записях в качестве имен используются точка (.) для текущего каталога и две точки (..) для родительского каталога. Пример — начало кластера 2 дискета, содержащего каталог, находящийся в корневом каталоге (подчеркнуты поля «первый кластер»):

```

0000: 2E 20 20 20 20 20 20 20 20 20 20 20 10 00 00 00 00 . . . . .
0010: 00 00 00 00 00 00 00 14 06 82 2F 02 00 00 00 00 00 . . . . .В/ . . . . .
0020: 2E 2E 20 20 20 20 20 20 20 20 20 20 20 10 00 00 00 00 .. . . . .
0030: 00 00 00 00 00 00 00 14 06 82 2F 00 00 00 00 00 00 . . . . .В/ . . . . .
  
```

Запись, первый байт которой равен изображению (вид этого байта зависит от кодировки), означает, что файл был удален.

Запись, первый байт которой равен 05h, означает, что фактически первый символ — E5h.

Запись, первый байт которой равен 0, означает конец каталога. Эта и все последующие записи никогда не использовались.

Специальная запись «метка тома» (*Volume Label*) создается в корневом каталоге, если пользователь при форматировании указал метку тома или выполнил команду присвоения метки тома диску. В операционной системе *Windows* эта операция может быть выполнена при помощи свойств диска. Эта запись имеет атрибут, в котором установлен бит 8 (бит «метка тома»), а поля «первый кластер» и «размер» равны нулю. Поле имени файла заполняется меткой длиной до 11 символов (подчеркнуты байт атрибутов и поле «первый кластер»):

```
0000: 4D 59 20 44 49 53 4B 20 20 20 28 00 00 00 00 MY DISK .....
0010: 00 00 00 00 00 00 14 06 82 2F 00 00 00 00 00 00 .....B/.....
```

### Допустимые символы MS-DOS

В качестве символов для имен файлов и каталогов в MS-DOS можно использовать:

- буквы латинского алфавита A..Z
- цифры 0..9
- знаки \$ % ' - \_ @ { } ~ ` ! # ( ) & ^

Точка разделяет имя и расширение имени при вводе и выводе имени, но в каталожной записи она опускается.

В каталожной записи буквы строчного регистра преобразуются к прописному.

### Каталожная запись FAT32 (Windows 98)

При создании файловой системы FAT32 нужно было решить множество проблем:

- расширить поле для первого кластера до 4-х байт;
- добавить поля для даты и времени создания и последнего доступа;
- разрешить длинные имена файлов;
- обеспечить обратную совместимость с MS-DOS;
- найти механизм для защиты новых записей от MS-DOS.

Обеспечение обратной совместимости чрезвычайно важно, так как на дисках и других сменных носителях применяется система FAT12/16, и использование этих носителей на компьютерах, в которых установлена ОС MS-DOS или, скажем, *Windows 3.1*, привело бы к невозможности обмена информацией или к ее потере.

В результате в системе FAT32 применяются два **формата каталожных записей** одновременно.

Прежде всего, чтобы обеспечить возможность чтения длинных имен файлов (появившихся в *Windows 95 OSR2*) в операционных системах, использующих формат имени **8+3**, было введено понятие **короткого имени MS-DOS**. Это имя файла или каталога, получаемое из длинного имени приведением к формату **8+3** по определенным правилам.

Так как в длинных именах файлов разрешается использовать намного больше символов, чем в MS-DOS, недопустимые символы заменяются на возможно похо-

жие символы из второй половины таблицы ASCII-8 (коды 128-175) и приводятся к верхнему регистру (прописные). Пробелы и точки (за исключением точки, которая соединяет имя и расширение) удаляются, недопустимые символы, например, точка с запятой, заменяются символом подчеркивания.

Далее имя файла отделяется от расширения и усекается до первых 6-ти символов, к полученному имени приписывается 2 символа: ~1.

Новое имя проверяется на допустимость — в каталоге не должно быть двух одинаковых имен. Если такое имя в каталоге есть, число 1 после тильды заменяется числом 2 и так далее, пока не будет образовано уникальное имя. Если число после тильды превысит один разряд (например, станет равным 10), то первые 6 символов усекаются до пяти. Если исходное имя короче 6-ти символов, ~1 приписывается все равно.

Далее приведены примеры получения короткого имени из длинного:

Мой.txt преобразуется к МОЙ~1.ТХТ  
 Мой файл.txt преобразуется к МОЙФАЙ~1.ТХТ  
 a.b;c.d преобразуется к АВ\_С~1.D

Короткое имя записывается в каталожную запись, формат которой приведен на рисунке 12:

Короткое имя MS-DOS - 11 байт						A	NT	S	Cre T
Cre D	Last	Cl Hi	Mod T	Mod D	Cl Lo	Size			

Рисунок 12 — Каталожная запись MS-DOS FAT32

Формат этой записи совместим с форматом записи MS-DOS, она используется в качестве основной, хранящей сведения о файле, и в FAT16 и в FAT32.

Назначение полей приведено в следующей таблице:

Смещение	Назначение	Размер, байт
00h	Короткое имя файла	11
0Bh	Байт атрибутов файла (поле A)	1
0Ch	Используется Windows NT (поле NT)	1
0Dh	Секунды и сотые доли секунды времени создания (S)	1
0Eh	Время создания файла (Cre T)	2
0Fh	Дата создания файла (Cre D)	2
12h	Дата последнего доступа (Last)	2
14h	Старший байт номера первого кластера (Cl Hi)	2
16h	Время изменения файла (Time)	2
18h	Дата изменения файла (Date)	2
1Ah	Младший байт номера первого кластера (Cl Lo)	2
1Ch	Фактический размер файла, байт (Size)	4

Поле NT используется FAT32 для своих целей. Если к записи добавляются записи с длинным именем, это поле равно нулю.

Поле S содержит число, прибавляемое к двухсекундам времени создания поля Cre T, чтобы получить время создания с точностью до сотых долей секунды. Например, если двухсекунды равны 36, S равно 154, секунды создания равны 37,54.

Для записи длинного имени файла введен еще один формат каталожной записи, приведенный на рисунке 13:



N	1	2	3	4	5	0F	0	CS	6
7	8	9	10	11	0	12	13		

Рисунок 13 — Каталогная запись Windows FAT32

В этом формате поле атрибутов файла всегда равно 0Fh, что является недопустимым сочетанием в записи MS-DOS. Благодаря этому новые записи «не видны» для MS-DOS.

Поле для начального кластера файла всегда равно нулю, чтобы избежать ошибок при трактовании этой записи, как правильной.

Поле NT всегда равно 0, а в поле CS находится байт контрольной суммы. Контрольная сумма используется для проверки каталога на случай, если система MS-DOS изменит данную запись (удалит).

Так как длинное имя файла слишком велико, чтобы уместиться в одной записи, система FAT32 записывает их столько, сколько нужно, с учетом того, что в одну запись можно записать до 13-ти символов в кодировке *Unicode*. Минимальное число новых записей 1, максимальное — 20.

Расположение символов в записи показано на рисунке цифрами от 1 до 13. Следует учитывать, что в кодировке *Unicode* каждый символ занимает минимум 2 байта. Последовательность символов заканчивается нулем, а неиспользуемые позиции символов заполняются кодом FFFFh.

Кроме того, имя файла записывается как есть, вместе с точкой, которая соединяет расширение. Недопустимыми символами в имени файла являются только следующие знаки: ":", "\", "/" и "|".

Каждая новая запись пронумерована двоичным числом от 1, которое записывается в поле, обозначенное N, при этом к номеру последней записи прибавляется 64 (40h). Наконец, новые записи располагаются в обратном порядке непосредственно перед записью с коротким именем.

Ниже приведен пример записей для файла с именем test.txt (длина имени до 8 символов):

```

0000: 41 74 00 65 00 73 00 74 00 2E 00 0F 00 8F 74 00 At.e.s.t.....Pt.
0010: 78 00 74 00 00 00 FF FF FF FF 00 00 FF FF FF FF x.t.....
0020: 54 45 53 54 20 20 20 20 54 58 54 20 00 75 B3 02 TEST TXT .u|.
0030: 84 2F 84 2F 00 00 B2 7E 83 2F 02 00 2B 02 00 00 Д/Д/...~Г/...+...

```

Еще один пример показывает, как располагаются несколько новых записей. Имя файла This Is A Long File Name.txt (длина 28 символов, имя — 24 символа):

```

0000: 43 78 00 74 00 00 00 FF FF FF FF 0F 00 43 FF FF Cx.t.....C..
0010: FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF .....
0020: 02 67 00 20 00 46 00 69 00 6C 00 0F 00 43 65 00 .g. .F.i.l...Ce.
0030: 20 00 4E 00 61 00 6D 00 65 00 00 00 2E 00 74 00 .N.a.m.e.....t.
0040: 01 54 00 68 00 69 00 73 00 20 00 0F 00 43 49 00 .T.h.i.s. ...CI.
0050: 73 00 20 00 41 00 20 00 4C 00 00 00 6F 00 6E 00 s. .A. .L...o.n.
0060: 54 48 49 53 49 53 7E 31 54 58 54 20 00 08 F1 03 THISIS~1TXT ..ё.
0070: 84 2F 84 2F 00 00 B2 7E 83 2F 02 00 2B 02 00 00 Д/Д/...~Г/...+...

```

Подчеркнуты поля, не относящиеся к символам имени файла.

## Дополнительные сведения

При переименовании файла файловая система FAT формирует новые каталожные записи, помечая текущие как удаленные. Если в каталоге есть удаленные записи, в качестве новых записей используются они.

Использование записей, сформированных FAT32, ограничено в MS-DOS: если имя содержит недопустимые для MS-DOS символы (например, русские), файл нельзя будет переименовать или удалить (но можно удалить командой удаления группы файлов, в которой имя файла не указывается, например, командой DEL.). Для чтения и редактирования файл доступен.

Для символов кириллицы в кодировке *Unicode* выделены коды, приведенные в таблице (приведены только младшие байты, старшие равны 4).

Буква	Код	Буква	Код	Буква	Код	Буква	Код
А	10h	Р	20h	а	30h	р	40h
Б	11h	С	21h	б	31h	с	41h
В	12h	Т	22h	в	32h	т	42h
Г	13h	У	23h	г	33h	у	43h
Д	14h	Ф	24h	д	34h	ф	44h
Е	15h	Х	25h	е	35h	х	45h
Ж	16h	Ц	26h	ж	36h	ц	46h
З	17h	Ч	27h	з	37h	ч	47h
И	18h	Ш	28h	и	38h	ш	48h
Й	19h	Щ	29h	й	39h	щ	49h
К	1Ah	Ъ	2Ah	к	3Ah	ъ	4Ah
Л	1Bh	Ы	2Bh	л	3Bh	ы	4Bh
М	1Ch	Ь	2Ch	м	3Ch	ь	4Ch
Н	1Dh	Э	2Dh	н	3Dh	э	4Dh
О	1Eh	Ю	2Eh	о	3Eh	ю	4Eh
П	1Fh	Я	2Fh	п	3Fh	я	4Fh
Ё	01h	ё	51h				

Кодировка символов английского алфавита, цифр и знаков в *Unicode* соответствует таблице ASCII с добавлением старшего байта, равного нулю.

## Литература

1. Данкан Р. Профессиональная работа в MS-DOS: Пер. с англ. — М.: Мир. 1993. — 509 с., ил.

2. Нортон П. Персональный компьютер фирмы IBM и операционная система MS-DOS: Пер. с англ. - М.: Радио и связь. 1992 г. - 416 с.: ил.

Дата создания: 30.12.2003.

Дата модификации: 30.09.2016.

Владимир Вадимович Пономарев

245-59@mail.ru

revol.ponomcom.ru