

FAT

Материал из Википедии — свободной энциклопедии

FAT (англ. *File Allocation Table* «таблица размещения файлов») — классическая архитектура файловой системы, которая из-за своей простоты всё ещё широко применяется для флеш-накопителей. Используется в дискетах, картах памяти и некоторых других носителях информации. Ранее находила применение и на жёстких дисках.

Разработана Биллом Гейтсом и Марком МакДональдом в 1976—1977 годах^{[1][2]}. Использовалась в качестве основной файловой системы в операционных системах семейств MS-DOS и Windows 9x.

Структура FAT следует стандарту ECMA-107 и подробно определяется официальной спецификацией Microsoft, известной под названием FATGEN^[3].

Содержание

Версии системы FAT

VFAT

Структура системы FAT

Загрузочная запись

Блок параметров BIOS (BPB)

FSInfo

Определение типа FAT тома

Серийный номер тома

Таблица FAT

Файловые записи

Корневой каталог

Структура файловой записи

Имя файла в FAT

Файловые атрибуты

Что происходит при создании каталога

Время и дата

LFN-записи

Смысл файловых операций в FAT

Отказоустойчивость системы

Характеристики^[3]

Лицензирование

См. также

Примечания

Ссылки

Версии системы FAT

Существует четыре версии FAT — **FAT12**, **FAT16**, **FAT32** и **exFAT (FAT64)**. Они отличаются разрядностью записей в дисковой структуре, то есть количеством бит, отведённых для хранения номера кластера. FAT12 применяется в основном для дискет, FAT16 — для дисков малого объёма. На основе FAT была разработана новая файловая система **exFAT** (extended FAT), используемая преимущественно для флеш-накопителей.

Изначально FAT не поддерживала иерархическую систему каталогов — все файлы располагались в корне диска. Это было сделано для упрощения, так как на односторонних дискетах ёмкостью всего 160–180 Кбайт сортировать немногочисленные файлы по каталогам попросту не было смысла. С распространением дискет на 320 и более килобайт хранение всех файлов в корне оказалось неудобным, к тому же малый размер корневого каталога ограничивал количество файлов на диске. Каталоги были введены с выходом MS-DOS 2.0.

В разных операционных системах также были внедрены различные расширения FAT. Например, в DR-DOS имеются дополнительные атрибуты доступа к файлам; в Windows 95, Linux — поддержка длинных имён файлов (LFN) в формате Unicode (Virtual FAT — VFAT); в OS/2 — расширенные атрибуты всех файлов.

VFAT

VFAT — расширение FAT, появившееся в Windows 95. В FAT имена файлов имеют формат 8.3 и состоят только из символов кодировки ASCII. В VFAT была добавлена поддержка длинных (до 255 символов) имён файлов (англ. *Long File Name, LFN*) в кодировке UTF-16LE, при этом LFN хранятся одновременно с именами в формате 8.3, ретроспективно называемыми SFN (англ. *Short File Name*). LFN нечувствительны к регистру при поиске, однако, в отличие от SFN, которые хранятся в верхнем регистре, LFN сохраняют регистр символов, указанный при создании файла^{[4][5]}.

Структура системы FAT

В файловой системе FAT смежные секторы диска объединяются в единицы, называемые кластерами. Количество секторов в кластере равно степени двойки (см. далее). Для хранения данных файла отводится целое число кластеров (минимум один), так что, например, если размер файла составляет 40 байт, а размер кластера 4 Кбайт, реально занят информацией файла будет лишь 1 % отведённого для него места. Во избежание подобных ситуаций целесообразно уменьшать размер кластеров, а для сокращения объёма адресной информации и повышения скорости файловых операций — наоборот. На практике выбирают некоторый компромисс. Так как ёмкость диска вполне может и не выражаться целым числом кластеров, обычно в конце тома присутствуют так называемые *surplus sectors* — «остаток» размером менее кластера, который не может отводиться ОС для хранения информации.

Пространство тома FAT32 логически разделено на три смежные области:

- Зарезервированная область. Содержит служебные структуры, которые принадлежат загрузочной записи раздела (Partition Boot Record — PBR, для отличия от Master Boot Record — главной загрузочной записи диска; также PBR часто некорректно называется загрузочным сектором) и используются при инициализации тома;
- Область таблицы FAT, содержащая массив индексных указателей («ячеек»), соответствующих кластерам области данных. Для повышения надёжности на диске обычно представлено две копии таблицы FAT;
- Область данных, где записано собственно содержимое файлов — то есть текст текстовых файлов, кодированное изображение для файлов рисунков, оцифрованный звук для аудиофайлов и т. д.

В FAT12 и FAT16 также специально выделяется область корневого каталога. Она имеет фиксированное положение (непосредственно после последнего элемента таблицы FAT) и фиксированный размер в 32-байтных элементах, то есть при описании в Partition Boot Record указывается именно количество 32-байтных элементов, каждый из которых описывает какой-либо элемент корневого каталога (будь то файл или другой вложенный каталог).

Если кластер принадлежит файлу, то соответствующая ему ячейка в таблице FAT содержит номер следующего кластера этого же файла. Если ячейка соответствует последнему кластеру файла, то она содержит специальное значение (0xFFFF для FAT16). Таким образом выстраивается цепочка кластеров файла. Неиспользуемым кластерам в таблице соответствуют нули. «Плохим» кластерам (которые исключаются из обработки, например, по причине нечитаемости соответствующей области устройства) также соответствует специальный код (0xFFFF7 для FAT16).

При удалении файла первый знак имени заменяется специальным кодом 0xE5, и цепочка кластеров файла в таблице размещения обнуляется. Поскольку информация о размере файла (которая располагается в каталоге рядом с именем файла) при этом остаётся нетронутой, в случае, если кластеры файла располагались на диске последовательно и не были перезаписаны новой информацией, удалённый файл можно восстановить.

Загрузочная запись

Первая структура тома FAT называется BPB (англ. *BIOS parameter block*) и расположена в зарезервированной области, в нулевом секторе. Эта структура содержит информацию, идентифицирующую тип файловой системы и физические характеристики носителя (дискеты или раздела на жёстком диске).

Блок параметров BIOS (BPB)

BPB отсутствовал в FAT, обслуживавшей MS-DOS 1.x, так как в то время предполагалось лишь два различных типа тома — одно- и двусторонние пятидюймовые дискеты на 320 Кбайт, причём формат тома определялся по первому байту области FAT. BPB был введён в MS-DOS 2.x в начале 1983 года как обязательная структура загрузочного сектора, по которой впредь следовало определять формат тома; старая схема определения по первому байту FAT больше не поддерживалась. Также в MS-DOS 2.0 была введена иерархия файлов и папок (до этого все

файлы хранились в корневом каталоге).

Структура BPB в MS-DOS 2.x содержала 16-битное поле «общего количества секторов», что означало принципиальную неприменимость этой версии FAT для томов объёмом более $2^{16} = 65\,536$ секторов, то есть более 32 Мбайт при стандартном размере сектора 512 байт. В MS-DOS 4.0 (1988 г.) поле BPB было расширено до 32 бит, что означало увеличение теоретического размера тома до $2^{32} = 4\,294\,967\,296$ секторов, то есть до 2 Тбайт при 512-байтовом секторе.

Следующая модификация BPB появилась вместе с Windows 95 OSR2, в которой была введена FAT32 (в августе 1996 года). Было снято ограничение в 2 Тбайт на размер тома, том FAT32 теоретически может иметь размер до 8 Тбайт. Впрочем, размер каждого отдельного файла при этом не может превышать 4 Гбайт. BIOS Parameter Block в FAT32 в целях совместимости с ранними версиями FAT повторяет BPB у FAT16 вплоть до поля BPB_TotSec32 включительно, далее следуют различия.

«Загрузочный сектор» FAT32 в действительности представляет собой три 512-байтовых сектора — сектора 0, 1 и 2. Каждый из них содержит сигнатуру 0xAA55 по адресу 0x1FE, то есть в последних двух байтах, в случае если размер сектора равен 512 байт. Если же размер сектора больше 512 байт, то сигнатура содержится как по адресу 0x1FE, так и в последних двух байтах нулевого сектора, то есть дублируется.

FSInfo

Загрузочная запись раздела FAT32 содержит структуру под названием *FSInfo*, используемую для хранения значения числа свободных кластеров тома. FSInfo, как правило, занимает сектор 1 (см. поле BPB_FSInfo) и имеет следующую структуру (адреса относительно начала сектора):

- **FSI_LeadSig.** 4-байтовая подпись 0x41615252, свидетельствует, что сектор используется для структуры FSInfo.
- **FSI_Reserved1.** Промежуток с 4 по 483 байт сектора включительно, обнуляется.
- **FSI_StrucSig.** Ещё одна подпись, расположена по адресу 0x1E4 и содержит значение 0x61417272.
- **FSI_Free_Count.** 4-байтовое поле по адресу 0x1E8, содержит последнее известное системе значение числа свободных кластеров тома. Значение 0xFFFFFFFF означает, что число свободных кластеров неизвестно и должно вычисляться.
- **FSI_Nxt_Free.** 4-байтовое поле по адресу 0x1EC, содержит номер кластера, от которого должен начинаться поиск свободных кластеров по таблице индексных указателей. Обычно это поле содержит номер последнего кластера FAT, отведённого для хранения файла. Значение 0xFFFFFFFF означает, что поиск свободного кластера должен проводиться с самого начала таблицы FAT, то есть со второго кластера.
- **FSI_Reserved2.** Зарезервированное 12-байтное поле по адресу 0x1F0.
- **FSI_TrailSig.** Подпись 0xAA550000 — последние 4 байта сектора FSInfo.

Смысл введения FSInfo — в оптимизации работы системы, так как в FAT32 таблица индексных указателей может быть весьма большой, а её побайтовый просмотр — занять

значительное время. Однако значения полей FSI_Free_Count и FSI_Nxt_Free могут не соответствовать действительности и должны проверяться на адекватность. Кроме того, они даже не обновляются в резервной копии FSInfo, расположенной, как правило, в секторе 7.

Определение типа FAT тома

Определение типа FAT тома (то есть выбор между FAT12, FAT16 и FAT32) производится ОС по количеству кластеров в томе, которое, в свою очередь, определяется из полей BPB. Прежде всего вычисляется количество секторов корневого каталога:

$$\text{RootDirSectors} = (\text{BPB_RootEntCnt} * 32) / \text{BPB_BytsPerSec}$$

Далее определяется, какие из полей BPB_FATSz16/32 и BPB_TotSec16/32 не равны нулю, и они используются при определении количества секторов области данных тома:

$$\text{DataSec} = \text{TotSec} - (\text{BPB_ResvdSecCnt} + (\text{BPB_NumFATs} * \text{FATSz}) + \text{RootDirSectors})$$

Наконец, определяется количество кластеров области данных:

$$\text{CountofClusters} = \text{DataSec} / \text{BPB_SecPerClus}$$

По количеству кластеров проводится однозначное соответствие с файловой системой:

- CountofClusters < 4085 — FAT12
- CountofClusters = 4085 ÷ 65 524 — FAT16
- CountofClusters > 65 524 — FAT32

Согласно официальной спецификации, это единственный допустимый способ определения типа FAT. Искусственное создание тома, нарушающего указанные правила соответствия, приведёт к его некорректной обработке Windows. Тем не менее рекомендуется избегать значений CountofClusters, близких к критическим (4085 и 65 525), для верного определения типа файловой системы любыми, часто некорректно написанными драйверами.

На диске при форматировании всегда создаётся FAT12. Что касается жёстких дисков и флеш-накопителей, то при размере накопителя до 512 Мбайт (при 512-байтовом секторе) по умолчанию создаётся FAT16, свыше 512 Мбайт — FAT32. Размер кластера определяется при форматировании, исходя из файловой системы и размера тома.

Серийный номер тома

Серийный номер тома (поле BS_VolID) в Windows 98 создаётся из даты и времени форматирования таким образом, что без дополнительной информации восстановить их невозможно.

Таблица FAT

Следующая важная структура тома FAT — сама таблица FAT, занимающая отдельную логическую область. Она определяет список (цепочку) кластеров, в которых размещаются файлы и папки тома. Между кластерами и индексными указателями таблицы имеется

взаимно однозначное соответствие — N-й указатель соответствует кластеру с тем же номером. Первому кластеру области данных присваивается номер 2. Значение индексного указателя соответствует состоянию соответствующего кластера. Возможны следующие состояния:

- кластер свободен — указатель обнулён;
- кластер занят файлом и не является последним кластером файла — указатель содержит номер следующего кластера файла;
- кластер является последним кластером файла — указатель содержит метку EOC (End Of Clusterchain), значение которой зависит от версии FAT: для FAT12 меткой EOC считается любое значение, большее или равное 0x0FF8 (по умолчанию 0x0FFF); для FAT16 — большее или равное 0xFFFF (по умолчанию 0xFFFF); для FAT32 — любое значение, большее или равное 0x0FFFFFF8 (по умолчанию 0x0FFFFFFF);
- кластер повреждён — указатель содержит специальную метку, значение которой для FAT12 0x0FF7, для FAT16 0xFFFF7 и для FAT32 0x0FFFFFF7. Повреждённый кластер не может использоваться файловой системой для хранения данных; соответствующие указатели не затрагиваются при форматировании тома, когда все остальные указатели обнуляются;
- кластер зарезервирован «для будущей стандартизации» — указатель содержит значение, превышающее CountofClusters, но меньше метки повреждённого кластера (то есть до 0xFFFF6 включительно для FAT16). В этом случае кластер, не соответствуя никаким реальным данным, считается занятым и пропускается при поиске свободного, но никакой другой информации о нём не предоставляется.

Кластеры 0 и 1 отражаются FAT особо. Индексный указатель, соответствующий нулевому кластеру (самый первый указатель таблицы FAT), содержит значение BPB_Media в нижних 8 битах; остальные биты устанавливаются в 1. Например, если BPB_Media = 0xF8 (жёсткий диск), FAT[0] = 0x0FFFFFF8 для FAT32. Таким образом, формально FAT[0] = EOC, что используется при обработке файлов нулевого размера (см. далее).

Второй зарезервированный указатель, FAT[1], при форматировании устанавливается в значение метки EOC. В FAT12 он не используется больше никак, а в FAT16 и FAT32 верхние два бита этого указателя могут содержать отметку о необходимости проверки тома (так называемый «грязный бит»), причём все остальные биты выставлены в 1. Наличие грязного бита проверяется в процессе загрузки Windows программой autochk.exe. Грязный бит формируется при некорректном отключении тома или при аппаратной ошибке носителя и, соответственно, принимает два возможных значения.

Индексный указатель FAT32 по определению является 32-битным, однако верхние 4 бита в действительности игнорируются, так что значение указателя, по сути, является 28-битным. Единственной операцией, оперирующей с верхними 4 битами указателя, является форматирование тома, когда обнуляется весь указатель. Это означает, что, например, значения указателя 0x10000000, 0xF0000000 и 0x00000000 все соответствуют свободному кластеру, так как они отличаются лишь в верхних 4 битах.

Значение размера таблицы FAT по BPB, то есть BPB_FATSz16/32, может превышать реальное, так что в конце каждой таблицы FAT могут находиться сектора, не соответствующие никаким реальным кластерам данных. При форматировании эти сектора обнуляются, а в

процессе функционирования тома никак не используются. Поэтому действительный адрес последнего сектора таблицы FAT, содержащего указатели на реальные кластеры тома, всегда должен рассчитываться из общего количества кластеров области данных, а не из поля BPB_FATSz16/32. Кроме того, последний сектор, занятый таблицей FAT, вовсе не обязательно весь занят ею — в этом случае избыточное пространство сектора также не используется и забивается нулями при форматировании тома.

Файловые записи

Непосредственно после окончания последней таблицы FAT следует область данных, содержащая файлы и папки. Каталог FAT является обычным файлом, помеченным специальным атрибутом. Данными (содержимым) такого файла в любой версии FAT является цепочка 32-байтовых файловых записей (записей каталога). Каталог не может штатно содержать два файла с одинаковым именем. Если программа проверки диска обнаруживает искусственно созданную пару файлов с идентичным именем в одном каталоге, один из них переименовывается.

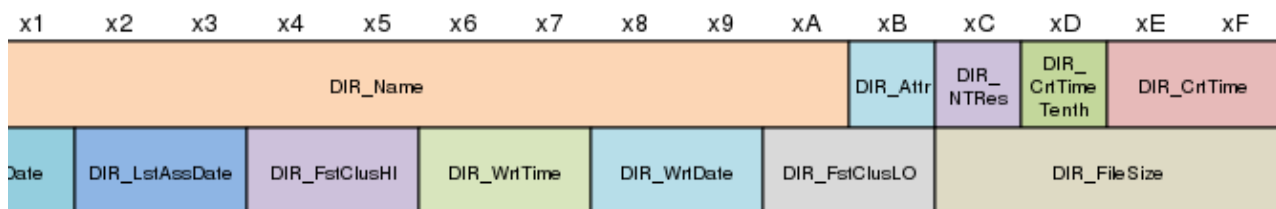
Корневой каталог

Единственным обязательно присутствующим каталогом является корневой каталог. В FAT12/FAT16 корневой каталог имеет фиксированный размер в секторах, который вычисляется из значения BPB_RootEntCnt, и следует на диске непосредственно после таблицы FAT.

В FAT32 корневой каталог, как и любой другой, имеет переменный размер и является цепочкой кластеров. Номер первого кластера корневого каталога отражается BPB_RootClus. Корневой каталог имеет следующие отличия от других каталогов тома FAT:

- у него нет меток даты и времени;
- нет собственного имени (кроме «\»);
- он не содержит файлов с именами «.» и «..» (см. далее);
- является единственным каталогом, в котором может штатно располагаться файл метки тома (см. далее).

Структура файловой записи



ение полей файловой записи

Файловая запись FAT32 состоит из следующих структур:

- DIR_Name. 11-байтовое поле по относительному адресу 0, содержит короткое имя файла (в рамках стандарта 8.3). По поводу имён файлов

см. далее.

- DIR_Attr. Байт по адресу 0x0B, отвечающий за атрибуты файла.
- DIR_NTRes. Байт по адресу 0x0C, используется в Windows NT.
- DIR_CrtTimeTenth. Байт по адресу 0x0D. Счётчик десятков миллисекунд времени создания файла, допустимы значения 0–199. Поле часто неоправданно игнорируется.
- DIR_CrtTime. 2 байта по адресу 0x0E. Время создания файла с точностью до 2 секунд.
- DIR_CrtDate. 2 байта по адресу 0x10. Дата создания файла.
- DIR_LstAccDate. 2 байта по адресу 0x12. Дата последнего доступа к файлу (то есть последнего чтения или записи — в последнем случае приравнивается DIR_WrtDate). Аналогичное поле для времени не предусмотрено.
- DIR_FstClusHI. 2 байта по адресу 0x14. Номер первого кластера файла (старшее слово, на томе FAT12/FAT16 равен нулю).
- DIR_WrtTime. 2 байта по адресу 0x16. Время последней записи (модификации) файла, например его создания.
- DIR_WrtDate. 2 байта по адресу 0x18. Дата последней записи (модификации) файла, в том числе создания.
- DIR_FstClusLO. 2 байта по адресу 0x1A. Номер первого кластера файла (младшее слово).
- DIR_FileSize. DWORD, содержащий значение размера файла в байтах. Фундаментальное ограничение FAT32 — максимально допустимое значение размера файла составляет 0xFFFFFFFF (то есть 4 Гбайт минус 1 байт).

Если первый байт записи FAT (то есть DIR_Name[0]) содержит 0xE5 или 0x05, это значит, что запись свободна (соответствующий файл был удалён). Ноль в DIR_Name[0] означает, что свободна не только эта запись, но и все следующие записи каталога; Windows не анализирует остаток каталога после обнулённой записи.

Имя файла в FAT

Поле DIR_Name логически разбивается на первые 8 символов, образующие имя файла, и последние 3, образующие расширение. Точка-разделитель добавляется на уровне операционной системы и не хранится в поле имени. Если имя и расширение файла не заполняют отведённое для них место, остальные байты поля DIR_Name заполняются пробелами (0x20). Имя и расширение файла могут содержать любую комбинацию букв, цифр или символов с ASCII-кодами выше 127; специальные символы распределяются на три группы:

- Разрешённые: ! # \$ % & () - @ ^ _ ` { } ~ ' "
- Запрещённые: + . ; = []
- Служебные: * ? < : > / \ | " "

Служебные символы имеют особое значение в DOS и Windows и не могут входить в состав имени файла (знаки * ? являются метасимволами, а знаки : / \ используются как разделители

в путях к файлам, остальные служебные и запрещённые знаки являются управляющими в интерпретаторах командной строки `COMMAND.COM` и `cmd.exe`), в то время как символы из числа запрещённых всё же можно включить в имя файла ценой возникновения LFN-записи (см. ниже). Например, каталог с именем, начинающимся точкой или содержащим несколько точек, можно создать в режиме командной строки (`mkdir .directory`) или в оболочках вроде FAR Manager, Total Commander, WinRAR. Имя файла не может начинаться или заканчиваться пробелом; ни в каком байте поля имени недопустимы управляющие символы ASCII (то есть 0x00-0x1F), за исключением оговорённого выше случая кода 5. Информация о текущей (на момент создания файла) кодовой странице DOS не сохраняется, поэтому доступ к файлам, в именах которых есть национальные коды из Extended ASCII (например знаки кириллицы из кодовой страницы 866), при другой кодовой странице может быть проблематичным или невозможным (поскольку перед поиском файла в каталоге его имя переводится в верхний регистр в соответствии с таблицей, заложенной в кодовой странице). Полный путь к файлу не может превышать 80 байт (3 — буква диска; 64 — путь; 12 — имя файла, включая точку-разделитель; 1 — терминальный нулевой знак).

Все буквенные символы 8.3 имени всегда переводятся и сохраняются в поле DIR_Name в верхнем регистре. Для сохранения исходного регистра имени Windows NT используется байт DIR_NTRes: 1 в бите 3 говорит, что имя следует отображать в нижнем регистре; за расширение отвечает бит 4. Если имя или расширение содержат символы обоих регистров, для такого файла создаётся LFN-запись (см. ниже). Windows 9x для сохранения нетривиального регистра имени всегда создаёт LFN-запись и игнорирует поле DIR_NTRes. Как следствие, имя одного и того же файла, лишённого ассоциированной LFN-записи, может отображаться Windows 9x целиком в верхнем регистре, а Windows NT — (частично) в нижнем.

Файловые атрибуты

В байте атрибутов верхние два бита являются резервными и всегда должны быть обнулены. Остальные биты распределяются таким образом, что значение 0x01 соответствует атрибуту «только для чтения», 0x02 — «скрытый», 0x04 — «системный», 0x20 — «архивный». Набор нескольких атрибутов составляется суммированием основных значений. Кроме этих стандартных атрибутов используются следующие: 0x10 — свидетельствует, что файл является каталогом (контейнером для других файлов); 0x08 — ATTR_VOLUME_ID, специальный атрибут уникального файла нулевого размера в корневом каталоге, имя которого считается меткой тома. Ограничение длины метки тома FAT в 11 символов связано с размером поля DIR_Name. Если файл имеет набор атрибутов READ_ONLY | HIDDEN | SYSTEM | VOLUME_ID (значение 0x0F), это свидетельствует, что запись не соответствует отдельному файлу, а содержит часть длинного имени другого файла, не вписывающегося в рамки 8.3 (см. далее).

Искусственное присвоение ненулевого значения верхним двум битам DIR_Attr используется для формирования файлов, которые невозможно удалить или переименовать штатными средствами файловой системы без форматирования. Это полезно, например, при борьбе с вирусами Autorun.inf (программа Panda USB and AutoRun Vaccine). С другой стороны, это же средство могут использовать сами вирусы. Значение DIR_Attr = 0x40 резервировано для внутреннего использования (устройство).

Что происходит при создании каталога

При создании каталога для него «пожизненно» выставляется DIR_FileSize = 0. Размер

содержимого каталога определяется простым следованием по цепочкам кластеров до метки End Of Chain. Размер самого каталога лимитируется файловой системой в 65 535 32-байтовых записей (то есть записи каталога в таблице FAT не могут занимать более 2 Мбайт). Это ограничение призвано ускорить операции с файлами и позволить различным служебным программам использовать 16-битное целое (WORD) для подсчёта количества записей в каталоге (как следствие, возникает теоретическое ограничение на количество файлов в каталоге — 65 535 при условии, что все имена файлов следуют стандарту 8.3). Каталог отводится один кластер области данных (за исключением случая, если это корневой каталог FAT12/FAT16), и поля DIR_FstClusHI / DIR_FstClusLO присваивается значение номера этого кластера. В таблицу FAT для записи, соответствующей этому кластеру, помещается метка ЕОС, а сам кластер забивается нулями. Далее создаются два специальных файла, без которых каталог FAT считается повреждённым (первые две 32-байтовых записи в области данных кластера) — файлы нулевого размера с именами «.» (одна точка, идентификатор каталога) и «..» (две точки, указатель на родительский каталог). Отметки даты и времени этих файлов приравниваются значениям для самого каталога на момент создания и не обновляются при изменениях каталога. Поля DIR_FstClusHI / DIR_FstClusLO файла «.» содержат значение номера содержащего его кластера, а файла «..» — номера первого кластера каталога, содержащего данный. Таким образом, файл «.» отсылает к самому каталогу, а файл «..» — к начальному кластеру родительского каталога; если родительский каталог — корневой, начальным кластером считается нулевой.

Время и дата

Двухбайтовая отметка даты имеет следующий формат:

- биты 0–4 — день месяца, допускаются значения 1–31;
- биты 5–8 — месяц года, допускаются значения 1–12;
- биты 9–15 — год, считая от 1980 года («эпоха MS-DOS»), возможны значения от 0 до 127 включительно, то есть 1980–2107 годы.

Двухбайтовая отметка времени имеет следующий формат:

- биты 0–4 — счётчик секунд (по две), допустимы значения 0–29, то есть 0–58 секунд;
- биты 5–10 — минуты, допустимы значения 0–59;
- биты 11–15 — часы, допустимы значения 0–23.

Из отметок даты и времени критическим является лишь время последней модификации (то есть DIR_WrtTime и DIR_WrtDate), остальные могут не поддерживаться многими системами; при операциях с файлом в такой системе (например в DOS или Windows 3.1) эти поля игнорируются. FAT сохраняет отметки даты и времени по местному часовому поясу, при его смене отметки не меняются.

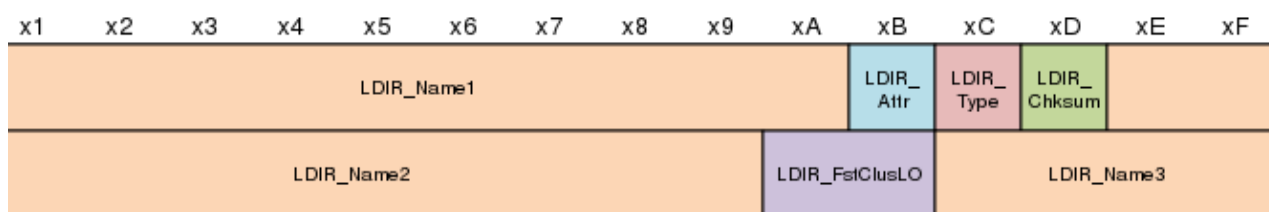
Временные отметки каталогов задаются при создании и не изменяются при записи новых файлов в каталог, переименовании или выделении ему нового кластера.

Дата последнего доступа к файлу обновляется при каждом обращении, например при просмотре свойств файла, при перемещении на другой том (но не в пределах тома). При копировании файла в Windows 98 дата последнего доступа исходного файла обновляется, в Windows XP — нет.

Дата-время модификации файла изменяется при каждой записи нового содержимого в области данных (не в файловой записи). Другими словами, дата-время модификации не изменяется при смене атрибутов или переименовании файла. Перемещение или копирование файла сохраняет исходную отметку модификации.

Дата и время создания задаётся при выделении файловой записи для нового, не существовавшего до этого файла. Иначе говоря, при переименовании или перемещении файла дата и время создания не изменяются, а при копировании новый файл получает новую отметку. Таким образом, при копировании файла в Windows он может получить более позднюю дату создания, чем дату модификации.

LFN-записи



LFN-записи

Файлы и каталоги с длинным именем (свыше 8.3) обрабатываются файловой системой FAT особым образом. Структура 32-байтовой записи для файла с LFN (Long File Name) отличается от обычной (SFN-записи):

- LDIR_Ord. Первый байт записи служит для нумерации записей в наборе.
- LDIR_Name1. 10-байтовое поле по адресу 0x01 содержит первые пять символов имени файла (вернее, той части его имени, которая отражена в данной LFN-записи).
- LDIR_Attr. Байт атрибута по адресу 0x0B, равен 0x0F (ATTR_LONG_NAME).
- LDIR_Type. Байт по адресу 0x0C, обнулён и дополнительно свидетельствует, что данная запись таблицы FAT относится к файлу с длинным именем.
- LDIR_Chksum. Байт по адресу 0x0D, содержит контрольную сумму SFN псевдонима файла, соответствующего набору LFN-записей.
- LDIR_Name2. 12-байтовое поле по адресу 0x0E, содержащее с 6-го по 11-й символы имени файла.
- LDIR_FstClusLO. 2-байтовое поле по адресу 0x1A, в контексте LFN-записи лишено смысла и обнуляется.
- LDIR_Name3. 4-байтовое поле по адресу 0x1C, содержащее 12-й и 13-й символы имени файла.

Набор LFN-записей каталога FAT всегда должен быть связан с обычной SFN-записью, которой физически предшествует на диске. Набор LFN-записей, обнаруженный без соответствующей обычной записи, называется орфаном, и запись считается повреждённой; подобный файл совершенно невидим в старых версиях MS-DOS/Windows.

В последовательности LFN-записей каждая из них имеет собственный порядковый номер, определяемый первым байтом (LDIR_Ord). Маска 0x40 свидетельствует, что данная запись

является крайней в ряду следующих за ней LFN-записей (то есть, например, для третьей в ряду LFN-записи значение байта LDIR_Ord будет 0x43, для 17-й — 0x51). В последующих записях этот байт изменяется от N для N-ой «длинной» записи по счету от соответствующей обычной до 1 для ближайшей к обычной записи.

Длинные имена файлов хранятся в кодировке Юникод (UTF-16), при этом сохраняется вводимый регистр буквенных символов. Если некоторый символ имени в кодировке OEM или Юникоде не может быть превращён в символ кодовой страницы, он всегда отображается как символ подчёркивания «_», причём сохранённый на диске действительный символ не изменяется.

Байт контрольной суммы вычисляется по определённому алгоритму на основе 8.3-имени обычной записи (для файла с длинным именем «имя» из обычной записи называется псевдонимом — alias) и копируется во все соответствующие ей «длинные» записи. Если какое-либо из значений не согласуется с именем файла (например если файл был переименован под ранней версией MS-DOS/Windows), возникает орфан.

SFN-псевдоним файла с длинным именем состоит из основной части и, при необходимости, цифрового «хвоста». Если файл имеет расширение, его первые три символа сохраняются в псевдониме. Соответствующее имя образуется переводом в кодировку OEM символов длинного имени файла, причём все пробелы длинного имени игнорируются, а символы, непереводимые в OEM либо запрещённые в контексте короткого имени, заменяются подчёркиванием «_». Цифровой хвост «~n», где $n = 1 \div 999999$, добавляется к псевдониму, если первоначально полученный псевдоним конфликтовал с именем какого-либо файла в том же каталоге либо был длиннее, чем определяет стандарт 8.3, или если какой-либо символ при смене кодировки не нашёл OEM-аналога и был заменён подчёркиванием. Таким образом, образуются псевдонимы типа NEWFIL~1.DJV (LFN = New file for me.djvu). Схема образования псевдонима файла оптимизирована для скорости операций и поэтому малопредсказуема в подробностях.

Имя файла длиной, не кратной 13 символам, не заполняет поля имени LFN-записей таблицы FAT полностью. В таком случае имя файла искусственно оканчивается символом NUL (0x00), а избыточные байты забиваются единицами (то есть символами 0xFF).

Для длинных имён длина имени ограничена 255 символами, не считая разделителя NUL, а полный путь — 260 символами, включая NUL. В длинном имени также допускается использование шести специальных символов, запрещённых в коротких именах: +, ;, = []

При попытке создания в томе FAT32 файла или каталога с именем, содержащим такой символ, автоматически генерируется LFN-запись независимо от длины имени файла. Аналогичный процесс происходит при создании файла/папки с именем, содержащим не-ASCII-символы.

Возможна ситуация, когда файл метки тома физически не предшествует всем записям тома с длинными именами (когда том не имеет метки или метка была назначена после записи какого-либо файла с длинным именем). Тогда метка тома в FAT12/FAT16 будет отображаться неправильно, так как будет взята из ближайшей LFN-записи (поскольку она также имеет атрибут VOLUME_ID), и при попытке изменения метки тома в действительности будет происходить нарушение имени соответствующего файла. При удалении файла, имеющего ассоциированные LFN-записи, последние не затрагиваются и становятся орфаном. При дальнейшем создании нового файла упомянутый орфан может быть ошибочно ассоциирован с ним в случае совпадения контрольных сумм имён старого и нового файлов,

однако используемый алгоритм вычисления контрольной суммы (ASCII-код первого символа псевдонима файла циклически сдвигается на бит вправо и прибавляется код следующего символа и т. д.) делает такую вероятность ничтожно малой.

Смысл файловых операций в FAT

Форматирование тома — таблица индексных указателей обнуляется, за исключением первых трёх (FAT[0] и FAT[1], зарезервированы, а FAT[2] содержит запись, соответствующую файлу метки тома, либо, при отсутствии её — метку EOC) и записей повреждённых кластеров; записи корневого каталога обнуляются (за исключением файла метки тома, если она есть), в остальном область данных не затрагивается.

Удаление файла — первый символ файловой записи и всех ассоциированных LFN-записей заменяется кодом 0xE5; занимаемые файлом кластеры помечаются в таблице FAT как свободные, а в области данных не затрагиваются.

Создание файла или каталога командой «Создать» контекстного меню — создаётся файловая запись для нового «пустого» файла с именем по умолчанию (например «Новая папка») и размером, определяемым типом файла; сам файл, если имеет ненулевой размер (что верно для практически всех «пустых» файлов, кроме каталогов и текстовых документов) записывается в области данных в выделенные ему кластеры; в таблице FAT создаётся соответствующая кластерная цепочка. После присвоения файлу действительного имени (не по умолчанию) первоначально созданная файловая запись помечается как удалённая и создаётся новая.

Переименование файла — создаётся новая запись с обновлённым именем; старая запись помечается как удалённая.

Сохранение файла из приложения (не из командной строки) — создаётся запись, содержащая все поля, кроме размера и начального кластера файла; после завершения сохранения файла создаётся новая запись, содержащая все поля, а прежняя удаляется.

Копирование файла — в новом местоположении создаётся идентичная файловая запись (возможно, за исключением некоторых временных отметок, см. выше), файлу выделяется первый свободный кластер и содержимое файла копируется в новое место, причём происходит копирование текущего кластера, поиск следующего свободного и заполнение таблицы FAT.

Перемещение файла (между разными томами) — копирование с последующим удалением файла из исходного местоположения.

Перемещение файла (в пределах тома) — кластерная цепочка не затрагивается, файловая запись копируется без изменения в новый каталог, после чего удаляется из прежнего.

Поиск свободного кластера по таблице индексных указателей для выделения новому файлу начинается в общем случае не с начала области данных (то есть с кластера 2), а с последнего выделенного какому-либо файлу кластера, номер которого сохраняется в структуре FSInfo. Другими словами, если файлу 1 был отведён кластер 30, а файлу 2 — кластер 31, после чего файл 1 был удалён, то при создании нового файла 3 он, скорее всего, будет физически размещён начиная с кластера 32.

Отказоустойчивость системы

Поскольку система FAT хранит данные о файлах и данные о свободном месте на диске в одной таблице, то операция записи файла, традиционно состоящая из двух этапов (добавление занимаемого блока в перечень занятых и исключение этого же блока из списка свободных), происходит в FAT в одно действие. Благодаря этому система FAT обладает врождённой устойчивостью к сбоям, то есть сбой (например, питания) в момент выполнения операции чтения или записи в большинстве случаев не приведёт к разрушению файловой системы. Однако в данном случае речь идёт именно о целостности файловой системы, а не самих файлов.

Характеристики^[3]

	FAT12	FAT16	FAT32
Разработчик	Microsoft		
Полное название	File Allocation Table (Таблица размещения файлов)		
	(12-битная версия)	(16-битная версия)	(32-битная версия)
Представлена	1980 (Microsoft Disk BASIC)	август 1984 (MS-DOS 3.00, урезанная) — июль 1988, MS-DOS 4.0 ^[6]	Август 1996 (Windows 95 OSR 2)
Идентификатор тома	0x01 (MBR)	0x04, 0x06, 0x0E (MBR)	0x0B, 0x0C (MBR) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (GPT)
Структуры			
Содержимое каталога	Таблица		
Размещение файлов	Линейный список		
Сбойные блоки	Тегирование кластера		
Ограничения			
Размер файла	32 MiB	2 GiB	4 GiB
Количество кластеров	4084	65 524	268 435 445 (2 ²⁸ −12)
Длина имени файла	8.3 или 255 символов при использовании LFN		

Размер тома	32 <u>MiB</u>	2 <u>GiB</u> 4 <u>GiB</u> (64 Кб на кластер, поддерживается не везде)	2 <u>TiB</u> 8 <u>TiB</u> (32 Кбайт на сектор)
Возможности			
Сохраняемые даты	Создания, модификации, доступа		
Диапазон дат	1 января 1980 — 31 декабря 2107		
Дополнительные данные	Изначально не поддерживаются		
Атрибуты файлов	Только для чтения, скрытый, системный, метка тома, подкаталог, архивный		
Разграничение прав доступа	Нет		
Прозрачное сжатие	Отдельные утилиты (<u>Stacker</u> , <u>DoubleSpace</u> , <u>DriveSpace</u>)		
Прозрачное шифрование	Сторонние утилиты или клоны DOS		

Лицензирование

Некоторые алгоритмы работы с FAT и VFAT запатентованы компанией Microsoft.

В США при повторном рассмотрении^[*когда?*] было принято решение аннулировать некоторые из патентов, но потом его отменили.

В октябре 2006 года в Германии был аннулирован за очевидность патент на VFAT, выданный Европейским патентным бюро^[7].

Со временем FAT стали широко использовать в различных устройствах для совместимости между DOS, Windows, OS/2, Linux. Майкрософт не выказывала намерений принуждать к их лицензированию^[*уточнить*]^[8].

В феврале 2009 года Microsoft подала в суд на компанию TomTom, производителя автомобильных навигационных систем на основе Linux, обвиняя её в нарушении патентов^[9].

По мнению Джереми Эллисона^[*уточнить*], цель Microsoft — поставить различные компании перед выбором: заключить с Microsoft договор о патентной защите (такой, который с ней заключила Novell в ноябре 2006 года), нарушив тем самым лицензию GNU GPL, и сделав невозможным для себя использование Linux, или не заключать такого договора и быть обвинённой в нарушении патентов, защита по которым предоставляется при его заключении при условии неразглашения^{[10][11]}.

В марте 2009 года TomTom подала встречный иск о нарушении патентов^[12].

См. также

- [exFAT](#)

Примечания

1. http://cd.textfiles.com/megademo2/INFO/OS2_HPFS.TXT
2. www.microsoft.com/mscorp/ip/tech/fathist.asp (http://web.archive.org/web/*/www.microsoft.com/mscorp/ip/tech/fathist.asp) в archive.org
3. Microsoft Extensible Firmware Initiative FAT32 File System Specification 1.03 (<http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc>) (недоступная ссылка). Microsoft (6 декабря 2000). — Документа в формате Microsoft Word, 268 Кб. Дата обращения: 5 апреля 2010. Архивировано (<https://www.webcitation.org/6181vmpl3?url=http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc>) 22 августа 2011 года.
4. What About VFAT? (<http://technet.microsoft.com/en-us/library/cc750354.aspx#XSLTsection125121120120>) (недоступная ссылка). *TechNet Archive*. Microsoft (15 октября 1999). Дата обращения: 5 апреля 2010. Архивировано (<https://www.webcitation.org/6181uHOVV?url=http://technet.microsoft.com/en-us/library/cc750354.aspx#XSLTsection125121120120>) 22 августа 2011 года.
5. Не следует путать расширение файловой системы VFAT с одноимённым драйвером файловой системы, который появился в Windows for Workgroups 3.11 и предназначен для обработки вызовов функций MS-DOS (INT 21h) в защищённом режиме (см.: [KB126746: Windows for Workgroups Version History](http://support.microsoft.com/kb/126746) (<http://support.microsoft.com/kb/126746>) (недоступная ссылка). *VERSION 3.11 → Non-Network Features*. Microsoft (14 ноября 2003). Дата обращения: 5 апреля 2010. Архивировано (<https://www.webcitation.org/6181uptwO?url=http://support.microsoft.com/kb/126746>) 22 августа 2011 года.)
6. MS-DOS Partitioning Summary (<http://support.microsoft.com/kb/q69912>) (недоступная ссылка). *microsoft.com*. Дата обращения: 23 октября 2012. Архивировано (<https://web.archive.org/web/20121023143346/http://support.microsoft.com/kb/q69912>) 23 октября 2012 года.
7. Federal Patent Court declares FAT patent of Microsoft null and void (<http://www.heise.de/english/newsticker/news/86141>) (англ.) (недоступная ссылка). *heise online*. Heise Zeitschriften Verlag (2 марта 2007). Дата обращения: 10 марта 2009. Архивировано (<https://www.webcitation.org/6181wE3ui?url=http://www.h-online.com/>) 22 августа 2011 года.
8. *Brian Kahin*. Microsoft Roils the World with FAT Patents (http://www.huffingtonpost.com/brian-kahin/microsoft-roils-the-world_b_171985.html) (англ.) (недоступная ссылка). *The Huffington Post* (10 марта 2009). Дата обращения: 10 марта 2009. Архивировано (https://www.webcitation.org/6181yXtEr?url=http://www.huffingtonpost.com/brian-kahin/microsoft-roils-the-world_b_171985.html) 22 августа 2011 года.

9. *Ryan Paul*. Microsoft suit over FAT patents could open OSS Pandora's Box (<http://arstechnica.com/microsoft/news/2009/02/microsoft-sues-tomtom-over-fat-patents-in-linux-based-device.ars>) (англ.) (недоступная ссылка). *Ars Technica*. Condé Nast Publications (25 февраля 2009). Дата обращения: 9 марта 2009. Архивировано (<https://www.webcitation.org/6181zQldD?url=http://arstechnica.com/microsoft/news/2009/02/microsoft-sues-tomtom-over-fat-patents-in-linux-based-device.ars>) 22 августа 2011 года.
10. *Glyn Moody*. The Real Reason for Microsoft's TomTom Lawsuit (<http://www.computerworlduk.com/toolbox/open-source/blogs/index.cfm?entryid=1953&blogid=14>) (англ.) (недоступная ссылка). *ComputerworldUK*. IDG (5 марта 2009). Дата обращения: 9 марта 2009. Архивировано (<https://www.webcitation.org/6181zvabs?url=http://blogs.computerworlduk.com/open-enterprise/2009/03/the-real-reason-for-microsofts-tomtom-lawsuit/index.htm>) 22 августа 2011 года.
11. *Steven J. Vaughan-Nichols*. Linux companies sign Microsoft patent protection pacts (http://blogs.computerworld.com/linux_companies_sign_microsoft_patent_protection_pacts) (англ.) (недоступная ссылка). *Блоги Computerworld*. IDG (5 марта 2009). Дата обращения: 9 марта 2009. Архивировано (https://www.webcitation.org/618211DQy?url=http://blogs.computerworld.com/linux_companies_sign_microsoft_patent_protection_pacts) 22 августа 2011 года.
12. *Erica Ogg*. TomTom countersues Microsoft in patent dispute (http://news.cnet.com/8301-10805_3-10200526-75.html) (англ.) (недоступная ссылка). *CNet* (19 марта 2009). Дата обращения: 20 марта 2009. Архивировано (https://www.webcitation.org/61821fwRj?url=http://news.cnet.com/8301-10805_3-10200526-75.html) 22 августа 2011 года.

Ссылки

- ECMA-107 (<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-107.pdf>) (англ.) стандарт FAT
 - работа с FAT средствами WinHEX (<http://litl-admin.ru/content/detail.php?ID=147>)
-

Источник — <https://ru.wikipedia.org/w/index.php?title=FAT&oldid=116296847>

Эта страница в последний раз была отредактирована 26 августа 2021 в 16:28.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.