

Файловая система в Операционных системах персональных компьютеров

Одной из главных задач ОС является обеспечение обмена данными между приложениями и периферийными устройствами компьютера. Собственно ради выполнения этой задачи и были разработаны первые системные программы, послужившие прототипами операционных систем. В современной ОС функции обмена данными с периферийными устройствами выполняет подсистема ввода-вывода. Клиентами этой подсистемы являются не только пользователи и приложения, но и некоторые компоненты самой ОС, которым требуется получение системных данных или их вывод, например подсистеме управления процессами при смене активного процесса необходимо записать на диск контекст приостанавливаемого процесса и считать с диска контекст активизируемого процесса. Основными компонентами подсистемы ввода-вывода являются драйверы, управляющие внешними устройствами, и файловая система.

Задачи ОС по управлению файлами и устройствами

Подсистема ввода-вывода (Input-Output Subsystem) мультипрограммной ОС при обмене данными с внешними устройствами компьютера должна решать ряд общих задач, из которых наиболее важными являются следующие:

- организация параллельной работы устройств ввода-вывода и процессора;
- согласование скоростей обмена и кэширование данных;
- разделение устройств и данных между процессами;
- обеспечение удобного логического интерфейса между устройствами и остальной частью системы;
- поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера;
- динамическая загрузка и выгрузка драйверов;
- поддержка нескольких файловых систем;
- поддержка синхронных и асинхронных операций ввода-вывода.

Поддержка нескольких файловых систем

Диски представляют особый род периферийных устройств, так как именно на них хранится большая часть как пользовательских, так и системных данных. Данные на дисках организуются в файловые системы, и свойства файловой системы во многом определяют свойства самой ОС – ее отказоустойчивость, быстродействие, максимальный объем хранимых данных. Популярность файловой системы часто приводит к ее миграции из «родной» ОС в другие операционные системы – например, файловая система FAT появилась первоначально в MS-DOS, но затем была

реализована в OS/2, семействе MS Windows и многих реализациях UNIX. Ввиду этого поддержка нескольких популярных файловых систем для подсистемы ввода-вывода также важна, как и поддержка широкого спектра периферийных устройств. Важно также, чтобы архитектура подсистемы ввода-вывода позволяла достаточно просто включать в ее состав новые типы файловых систем, без необходимости переписывания кода.

Логическая организация файловой системы

Одной из основных задач операционной системы является предоставление удобств пользователю при работе с данными, хранящимися на дисках. Для этого ОС подменяет физическую структуру хранящихся данных некоторой удобной для пользователя логической моделью. Логическая модель файловой системы материализуется в виде дерева каталогов, выводимого на экран такими утилитами, как Total Commander или Windows Explorer, в символьных составных именах файлов, в командах работы с файлами. Базовым элементом этой модели является файл, который так же, как и файловая система в целом, может характеризоваться как логической, так и физической структурой.

Цели и задачи файловой системы

Файл – это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Файлы хранятся в памяти, не зависящей от энергопитания. Однако нет правил без исключения. Одним из таких исключений является так называемый электронный диск, когда в оперативной памяти создается структура, имитирующая файловую систему.

Основные цели использования файла перечислены ниже.

Долговременное и надежное хранение информации. Долговременность достигается за счет использования запоминающих устройств, не зависящих от питания, а высокая надежность определяется средствами защиты доступа к файлам и общей организацией программного кода ОС, при которой сбои аппаратуры чаще всего не разрушают информацию, хранящуюся в файлах.

Совместное использование информации. Файлы обеспечивают естественный и легкий способ разделения информации между приложениями и пользователями за счет наличия понятного человеку символьного имени и постоянства хранимой информации и расположения файла. Пользователь должен иметь удобные средства работы с файлами, включая каталоги-справочники, объединяющие файлы в группы, средства поиска файлов по признакам, набор команд для создания, модификации и удаления файлов. Файл может быть создан одним пользователем, а затем использоваться совсем другим пользователем, при этом создатель файла или администратор могут определить

права доступа к нему других пользователей. Эти цели реализуются в ОС файловой системой.

Файловая система (ФС) – это часть операционной системы, включающая:

- совокупность всех файлов на диске;
- наборы структур данных, используемых для управления файлами, такие, например, как каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске;
- комплекс системных программных средств, реализующих различные операции над файлами, такие как создание, уничтожение, чтение, запись, именование и поиск файлов.

Файловая система позволяет программам обходиться набором достаточно простых операций для выполнения действий над некоторым абстрактным объектом, представляющим файл. При этом программистам не нужно иметь дело с деталями действительного расположения данных на диске, буферизацией данных и другими низкоуровневыми проблемами передачи данных с долговременного запоминающего устройства. Все эти функции файловая система берет на себя. Файловая система распределяет дисковую память, поддерживает именование файлов, отображает имена файлов в соответствующие адреса во внешней памяти, обеспечивает доступ к данным, поддерживает разделение, защиту и восстановление файлов.

Таким образом, файловая система играет роль промежуточного слоя, экранирующего все сложности физической организации долговременного хранилища данных, и создающего для программ более простую логическую модель этого хранилища, а также предоставляя им набор удобных в использовании команд для манипулирования файлами.

Задачи, решаемые ФС, зависят от способа организации вычислительного процесса в целом. Самый простой тип – это ФС в однопользовательских и однопрограммных ОС, к числу которых относится, например, MS-DOS. Основные функции в такой ФС нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- устойчивость файловой системы к сбоям питания, ошибкам аппаратных и программных средств.

Задачи ФС усложняются в операционных однопользовательских мультипрограммных ОС, которые, хотя и предназначены для работы одного пользователя, но дают ему возможность запускать одновременно несколько процессов. К перечисленным выше задачам добавляется новая задача совместного доступа к файлу из нескольких процессов. Файл в этом случае является разделяемым ресурсом, а значит, файловая система должна решать весь комплекс проблем, связанных с такими

ресурсами. В частности, в ФС должны быть предусмотрены средства блокировки файла и его частей, предотвращения гонок, исключение тупиков, согласование копий и т. п.

В многопользовательских системах появляется еще одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя. Еще более сложными становятся функции ФС, которая работает в составе сетевой ОС.

Типы файлов

Файловые системы поддерживают несколько функционально различных типов файлов, в число которых, как правило, входят обычные файлы, файлы-каталоги, специальные файлы, именованные конвейеры, отображаемые в память файлы и другие.

Обычные файлы, или просто *файлы*, содержат информацию произвольного характера, которую заносит в них пользователь или которая образуется в результате работы системных и пользовательских программ. Большинство современных операционных систем (например, UNIX, Windows) никак не ограничивает и не контролирует содержимое и структуру обычного файла. Содержание обычного файла определяется приложением, которое с ним работает. Например, текстовый редактор создает текстовые файлы, состоящие из строк символов, представленных в каком-либо коде. Это могут быть документы, исходные тексты программ и т. п. Текстовые файлы можно прочесть на экране и распечатать на принтере. Двоичные файлы не используют коды символов, они часто имеют сложную внутреннюю структуру, например исполняемый код программы или архивный файл. Все операционные системы должны уметь распознавать хотя бы один тип файлов – их собственные исполняемые файлы.

Каталоги – это особый тип файлов, которые содержат системную справочную информацию о наборе файлов, сгруппированных пользователями по какому-либо неформальному признаку (например, в одну группу объединяются файлы, содержащие документы одного договора, или файлы, составляющие один программный пакет). Во многих операционных системах в каталог могут входить файлы любых типов, в том числе другие каталоги, за счет чего образуется древовидная структура, удобная для поиска. Каталоги устанавливают соответствие между именами файлов и их характеристиками, используемыми файловой системой для управления файлами. В число таких характеристик входит, в частности, информация (или указатель на другую структуру, содержащую эти данные) о типе файла и расположении его на диске, правах доступа к файлу и датах его создания и модификации. Во всех остальных отношениях каталоги рассматриваются файловой системой как обычные файлы.

Специальные файлы – это фиктивные файлы, ассоциированные с устройствами ввода-вывода, которые используются для унификации механизма доступа к файлам и внешним устройствам. Специальные файлы позволяют пользователю выполнять операции ввода-вывода посредством обычных команд записи в файл или чтения из файла. Эти команды обрабатываются сначала программами файловой системы, а затем

на некотором этапе выполнения запроса преобразуются операционной системой в команды управления соответствующим устройством.

Современные файловые системы поддерживают и другие типы файлов, такие как символьные связи, именованные конвейеры, отображаемые в память файлы.

Физическая организация файловой системы

Представление пользователя о файловой системе как об иерархически организованном множестве информационных объектов имеет мало общего с порядком хранения файлов на диске. Файл, имеющий образ цельного, непрерывающегося набора байт, на самом деле очень часто разбросан «кусочками» по всему диску, причем это разбиение никак не связано с логической структурой файла, например, его отдельная логическая запись может быть расположена в несмежных секторах диска. Логически объединенные файлы из одного каталога совсем не обязаны соседствовать на диске. Принципы размещения файлов, каталогов и системной информации на реальном устройстве описываются *физической организацией файловой системы*. Очевидно, что разные файловые системы имеют разную физическую организацию.

Диски, разделы, секторы, кластеры

Основным типом устройства, которое используется в современных вычислительных системах для хранения файлов, являются дисковые накопители.

Сектор – наименьшая адресуемая единица обмена данными дискового устройства с оперативной памятью. Для того чтобы контроллер мог найти на диске нужный сектор, необходимо задать ему все составляющие адреса сектора: номер цилиндра, номер поверхности и номер сектора. Так как прикладной программе в общем случае нужен не сектор, а некоторое количество байт, не обязательно кратное размеру сектора, то типичный запрос включает чтение нескольких секторов, содержащих требуемую информацию, и одного или двух секторов, содержащих наряду с требуемыми избыточные данные.

Операционная система при работе с диском использует, как правило, собственную единицу дискового пространства, называемую *кластером (cluster)*. При создании файла место на диске ему выделяется кластерами. Например, если файл имеет размер 2560 байт, а размер кластера в файловой системе определен в 1024 байта, то файлу будет выделено на диске 3 кластера.

Дорожки и секторы создаются в результате выполнения процедуры *физического*, или *низкоуровневого, форматирования диска*, предшествующей использованию диска. Для определения границ блоков на диск записывается идентификационная информация. Низкоуровневый формат диска не зависит от типа операционной системы, которая этот диск будет использовать.

Разметку диска под конкретный тип файловой системы выполняют процедуры *высокоуровневого*, или *логического*, *форматирования*. При высокоуровневом форматировании определяется размер кластера и на диск записывается информация, необходимая для работы файловой системы, в том числе информация о доступном и неиспользуемом пространстве, о границах областей, отведенных под файлы и каталоги, информация о поврежденных областях. Кроме того, на диск записывается загрузчик операционной системы – небольшая программа, которая начинает процесс инициализации операционной системы после включения питания или рестарта компьютера.

Прежде чем форматировать диск под определенную файловую систему, он может быть разбит на разделы. *Раздел* – это непрерывная часть физического диска, которую операционная система представляет пользователю как *логическое устройство* (используются также названия *логический диск* и *логический раздел*). Логическое устройство функционирует так, как если бы это был отдельный физический диск. Именно с логическими устройствами работает пользователь, обращаясь к ним по символьным именам, используя, например, обозначения А, В, С, SYS и т. п. Операционные системы разного типа используют единое для всех них представление о разделах, но создают на его основе логические устройства, специфические для каждого типа ОС. Так же как файловая система, с которой работает одна ОС, в общем случае не может интерпретироваться ОС другого типа, логические устройства не могут быть использованы операционными системами разного типа. На каждом логическом устройстве может создаваться только одна файловая система.

В частном случае, когда все дисковое пространство охватывается одним разделом, логическое устройство представляет физическое устройство в целом. Если диск разбит на несколько разделов, то для каждого из этих разделов может быть создано отдельное логическое устройство. Логическое устройство может быть создано и на базе нескольких разделов, причем эти разделы не обязательно должны принадлежать одному физическому устройству. Объединение нескольких разделов в единое логическое устройство может выполняться разными способами и преследовать разные цели, основные из которых: увеличение общего объема логического раздела, повышение производительности и отказоустойчивости. Примерами организации совместной работы нескольких дисковых разделов являются так называемые RAID-массивы, подробнее о которых будет сказано далее.

На разных логических устройствах одного и того же физического диска могут располагаться файловые системы разного типа. На рисунке 1 показан пример диска, разбитого на три раздела, в которых установлены две файловых системы NTFS (разделы С и Е) и одна файловая система FAT (раздел D).

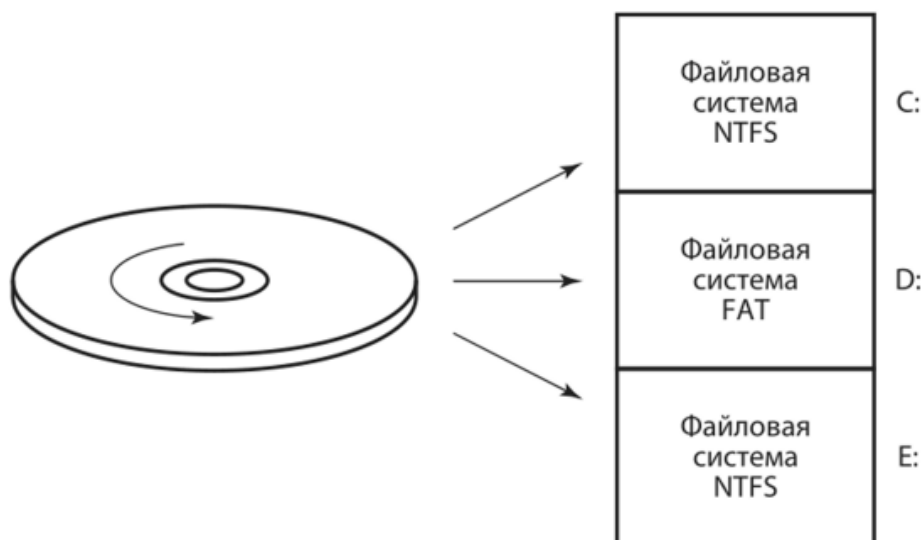


Рисунок 1 – Разбиение диска на разделы

Все разделы одного диска имеют одинаковый размер блока (сектора), определенный для данного диска в результате низкоуровневого форматирования. Однако в результате высокоуровневого форматирования в разных разделах одного и того же диска, представленных разными логическими устройствами, могут быть установлены файловые системы, в которых определены кластеры отличающихся размеров. Операционная система может поддерживать разные статусы разделов, особым образом отмечая разделы, которые могут быть использованы для загрузки модулей операционной системы, и разделы, в которых можно устанавливать только приложения и хранить файлы данных. Один из разделов диска помечается как загружаемый (или активный). Именно из этого раздела считывается загрузчик операционной системы.

Во многих операционных системах используется термин «том» (*volume*). В разных ОС толкование этого термина имеет свои нюансы, но чаще всего он обозначает логическое устройство, отформатированное под конкретную файловую систему.

Физическая организация и адресация файла

Важным компонентом физической организации файловой системы является *физическая организация файла*, то есть способ размещения файла на диске. Основными критериями эффективности физической организации файлов являются:

- скорость доступа к данным;
- объем адресной информации файла;
- степень фрагментированности дискового пространства;
- максимально возможный размер файла.

Непрерывное размещение – простейший вариант физической организации, при котором файлу предоставляется последовательность кластеров диска, образующих

непрерывный участок дисковой памяти. Основным достоинством этого метода является высокая скорость доступа, так как затраты на поиск и считывание кластеров файла минимальны. Также минимален объем адресной информации – достаточно хранить только номер первого кластера и объем файла. Данная физическая организация максимально возможный размер файла не ограничивает. Однако этот вариант имеет существенные недостатки, которые затрудняют его применимость на практике, несмотря на всю его логическую простоту. При более пристальном рассмотрении оказывается, что реализовать эту схему не так уж просто. Действительно, какого размера должна быть непрерывная область, выделяемая файлу, если файл при каждой модификации может увеличить свой размер? Еще более серьезной проблемой является фрагментация. Спустя некоторое время после создания файловой системы в результате выполнения многочисленных операций создания и удаления файлов пространство диска неминуемо превращается в «лоскутное одеяло», включающее большое число свободных областей небольшого размера. Как всегда бывает при фрагментации, суммарный объем свободной памяти может быть очень большим, а выбрать место для размещения файла целиком невозможно. Поэтому на практике используются методы, в которых файл размещается в нескольких, в общем случае несмежных областях диска.

Следующий способ физической организации – размещение файла в виде связанного списка кластеров дисковой памяти. При таком способе в начале каждого кластера содержится указатель на следующий кластер. В этом случае адресная информация минимальна: расположение файла может быть задано одним числом – номером первого кластера. В отличие от предыдущего способа каждый кластер может быть присоединен к цепочке кластеров какого-либо файла, следовательно, фрагментация на уровне кластеров отсутствует. Файл может изменять свой размер во время своего существования, наращивая число кластеров. Недостатком является сложность реализации доступа к произвольно заданному месту файла – чтобы прочитать пятый по порядку кластер файла, необходимо последовательно прочитать четыре первых кластера, проследив цепочку номеров кластеров. Кроме того, при этом способе количество данных файла, содержащихся в одном кластере, не равно степени двойки (одно слово израсходовано на номер следующего кластера), а многие программы читают данные кластерами, размер которых равен степени двойки.

Популярным способом, применяемым, например, в файловой системе FAT, является использование связанного списка индексов. Этот способ является некоторой модификацией предыдущего. Файлу также выделяется память в виде связанного списка кластеров. Номер первого кластера запоминается в записи каталога, где хранятся характеристики этого файла. Остальная адресная информация отделена от кластеров файла. С каждым кластером диска связывается некоторый элемент – индекс. Индексы располагаются в отдельной области диска – в MS-DOS к примеру это таблица FAT (*File Allocation Table*), занимающая один кластер. Когда память свободна, все индексы имеют нулевое значение. ***Если некоторый кластер N назначен некоторому файлу, то индекс этого кластера становится равным либо номеру M следующего***

кластера данного файла, либо принимает специальное значение» являющееся признаком того, что этот кластер является для файла последним. Индекс же предыдущего кластера файла принимает значение N, указывая на вновь назначенный кластер.

При такой физической организации сохраняются все достоинства предыдущего способа: минимальность адресной информации, отсутствие фрагментации, отсутствие проблем при изменении размера. Кроме того, данный способ обладает дополнительными преимуществами. Во-первых, для доступа к произвольному кластеру файла не требуется последовательно считывать его кластеры, достаточно прочитать только секторы диска, содержащие таблицу индексов, отсчитать нужное количество кластеров файла по цепочке и определить номер нужного кластера. Во-вторых, данные файла заполняют кластер целиком, а значит, имеют объем, равный степени двойки.

Еще один способ задания физического расположения файла заключается в простом перечислении номеров кластеров, занимаемых этим файлом. Этот перечень и служит адресом файла. Недостаток данного способа очевиден: длина адреса зависит от размера файла и для большого файла может составить значительную величину. Достоинством же является высокая скорость доступа к произвольному кластеру файла, так как здесь применяется прямая адресация, которая исключает просмотр цепочки указателей при поиске адреса произвольного кластера файла. Фрагментация на уровне кластеров в этом способе также отсутствует.

Для того чтобы корректно принимать решение о выделении файлу набора кластеров, файловая система должна отслеживать информацию о состоянии всех кластеров диска: свободен/занят. Эта информация может храниться как отдельно от адресной информации файлов, так и вместе с ней.

Физическая организация FAT

Логический раздел, отформатированный под файловую систему FAT, состоит из следующих областей (рисунок 2).

1. *Загрузочный сектор* содержит программу начальной загрузки операционной системы. Вид этой программы зависит от типа операционной системы, которая будет загружаться из этого раздела.
2. *Основная копия FAT* содержит информацию о размещении файлов и каталогов на диске.
3. *Резервная копия FAT*.
4. *Корневой каталог* занимает фиксированную область размером в 32 сектора (16 Кбайт), что позволяет хранить 512 записей о файлах и каталогах, так как каждая запись каталога состоит из 32 байт.
5. *Область данных* предназначена для размещения всех файлов и всех каталогов, кроме корневого каталога.



Рисунок 2 – Физическая структура файловой системы FAT

Файловая система FAT поддерживает всего два типа файлов: обычный файл и каталог. Файловая система распределяет память только из области данных, причем использует в качестве минимальной единицы дискового пространства кластер.

Структура записей каталога включает следующие поля:

1. Имя файла.
2. Расширение имени файла.
3. Свойства (атрибуты) файла:
 - 00: зарезервированные биты;
 - A: бит "архивный";
 - R: бит "только для чтения";
 - S: бит "системный";
 - D: бит "каталог";
 - V: бит "том".
4. Время и дата создания.
5. Первый кластер, в котором располагается начало файла.
6. Размер файла.

Таблица FAT (как основная копия, так и резервная) состоит из массива индексных указателей, количество которых равно количеству кластеров области данных. Между кластерами и индексными указателями имеется взаимно однозначное соответствие – нулевой указатель соответствует нулевому кластеру и т. д.

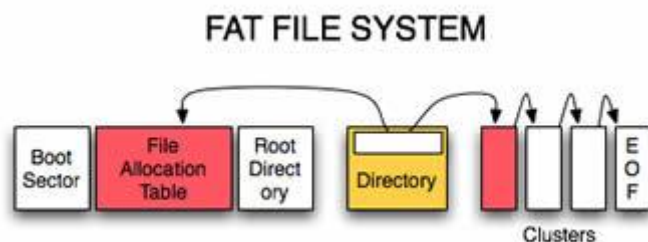


Рисунок 3 – Соответствие записей в FAT и содержимого кластеров

Индексный указатель может принимать следующие значения, характеризующие состояние связанного с ним кластера:

- кластер свободен (не используется, “0”);
- кластер используется файлом и не является последним кластером файла; в этом случае индексный указатель содержит номер следующего кластера файла;
- последний кластер файла (“EOF”);
- дефектный кластер;
- резервный кластер.

Таблица FAT является общей для всех файлов раздела. В исходном состоянии (после форматирования) все кластеры раздела свободны и все индексные указатели (кроме тех, которые соответствуют резервным и дефектным блокам) принимают значение «кластер свободен». При размещении файла ОС просматривает FAT, начиная с начала, и ищет первый свободный индексный указатель. После его обнаружения в поле записи каталога «номер первого кластера» фиксируется номер этого указателя. В кластер с этим номером записываются данные файла, он становится первым кластером файла. Если файл уместается в одном кластере, то в указатель, соответствующий данному кластеру, заносится специальное значение «последний кластер файла». Если же размер файла больше одного кластера, то ОС продолжает просмотр FAT и ищет следующий указатель на свободный кластер. После его обнаружения в предыдущий указатель заносится номер этого кластера, который теперь становится следующим кластером файла. Процесс повторяется до тех пор, пока не будут размещены все данные файла. Таким образом, создается связный список всех кластеров файла.



В начальный период после форматирования файлы будут размещаться в последовательных кластерах области данных, однако после определенного количества удалений файлов кластеры одного файла окажутся в произвольных местах области данных, чередуясь с кластерами других файлов.

Размер таблицы FAT и разрядность используемых в ней индексных указателей определяется количеством кластеров в области данных. Для уменьшения потерь из-за фрагментации желательно кластеры делать небольшими, а для сокращения объема адресной информации и повышения скорости обмена наоборот – чем больше, тем лучше. При форматировании диска под файловую систему FAT обычно выбирается компромиссное решение и размеры кластеров выбираются из диапазона от 1 до 128 секторов, или от 512 байт до 64 Кбайт.

Очевидно, что разрядность индексного указателя должна быть такой, чтобы в нем можно было задать максимальный номер кластера для диска определенного объема. Существует несколько разновидностей FAT, отличающихся разрядностью индексных указателей, которая и используется в качестве условного обозначения: FAT12, FAT16 и FAT32. В файловой системе FAT12 используются 12-разрядные указатели, что позволяет поддерживать до 4096 кластеров в области данных диска, в FAT16 – 16-разрядные указатели для 65 536 кластеров и в FAT32 – 32-разрядные для более чем 4 миллиардов кластеров. Реально это число немного меньше, так как несколько значений индексного указателя расходуется для идентификации специальных ситуаций, таких как «Последний кластер», «Неиспользуемый кластер», «Дефектный кластер» и «Резервный кластер».

Форматирование FAT 12 обычно характерно только для небольших дисков объемом не более 16 Мбайт, чтобы не использовать кластеры более 4 Кбайт. По этой же причине считается, что FAT16 целесообразнее для дисков с объемом не более 512 Мбайт, а для больших дисков лучше подходит FAT32, которая способна использовать кластеры 4 Кбайт при работе с дисками объемом до 8 Гбайт и только для дисков большего объема начинает использовать 8, 16 и 32 Кбайт. Максимальный размер раздела FAT16 ограничен 4 Гбайт, такой объем дает 65 536 кластеров по 64 Кбайт каждый, а максимальный размер раздела FAT32 практически не ограничен – 2^{32} кластеров по 32 Кбайт.

Таблица FAT при фиксированной разрядности индексных указателей имеет переменный размер, зависящий от объема области данных диска.

При удалении файла из файловой системы FAT в первый байт соответствующей записи каталога заносится специальный признак, свидетельствующий о том, что эта запись свободна, а во все индексные указатели файла заносится признак «кластер свободен». Остальные данные в записи каталога, в том числе номер первого кластера файла, остаются нетронутыми, что оставляет шансы для восстановления ошибочно удаленного файла. Существует большое количество утилит для восстановления удаленных файлов FAT, выводящих пользователю список имен удаленных файлов с отсутствующим первым символом имени, затертым после освобождения записи. Очевидно, что надежно можно восстановить только файлы, которые были расположены в последовательных кластерах диска, так как при отсутствии связанного списка выявить принадлежность произвольно расположенного кластера удаленному файлу невозможно (без анализа содержимого кластеров, выполняемого пользователем «вручную»).

Резервная копия FAT всегда синхронизируется с основной копией при любых операциях с файлами, поэтому резервную копию нельзя использовать для отмены ошибочных действий пользователя, выглядевших с точки зрения системы вполне корректными. Резервная копия может быть полезна только в том случае, когда секторы основной памяти оказываются физически поврежденными и не читаются.

Используемый в FAT метод хранения адресной информации о файлах не отличается большой надежностью – при разрыве списка индексных указателей в одном

месте, например из-за сбоя в работе программного кода ОС по причине внешних электромагнитных помех, теряется информация обо всех последующих кластерах файла.

Файловые системы FAT12 и FAT16 оперировали с именами файлов, состоящими из 12 символов по схеме «8.3». В версии FAT16 операционной системы Windows NT был введен новый тип записи каталога – «длинное имя», что позволяет использовать имена длиной до 255 символов, причем каждый символ длинного имени хранится в двухбайтном формате Unicode. Имя по схеме «8.3», названное теперь коротким (не нужно путать его с простым именем файла, также называемого иногда коротким), по-прежнему хранится в 12-байтовом поле имени файла в записи каталога, а длинное имя помещается порциями по 13 символов в одну или несколько записей, следующих непосредственно за основной записью каталога. Каждый символ в формате Unicode кодируется двумя байтами, поэтому 13 символов занимают 26 байт, а оставшиеся 6 отведены под служебную информацию. Таким образом, у файла появляются два имени – короткое, для совместимости со старыми приложениями, не понимающими длинных имен в Unicode, и длинное, удобное в использовании имя. Файловая система FAT32 также поддерживает короткие и длинные имена.

Краткие выводы

Файл – это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Основными целями использования файлов являются: долговременное и надежное хранение информации, а также совместный доступ к данным.

Файловая система представляет собой комплекс системных программных средств, реализующих различные операции с файлами, таких как создание, уничтожение, чтение, запись, именование и поиск файлов. Под файловой системой понимают также набор всех файлов и служебных структур данных, хранящихся на внешнем устройстве.

Кроме обычных файлов ОС, как правило, поддерживает такие типы файлов, как каталоги, символьные связи, именованные конвейеры и специальные файлы.

Специальный файл является универсальной моделью устройства ввода-вывода, представляя его для остальной части операционной системы и прикладных процессов в виде неструктурированного набора байт, то есть в виде обычного файла.

Современные файловые системы имеют иерархическую структуру, упрощающую именование файлов и их поиск.

Физическая организация файловой системы подразумевает способы размещения и адресации отдельных частей файлов в разделах и секторах дисковой памяти, а также способы организации служебной информации, описывающей размещение файлов и их атрибуты.