

**ГЛАВА 25**

# **Файловые системы и восстановление данных**



# Структуры диска FAT

Жесткий диск — основное устройство хранения данных. Он может иметь различную структуру хранения файлов и каталогов, которая обеспечивает непосредственное расположение данных на диске. Файловая система чаще всего интегрирована в операционную, а некоторые операционные системы поддерживают несколько файловых систем.

Большинство существующих на сегодняшний день файловых систем построены на основе *таблицы размещения файлов (File Allocation Table — FAT)*, которая содержит дорожки данных в каждом кластере на диске. Существует несколько типов файловой системы FAT — FAT 12, FAT 16 и FAT 32. Они отличаются количеством цифр, используемых в таблице размещения файлов. Другими словами, в FAT 32 используется 32-разрядное число для хранения дорожки данных в каждом кластере, в FAT 16 — 16-разрядное число и т.д. В настоящее время существуют следующие типы FAT:

- *FAT 12*, используемая в разделах емкостью не более 16 Мбайт (например, дискета);
- *FAT 16*, используемая в разделах емкостью от 16 Мбайт до 2 Гбайт; операционные системы Windows NT, Windows 2000 и Windows XP поддерживают разделы FAT 16 емкостью до 4 Гбайт;
- *FAT 32*, используемая (необязательно) в разделах емкостью от 512 Мбайт до 2 Тбайт.

Файловые системы FAT 12 и FAT 16 изначально применяются в DOS и Windows и поддерживаются практически всеми известными на сегодняшний день операционными системами. Большинство персональных компьютеров поставляется с жесткими дисками, на которых установлена одна из файловых систем FAT.

Файловая система FAT 32 поддерживается операционной системой Windows 95B и более поздними версиями, а также Windows 2000, которая также поддерживает файловую систему NTFS. Некоторые операционные системы имеют собственную файловую систему. Например, Windows NT и Windows 2000 поддерживают файловую систему NT File System (NTFS); OS/2 поставляется с собственной файловой системой High Performance File System (HPFS).

В этой главе речь идет о файловой системе FAT, а также рассматриваются возможности новой системы FAT 32 и NTFS.

Для обеспечения пользовательским приложениям доступа к файлам независимо от типа используемого диска в операционной системе предусмотрено несколько структур. Эти структуры поддерживаются системами Windows 9x, Windows NT и Windows 2000 и представлены ниже в порядке расположения на диске:

- загрузочные секторы главного и дополнительного разделов;
- загрузочный сектор логического диска;
- таблицы размещения файлов (FAT);
- корневой каталог;
- область данных;
- цилиндр для выполнения диагностических операций чтения/записи.

В отличие от жесткого диска, на дискетах нет загрузочных секторов главного и дополнительного разделов и диагностического цилиндра. Эти структуры создаются программой Fdisk, которая не применяется для дискет, так как они не могут быть разбиты на разделы. На рис. 25.1 представлено взаимоотношение этих структур на диске Western Digital Caviar AC12100 размером 2 111 Мбайт.

Каждая дисковая область применяется для конкретной цели. Повреждение одной из перечисленных областей обычно приводит к ограничению доступа к другим областям, вызывая

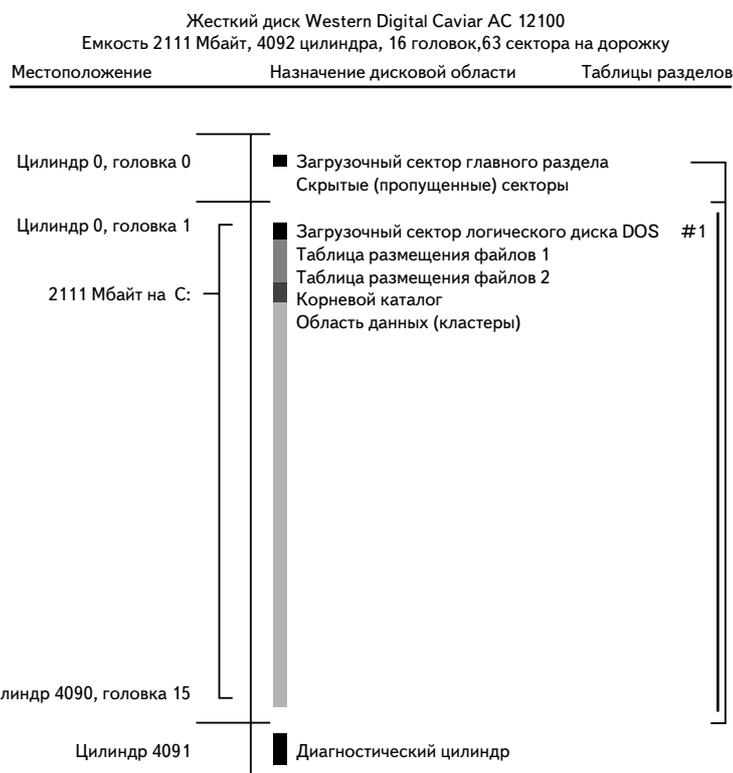


Рис. 25.1. Принцип размещения файлов на обычном жестком диске

сбои в работе. Например, операционная система не сможет получить доступ к диску, если повреждена главная загрузочная запись. Таким образом, понимание логики работы каждой структуры и их взаимодействия оказывает значительную помощь в устранении неполадок.

### Замечание

Некоторые съемные носители, например *Imega Zip*, функционируют подобно "высокоемким дискетам", т.е. на них нет загрузочных секторов главного и дополнительного разделов, а также диагностического цилиндра. Однако такие устройства, как *Imega Jaz*, похожи по структуре на жесткие диски.

## Загрузочный сектор главного раздела

Впервые поддержка жестких дисков была реализована в DOS 2.0, выпущенной в 1983 году. В этой операционной системе впервые использовалась 16-разрядная файловая система FAT и поддерживалось деление диска на разделы, т.е. создание на диске логических томов. Выполнять деление диска необходимо даже в том случае, если вы собираетесь использовать только один раздел. Разделы диска иногда называют логическими томами, поскольку операционная система присваивает каждому из них отдельную букву.

В настоящее время практически все накопители на жестких дисках делятся на несколько разделов, с которыми работает установленная операционная система. Однако, разделив диск, можно установить несколько операционных систем — по одной в каждый раздел, и эти сис-

темы смогут нормально сосуществовать на одном диске. Такая мультизагрузка становится все более популярной в настоящее время.

Чтобы установить на один жесткий диск несколько операционных систем, его надо разбить на разделы. Например, вы можете с помощью программы Fdisk создать один или несколько разделов для установки на них DOS или Windows 9x, а оставшуюся часть диска предоставить для другой операционной системы. Каждый раздел в операционной системе будет иметь вид отдельного диска.

Информация о разделах сохраняется в загрузочном секторе раздела (или логического диска), который находится в начале каждого раздела. Существует также основная таблица списка разделов, помещенная в загрузочный сектор главного раздела.

*Загрузочный сектор главного раздела* (или *главная загрузочная запись (Master Boot Record — MBR)*) является первым сектором на жестком диске (цилиндр 0, головка 0, сектор 1) и состоит из двух элементов.

- *Таблица главного раздела.* Содержит список разделов на диске и расположение загрузочных секторов соответствующих логических дисков. Эта таблица очень маленькая и может содержать максимум четыре записи. Таким образом, для получения большего количества разделов в операционной системе (например, DOS) можно создать один дополнительный раздел и поместить в него несколько логических дисков.
- *Главный загрузочный код.* Небольшая программа, которая выполняется системной BIOS. Основная функция этого кода — передача управления в раздел, который обозначен как активный (или загрузочный).

## Основные и дополнительные разделы FAT

Количество разделов на всех жестких дисках в системе может достигать 24. Это означает, что в компьютере может быть установлено либо 24 отдельных накопителя, в каждом из которых имеется по одному разделу, либо один жесткий диск с 24 разделами, либо несколько накопителей с разным количеством разделов, но при условии, что общее их количество не больше 24. Если общее количество разделов превысит эту цифру, DOS просто проигнорирует их, хотя другие операционные системы могут работать и с большим количеством томов. Такое ограничение DOS связано с тем, что в латинском алфавите от C до Z всего 24 буквы.

В начале каждого раздела DOS содержится загрузочный сектор логического диска. Разбивая диск на разделы, необходимо создать активный (или загрузочный) раздел. Программа, содержащаяся в самом первом секторе на жестком диске, определяет, какой раздел активен, и передает управление его загрузочному сектору. Вы также можете создать дополнительный раздел диска для Novell NetWare, NTFS (Windows NT), HPFS (OS/2), AIX (UNIX), XENIX или другой файловой системы, используя системный диск с соответствующей программой разделения диска.

Разделы, используемые этими операционными системами, недоступны при работе в DOS. Все дело в различиях между файловыми структурами. DOS использует структуру FAT, которая также поддерживается OS/2, Windows NT и некоторыми другими операционными системами. В то же время в OS/2 обычно вместо FAT применяется файловая система HPFS (High Performance File System), а Windows NT пользуется собственной файловой системой — NTFS (NT File System) и т.д.

В табл. 25.1 приведен формат таблицы разделов, которая хранится в секторе главной загрузочной записи.

В табл. 25.2 приведены стандартные, а в табл. 25.3 — нестандартные значения байта идентификации системы.

При восстановлении поврежденного диска приведенные в табл. 25.2 и 25.3 значения можно ввести с помощью программы Diskedit из пакета Norton Utilities.

**Таблица 25.1. Главная загрузочная запись (таблица разделов)**

Смещение	Длина	Описание
<b>Первая запись в таблице разделов</b>		
1BEh 446	1 байт	Индикатор загрузки (80h – активный, иначе 00h)
1BFh 447	1 байт	Первая головка (или сторона) раздела
1C0h 448	16 бит	Первый цилиндр (10 бит) и сектор (6 бит)
1C2h 450	1 байт	Байт идентификации системы (табл. 25.2)
1C3h 451	1 байт	Последняя головка (сторона) раздела
1C4h 452	16 бит	Последний цилиндр (10 бит) и сектор (6 бит)
1C6h 454	Одно двойное слово	Относительное смещение первого сектора на диске
1CAh 458	Одно двойное слово	Количество секторов в разделе
<b>Вторая запись в таблице разделов</b>		
1CEh 462	1 байт	Индикатор загрузки (80h – активный, иначе 00h)
1CFh 463	1 байт	Первая головка (или сторона) раздела
1D0h 464	16 бит	Первый цилиндр (10 бит) и сектор (6 бит)
1D2h 466	1 байт	Байт идентификации системы (см. табл. 25.2)
1D3h 467	1 байт	Последняя головка (сторона) раздела
1D4h 468	16 бит	Последний цилиндр (10 бит) и сектор (6 бит)
1D6h 470	Одно двойное слово	Относительное смещение первого сектора на диске
1DAh 474	Одно двойное слово	Количество секторов в разделе
<b>Третья запись в таблице разделов</b>		
1DEh 478	1 байт	Индикатор загрузки (80h – активный, иначе 00h)
1DFh 479	1 байт	Первая головка (или сторона) раздела
1E0h 480	16 бит	Первый цилиндр (10 бит) и сектор (6 бит)
1E2h 482	1 байт	Байт идентификации системы (см. табл. 25.2)
1E3h 483	1 байт	Последняя головка (сторона) раздела
1E4h 484	16 бит	Последний цилиндр (10 бит) и сектор (6 бит)
1E6h 486	Одно двойное слово	Относительное смещение первого сектора на диске
1EAh 490	Одно двойное слово	Количество секторов в разделе
<b>Четвертая запись в таблице разделов</b>		
1EEh 494	1 байт	Индикатор загрузки (80h – активный, иначе 00h)
1EFh 495	1 байт	Первая головка (или сторона) раздела
1F0h 496	16 бит	Первый цилиндр (10 бит) и сектор (6 бит)
1F2h 498	1 байт	Байт идентификации системы (см. табл. 25.2)
1F3h 499	1 байт	Последняя головка (сторона) раздела
1F4h 500	16 бит	Последний цилиндр (10 бит) и сектор (6 бит)
1F6h 502	Одно двойное слово	Относительное смещение первого сектора на диске
1FAh 506	Одно двойное слово	Количество секторов в разделе

Смещение	Длина	Описание
<b>Байты сигнатуры</b>		
1Feh 510	2 байт	Сигнатура загрузочного сектора (55AAh)

Примечание. В этой и последующих таблицах слово соответствует двум байтам в обратном порядке, двойное слово — двум словам в обратном порядке.

**Таблица 25.2. Байт идентификации системы в таблице разделов (стандартные значения)**

Значение	Тип раздела	Режим трансляции	Размер раздела
00h	Нет	–	–
01h	Основной, FAT 12	CHS	0–15 Мбайт
04h	Основной, FAT 16	CHS	16–32 Мбайт
05h	Дополнительный	CHS	16–32 Мбайт
06h	Основной, FAT 16	CHS	32 Мбайт–2 Гбайт
07h	NTFS/HPFS	Все	Любой
0Ah	Диспетчер загрузки OS/2	Все	Любой
0Bh	Основной, FAT 32	LBA	512 Мбайт–2 Тбайт
0Ch	Основной, FAT 32	LBA	512 Мбайт–2 Тбайт
0Eh	Основной, FAT 16	LBA	32 Мбайт–2 Гбайт
0Fh	Дополнительный	LBA	32 Мбайт–2 Гбайт

**Таблица 25.3. Байт идентификации системы в таблице разделов (нестандартные значения)**

Значение	Тип раздела
02h	Корневой раздел MS-XENIX
03h	Пользовательский раздел MS-XENIX
08h	Раздел файловой системы AIX
09h	Загрузочный раздел AIX
12h	Диагностический раздел Compaq
50h	Раздел Ontrack Disk Manager только для чтения (DOS)
51h	Раздел Ontrack Disk Manager для чтения и записи (DOS)
54h	Раздел Ontrack Disk Manager (не DOS)
55h	Раздел Micro House EZ-Drive (не DOS)
56h	Раздел Golden Bow Vfeature
61h	Раздел Storage Dimensions Speedstor
63h	Раздел IBM 386/ix или UNIX System V/386
64h	Раздел Novell NetWare 286
65h	Раздел Novell NetWare 386
75h	Раздел IBM PCIX

Значение	Тип раздела
82h	Файл подкачки Linux
83h	Файловая система Linux
DBh	Раздел Digital Research Concurrent DOS/CPM-86
E1h	Дополнительный раздел FAT 12 SpeedStor
E4h	Дополнительный раздел FAT 16 SpeedStor
F4h	Первичный раздел SpeedStor
FFh	Раздел дефектных блоков UNIX

## Недокументированные возможности программы Fdisk

Программа Fdisk обладает очень большими возможностями, которые были еще более расширены в DOS 5 и следующих версиях. К сожалению, эти возможности никогда не документировались в руководстве по DOS и не были описаны даже в Windows. Самым важным из недокументированных параметров является /MBR. С его помощью программа Fdisk перезаписывает данные в главном загрузочном секторе, оставляя неизменными таблицы разбиения.

Параметр /MBR словно специально предназначен для уничтожения вирусов, “заражающих” главный загрузочный сектор диска (цилиндр 0, головка 0, сектор 1). Чтобы воспользоваться этой возможностью, введите следующую команду: **fdisk /mbr**.

После этого Fdisk перезапишет код загрузочного сектора, оставляя таблицы разбиения неизменными. В нормально работающей системе это не приведет к проблемам, но на всякий случай создайте резервную копию таблиц разбиения на дискете.

Имейте в виду: таблицы разбиения будут перезаписаны в том случае, если 2 байта контрольного кода (сигнатуры) 55AAh в конце сектора окажутся поврежденными. Но эта ситуация маловероятна. На самом деле в случае повреждения байтов сигнатуры вы сразу же об этом узнаете: система перестанет загружаться и будет вести себя так, как будто разбиения диска вообще не существует. Если система после загрузки с проверенной дискеты или оптического диска не распознает жесткий диск, то его загрузочный сектор может быть инфицирован. В этом случае следует воспользоваться какой-либо современной антивирусной программой и попытаться восстановить поврежденные данные.

### **Внимание!**

Команду FDISK/MBR следует использовать только в системах с нормальной структурой главной загрузочной записи. Если в системе для доступа ко всем разделам диска используются такие программы, как Disk Manager, EZ-Drive, MaxBlast, Data Lifeguard Tools и им подобные, *не применяйте* команду FDISK/MBR. Это связано с тем, что в указанных программах для доступа к диску используется модифицированная версия MBR. Команда FDISK/MBR может привести к уничтожению изменений, сделанных этими программами в структуре диска, в результате чего данные станут недоступными.

В служебной программе Recovery Console ОС Windows NT/2000 существует функция, эквивалентная команде FDISK/MBR, которая называется FIXMBR. Аналогом команды FDISK в Recovery Console является функция DISKPART. Для получения более подробной информации по использованию этих функций загрузите программу Recovery Console и введите команду HELP.

## Загрузочный сектор

*Загрузочный сектор* — это первый сектор на любом логическом диске DOS. Например, на дискете или на диске Zip это самый первый физический сектор, так как дискету нельзя разбить на разделы и она имеет только один логический диск. На жестком диске загрузочный сектор (секторы) располагается в начале каждого раздела, не являющегося дополнительным, или в начале любой области диска, распознаваемой как логический диск DOS.

Эти секторы немного похожи на загрузочные секторы разделов, так как содержат таблицы со специальной информацией о логическом диске.

- *Блок параметров диска*, в котором содержится специфическая информация, например размер раздела, количество используемых секторов диска, размер кластера и метка тома.
- *Загрузочный код* — программа, которая начинает процесс загрузки операционной системы. Для DOS и Windows 9x/Me это файл Io.sys.

Загрузочный сектор дискеты загружается ROM BIOS, а при загрузке системы с жесткого диска MBR передает управление загрузочному сектору активного раздела. В обоих случаях загрузочный сектор логического диска получает управление. Он выполняет некоторые проверки и затем пытается прочитать с диска первый системный файл (в DOS/Windows это файл Io.sys). Загрузочный сектор не виден, так как находится вне области хранения файлов логического диска.

### Замечание

---

Большинство современных систем поддерживают загрузку с других устройств, а не только с дискеты. Эта возможность обеспечивается системной BIOS. Например, некоторые системы, помимо загрузки с жесткого диска и дискеты, могут загружаться с накопителя CD-ROM или диска Zip.

---

Загрузочный сектор логического диска создается программой DOS и Windows 9x Format. На жестком диске загрузочные секторы есть в начале каждого логического диска как в основном, так и в дополнительном разделах. Все загрузочные секторы наряду с данными о логическом диске содержат специальную запись, однако при загрузке выполняется код только того сектора, который находится в активном разделе. Остальные секторы просто считываются операционной системой для определения параметров логических дисков.

Загрузочный сектор логического диска состоит из программы (выполняемого кода) и области данных. Эта информация необходима операционной системе для определения размера логического диска и размещения таких структур, как FAT. Формат блока параметров диска весьма специфичен. Ошибки в этом блоке могут привести к проблемам при загрузке DOS или к отсутствию доступа к диску.

В табл. 25.4 приведены форматы загрузочного сектора DOS различных версий.

## Корневой каталог

*Каталог* — это база данных, содержащая информацию о записанных на диске файлах. Каждая запись в ней имеет длину 32 байт, и между записями не должно быть никаких разделителей. В каталоге сохраняется практически вся информация о файле, которой располагает операционная система.

- *Имя файла и расширение* — восемь символов имени и три символа расширения; точка между именем и расширением файла подразумевается, но не включается в эту запись.

### Замечание

---

В Windows 9x/Me имя файла может состоять из 255 символов в структуре каталога 8.3.

---

- *Байт атрибутов файла*, содержащий флаг, который представляет стандартные атрибуты файла.
- *Время и дата* создания файла или его модификации.
- *Размер файла* в байтах.
- *Ссылка на начальный кластер* — номер кластера, с которого начинается файл.

Информация о расположении файла, т.е. расположении оставшихся кластеров, содержится в FAT.

Существует два основных типа каталогов: *корневой каталог* и *подкаталог*. Различаются они максимальным количеством хранящихся файлов. На каждом логическом диске в фиксированном месте, сразу же за копиями FAT, располагается корневой каталог. Размеры корневых каталогов варьируются в зависимости от размера диска, но каждый конкретный корневой каталог

**Таблица 25.4. Форматы загрузочной записи различных версий DOS**

Смещение			
HEX	DEC	Длина поля	Описание
00h	0	3 байт	Команда перехода на код загрузки
03h	3	8 байт	Имя производителя и версия DOS
0Bh	11	Одно слово	Размер сектора в байтах (обычно 512)
0Dh	13	1 байт	Размер кластера в секторах (степень числа 2)
0Eh	14	Одно слово	Количество зарезервированных секторов (обычно 1)
10h	16	1 байт	Количество копий FAT (обычно 2)
11h	17	Одно слово	Максимальное количество записей в корневом каталоге (обычно 512)
13h	19	Одно слово	Всего секторов (если раздел не больше 32 Мбайт, в противном случае 0)
15h	21	1 байт	Байт описания диска (F8h для жесткого диска)
16h	22	Одно слово	Размер FAT в секторах
18h	24	Одно слово	Количество секторов на дорожке
1Ah	26	Одно слово	Количество головок
1Ch	28	Одно двойное слово	Количество скрытых секторов (если раздел не больше 32 Мбайт, то только одно слово)
<b>Для DOS версии 4.0 и более поздних, иначе 00h</b>			
20h	32	Одно двойное слово	Всего секторов (если раздел больше 32 Мбайт, иначе 0)
24h	36	1 байт	Физический номер диска (00h — дисковод, 80h — жесткий диск)
25h	37	1 байт	Зарезервировано (00h)
26h	38	1 байт	Сигнатура расширенной загрузочной записи (29h)
27h	39	Одно двойное слово	Серийный номер тома (32-разрядное случайное число)
2Bh	43	11 байт	Метка тома ("NO NAME", если нет метки)
36h	54	8 байт	Идентификатор файловой системы ("FAT12" или "FAT16")
<b>Для всех версий DOS</b>			
3Eh	62	448 байт	Код программы загрузки
1FEh	510	2 байт	Байты сигнатуры (55AAh)

имеет *фиксированное* максимальное число файлов. Длина корневого каталога фиксируется при создании логического диска и не может быть изменена в процессе работы. Размер корневого каталога различных накопителей приведен в табл. 25.5. В отличие от корневого каталога, подкаталог может хранить произвольное количество файлов и расширяться по мере необходимости.

**Таблица 25.5. Размер корневого каталога**

Тип накопителя	Максимальное количество записей
Жесткий диск	512
Дисковод 1,44 Мбайт	224
Дисковод 2,88 Мбайт	448
Jaz и Zip	512
LS-120	512

**Замечание**

Одно из преимуществ FAT 32 заключается в том, что корневой каталог может располагаться в любом месте диска и содержать неограниченное количество записей.

Все каталоги имеют одинаковую структуру. Записи в этой базе данных сохраняют важную информацию о файлах, которая связана с информацией, хранящейся в FAT, посредством одного из полей записи — номера первого занимаемого файлом кластера на диске. Если бы все файлы на диске не превышали размеров одного кластера, потребности в FAT вообще бы не возникло. В FAT содержится информация о файле, отсутствующая в каталоге, — номера кластеров, в которых расположен *весь* файл.

Чтобы отследить расположение всего файла на диске, обратитесь к каталогу и выясните номер первого кластера и длину файла. Затем, используя таблицу размещения файлов, просмотрите *цепочку* кластеров, занимаемых файлом, пока не дойдете до конца файла.

Формат 32-байтовой записи в каталоге приведен в табл. 25.6.

Имена файлов и их расширения записаны с привязкой к левому краю и дополнены до максимальной длины пробелами, т.е. имя файла *AL* будет реально сохранено как *AL.....*, где точки обозначают пробелы. Первый байт имени файла также может обозначать его состояние, как приведено в табл. 25.7.

В табл. 25.8 приводятся используемые в записях каталогов атрибуты файлов.

**Таблица 25.6. Формат каталога**

Смещение			
Hex	Dec	Длина поля	Описание
00h	0	8 байт	Имя файла
08h	8	3 байт	Расширение файла
0Bh	11	1 байт	Атрибуты файла
0Ch	12	10 байт	Зарезервировано (00h)
16h	22	Одно слово	Время создания
18h	24	Одно слово	Дата создания
1Ah	26	Одно слово	Начальный кластер
1Ch	28	Одно двойное слово	Размер файла в байтах

**Таблица 25.7. Байт состояния записи каталога (первый байт)**

Hex	Состояние файла
00h	Запись никогда не использовалась; ниже этой записи поиск не выполняется
05h	Первый символ имени файла в настоящее время – E5h
E5h	Файл удален
2Eh	Точка (.) показывает, что запись является каталогом. Если и второй байт – 2Eh, то поле начального кластера содержит номер кластера родительского каталога (0000h, если родительский каталог корневой)

**Таблица 25.8. Атрибуты файлов**

Позиция бита в шестнадцатеричном формате										
7	6	5	4	3	2	1	0	Значение	Описание	
0	0	0	0	0	0	0	1	01h	Только для чтения	
0	0	0	0	0	0	1	0	02h	Скрытый	
0	0	0	0	0	1	0	0	04h	Системный	
0	0	0	0	1	0	0	0	08h	Метка тома	
0	0	0	1	0	0	0	0	10h	Подкаталог	
0	0	1	0	0	0	0	0	20h	Архивный (измененный)	
0	1	0	0	0	0	0	0	40h	Зарезервировано	
1	0	0	0	0	0	0	0	80h	Зарезервировано	
<b>Примеры</b>										
0	0	0	0	0	1	1	1	07h	Системный, скрытый, только для чтения	
0	0	1	0	0	0	0	1	21h	Только для чтения, архивный	
0	0	1	1	0	0	1	0	32h	Скрытый, подкаталог, архивный	
0	0	1	0	0	1	1	1	27h	Только для чтения, скрытый, системный, архивный	

## Таблица размещения файлов

Таблица размещения файлов (FAT) содержит номера кластеров, в которых расположены файлы на диске. Каждому кластеру в FAT соответствует одно число. Секторы, не содержащие пользовательских данных (файлов), не отражены в FAT. К таким секторам относятся загрузочные секторы, таблицы размещения файлов и секторы корневого каталога.

В файловой системе FAT дисковое пространство разбивается не на секторы, а на группы секторов, которые называются *кластерами* (*ячейками размещения*). Кластер содержит один или несколько секторов. Размер кластера определяется при делении диска на разделы с помощью программы Fdisk и зависит от размера создаваемого раздела. Наименьший размер диска, который может занимать файл ненулевого размера, — один кластер. Каждый файл использует целое число кластеров. Например, если файл занимает на один байт больше размера кластера, то для его размещения на диске будет выделено два кластера.

FAT — это электронная таблица, управляющая распределением дискового пространства. Каждая ячейка этой таблицы связана с определенным кластером на диске. Число, содержа-

щеся в этой ячейке, сообщает о том, использован ли данный кластер под какой-либо файл и, если использован, где находится следующий кластер этого файла.

Каждая ячейка FAT хранит шестнадцатеричное значение длиной 12 или 16 бит. Шестнадцатизрядные FAT более удобны в работе, так как значительно легче редактировать поля размером в два байта, чем в полтора. Чтобы самостоятельно отредактировать FAT, вы должны заполнить некоторые математические преобразования для получения номера кластера. К счастью, многие программы позволяют отредактировать FAT автоматически. В большинстве этих программ номера кластеров представлены в десятичном виде, наиболее удобном для пользователей. В табл. 25.9 приведены данные о каталоге и FAT (файл не фрагментирован).

В данном примере запись каталога указывает начальный кластер (1000), в котором размещается файл. В FAT кластеры с ненулевыми значениями используются, а специальное значение указывает дальнейшее расположение файла. В рассматриваемом примере в кластере 1000 указывается кластер 1001, в 1001 — 1002, в 1002 — 1003, а в 1003 записано значение FFFFh, т.е. на этом кластере файл заканчивается.

Рассмотрим пример с фрагментированным файлом. Пусть файл Usconst.txt записан, начиная с кластера номер 1000. А файл Pledge.txt начинается с кластера 1002. Таким образом, файл Usconst.txt становится фрагментированным. Описанная ситуация иллюстрируется данными в табл. 25.10.

В данном примере в файл Usconst.txt “внедряется” файл Pledge.txt, что приводит к непоследовательному расположению файлов на диске, т.е. фрагментации. В операционных системах DOS и Windows есть программы дефрагментации, которые перемещают файлы для их последовательного размещения на диске.

Первые две записи FAT зарезервированы и содержат информацию о самой FAT, все остальные указывают на соответствующие кластеры диска. Большинство записей FAT состоит

**Таблица 25.9. Записи о нефрагментированном файле в каталоге и FAT**

Каталог		
Имя	Начальный кластер	Размер
Usconst.txt	1000	4

FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
01005	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

из ссылок на кластеры, в которых представлены части определенного файла, а некоторые содержат специальные шестнадцатеричные значения:

- *0000h* — кластер не используется;
- *FFF7h* — как минимум один сектор в кластере поврежден и не может быть использован для хранения данных;
- *FFF8h–FFFFh* — кластер содержит конец файла.

Тип используемых FAT определяется программой *Fdisk*, хотя записываются они в процессе форматирования высокого уровня программой *Format*. На всех дискетах применяется 12-разрядная FAT, а на жестком диске может использоваться как 12-, так и 16-разрядная FAT, в зависимости от размера логического диска. На дисках размером меньше 16 Мбайт (32 768 секторов) применяется 12-разрядная FAT, на дисках большего размера — 16-разрядная, а на дисках размером более 512 Мбайт при использовании Windows 95 OSR2 и Windows 98 — 32-разрядная FAT.

Программа *Fdisk* обычно создает на одном диске две копии FAT. Каждая копия занимает несколько последовательных секторов на диске, и вторая копия записывается непосредственно после первой. К сожалению, операционная система использует вторую копию FAT только в том случае, когда невозможно прочитать секторы, содержащие первую копию. Таким образом, если первая копия FAT пропадет (весьма распространенная ситуация), операционная система не будет использовать вторую копию. Даже команда *Chkdsk* не проверяет вторую копию FAT. Кроме того, каждый раз, когда операционная система обновляет первую копию FAT, большие участки первой копии автоматически копируются во вторую. Если же первая копия повреждена, то и вторая окажется поврежденной: после обновления FAT вторая копия отражает все изменения в первой копии, включая и ошибки. Обе копии FAT редко отличаются

**Таблица 25.10. Записи о фрагментированном файле в каталоге и FAT**

Каталог		
Имя	Начальный кластер	Размер
Pledge.txt	1002	2
Usconst.txt	1000	4

FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1004	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	1005	Используется; ссылка на следующий кластер
01005	FFFFh	Конец файла
...	...	...
65526	0	Последний доступный кластер

ся одна от другой, по крайней мере в течение продолжительного срока: при обновлении первая копия FAT автоматически копируется во вторую. Учитывая все это, можно сказать, что применение второй копии FAT ограничивается только операциями по восстановлению дефектных данных. Но даже в такой ситуации использовать вторую копию FAT можно только в том случае, когда проблема решается *немедленно*, не дожидаясь очередного обновления FAT.

## Кластер (ячейка размещения)

Термин *кластер* в DOS 4.0 был заменен термином *ячейка размещения (allocation unit)*. Новый термин — синоним старого, так как кластер является наименьшей ячейкой на диске, которой может оперировать система при чтении или записи файла на диск. Кластер соответствует одному или (чаще всего) нескольким секторам. Это позволяет уменьшить размер FAT и ускорить работу операционной системы, так как ей приходится оперировать меньшим числом распределяемых ячеек. В то же время с увеличением размера кластера на диске растет и размер неиспользуемого дискового пространства, так как его распределение происходит с дискретностью в один кластер.

В табл. 25.11 приведены стандартные размеры кластеров для различных форматов дискет.

Довольно странным является то обстоятельство, что некоторые дискеты высокой плотности имеют меньший размер кластера, чем дискеты низкой плотности. Увеличивается размер FAT, увеличивается количество записей, которые должна обрабатывать операционная система, и замедляется работа самой системы. Меньший размер кластера позволяет сократить размер неиспользуемого дискового пространства. Все пространство между концом файла и концом последнего занимаемого кластера не используется, и в результате, чем больше размер кластера, тем больше потери дискового пространства. Кроме того, дисководы высокой плотности работают быстрее, чем их “родственники” низкой плотности. Все это позволило IBM и Microsoft пойти на уменьшение размера кластера в дискетах высокой плотности, хотя при этом и увеличивается FAT.

Для жестких дисков размер кластера может варьироваться в зависимости от размера раздела диска. В табл. 25.12 приведены размеры кластеров в зависимости от размера логического диска.

Использование кластеров больших размеров ощутимо сказывается на работе системы. Например, на диске емкостью 2 Гбайт, содержащем 5 000 файлов, со средней потерей дискового пространства в полкластера на один файл суммарные потери дискового пространства составят около 78 Мбайт [ $5000 \times (0,5 \times 32)$ ].

Размер кластера и структура FAT определяют максимально возможный размер раздела. Поскольку FAT использует записи размером 16 байт для ссылки на кластер в разделе, максимально возможное число кластеров может равняться 65 536 ( $2^{16}$ ). Максимальный размер кластера — 32 Кбайт, следовательно, максимально возможный размер раздела — 2 047,6875 Мбайт.

Операционные системы Windows 95 OSR2x и Windows 98 поддерживают 32-разрядную FAT с размером кластера до 64 Кбайт. С одной стороны, использование большего количества

**Таблица 25.11. Стандартные размеры кластеров для дискет**

Тип диска	Размер кластера (ячейки размещения)	Плотность
5,25-дюймовый на 360 Кбайт	Два сектора (1 024 байт)	Низкая
5,25-дюймовый на 1,2 Мбайт	Один сектор (512 байт)	Высокая
3,5-дюймовый на 720 Кбайт	Два сектора (1 024 байт)	Низкая
3,5-дюймовый на 1,44 Мбайт	Один сектор (512 байт)	Высокая
3,5-дюймовый на 2,88 Мбайт	Два сектора (1 024 байт)	Экстра

маленьких кластеров позволяет уменьшить потери дискового пространства, а с другой — большие кластеры необходимы для больших логических дисков. Так, использование 32-разрядных FAT позволяет превысить существующий на данный момент лимит в 2 Гбайт для одного раздела до 2 Тбайт (2 048 Гбайт). Вообще же, предел в 2 Гбайт существует только для DOS; такие операционные системы, как Windows 9x и Windows NT, давно уже его преодолели.

**Таблица 25.12. Стандартные размеры кластеров**

Размер диска, Мбайт	Размер кластера	Тип FAT
Меньше 16	8 секторов (4 096 байт)	12-разрядная
16–128	4 сектора (2 048 байт)	16-разрядная
128–256	8 секторов (4 096 байт)	16-разрядная
256–512	16 секторов (8 192 байт)	16-разрядная
512–1 024	32 сектора (16 384 байт)	16-разрядная
1 024–2 048 и более	64 сектора (32 768 байт)	16-разрядная

## Область данных

*Область данных* диска — это область, следующая за загрузочным сектором, таблицами размещения файлов и корневым каталогом на любом логическом диске. Эта область контролируется с помощью FAT и корневого каталога и делится на ячейки размещения, называемые *кластерами*. В этих кластерах и располагаются сохраняемые на диске файлы.

## Цилиндр для диагностических операций чтения и записи

Программа разбивки диска на разделы Fdisk всегда резервирует последний цилиндр жесткого диска для выполнения диагностических операций. Из-за этого Fdisk указывает меньшее количество цилиндров, чем существует на самом деле. Операционная система не использует этот цилиндр, поскольку он находится вне разделов.

В системах с дисковыми интерфейсами IDE или SCSI контроллер должен выделить дополнительную область после разделов для хранения таблицы испорченных дорожек и запасных секторов. В этом случае разница между фактическим числом цилиндров и тем, что показывает Fdisk, будет еще больше.

Область диагностики позволяет выполнять тестирование чтения/записи жесткого диска, не повреждая данных на диске. Программы форматирования жестких дисков на низком уровне обычно используют этот цилиндр для тестирования чередования диска либо для хранения необходимой во время форматирования информации.

## VFAT и длинные имена файлов

В оригинальной операционной системе Windows 95 используется та же файловая система, что и в DOS, но с весьма существенными улучшениями. В Windows 95 поддерживается файловая система FAT, переписанная в 32-разрядный код и названная *виртуальной таблицей*

размещения файлов (*virtual file allocation table* — VFAT). VFAT используется вместе с 32-разрядной программой VCACHE (заменившей 16-разрядную программу SMARTDrive из DOS и Windows 3.1), что обеспечивает более высокую производительность файловой системы. Однако наиболее существенным улучшением новой файловой системы является поддержка длинных имен файлов. Системы DOS и Windows 3.1 ограничивались стандартом “восемь-точка-три” при именовании файлов, поэтому добавление поддержки длинных имен файлов было приоритетной задачей, которую необходимо было решить разработчикам Windows 95, тем более что пользователи операционных систем Macintosh и OS/2 уже всюду применяли эти возможности. Таким образом, создатели Windows 95 должны были обеспечить обратную совместимость, т.е. необходимо было реализовать в файловой системе все новые свойства и, кроме того, не “обделить” пользователей предыдущих версий DOS и Windows. Кстати, обратная совместимость — одна из самых распространенных проблем в мире персональных компьютеров.

В системе VFAT файлу или каталогу можно присваивать имя длиной до 255 символов (включая путь к этому файлу или каталогу). В Windows 95 от трехсимвольного расширения не отказались, поскольку в этой операционной системе (как и в предыдущих версиях Windows) с помощью расширения создается ассоциация типа “файл–приложение”. В длинных именах файлов можно использовать пробелы, а также символы + , ; = [ ], которые нельзя было использовать в стандартных (восемь-точка-три) именах файлов DOS.

При создании длинного имени файла создается его псевдоним, удовлетворяющий стандарту “восемь-точка-три”. В Windows 9x файловая система VFAT выполняет это следующим образом.

1. Первых три символа после последней точки в длинном имени файла становятся расширением псевдонима.
2. Первых шесть символов длинного имени файла (за исключением пробелов, которые игнорируются) преобразуются в символы верхнего регистра и становятся первыми шестью символами стандартного имени файла. Недопустимые в стандартном имени файла символы (+ , ; = [ ]) преобразуются в символы подчеркивания.
3. VFAT добавляет символы ~1 (седьмой и восьмой) к псевдониму имени файла. Если первых шесть символов нескольких файлов одинаковы, то для разрешения конфликтов имен добавляются символы ~2, ~3 и т.д.

### **Длинные имена файлов в Windows NT/2000/XP**

Обратите внимание, что в Windows NT/2000/XP псевдонимы имен файлов создаются иначе, чем в Windows 9x. Операционная система Windows NT/2000/XP использует для создания “короткого” имени файла первые шесть допустимых символов длинного имени и, если созданное имя уникально, добавляет символы ~1. Если же первые шесть символов уже используются другим файлом, то добавляются символы ~2. Для создания расширения Windows NT/2000/XP использует первые три допустимых символа после последней точки в длинном имени файла. Если после добавления символов ~5 появляется еще одно аналогичное “короткое” имя файла, то для создания следующих имен файлов используется такой алгоритм: длинное имя файла преобразуется в четыре шестнадцатеричных символа, которые помещаются после двух допустимых символов длинного имени с добавлением символов ~5. Таким образом, в Windows NT/2000/XP окончание ~5 появляется у всех псевдонимов файлов, а изменяются только шестнадцатеричные значения.

### **Совет**

Вы можете отключить в системе VFAT механизм создания стандартного имени из длинного имени файла. Для этого в системном реестре добавьте двоичный параметр NameNumericTail в ветви HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\FileSystem и установите его значение равным 0. Для возврата к использованию псевдонимов длинных имен файлов установите значение этого параметра равным 1.

VFAT хранит псевдонимы длинных имен в поле стандартных имен файлов записи каталога файлов. Таким образом, все версии DOS и Windows могут получить доступ к файлу под длинным именем с помощью его псевдонима. Остается еще одна проблема: как хранить 255 символов имени файла в 32 байтах записи каталога, ведь каждый символ имени файла — это один байт? Модифицировать структуру записи каталога нельзя, поскольку тогда предыдущие версии DOS не смогут использовать ее.

Разработчики файловой системы решили эту проблему следующим образом: были добавлены дополнительные записи каталога для хранения длинных имен файлов. Чтобы предыдущие версии DOS не повредили этих дополнительных записей каталога, VFAT устанавливает для них атрибуты, которые нельзя использовать для обычного файла: только для чтения, скрытый, системный и метка тома. Такие атрибуты DOS игнорирует, а следовательно, длинные имена файлов остаются “нетронутыми”.

Существует еще одна проблема, связанная с длинными именами файлов: приложения, не умеющие работать с длинными именами файлов, при открытии такого файла и его последующем сохранении будут записывать его псевдоним в дополнительные записи каталога, следовательно, длинное имя файла будет утрачено.

Старые программы для работы с диском, такие как Norton Disk Doctor, не могут работать с VFAT. Эти программы игнорируют дополнительные записи каталога. Поэтому после “восстановления” диска с помощью таких программ вы можете не найти длинных имен файлов.

Существует еще одна проблема с длинными именами файлов: VFAT создает новый псевдоним всякий раз при создании или копировании файла в новый каталог. Например, файл Expenses-January98.doc сохраняется в папке под псевдонимом EXPENS~1.DOC. Если с помощью программы Windows 9x Explorer скопировать этот файл в папку, в которой уже существует файл Expenses-December97.doc с псевдонимом EXPENS~1.DOC, то VFAT создаст в этой папке для копируемого файла новый псевдоним EXPENS~2.DOC. Причем пользователь не будет уведомлен о таком “самоуправстве”. Для программ, поддерживающих длинные имена файлов, такое копирование не составит проблемы: все длинные имена файлов сохраняются. Если же запустить приложение, которое не поддерживает длинные имена файлов, то, открыв файл EXPENS~1.DOC, пользователь обнаружит, что это файл Expenses-December97.doc, а не Expenses-January98.doc.

### **Замечание**

---

При использовании VFAT рекомендую применять дисковые утилиты, которые поддерживают эту файловую систему. Windows 9x содержит необходимые программы для проверки, восстановления, дефрагментации диска и резервного копирования. Кстати, при запуске старых дисковых утилит в Windows 9x вы будете предупреждены о возможных последствиях.

Если необходимо использовать длинные имена файлов со старыми программами, установите программу Lfnbk.exe с компакт-диска Windows 9x. Эта программа восстанавливает длинные имена файлов, но только в том случае, если структура каталога не изменялась.

---

## **FAT 32**

Когда разрабатывалась FAT, жесткие диски размером 2 Гбайт можно было встретить разве что в научно-фантастических романах. В настоящее время практически все системы нижнего уровня оснащаются жестким диском не менее 2 Гбайт, а чаще всего 16 или 20 Гбайт. При использовании стандартной FAT вы можете создать раздел размером не более 2 Гбайт. Это ограничение приводит к тому, что пользователи больших жестких дисков испытывают неудобства в организации файлов: в их распоряжении несколько дисков.

Для устранения этого ограничения Microsoft предложила новую файловую систему с расширенными возможностями, называемую *FAT 32*. Эта файловая система работает как стандартная FAT, но имеет отличия в организации хранения файлов. Кроме того, FAT 32 можно установить с помощью программы *Fdisk*, в отличие от VFAT, которая является частью *Vmm.vxd*. Система FAT 32 была впервые реализована в Windows 95 OEM Service Release 2 (OSR2). Она встроена также в Windows 98/Me и в Windows 2000.

### **Замечание**

---

Поскольку FAT 32 устанавливается только с помощью программы *Fdisk*, вы не сможете использовать эту файловую систему на дисках и дисках съемных устройств, например *Imega Zip*, которые не имеют файловой системы. Однако такие устройства, как *Imega Jaz*, имеют структуру жестких дисков и на них можно устанавливать FAT 32.

---

Основное преимущество FAT 32 — это возможность использования 32-разрядных записей, вместо 16-разрядных, что приводит к увеличению числа кластеров в разделе до 268 435 456 (вместо 65 536, или  $2^{16}$ ). Это значение эквивалентно  $2^{28}$ , а не  $2^{32}$ , поскольку четыре бита из 32 зарезервированы для других целей.

При использовании FAT 32 размер раздела может достигать 2 Тбайт (1 Тбайт равен 1 024 Мбайт). Новая файловая система может иметь 4 294 967 296 ( $2^{32}$ ) кластеров размером 512 байт, а размер единичного файла может составлять 4 Гбайт. В целом Windows 9x и интерфейс ATA-5 ограничивают размер жесткого диска до 128 Гбайт. В то же время операционные системы Windows 2000/XP могут форматировать раздел FAT 32 размером до 32 Гбайт и читать гораздо большие разделы FAT32.

Существует еще одно отличие FAT 32 от ее предшественниц — положение корневого каталога: он не занимает фиксированного места на диске, как в FAT 16. Корневой каталог в FAT 32 может располагаться в любом месте раздела и иметь любой размер. Устранение ограничений записей корневого каталога обеспечивает динамическое изменение размера раздела FAT 32. Однако Microsoft не реализовала это замечательное свойство в операционных системах Windows 9x, чем и воспользовались независимые разработчики, в частности PowerQuest, создавшая программу PartitionMagic.

Основной недостаток FAT 32 — несовместимость с предыдущими версиями DOS и Windows 95. Вы не сможете загрузить предыдущую версию DOS или оригинальную Windows 95 с диска с файловой системой FAT 32, кроме того, раздел с FAT 32 будет недоступен этим системам при их загрузке с другого диска.

## **Размер кластера FAT 32**

Поскольку раздел FAT 32 имеет больше кластеров, чем раздел FAT 16, размер кластера уменьшается. Использование меньшего кластера снижает потери дискового пространства. Например, раздел размером 2 Гбайт с 5 000 файлов в FAT 32 использует кластер размером 4 Кбайт, вместо 32 Кбайт в FAT 16. Такое уменьшение размера кластера снижает потери дискового пространства с 78 до 10 Мбайт.

Для сравнения FAT 16 и FAT 32 необходимо посмотреть, как в этих файловых системах организовано хранение данных. Номера кластеров в FAT 16 хранятся в виде 16-разрядных записей (0000h–FFFFh). Максимальное значение FFFFh соответствует десятичному 65 536, но несколько значений зарезервированы для специальных целей. Реальное число кластеров в FAT 16 лежит в диапазоне 0002h–FFF6h, или 2–65 526. Таким образом, для хранения файлов используется 65 524 кластера. Типичная запись о файле в FAT 16 представлена в табл. 25.13.

**Таблица 25.13. Записи файлов в файловой системе FAT 16**

Каталог		
Имя	Начальный кластер	Размер
Usconst.txt	1000	4

FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

В FAT 32 количество кластеров лежит в диапазоне 00000000h–FFFFFFFh, или 0–4 294 967 295. Как и в FAT 16 верхние и нижние кластеры зарезервированы для специальных целей и их номера лежат в диапазоне 00000002h–FFFFFF6h, или 2–4 294 967 286. Таким образом, для хранения файлов можно использовать 4 294 967 284 кластера. Накопитель на жестких дисках разбит на большее количество кластеров, каждый из которых становится меньше, что снижает потери дискового пространства. Пример записей о файле в FAT 32 приведен в табл. 25.14.

### Замечание

С сервера журнала *PC Magazine* (<ftp://ftp.zdnet.com/pcmag/1995/0627/chkdrv.zip>) можно загрузить программу CHKDRV для оценки потерь при использовании файловых систем FAT 16 и FAT 32.

В табл. 25.15 приведен размер кластера при использовании FAT 32 с различными размерами разделов.

Уменьшение размера кластера приводит к увеличению записей в FAT. Раздел размером 2 Гбайт с FAT 32 использует 524 288 записей, в то время как аналогичный раздел с FAT 16 использует 65 536 записей. Следовательно, таблица FAT 16 имеет размер 128 Кбайт (65 536 записей × 16 бит = 1 048 576 бит, или 131 072 байт, или 128 Кбайт), а таблица FAT 32 — 2 Мбайт.

Размер FAT существенно влияет на производительность файловой системы. В Windows 9x модуль VCACHE пытается загрузить FAT в оперативную память для повышения производительности системы. Выбор кластера размером 4 Кбайт на дисках емкостью до 8 Гбайт обеспечивает компромисс между производительностью и размером FAT в оперативной памяти.

Несмотря на то что размер FAT в файловой системе FAT 32 практически в двадцать раз больше, чем в FAT 16, появляется незначительный (менее 5%) прирост производительности FAT 32 в операционной системе Windows 9x. Это отчасти достигается использованием в персональных компьютерах самых современных накопителей на жестких дисках.

**Таблица 25.14. Записи файлов в системе FAT 32**

Каталог		
Имя	Начальный кластер	Размер
Usconst.txt	1000	8

FAT 32		
Номер кластера	Значение	Назначение
000000002	0	Первый доступный кластер
...	...	...
000000999	0	Кластер доступен
000001000	1001	Используется; ссылка на следующий кластер
000001001	1002	Используется; ссылка на следующий кластер
000001002	1003	Используется; ссылка на следующий кластер
000001003	1004	Используется; ссылка на следующий кластер
000001004	1005	Используется; ссылка на следующий кластер
000001005	1006	Используется; ссылка на следующий кластер
000001006	1007	Используется; ссылка на следующий кластер
000001007	FFFFFFFFh	Конец файла
000001008	0	Кластер доступен
...	...	...
4294967286	0	Последний доступный кластер

**Таблица 25.15. Размер кластера файловой системы FAT 32**

Размер раздела	Размер кластера, байт
До 260 Мбайт	512
260 Мбайт–8 Гбайт	4 096
6–16 Гбайт	8 192
16–32 Гбайт	16 384
32 Гбайт–2 Тбайт	32 768

## Зеркальная копия файловой системы

Система FAT 32 также использует преимущества двух копий FAT в разделе диска. Как и в FAT 16, в FAT 32 первая копия является основной и периодически копирует данные в дополнительную копию FAT. В FAT 32 при появлении проблем с главной копией FAT система переключается на дополнительную копию, которая становится главной. Помимо этого, система прерывает процесс создания зеркальной копии FAT, чтобы предотвратить потерю данных.

## Создание раздела FAT 32

Для создания раздела с FAT 32 в Windows 9x необходимо использовать программу Fdisk в командной строке так же, как при создании раздела с FAT 16. При запуске этой программы будет выполнено тестирование диска и, если его размер превышает 512 Мбайт, появится следующее сообщение:

Компьютер имеет диск емкостью более 512 Мбайт. Данная версия Windows включает поддержку больших дисков и позволяет эффективнее использовать место на таких дисках, а также форматировать диски размером более 2 Гбайт как один диск.

**ВНИМАНИЕ!** Если включить поддержку больших дисков и создать на них новый диск, невозможно будет получить доступ к новому диску из другой операционной системы, включая некоторые версии Windows 95 и Windows NT, а также более ранние версии Windows и MS DOS. Кроме того, дисковые служебные программы, которые не поддерживают явно файловую систему FAT32, не смогут работать с этим диском. Если собираетесь обращаться к этому диску из других операционных систем или более старых служебных программ, не включайте поддержку больших дисков.

Включить поддержку больших дисков (Y/N).....? [N]

Если вы ответите на этот вопрос утвердительно, все разделы размером более 512 Мбайт будут иметь файловую систему FAT 32. Кроме того, утвердительный ответ требуется для создания раздела размером более 2 Гбайт. Последующие окна работы программы Fdisk аналогичны окнам предыдущих версий этой программы.

Программа Fdisk автоматически определяет размер кластера на основе выбранной файловой системы и размера раздела. Однако существует недокументированный параметр команды Format, позволяющий явно указать размер кластера: Format / Z:n, где n — размер кластера в байтах, кратный 512. С помощью этой команды вы можете создать файловую систему с размером кластера, меньшим установленного по умолчанию.

### **Внимание!**

---

При использовании переключателя /z необходимо помнить об ограничении FAT 16 на количество кластеров (65 536). Используйте этот переключатель с файловой системой FAT 32. Не забывайте, что модификация размера кластера может отразиться на производительности файловой системы.

---

## Преобразование FAT 16 в FAT 32

Операционная система Windows 95 OSR2 может создавать разделы с FAT 32 только на пустом диске. Для преобразования раздела с FAT 16 необходимо скопировать данные на другой носитель, удалить раздел с FAT 16 и создать новый раздел с FAT 32, а затем восстановить все данные. В Windows 98/Me существует программа-мастер для преобразования раздела в FAT 32 без потери данных.

При запуске программа преобразования диска отображает информацию о существующих разделах и установленных файловых системах. Вам необходимо лишь выделить диск и выполнить все операции мастера.

Обратите внимание, что после преобразования диска в FAT 32 выполнить обратное преобразование нельзя. Необходимо принимать “радикальные” меры, т.е. сохранить данные, запустить программу Fdisk, удалить раздел с FAT 32 и заново создать раздел с FAT 16.

## **FAT 32 и PartitionMagic**

Операционные системы Windows 95 OSR2 и Windows 98 содержат базовые инструменты для создания разделов с файловой системой FAT 32. Компания PowerQuest создала программу PartitionMagic, которая обладает широкими возможностями работы с файловыми системами. С помощью этой программы вы можете выполнить преобразование FAT 16 в FAT 32 и обратно, а также изменить размеры разделов без потери данных.

# Ошибки файловой системы FAT

Ошибки в файловой системе появляются скорее из-за программных, нежели из-за аппаратных сбоев (например, при неверном завершении работы Windows). Некоторые программные ошибки описаны ниже.

## Потерянные кластеры

Это наиболее распространенная ошибка файловой системы, при которой кластеры в FAT помечаются как используемые, хотя на самом деле таковыми не являются. Эти потерянные кластеры появляются при неверном завершении работы приложения или крахе системы. Программы восстановления диска могут обнаружить эти кластеры и восстановить их.

Потерянные кластеры появляются в файловой структуре (табл. 25.16).

Появляются цепочки кластеров, не имеющие записей в каталоге. Чаще всего это происходит при “зависании” программы при операции открытия файла.

Программы восстановления диска просматривают диск и создают копию FAT в оперативной памяти. Затем эта копия сравнивается с “настоящей” FAT и таким образом выявляются

**Таблица 25.16. Потерянные кластеры в файловой структуре**

Каталог		
Имя	Начальный кластер	Размер
(Нет записей)	0	0
FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

потерянные кластеры, т.е. не принадлежащие ни одному из существующих файлов. Практически все программы восстановления могут сохранять информацию из потерянных кластеров в файл, а затем обнулять их.

Например, программа Chkdsk из цепочек потерянных кластеров создает файлы с именами FILE0001.CHK, FILE0002.CHK и т.д. Программа Chkdsk преобразует потерянные кластеры в файлы так, как показано в табл. 25.17.

Как видно из приведенного примера, оригинальное имя файла не восстанавливается. Однако его можно восстановить, просмотрев содержимое файлов, которые созданы программой восстановления диска.

## Пересекающиеся файлы

Такие файлы появляются, когда две записи каталога неправильно указывают на один кластер. В результате кластер “содержит” данные из нескольких файлов, что, естественно, недопустимо.

В табл. 25.18 приведен пример записи файловой системы с пересекающимися файлами.

В рассматриваемом примере два файла занимают два кластера — 1002 и 1003. Это пересечение файлов начинается с кластера 1002. Чаще всего один из пересекающихся файлов поврежден. Программы восстановления данных обычно решают проблему пересекающихся файлов следующим образом: файлы копируются с новыми именами в свободное место диска, а пересекающаяся область *обоих* файлов (и их остальные части) удаляется. Обратите внимание, что удаляются оба файла, т.е. устранение подобной ошибки не порождает новых проблем: например, запись в каталоге указывает на несуществующий файл. Просмотрев два восстановленных файла, можно определить, какой из них поврежден.

Для программ восстановления диска поиск пересекающихся файлов — очень простая задача, и практически все дисковые утилиты могут устранить эту проблему.

**Таблица 25.17. Потерянные кластеры найдены**

Каталог		
Имя	Начальный кластер	Размер
FILE0001.CHK	1000	4

FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

**Таблица 25.18. Пересекающиеся файлы**

<b>Каталог</b>		
<b>Имя</b>	<b>Начальный кластер</b>	<b>Размер</b>
Usconst.txt	1000	4
Pledge.txt	1002	2

<b>FAT 16</b>		
<b>Номер кластера</b>	<b>Значение</b>	<b>Назначение</b>
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

## Неверный файл или каталог

Иногда информация в записи каталога для файла или подкаталога не соответствует действительности: запись содержит кластер с неверной датой или неправильным форматом. Практически все программы восстановления диска устраняют и эту проблему.

## Ошибки FAT

Как уже указывалось выше, при повреждении основной FAT доступ к файлам осуществляется с помощью дополнительной FAT. Программы восстановления диска возвращают поврежденную FAT в ее оригинальное местоположение и активизируют зеркальное копирование. FAT 32 обладает большими способностями к восстановлению, поскольку в ней используются более развитые средства зеркального копирования.

Пример поврежденной FAT приведен в табл. 25.19.

В рассматриваемом примере размер файла в каталоге не соответствует количеству кластеров в FAT (потерян кластер 1001), а кроме того, кластеры 1002 и 1003 являются потерянными. При восстановлении данных поврежденная FAT восстанавливается из резервной копии. Практически все программы восстановления данных успешно справляются с таким типом ошибки FAT.

**Таблица 25.19. Поврежденная FAT**

Каталог		
Имя	Начальный кластер	Размер
Usconst.txt	1000	4

FAT 16		
Номер кластера	Значение	Назначение
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	0	Кластер доступен
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	0	Кластер доступен
...	...	...
65526	0	Последний доступный кластер

## Восстановление диска и данных

Команды `Chkdsk`, `Recover` и `Scandisk` — это “реанимационная бригада” DOS, занимающаяся восстановлением поврежденных данных на диске. Эти команды имеют очень простой и не слишком дружелюбный интерфейс, их применение зачастую оказывает значительное воздействие на систему, но иногда только они и могут помочь. Из перечисленных утилит наиболее известны, пожалуй, `Recover`, которая восстанавливает программы, и `Chkdsk`, используемая для проверки файловой структуры диска. Многие пользователи даже не подозревают, что `Chkdsk` может не только проверять, но и восстанавливать поврежденную файловую структуру диска. Еще одна программа — простая утилита `Debug` — может помочь вам в беде, но только в том случае, если вы точно знаете, что и как делать.

`Scandisk` — более мощная утилита, чем `Chkdsk` и `Recover`, заменяющая эти две утилиты в DOS 6 и более поздних версиях, а также в Windows 9x.

MS DOS 5.0 и более ранние версии поддерживают только две утилиты, используемые для проверки диска — `CHKDSK` и `RECOVER`. Использование и принципы работы команды `CHKDSK` рассматривается в 11-м издании книги, представленном на прилагаемом компакт-диске.

## Программа Scandisk

Программа `Scandisk` входит в поставку DOS версий 6 и более поздних, а также в Windows 9x. Она значительно мощнее утилит `Chkdsk` и `Recover` и выполняет функции их обеих. Программа `Scandisk` из Windows 95 OSR2 и Windows 98 может работать с FAT 32.

### **Замечание**

---

В операционных системах Windows NT 4.0, Windows 2000 и Windows XP используется программа CHKDSK, которая является более мощным аналогом утилиты SCANDISK. Для получения дополнительной информации о программе SCANDISK, используемой в MS DOS, обратитесь к 12-му изданию книги, представленному на прилагаемом компакт-диске.

---

Программа Scandisk больше похожа на упрощенную версию Norton Disk Doctor и позволяет проверять как целостность файловой структуры, так и работу секторов на физическом уровне. Обнаружив ошибки в каталогах или в FAT, Scandisk может их исправить. После определения дефектного сектора в FAT помечается дефектный кластер, содержащий этот сектор. При этом программа пытается восстановить поврежденный файл, причем сохраняются данные как до дефектного участка, так и после него.

В Windows 9x есть программа Scandisk для DOS и Windows. Файлы этих программ называются Scandisk.exe и Scandiskw.exe соответственно. Windows 9x проверяет диск в процессе установки операционной системы, а также после неверного завершения работы с системой. Вы можете также запустить программу Scandisk и ее “оконную” версию из командной строки.

Особенности работы программы Scandisk вы можете найти в книгах по операционным системам или в справочной системе Windows 9x.

## **Дефрагментация диска**

Как уже упоминалось выше, в файловой системе FAT данные в кластерах могут располагаться в любом месте диска. И при поиске файла последний считывается из нескольких мест, что, естественно, приводит к снижению производительности системы. Для перемещения файла в одно место служат программы дефрагментации диска.

В Windows 9x есть программа дефрагментации диска, которая работает с файловыми системами FAT 16 и FAT 32. В Windows 98 в программу дефрагментации была добавлена функция ускорения запуска приложений — перемещение часто запускаемых программ к началу диска.

Рассмотрим работу программы дефрагментации диска на примере. В табл. 25.20 приведены данные о расположении файлов в FAT.

В рассматриваемом примере файл Usconst.txt фрагментирован на две части. После запуска программы дефрагментации этот файл может быть расположен на диске так, как показано в табл. 25.21.

В процессе дефрагментации части файла, разбросанные по всему диску, были соединены. Дефрагментация диска — довольно продолжительный процесс, поскольку необходимо выполнить большое количество операций чтения и записи.

Кроме программы дефрагментации диска, поставляемой с операционной системой Windows 9x, существуют программы независимых разработчиков. Примером одной из таких программ может служить Speed Disk из комплекта Norton Utilities.

### **Внимание!**

---

Помните, что процесс дефрагментации — довольно опасная процедура. При дефрагментации происходит считывание, удаление и перезапись данных. Сбой в системе питания при выполнении этой операции может привести к плачевным последствиям. Поэтому перед дефрагментацией критически важных данных следует выполнять резервное копирование.

---

Операционная система Windows NT 4.0, в отличие от Windows 2000 или Windows XP, не содержит программу дефрагментации диска.

**Таблица 25.20. Фрагментированные файлы**

<b>Каталог</b>		
<b>Имя</b>	<b>Начальный кластер</b>	<b>Размер</b>
Pledge.txt	1002	2
Usconst.txt	1000	4
<b>FAT 16</b>		
<b>Номер кластера</b>	<b>Значение</b>	<b>Назначение</b>
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1004	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	1005	Используется; ссылка на следующий кластер
01005	FFFFh	Конец файла
...	...	...
65526	0	Последний доступный кластер

**Таблица 25.21. Дефрагментированный файл**

<b>Каталог</b>		
<b>Имя</b>	<b>Начальный кластер</b>	<b>Размер</b>
Pledge.txt	1004	2
Usconst.txt	1000	4
<b>FAT 16</b>		
<b>Номер кластера</b>	<b>Значение</b>	<b>Назначение</b>
00002	0	Первый доступный кластер
...	...	...
00999	0	Кластер доступен
01000	1001	Используется; ссылка на следующий кластер
01001	1002	Используется; ссылка на следующий кластер
01002	1003	Используется; ссылка на следующий кластер
01003	FFFFh	Конец файла
01004	1005	Используется; ссылка на следующий кластер
01005	FFFFh	Конец файла
...	...	...
65526	0	Последний доступный кластер

## Программы независимых разработчиков

Кроме стандартных программ для работы с диском, поставляемых с операционной системой, существует огромное количество дисковых программ независимых разработчиков. Самый известный пакет таких программ — Norton Utilities, разработанный компанией Symantec. Большинство подобных программ созданы для операционных систем DOS и Windows и могут работать с файловой системой FAT 32. Все эти программы имеют существенный недостаток — их необходимо приобретать дополнительно; однако, как правило, игра стоит свеч.

### Norton Utilities и Norton System Works

Утилита Norton Diagnostics, входящая в состав Norton Utilities, позволяет идентифицировать и проверить существующее аппаратное обеспечение, а также создать загрузочную дискету, используемую для перезапуска системы и тестирования жесткого диска при его выходе из строя, для восстановления ошибочно удаленных файлов и отмены случайного форматирования жесткого диска.

В настоящее время пакет Norton Utilities версии 2001 поставляется в качестве самостоятельного программного продукта или же в виде одного из компонентов программы Norton System Works 2001 (для операционных систем Windows 95 OSR 2.x, Windows 98, Windows Me, Windows NT и Windows 2000). Norton System Works 2001 также включает в себя Norton Anti-Virus и деинсталлятор Norton CleanSweep; в версию Professional вошли программа архивирования системной информации Norton Ghost, приложения WinFax Basic и Web Services. Некоторые из программ, вошедших в состав Norton Utilities, предназначены для запуска из командной строки или работы в режиме MS DOS:

- Norton Disk Doctor (NDD.EXE);
- Disk Editor (DISKEDIT.EXE);
- UnErase (UNERASE.EXE);
- UnFormat (UNFORMAT.EXE);
- Rescue Restore (RESCUE.EXE).

Тем не менее большинство программ, вошедших в Norton Utilities версии 2001, предназначены непосредственно для работы в графической среде Windows.

Не следует использовать более ранние версии Norton Utilities, например 8.0 (которая предназначена для работы в Windows 3.1 и MS DOS), с 32-разрядными версиями Windows, так как это может привести к потере данных из-за отсутствия поддержки длинных имен файлов и дисков большой емкости.

### Программа SpinRite 5

Тестовая программа Calibrate, существовавшая в ранних версиях Norton Utilities, не была включена в состав Norton Utilities 2001. Однако вместо нее можно воспользоваться программой SpinRite 5 от компании Gibson Research (<http://grc.com>), которая позволяет тестировать жесткие диски IDE и SCSI, а также накопители со сменными носителями любой емкости, имеющими файловую систему FAT, включая FAT32. Возможности технологии Dynastat Data Recovery, используемой SpinRite 5 для статистического анализа трудночитаемых секторов, позволяют определить их первоначальное содержание и переместить восстановленные данные в неповрежденную область диска. Это занимает больше времени, чем при использовании, например, программы DiskDoctor, но для восстановления данных существенно поврежденных дисков и разделов, не имеющих резервной копии, этот вариант наиболее приемлем.

SpinRite 5 также позволяет обработать таблицу разделов жесткого диска для определения его первоначальной структуры.

Приобретая программы для работы с жестким диском, следует обращать внимание, в первую очередь, на поддержку используемой файловой системы. Например, программа Norton Utilities 2001 поддерживает файловые системы FAT (включая FAT 32) и NTFS, в то время как ее ранние версии поддерживают только FAT 16. Программа SpinRite 5 поддерживает все разновидности FAT, однако не работает с NTFS. Обратите особое внимание на то, что никогда не следует применять программу для работы с диском, предназначенную для операционной системы более ранней версии. Это может привести к непоправимым последствиям.

## NTFS

Это собственная файловая система Windows NT, Windows 2000 и Windows XP. В Windows 2000 и Windows XP используется расширенная версия NTFS, получившая название NTFS 5 (NTFS 2000). Для того чтобы в Windows NT 4.0 получить доступ к разделам NTFS 5/NTFS 2000, необходимо установить Service Pack 4 или более поздней версии. Несмотря на то что операционные системы Windows NT/2000/XP поддерживают разделы FAT (Windows 2000/XP поддерживает и FAT32), файловая система NTFS обеспечивает целый ряд преимуществ: длинные имена файлов, поддержка файлов и разделов большого размера, дополнительные атрибуты файлов и расширенные средства безопасности. NTFS, как и все операционные системы Windows NT, создавалась на совершенно новой основе. Ее разработчики, увлеченно создавая новую 32-разрядную платформу, упустили из виду обратную совместимость с предыдущими операционными системами Microsoft. В результате разделы NTFS могут читаться только операционными системами Windows NT/2000/XP.

Все операционные системы Windows (кроме Windows NT/2000) основаны на DOS, поэтому внутри системы существует часть кода DOS. В Windows NT/2000 DOS-программы работают в режиме эмуляции DOS. При загрузке другой операционной системы раздел с NTFS недоступен.

В файловой системе имя файла может содержать до 255 символов, включая пробелы, точки и другие символы, кроме \* ? \ / ; < > |. Поскольку NTFS — 64-разрядная файловая система, размер файла и раздела может быть просто огромным —  $2^{64}$  байт (17 179 869 184 Тбайт)!

Начиная с Windows NT 3.51, файловая система NTFS также поддерживает сжатие файлов, осуществляемое с помощью списка свойств данного файла. Для сжатия или разархивирования файлов, хранящихся на диске NTFS, не нужны специальные программы наподобие WinZip или PKZip.

## Архитектура NTFS

Несмотря на существующие различия в структуре раздела файловых систем FAT и NTFS, они имеют подобные элементы, например загрузочную область. Раздел NTFS состоит из *главной таблицы файлов* (*master file table* — *MFT*). MFT — это не то же самое, что FAT. Вместо использования таблицы со ссылками на кластеры MFT содержит больше информации о файлах и каталогах в разделе. В некоторых случаях MFT может даже содержать файлы и каталоги.

При организации раздела NTFS система создает 10 системных файлов NTFS (табл. 25.22).

Первая запись в MFT называется *дескриптором* (*descriptor*) и содержит информацию о расположении самой MFT. Загрузочный сектор в разделе NTFS содержит ссылку на расположение записи дескриптора.

**Таблица 25.22. Системные файлы NTFS**

Имя файла	Назначение	Описание
\$mft	Master File Table (MFT)	Содержит запись для каждого файла в разделе NTFS в его атрибуте Data
\$mftmirr	Master File Table2 (MFT2)	Зеркальная копия MFT, используемая для восстановления
\$badclus	Файл поврежденных секторов	Содержит все поврежденные секторы раздела
\$bitmap	Карта распределения кластеров	Содержит карту всего раздела, указывающую на занятые кластеры
\$boot	Загрузочный файл	Содержит загрузочную информацию (если раздел загрузочный)
\$attrdef	Таблица определения атрибутов	Содержит определение всех системных и пользовательских атрибутов раздела
\$logfile	Файл журнала	Файл журнала транзакций, используемый для восстановления
\$quota	Таблица квот	Таблица квот пользователей на данном разделе (используется только в NTFS 5)
\$upcase	Таблица символов	Таблица, используемая для преобразования символов верхнего и нижнего регистров в символы верхнего регистра Unicode
\$volume	Раздел	Содержит информацию о разделе, например имя раздела и версию

Вторая запись в MFT — это зеркальная копия дескриптора. Такое избыточное хранение данных обеспечивает большую устойчивость к ошибкам.

Третья запись — это запись файла журнала. Все операции (транзакции) в NTFS записываются в специальный файл журнала, что позволяет восстановить данные после сбоя. Остальная часть MFT состоит из записей для файлов и каталогов, которые хранятся в разделе. В файле NTFS хранятся атрибуты, определенные пользователем и системой. Атрибуты в разделе NTFS — это не простые флаги из раздела FAT. Вся информация о файле, т.е. атрибуты, в NTFS сохраняется вместе с файлом и является частью самого файла. Каталоги в NTFS состоят в основном из индексов файлов в этом каталоге и не содержат такой информации о файле, как размер, дата, время и др.

Таким образом, MFT — это не просто список кластеров, это основная структура хранения данных в разделе. Если файл или каталог относительно небольшой (около 1 500 байт), его запись может храниться в MFT. Для больших массивов данных в MFT помещается указатель на файл или каталог, а сами данные располагаются в других кластерах в разделе. Эти кластеры называются *экстендами* (*extents*). Все записи в MFT, включая дескрипторы и файл журнала, могут использовать экстенды для хранения дополнительных атрибутов. Атрибуты файла, которые являются частью записи MFT, называются *резидентными* (*resident*), а атрибуты, расположенные в экстендах, — *нерезидентными* (*nonresident*).

## NTFS 5.0

В Windows 2000 используется новая версия NTFS — файловая система NTFS 5. При установке Windows 2000 все существующие разделы NTFS автоматически обновляются до NTFS 5. Если на компьютере также используется Windows NT (мультизагрузка), то необходимо установить пакет обновления Service Pack 4 (SP4) или последующий, чтобы эта система могла работать с разделами NTFS 5. В процессе обновления изменяется версия драйвера NTFS.SYS.

Файловая система NTFS 5 обладает несколькими новыми свойствами.

- *Квотирование диска.* Администраторы системы могут устанавливать для пользователей ограничения на использование диска. Эти квоты могут быть нескольких уровней: Off, Tracking и Enforced.

- *Шифрование.* NTFS 5 поддерживает автоматическое шифрование и дешифрование файлов при их записи и считывании с диска.
- *Особые объекты файловой системы.* Позволяют использовать точки монтирования, т.е. перенаправление записи и считывания данных из папки на другой раздел или физический диск.
- *Поддержка больших файлов.* Это свойство позволяет более экономно расходовать дисковое пространство.
- *Журнал номеров последовательных обновлений.* Обеспечивает ведение журнала всех изменений файлов раздела.

Обратите внимание, что большинство этих свойств поддерживается только операционной системой Windows 2000.

## Совместимость NTFS

Получить доступ к разделу NTFS из DOS и других операционных систем нельзя. Windows NT предназначена для использования в качестве сетевой операционной системы, поэтому доступ к файлам в разделе NTFS можно получить с помощью сети. Для этого в NTFS поддерживаются имена файлов, удовлетворяющие стандарту “восемь-точка-три”.

Основное преимущество NTFS — обеспечение безопасности файлов и каталогов. Атрибуты безопасности в NTFS называются *разрешениями (permissions)* и устанавливаются системным администратором посредством предоставления доступа к данным на уровне прав пользователей и групп пользователей.

Однако вы можете установить FAT-атрибуты файлов в NTFS с помощью стандартных инструментов операционных систем Windows NT/2000, например программы Windows NT Explorer или команды DOS `Attrib`. При копировании файлов из раздела NTFS в FAT все атрибуты файла сохраняются, и пользователь с правами полного доступа не сможет удалить файл с FAT-атрибутом “только для чтения”.

Алгоритм создания коротких имен файлов в Windows NT практически такой же, как и в файловой системе VFAT Windows 9x. Процесс создания имени файла, удовлетворяющего стандарту “восемь-точка-три”, для операционных систем Windows 9x, а также особенности этого процесса в Windows NT/2000 описаны выше в этой главе.

## Создание раздела NTFS

Создать раздел NTFS можно только на жестком диске. Его нельзя создать на дискете, а на сменном устройстве, таком, как Iomega Zip или Jaz, можно. Существует три способа создания раздела NTFS:

- при установке операционных систем Windows NT/2000 или после установки с помощью программ работы с диском;
- путем форматирования существующего раздела в NTFS (с удалением всех данных) с помощью команды `Format` систем Windows NT/2000 (параметр `/fs:ntfs`);
- посредством преобразования существующего раздела FAT в NTFS (с сохранением всех данных) при установке или после установки Windows NT/2000 с помощью программы `Convert`.

## Инструменты для NTFS

В связи с тем, что файловые системы NTFS и FAT различаются структурой, в NTFS нельзя использовать дисковые утилиты для FAT. Сама NTFS содержит средства восстановления данных. Кроме того, некоторые дисковые утилиты поставляются с Windows NT/2000. Последние версии программ для работы с диском, например Norton Utilities 2000, могут работать со всеми версиями Windows, значит, и со всеми операционными системами — FAT 16, FAT 32 и NTFS.

## Наиболее часто встречающиеся сообщения об ошибках

В этом разделе речь пойдет о наиболее часто встречающихся ошибках файловых систем и способах их устранения. Чаще всего могут появляться следующие системные сообщения об ошибках:

- Missing Operating System;
- NO ROM BASIC - SYSTEM HALTED;
- Boot error Press F1 to retry;
- Invalid drive specification;
- Invalid Media Type;
- Hard Disk Controller Failure.

### Missing Operating System

Такое сообщение об ошибке указывает на проблемы в главной загрузочной записи или записях таблицы раздела. Запись в таблице раздела может указывать на сектор, который не является началом раздела. Такая ошибка иногда появляется вследствие разряда батареи на системной плате, что приводит к удалению параметров BIOS.

Для решения проблемы прежде всего необходимо проверить правильность установки параметров в BIOS. Главную загрузочную запись можно восстановить с помощью команды `Fdisk /MBR`. В других случаях решить возникшую проблему можно с помощью радикальных средств — разбивки диска на разделы и форматирования с последующей повторной установкой операционной системы и необходимых приложений.

### NO ROM BASIC – SYSTEM HALTED

Эту ошибку генерирует AMI BIOS в случае повреждения или отсутствия загрузочного сектора или главной загрузочной записи на загрузочном диске. Кроме того, такая ошибка может появиться при неправильной установке параметров жесткого диска в BIOS. Для решения этой проблемы необходимо проверить параметры диска в BIOS или же восстановить главную загрузочную запись с помощью команды `Fdisk /MBR`.

## Boot error Press F1 to retry

Эта ошибка генерируется Phoenix BIOS при отсутствии жесткого диска или загрузочных областей. Наиболее частая причина появления этой ошибки — отсутствие активного раздела.

## Invalid drive specification

Такая ошибка появляется в том случае, если жесткий диск не разбит на разделы, записи таблицы разделов повреждены или же содержат неверные данные. Для устранения подобных проблем воспользуйтесь программой Fdisk либо программами пакета Norton Utilities.

## Invalid Media Type

При появлении такого сообщения скорее всего поврежден (или не инициализирован) загрузочный сектор, каталог или таблица размещения файлов. Например, такая ошибка появляется, если диск разбит на разделы, но не отформатирован с помощью команды Format. Для устранения этой ошибки необходимо использовать одну из программ восстановления диска или же просто отформатировать диск.

## Hard Disk Controller Failure

Такого типа ошибка появляется вследствие неверных параметров накопителя, установленных в BIOS, а также плохого подключения кабелей к накопителю или системной плате. Для устранения этой проблемы прежде всего проверьте подключение накопителя, а затем установите в BIOS его правильные параметры.

## Наиболее общие способы решения проблем с файловой системой

Чтобы устранить проблемы при доступе к жесткому диску, выполните ряд действий.

1. Загрузите компьютер с загрузочной дискеты (ее иногда называют аварийным диском). Это может быть загрузочная дискета как DOS, так и Windows, главное, чтобы на ней были записаны следующие программы: Fdisk.exe, Format.com, Sys.com и Scandisk.exe. Лучше, если это будет аварийный диск с операционной системой Windows 95B или более поздней.
2. Если с загрузочной дискеты нельзя загрузить операционную систему, скорее всего существуют проблемы с аппаратным обеспечением. В некоторых системах гибкий диск не определен как загрузочное устройство. В этом случае повторно запустите BIOS, укажите гибкий диск как первое загрузочное устройство и перезагрузите компьютер.
3. Запустите с загрузочной дискеты программу Fdisk. В меню выберите вывод сведений об имеющихся разделах (четвертый пункт меню).

4. Если отображается список разделов, проверьте наличие активного раздела: в столбце состояния возле одного из разделов должна быть буква А.
5. Если в списке не отображается ни одного раздела и вы не желаете восстанавливать данные на диске, создайте новый раздел (или разделы), а затем отформатируйте его. При выполнении этих действий все данные на диске будут уничтожены.
6. Если вам необходимо восстановить данные, воспользуйтесь одной из программ восстановления данных.
7. Если список разделов отображается и один из них активный, очевидно, повреждены системные файлы. Для их восстановления введите команду **Sys C:**.
8. Теперь ваш жесткий диск содержит системные файлы той операционной системы, которая была на загрузочном диске.
9. Извлеките дискету из дисковода и перезагрузите компьютер. Если и сейчас при загрузке появляются ошибки, они, скорее всего, вызваны неверной конфигурацией жесткого диска в BIOS.
10. Запустите программу Scandisk с загрузочного диска и проверьте диск на наличие ошибок.
11. При проверке диска с помощью программы Scandisk не забудьте выполнить проверку поверхности диска. При появлении большого количества поврежденных секторов необходимо заменить накопитель на жестких дисках.

## Решение общих проблем файловой системы в Windows 2000/XP

Способы решения проблем файловой системы в Windows 2000/XP практически те же, что и в Windows 9x. Их основным отличием является использование служебной программы Recovery Console, входящей в Windows 2000/XP.

- Если программа Recovery Console введена в загрузочное меню, запустите систему в обычном порядке, войдите в нее как администратор, если это необходимо, и выберите опцию Recovery Console.
- В том случае, если программа Recovery Console не была предварительно введена в загрузочное меню, загрузите систему с помощью установочного компакт-диска Windows или диска Windows Setup. Выберите опцию Repair в меню Welcome to Setup и нажмите клавишу <C> для запуска Recovery Console.

Если система не загружается с установочного компакт-диска или загрузочной дискеты, то возможны некоторые проблемы с аппаратным обеспечением. Проверьте жесткие диски, конфигурацию BIOS и установочные параметры системной платы. Определите гибкий диск как первое загрузочное устройство, а CD-ROM — как второе и перезапустите систему.

После запуска Recovery Console выполните ряд действий.

1. Для получения справки и списка команд Recovery Console введите HELP в командной строке.
2. Запустите программу DISKPART, чтобы получить сведения о существующих разделах диска.
3. Когда отобразится перечень разделов, проверьте, определен ли загрузочный раздел в качестве активного.

4. В том случае, если в списке не отображены имеющиеся разделы и у вас нет желания восстанавливать какие-либо данные, существующие в настоящее время на диске, создайте новый раздел (или разделы) с помощью команды FDISK. Для форматирования созданных разделов воспользуйтесь командой FORMAT. При выполнении этих действий все данные на диске будут уничтожены.
5. Если необходимо восстановить данные, воспользуйтесь одной из существующих программ восстановления данных, например Norton Utilities от компании Symantec или Lost and Found от PowerQuest.
6. Если при выполнении программы DISKPART отображается список разделов и один из них определен как активный, значит, возможно повреждение системных файлов. Для их восстановления введите команду FIXBOOT.
7. Для того чтобы перезапустить систему, введите команду EXIT. Перед этим извлеките загрузочную дискету из дисковода A: или же установочный компакт-диск Windows 2000 из дисковода CD-ROM.
8. Если и после перезагрузки компьютера появятся те же ошибки, то они, скорее всего, вызваны повреждением или неверной конфигурацией жесткого диска.
9. Перезапустите Recovery Console и проверьте диск на наличие ошибок, введя команду CHKDSK в командной строке.