

...SECRETS **INFO**
WORLD

Секреты жесткого диска



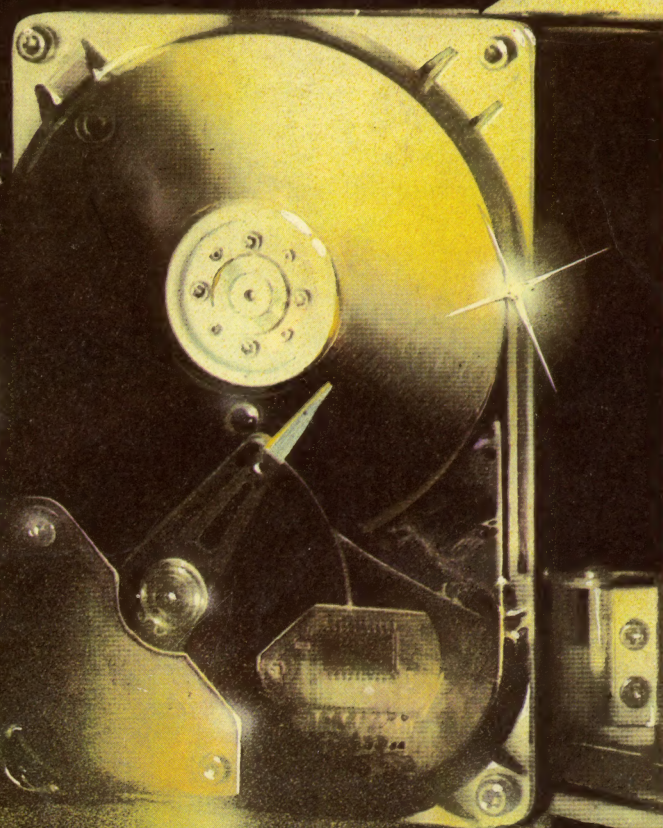
Лучшая серия
международных
бестселлеров
по компьютерам

Решение
проблем
еще до их
возникно-
вания

100
советов
и
секретов

Новинка!
DOS 6.X

Джон Гудмэн, доктор физики
Еще один бестселлер от
издателей "Секретов Windows 3.1"



роль
жестких
дисков

•
устройство
накопителей

•
кэширование
диска

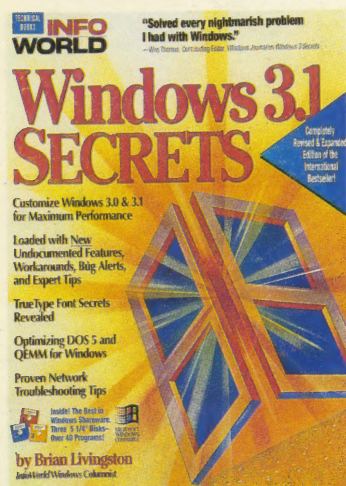
•
жесткие
диски и
DOS

•
подготовка
жесткого
диска

Только книги серии "Секреты..."

от "Диалектики" и ICE помогут вам досконально разобраться в секретах компьютерных технологий

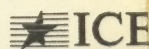
SECRETS™



Б.Ливингстон. Секреты Windows 3.1

384 с., мягкий переплет.
Известный бестселлер дает возможность вполне освоить и в дальнейшем использовать средства, предоставляемые графической средой Windows 3.1. Новичкам книга поможет подружиться с новой средой и стать грамотными пользователями, знающими Windows "снаружи" и "изнутри". Умудренные опытом пользователи Windows найдут много уникальной информации, которую невозможно получить даже в процессе длительного изучения документации.

СЕКРЕТЫ





Секреты жесткого диска

Джон Гудмэн

Совместное издание

"Диалектика" (Киев) и "Информейшн Компьютер Энтерпрайз" (Москва)

**Киев
1994**

ББК 32.973

Г21

Г21 Секреты жесткого диска

Киев: "Диалектика", 1994. — 256 с., ил.

ISBN 1-878058-64-9 (англ.)

ISBN 5-7707-6513-5 (рус.)

В книге приведены конкретные рекомендации для устранения проблем, возникающих при работе с жестким диском. Вице-президент Orange Coast IBM PC Users Group (OCIPUG) — одной из наибольших в США групп пользователей ПК, рассказывает о том, как наилучшим образом подготовить и использовать винчестер для хранения на нем информации. Приведено свыше ста практических советов и недокументированных возможностей. Рассмотрена DOS 6.

ББК 32.973

Группа подготовки издания:

Литературный редактор

Л.Н.Важенина

Технические редакторы

С.А.Чернокозинский, И.В. Родюк

Художественный редактор Е.Е.Полищук

Рассылка по почте А.Н.Тимченко

Секреты жесткого диска

Подписано к печати 22.07.1994. Формат 70х108/16. Бумага офсетная. Гарнитура Таймс. Печать офсетная. Усл.печ.л. 33.6. Заказ N0214071. Цена договорная.

Подготовлено к печати книжной редакцией фирмы "Диалектика" на базе лицензионного соглашения с АО "Информейшн Компьютер Энтерпрайз" (ICE).

© Джон Гудмэн, 1990, 1993

© В.Л.Григорьев, 1994, перевод, предисловие к русскому изданию

© фирма "Диалектика" и АО "Информейшн Компьютер Энтерпрайз" (ICE), подготовка и оформление, 1994

Украина, 252124 Киев-124, а/я 506.

Телефон/факс (044) 266-4074

E-mail:dial@dial.kiev.ua

ISBN 1-878058-64-9 (англ.)

ISBN 5-7707-6513-5 (рус.)

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства **IDG Books Worldwide, Inc.**

Отпечатано с готовых фотоформ

на комбинате печати издательства "Пресса Украины".

Украина 252047, Киев-47, проспект Победы 50.

Книги серии "Секреты..."™ от "Диалектики" и ICE

«Secrets...» («Секреты...») — это одна из всемирно известных книжных серий издательства IDG, представляемая вашему вниманию в переводе на русский язык фирмой "Диалектика" и ICE. Вы, несомненно, оцените доступные для понимания и исчерпывающие работы компьютерных экспертов. Каждая книга посвящена определенной области компьютерных знаний, а авторы — признанные авторитеты в этих областях — делятся с вами своим личным опытом, помогая решать повседневные задачи.

Суть любой книги из серии «Секреты...» проста — предоставить знающим специалистам лекционный зал и многомиллионную аудиторию, способную оценить и понять то, что им излагается. Каждый из авторов «Секретов» лучше, чем целое издательство, даст ответ на любой вопрос, заинтересовавший вас. В свое время авторы этих книг непосредственно общались с читателями на страницах журналов, беседовали в группах пользователей и вели консультационную работу. Такой тесный контакт дал книгам серии «Секреты...» преимущество перед многими изданиями подобного рода. Авторы не оторваны от реальных повседневных задач, наоборот они постоянно держат руку на пульсе времени.

Издательство уверяет, что квалификации авторов вполне достаточно для того, чтобы предложить наиболее эффективный способ решения возникшей задачи, и думает, что читателю будет полезна непосредственная беседа с "гуру" на протяжении всей книги. Авторский голос всегда чувствуется в серии «Секретов...». Материал в таких книгах преподносится так, что возникает впечатление, что вы встретились с автором в кафе и он разговаривает лишь с вами.

И, конечно, автор пользуется правом рекомендовать вам понравившееся ему программное или аппаратное обеспечение. Это логически связано с содержанием книги и, безусловно, заинтересует многих читателей.

Вам будет легко найти интересующую вас информацию независимо от того, читаете ли вы книгу от корки до корки, раздел за разделом, или просто просматриваете отдельную тему. Вы заслужили исчерпывающие и содержательные ответы на возникающие в процессе работы вопросы, именно их и дают вам книги серии «Секреты...» от фирмы "Диалектика" и ICE.

Предисловие к русскому изданию

Уважаемый читатель!

Вы держите в руках уникальную книгу. Ее уникальность объясняется, во-первых, рассмотрением вопросов, которые мало освещены в литературе, во-вторых, глубиной и всесторонностью изложения довольно сложного материала и, наконец, живой и образной манерой подачи материала, напоминающей беседу автора с читателем.

По-видимому, наиболее сложным, дорогим и несколько таинственным для пользователей устройством любого современного персонального компьютера является накопитель на жестких дисках. Все мы представляем важность и ценность этого устройства как огромного хранилища еще более ценных данных. Пока накопитель работает исправно, его не замечают, а узнать его истинную ценность можно, только когда он выходит из строя (дай Бог, чтобы этого вы подольше этого не узнали). А компьютер при этом становится практически бесполезным.

Автор, давно работающий на персональных компьютерах, обладает обширными сведениями о работе жестких дисков и их использовании в ПК. На страницах книги содержатся такие тонкости, которые доступны только знающему специалисту. Более того, в книге приведены практические советы и рекомендации, которые помогут отодвинуть неизбежную гибель накопителя и смягчить ее последствия. Да, об этом приходится помнить, так как, по образному выражению автора, есть только два типа жестких дисков — те, которые погибли, и те, которые погибнут.

В известном смысле эту книгу можно считать энциклопедией жестких дисков, и она по крайней мере в течение нескольких лет не утратит своей актуальности.

Книга в оригинале состоит из трех частей, мы публикуем только первую часть. Остальные части мы не печатаем по следующим причинам. Вторая часть книги посвящена дисковой утилите SpinRite (часть ссылок на нее в тексте сохранена), разработанной хорошим знакомым автора С.Гибсоном. По-видимому, для большинства читателей эта утилита недоступна и ее описание просто теряет смысл. Кроме того, практически все функции SpinRite реализованы дисковыми утилитами великолепных утилит Нортон, освоению которых будет способствовать настоящая книга. В третьей части содержится описание нескольких программ, помещенных на прилагаемой к оригинальной книге дискете¹.

При переводе книги я старался передать живой и образный стиль автора, возможно, несколько в ущерб технической строгости. В частности, я сохранил его манеру обращения к читателю, что у нас встречается очень редко.

Надеюсь, что великолепная книга Д.М.Гудмэна вызовет у вас живой интерес.

В.Л.Григорьев

¹ Эту дискету фирма "Диалектика" может предоставить дополнительно. За справками обращаться по тел.: (044) 266-40-74 или по адресу: Украина 252124 Киев-124, а/я 506, E-mail: dial&dial.kiev.ua

Введение

Прочитав введение, вы поймете, что изучение материала книги поможет сделать вашу жизнь с компьютерами лучше и легче, а ваши данные более защищенными. Во введении даются также советы о том, как прочитать книгу с минимальными усилиями и за кратчайшее время.

Кому предназначена книга

Нужна ли мне эта книга? — Это хороший первый вопрос о любой книге. Читая введение в книжном магазине, вы хотите как можно скорее узнать, что даст приобретение книги.

Книга написана для трех основных групп читателей, но четвертая группа также может прочитать ее, хотя книга не рассчитана на нее. Из дальнейшего материала вы поймете, к какой группе читателей вы относитесь.

Те, кому особенно нужна книга

В первую группу входят те читатели, которым содержащаяся в книге информация нужна немедленно. Читатели этой группы имеют персональный компьютер, жесткий диск и затруднения с жестким диском, облегчения которых они ищут. Если вы относитесь к этой группе, чтение книги окажет существенную помощь. Вы найдете в ней материал, который поможет вам идентифицировать и решить практически любую проблему жесткого диска.

Кроме того, вы найдете массу полезной и интересной информации о том, как работают жесткие диски. Наконец, вы узнаете некоторые приемы, как сделать будущие затруднения с диском менее серьезными.

Те, кто хочет знать все

Возможно, вы заинтересуетесь книгой, поскольку просто "пожираете" информацию. Вы хотите знать все! Конечно, вы не найдете здесь все обо всем, но в этой книге изложено почти все о жестких дисках персональных компьютеров.

Не имеет значения, кто вы — новичок или несколько лет уже работаете с компьютером. Для чтения этой книги потребуются самые умеренные начальные знания, хотя в книге довольно много подробностей, которые удовлетворят и технически подготовленного читателя.

Любознательные читатели

История о том, как разрабатывались жесткие диски для персональных компьютеров и как они работают, довольно интересная. Конструкция жестких дисков основана на достижениях физики, содержит оригинальные инженерные решения и учитывает экономические показатели. В эволюции жестких дисков встречались и неудачи, и значительные успехи, закономерные и хаотические.

Конечно, эта книга — не исторический трактат, но она предоставляет основную информацию, необходимую для понимания этих удивительных инженерных достижений. Приведено также несколько анекдотов о жестких дисках и их создателях.

Кто не имеет IBM PC или не работает с DOS

Часть материала этой книги вначале появилась в книге *Official SpinRite II and Hard Disk Companion*. Автор был приятно удивлен тем, что ее покупали и посчитали интересной даже владельцы компьютеров Macintosh и других не-IBM компьютеров.

Возможно, автору и не стоило удивляться! Жесткие диски в компьютерах Macintosh, Amiga, Atari, NeXT и других малых компьютерах похожи на жесткие диски в PC. По существу, жесткие диски мини-компьютеров и больших компьютеров одинаковы. В этих компьютерах отличается только программная поддержка операционной системы.

Если вы оказались в этой группе читателей, знайте, что материал глав 1–5 применим к любому жесткому диску почти любого современного компьютера; то же самое относится и к главам 8–11. Только четверть книги не рассчитана на такую широкую применимость. Если у вас не-DOS компьютер, даже в тех главах, которые автор не считает существенными, можно найти много положений, относящихся и к нему. Просто это сделано по-другому операционной системой вашего компьютера.

Что нужно знать предварительно

Вы не будете читать введение, не понимая языка, на котором оно написано. Это замечание относится к любой книге (исключая книжки-картинки). Чтобы получить от книги максимум полезного, у вас должны быть определенные знания и вы уже должны уметь делать нечто большее, чем просто читать.

Предположения автора

Автор полагает, что вы уже кое-что знаете о персональных компьютерах, а также, что вы заинтересованы получить больше информации об IBM-совместимых компьютерах и их жестких дисках.

В этой книге термин PC (Personal Computer, ПК — персональный компьютер) относится к компьютеру, аналогичному одной из моделей, производимых фирмой IBM. Это означает, что центральным процессором (ЦП) компьютера является одна из микросхем семейства 80x86 фирмы Intel или ее клон, производимый другими фирмами (AMD, Cyrix, Harris, NEC, Phoenix и др.).

Первые персональные компьютеры производились не фирмой IBM; их выпускали фирмы, о которых вы, возможно, и не слышали, например MITS, IMSAI и Southwest Technical Products. Но о Apple Computer вы, наверняка, знаете. Фирма Apple выпустила персональный компьютер Apple II в апреле 1977 г. Фирма Tandy ("родительская" для фирмы Radio Shack) начала поставлять свое семейство TRS-80 в июне 1977 г. Эти фирмы были самыми известными среди множества фирм, выпускавших малые компьютеры в конце 70-х и начале 80-х годов.

В августе 1981 г. фирма IBM выпустила свой первый персональный компьютер IBM Personal Computer, который вскоре стал известным просто как PC (ПК). Было продано столь много этих компьютеров, что IBM PC вскоре стал новым промышленным стандартом.

Фирма IBM подробно сообщила о том, как устроен IBM PC. За довольно умеренную плату фирма продавала полную техническую спецификацию компьютера. Фирма IBM предложила эту информацию в расчете на то, что другие фирмы начнут разрабатывать IBM-совместимые съемные платы и программы.

Многие фирмы вскоре создали такие изделия, что способствовало успеху ПК. Однако произошло и нечто неожиданное. Некоторые фирмы стали производить не только совместимые съемные изделия, но и клоны IBM PC и следующей модели PC/XT. Такие клоны или IBM-совместимые компьютеры были настолько похожи

на оригинальные модели, что определенный фирмой IBM стандарт стал повсеместно принятым промышленным стандартом.

Нам важны еще два определения: одно для DOS и одно для ПК. В этой книге DOS означает любую версию MS DOS, PC DOS или DR DOS, включая лицензионные версии MS DOS, созданные производителями IBM-совместимых персональных компьютеров.



Фирма Microsoft разработала PC DOS как оригинальную операционную систему для IBM PC. DOS (Disk Operating System — Дисковая операционная система) — это набор программ, которые управляют аппаратными средствами и позволяют выполнять прикладные программы, например текстовые процессоры, электронные бланки, коммуникационные программы и системы управления базами данных.

Фирма Microsoft продает лицензионный вариант PC DOS, называемый MS DOS, производителям IBM-совместимых компьютеров. Фирма Digital Research создала функционально эквивалентную систему DR DOS. (Почти для всего материала книги безразлично, с какой из этих трех операционных систем вы работаете.)

Таким образом, в этой книге под ПК понимается компьютер, который может выполнять операционную систему MS DOS. (Для ПК — это не единственная система; многие пользователи работают с UNIX, OS/2 или другой операционной системой, но их компьютеры по желанию владельцев могут работать и с MS DOS.)

Далее, автор полагает, что вы знаете, как включить ПК; что вы умеете выполнять простые команды и программы; что вы знаете, что такое приглашение DOS и как создавать каталоги и переходить к ним; что вы обладаете тем базовым искусством, которое необходимо для работы на ПК.

Конечно, вы можете что-то не знать из этого. Тогда почитайте некоторые рекомендуемые источники, чтобы извлечь из этой книги максимум полезного.

Нужно знать, как открыть компьютер

Фирма IBM спроектировала ПК с расчетом на то, что их владельцы время от времени будут открывать компьютер для доступа к внутренним компонентам даже без специальной технической подготовки. Производители большинства клонов также предвидят такую ситуацию и так же, как фирма IBM, предоставляют полную документацию о том, как открыть корпус и провести такие операции, как вставить или удалить съемные платы и дисковые накопители.

В книге не так много мест, где предлагается открыть компьютер. Если вы боитесь открывать ПК, пропустите соответствующий материал или попросите кого-нибудь помочь вам.

Если вы собираетесь открыть ПК и выполнить некоторые более сложные операции, внимательно прочитайте совет, как это сделать безопасно (материал в конце главы 1 о проверке новой загрузочной дискеты). Невозможно сломать ПК, печатая что-либо на клавиатуре, но его можно вывести из строя, совершая неправильные действия или правильные действия неправильными способами внутри системного блока.

Что делать, если вы чего-то не знаете

Необходимо знать хотя бы основы DOS, которые можно почерпнуть в книге Dan Gookin, "DOS for Dummies" или в книге John Socha и Clint Hicks, "PC World DOS 5 Complete Handbook".

В этой книге есть все, что нужно знать о подсистеме жесткого диска ПК (включая контроллер диска, кабели и собственно накопитель) и взаимодействии этой подсистемы с остальными компонентами. Информацию о других подсистемах ПК можно найти в великолепной книге Dan Gookin и Andy Rathbone, "PCs for Dummies"

(книга «ПК для "чайников"» готовится в издательстве "Диалектика" и, вполне вероятно, выйдет в свет одновременно с этой).

Как лучше всего пользоваться книгой

Оглавление книги специально сделано очень подробным, чтобы с его помощью можно было найти необходимую информацию.

Во настоящем введении показано, как пользоваться книгой и какую информацию в ней можно найти. В главе 1 преследуются две цели: показать как нужно читать эту книгу и дать важные советы о безопасности.

В главах 2–11 содержится все, что вам захочется узнать о жестких дисках ПК, включая принципы их работы, многочисленные причины их гибели и приемы восстановления после большинства таких неисправностей. (Когда жесткий диск "погибает", причиной служит какой-либо дефект; многие из этих дефектов стали исправимыми благодаря искусным инженерным приемам.)

Три способа первого чтения книги

Можно подумать, что у вас в руках только одна книга. На самом деле, ее можно считать тремя или даже четырьмя книгами под одной обложкой.

Первый маршрут по книге рассчитан на людей, которые имеют средний уровень знаний о жестких дисках, но хотят изучить их подробнее и располагают для этого временем. Маршрут таких людей очевиден — нужно просто читать книгу последовательно, раздел за разделом.

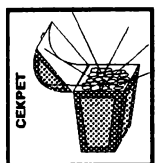
Второй маршрут предназначен тем, кто не очень хорошо знает ПК или не имеет много времени или терпения для изучения тонких деталей. Если вы относитесь к такой категории, просто пропускайте все абзацы "Технический секрет". В них содержится дополнительная информация для читателей, желающих глубже разобраться с жесткими дисками. С пиктограммы "Технический секрет" начинаются такие абзацы, а пропущенная строка показывает конец технического текста. Такое выделение удобно, если вы хотите пропустить технические подробности.

С другой стороны, специально выделив "Технические секреты", мы помогаем техническим специалистам, которые захотят воспользоваться третьим маршрутом по книге — выбрать новую и сугубо техническую информацию, которую можно почерпнуть из книги. Эти читатели будут действовать прямо противоположно читателям, идущим по двум первым маршрутам. Если вы относитесь к этой группе читателей, сразу идите к пиктограмме "Технический секрет", читайте материал, а затем переходите к следующим пиктограммам.

Мы показали три способа начального чтения книги. Четвертый способ — использование книги как справочника, когда вы читаете только то, что нужно. Автор полагает, что большинство читателей после первого знакомства с книгой по одному из трех рассмотренных способов будут пользоваться ею именно как справочником.

Принятые в книге соглашения

Мы уже рассмотрели оформление "Технических секретов". Кроме того, в книге приняты некоторые другие элементы, которые помогут в поиске конкретной информации. Приведенное ниже описание показывает, как оформлена пиктограммами такая "визуальная помощь".



Такая отметка показывает самые сложные технические детали и используемый жаргон. В зависимости от ваших намерений и уровня подготовки этот материал можно пропускать или внимательно изучать.



Этим мы предупреждаем об опасных ситуациях. Внимательно прочитайте все предупреждения! Содержащуюся в них информацию необходимо знать; в противном случае вы можете повредить свои данные или аппаратные средства компьютера.



Материал с такой пиктограммой содержит полезные приемы. Советы рассчитаны на всех читателей.



Этот материал вспомогательный, но довольно интересный. Читать его не обязательно, но при чтении вы получите определенное удовольствие.

Команды, которые вам предлагается напечатать, выделены особым шрифтом. Выводимые на экран сообщения также приведены специальным шрифтом.

Приглашение к сотрудничеству

Автор хотел бы знать мнение читателей о книге. Пожалуйста, напишите мне о всех ваших впечатлениях о книге, в том числе о любых возможных исправлениях.

Автор также хотел бы услышать от вас как приятные, так и ужасные истории о жестких дисках. Ему хотелось бы знать, за что иногда вы ненавидели свой жесткий диск, а также о том, что вы его полюбили, разобравшись, как работают накопители на жестких дисках в компьютере.

Адрес автора:

*John M. Goodman, Ph.D.
P.O. Box 746
Westminster, CA 92684-0746*

Глава 1. Значение жестких дисков

В этой главе вы узнаете

- Какова истинная ценность жесткого диска вашего персонального компьютера
 - Несчастные случаи — это не всегда катастрофы
 - Как защитить ценные данные
 - Как улучшить производительность жесткого диска
 - Как относиться к "гибели" жесткого диска — просто как к неприятности, а не как к катастрофе.
-

1.1. Жесткий диск — это не просто одно из периферийных устройств

Возможно, вы удивитесь, зачем целую книгу посвящать жестким дискам? Нужно ли на самом деле так много знать о них? Может быть, вам просто любопытно, как работают жесткие диски. В этом случае никакие аргументы вам не нужны, и вы можете сразу перейти к главе 2, но, пожалуй, не стоит это делать. Некоторая информация из этой главы потребуется вам для правильной защиты диска и данных.

По-видимому, вы сумеете работать, не вникая глубоко в проблемы жесткого диска вашего персонального компьютера. Однако, чем больше вы будете знать о его конструкции (и о том, как он может выйти из строя), тем успешнее устраните неисправность жесткого диска. И помните, неисправность когда-нибудь возникнет — возможно, в ближайшие дни.

Хуже того, возможно, этот день уже наступил. Тогда вы броситесь к главе 11. Потом вы вынужденно вернетесь к главе 1, так как именно в ней предполагается внимательное изучение материала. Здесь, в главе 1, вы сможете успокоиться, прийти в себя и найдете полезные сведения о постигшей вас неудаче. Конечно, в ПК может выйти из строя не только жесткий диск. Почему же гораздо важнее понимать потенциальные неисправности жесткого диска, чем, например, видеоплаты или платы порта принтера?

1.2. Жесткий диск — это самое дорогое устройство в ПК

Ответ на поставленный выше вопрос можно найти, узнав стоимость разных частей ПК. Чаще всего жесткий диск является наиболее дорогим компонентом ПК. Да, в некоторых компьютерах на СБИС стоимость материнской платы выше стоимости жесткого диска, особенно если емкость его невелика. Но даже в этом случае стоимость жесткого диска почти равна стоимости материнской платы.

С другой стороны, плату порта принтера можно заменить менее чем за 100 долларов (и часто за меньшую сумму). Но, может быть, на материнской плате вашего ПК есть СБИС с последовательными и параллельными портами. Даже в этом случае ваши перспективы не так печальны. Если отказал порт на материнской плате, обычно можно запретить его и заменить новой недорогой съемной платой.

1.2.1. Жесткие диски портятся чаще остальных компонентов

В дисководах жестких дисков имеются движущиеся части. После включения компьютера диски вращаются непрерывно. Разумеется, компоненты с движущимися частями изнашиваются быстрее компонентов без движущихся частей. Жесткие диски не будут исключением из этого правила.

Накопители на гибких дисках тоже имеют движущиеся части; однако они вращаются только при обращении к ним, а это происходит довольно редко. Единственной другой движущейся частью, которая часто отказывает в ПК, является вентилятор в блоке питания. (Когда вентилятор "умирает", блок питания быстро выходит из строя и его приходится заменять. Даже не думайте о его ремонте, потому что сейчас ремонт дороже замены.)

Кроме движущихся частей в ПК, часто отказывают большие электролитические конденсаторы в блоке питания. Срок службы этих конденсаторов намного сокращается при работе в условиях повышенной температуры, поэтому никогда не закрывайте вентиляционных отверстий ПК.



В некоторых портативных компьютерах есть специальные средства управления питанием. Обычно они предусматривают определенный временный интервал, по истечении которого жесткий диск "засыпает". Другими словами, ПК отключается, если вы не обращаетесь к нему дольше определенного времени. Если в вашем компьютере есть такое средство и вы много работаете от внутренних батарей, выигрыш от активизации этого средства будет достаточно велик. Но знайте, что это не обязательно продлит срок службы жесткого диска — он может даже сократиться. Самый тяжелый момент в "жизни" жесткого диска настает при его включении, поэтому целесообразно установить относительно продолжительный интервал для перехода в пассивное состояние, когда управление питанием отключает ПК.

1.2.2. Самое ценное на дисках

Жесткий диск — безусловно, дорогое устройство ПК, но еще ценнее хранящаяся на нем информация. Эта ценность определяется временем и работой, которые вы вложили в создание и сопровождение информации. Ваши данные незаменимы. Вы можете купить другую копию любой коммерческой программы, но полученные с помощью этих программ данные для вас уникальны.



Пожарные страховые фирмы свидетельствуют, что предприятия, пострадавшие от пожара, прекращают свою деятельность не потому, что лишаются своих построек и оборудования; они гибнут из-за потери информации (особенно, если теряют свои платежные документы).

Вашей первой линией обороны является поддержание хороших резервных копий ваших данных. Целесообразно также создать и хранить копии программ точно в таком же виде, как они инсталлированы на жестком диске. Что такое хорошая резервная копия, обсуждается в разделе 1.5.

Но иногда у вас нет надежных резервных копий, а жесткий диск работает хорошо. Только когда он гибнет, вы понимаете истинную ценность жесткого диска или, точнее, — содержащейся на нем информации. Если вы решите восстанавливать данные с действительно погибшего диска, вам придется прибегать к (очень дорогой!) помощи специализированных фирм.

1.3. Погибший жесткий диск можно исправить — иногда легко

Наиболее ценным в компьютере являются ваши данные, а эти данные находятся на жестком диске, поэтому при выходе его из строя вы можете запаниковать. Это чувство знакомо всем.

Конечно, когда вы абсолютно уверены в своевременности и надежности своих резервных копий, оставайтесь спокойными даже при серьезном отказе жесткого диска. Спокойствие можно сохранять и при не столь совершенных резервных копиях, просто зная, что они есть. Прекрасной новостью для вас должно послужить сообщение о том, что несложно исправить отказ накопителя. Не все неисправности диска фатальны!

Прежде всего, большинство неисправностей дисков связано с данными. А это означает, что испортился бит информации, хранимый где-то в ПК. Если этот бит исправить, ПК вновь заработает.

Конечно, сначала нужно узнать, связана ли поломка диска с данными или вызвана аппаратной неисправностью. Узнав тип встретившейся неисправности, необходимо определить, что конкретно с ней делать.

1.4. И небольшие знания помогут в беде

Неисправности жестких дисков и предпринимаемые при этом меры рассматриваются в главе 11. Лучше воспринимать этот материал вам помогут главы 2–10, особенно сведения, приводимые в последующих разделах этой главы. Ознакомившись с ними, вы сами разберетесь, какие другие сведения помогут вам в работе.

1.4.1. Небрежность плоха, но паника еще хуже

Чаще всего сбой ПК происходит, когда пользователь делает что-то ошибочное. Иногда жесткие диски выходят из строя и сами по себе, но гораздо чаще пользователь неосторожно удаляет файлы или делает другие, столь же простые ошибки.

Когда с вашим ПК что-то происходит, особенно когда вы не можете получить все данные, записанные на диске, вы обычно впадаете в панику. Если вы оказались в такой ситуации, встаньте, отойдите от ПК и успокойтесь. Пригласите приятеля и расскажите о ваших проблемах. Не злитесь и не упрямитесь. В паническом состоянии заниматься ремонтом нельзя.

Помните, что в большинстве случаев вы сами можете решить проблему или пригласить того, кто поможет вам. На самом деле серьезной причины для паники нет.

1.4.2. Не выплескивайте с водой ребенка

Большинство неисправностей в подсистеме жесткого диска ПК вызывается простыми ошибками. Например, простейшее изменение всего одного числа в определенном месте может сделать весь жесткий диск недоступным. Конечно, возникают и серьезные, катастрофические неисправности жесткого диска, но они относительно редко случаются.

Первая реакция человека, столкнувшегося с неисправностью жесткого диска, — предположение о наихудшем. Следующей в голову приходит мысль о необходимости форматирования низкого уровня. (Возможно, что эта мысль у вас никогда не появится — просто потому, что вы не понимаете смысла приведенной фразы. В главе 8 вы узнаете, что означает форматирование низкого уровня и почему не нужно его делать в такой ситуации.)

Вначале попробуйте простые приемы. И только когда это не помогло, переходите к среднему уровню и попробуйте средние по сложности приемы. Когда и они не помогают, переходите к решительным и сильнодействующим средствам.

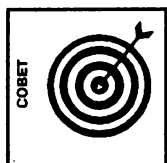
Одним из таких действий и является форматирование низкого уровня, прибегнуть к которому можно, только исчерпав весь арсенал более простых лекарств. Не выполнив всех предшествующих мероприятий, нельзя переходить к форматированию низкого уровня. (Более того, в некоторых дисковых накопителях форматирование низкого уровня невозможно.)

Последнее средство — заменить некоторые или все аппаратные компоненты подсистемы жесткого диска. Некоторые из этих компонентов дешевле других. Замена каких-то компонентов наверняка приведет к потере всех данных, а замена других позволит восстановить данные. По-прежнему, в успешном поиске и устранении неисправности помогут знания.

1.4.3. Великое дело — помощь

Иногда ваших знаний недостаточно для устранения возникшей неисправности. Вы всегда должны знать, как найти того, кто поможет.

Хорошо, когда вы знаете, к кому обратиться. Если не знаете такого специалиста, обратитесь за советом к своим приятелям. Может быть, в вашей фирме есть менеджер по ПК, — не стоит пренебрегать его советами.



Помимо общих советов при поиске специалистов по ПК, есть еще одна необычная, но хорошая рекомендация — обратитесь за помощью в группу пользователей ПК. Группы пользователей — это добровольные объединения людей с общими интересами. Среди членов группы есть люди, знающие больше или меньше вас. Поделитесь своими знаниями и поучитесь у тех, кто знает больше. Как и любая группа добровольцев, группы пользователей действуют лучше всего, когда каждый не только берет, но и отдает.

Многие группы пользователей организуют "электронные доски объявлений" (Bulletin Board System — BBS), бесплатные для членов группы или даже для всех. Пошлите объявление на доску и, очень вероятно, через несколько дней вы получите массу советов. Поскольку большинство пользующихся доской живут поблизости, вы сможете даже встретиться с откликнувшимся специалистом.



Будьте осторожны. Не все рекламирующие себя специалисты знают, как устранять затруднения с жестким диском. После изучения этой книги вы будете знать столько же или даже больше, чем большинство из них. Прислушайтесь к полученному совету: подробно расспросите человека, которому вы доверяете, о том, какие приемы он рекомендует и почему. Если вы не понимаете сказанного, поищите кого-то еще, прежде чем разрешать портить ваши ценные данные.

Если вы покупали компьютер у дилера, обратитесь к нему за помощью или рекомендациями. Когда вы уверены, что неисправность ПК связана с жестким диском, фирма-производитель сможет помочь вам или направить к специалисту. Позвоните в отдел технического обслуживания фирмы или проверьте информацию на доске объявлений фирмы.

Еще одним ценным помощником служит информационное обслуживание Compu-Serve, на форумах которого вы можете задать вопросы и получить ответы от весьма компетентных людей. (Многие производители выступают спонсорами форумов Compu-Serve. Здесь вы можете получить помощь от тех, кто разрабатывали накопитель, ПК или программы. Можно также послушать специалистов, не связанных с произ-

водителями аппаратных и программных средств. Внешние специалисты дадут совет, не связанный с интересами тех людей, которые работают у производителя.)

Если вы застенчивы, посмотрите вопросы других пользователей. Возможно, ваш вопрос уже возникал и на него готов ответ. CompuServe оказывает платные услуги, но полученная вами информация оправдывает расходы.

1.5. Данные нельзя застраховать — их нужно защищать самому

Теперь вы знаете, что данные на жестком диске ценнее самого накопителя. Для защиты вложений в аппаратные средства можно прибегнуть к страхованию. Некоторые страховые фирмы защищают даже вложения в коммерческие программы. Но свои программы и свои данные вы должны защищать сами. Никакая страховая фирма не поможет вам в случае их потери.



Даже гибкие диски выходят из строя. Хранение важных файлов только на гибких дисках не решает проблемы, если не иметь нескольких копий, находящихся в разных зданиях.

1.5.1. Все диски портятся — никто не знает, когда это случится

Если у вас есть только одна копия данных и она хранится на жестком диске, когда-нибудь вы потеряете свои данные. Знайте об этом и помните следующее изречение: "Есть только два типа жестких дисков — те, которые погибли, и те, которые погибнут".

1.5.2. Будьте готовыми — имейте надежный резерв

Самая лучшая защита — иметь надежные резервные копии всех файлов программ и данных. Тогда, в случае гибели жесткого диска вы всегда сумеете восстановить всю информацию, которую жесткий диск "отправил на небеса".



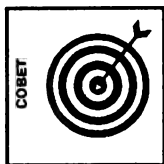
Не станьте жертвой ложной безопасности. Проверяйте работоспособность своих резервных копий периодически. Слишком часто их не проверяют до того, как они потребовались. А затем обнаруживается, что нельзя восстановить все файлы (или их часть) с резервных копий. Если нельзя восстановить всего один файл и если он один из самых нужных, смысл резервирования теряется.

Конечно, не нужно сразу мириться с мыслью о том, что ваш жесткий диск погиб. Не следует выбрасывать то, что можно отремонтировать. Прежде всего убедитесь в том, что жесткий диск оживить невозможно.

Если после тщательной проверки вы пришли к выводу, что жесткий диск погиб, его можно заменить новым. (Приятная новость: по-видимому, вы сможете купить жесткий диск большей емкости, чем старый, по цене, не превосходящей цену старого диска.)

После установки нового жесткого диска в ПК нужно только восстановить программы и данные из (предположительно почти надежных) резервных копий. После этого компьютер готов к работе. При наличии надежного резерва гибель жесткого

диска представляется дорогой неприятностью, но не более того. Однако без такого резерва неисправность жесткого диска может вызвать все — от головной боли до полной катастрофы.



Воспользуйтесь этим случаем и реорганизируйте жесткий диск. Вполне вероятно, что у вас появились лучшие идеи на этот счет с тех пор, когда вы готовили предшествующий диск.

На рис. 1.1 показано, что надежные резервные копии обеспечивают восстановление после отказа жесткого диска за минимальное время и с минимумом огорчений. Скопируйте рис. 1.1 и повесьте его на видном месте.

В главе 11 рассмотрены различные способы резервирования и их применение. Там же указаны программы, полезные для создания и сопровождения ваших резервных копий.

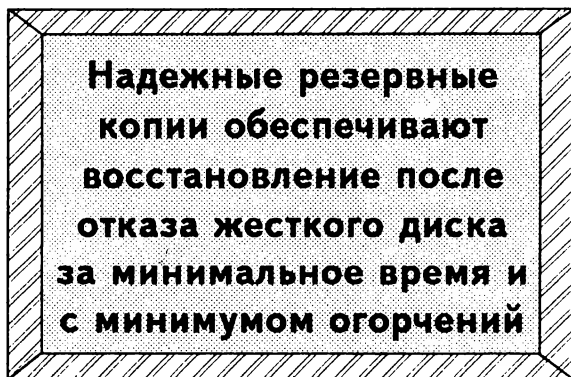


Рис. 1.1. Повесьте это ценное изречение над компьютером

1.5.3. Способы защиты от ошибок

Потери данных в большинстве случаев вызываются не аппаратными отказами в накопителе, а ошибками пользователя. Для минимизации таких ошибок полезно выработать несколько простых привычек.

Пользуйтесь программами, которые помогают вам делать то, что нужно, и не делать того, что не нужно. Одной из таких программ, по мнению многих, является программа оболочки DOS. Такая программа DOSSHELL поставляется, начиная с DOS 4. Однако только с DOS 5 эта программа стала удовлетворять пользователей. К счастью, есть много великолепных оболочек, например XTreeGold, других фирм.

При работе с Windows хорошие услуги оказывает менеджер файлов (File Manager). В версии Windows 3.0 его возможности ограничены, но в Windows 3.1 менеджер файлов улучшен настолько, что вам не потребуется какая-нибудь замена. Но при желании можно найти и другие оболочки, например Norton Desktop для Windows.



При использовании программы оболочки DOS или менеджера файлов Windows не забудьте включать подтверждение удаления файла или каталога.

Сохраняйте копии критических информационных структур диска. Для этого есть несколько способов. Начиная с версии 5, в DOS для этого появилась программа MIRROR. Без указания параметров

MIRROR сохраняет большую часть критической информации. (Необходимо выполнять MIRROR для каждого логического накопителя.) При задании опции /PART программа MIRROR сохраняет еще одну часть критической информации. Наконец, при использовании резидентной программы с опцией /Td (где d — буква накопителя) MIRROR поддерживает сохраняемую информацию свежей.

В DOS 6 программа MIRROR убрана, а большинство ее функций реализует программа UNDELETE. Однако средство сохранения таблицы разделов (/PART) отсутствует.

Наконец, внимательно изучите и используйте рекомендации из раздела 1.7 для создания безопасной загрузочной дискеты.

1.6. Как заставить жесткий диск работать лучше и безопаснее

Закрепив хорошие привычки для защиты файлов, можно переходить к некоторым шагам, которые помогут повысить производительность при работе с файлами и ускорить операции DOS.

1.6.1. Хорошая организация помогает вам и DOS

Организация жесткого диска — довольно сложный вопрос, только косвенно относящийся к материалу этой книги. Написано много книг и статей о том, как организовывать программы и данные на жестком диске. Автор хочет только подчеркнуть важность этого и предложить одну из великолепных книг по этому вопросу: Beth Slick, *Official XTree MS-DOS, Windows, and Hard Disk Management Companion*.

Упоминаем здесь об это только для того, чтобы рассмотреть его до принятия обоснованного решения об организации разделов на жестком диске. Подробнее об организации разделов речь идет в главе 9, но в ней рассмотрены не все вопросы, относящиеся к определению размеров разделов жесткого диска. Необходимо учитывать также то, как вы собираетесь организовать свои файлы и обращаться к ним.

При работе на ПК и накоплении все большего числа файлов программ и данных станет необходимым изменить организацию разделов жесткого диска. В этом случае внимательно изучите главу 9 или другие книги по организации жесткого диска. Эта информация поможет изменить разделы самым простым и эффективным способом.

Имеются несколько приемов управления файлами. Например, можно организовать файлы в осмысленные группы и поместить каждую группу в самостоятельный подкаталог. Можно также сгруппировать родственные подкаталоги в другой отдельный подкаталог.

Интересно отметить, что такая организация не только помогает вам следить за файлами, но и помогает DOS быстрее искать и считывать файлы. Подробнее об этом сказано в главе 10.

1.6.2. Помогает и ускорение диска

Для достижения оптимальной производительности, кроме организации файлов и структуры каталогов еще две операции полезны. Одна из них состоит в периодической дефрагментации файлов, а вторая — в установке правильного коэффициента чередования секторов (и, возможно, перекоса головки и цилиндра). Дефрагментация диска — это переупорядочение блоков хранимой информации так, чтобы все части файла находились рядом друг с другом. Чередование и перекос относятся к нумерации физических секторов по одной дорожке и изменению нумерации при переходе с дорожки на дорожку.

Дефрагментация диска рассматривается в главе 11. Там же указаны некоторые хорошие программы, помогающие дефрагментировать жесткий диск.

Чередование секторов и перекос головки и цилиндра являются довольно сложными вопросами. Наверное, раньше вы и не слышали о них. Однако важно разобраться в этом и, к счастью, сделать это очень просто. В главе 3 они изложены на уровне, понятном любому читателю, хотя для этого требуется внимательное чтение и размышление.

1.7. Некоторые предложения для восстановления

Повторим еще раз: если ваш жесткий диск еще не погиб, готовьтесь к тому, что он погибнет. В конце концов, гибнут все жесткие диски. Даже если вы счастливчик и ваши диски (пока) работают хорошо, прочитайте данный раздел и воспользуйтесь приведенными в нем рекомендациями.

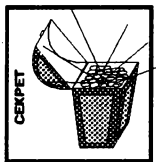
1.7.1. Готовьтесь — создайте безопасную загрузочную дискету

Важным средством в вашем арсенале для работы с жестким диском ПК является безопасная загрузочная дискета. Один из приятелей автора называет ее "хранительницей счастья". Но и без названия безопасная загрузочная дискета оказывается удивительным средством.

Безопасная загрузочная дискета — это дискета, с которой можно загрузить компьютер независимо от того, что случилось с жестким диском. Эта дискета обеспечивает также доступ к жесткому диску при всех неисправностях, кроме самых серьезных. В простейшей форме безопасная загрузочная дискета представляет собой загрузочную дискету DOS с некоторыми дополнительными файлами; однако ее роль значительно важнее.

Начните с системных файлов. Как вы знаете, дискета является загрузочной, если она правильно форматирована и содержит системные файлы. (Системные файлы — это существенные элементы DOS, которые содержатся в специальных файлах на диске. Эти файлы вводятся в каталог так, что команда DIR их не показывает. Можно убедиться в наличии системных файлов, выполнив команду CHKDSK и посмотрев ее отчет о скрытых файлах.)

Проще всего создать загрузочную дискету с помощью команды `FORMAT A: /S`. Если вы создаете такую дискету после загрузки компьютера с жесткого диска, можно быть уверенным, что версия MS DOS на дискете та же самая, которая применяется для загрузки. Точное соответствие имеет очень важное значение.



Загрузочная дискета DOS имеет, как минимум, загрузочный сектор, два скрытых системных файла и командный интерпретатор — все созданные одной и той же версией DOS. Если вы форматировали дискету и превратили ее в загрузочную с помощью MS DOS или большинства ее клонов, скрытые файлы называются MSDOS.SYS и IO.SYS. При использовании PC DOS или DR DOS эти файлы называются IBMBIO.COM и IBMDOS.COM. В некоторых клонах MS DOS применяются другие имена.

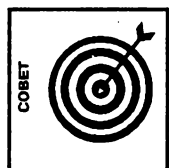
Загрузочная запись создается при форматировании диска. Она содержит имена системных файлов, которые находятся на диске. Если просто форматировать дискету, не помещая на нее системных файлов, можно превратить дискету в загрузочную позднее, копируя эти файлы на дискету.

Можно добавить системные файлы командой `DOS SYS`, но помните о следующем. Если вы использовали одну марку DOS для форматирования и другую — в команде `SYS`, дискета может оказаться не загрузочной, если имена системных фай-

лов в двух марках DOS отличаются. Поскольку вы не видите имен файлов в листинге команды DIR, эту ошибка может остаться незамеченной. В результате с безопасной загрузочной дискеты загрузиться нельзя, и на выяснение причин этого уйдет много времени.

Во всех версиях DOS командным интерпретатором служит файл COMMAND.COM, хотя его можно заменить несколькими другими, например 4DOS.COM и NDOS.COM из утилит Нортон (версия 6+). Для замены командного интерпретатора на загрузочном диске нужно иметь файл CONFIG.SYS, в котором оператор SHELL определяет имя и местонахождение командного интерпретатора.

На безопасной загрузочной дискете должен находиться обычно используемый вами командный интерпретатор. Конечно, это требование подразумевает, что при работе не с копией COMMAND.COM в корневом каталоге, на безопасной загрузочной дискете должен находиться файл CONFIG.SYS.



Будьте готовы к катастрофе

Вы хотите, чтобы безопасная загрузочная дискета отвела вас от катастрофы. Создавая эту дискету, не тратьте времени на нестандартные приемы сомнительной надежности!

Если у вас есть накопитель А малой емкости (который работает только с дискетами 360 Кбайт диаметром 5.25" или 720 Кбайт диаметром 3.6"), нужно взять дискету малой емкости. При наличии накопителя А большой емкости (который работает с дискетами 1.2, 1.44 или 2.88 Мбайт) вы можете задуматься о том, что взять для безопасной загрузочной дискеты — дискету малой или большой емкости; в таком случае читайте дальше.

Накопители большой емкости допускают дискеты малой емкости, они работают с ними не так надежно, как с теми дискетами, на которые сконструированы. (Еще хуже применять дискету малой емкости как дискету большой емкости. Некоторые накопители допускают и это, но надежность работы невысока.)

При создании безопасной загрузочной дискеты важно работать с накопителем А. Если накопитель В имеет другую емкость, подготовленный в нем диск в последующем нельзя использовать для загрузки компьютера, так как обычно ПК загружается либо с накопителя А (это первый накопитель на гибком диске), либо с накопителя С (первый накопитель на жестких дисках). Если при одинаковой емкости накопителей А и В вы захотите воспользоваться накопителем В, то трудно предусмотреть, каким будет результат — благополучным или нет. Дело в том, что некоторые накопители неправильно выровнены. Дискету, записанную в таком накопителе, считать в нем же можно; однако номинально идентичный другой накопитель может и не считать ее.

Добавление некоторых специальных файлов. Кроме необходимых для загрузки системных файлов, на безопасной загрузочной дискете должны находиться специальные версии файлов CONFIG.SYS и AUTOEXEC.BAT, обычно применяемые на жестком диске. Эти файлы проводят адаптацию DOS при каждой загрузке. Их следует поместить на безопасную загрузочную дискету, чтобы DOS была конфигурирована ожидаемым образом и обеспечила оптимальное восстановление.

Создаваемые файлы, конечно, похожи на оригинальные файлы, но имеют два важных отличия.

- ✓ ПК целесообразно привести в самое простое состояние. Максимально упростите файлы CONFIG.SYS и AUTOEXEC.BAT, помещаемые на безопасную загрузочную дискету. Не вызывайте сложного менеджера памяти. Загружайте только действительно необходимые драйверы устройств и резидентные программы
- ✓ Вы хотите воспользоваться безопасной загрузочной дискетой для запуска компьютера даже при полностью недоступном жестком диске. Для этого все файлы, которые вызываются в CONFIG.SYS и AUTOEXEC.BAT, должны находиться на

безопасной загрузочной дискете. Более того, маршруты обращения к этим файлам должны показывать на накопитель А

Заполнение дискеты. Заполните оставшееся место на безопасной загрузочной дискете полезными утилитами. Среди наиболее полезных отметим программы DOS CHKDSK, FDISK и FORMAT. (Проверьте, они должны быть той же версии DOS, что и в скрытых системных файлах.) Если все это не помещается на безопасную загрузочную дискету, возьмите еще одну дискету (подробнее об этом говорится в разделе 1.7.2).

Проверка безопасной загрузочной дискеты. Последних три этапа создания безопасной загрузочной дискеты очень важны: защита от записи, проверка и образование копии с помощью команды DISKCOPY (копию также нужно защитить от записи). Держите одну копию около ПК, а вторую в другом здании. Не пропускайте ни одного этапа. До завершения всех этапов ваш уровень защиты минимален.

Проверяя безопасную загрузочную дискету загрузите с нее компьютер. Но если у вас типичный настольный ПК, можно поступить проще. (Дальнейшие инструкции предполагают, что у вас есть клон ПК обычной конструкции. Если у вас имеется IBM PS/2 или другая машина с архитектурой Микроканала, некоторые детали удаления или отключения питания жесткого диска немного отличаются. Посмотрите документацию вашего ПК.)



Перед работой внутри системного блока ПК проверьте парковку головок жесткого диска, а затем выключите компьютер. Отключите все кабели и шнуры питания от системного блока. Затем аккуратно откройте системный блок. (Причина отключения всех кабелей связана с тем, чтобы исключить прохождение электричества через ПК на землю и защитить вас от сетевого напряжения. Этим вы защищаетесь от электрического удара, а ПК от повреждения статическим электричеством.)

При открытом системном блоке ПК более уязвим. Прежде чем залезать внутрь системного блока, обязательно прикоснитесь к блоку питания или шасси ПК перед касанием чего-нибудь еще. Этот прием гарантирует, что заряд статического электричества безопасно ушел на корпус ПК и не повредит микросхемам. Если разряд пройдет через микросхему, она выйдет из строя.

Если ваш единственный накопитель на жестком диске смонтирован на плате (или если это накопитель С), отверните фиксирующий винт и аккуратно выньте плату жесткого диска из разъема. Если же накопитель смонтирован отдельно от съемных плат, просто отсоедините кабель, связывающий накопитель с блоком питания (нужно отсоединять этот кабель на накопителе, так как второй его конец постоянно прикреплен к блоку питания).

Вставьте безопасную загрузочную дискету в накопитель А и включите компьютер. Убедитесь в том, что компьютер загружается правильно и никакие сообщения об ошибках при этом не появляются. Внимательно посмотрите сообщения о том, что DOS не может найти какой-нибудь драйвер устройства. (В компьютере AT будет выдано сообщение "Bad or missing C drive" — раз вы удалили или запретили жесткий диск, следует ожидать этого сообщения.) По окончании проверки выключите питание, восстановите кабель питания жесткого диска и закройте ПК.

Следующие этапы детализируют процедуру создания безопасной загрузочной дискеты. Выполните их, а затем всегда держите эту дискету под рукой.

1. Загрузите компьютер с жесткого диска и выйдите на стимул DOS для накопителя С.
2. Вставьте дискету в накопитель А. Применяйте дискету, которая точно соответствует емкости этого накопителя.

3. Напечатайте `FORMAT A: /S /V` и нажмите клавишу `Enter`. Назовите дискету каким-либо именем. (При работе с DOS 5 добавьте двоеточие и имя после `V` для придания диску метки тома. В прежних версиях DOS при указании ключа `/V` без имени или двоеточия DOS стимулировала ввод имени после окончания форматирования.).
4. Создайте на дискете минимальные файлы `CONFIG.SYS` и `AUTOEXEC.BAT` (помните, что ваша цель — обеспечить доступ к жесткому диску. Укажите только нужные для этого резидентные программы и драйверы устройств, а остальные не включайте) Введите в файл `AUTOEXEC.BAT` такие строки:

```
PATH=A:\DOS;A:\UTIL;A:\VIRSTUFF
PROMPT $P $G
```

5. Создайте на гибком диске каталоги `DATA`, `DOS`, `UTIL` и `VIRSTUFF`.
6. Скопируйте программы `DOS CHKDSK`, `FDISK` и `FORMAT` в каталог `A:\DOS`.
7. Скопируйте программы `MH-SYS` и `MH-RESTR` с приложенного жесткого диска в каталог `A:\UTIL`. Скопируйте также в каталог `A:\UTIL` программу `Disk Editor` из утилит Нортон.
8. Скопируйте свою любимую антивирусную программу в каталог `A:\VIRSTUFF`.
9. Напечатайте команду `MH-SAVE /BA:\DATA\DATA.FIL /C` и нажмите клавишу `Enter`. (Предполагается, что программы приложенного диска помещены в каталог жесткого диска, включенный в `PATH`.) Можно также заставить программу `MH-SAVE` поместить на дискету больше информации о жестком диске.
10. Скопируйте другие утилиты по своему выбору в каталог `A:\UTIL` до заполнения дискеты.
11. Проверьте свою безопасную загрузочную дискету.
12. Создайте резервную копию дискеты с помощью команды `DISKCOPY`.
13. Защитите обе дискеты от записи.
14. Положите одну копию рядом с ПК, а другую отнесите в другое здание.
15. Обновляйте свою безопасную загрузочную дискету (выполняя приведенные этапы) при переходе на новую марку или версию DOS, используемую на жестком диске. Вам придется также обновлять файл `DATA.FIL` хотя бы тогда, когда проводится значительное изменение аппаратной конфигурации (при изменении содержимого КМОП-ЗУПВ). Если программа `MH-SAVE` сохраняла больше информации о жестком диске, обновляйте файл данных при изменении тех параметров, которые сохранены о жестком диске.
16. Повторите приведенную процедуру для каждого имеющегося у вас ПК. Для каждого ПК требуется индивидуальная загрузочная дискета.
17. Когда ПК не загружается с жесткого диска, возьмите свою безопасную загрузочную дискету. После этого откройте главу 11 и посмотрите, как определить возникшую у вас неисправность и исправить ее.



DOS и OS/2

Некоторые устанавливают на своем ПК операционные системы DOS и OS/2. Начиная с OS/2 2.0, такое инсталлирование можно провести двумя совершенно различными способами. Фирма IBM называет один из способов `Dual Boot` (Сдвоенная Загрузка), а второй

— Boot Manager (Менеджер Загрузки). Различие оказывается весьма существенным, особенно в контексте безопасной загрузочной дискеты. Если вы применяете способ IBM Dual Boot, ваш ПК всегда загружает ту операционную систему, с которой он последний раз работал. Если ваш жесткий диск погибает, когда текущей операционной системой является OS/2, и особенно если вы установили OS/2 в логическом накопителе HPFS (High Performance File System — высокопроизводительная файловая система), вам потребуется вторая безопасная загрузочная дискета, содержащая важнейшие загрузочные файлы OS/2. (Помогают создать такую дискету файлы CMD, аналогичные пакетным файлам DOS, от фирмы IBM и различных досок объявлений.)

Если вы применяете способ IBM Boot Manager (его можно использовать для нескольких версий DOS, а также для DOS и OS/2; для него есть много программ), можно не беспокоиться. Для восстановления из неисправности жесткого диска достаточно одной безопасной загрузочной дискеты, рассчитанной на DOS.

1.7.2. Держите другие полезные дискеты под рукой

По-видимому, все полезные утилиты не поместятся на одну дискету, независимо от ее емкости. Очевидно необходимо подготовить комплект дискет, рассчитанный на аварии жесткого диска. В такой комплект обязательно входит безопасная загрузочная дискета для конкретного ПК.

Целесообразно включить в аварийный комплект утилиты Нортон. (В версии 5+ имеется Disk Editor — великолепное средство анализа гибких и жестких дисков и исправления многих их сбоев. Если вы умеете пользоваться Disk Editor, то его ценность оправдывает запись этой программы на безопасную загрузочную дискету.) Из других утилит Нортон в аварийный комплект автор рекомендует включить Disk Doctor, File Fix, UnErase и UnFormat.

Полезно иметь полный набор дискет DOS, включая файлы всех внешних команд DOS. Согласно ранее приведенным советам вы уже скопировали на безопасную загрузочную дискету программы CHKDSK, FDISK и FORMAT. Большинство файлов остальных внешних команд также могут оказаться полезными, например программы DISKCOPY, COMP, DISKCOMP и SYS, а для пользователей DOS 5 — программы MIRROR, UNFORMAT и UNDELETE.

Наконец, рекомендуется иметь самую свежую копию хорошей антивирусной программы, например VIRUSCAN фирмы McAfee. Большинство неисправностей жестких дисков вызываются не вирусами, но некоторые связаны именно с ними. Можно проверить, заражен ваш ПК или нет. Это либо успокоит вас (если не заражен), либо подскажет, что делать дальше (если заражен). Почаще обновляйте эту программу. Большая часть антивирусных программ основана на знании текущей сигнатуры компьютерных вирусов и при старении становятся почти бесполезными.

1.7.3. Создание безопасной загрузочной дискеты после гибели жесткого диска

Что делать, если ваш жесткий диск погиб до того, как вы прочитали эту главу? Неужели вы такой невезучий? Это совсем не обязательно. Просто ваша работа окажется потрудней (и придется поискать несколько дополнительных сведений о ПК).

Прежде всего, знаете ли вы версию DOS, с которой вы работали до гибели ПК? Найдите другой ПК, работающий с такой же версией, и создайте базовую загрузочную дискету с помощью команды FORMAT A: /S. (Помните прежний совет о емкости дискет.)

Далее, знаете ли вы специальные драйверы устройств для обращения к жесткому диску? Например, не могли ли вы работать со специальной программой организации разделов, например Disk Manager (имя драйвера устройства DMDRVR.BIN)? Или с программой уплотнения файлов типа Stacker (имя драйвера устройства STACKER.COM)?

Не беспокойтесь, если вы применяли эти драйверы для обращения к логическому накопителю D или к накопителям с последующими буквами. Если доберетесь к файлам в накопителе C, прочитаете старые файлы CONFIG.SYS и AUTOEXEC.BAT и, возможно, восстановите требуемые драйверы устройств.

Если вам требуются любой такой драйвер устройства или резидентная программа, создайте минимальные файлы CONFIG.SYS и AUTOEXEC.BAT, которые содержат строки для загрузки этих программ и поместите эти файлы и программы на новую дискету. Почти во все ситуациях такая простая загрузочная дискета DOS, дополненная минимальным набором драйверов для доступа к накопителю C, может стать настоящей безопасной загрузочной дискетой.

Для одной из неисправностей, связанной с потерей содержимого КМОП-ЗУПВ, такой дискеты недостаточно. В КМОП-ЗУПВ хранится информация о типах накопителей на гибких и жестких дисках вашего ПК. Вы можете знать емкость накопителей на гибких дисках, но вряд ли помните тип накопителя на жестких дисках. (Тип накопителя — это число, которое хранится в КМОП-ЗУПВ конфигурации и показывает на таблицу описаний накопителей в ROM BIOS на материнской плате. Чаще всего это число находится в диапазоне от 1 до 47, хотя возможны и другие значения. Тип 47 обычно означает определяемый пользователем тип накопителя. Если ваш накопитель этого типа, его описание также хранится в КМОП-ЗУПВ.) Каждый ПК с КМОП-ЗУПВ конфигурации имеет таблицу типов накопителей, и вы должны знать, какой элемент КМОП-ЗУПВ идентифицирует ваш накопитель.

Если вы потеряли информацию из КМОП-ЗУПВ и не знаете типа жесткого диска, обратитесь к такому приему: войдите в программу настройки и по очереди выберите тип накопителя. После каждого нового выбора перезагружайте компьютер и проверяйте возможность успешного доступа к накопителю C. В конце концов, вы наткнетесь на правильное значение.

Ситуация усложняется, если правильный тип вашего накопителя отсутствует в таблице типов накопителей. Для решения этой проблемы вам придется воспользоваться несколькими способами, а если жесткий диск для вас готовил другой человек, вы можете не знать выбранный им способ. В зависимости от используемого способа потеря информации в КМОП-ЗУПВ вызовет небольшую досаду или окажется серьезной трудностью.

Практические способы включают в себя определяемый пользователем тип накопителя, поддерживаемый некоторыми ROM BIOS на материнской плате, автоконфигурирующий контроллер жесткого диска или загружаемый драйвер устройства. Если вы не знаете какой способ применялся раньше, вместо простого гадания лучше воспользоваться специальной утилитой, рассчитанной (кроме других возможностей) именно на этот случай. Такая специальная программа фирмы Micro House называется DrivePro.

Далее с безопасной загрузочной дискетой на руках при возникновении неисправностей в жестком диске обратитесь к главе 11. Если рекомендации из этой главы покажутся вам сложными, прочитайте главы 2—10 и вновь вернитесь к главе 11.

Заключение

Повторим положение, приведенное в начале этой главы: не все неисправности жесткого диска означают его окончательную гибель. Часто их можно довольно легко исправить, но для этого нужно знать, как работают жесткие диски и как они погибают. В этой главе вы получили следующие сведения:

- Жесткий диск является ценным устройством ПК из-за своей фактической стоимости и еще более — из-за содержащихся в нем данных.
- Простая подготовка позволяет смягчить неизбежные проблемы с жестким диском. Без мер предосторожности, вы подвергаете свои данные и всю работу серьезному риску.

- Обязательно создайте безопасную загрузочную дискету для вашего ПК. Это послужит основой комплекта выживания для конкретного ПК.

Перед попыткой использовать комплект выживания для идентификации и устранения неисправностей жесткого диска, внимательно изучите остальной материал книги, начиная с главы 2. Эти знания облегчат диагностику и ремонт жестких дисков.

•

Глава 2. Магнитное хранение информации

В этой главе вы узнаете

- Три основных физических принципа, на которых базируется магнитная запись данных
 - Основные элементы конструкции жесткого диска
 - Смысл терминов "дорожка", "сектор" и "цилиндр"
 - Размер одного хранимого на диске бита и почему этот размер не может быть произвольно малым
 - Почему иногда битам отводится для хранения больше места, чем необходимое
 - Как математики и инженеры "прячут" ваши данные так, чтобы в последующем их можно было точно и надежно найти
 - Различия способов кодирования данных
 - Как введение небольшой дополнительной информации делает диск в сотни раз надежнее
 - Как коды с исправлением ошибок скрывают проблемы до тех пор, пока их нельзя исправить без потери некоторых данных.
-

2.1. Физические принципы

Жесткий диск служит хранилищем данных в те интервалы времени, когда компьютер не обрабатывает их. Главное назначение жесткого диска — хранить информацию и выдавать ее по запросу. Жесткие диски действуют аналогично ленточным магнитофонам: для хранения информации применяется местная намагниченность в тонком слое магнитного материала, находящегося на некотором носителе (называемом подложкой).

Различие между этими двумя способами состоит в том, что в магнитофоне носителем служит длинная пластиковая лента, намотанная на катушки, а в дисковых накопителях применяются металлические диски. Гораздо важнее, однако, что на ленте информация хранится в аналоговой форме, т.е. магнитные сигналы прямо имитируют хранимую звуковую информацию. Жесткие диски являются цифровыми устройствами — в них магнитные сигналы хранят числа, представляющие информацию, подлежащую хранению.

Магнитный носитель в обоих случаях может поддерживать только два уровня намагниченности: только в одном направлении или только в другом направлении. Другими словами, носитель в каждой точке намагничен до насыщения. При записи аналоговой информации добавляется аналоговый сигнал для смещения сигнала генератора и записывается этот составной сигнал. Результирующая намагниченность колеблется между двумя уровнями насыщения, причем коэффициент заполнения отражает хранимый аналоговый сигнал. Для считывания такого сигнала головка считывания усредняет сигналы смещения и восстанавливает исходный аналоговый сигнал.

Хранение цифровых данных осуществляется совсем по-другому. Далее показано, что цифровые данные представлены блоками чисел, которые кодируются и хранятся в виде переходов намагниченности в четко определенных местах на поверхности диска. Головка считывания воспринимает каждое из этих изменений, а схемы декодирования преобразуют изменения в копии исходных чисел.

Последнее отличие состоит в том, что при аналоговой записи на ленту допускаются небольшие дефекты, например помехи или небольшие пропуски информации.

С другой стороны, подсистема жесткого диска в компьютере должна хранить цифровые данные абсолютно точно. Ошибка всего в одном бите из нескольких миллионов битов может существенно повлиять на работу компьютера.

Конечно, реальные жесткие диски несовершенны, поэтому в них бывают ошибки. Для минимизации влияния таких ошибок в накопители встраиваются специальные схемы, которые скрывают дефекты и исправляют ошибки. В результате накопитель выглядит намного надежнее, чем он есть на самом деле.

Для понимания работы жестких дисков и причин их гибели вначале необходимо разобраться в магнитной записи цифровых данных. Вы узнаете также о том, какие меры фирмы-производители предпринимают для того, чтобы их изделия удовлетворяли потребителей. Однако из главы 11 вы узнаете, что эти приемы помогают не всегда и что жесткие диски все-таки погибают.

2.1.1. Три важных физических факта

Ленточные магнитофоны и накопители на жестких дисках базируются на одних и тех же физических принципах. Для общего понимания их работы вам достаточно знать всего три факта. (Этого достаточно для понимания работы устройств с магнитной записью, в которых применяются наиболее распространенные индуктивные головки считывания-записи. Для устройств с головками, воспринимающими магнитный поток (глава 4), потребуются еще один факт.)

1. При пропускании тока через обмотку (или катушку) создается магнитное поле.
2. При воздействии достаточно сильного магнитного поля на ферромагнитное или ферримагнитное вещество поле "зависает", т.е. часть материала намагничивается в том же направлении, и эта остаточная намагниченность существует до подачи другого, достаточно сильного магнитного поля, которое намагничивает вещество в другом направлении. Этот процесс показывает, как записывается магнитная информация.
3. При изменении магнитного поля, проходящего через обмотку (или катушку), на ее зажимах появляется напряжение. Если, например, опускать постоянный магнит через катушку, можно обнаружить магнитное поле по наведенному в катушке напряжению. Этот процесс показывает, как можно считать записанную информацию.

В главе 4 будет показано, что в некоторых дисковых накопителях применяется другой способ считывания, основанный еще на одном факте. Однако в большинстве жестких дисков ПК сейчас применяются индуктивные головки считывания-записи, а их работа основана на трех приведенных физических фактах.

В системах магнитной записи используются два типа магнитных материалов. Материалы первого типа концентрируют и проводят силовые магнитные линии (магнитный поток), но они не сохраняют заметной намагниченности после снятия внешнего магнитного поля. Они далее называются просто магнитными материалами. (Замечание о проводимости магнитного поля означает, что поле "предпочитает" проходить через магнитный, а не через немагнитный материал.) В других книгах такие материалы называются магнитомягкими.

Ко второму типу относятся материалы, которые обладают свойством остаточной намагниченности. Такие материалы не только концентрируют и проводят магнитные поля, но и сохраняют некоторую намагниченность после снятия внешнего поля. Эта группа материалов подразделяется на ферромагниты и ферримагниты. (Различие между ними для нас несущественно, мы просто привели их точные названия; в накопителях на жестких дисках применяются вещества обеих подгрупп.) Чтобы не употреблять длинные термины "ферромагнитный материал" и "ферримагнитный материал", далее будем называть их "намагничивающимся материалом". В других книгах такие материалы называются магнитожесткими, или остаточными (постоянными) магнитами.

Зная о трех физических явлениях, на которых базируется магнитная запись информации, можно рассмотреть, как они используются для создания жестких дисков.

2.1.2. Базовая конструкция жесткого диска

Основные компоненты накопителя на жестких дисках представлены на рис. 2.1. На шпинделе, который вращается с постоянной высокой скоростью, смонтированы один или несколько металлических дисков (platters — "блинов" или даже "пластин"), с обеих сторон покрытых очень тонким слоем намагничивающегося материала. Для каждой поверхности (сверху и снизу каждого диска) имеется головка считывания /записи. Головки оформлены в блок головок, который может перемещаться к шпинделю или к краю дисков.

Диски и головки. Совсем недавно некоторые фирмы стали применять в накопителях на жестких дисках стеклянные диски, которые обладают большей жесткостью и могут быть более плоскими. Повышенная жесткость особенно важна в некоторых новых накопителях, скорость вращения которых больше обычных 3600 обор./мин. Пока накопителей со стеклянными дисками мало, но в недалеком будущем они получат широкое распространение.

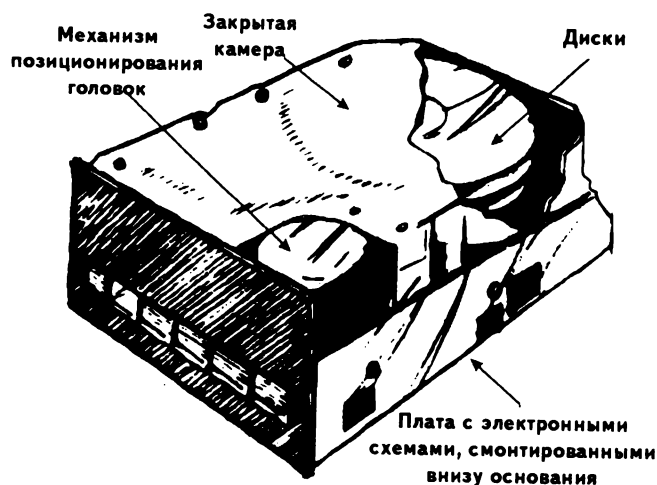
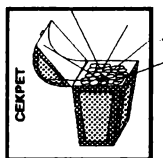


Рис. 2.1. Разрез типичного накопителя, показывающий диски

Когда диск не вращается (накопитель выключен), слабая пружина прижимает головку к поверхности диска. При вращении диска головки поднимаются ветром, который создает вращающийся диск. Слой воздуха под головкой оказывается достаточно сильным, чтобы поднять головку с поверхности, хотя и на очень малую высоту.



В первых накопителях высота полета головок над поверхностью составляла несколько микронов. В современных накопителях это расстояние уменьшено до 0,1–0,5 микронов, что в тысячу раз меньше толщины человеческого волоса.

Поток воздуха одновременно поднимает головку над поверхностью и удерживает ее на малом расстоянии. Следовательно, поток воздуха управляет полетом головки, заставляя ее следить за поверхностью диска.

Головка должна летать над поверхностью, а не касаться ее; это предотвращает износ головки из-за трения и, что важнее, предотвращает износ покрытия головкой. Но головка не должна удаляться от поверхности диска, так как иначе она не сможет намагнитить носитель или считать имеющуюся намагниченность.

В главе 4 будут рассмотрены различные варианты конструкции головок и влияние высоты полета на работу головки. вы узнаете также, как новый способ перпендикулярной записи существенно изменил прежние положения.

Дорожки и цилиндры. При вращении диска и нахождении блока головок в некоторой позиции каждая головка образует на поверхности диска окружность, или кольцо. Такие кольца называются дорожками. Информация записывается на дорожки пакетами, которые называются секторами.

Данные в одном секторе считываются и записываются как единое целое. На рис. 2.2 показаны один диск и одна головка, а также дорожки и секторы. (На этом рисунке размер головки относительно диска сильно увеличен, поэтому ширина дорожек также увеличена. Вместо показанных на рисунке семи дорожек типичный накопитель на жестком диске имеет от нескольких сотен до нескольких тысяч дорожек на каждой поверхности диска.) Дорожки невидимы, поскольку это просто участки поверхности, которые намагничиваются специальным образом.

Если бы можно было рассмотреть дорожки, они оказались бы похожими на бороздку грампластинки. Сходство есть, но есть и различия. Важнейшее отличие состоит в том, что дорожки на диске круговые и отдельные. Пластинка же обычно имеет только одну бороздку, которая спирально тянется от края к центру. При проигрывании пластинки игла непрерывно движется от края к центру. При считывании или записи данных на жестком диске головки неподвижны. Они перемещаются только при необходимости перейти на другую дорожку, а затем останавливаются.

Необычная бороздка

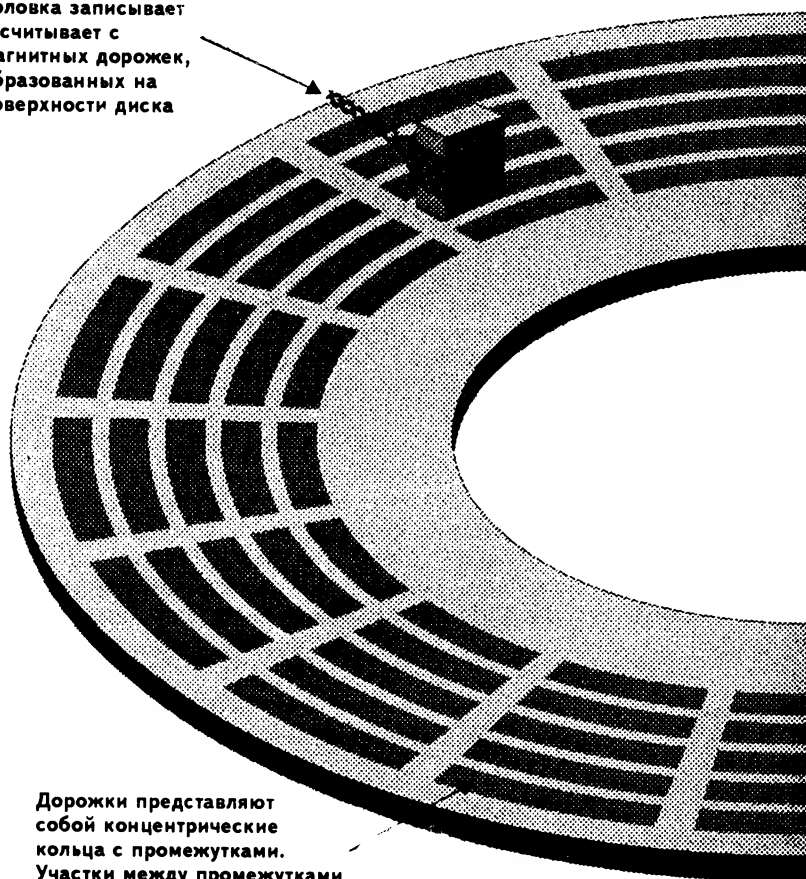
Вы можете заметить, что на пластинке может быть несколько бороздок. Действительно, в некоторых новых пластинках есть несколько спиральных бороздок, например четыре, которые начинаются от края пластинки. Для проигрывания нужно поместить иглу на одну из бороздок. В этом случае можно проиграть одну из четырех записей.

Блок головок перемещает так называемый привод. В накопителях применяются два типа приводов: привод с шаговым двигателем, который управляет ведущей шестерней и рейкой, и привод со звуковой катушкой. Первый позиционер базируется на счислении пути, а во втором для правильного позиционирования головок применяется отрицательная обратная связь. Подробнее об устройстве приводов сказано в главе 4.

Привод блока головок перемещает блок в одном из двух направлений небольшими шагами. В любой конкретной позиции блока головок каждая головка образует на поверхности вращающегося диска дорожку. При перемещении головок в новую позицию они выбирают новый набор дорожек. (Головки формируют дорожки, когда осуществляется форматирование диска низкого уровня. Поэтому в дальнейшем при позиционировании головок они отслеживают ранее записанные дорожки.)

Дорожки представляют собой концентрические окружности, которые нумеруются с нуля от внешнего края диска, как показано на рис. 2.3. (Показан разрез накопителя с четырьмя дисками, вертикальный размер которого для наглядности сильно увеличен. Кроме того, показанная ширина дорожек также намного больше реальной ширины.) Дорожки с одним и тем же номером, т.е. все дорожки на всех поверхностях для некоторой позиции блока головок, образуют так называемый цилиндр. (Максимальное число дисков в накопителе равно 14; в типичных накопителях число дисков варьируется от 2 до 8.)

Головка записывает
и считывает с
магнитных дорожек,
образованных на
поверхности диска



Дорожки представляют
собой концентрические
кольца с промежутками.
Участки между промежутками
называются секторами

Рис. 2.2. Когда диск вращается под головками считывания-записи, головки проходят по дорожкам

Важность введения понятия цилиндра объясняется очень просто: дорожки цилиндра проходят под соответствующими головками одновременно. Это означает, что компьютер может считывать или записывать данные на любой дорожке текущего цилиндра, просто выбирая электронным коммутатором нужную головку. Такие коммутаторы действуют очень быстро, намного быстрее физического перемещения головок на соседний цилиндр.

Число дорожек в цилиндре равно числу головок или удвоенному числу дисков (в некоторых накопителях одна или две головки резервируются для механизма позиционирования. Если в накопителе для сервосистемы выделена одна головка, указываемое в таблицах число головок оказывается нечетным) Головки нумеруются с нуля, начиная с верхней поверхности верхнего диска (В вычислительной технике счет обычно ведется с нуля.)

Число цилиндров (или дорожек на диске) в накопителе зависит от ширины дорожки (а она зависит от размера головки) и размера шага привода блока головок. Напомним, что конструкция на рис. 2.3 сильно увеличена; в типичных накопителях число цилиндров составляет от нескольких сотен до нескольких тысяч.

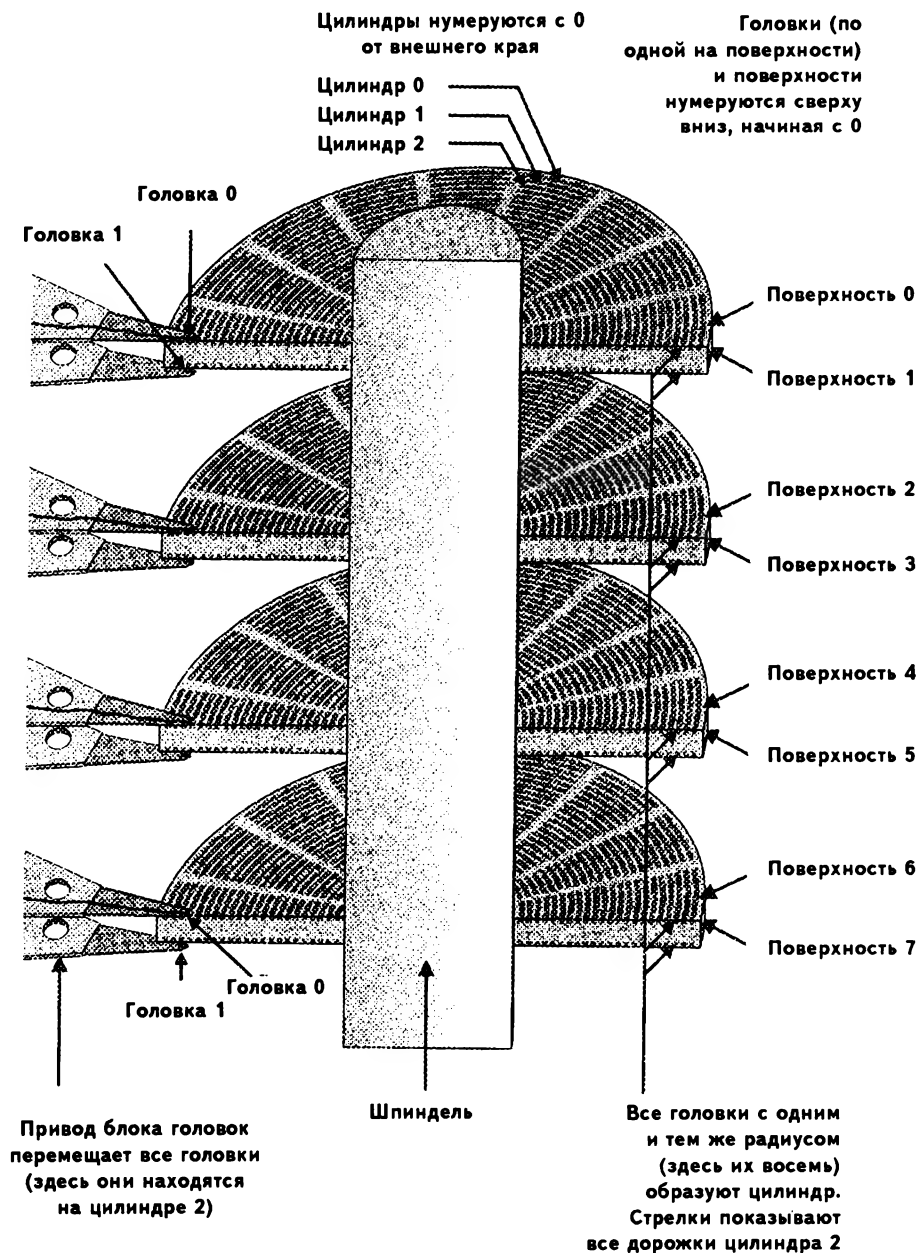
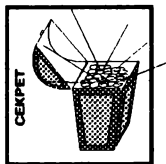


Рис. 2.3. Расположение и нумерация головок (поверхностей) и цилиндров

2.1.3. Важность чистоты

Даже крошечные пылинки могут вызвать катастрофу в накопителе, поэтому подвижные части заключаются в плотно закрытую камеру, заполненную очень чистым воздухом. Камера открывается только специалистами в так называемой чистой комнате. Если открыть камеру в обычных условиях, гарантия фирмы-производителя снимается и накопитель почти наверняка выйдет из строя.



Камера с подвижными частями накопителя не герметична. В камеру встроен вентилятор, который подает воздух через очень тонкий фильтр с отверстиями диаметром в доли микрона. Воздух выходит через имеющиеся в камере щели. Внутреннее давление воздуха выше внешнего, поэтому пыль выходит через неизбежно имеющиеся щели.

Если открыть камеру накопителя, то вентилятор обнаружить не легко. Вентилятор фактически встроен в конструкцию шпинделя и дисков. При вращении диски увлекают за собой воздух и отбрасывают его к внешним краям. Такой центробежный вентилятор засасывает воздух через фильтр, находящийся сверху шпинделя.

2.1.4. Запись и считывание информации

В каждой головке на магнитном сердечнике есть обмотка. Одна из возможных конструкций головки показана на рис. 2.4,а. Отметим, что в сердечнике есть небольшой зазор, в котором происходят все действия, касающиеся магнитного покрытия диска. Когда головка считывания-записи находится над диском, она записывает на дорожке участки намагниченности (см. рис. 2.4,а). Магнитные силовые линии от тока тока в обмотке головки образуют на покрытии поверхности диска постоянный магнит (см. рис. 2.4,б). Силовые линии постоянных магнитов на покрытии замыкаются внутри. У магнита, находящегося под зазором головки, силовые линии замыкаются через сердечник и обмотку головки (см. рис. 2.4,в).

Важно отметить, что все магнитные поля представляют собой замкнутые петли. Независимо от того, создаются они током в обмотке сердечника или постоянным магнитом, силовые линии замыкаются. В постоянном магните петля проходит через магнитный материал, выходит из его "северного полюса", замыкается снаружи и входит в "южный полюс". В случае обмотки сердечника петли проходят через обмотку, замыкаются снаружи и возвращаются с другой стороны обмотки.

Силовые магнитные линии стремятся проходить через магнитопроводящий материал. Поэтому в электромагнитах сердечник часто делается из магнитного материала. Сердечник направляет и концентрирует магнитное поле.

Когда головка считывания-записи в накопителе записывает информацию на диск, она действует как крошечный электромагнит с магнитным сердечником. Но этот сердечник оказывается не единственным маршрутом для магнитных силовых линий. Сердечник имеет небольшой зазор. Сами силовые линии не могут иметь никаких разрывов, поэтому они выходят из сердечника и пересекают зазор.

Если поблизости есть магнитный материал, силовые линии проходят через него, а не просто пересекают зазор. В жестком диске таким соседним материалом оказывается поверхностное покрытие диска, как показано на рис. 2.4,а. При таком отклонении силовые линии от обмотки намагничивают материал поверхности. При изменении тока в обмотке направление магнитных силовых линий изменяется; такое изменение регистрируется как противоположное магнитное поле на поверхности диска.

Когда диск вращается под головкой, ток в обмотке сердечника быстро изменяет направление, заставляя ток записывать изменяющееся магнитное поле на поверхности диска. Такие поля действуют на небольшом участке дорожки. Таким образом, изменения тока позволяют сохранить информацию, как показано на рис. 2.4,в.

Когда обмотка не проводит никакого тока (и не создает никакого магнитного поля), она может считывать магнитные поля с поверхности диска. Если участок поверхности диска рядом с зазором сердечника намагничен, магнитные силовые линии этого участка замыкаются через сердечник (и через обмотку), а не просто через зазор (см. рис. 2.4,в).

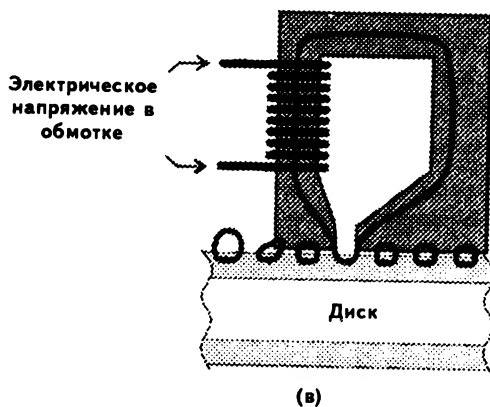
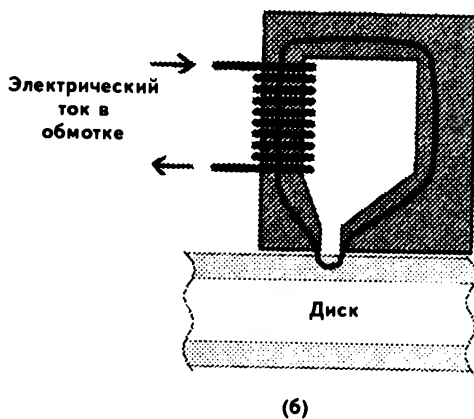
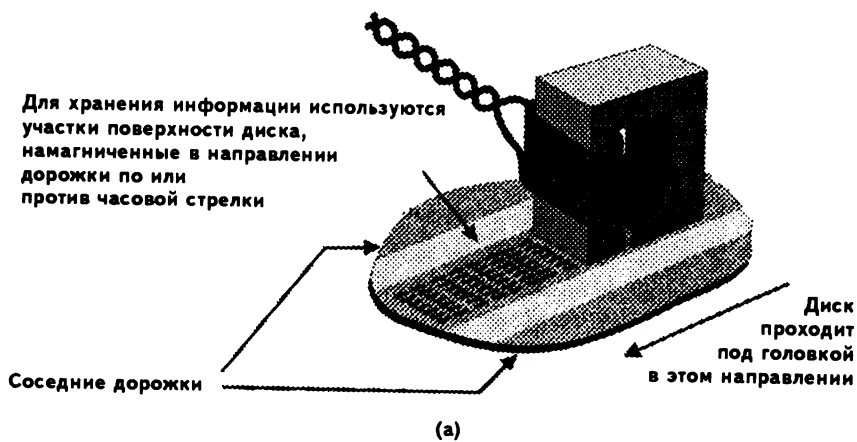


Рис. 2.4. Типичная конструкция головки считывания-записи

После записи данных участок диска, проходящий под головкой, содержит, по существу, цепочку постоянных магнитов, ориентированных в разных направлениях.

Когда головка находится над таким магнитом, она замыкает магнитные силовые линии через обмотку. Следовательно, когда головка проходит поверх магнита, указывающего в одном направлении, к соседнему магниту, показывающему в другом направлении, проходящее через обмотку магнитное поле изменяется. В соответствии с отмеченным ранее третьим физическим фактом, именно в течение этого короткого временного интервала на выводах обмотки появляется импульс напряжения. Такие выбросы напряжения позволяют жесткому диску воспринять записанную информацию.

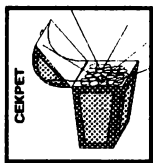
Кроме дисков, шпинделя, шпиндельного двигателя, головки считывания-записи, блока головок и привода и закрытой камеры, содержащей и защищающей эти части, все накопители должны иметь еще одну часть — плату с электронными схемами. С одной стороны эти схемы подключаются к приводу головок, головкам и шпиндельному двигателю. С другой стороны схемная плата подключается к компьютеру или к плате контроллера диска, а также к блоку питания компьютера. Назначение схем на этой плате рассмотрено в главе 4.

2.1.5. Размер бита

Для пользователей ПК жесткий диск должен иметь максимальную емкость. Информация хранится единицами, которые называются битами. Каждый бит может быть нулем или единицей. Байт состоит из восьми бит. Например, в программе обработки текста байт представляет в документе один символ или форматирующий код. Емкость накопителя в битах зависит от числа имеющихся в нем поверхностей (сколько дисков и каковы их размеры) и от размера участка поверхности для хранения одного бита.

Биты записываются в виде элементов намагниченности магнитных полей вдоль дорожек, причем каждый элемент ориентирован вперед или назад. При вращении диска под головкой ток в головке изменяется, что вызывает изменения записываемых магнитных полей. Можно для простоты считать, что при записи на дорожке укладываются постоянные магниты. На рис. 2.5 схематически показано, как выстраиваются магнитные частицы вещества на поверхности диска и как головка считывания-записи оставляет на дорожке намагниченные участки. (Конечно, в реальном накопителе число участков намного больше чем показано на рисунке.)

Размер (или площадь) бита есть произведение его ширины на длину. Ширина бита равна ширине дорожки или, упрощенно, ширине головки считывания-записи. Чтобы запись на одну дорожку не "перезаписывала" части соседних дорожек и чтобы головка не воспринимала информацию более чем с одной дорожки, приходится оставлять между дорожками небольшой промежуток. Такая буферная зона увеличивает площадь, занимаемую каждым битом, поэтому эффективная ширина бита равна расстоянию между дорожками, а не просто (несколько меньшей) ширине дорожки.



Фактически дорожки несколько шире головок. Различие вызывается тем, что некоторые магнитные силовые линии, выходящие из зазора и воздействующие на участки поверхности диска, оказываются просто вблизи головки, а не точно под ней. Такое "растекание" влияет на точность считывания, если разместить головки слишком близко; чтобы избежать этого, задается минимальное расстояние между дорожками.

В видеомэгнитофонах для подавления перекрестных помех взаимодействия соседних дорожек применяют две или более головок и выравнивают зазоры головок так, чтобы они не были параллельными (так называемое азимутальное выравнивание). Благодаря этому дорожки можно размещать почти вплотную. Такой способ в накопителях на жестких дисках не применяется, так как перемещение головок с дорожки на дорожку заставляет использовать одну и ту же головку для всех дорожек. В видеомэгнитофонах, имеющих вращающуюся головку и спиральную по-

[illegible]

Описать длину бита несколько труднее. На нее влияют три фактора. Во-первых, имеется некоторый предел того, как близко можно размещать переходы магнитного поля вдоль дорожки. Во-вторых, для хранения бита информации может потребоваться больше чем длина одного перехода. Умножение минимальной длины магнитного "триггера" на требуемое число таких длин дает абсолютно минимальную длину бита для каждого конкретного накопителя. В-третьих, у разработчиков накопителей могут быть веские причины для того, чтобы применять для хранения бита участки, большие абсолютно минимального размера.

Минимальная длина магнитного триггера. Насколько малой может быть длина магнитного триггера? Ответ на этот вопрос можно разделить на три части.

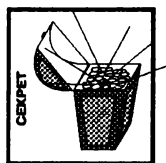
Прежде всего приходится учитывать характер материала поверхности. В накопителях с оксидным покрытием поверхность состоит из небольших частиц магнитного материала, находящихся в связующем веществе. (Связующим веществом является специальная краска. Обычная краска состоит из частиц окрашенного вещества, находящихся в материале, который приклеивается к окрашиваемому предмету. Краска для накопителей просто заменяет магнитный наполнитель окрашенным наполнителем.) В накопителях с металлическим покрытием поверхность представляет собой осажденную пленку чистого магнитного материала. Для любой поверхности существует минимальный участок, который можно намагнитить; этот участок называется зерном или доменом. Длина триггера не может быть меньше размера домена.

Второе ограничение на минимальное расстояние между магнитными переходами связано с последовательностью ориентации записанных участков. Отметим по рис. 2.6,б, что у соседних участков с противоположной намагниченностью "северный полюс" находится рядом с другим "северным полюсом", а "южный полюс" — рядом с другим "южным полюсом". Биты на диске отталкивают друг друга, аналогично отталкиванию магнитов при их приближении; чем ближе расположены участки, тем сильнее магнитное поле одного участка (или бита) отталкивает соседние участки. Если участки оказываются слишком близко, поле одного участка может изменить поле соседнего участка, т.е. два участка сольются. Следовательно, минимальная длина триггера должна предотвращать такое событие.

На рис. 2.6,а показан объемный вид головки и дорожки; рис. 2.6,б — увеличенный вид сбоку, а на рис. 2.6,в показан график намагниченности поверхности (знак "+" означает вдоль дорожки, а знак "-" противоположное направление). На этих рисунках упрощенный способ кодирования данных. На практике применяются более сложные способы, которые рассматриваются в следующем разделе.

Третьим ограничивающим фактором является размер зазора сердечника головки. Обычно длина триггера делается, как минимум, равной ширине зазора. Несмотря на потенциальную возможность записи меньших участков, это не рекомендуется, так как через обмотку могут одновременно проходить поля от нескольких участков, что затрудняет выделение магнитного поля каждого участка. Кроме того, при записи на диск ток в обмотке намагничивает в одном направлении дорожку по всей длине зазора и может "перезаписать" соседние биты.

Современная технология производства накопителей позволяет свести все эти пределы примерно к одной величине. Поэтому размер домена поверхности, минимальный размер участков, которые не намагничивают друг друга, и ширина зазора головки оказываются сравнимыми друг с другом.



Минимальный размер участка покрытия диска, который намагничивается независимо, является одним из двух важнейших физических параметров носителя. Вторым критическим параметром называется порог силы магнитного поля, необходимого для намагничивания покрытия поверхности (коэрцитивная сила). В течение многих лет ведутся исследования по управлению этими параметрами в процессе производства. Минимальный размер независимо намагничиваемого участка делается все меньше, что позволяет упаковать на единице длины дорожки все больше битов. Коэрцитивная сила также повышается, что позволяет образовать на поверхности диска все больше устойчивых магнитных участков.

Есть несколько способов образования магнитного покрытия поверхности. В первых накопителях крошечные игольчатые частицы оксида железа смешивались с эпоксидным связующим веществом и наносились на поверхность диска. Часто при отверждении эпоксидной смолы диски вращались в поле сильного магнита и частицы выравнивались вдоль дорожек. Такой процесс обеспечивает намагниченность в плоскости поверхности диска в прямом или обратном направлении вдоль дорожки (см. рис. 2.5).

Дальнейшие исследования привели к покрытию из почти чистого магнитного материала, который напылялся или осаждался на поверхность диска. Материал ориен-

тируется так, что намагниченность оказывается такой же, как для поверхности с оксидным покрытием. Достоинства металлических носителей состоят в том, что их можно сделать тоньше с получением сигналов той же силы. Уменьшение толщины позволяет сократить размеры участков для хранения информации.

Оксидные поверхности обеспечивают плотность записи около 20 тыс. битов на дюйм дорожки, а плотность металлических носителей в четыре раза больше. Кроме того, для металлического носителя можно применять более узкие головки, а это увеличивает число дорожек; в результате емкость диска с металлическим покрытием примерно в десять раз больше емкости диска с оксидным покрытием.

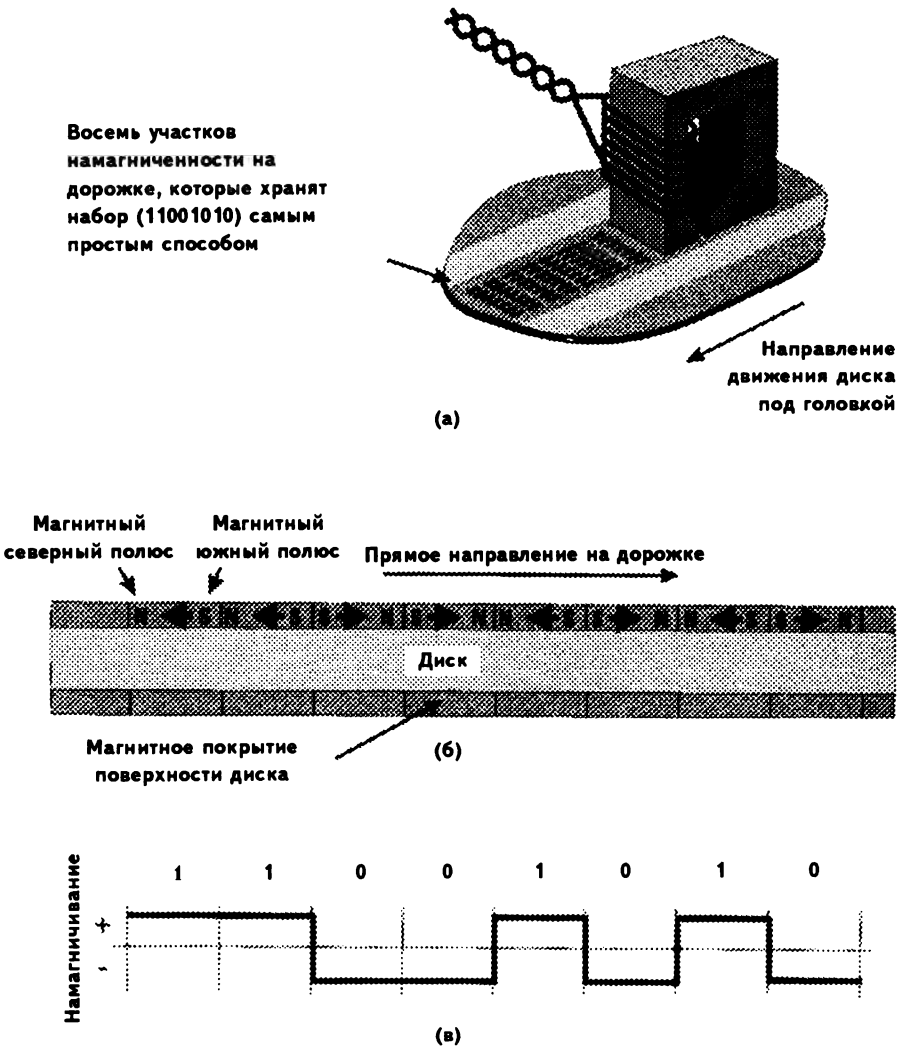


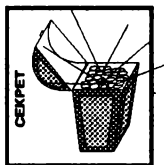
Рис. 2.6. Для хранения информации на поверхности диска применяются участки с различной намагниченностью

Два этих фактора ограничивают размер независимо намагничиваемого участка на поверхности диска (длину магнитного триггера). Способ кодирования данных опре-

деляет, сколько таких длин необходимо для хранения одного бита. Этот вопрос рассматривается в следующем разделе; пока нам достаточно знать, что для каждого бита требуется некоторое фиксированное число (необязательно целое) таких участков.

Теоретическая и практическая плотность битов. Длину дорожки, используемую для хранения одного бита, можно назвать битовой ячейкой. Реальные битовые ячейки в дисковых накопителях не всегда столь малы, как они могли бы быть (т.е. минимальная длина магнитного триггера, умноженная на число длин триггеров, необходимое для кодирования бита). Такая ситуация объясняется экономическими, а не физическими причинами. Дорожки около шпинделя короче внешних дорожек, т.е. на внешних дорожках для битов больше места. Это — физика. Поскольку шпиндель вращается с постоянной скоростью независимо от позиции блока головок (это не относится к некоторым накопителям на гибких дисках, которые, правда, в ПК не применяются), время прохода всей дорожки под головкой одинаково для всех дорожек. Более длинные внешние дорожки просто движутся под головкой быстрее.

Другими словами, головка считывает данные без учета скорости движения диска. Головка воспринимает только частоту переходов магнитного поля. Битовые ячейки на краю длиннее, чем вблизи центра. Проще всего было бы считать длины всех дорожек одинаковыми. Каждая дорожка проходит под головкой одно и то же время. Электронные схемы считывания и записи не учитывают, какая дорожка проходит под головкой.



В приведенном рассуждении не затронут один важный момент. Если на каждую дорожку поместить одно и то же число битов, биты на внутренних дорожках окажутся ближе друг к другу. Для достижения максимальной емкости накопителя биты на внутренних дорожках "упаковываются" как можно плотнее. Иногда при очень плотном размещении бит возникает интересное явление. Магнитные переходы сами по себе сближаются. Это явление называется битовым сдвигом или группированием импульсов. (Под магнитным переходом автор понимает границу между участком, намагниченным в одном направлении, и участком, намагниченным в другом направлении.) В зависимости от способа кодирования данных и конкретных записываемых данных можно заметить, что биты сдвинуты больше обычного.

Это явление приводит к тому, что сигналы от головки считывания возникают в неправильные моменты времени, и контроллер может не распознать информацию, записанную в этом месте диска. Приходится предвидеть битовый сдвиг и корректировать его еще до фактического появления.

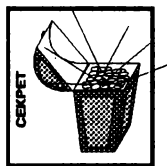
Подаваемые в головку записи сигналы вызывают изменения тока немного раньше или немного позже номинальных моментов. Конкретный сдвиг каждого перехода зависит от многих факторов, но для каждого данного накопителя (с известным способом кодирования данных и фиксированными физическими свойствами) правила определены и хорошо изучены. Если позиционирование осуществляется точно, считанные сигналы возникнут точно там, где они и должны быть. Разработанный способ называется предкомпенсацией записи.

Предкомпенсация записи требуется только для цилиндров, близких к шпинделю. Биты на внешних цилиндрах размещены достаточно далеко, и битовый сдвиг не возникает. Наилучшая предкомпенсация должна быть некоторой функцией от номера цилиндра, но гораздо экономичнее ввести фиксированную величину для всех цилиндров, которые ближе к шпинделю, чем некоторое расстояние, а для остальных предкомпенсацию записи не применять.

Старые контроллеры дисков предкомпенсировали записи для диапазона цилиндров по специальному запросу. Новые контроллеры осуществляют предкомпенсацию записи автоматически без участия пользователя.

При построении гораздо более сложного накопителя или контроллера можно разместить на внешних дорожках больше битов, но это оказывается слишком доро-

гим. Именно здесь приходится учитывать экономические факторы. В дешевых накопителях небольшой емкости для персональных компьютеров число бит на каждой дорожке одинаково и не зависит от длины дорожки. При проектировании более дорогих и емких накопителей можно отойти от этого правила.



До недавнего времени почти во всех накопителях на каждой дорожке хранился фиксированный объем информации (возможно, различный в разных накопителях), поэтому стандартные контроллеры дисков не допускали никакой другой возможности. Следовательно, для размещения на внешних дорожках больше битов, чем на внутренних, требуется нестандартный контроллер.

В накопителях с интерфейсами SCSI (Small Computer System Interface — интерфейс малых компьютеров) и IDE/ATA (Integrated Drive Electronics/AT Attachment — Интегрированная электроника накопителей/Подключение AT) применяются встроенные контроллеры (подробнее рассмотрены в главе 3.)

Некоторые фирмы выпустили накопители с интерфейсами SCSI и IDE/AT, в которых учтено, что внешние дорожки длиннее внутренних, и на внешних дорожках размещено больше данных. При этом дорожки разделены на зоны. Например, дорожки на внешней трети диска образуют зону #1, дорожки средней трети объединены в зону #2, а дорожки внутренней трети представляют зону #3. Объем данных на каждой трети внутри зоны постоянный, но в зонах различный. Такой способ иногда называется зонной записью. Он применяется в некоторых комбинациях накопитель/контроллер с MFM- или RLL-кодированием, а также в жестких дисках со специальной платой.

Конечно, размер бита на поверхности диска зависит от конкретного накопителя, но обычно он находится в следующих диапазонах: длина по дорожке от 0,5 до 2,5 микрон, а ширина дорожки от 12 до 75 микрон. При этом минимальная площадь бита варьируется от 6 до 200 квадратный микрон. Взяв обратную величину от приведенных чисел, можно найти поверхностную плотность; она составляет от 5,000 до 170,000 бит на квадратный миллиметр. Чтобы сделать приведенные величины более наглядными, скажем, что площадь бита на поверхности диска примерно равна площади поперечного сечения человеческого волоса, а в максимальной плотности в современных жестких дисках примерно в 30 раз выше.

2.2. Способы кодирования данных

Сложная структура жесткого диска предназначена для хранения информации, т.е. блоков двоичных чисел, под которыми понимаются просто наборы единиц и нулей. Для считывания и записи данных эти единицы и нули, т.е. биты, должны быть закодированы как магнитные сигналы на поверхности диска. Представьте себе, что отдельные биты можно хранить один за другим вдоль дорожки, назначая каждому биту одно из двух направлений магнитных полей в зависимости от того, является бит единицей или нулем. Если такое представление возможно, для хранения каждого бита потребуются длина одного магнитного триггера.

На рис. 2.6,а и 2.6,б такой способ кодирования показан для конкретного набора данных, на рис. 2.6,в показан график силы магнитного поля в зависимости от расстояния по дорожке для того же набора данных. Если внимательно разобраться в смысле третьего физического факта, приведенного в начале этой главы, оказывается, что такой способ кодирования не работает. Чтобы на выводах обмотки появилось напряжение, магнитное поле в обмотке должно изменяться, а этого данный способ не обеспечивает при записи идентичных данных, например всех единиц или нулей. В такой ситуации сигналы при считывании не возникают.

Кроме того, если скорость диска при считывании даже незначительно отличается от скорости при записи, головка формирует биты с другой скоростью. Если было бы можно "видеть" каждый бит, такое различие в скорости можно учесть, но если

"плыть" над идентичными магнитными полями и совсем не получать сигналов, то невозможно узнать, сколько бит прошло под головкой. Чтобы накопитель мог надежно считывать информацию, требуется более сложный способ кодирования данных.

Считайте кодирование игрой, в которой нужно найти способ "спрятать" биты информации в магнитных триггерах так, чтобы накопитель мог надежно "найти" их при считывании данных. Очевидное решение состоит в том, чтобы сделать каждый бит автономным. По-видимому, проще всего для этого ввести между битами участки без намагниченности. К сожалению, осуществить такой прием не так просто. Зерна, образующие поверхность диска, сильно намагничиваются в том или ином направлении и можно управлять только тем, в каком направлении они намагничиваются.

2.2.1. Простой импульсный подход

Простое работоспособное решение состоит в том, чтобы превратить двоичную единицу в двунаправленный импульс, а каждый двоичный ноль — в отсутствие импульса (обратите внимание на "Предупреждение"). Кроме того, между каждой парой битов данных нужно ввести фиктивный импульс или сигнал синхронизации, аналогичный импульсам двоичных единиц.



Если посмотреть другие книги по рассматриваемому вопросу, можно немного запутаться. Под импульсами понимаются сигналы, которые возрастают и спадают. Однако некоторые авторы говорят о записи импульсов, подразумевая фактически запись перехода магнитного поля от минуса к плюсу или от плюса к минусу (но без возвращения).

Такие записанные переходы приведут при считывании к появлению импульсов напряжения в головке считывания. Во время перехода от отрицательной намагниченности к положительной, напряжение быстро растет к некоторому положительному значению, а затем быстро спадает к нулю, т.е. формируется положительный импульс. При прохождении перехода противоположного направления напряжение в головке быстро спадает до некоторого отрицательного значения, а затем возвращается к нулю, образуя отрицательный импульс.

Некоторые авторы вносят дополнительную путаницу, обозначая как биты данных 1 и 0 переход и его отсутствие, не делая четкого различия между ними. В данной книге запутаться невозможно.

На рис. 2.7 показано несколько диаграмм силы магнитного поля в зависимости от позиции на дорожке для того же набора данных, что и на рис. 2.6. Рис. 2.7 иллюстрирует несколько способов кодирования применительно к байту данных 11001010.

Каждая диаграмма на рис. 2.7 имеет 8 битовых ячеек. Границы ячеек показаны светлыми линиями, а хранимые в ячейках биты (0 или 1) показаны слева. Горизонтальное направление показывает расстояние по дорожке. Серые линии соединяют границы битовых ячеек на каждой диаграмме. Нетрудно заметить, что новые способы кодирования данных значительно сокращают размер битовых ячеек. Как это достигается, Вы вскоре узнаете.

На рис. 2.7,а показан предложенный ранее способ с применением чередующихся импульсов синхронизации и данных. Импульсы от двоичных единиц четко фиксируются при их прохождении под головкой, а импульсы от двоичных нулей, которые не изменяют магнитное поле, нет. Импульсы синхронизации возникают всегда и позволяют головке обнаружить даже длинные цепочки двоичных нулей. Такой способ обеспечивает простоту операций записи и считывания данных, но недостаточно эффективен.

На рис. 2.7 представлены следующие способы кодирования:

- а) Двунаправленная частотная модуляция

- б) Частотная модуляция без возврата к нулю (NRZ — Non Return to Zero), обычно называемая частотной модуляцией (FM — Frequency Modulation)
- в) Модифицированная частотная модуляция (MFM — Modified Frequency Modulation)
- г) Идеализированное напряжение в головке считывания при прохождении под ней набора из в)
- д) Кодирование с ограниченной длиной промежутка (2,7 RLL — Run-Length Limited)

2.2.2. Частотная модуляция

До сих пор мы касались способов кодирования применительно к расстоянию вдоль дорожки от одного участка намагниченности к следующему. Но можно посмотреть на эту же информацию в терминах временных интервалов между прохождением этих участков под головкой считывания-записи или (после деления единицы на значения этих интервалов) частот, в которыми участки проходят под головкой.

Рис. 2.7,а можно рассматривать как прямоугольный сигнал с переменной частотой. При хранении нуля имеется один цикл на битовую ячейку, а при хранении единицы — два цикла. Такой способ аналогичен передаче информации в радиовещательной станции, хотя там для кодирования тонкостей музыки и речи требуется целый диапазон частот, а в дисковых накопителях для кодирования единиц и нулей требуются всего две частоты. Такой способ кодирования называется частотной модуляцией или FM-модуляцией. Несмотря на работоспособность этого способа, разработчики дисковых систем вскоре поняли, что можно применить значительно более эффективный способ кодирования.

2.2.3. Кодирование без возврата к нулю

В показанном на рис. 2.7,а способе для каждого хранимого элемента имеются точно два изменения намагниченности носителя: когда поле изменяется от отрицательного к положительному и когда оно возвращается. Когда под головкой проходит первый переход, в ее обмотке возникает положительный импульс напряжения, а при прохождении обратного перехода появляется отрицательный импульс напряжения. В этом способе кодирования данных электронная схема для считывания данных использует только положительные импульсы, а отрицательные импульсы игнорируются.

Данный способ имеет серьезный недостаток. Напомним, что поле на дорожке нельзя изменять чаще, чем позволяет минимальная длина магнитного триггера. На рис. 2.7 короткая двусторонняя стрелка внизу под каждой диаграммой намагниченности показывает минимальное расстояние между переходами для данного способа кодирования.

В лучшем случае байты данных можно сжимать до тех пор, пока длина, соответствующая двусторонней стрелке, не будет равна минимальной длине магнитного триггера. (На рис. 2.7 так и сделано — все двусторонние стрелки имеют одну и ту же длину. Следовательно, уменьшение размера битовых ячеек на рисунке соответствует фактическому повышению плотности битов, достигаемой в каждом способе кодирования.)

Из рис. 2.7,а видно, что минимальное расстояние равно четверти размера битовой ячейки; следовательно, для варианта FM-кодирования с двуправленным импульсом для хранения одного бита данных вместе с импульсом синхронизации требуются четыре минимальных длины магнитного триггера.

Простой способ сокращения вдвое длины дорожки, необходимой для хранения некоторого объема данных, состоит в том, чтобы при считывании данных игнорировать направление каждого перехода (направление определяется знаком импульса напряжения в обмотке головки, когда под ней проходит переход) и учитывать каждый

переход как значащий. Конечно, для этого необходимо изменить процесс записи данных.

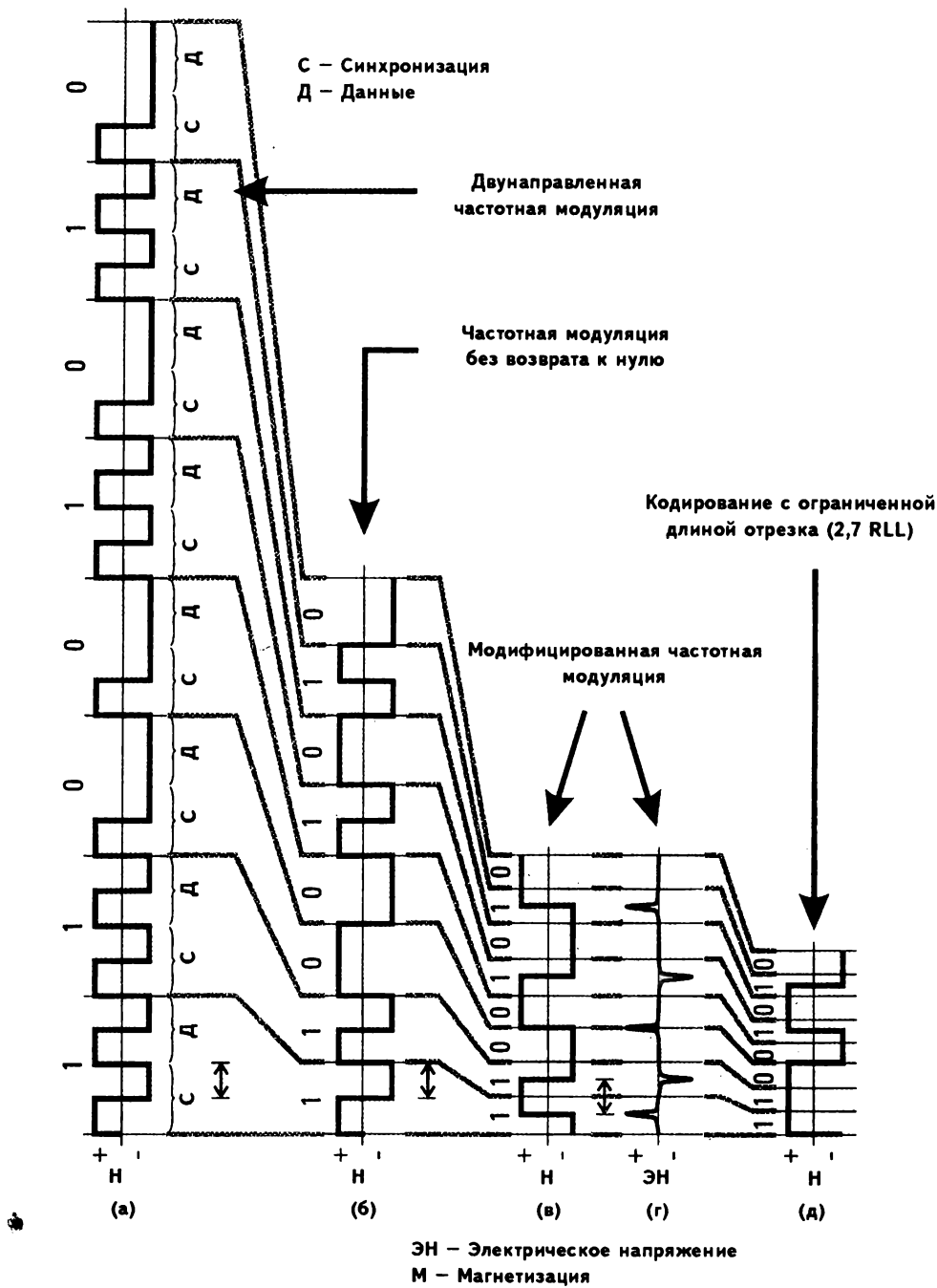
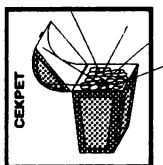


Рис. 2.7. Способы кодирования данных

Такая модификация представлена на рис. 2.7,б. При сравнении этой диаграммы с диаграммой на рис. 2.7,а видно, что при переходе сигнала на рис. 2.7,а от минуса к плюсу сигнал на рис. 2.7,б изменяется вверх или вниз, а при возвращении сигнала на рис. 2.7,а от плюса к минусу никаких изменений на рис. 2.7,б не происходит. Получается, что число переходов и число импульсов при считывании сокращается вдвое, но на дорожке хранится та же информация. Такой способ кодирования данных называется "без возврата к нулю" или NRZ. В этом способе можно преобразовать любой набор импульсов в сокращенное вдвое число импульсов без всякой потери информации.

Поскольку при NRZ-кодировании число переходов уменьшается вдвое, для хранения одного и того же набора потребуется участок дорожки вдвое короче, что повышает эффективность хранения данных и оправдывает некоторые дополнительные усилия. Простой вариант FM-кодирования нигде не применяется, поэтому далее термин "FM-кодирование" относится именно к способу NRZ.



В свое время было предложено много способов кодирования данных, например NRZ, NRZI, PE, ZM и др. В большинстве книг под FM-кодированием понимается рассмотренный выше вариант NRZ; в несколько отличающихся от него способах NRZ и NRZI также реализован принцип "без возврата к нулю".

2.2.4. Модифицированная частотная модуляция

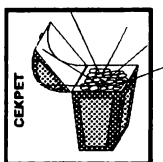
В первых ленточных и дисковых накопителях применялось FM-кодирование; через некоторое время инженеры и математики нашли способ удвоить эффективность системы. Сравнительно простой вариант FM-кодирования ранее назывался кодированием с двойной плотностью, а сейчас называется модифицированной частотной модуляцией или MFM-кодированием, и наиболее распространен в дисковых накопителях для персональных компьютеров.



Дискеты с маркой DSDD имеют две рабочие стороны (Double-Sided) и рассчитаны на двойную плотность (Double-Density) кодирования. Такое кодирование данных применяется во всех накопителях на гибких дисках и применялось в первом накопителе на жестких дисках для компьютера IBM PC/XT.

Отметим, что в MFM-кодировании переходы синхронизации требуются только при отсутствии близко расположенных переходов данных. Следовательно, большинство сигналов синхронизации можно убрать и все-таки надежно знать, какие переходы несут данные, а какие — информацию о синхронизации.

На рис. 2.7,в показано изменение намагниченности для того же набора данных, что и на рис. 2.7,б. Изменения сверху вниз или наоборот показывают переходы поля. На рис. 2.7,г показаны импульсы напряжения, возникающие в головке считывания.



Отметим, что на рис. 2.7,б имеется переход (синхронизации) в начале каждой битовой ячейки и еще один переход в середине битовой ячейки, который хранит двоичную единицу. В MFM-кодировании просто убирается максимум переходов синхронизации, но с возможностью восстановления записанных данных.

Правило MFM-кодирования формулируется довольно просто. В FM-кодировании убираем все импульсы синхронизации, кроме случая, когда данная и предыдущая ячейки содержат двоичные нули. Все битовые ячейки, хранящие единицы, и ячейки сразу после них достаточно близки к переходу, поэтому только для синхронизации дополнительных переходов не нужно.

Правило отделения при считывании сигналов синхронизации и данных также довольно простое. Необходимо различать только сигнал в начале битовой ячейки (это

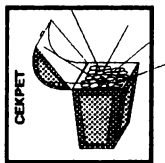
импульс синхронизации между двумя нулевыми битами) от сигнала в середине ячейки, означающего двоичную единицу.

Минимальное и максимальное расстояния переходов в этом способе кодирования данных в два раза больше, чем в FM-кодировании. Минимальный промежуток удваивается, а данные в MFM-кодировании требуют вдвое меньше длины дорожки — отсюда и появился термин "двойная плотность". FM-кодирование имеет только две частоты (f и $2f$), а в MFM-кодировании есть три частоты (f , $1.5f$ и $2f$). Требование того, чтобы контроллер различал больше частот, означает, что электронные схемы и дисковые двигатели должны быть стабильнее, чем в FM-кодировании; оказалось, что удовлетворить это требование не так сложно.

2.2.5. RLL-кодирование

Оказалось, что, несмотря на популярность и эффективность, MFM-кодирование не самый эффективный способ кодирования данных. В MFM-кодировании размер битовой ячейки уменьшается до минимальной длины магнитного триггера, но можно достичь лучшего результата при дальнейшем сокращении числа сигналов синхронизации. Степень сокращения зависит от постоянства скорости вращения диска и от точности выделения импульсов, поступающих от головки считывания.

Разработчики дисковых систем проверили много способов снижения среднего числа импульсов синхронизации на бит данных и соответствующего повышения максимальной плотности битов на поверхности диска. Наибольшее распространение, по крайней мере, для жестких дисков PC, получило так называемое кодирование с ограниченной длиной отрезка или RLL-кодирование. В этом способе совершенно нет сигналов синхронизации! Это стало возможно при записи на диск наборов, которые отличаются от наборов сохраняемых данных. При правильном выборе записываемых наборов контроллер при считывании данных может "обратить" этот процесс.

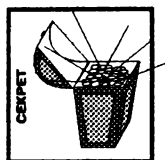


Математики и инженеры считают способ 2,7 RLL разновидностью записи с групповым кодированием (Group-coded Recording — GCR). Идея группового кодирования состоит в том, что группа битов данных заменяется большей группой записываемых битов. (Здесь у вас, конечно, появится вопрос: "Нужно ли расходовать поверхность, записывая больше битов, чем имеется в данных?" Минуточку терпения, и ответ будет очевиден.)

Обычная форма способа 2,7 RLL относится к кодированию GCR с переменной длиной. Другими словами, размер групп заменяющих битов зависит от фактических битов данных.

RLL-кодирование использует два ограничения на любой набор переходов магнитных полей, которые можно записать на поверхности диска. Во-первых, переходы магнитного поля не должны следовать чаще, чем минимальная длина магнитного триггера (этим предотвращается риск стирания предшествующего магнитного поля при записи нового). Во-вторых, промежутки без переходов не должны быть столь длинными, чтобы контроллер диска потерял текущую позицию на диске.

Таким образом, для переходов магнитного поля имеются максимальная и минимальная допустимые частоты. (Другими словами, в терминах расстояний по дорожке между переходами магнитного поля имеются максимальная и минимальная допустимые длины промежутка или "отрезка" без переходов. Кодирование RLL и означает, что эти "отрезки" дорожки между переходами ограничены верхней и нижней величинами.)



В RLL-кодировании, как и в FM- и MFM-кодировании, переходы происходят только в начале или середине любой битовой ячейки. Вместо простого преобразования поступающих битов данных в переходы контроллер анализирует их небольшими группами, и для каждой группы сохраняется специально подобранная последовательность переходов и отсутствий переходов. Далее буква Т (Transition) означает переход, а буква О (Open) — пустой участок. По дорожке переходы Т и пустые участки О занимают точно половину битовой ячейки.

В 2,7 RLL-кодировании последовательности Т и О необходимо выбрать так, чтобы независимо от поступающего потока данных всегда было минимум два и никогда не больше семи О между любыми двумя Т. При выполнении этого правила минимальное расстояние между переходами равно трем полубитовым ячейкам, а максимальное восемь.

На рис. 2.8 показана схема формирования последовательности Т и О из входного потока двоичных данных. Обратите внимание на показанные справа последовательности: некоторые из них начинаются с Т, а другие перед Т имеют несколько О. Часть последовательностей имеет внутри более одного Т, но всегда между последовательными Т есть минимум два О. Наконец, каждая последовательность имеет в конце минимум два О. В этом случае независимо от способа объединения последовательностей всегда между двумя соседними Т будет минимум два О.

Кроме того, из рис. 2.8 видно, что ни в одной последовательности нет более трех О в конце и более четырех О в начале. Следовательно, между двумя соседними Т никогда не может быть более семи О.

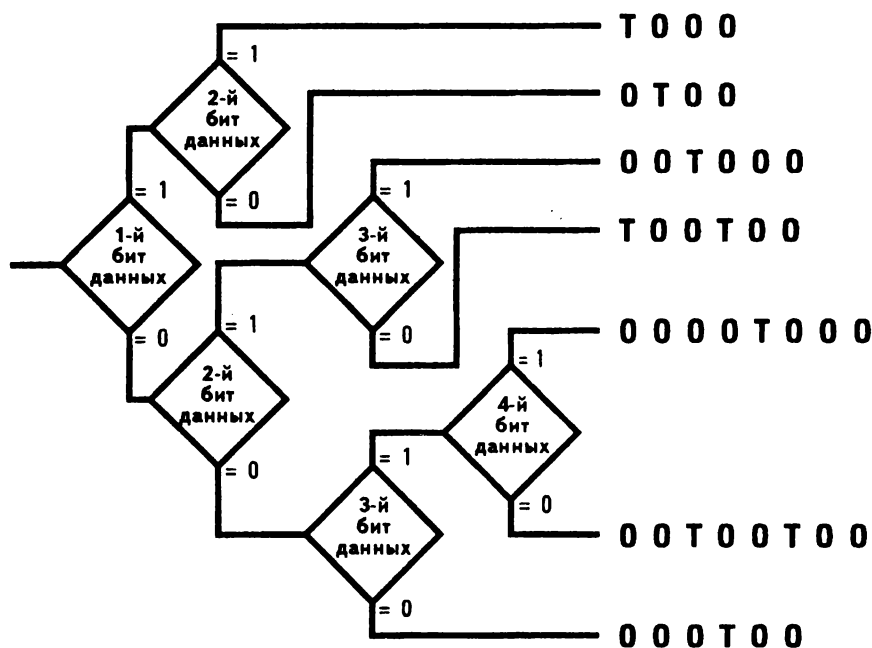


Рис. 2.8. Схема преобразования двоичного потока данных в магнитные переходы Т и промежутки между ними О в 2,7 RLL-кодировании

Каждая последовательность Т и О имеет точно в два раза больше знаков, чем набор кодируемых единиц и нулей. Поскольку от О до Т минимум три знака, то в минимальную длину магнитного триггера можно поместить три знака. Поскольку два

знака равны одному биту, размер битовой ячейки сокращается до двух третей длины триггера. Такой размер позволяет накопителю разместить в полтора раза (150%) больше битов на длине дорожки, чем в MFM-кодировании, и в три раза больше битов, чем в FM-кодировании.

Таким образом, в RLL-кодировании специальные наборы переходов заменяют фактические записываемые наборы данных. Наборы выбраны так, чтобы отношение максимального промежутка между переходами поля к минимальному было как 8 к 3. Применение такого способа позволяет в том же пространстве сохранить на 50% больше данных. Большинство жестких дисков вращаются с одинаковой скоростью, на запись и считывание одной и той же информации уходит только две трети времени. На рис. 2.7,д показано, как выглядит RLL-кодирование одного и того же байта по сравнению с другими способами кодирования. Несмотря на возможность улучшения RLL-кодирования, этот процесс оказывается непростым и недостаточно надежным для большинства пользователей. Как всегда, на практике RLL-кодирование встречает определенные ограничения.

Поскольку в RLL-кодировании переходы магнитного поля оказываются не ближе, чем в MFM-кодировании, многие полагают, что можно применять MFM-накопитель как RLL-накопитель, просто подключив к MFM-накопителю RLL-контроллер. К сожалению, при первом появлении на рынке несколько лет назад для надежного применения RLL-кодирования пришлось усложнить электронику жестких дисков и контроллера. Накопители и контроллеры довольно часто и с драматическими последствиями отказывали.

С тех пор фирмы-производители RLL-контроллеров разработали лучшие методы, а производители накопителей стали более жестко тестировать свои изделия. В результате сейчас появились накопители, "сертифицированные для RLL-кодирования", а других пока нет. Объединение RLL-сертифицированного накопителя с RLL-контроллером оказывается довольно надежным устройством.



До введения тестирования на RLL-совместимость у вас была вероятность примерно в 50% на то, что MFM-накопитель мог приемлемо работать с RLL-контроллером. Почти все такие накопители сейчас маркируются как "допускающие RLL"; по-видимому, все накопители, маркированные MFM, совместимостью с RLL не обладают.

2.2.6. Улучшенное ARLL-кодирование

Некоторым людям нравится рисковать. Часть фирм-производителей жестких дисков пошла дальше обычного 2,7 RLL-кодирования и предложили улучшенное ARLL-кодирование. Суть этих способов сводится к тому, что в них допускается большее отношение максимального промежутка между переходами магнитного поля к минимальному, что позволяет упаковать данные на дорожке почти в два раза плотнее, чем в MFM-кодировании.

Про накопители с интерфейсом ESDI (Enhanced Small Device Interface — Улучшенный интерфейс малых устройств) и контроллерами SCSI иногда говорят, что в них применяется Enhanced RLL-кодирование (ERLL). Фактически же в них применяется 2,7 RLL-кодирование, но благодаря большей скорости битов они помещают на дорожку больше данных, чем накопители и контроллеры со стандартным RLL-кодированием.

Когда в накопителе и контроллере применяется специальное кодирование данных, очень важно их согласование друг с другом. Часто такую настройку невозможно осуществить, если оба компонента не одной и той же фирмы. Поскольку накопители с интерфейсами ESDI, SCSI и IDE/AT, а также жесткие диски, смонтированные на плате, обязательно имеют встроенные контроллеры, такая настройка возможна. Когда же контроллер продается отдельно от накопителя и обещает большую плотность данных, чем RLL-кодирование, такую ситуацию следует считать пригда-

шением к потере данных. Примечание: в ESDI-накопителях применяется отдельный контроллер, но кодирование данных при записи и разделение синхронизации и данных при считывании выполняются в накопителе. Поэтому мы включили их в одну группу с IDE/AT- и SCSI-накопителями, а также дисками на плате.

2.3. Проблема несовершенства и ее решение

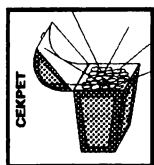
Мы разобрались с тем, как можно сохранить данные на жестком диске. Все способы кодирования должны обеспечить считывание точно той намагниченности, которая была записана. Однако ничего совершенного в реальной жизни нет. Чтобы справиться с неизбежными ошибками, производители накопителей и контроллеров вынуждены встраивать в свои изделия некоторые дополнительные возможности. В этом разделе показано, что делают производители для того, чтобы их накопители стали более совершенными, или, по крайней мере, информировали компьютер о том, когда они не могут быть совершенными. Последняя возможность позволяет компьютеру не использовать плохие данные так, как будто они хорошие.

2.3.1. Частичное решение — обнаружение ошибок

Компьютер просто сохраняет больше информации, чем ему требуется получить. Он также сохраняет дополнительные биты, которые помогают определить, можно ли доверять считанной информации.

Простейший способ обнаружения ошибок, который применяется в системном ЗУПВ компьютера, заключается во введении бита паритета. В дисковых накопителях применяется несколько более сложный и мощный способ, называемый циклическим избыточным контролем.

Паритет. В системном ЗУПВ почти всех персональных компьютеров каждый байт (8 бит) данных сопровождается девятым битом, называемым битом паритета. При записи байта в ЗУПВ компьютер вычисляет его паритет, подсчитывая число единиц в байте. Если число единиц четное, соответствующий бит паритета устанавливается в 1, а в противном случае он сбрасывается в 0. Во всех девяти битах всегда должно быть нечетное число единичных битов, а иное означает возникновение ошибки. Отсюда появился и термин "нечетный паритет" (или нечетность). Если любой бит считан или сохранен неправильно, паритет оказывается нарушенным и компьютер узнает об ошибке. Конечно, компьютер не может определить ошибочный бит.



Если искажено четное число битов, паритет не обнаружит ошибку. К счастью, большинство ошибок в ЗУПВ приходится на битовые и даже они возникают редко. Компьютер не может исправить ошибочные биты, поэтому он просто извещает об ошибке пользователя и останавливается. О таких ошибках свидетельствуют сообщения PARITY CHECK ONE и PARITY CHECK TWO.

Некоторые клоны в такой ситуации предоставляют больше информации. Они могут вывести примерно такое сообщение: I/O memory error at XXXX (Ошибка ввода-вывода памяти в XXXX). Буквы X заменяются 16-ричными цифрами. Показанное 16-ричное число представляет собой адрес, по которому компьютер производил считывание при появлении ошибки паритета. После вывода сообщения клон, как и IBM PC, по-видимому, остановится.

Циклический избыточный контроль. Когда MS DOS записывает информацию на гибкий диск, она использует аналогичный способ защиты. Во-первых, она группирует информацию в секторы по 512 байт. После этого в конце данных она добавляет еще два байта. MS DOS вычисляет 16-битовое число, комбинируя все 512 байт сек-

тора специальным образом, который называется циклическим избыточным контролем (Cyclical Redundancy Check — CRC).

Как и паритет, CRC является средством проверки искаженных данных. Он может выявить все однобитные и часть многобитных ошибок, но искаженные биты определить не может. Сообщение Error reading drive A (Ошибка считывания в накопителе A) обычно означает, что компьютер считал сектор с неправильным значением CRC.

2.3.2. Лучшее решение — исправление ошибок

Жесткие диски сохраняют значительно больше данных и с большей скоростью, чем гибкие диски, поэтому вероятность ошибок возрастает. Без средств обнаружения и исправления ошибок в процессе считывания появится недопустимо высокая частота ошибок, например несколько раз в день.

К счастью, математики и инженеры разработали так называемые коды с исправлением ошибок (Error Correction Codes — ECC). Во всех современных контроллерах жестких дисков есть специальные схемы для формирования и использования этих ECC-чисел. Большую часть времени пользователь даже не знает о появлении дисковых ошибок; контроллер диска благодаря ECC просто автоматически исправляет ошибки.

"Садовая" модель ECC. Для понимания некоего "волшебства" ECC требуется немного терпения. До обсуждения работы ECC в контроллере жесткого диска полезно обратиться к более простой для понимания модели. Предположим, что вы хотите защищать 16 бит информации. Выстроим эти единицы и нули, как деревья в саду, в четыре строки и четыре столбца. Для каждой строки и столбца при наличии в них четного числа единиц добавим дополнительный единичный бит; при наличии нечетного числа единиц в строке или столбце добавим нулевой бит; наконец, введем последний бит в месте пересечения дополнительных строки и столбца, поместив в него 1 или 0 так, чтобы сделать нечетным число единиц в нижней строке (рис. 2.9).

При такой организации каждая строка и столбец имеют нечетный паритет. Если искажается любой бит, его можно найти, потому что содержащие его строка и столбец больше не имеют нечетного паритета. Можно не только обнаружить ошибочный бит, но и исправить его!

Такая возможность представляется почти волшебной. Многие люди, сталкиваясь с необходимостью правильного хранения данных, полагают, что, как минимум, необходимо хранить дополнительную копию каждого бита. При расхождении двух копий можно наверняка сказать, что одна из них содержит ошибку. Разумеется, такой подход недостаточен для исправления ошибки, так как невозможно узнать, какая копия правильная.

Идея возможности обнаруживать и исправлять ошибки, применяя умеренный объем избыточной информации (меньше полной копии оригинальных данных), представляется удивительной и неожиданной. Но именно благодаря ей ECC позволяет компьютеру обнаруживать и исправлять ошибки.

"Садовая" модель обладает такими замечательными свойствами, но только для однобитных ошибок; поэтому она хорошо показывает сущность ECC. Отметим, что для защиты от однобитных ошибок 16 бит данных потребовались дополнительные 9 бит. Эффективность защиты значительно возрастает, если выбрать большее число защищаемых битов.

Предположим, что необходимо защитить от ошибок один сектор информации на жестком диске. Сектор содержит 512 байт или 4096 бит. Такой размер означает, что можно создать "сад", имеющий 64 строки и 64 столбца. Для завершения "сада" требуется 129 бит, обеспечивающих защиту от всех однобитных ошибок.

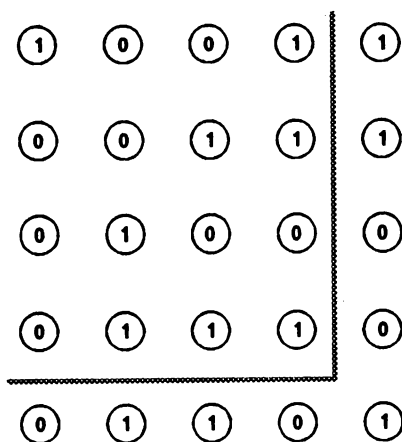


Рис. 2.9. "Садовая" модель показывает принцип построения кодов с исправлением ошибок (ECC)

Несмотря на то, что рассмотренный "садовый" подход на практике не применяется, он все же наглядно иллюстрирует принципы всех практических ЕСС. В этой модели отсутствуют две вещи. Во-первых, данный способ не справляется с многобитными ошибками. Во-вторых, несмотря на приемлемую эффективность, "садовая" модель не обеспечивает максимальной эффективности.

"Садовая" модель полезна и в том отношении, что она помогает вывести явные формулы для каждого из защитных битов. Эти формулы представляют собой суммы групп битов, причем для каждой группы выбираются свои биты. (При вычислении сумм переносы игнорируются. Подобные манипуляции математики называют модульной арифметикой.) Реальный ЕСС похож на рассмотренный, но в нем применяется больше групп битов, они комбинируются более сложным способом, а сами группы выбираются довольно таинственным образом.

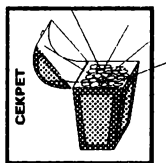
Реальный ЕСС. Фирмы-производители включают схемы ЕСС в контроллеры жестких дисков по-разному. Они считаются собственностью фирм, поэтому практически подробно разобраться в них и понять отличия друг от друга невозможно. По-видимому, это одна из причин того, что новый контроллер часто не может считать данные с жесткого диска форматированного другим контроллером. Из-за записи другим способом секторы в накопителе выглядят странными для второго контроллера и он считает жесткий диск неформатированным.

Обычно схемы ЕСС контроллера могут обнаружить и исправить любой набор ошибок во всем секторе из 512 байт (4096 бит) данных, если все ошибки содержатся в одном пакете, длина которого не превышает некоторого конкретного значения. В стандарте фирмы IBM для первого MFM-накопителя в IBM PC/XT исправление гарантировалось, если длина пакета не превышала 11 бит. Для этого в конце каждого сектора приходилось хранить всего 4 байта ЕСС. Это же количество байт ЕСС обеспечивает высокую вероятность обнаружения более длинного пакета или группы пакетов, но такие ошибки исправить нельзя.



Очень полезно просто знать о возможности таких больших ошибок. Во-первых, вы можете заставить накопитель повторно считать нужные данные. Часто этого оказывается вполне достаточно для их точного считывания. Если же ошибки остаются, вы хотя бы знаете о том, что их нельзя использовать как правильную копию.

Поскольку RLL-накопитель хранит в том же пространстве на 50% данных больше, чем MFM-накопитель, дефект на поверхности диска, который повреждает 11 соседних битов в MFM-накопителе, повредит в полтора раза больше битов в RLL-накопителе. В связи с этим, в RLL-контроллерах часто применяется больше битов ECC. Во многих RLL-контроллерах для защиты от ошибок в 17 соседних битах данных применяются 56 бит (7 байт). (Такое число по-прежнему невелико по сравнению с защищаемыми байтами.)



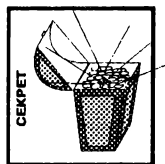
Несмотря на возможность ошибок длиннее 11 бит (или 17 бит в RLL-накопителе), такие ошибки очень редки. Большинство ошибок в жестких дисках возникают из-за дефектного пятна на поверхности, которое не позволяет надежно сохранить данные в небольшой части сектора, длина которой обычно намного меньше 11 бит.

Плохо, конечно, если у вас в пределах одного сектора несколько дефектных участков; в этом случае ECC может казаться достаточным для исправления ошибок, но его исправления могут быть неправильными. Имеется несколько способов борьбы с такими ошибками (но, к сожалению, ни один из них пока не получил распространения в накопителях на жестких дисках для персональных компьютеров).

В наиболее совершенных дисковых накопителях больших компьютеров применяется многоуровневый способ обнаружения и исправления ошибок. Каждая защищаемая область на диске (обычно целая дорожка) подразделяется на несколько подобластей, а каждая подобласть делится на меньшие секции, каждая из которых защищается своим ECC. Вся область защищается также встроенными значениями CRC.

Если при считывании области окажется, что некоторые меньшие секции содержат ошибки, они исправляются с привлечением их ECC. Значения CRC защищают подобласти и всю область и, если окажется, что исправления ECC неправильные, это можно обнаружить на уровне CRC. В этом способе не исправляются такие ошибки, но они почти наверняка обнаруживаются и проблема неправильных исправлений снимается. Возможно, в будущих разработках дисковых накопителей для персональных компьютеров этот способ будет применяться, но сейчас просто приходится надеяться на то, что подобные ошибки встречаются редко.

Что делает DOS при действии ECC. Когда контроллер диска пытается считать сектор данных с жесткого диска, могут произойти три события: вычисленный ECC соответствует ECC на диске; контроллер обнаруживает, что данные не согласуются с ECC, но ECC определяет, какими должны быть данные (это исправимая ошибка считывания); ошибка в данных столь велика, что ECC недостаточно для определения искаженных битов (это неисправимая ошибка).



Когда при считывании данных обнаруживаются ошибки, контроллер диска может привлечь ECC для исправления воспринятых данных и перейти к следующему сектору, но он может попытаться вновь и вновь считывать подозрительный сектор, определяя, нельзя ли получить правильные данные. Попытки повторного считывания данных зависят не только от имеющегося контроллера, но и от BIOS на материнской плате компьютера, а, возможно, от версии DOS и выполняемой прикладной программы.

После этого контроллер диска посылает данные в компьютер с такими соответствующими сообщениями:

1. "Вот данные. Они считаны без ошибок"
2. "Вот данные. Я не смог считать их, но определил, какими они должны быть"
3. "Вот то, что я считал, но я уверен, что это не то, что требуется"

DOS управляет дисками и не отвлекает пользователя больше чем необходимо. Например, вы не можете приказать DOS, где на диске сохранить информацию; это она определяет сама. Когда вам нужна хранимая информация, вы просто сообщаете DOS имя файла, а она сама находит данные. В общем, благодаря такому подходу компьютерами пользоваться проще, но один аспект здесь довольно интересен.

Сообщения контроллера диска поступают в компьютер и принимаются BIOS (базовой системой ввода-вывода), представляющей набор процедур очень низкого уровня для управления аппаратными средствами. После этого BIOS передает данные и сообщение о состоянии программе более высокого уровня, обычно операционной системе (которой в большинстве случаев является DOS).

Когда DOS получает одно из трех приведенных ранее сообщений, она предпринимает одно из двух действий. В случае сообщений 1 или 2 (успешное считывание или ошибка при считывании, которая почти наверняка исправлена ECC) DOS просто воспринимает данные и без комментариев передает их прикладной программе. Только если контроллер диска не может считать или восстановить данные, DOS сообщает вам (или вашей прикладной программе) о возникшей проблеме. Однако к этому моменту проблема настолько серьезна, что часть данных потеряна.

Такая ситуация неблагоприятна, так как часто дисковые ошибки начинают прогрессировать. Исправимая сейчас ошибка может со временем превратиться в непоправимую. Если бы DOS информировала вас об исправимых ошибках, вы смогли бы переместить данные в более безопасное место на диске или как-то еще заранее отреагировать на ожидаемую потерю данных.



Отметим, что авторы некоторых программ управления дисками уверяют, что они могут "выключить" ECC. На самом деле этого сделать нельзя, так как нельзя заставить контроллеры диска не применять ECC. Обычно такие программы просто отмечают использование ECC.

Заключение

В этой главе рассмотрены следующие вопросы:

- Работа жестких дисков и звуковых магнитофонов базируется на одних и тех же принципах записи и считывания цифровых данных (в случае жестких дисков) или музыки и речи (в случае звукового магнитофона).
- В накопителях один или несколько дисков с магнитным покрытием вращаются под головками считывания-записи, причем для каждой поверхности имеется своя головка. Головки перемещаются в радиальном направлении для доступа к дорожкам, на которых хранится информация.
- Все головки на одном и том же радиусе образуют цилиндр. Важность цилиндра состоит в том, что все дорожки в цилиндре (т.е. все дорожки, которые проходят под одной из головок в данной позиции блока головок) доступны с электронной скоростью (требуется только коммутация схем считывания или записи с одной головки на другую). Доступ к дорожкам на разных цилиндрах требует механического перемещения блока головок, на что требуется гораздо больше времени.
- В жестких дисках персональных компьютеров применяется изменяющийся ток через одну из головок для образования изменяющегося магнитного поля, которое, в свою очередь, создает множество постоянных магнитов на поверхности диска. Расположение этих магнитов определяет, какая информация сохранена на диске.
- Диск считывает записанную на поверхности информацию, обнаруживая границы постоянных магнитов, когда они проходят под головкой считывания.

- Двоичные данные нельзя просто записать на поверхности диска точно в том виде, в каком они подаются в контроллер диска. Вначале их необходимо некоторым образом "упаковать" или закодировать. Для жестких дисков персональных компьютеров наиболее распространены MFM- и 2,7 RLL-кодирование. RLL-кодирование эффективнее на 50%, но требует большей надежности от диска и контроллера.
- Для всех жестких дисков существуют проблемы надежности, которые просто скрыты от потребителя мощным набором средств обнаружения и исправления ошибок.
- Для обнаружения и исправления ошибок важнейшую роль играют коды ECC. Они позволяют накопителю и контроллеру обнаружить наличие в секторе поврежденных данных. Во многих случаях коды ECC позволяют восстановить данные.
- К сожалению, DOS скрывает все усилия контроллера диска по восстановлению испорченных данных. В результате вы или ваша прикладная программа узнаете о проблеме только тогда, когда уже поздно, а это приводит к значительной потере данных.

В следующей главе вы продолжите знакомство с магнитной записью данных и новейшими техническими разработками в этой области. Кроме того, там приведены сведения о наиболее популярных жестких дисках для персональных компьютеров.

Глава 3. Некоторые важные технические вопросы

В этой главе вы узнаете

- Что такое чередование секторов и как настроить его
 - Что такое перекосы головок и цилиндров и когда их необходимо регулировать
 - Различия между популярными дисковыми интерфейсами (ESDI, ST412/ST506, SCSI и IDE)
 - Чем отличаются платы жестких дисков от других жестких дисков ПК
 - Различия между логическими и физическими адресами хранимых на жестком диске данных.
-

3.1. Повышение производительности жесткого диска с помощью чередования секторов и перекоса головок

Конечно, для повышения производительности жесткого диска можно всегда приобрести более быстрый накопитель. Однако это не всегда оправдано, а при неправильной настройке чередования секторов и перекоса головок деньги будут потрачены впустую.

Для оценки скорости работы жесткого диска обычно интересуются двумя параметрами. Один из них называется временем обращения (или доступа), и показывает, как долго контроллер ищет файл на диске. Второй параметр называется скоростью передачи данных, и он показывает, как быстро контроллер может считывать файл после его локализации.

Уменьшить время обращения жесткого диска практически невозможно. (Конечно, можно разделить накопитель на несколько логических накопителей; об этом сказано в главе 9). Однако в большинстве случаев вы можете значительно повысить скорость передачи данных жесткого диска, установив оптимальные значения чередования секторов и перекоса головок.

3.1.1. Анатомия дискового сектора

В главе 2 показано, что MS DOS хранит информацию в дисковых накопителях секторами, содержащими по 512 байт данных. Кроме данных, в секторах находится кое-что еще. Поставляемый с предприятия жесткий диск необходимо подготовить для того, чтобы MS DOS и компьютер могли хранить на нем данные (подробнее в главах 8–10). Для образования секторов на каждой дорожке каждой поверхности нужно записать определенные двоичные наборы. Сектор состоит из двух основных частей: идентификатора и сегмента, в котором хранятся данные (рис. 3.1).

Идентификатор, или заголовок сектора, содержит три числа, образующих адрес сектора: головка (или поверхность), на которой находится этот сектор, номер дорожки (или цилиндра) и позиция сектора на дорожке. Заголовок содержит также поле, метка в котором показывает, может ли сектор надежно хранить данные или обнаружен дефект, исключающий использование сектора. Контроллеры некоторых жестких дисков помещают в заголовки секторов указатели, направляющие диск на альтернативные сектор или дорожку, если сектор оказался плохим. Наконец, заголовок сектора заканчивается значением циклического избыточного контроля CRC, которое контроллер привлекает для контроля правильности считывания заголовка.

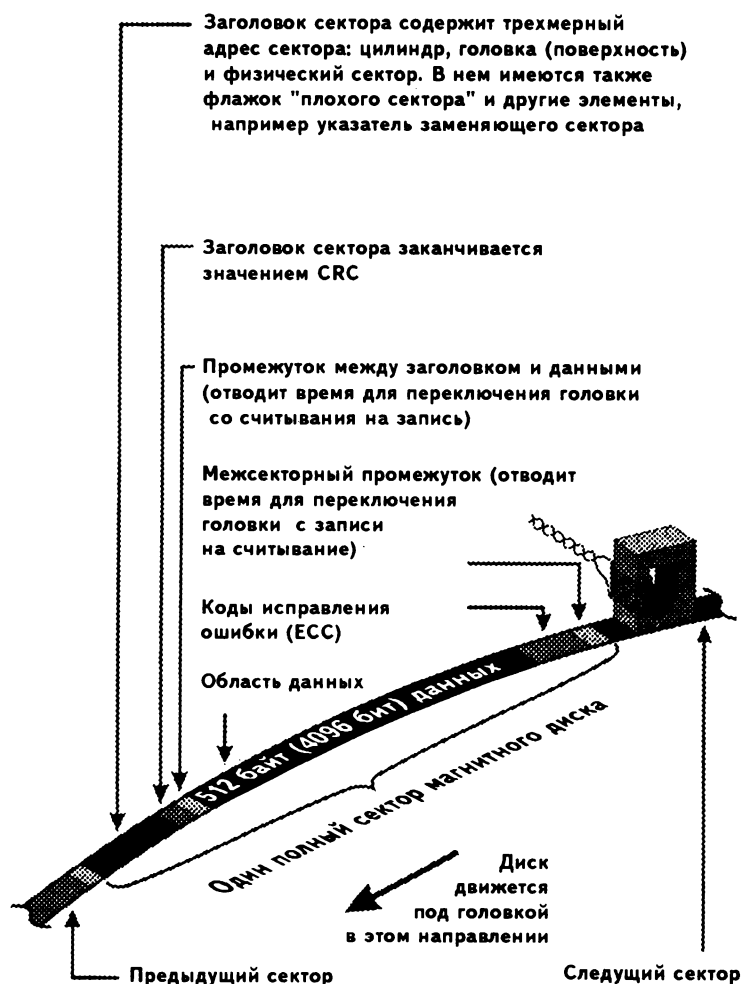


Рис. 3.1. Анатомия сектора жесткого диска

Вторую часть сектора, в которой компьютер хранит данные, можно разделить на собственно данные и коды исправления ошибки ЕСС, защищающие данные. В процессе начальной подготовки компьютер заполняет эту область 512 байтами фиктивной информации и значениями ЕСС, соответствующими этим байтам.

3.1.2. Чередование секторов

Заголовок сектора содержит номер сектора, который идентифицирует данный сектор на дорожке. Номера не обязательно назначать в каком-то конкретном порядке. Формат заголовка сектора позволяет нумеровать секторы от 1 до некоторого максимального значения, например 256.

Контроллер диска не заботится о том, какой номер из указанного диапазона помещен в заголовок конкретного сектора; секторы могут разделять один и тот же номер, хотя такая ситуация самая необычная. Контроллер диска не заботится даже о размере области данных. Контроллер просто считывает все, что находит, или записывает то, что ему приказано записать.



Наиболее распространенное применение такой возможности связано с гибкими дисками, на которых поставляются некоторые защищенные от копирования программы. Такие дискеты имеют минимум на одной дорожке несколько нестандартных секторов. Команды DOS COPY и DISCOPY не могут дублировать такие секторы, на чем и основана защита от копирования. Программы работают только при наличии на дискете в накопителе А специальных особо пронумерованных секторов, возможно, необычного размера.

С другой стороны, программа подготовки диска полагает, что секторы нумеруются в некотором простом порядке. Например, секторы можно пронумеровать 1, 2, 3, 4, 5 и т.д., но такой порядок необязателен.

Другой способ нумерации секторов называется чередованием секторов. Такое название объясняется тем, что группы секторов с соседними номерами чередуются с другими группами секторов, имеющими другие соседние номера. Можно считать также, что после нумерации одного сектора вместо назначения следующего номера следующему сектору вы пропускаете ("перепрыгиваете") через одно и то же число секторов, а затем назначаете следующий номер.

Величина чередования определяется коэффициентом чередования, представленным в виде двух чисел, разделенных двоеточием. Если, например, коэффициент равен 3:1, то после назначения первому сектору на дорожке адреса 1 два следующих сектора пропускаются, а четвертому сектору на дорожке назначается адрес 2. После этого вновь пропускаются два сектора, а седьмому сектору назначается адрес 3. Этот процесс продолжается до назначения адресов всем секторам. (Пока такой способ нумерации кажется надуманным и искусственным, но далее показано, что он имеет большое практическое значение.)

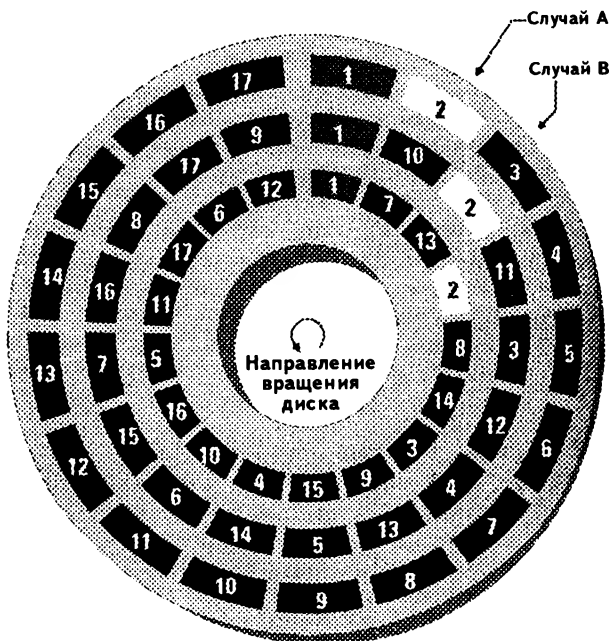


Рис. 3.2. Примеры разных коэффициентов чередования

Чтобы посмотреть, какой выигрыш дает выбор конкретного коэффициента чередования секторов, рассмотрим три случая, показанных на рис. 3.2 для трех дорожек диска. Число секторов на дорожке 17, что типично для накопителей с MFM-кодированием; дорожки показаны так, как будто они фактически записаны на диск, но, конечно, их ширина намного больше ширины реальных дорожек. (В современных накопителях число секторов на дорожке больше, обычно от 23 до 75 секторов, а иногда число секторов на дорожке не постоянно.)

Секторы на внешней дорожке (цилиндр 0) пронумерованы в простейшем порядке (последовательно), что соответствует коэффициенту чередования секторов 1:1. При вращении диска секторы проходят под головкой считывания-записи в порядке 1, 2, 3, 4, 5 и т.д. Секторы на соседней дорожке (цилиндр 1) пронумерованы с коэффициентом чередования секторов 2:1, а на третьей дорожке (цилиндр 2) — с коэффициентом чередования секторов 3:1.

Рассмотрим теперь, что произойдет, когда диск вращается, а контроллер пытается считать сектор номер 1 и сразу после этого сектор номер 2. Стрелки, отмеченные как "Случай 1" и "Случай 2", показывают две точки, в которых головка может начать считывание после обработки ("переваривания") данных, считанных из сектора 1. В "Случае А" второй сектор на дорожке уже частично прошел под головкой считывания-записи к тому времени, когда головка готова начать считывание второго сектора, а третий сектор еще не подошел. В этой ситуации коэффициент чередования 2:1 оказывается оптимальным. В "Случае Б" даже третий сектор частично прошел под головкой, когда она готова считывать этот сектор, а четвертый сектор еще не появился под головкой. Здесь оптимальным будет коэффициент чередования 3:1.

Как DOS хранит и считывает файлы. Если вы просите DOS сохранить файл на диске и все секторы в этой области доступны, DOS последовательно заполняет секторы в порядке их номеров наборами по 512 байт. DOS переходит к следующей головке в том же цилиндре, когда первая дорожка заполнена, и продолжает последовательное заполнение секторов.

Данный процесс продолжается до заполнения цилиндра. Здесь при наличии оставшихся для записи данных DOS командует контроллеру диска передвинуть все головки на следующий цилиндр (и контроллер перемещает блок головок на одну дорожку). Весь этот процесс продолжается до записи на диск всего файла. DOS будет считывать секторы в том же порядке. DOS считывает сектор, сообщая контроллеру диска номера цилиндра, головки и сектора (три компонента физического адреса) считываемого сектора.

В свою очередь, контроллер диска просто перемещает блок головок на нужный цилиндр, выбирает нужную головку и ожидает появления нужного сектора под головкой. Контроллер считывает заголовок каждого сектора, сравнивает адресную информацию из заголовков с отыскиваемыми цилиндром и головкой (проверяя правильность работы привода блока головок) и отыскивает нужный номер сектора. Когда контроллер находит заголовок этого сектора, он либо включает схему записи или считывает данные и "хвост" в зависимости от выполняемой операции записи или считывания.

После локализации сектора контроллер диска должен обработать информацию из этого сектора до перехода к следующему сектору. При считывании данных контроллер должен вычислить ЕСС для считанных данных и сравнить его с записанным ЕСС; при записи контроллер должен вычислить ЕСС для сохранения вместе с данными. Когда контроллер проводит обработку информации от данного сектора, диск продолжает вращаться. Обработка информации занимает продолжительное время, поэтому диск может повернуться за это время на заметную величину.

Простейший (и иногда наихудший) случай. Напомним, что в рассматриваемом примере секторы по дорожке пронумерованы последовательно. Если во время обработки контроллером данных одного сектора диск поворачивается больше, чем на (небольшой) межсекторный промежуток, следующий сектор, который должен участво-

вать в операции считывания или записи, уже пройдет под головкой на, возможно, значительное расстояние. В этом случае контроллер диска должен ожидать, пока диск не сделает почти полный оборот, прежде чем нужный сектор подойдет к головке.

Отметим, что секторы на трех дорожках рис. 3.2 пронумерованы по-разному. Обратимся вначале к внешней дорожке. Ее секторы пронумерованы последовательно, т.е. так, как только что рассмотрено в примере. Остальные дорожки рассматриваются далее.

Стрелка, обозначенная "Случай А", показывает ту точку на диске, где система может начать считывание после обработки информации, считанной из сектора 1. Отметим, что эта стрелка показывает на середину сектора внешней дорожки (сектор 2). Этот пример соответствует только что описанной ситуации, в которой контроллер диска не готов начать считывание сектора 2 после того, часть сектора уже прошла ПОД головкой. Так как контроллер должен ожидать до тех пор, пока он не "увидит" под головкой заголовок сектора 2, ему до начала считывания придется ожидать почти полный оборот диска.

В этой ситуации контроллер жесткого диска обеспечивает считывание за каждый оборот всего одного сектора. При такой скорости данные нельзя считать или записать быстрее, чем на гибкий диск. К счастью, немного изменив нумерацию секторов, можно сделать очень много для повышения скорости.

Как чередование секторов решает проблему. Много лет назад один из талантливых инженеров фирмы IBM нашел способ решения этой проблемы. Он не захотел увеличивать межсекторный промежуток, поскольку это уменьшает емкость накопителя. Вместо этого он перенумеровал секторы не в последовательном порядке. Например, для получения коэффициента чередования 2:1 для дорожки с 17 секторами применяется такая нумерация: 1, 10, 2, 11, 3, 12, 4, 13, 5, 14, 6, 15, 7, 16, 8, 17, 9 и далее вновь 1 (см. среднюю дорожку на рис. 3.2). Для коэффициента чередования 3:1 получается такая нумерация: 1, 7, 13, 2, 8, 14, 3, 9, 15, 4, 19, 16, 5, 11, 17, 6, 12 и вновь 1 (см. внутреннюю дорожку на рис. 3.2).

Таким образом, показатель чередования на единицу больше числа пропускаемых секторов между секторами с соседними номерами (например, при коэффициенте чередования 2:1 пропускается один сектор). Кроме того, показатель чередования равен числу обходов по дорожке для считывания всех секторов по порядку, и ни один из секторов не проходит до того, как контроллер будет готов к его восприятию. Например, при коэффициенте чередования 1:1 считывание всех секторов на одной дорожке требует всего одного оборота — если, конечно, контроллер диска может обрабатывать информацию достаточно быстро. Если контроллер диска оказывается недостаточно быстрым, ему для считывания всей дорожки потребуется столько оборотов, сколько секторов имеется на одной дорожке.

Увеличение коэффициента чередования до 2:1 означает, что дорожка должна пройти под головкой минимум дважды, чтобы считать или записать все секторы последовательно. При этом замедляется появление секторов под головкой так, что контроллер диска будет готов к моменту появления нового сектора. Если коэффициент чередования 2:1 оказывается недостаточно медленным, для считывания или записи всей дорожки диск должен сделать 17 или даже 18 оборотов (при наличии на дорожке 17 секторов и соответственно больше оборотов при большем числе секторов).

Средняя дорожка на рис. 3.2 показывает, что если задержка между считыванием одного сектора и готовностью системы к считыванию другого представлена стрелкой "Случай А", то коэффициент чередования 2:1 оказывается оптимальным. Следующим сектором, который появляется после готовности системы в считыванию (после обработки сектора 1), будет именно сектор 2.

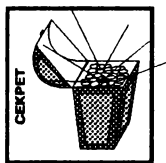
Внутренняя дорожка на рис. 3.2 показывает, что если задержка больше и головка находится в позиции "Случай Б", когда контроллер готов к считыванию следующего сектора, наилучшим оказывается коэффициент чередования 3:1. В этом случае коэффициент чередования 2: 1 столь же плох, как и 1:1, так как оба эти коэффициента

заставляют контроллер ожидать полного оборота для доступа к содержимому сектора со следующим номером.

Чередование — общесистемная проблема. При определении идеального коэффициента чередования для данного накопителя приходится учитывать несколько факторов. Кроме быстродействия контроллера диска необходимо учесть частоту синхронизации материнской платы и, в частности, скорость работы шины ввода-вывода того разъема, в который вставлен контроллер. После того как большинство контроллеров диска считало сектор, до считывания следующего сектора они должны передать данные в компьютер. Аналогично контроллеры диска не воспринимают данные для записи в следующий сектор до окончания записи в текущий сектор. Следовательно, скорость работы шины ввода-вывода компьютера может повлиять на правильное значение коэффициента чередования.

Все эти факторы показывают, что определить правильный коэффициент чередования можно только на системном уровне, т.е. только тем человеком, который собирает диск, контроллер и компьютер.

Из-за сложности определения и задания правильного коэффициента чередования секторов и значительных усилий для получения ответа (особенно до появления программ типа SpinRite) до недавнего времени почти во всех персональных компьютерах чередование было неправильным. С расширением популярности IDE- и SCSI-накопителей, которые обычно правильно работают с коэффициентом чередования 1:1, эта проблема исчезает. Однако при наличии в вашем компьютере старого накопителя вполне вероятно, что его коэффициент чередования далеко не оптимален.



Во многих современных накопителях (особенно в накопителях IDE и SCSI и платах жестких дисков) встроен буфер дорожки. Этот буфер позволяет накопителям считать сразу целую дорожку, не обрабатывая каждый сектор до считывания следующего.

Иногда проблема остается, но только при переходе операции считывания с одной дорожки на следующую. Это случается намного реже перехода от одного сектора к следующему, и особой роли эта проблема не играет. (Когда же она оказывается важной, ее решает перекос головки, рассматриваемый далее.) Некоторые контроллеры имеют даже два буфера дорожки: один для хранения считываемых или записываемых данных, а второй — для обработки или подготовки к предыдущей или следующей дорожке.

Важный практический результат такого подхода состоит в том, что эти накопители могут работать с коэффициентом чередования 1:1 независимо от быстродействия системного блока. Быстродействие системного блока влияет только на то, сколько времени должно пройти после окончания обработки одной дорожки до перехода к следующей.

Некоторые контроллеры ST412/ST506 и ESDI также имеют буферы дорожек и могут поддерживать коэффициент чередования 1:1 для управляемых ими дисков.

Если коэффициент чередования вашего диска установлен слишком большим, в дисковых операциях придется ожидать лишь немного больше времени для получения данных. Если же коэффициент чередования слишком мал, производительность диска резко снижается (затемненные участки в табл. 3.1).

В табл. 3.1 показывают влияние неправильного коэффициента чередования в некоторых распространенных ситуациях. Единственный способ проверить правильность коэффициента чередования заключается в том, чтобы выполнить программу, которая определяет производительность диска в вашей системе для нескольких значений коэффициента чередования. Такой прием характерен для программы SpinRite и большинства программ регулировки коэффициента чередования.

Таблица 3.1. Снижение производительности из-за неправильного коэффициента чередования секторов

Пиковая скорость передачи данных, Кбайт/с					
Минимальный коэффициент чередования, допустимый в системе без пропуска последовательно пронумерованных секторов:	1:1	2:1	3:1	4:1	5:1
MFM (17 секторов на дорожке)					
Коэффициент чередования 1:1	510	30	30	30	30
Коэффициент чередования 2:1	255	255	30	30	30
Коэффициент чередования 3:1	170	170	170	30	30
Коэффициент чередования 4:1	128	128	128	128	30
Коэффициент чередования 5:1	102	102	102	102	102
RLL (26 секторов на дорожке)					
Коэффициент чередования 1:1	780	30	30	30	30
Коэффициент чередования 2:1	390	390	30	30	30
Коэффициент чередования 3:1	260	260	260	30	30
Коэффициент чередования 4:1	195	195	195	195	30
Коэффициент чередования 5:1	156	156	156	156	156
ESDI (34 сектора на дорожке)					
Коэффициент чередования 1:1	1620	30	30	30	30
Коэффициент чередования 2:1	810	810	30	30	30
Коэффициент чередования 3:1	540	540	540	30	30
Коэффициент чередования 4:1	405	405	405	405	30
Коэффициент чередования 5:1	324	324	324	324	324
Приведенные числа получены на основе следующей информации:					
Данные в различных способах кодирования					
	MFM	RLL	ESDI		
Оборотов в секунду	60	60	60		
Килобайт в секунду	0.5	0.5	0.5		
Секторов на дорожке	17	26	34		
Максимальная скорость передачи данных	510	780	1620		
Скорость передачи для одного сектора на оборот	30	30	30		

Некоторые накопители RLL имеют от 23 до 27 секторов на дорожке. Диапазон для ESDI-накопителей составляет от 34 до 56 секторов на дорожке. Некоторые SCSI- и IDE-накопители имеют до 75 секторов на дорожке и часто переменное число секторов в разных зонах диска.



До появления программ типа SpinRite были другие программы, например NORTIMUM, которые могли помочь определить текущий коэффициент чередования жесткого диска и то, каким он должен быть для достижения оптимальной производительности. К сожалению, программы этого раннего поколения работали с разрушающим переформатированием одной или нескольких дорожек и уничтожением данных, если вы были невнимательными или неопытными. Такие программы не подходят для боязливых или безынициативных пользователей. Современные программы типа SpinRite стали совершенно безопасными для любого пользователя.



Все современные программы изменения коэффициента чередования диска безопасны за исключением программы IAU (Interleave Adjustment Utility).

3.1.3. Перекос головки

Многие пользователи плохо разбираются в чередовании секторов. Следующий вопрос — перекас головки — также довольно сложен, но понимание чередования секторов поможет разобраться и с ним.

Напомним действия DOS при записи файла на диск. Вначале DOS заполняет одну дорожку, а затем переходит к следующей головке в том же цилиндре; когда цилиндр заполнен, DOS переходит к следующей дорожке. Каждый переход с одной дорожки на другую и с одного цилиндра на следующий занимает время, в течение которого диск неумолимо вращается. С учетом этого обстоятельства нетрудно понять, что производительность диска снижается при считывании или записи файла, который пересекает границу дорожек или цилиндров.

Идея перекаса головки достаточно проста. Предположим, что вы только что закончили запись в последний сектор на одной дорожке. Полагая, что коэффициент чередования установлен оптимальным, теперь вы готовы записывать в первый сектор следующей дорожки. Но вам придется подождать переключения головки и перемещения блока головок на следующую дорожку. Если эти операции занимают заметное время, головка опоздает, несмотря на чередование. Решение состоит в том, чтобы сдвинуть номера всех секторов на новой дорожке на одну (или более) позиций относительно их позиций на предыдущей дорожке.

Чтобы подробнее разобраться в происходящем, рассмотрим простой пример. Почти для каждой комбинации накопителя и контроллера, общая производительность которой достаточно высока, чтобы оправдать введение перекаса, оптимальный коэффициент чередования составляет 1:1. Предположим, ради простоты, что при переходе с одной дорожки на другую оптимальная производительность достигается, если дорожка на следующей поверхности смещена относительно только что считанной на один сектор. Перемещение головок длится дольше, чем коммутация с одной головки на другую. По-прежнему, ради простоты, предположим, что при переходе с одной дорожки на соседнюю достаточно смещения в три сектора.

Рассмотренный пример показан на рис. 3.3 для двухдискового накопителя (четыре головки) с коэффициентом чередования секторов 1:1 на одной поверхности, перекасом головок +1 и перекасом цилиндров +3. Отметим, что в накопителе есть

всего два диска; дорожки снизу каждого диска показаны так, как будто вы смотрите сквозь диск. Предположим теперь, что происходит запись (или считывание) файла в последовательно пронумерованные секторы и для файла требуется много дорожек.

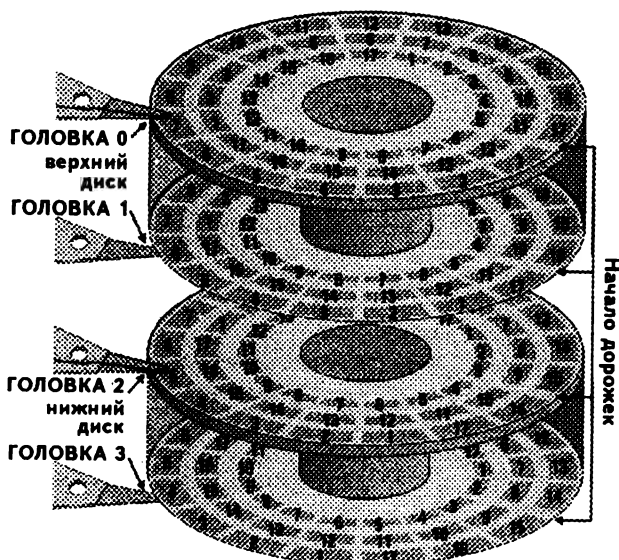
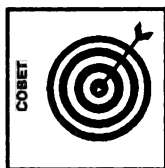


Рис. 3.3. Перекос головки

- ✓ На первой дорожке (головка 0, цилиндр 0 — это внешняя дорожка верхней поверхности верхнего диска) секторы пронумерованы последовательно с номера 1, начиная сразу после промежутка дорожки. Следующая дорожка, которая записывается или считывается в той же позиции, находится на противоположной стороне верхнего диска — это дорожка 0 для головки 1. Отметим, что номера ее секторов начинаются с 17, а затем идут номера 1, 2, 3 и т.д. Такое смещение на один сектор при переходе с верхней стороны на нижнюю данного диска является примером перекоса головки +1.
- ✓ После записи всех секторов на дорожке с головкой 0, цилиндром 0 и всех секторов на следующей дорожке (головка 1, цилиндр 0) следующей должна записываться дорожка на верхней стороне нижнего диска (головка 2, цилиндр 0). Отметим, что на этой дорожке секторы нумеруются (после конца промежутка дорожки) как 16, 17, 1, 2 и т.д. Эти номера секторов смещены относительно секторов предыдущей дорожки точно на один сектор — по-прежнему перекос головки равен +1.
- ✓ Такой порядок сохраняется до достижения последней дорожки в цилиндре 0 (головка 3). Последняя дорожка в цилиндре 0 начинается с номера сектора 15.
- ✓ Теперь блок головок должен перейти на цилиндр 1 и запись продолжается на верхнем диске с использованием головки 0. Чтобы отвести время на перемещение головок, внешняя дорожка нумеруется, начиная с сектора 12. Напомним, что последняя дорожка (головка 3, цилиндр 0) начиналась с номера сектора 15. Вычитание 3 для учета перекоса цилиндра дает 12 для начального номера сектора этой дорожки.



Некоторые высокопроизводительные контроллеры диска предоставляют пользователю возможность изменения перекаса головки, но большинство контроллеров этого допускает. Пока вопрос о перекасе головки не привлекает достаточного внимания, но, без сомнения, при появлении более быстрых дисковых накопителей ему (в частности, контроллерам и программам, которые помогают установить перекас) будет уделяться большее внимание.

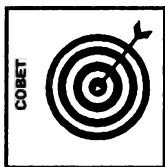
3.1.4. Где хранится информация о чередовании секторов и перекасе головки

Номера секторов хранятся в заголовках секторов, поэтому значения чередования секторов и перекасе головки определяются при записи информации в заголовки секторов. Первоначально программа подготовки диска (форматтер низкого уровня) решает эту задачу, просто вызывая специальную функцию контроллера диска. Поскольку этот процесс разрушает все данные на дорожках, которые подвергаются форматированию низкого уровня, он выполняется как можно реже.

Раньше для изменения коэффициента чередования приходилось резервировать все данные, осуществлять форматирование низкого уровня, вновь создавать таблицу разделов, производить форматирование высокого уровня, а затем восстанавливать все данные. Все эти операции занимали минимум один рабочий день. (Смысл упомянутых терминов рассмотрен в главах 8–10.)

3.1.5. Как изменить чередование и перекас

Появление неразрушающих программ изменения коэффициента чередования типа SpinRite позволило выполнять эту операцию в любое время. Теперь можно легко проверить различные значения и установить оптимальный коэффициент чередования для вашего компьютера.



Несмотря на то, что всегда полное резервирование системы целесообразно осуществить до первого применения утилиты, работающей с жестким диском на таком "глубоком" уровне, вам вряд ли оно потребуется. После однократной проверки совместимости вашей системы и программы изменения чередования можно применять ее, без предварительного полного резервирования.

Изменить перекасы головки и цилиндра сложнее. К счастью, они сказываются только при считывании или записи файла, который переходит за границы дорожки, поэтому их отрицательное влияние намного меньше неправильно установленного коэффициента чередования. Считая себя достаточно опытным пользователем, вы можете поэкспериментировать с одной из программ изменения перекаса, например программой НТНФ фирмы Kolod Research.

3.1.6. Когда и как можно установить неправильное чередование

Если вы хотите установить коэффициент чередования, отличающийся от оптимального, следует задать его большим. Напомним, что при оптимальном коэффициенте чередования 3:1 установка его на 4:1 означает считывание всей дорожки за четыре оборота вместо трех. С другой стороны, установка коэффициента чередования 2:1 приведет к тому, что число оборотов возрастает до 17 и даже до 18.

Но зачем вообще устанавливать неоптимальный коэффициент чередования? Напомним еще один факт: оптимальное значение зависит от нескольких факторов, включая скорость работы шины компьютера. Если в разное время шина работает с

различной скоростью (например, компьютер Turbo XT может работать на частоте 4,77 или 10 МГц), оптимальный коэффициент чередования для двух частот будет неодинаков. Поэтому нужно установить оптимальный коэффициент чередования для низшей скорости шины, если, конечно, вы не собираетесь большую часть времени работать с высокой скоростью.

3.1.7. Несовместимые значения чередования

Коэффициент чередования секторов определяется номерами, записываемыми в заголовки секторов, каждый диск может иметь и имеет свой коэффициент чередования. В большинстве накопителей все дорожки имеют один и тот же коэффициент чередования, но это не обязательно.

Если прервать работу программы изменения чередования, то некоторые дорожки окажутся измененными, а другие нет. Такая несовместимость не окажет заметного влияния на работу компьютера. Просто дорожки с оптимальным коэффициентом чередования будут действовать быстрее остальных.

3.2. Популярные изделия и стандарты

При появлении IBM PC в 1981 г. жесткий диск не был стандартным компонентом компьютеров. Фирма IBM не ввела его не потому, что для персональных компьютеров не было жестких дисков; дисковые накопители уже применялись в системах Apple II, CP/M и S-100. Просто жесткие диски считались слишком дорогими для индивидуального домашнего компьютера.

К 1983 г. фирма IBM поняла, что можно успешно продавать ПК небольшим конторам и что люди готовы платить деньги за большую емкость и удобство жесткого диска. Поэтому в компьютере IBM PC/XT, появившемся в 1983 г., жесткий диск на 10 Мбайт стал стандартным компонентом.



Тот факт, что фирма IBM определила PC/XT как имеющий "пятнадцатилетний срок службы" (среднее время между отказами 8,000 часов), показывает, что она не рассчитывала на эксплуатацию компьютера более 40 часов в неделю, а сейчас многие ПК работают круглосуточно.

Фирма IBM реализовала жесткий диск с помощью съемной платы, которая вставляется в один из разъемов (слотов) материнской платы PC/XT. Эта плата, называемая Адаптером фиксированного диска, соединялась с накопителем двумя плоскими кабелями (рис. 3.4).

Как видно на рис. 3.4, в подсистеме жесткого диска типичного ПК плата контроллера вставлена в шину ввода-вывода системного блока (в один из разъемов материнской платы). В посадочном месте монтируются один или два накопителя на жестких дисках. Накопители подключаются к контроллеру плоскими кабелями. Кроме того, накопители подключены непосредственно к блоку питания. (Такие монтаж и кабели применяются не только в ПК, но они наиболее распространены.)

В PC/XT накопитель соединен с блоком питания четырехжильным кабелем, аналогичным кабелю питания гибкого диска. Во многих последующих вариантах накопителей для PC питание осуществлялось таким же способом. Правда, в некоторых современных конструкциях в силовом кабеле имеются только два проводника. Кроме того, жесткие диски, монтируемые на платах, и некоторые накопители получают питание от шины ввода-вывода. IDE

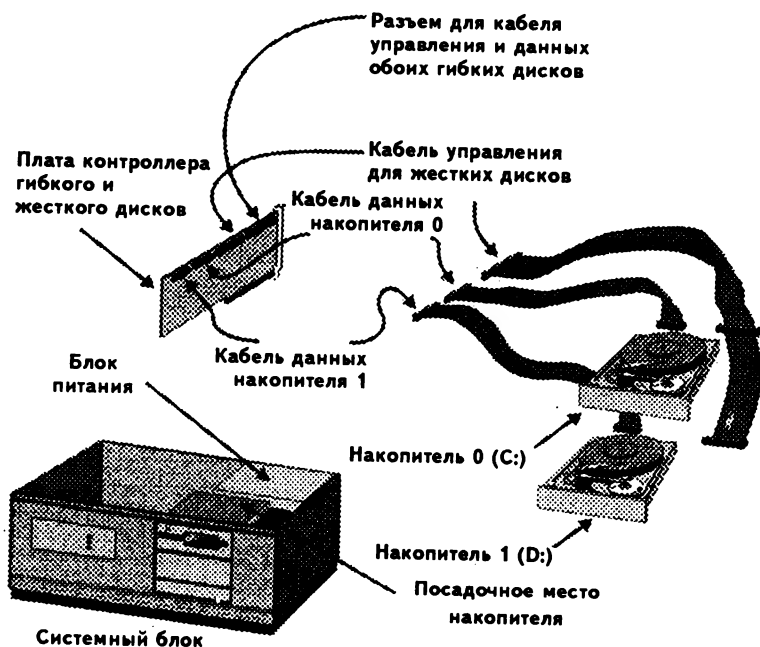
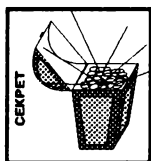


Рис. 3.4. Основные аппаратные элементы подсистемы жесткого диска типичного персонального компьютера



Можно вставить оригинальные контроллер жесткого диска и жесткий диск XT в компьютер IBM PC, обеспечив достаточную мощность блока питания для жесткого диска; собственный блок питания IBM PC недостаточно мощный.

Если вы решитесь на такую модернизацию, нужно проверить также, что микросхема ROM BIOS не устарела. Если на материнской плате вашего ПК есть только 64 Кбайт ЗУПВ, то ROM BIOS устарел и вам придется приобрести ROM BIOS, датированный октябрём 1982 г. В этой и всех последующих микросхемах ROM BIOS есть процедура сканирования ПЗУ, без которой контроллер жесткого диска (и некоторые другие съемные платы) не работает. Во всех клонах PC и последующих компьютерах самой фирмы IBM уже имеются достаточно современные ROM BIOS. (Многие диагностические программы, например CheckIt, показывают дату ROM BIOS. Можно также воспользоваться программой DEBUG и ввести команду d FFFF:5 <Enter>. Для выхода из DEBUG нужно ввести команду Q <Enter>.)

Как и для многих других компонентов PC и XT, фирма IBM применила для жестких дисков технологию "с полки". Она выпускает свои контроллеры, которые очень похожи на некоторые существующие контроллеры, и поставляет накопители, удовлетворяющие спецификациям фирм, которые выпускают накопители на жестких дисках.

Из-за большого объема продаж компьютеров IBM PC, XT, AT и PS/2 их базовая конструкция стала стандартом. Именно они определяют то, что сейчас называется персональным компьютером. (Конечно, есть и другие модели компьютеров, но львиная доля их приходится на IBM-совместимые.)

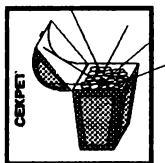
Выбранная фирмой IBM конструкция контроллеров и накопителей на жестких дисках определила важный стандарт для таких изделий. Специального названия для этого стандарта фирма IBM никогда не вводила. Конструкция первых накопителей

на жестких дисках в IBM PC/XT очень напоминала конструкцию двух более ранних разработок фирмы Seagate Technology Company, поэтому сейчас такая конструкция называется интерфейсом ST412/ST506. (Можно встретить также обозначения ST412/506, ST506/412 или просто ST506.)

3.2.1. Интерфейс ST412/ST506

Данное обозначение с номером модели описывает конкретный интерфейс жесткого диска, детально определяющий, как накопитель и контроллер "общаются" друг с другом. Взаимодействие контроллера с компьютером определяется шиной ввода-вывода ПК, которая соединяет все разъемы материнской платы, так как контроллер жесткого диска вставляется в эту шину.

Спецификация данного (и любого другого) интерфейса имеет несколько аспектов. Первый аспект касается физического описания кабелей, по которым взаимодействуют контроллер и накопитель, и описания разъемов на концах кабелей. Вторым оказывается электрический аспект, определяющий уровни напряжения и временную диаграмму сигналов в каждом проводнике. Далее, логический аспект связан с описанием функций сигналов в каждом проводнике. Наконец, обычно определяются критические временные соотношения, касающиеся длительностей сигналов и скорости их перехода из одного состояния в другое. Когда говорят об интерфейсе, например ST412/ST506, имеют в виду все эти аспекты.



Контроллер жесткого диска подключается к каждому накопителю двумя плоскими кабелями. Один кабель имеет 34 проводника (и 34-контактный разъем, поэтому проводники иногда называются контактами). По этому кабелю передаются управляющие сигналы в накопитель и из него.

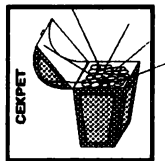
Среди сигналов от контроллера в накопитель есть сигналы, используемые для команд приводу головок перейти на одну дорожку внутрь или наружу, сигналы задания уровня тока записи для головок и сигнал, определяющий операцию считывания или записи. Накопитель выдает в контроллер сигналы об успешном окончании запрошенной операции или возникших проблемах. Имеются также сигналы о результате выполнения команды контроллера и готовности накопителя к работе.

Эти цифровые сигналы большей частью изменяются медленно. Если к контроллеру подключены два накопителя, кабель подключается вначале к первому накопителю, а затем ко второму. (Такое подключение кабеля от одного устройства к другому, а затем еще к одному называется гирляндным соединением.)

Второй кабель имеет 20 проводников, большинство из которых соединено с землей и служат защитой для других. Все биты данных, которые контроллер записывает в накопитель, передаются всего по одному из проводников, а по другому передается дополняющий сигнал (дополняющий означает, что при передаче по первому проводнику двоичного нуля по второму передается двоичная единица и наоборот). Еще по одной паре проводников передается считанный сигнал и его дополнение от головок в контроллер. Это высокочастотные сигналы, передающие до 5 млн. бит/с, и они более подвержены помехам, чем сигналы в 34-проводном кабеле. Чтобы снизить уровень помех, каждый накопитель имеет свой 20-проводный кабель и он должен быть максимально коротким.

В сущности, по 20-проводному кабелю может передаваться всего один бит данных. При скорости в 5 млн. бит/с скорость в байтах составляет одну восьмую часть, т.е. 625 Кбайт/с. В накопителях SCSI данные передаются по 8 бит, а иногда по 32 бита одновременно, поэтому скорость передачи данных по параллельному кабелю гораздо выше. (Недавно появились накопители SCSI, в которых применяется последовательный кабель, в котором данные передаются битами, т.е. так, как в интерфейсе ST412/ST506.)

Название ST412/ST506 не подразумевает какого-то конкретного способа кодирования данных, хотя ранее оно обычно обозначало накопитель MFM с 17 секторами на дорожке. Поскольку кодирование данных по способу 2,7 RLL дает одинаковую максимальную частоту переходов магнитного поля (но с меньшей минимальной частотой), сигналы в кабеле между контроллером RLL и накопителем очень похожи на сигналы комбинации контроллера MFM и накопителя. Поэтому можно сказать, что накопитель RLL также имеет интерфейс ST412/ST506.



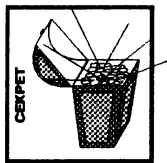
RLL-кодирование помещает больше данных на меньшее расстояние по дорожке, поэтому накопитель RLL обычно имеет 25 или 26 секторов, а не 17 секторов типичного накопителя MFM. Далее, так как в RLL-накопителе в данном числе переходов поля закодировано больше битов, даже без повышения частоты сигналов по сравнению с MFM-накопителем, RLL-накопитель может передавать по кабелю на 50% больше бит/с (70 Кбайт/с в случае 26 секторов на дорожке и иногда 937 Кбайт/с при наличии на дорожке 31 сектора).

Более сложные контроллеры ARLL иногда описываются так, как имеющие этот интерфейс. Но поскольку в них частота сигналов от головки и в нее выше спецификации частоты стандарта, их интерфейс следовало бы назвать по-другому.

В 1983 г. схему интерфейса ST412/ST506 несложно было реализовать на доступных в то время аппаратных средствах. Он удовлетворял запросы владельцев персональных компьютеров, когда емкость накопителя составляла всего 10 Мбайт, а шина компьютера работала на частоте 4,77 МГц. Однако вскоре эта ситуация изменилась. Появились накопители большей емкости, а новые компьютеры стали работать на большей частоте. Вскоре ограничения интерфейса стали тормозом развития, поэтому производители жестких дисков быстро предложили два разных решения этой проблемы.

3.2.2. Интерфейс ESDI

Фирма Maxtor начала разрабатывать стандарт ESDI (Enhanced Small Device Interface — улучшенный интерфейс малых устройств) в начале 80-х годов. Этот стандарт родился на базе интерфейса ST412/ST506, в котором проведены небольшие, но существенные изменения. Этот стандарт по-прежнему работает с платой контроллера диска, которая вставлена в шину компьютера и соединена с накопителем теми же двумя плоскими кабелями; один из них имеет 34 проводника (и идет от контроллера к первому накопителю, а затем ко второму), а второй содержит 20 проводников (для двух накопителей требуются два таких кабеля).



Самое важное отличие стандарта ESDI от интерфейса ST412/ST506 состоит в том, что схема сепаратора синхронизации и данных находится в самом накопителе, а не в контроллере. (Схема сепаратора разделяет поступающие от накопителя сигналы на сигналы, содержащие информацию о синхронизации, и сигналы, несущие собственно считываемые данные. Подробнее в главе 2.)

Фирмы, производящие накопители, обычно не выпускают контроллеры. Поэтому в интерфейсе ST412/ST506 MFM или RLL должны были включать сепараторы синхронизации и данных, которые превратились в отдельную схему и могли работать с любым накопителем с интерфейсом ST412/ST506. Теперь же в интерфейсе ESDI производители накопителей смогли встроить такой сепаратор данных, который тщательно согласован со схемами накопителя.

Еще одно достоинство такого решения связано с тем, что данные извлекаются из сигналов от головок до подачи в кабель, поэтому скорость передачи можно повысить. Максимальная частота передачи бит в интерфейсе ESDI составляет 24 МГц

(или 24 млн. бит/с). Современные накопители достигли частоты 15 МГц. Предлагаемые расширения стандарта ESDI повышают частоту примерно до 50 МГц.

Не забывайте, однако, что в этом стандарте данные передаются по кабелю битами, поэтому скорость в байтах составляет одну восьмую приведенных значений. Поэтому современные накопители с интерфейсом ESDI обеспечивают скорость передачи данных около 2 млн. байт/с.

Среди менее значительных отличий стандарта ESDI от предшественника отметим новые команды, которые может посылать контроллер, и новые сообщения о состоянии, которые накопитель передает контроллеру. Например, контроллер может приказывать накопителю перейти на определенную дорожку, указывая номер дорожки (вместо нескольких команд шагов внутрь или наружу). Кроме того, контроллер может указать накопителю выполнить некоторые диагностические тесты и сообщить их результаты.

3.2.3. Интерфейс SCSI

Интерфейс SCSI (Small Computer System Interface — интерфейс малых компьютеров) — это очень своеобразный дисковый интерфейс. Правильно говоря, SCSI даже не является дисковым интерфейсом.

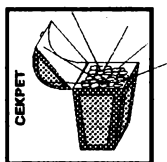
Стандарт SCSI появился в конце 1979 г. под названием SASI (Shugart Associates System Interface). За полтора десятка лет производители накопителей несколько раз модернизировали его. Современные накопители и контроллеры подчиняются версии стандарта под названием SCSI-2. Комитет, разрабатывающий этот стандарт, сейчас работает над стандартом SCSI-3.

Стандарт SCSI отличается от рассмотренных некоторыми важными моментами. Самая важная идея заключается в том, что в этом интерфейсе предполагается подключение к кабелю (шина SCSI) только "интеллектуальных" устройств. К кабелю одного SCSI-ведущего можно подключить до семи SCSI-ведомых устройств.

Примечание. Поясним смысл терминов "ведущий" и "ведомый". Ведущее устройство ("хозяин") сообщает ведомому устройству ("рабу"), что делать, и ведомое устройство выполняет заданное действие. Может быть только один ведущий, так как кто-то должен решать, что делать дальше. Ведущий может приказывать ведомому взять на себя полное управление шиной, но, в конце концов, ведомый должен вернуть управление ведущему.

Природа SCSI-устройств может быть разнообразной. К шине можно подключать быстрые жесткие диски большой емкости, ленточные накопители, оптические диски, лазерные принтеры и даже медленные устройства, например мышь, если каждое устройство имеет соответствующую интерфейсную схему между ним и кабелем. Во многих отношениях SCSI больше похож на небольшую локальную сеть, чем на обычный интерфейс персонального компьютера и жесткого диска.

Подключение SCSI-устройств к персональному компьютеру. Для подключения шины SCSI к персональному компьютеру необходимо иметь главный (хост) адаптер интерфейса SCSI. Адаптер — это схемная плата, которая вставляется в шину ввода-вывода компьютера и к которой подключается кабель шины SCSI.



Большинство устройств SCSI взаимодействуют по 50-контактному плоскому кабелю (или в соответствии со стандартом SCSI-2 по 50-контактному кабелю и вспомогательному 68-контактному кабелю). Все сигналы — цифровые. Данные передаются параллельно по 8 (16 или 32) бит. Кабель имеет относительно мало управляющих сигналов.

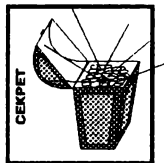
Каждое из семи SCSI-ведомых устройств может иметь внутри до восьми логических подустройств. Таким образом, одним главным адаптером SCSI можно взаимодействовать с 56 логическими отдельными подустройствами.

Скорость передачи данных первого интерфейса SASI составляла всего 12 Мбит/с или 1,5 Мбайт/с по восьми параллельным проводникам. В новейшей версии стандарта SCSI (она часто называется SCSI-2) скорость битов доходит до 10 МГц с соответствующей скоростью данных до 40 Мбайт/с (по 32 параллельным проводникам).

Главный адаптер SCSI посылает команду в одно из ведомых устройств, обладающее достаточным "интеллектом" для выполнения последовательности операций с использованием кабеля SCSI (другими словами, ведомый временно становится ведущим). Когда ведомый заканчивает свою задачу, он освобождает кабель и главный адаптер SCSI может разрешить другому ведомому временно управлять кабелем.

Фактически накопитель SCSI состоит из трех частей. Первая, конечно, — это сам дисковый накопитель. Вторая часть — это электронный модуль, который в ранних версиях вставлялся в контроллер диска. И, наконец, должен быть специальный интерфейс ведомого SCSI. Именно в этой части сосредоточена электроника, которая взаимодействует с шиной SCSI.

Посту популярности интерфейса SCSI способствовали компьютеры Macintosh фирмы Apple. (Все компьютеры Macintosh, кроме самых ранних, поставляются со встроенным портом SCSI.) До недавнего времени компьютеры PC оказывали слабую поддержку интерфейсу SCSI. Однако, после того как фирма IBM выпустила главный адаптер SCSI и некоторые периферийные устройства с интерфейсом SCSI, ситуация стала быстро меняться.



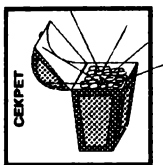
Первоначально интерфейс SCSI был рассчитан на широкий плоский кабель и параллельную передачу минимум 8 бит данных. В версию SCSI-2 встроена поддержка еще более широких кабелей с передачей до 32 бит одновременно. В некотором смысле версия SCSI-3 изменяет тенденцию. Говоря точнее, SCSI-3 поддерживает и широкий и узкий варианты. Кроме параллельного широкого кабеля SCSI-3 поддерживает последовательный кабель с передачей отдельными битами.

Такая возможность, на первый взгляд, приводит к отступлению, но это не так. Нужно понять огромную мощь системы команд SCSI. Последовательная версия SCSI-3 включена для того, чтобы использовать все новые возможности (кроме параллельной передачи данных) интерфейса SCSI в очень небольших компьютерах, например laptop, в которых просто нет места для больших разъемов параллельного интерфейса SCSI.

Беспокойство в мире SCSI. Аппаратный интерфейс для SCSI определен очень строго. К сожалению, программная поддержка SCSI на рынке ПК (в основном, это драйверы устройств, содержащиеся в файлах устанавливаемых драйверов устройств и в ПЗУ главного адаптера SCSI) далека от стандартизации. Есть три конкурирующих подхода, в которых заинтересованы различные производители.

Фирма Microsoft предлагает LADDR (Layered Device Driver Architecture — Слоистая архитектура драйвера устройства). Группа фирм, возглавляемая Adaptec, поддерживает ASPI (Advanced SCSI Programming Interface — Улучшенный программный интерфейс SCSI). Наконец, Американский национальный институт стандартов разработал еще один стандарт — CAM (Common Access Method — Общий метод доступа).

Не все поставщики главных адаптеров SCSI предлагают драйверы устройств для всех трех вариантов. Большинство производителей обычно поддерживают только один. Программы, которые обращаются к устройствам SCSI, разрабатываются для одного из трех методов доступа. В результате в вашем ПК может оказаться до трех различных главных адаптеров SCSI каждый со своим драйвером устройства (по одному для LADDR, ASPI и CAM).



Проблема заключается в том, что данный главный адаптер SCSI должен управляться и реагировать на один уровень запроса прерываний (IRQ). Следовательно, для поддержки нескольких устройств SCSI (причем одни из них взаимодействуют в соответствии с каждым из трех стандартов) потребуются три главных адаптера с тремя взаимодействующими с ними драйверами устройств (для этого потребуются до трех уровней IRQ).

В идеальном случае вам нужен только один главный адаптер SCSI для любого количества устройств SCSI, но это невозможно при произвольной смеси типов устройств. Если все ваши устройства подчиняются одному из трех стандартов доступа (и вы имеете не более семи физических и не более 56 логических устройств), вы уже достигли идеала. В противном случае вам придется иметь в ПК несколько главных адаптеров SCSI.

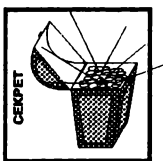
Конечно, в одном ПК можно иметь и больше трех главных адаптеров SCSI; вопрос упирается только в число свободных разъемов. В некоторых случаях можно иметь несколько главных адаптеров SCSI, использующих один и тот же метод доступа (например, ASPI), и один драйвер устройства. (Один из пользователей сообщил, что он применяет четыре главных адаптера SCSI и драйвер ASPI для управления 28 ленточными накопителями одновременно. С помощью номеров логических устройств каждого ведомого интерфейса SCSI в такой конфигурации можно управлять 224 устройствами.)

До недавнего времени при подключении к ПК одного или двух высокопроизводительных накопителей проще всего было применить контроллер и накопитель интерфейса ESDI. Однако этот способ совсем не помогает при подключении других периферийных устройств, например CDROM или ленточного накопителя.

Теперь при программной поддержке SCSI интерес к интерфейсу ESDI снижается. Почти во всех разработках новых высокопроизводительных дисковых накопителей большой емкости применяется SCSI или IDE.

ИЛЛЮЗИЯ СОВЕРШЕННОГО НАКОПИТЕЛЯ. По существу, жесткий диск с интерфейсом SCSI имеет свою электронику контроллера диска. Главный адаптер SCSI просто управляет шиной SCSI, которая действует как разновидность локальной мини-сети. Важное следствие такого многослойного управления накопителем состоит в том, что многие физические детали накопителя скрываются от компьютера.

Теперь ни компьютер, ни контроллер диска, вставленный в шину компьютера, не командуют накопителю перейти на одну дорожку внутрь или наружу. Накопитель SCSI выглядит для компьютера как идеализированная большая группа логических блоков, в которых могут храниться данные. Главный адаптер SCSI адресует каждый из этих блоков по адресу логического блока (Logical Block Address — LBA). Чтобы ПК рассматривал накопитель SCSI как и любой другой жесткий диск, главный адаптер представляет накопитель SCSI компьютеру как обычный трехмерный диск. Но главный адаптер создает количество головок, цилиндров и секторов на дорожке для такого фиктивного диска наиболее удобным для себя образом.



Накопитель SCSI большой емкости может фактически иметь пять дисков с десятью головками, из которых одна или две головки применяются для позиционирования блока головок. Накопитель имеет несколько тысяч головок. Внутренние дорожки имеют более 30 секторов, внешние дорожки — в два раза больше, а часть средних дорожек может иметь какое-то промежуточное количество секторов.

Интерфейс ведомого SCSI превращает все это в простой одномерный массив запоминающих блоков. В свою очередь, главный адаптер SCSI создает из линейного массива логических блоков иллюзию фиктивного, или логического, накопителя, имеющего 64 головки, 32 сектора на дорожке и всего несколько сотен цилиндров. Именно с таким накопителем работает BIOS персонального компьютера.

Накопители SCSI всегда выглядят совершенными. В таком накопителе при выполнении программы CHKDSK никогда не видны плохие секторы. Причина этого скрыта в интеллектуальном интерфейсе SCSI-ведомого.

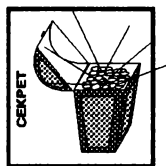
При начальной подготовке накопитель SCSI проверяет каждую ячейку, в которой он может хранить данные, и отвергает ячейки с дефектами. Остальным ячейкам назначаются адреса логических блоков. А если дефект появится позже?

Многие накопители SCSI контролируют работу диска, и если какой-то сектор начинает портиться, накопитель перемещает содержащиеся в нем данные в ближайший запасной сектор не извещая об этом пользователя (точнее, главный адаптер SCSI). Про такие накопители говорят, что они поддерживают "горячий ремонт" (hot fixes).

Рассмотренный процесс служит примером преобразований головки и сектора, которые подробнее рассматриваются в главе 7. Так как накопители SCSI представляются в некотором смысле "ирреальными", они значительно ограничивают функции таких программ, как SpinRite.

3.2.4. Интерфейс IDE

Сейчас наиболее популярным интерфейсом жестких дисков для ПК является так называемая интегрированная электроника накопителя (Integrated Drive Electronics — IDE). Этот интерфейс популяризировала фирма Conner Peripherals; здесь RLL-или MFМ-контроллер находится в самом накопителе, что позволяет исключить плату контроллера. Для подключения накопителя IDE к компьютеру не требуется разъем шины, что больше всего привлекает в этом интерфейсе. Подключение осуществляется через специальный разъем на материнской плате, служащий мини-разъемом, к которому подведены только нужные для интерфейса IDE сигналы.



Формальное название такого варианта соединения — AT Attachment (подключение AT) или ATA. Формальный стандарт ATA принят компьютерной индустрией. Стандарт опирается на 40-контактный разъем особой конструкции, сигналы на который берутся со стандартной (ISA) шины ввода-вывода.

Многие жесткие диски, в которых электроника контроллера встроена в накопитель, в общем, являются разновидностями IDE-накопителя, но их чаще называют по-другому. Во всех накопителях SCSI встроенная электроника. В некоторых моделях PS/2 фирма IBM поставляет накопитель ESDI, контроллер которого смонтирован в накопителе. Однако они не относятся к накопителям IDE/ATA, так как в них не применяется 40-контактный разъем, определенный в спецификации ATA. Применение в таких случаях обозначений SCSI или ESDI полезнее, так как они сообщают кое-что о возможностях накопителя, а не только о его физической конструкции.

Далее в книге обозначение IDE применяется как сокращенное название жестких дисков, которые имеют IDE с интерфейсом ATA (см. приведенный выше "Технический секрет").

Можно подключить накопитель IDE к ПК, не имеющему разъема по спецификации ATA на материнской плате. Для этого применяется так называемая переходная плата IDE. Эта плата вставляется в системную шину ввода-вывода и имеет разъем ATA. Поскольку многие контроллеры жесткого диска AT других типов имеют и контроллер гибкого диска, такой контроллер может быть и на переходной плате.

Поскольку как накопители IDE, так и SCSI, содержат в накопителе весь контроллер, они реализуют многие нестандартные приемы, например преобразование секторов. Однако, большинство накопителей IDE могут иметь любые параметры, которые вы им назначите. Другими словами, накопители IDE могут иметь любые числа головок, цилиндров и секторов на дорожке, которые ожидает ПК, при условии, что описываемая этими параметрами емкость не превышает физической емкости накопителя.

Такая возможность объясняется тем, что на нижнем уровне накопитель IDE работает аналогично SCSI. Его интерфейс создает единый линейный набор блоков, адресуемых их логическими адресами, а электроника преобразует этот набор в трехмерную адресацию, которую ожидает компьютер.

Жесткие диски с интерфейсами IDE и SCSI имеют так много общего, что вы, возможно, удивитесь наличию различий между ними. Прежде всего, они отличаются интерфейсами. Нельзя подключить накопитель SCSI в разъем ATA, а накопитель IDE к шине SCSI.

Во многих случаях это единственное существенное отличие. Часто можно приобрести одну и ту же модель накопителя в варианте SCSI или IDE. Но все же это разные накопители. В общем, промышленность предлагает накопители меньшей емкости в формате IDE и большей — в формате SCSI (но, разумеется, эти диапазоны перекрываются). Накопители большей емкости, т.е. в основном SCSI, обладают большим встроенным "интеллектом". Это означает, что в большинстве накопителей SCSI поддерживается "горячий ремонт", что допустимо лишь в некоторых накопителях IDE.

Каким образом конструкция накопителей SCSI и IDE влияет на программы, работающие на нижнем уровне, например SpinRite и другие программы изменения коэффициента чередования? Если диск выглядит для компьютера как накопитель MFM или RLL, программа SpinRite часто может считать его именно таким. Но если, как это обычно бывает, диск маскирует свои реальные параметры, он препятствует SpinRite и другим программам реализовать все их функции. Такие программы могут все же обеспечить некоторые полезные возможности, хотя и не в таком объеме, как для накопителей MFM, RLL и ESDI.

3.2.5. Жесткие диски на съемной плате

За несколько лет до появления накопителей IDE фирма Plus Development Corporation (сейчас Quantum Corporation) отошла от обычной конструкции, но в другом направлении. Фирма захотела объединить накопитель и его электронику, но не размещая электронику в обычной конструкции накопителя, она "сдвинула" жесткий диск на плату контроллера. Первая плата Hardcard появилась в 1985 г., а затем несколько других фирмы выпустили аналогичные изделия. Модели фирмы Quantum от первоначальной емкости 10 Мбайт дошли до нескольких сотен Мбайт, по-прежнему занимая всего один слот в ПК.

Hardcard является зарегистрированной торговой маркой фирмы Quantum, но обычно это название относится к аналогичным изделиям всех производителей. В этой книге термины "жесткий диск на съемной плате" и "плата жесткого диска" служат общим названием для подобных изделий.

Производители плат жестких дисков часто используют преимущества объединения контроллера и накопителя, как и производители SCSI и IDE накопителей, применяя зонную плотность (подробнее см. главу 2), преобразование секторов и другие нестандартные приемы достижения максимальной емкости в минимальном пространстве. Платы жесткого диска для компьютера выглядят как более или менее стандартные накопители MFM, но они отвергают нижние уровни управления компьютера, которые характерны для накопителей MFM. По-прежнему, программы типа SpinRite могут реализовать для платы жесткого диска только часть своих функций.

На рис. 3.5 в наглядной форме представлена сводка рассмотренных ранее наиболее популярных подсистем жестких дисков ПК.

3.2.6. Интерфейс PCMCIA

Одна из новейших тенденций в интерфейсах жестких дисков состоит в монтаже накопителя на плате PCMCIA (PC Memory Card International Association — международная ассоциация плат памяти ПК). Эта ассоциация создана для разработки и внедрения стандартных средств взаимодействия съемных модулей и других электронных систем. Вначале к этим системам относились карманные и портативные компью-

теры и ноутбуки, а также некоторые миди-машины. Сейчас же слоты РСМСІА появились в настольных компьютерах и множестве других систем.

Стандарт РСМСІА определяет три размера платы РСМСІА. По размерам все они напоминают толстую кредитную карточку, а различия Типов 1, 2 и 3 касаются только толщины платы. Во всех платах применяется один и тот же 68-контактный разъем.

Одно из преимуществ плат РСМСІА состоит в том, что их в любое время можно вставить в слот ПК или вынуть оттуда. При работе на компьютере вы можете сменить такие платы так же легко, как и гибкие диски.

Вначале такие платы применялись только для микросхем ЗУПВ и ПЗУ. Однако спецификация интерфейса РСМСІА, по существу, является надмножеством интерфейса АТА, поэтому на плате РСМСІА можно разместить практически любое периферийное устройство, рассчитанное на съемные платы для шины АТ.

Стандарт РСМСІА разработали фирмы — производители микросхем памяти. Они предусмотрели настолько хороший периферийный интерфейс, что его стали применять и для других изделий. Были предложены минимум три варианта стандарта РСМСІА для размещения на платах накопителей ІДЕ/АТА.

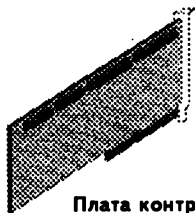
Низший уровень совместимости достигается, когда платы удовлетворяют Типу 2 РСМСІА, используют 68-контактный разъем РСМСІА и электрические сигналы на нем совместимы с сигналами РСМСІА. Однако смысл этих сигналов не аналогичен настоящей спецификации РСМСІА.

Уровень совместимости, достигнутый фирмой Hewlett-Packard для 1,3" жестких дисков Kittyhawk, позволяет производителю ПК ввести стандартный промышленный слот (РСМСІА) и подключить к нему накопитель ІДЕ/АТА. Это был первый случай применения слотов РСМСІА с жесткими дисками. Следом за ним появилась вторая модификация стандарта РСМСІА.

Эта модификация обеспечивала полную совместимость АТА с разъемом РСМСІА. Когда удовлетворяющий такому стандарту слот имеется в ПК, в него можно вставить плату памяти РСМСІА, плату модема РСМСІА или накопитель ІДЕ/АТА на плате РСМСІА, и все они будут правильно работать. Но все же платы в этом стандарте не поддерживают всех возможностей стандарта РСМСІА. До обращения к накопителю необходимо загружать драйвер устройства, который рассчитан на работу с жестким диском. Такой вариант спецификации разработан фирмами Western Digital и Hewlett-Packard.

Последний вариант представляет собой полную реализацию спецификации АТА для жестких дисков на платах РСМСІА с полной поддержкой спецификации РСМСІА (поставка 2.0). Когда жесткие диски удовлетворяют этой спецификации, их можно использовать попеременно с платами памяти как сменные носители в ПК, имеющих обычные гнезда РСМСІА. Этот вариант разработан комитетом РСМСІА.

Все три стандарта находятся на стадии становления и к моменту подготовки книги окончательно не приняты. Несмотря на это, уже появляются первые накопители, удовлетворяющие разным стандартам, в виде съемных плат для персональных компьютеров.



Плата контроллера
жесткого (и гибкого) диска

Эти накопители обычно подключаются к контроллеру 20-жильным кабелем данных (по одному для каждого накопителя) и 34-жильным кабелем управления (один для обоих накопителей 0 и 1). Питание подается от блока питания по отдельному 4-жильному кабелю



Жесткий диск

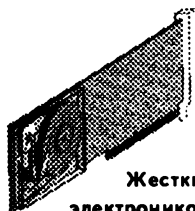
ST412/ST506 (MFM и RLL) и ESDI

Накопители IDE/ATA могут подключаться к специальным разъемам ATA, встроенным в системные блоки некоторых ПК. Эти 40-контактные разъемы просто расширяют шину системного блока для использования в устройстве ATA. В других ПК накопитель IDE подключается через переходную плату, которая вставляется в системную шину. Такая шина имеет разъем ATA (и, возможно, контроллер гибкого диска)



Жесткий диск с
электроникой
контроллера

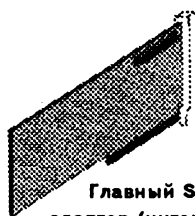
Интерфейс интегрированной электроники накопителя с подключением ATA (IDE/ATA)



Жесткий диск с
электроникой контроллера

Жесткий диск на съемной плате (например, Hardcard фирмы Plus Development) содержит на одной плате электронику контроллера и подвижные части. Питание подается от блока питания через слот шины, в который плата вставляется в системном блоке

Плата жесткого диска



Главный SCSI
адаптер (интерфейс с
шиной SCSI)

Главный SCSI-адаптер служит мостом между двумя шинами: шиной ввода-вывода системного блока ПК и шиной SCSI. Второй шиной часто служит 50-жильный кабель, который можно подключить к 7 периферийным устройствам (и к 56 логическим устройствам)



Жесткий диск,
контроллер и
интерфейс
SCSI-ведомого

Интерфейс малых компьютеров (SCSI)

Рис. 3.5. Популярные типы подсистем жесткого диска для ПК

3.3. Два вида архитектуры накопителя

До сих пор мы говорили об архитектуре накопителей на жестких дисках в терминах их физических параметров (число головок, число цилиндров и число секторов на дорожке). Но физический аспект не является единственно полезным при рассмотрении жестких дисков. Фактически в DOS этот способ не применяется.

3.3.1. Головки, цилиндры и секторы

В главе 2 было показано, что в накопителе есть диски и головки. Головки (по одной для каждой поверхности) объединены в блок головок и перемещаются внутри или снаружи дисков с помощью привода. Для позиции привода и блока головок каждая головка считывает или записывает на одной дорожке поверхности диска. Совокупность дорожек для одной позиции блока головок называется цилиндром (см. рис. 2.3).

Цилиндры нумеруются с нуля от внешнего края дисков. Головки нумеруются также с нуля, начиная с верхней поверхности. В некоторых накопителях одна из головок используется в сервосистеме позиционирования, поэтому для хранения данных остается нечетное число головок.

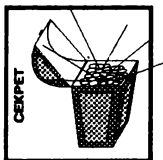
Дорожки разделены на секторы, содержащие 512 байт данных (плюс заголовок сектора и информация ECC). Секторы нумеруются с единицы.

Таким образом, для сохранения информации в определенном месте диска необходимо сообщить контроллеру накопителя, где ее поместить на конкретной поверхности, в определенном цилиндре и секторе с некоторым номером. Именно эти операции реализует BIOS, но DOS этого не делает, так как она смотрит на диски гораздо проще. Ее представление дисков называется логическим.

3.3.2. Логические структуры информации

Выше показана физическая адресация данных на диске, которой должны подчиняться аппаратные средства. Программисты и пользователи жестких дисков редко хотят связываться с такой точкой зрения, поскольку логическое представление дисков намного проще и удобнее.

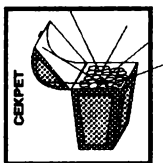
MS DOS спроектирована так, чтобы освободить нас (и наши программы) от знания деталей дисковых накопителей. Вместо этого DOS пытается управлять этим ресурсом так, чтобы мы и наши программы давали ей простейшие команды, например сохранить или считать информацию.



Операционная система в ПК состоит из нескольких частей: ROM BIOS (микросхемы Basic Input/Output System — базовой системы ввода-вывода) на материнской плате и различных съемных платах, которые содержат программы низкого уровня управления аппаратными средствами; программа BIOS, которая находится в одном из двух скрытых системных файлов на загрузочных дисках; программа DOS высокого уровня, которая содержится во втором скрытом системном файле. Программы BIOS, находящиеся в ПЗУ и на диске, работают с аппаратными средствами таким способом, который наиболее близок к реальности. Программа DOS стремится отойти от аппаратных средств к более простой логической форме.

В главе 9 будут показаны те части, на которые подразделяется дисковое пространство при подготовке диска для работы с DOS. Сейчас достаточно сказать только, что есть область диска до области хранения DOS и может быть другая область после нее. Пока мы будем игнорировать эти не-DOS области.

На своем высшем уровне DOS считает ту часть диска, где хранятся данные, просто набором логических секторов и нумерует их как простейший список, начиная с нулевого логического сектора.



Способ нумерации логических секторов DOS не следует путать с нумерацией физических секторов, которые начинаются на каждой дорожке с первого сектора.

Не путайте также эту нумерацию логических секторов с (очень похожими) логическими адресами блоков, применяемых в накопителях IDE и SCSI. Номер логического сектора DOS и логический ад-

рес блока используют одномерный способ указания области хранения. Они различаются тем, что номера логических адресов блоков идут от самого начала диска, а номера логических секторов DOS — от начала каждого доступного DOS раздела или логического накопителя на диске.

Для каждой программы, работающей под DOS, упрощаются дисковые операции. Даже многие дисковые программы-утилиты имеют дело с абстрактным логическим накопителем, а не с реальным физическим. Именно такие свойства DOS позволяют программам работать с множеством типов накопителей на жестких дисках. DOS скрывает их разнообразие как от нас, так и от наших программ.

Для части своих функций программа SpinRite работает с диском на абстрактном уровне, но для полного использования всех ее потенциальных возможностей SpinRite должна обходить DOS и обращаться прямо к контроллеру. Если же, реальность скрывает не DOS, а сам контроллер, то даже SpinRite не может проникнуть сквозь такое покрывало, что ограничивает ее возможности при решении некоторых проблем.

Заключение

- Скорость передачи данных — это важнейший показатель быстродействия жесткого диска. Задание оптимального коэффициента чередования секторов позволяет значительно повысить этот показатель
- Установка слишком малого коэффициента чередования секторов намного снижает фактическое быстродействие накопителя
- Производители диска или контроллера не могут правильно задать коэффициент чередования. Только специалист, объединяющий накопитель и контроллер, может установить правильное значение (если в контроллере нет буфера дорожки)
- Перекос головки и цилиндра также помогают оптимизировать скорость накопителя. Пока такая тонкая настройка достигается не для всех накопителей, но при повышении скорости дисков она будет применяться все шире
- В персональных компьютерах применяются шесть популярных дисковых подсистем: MFM и RLL (с интерфейсом ST412/ST506), ESDI, SCSI, платы жестких дисков и накопители IDE/ATA. Сейчас наиболее распространены накопители SCSI (большой емкости) и IDE (меньшей емкости)
- Монтаж накопителя на плате PCMCIA позволяет применять сменные накопители практически в любых персональных компьютерах.
- Местоположение блока информации в накопителе можно определить четырьмя способами: физическим адресом (поверхность, цилиндр и номер физического сектора), логическим адресом блока, фиктивным физическим адресом или логическим номером сектора. Накопитель и контроллер должны использовать первый способ. Накопители IDE и SCSI могут воспринимать местоположение от BIOS, задаваемое вторым и третьим способами. DOS применяет для определения областей диска только последний способ.

В этой и предыдущей главах рассмотрены наиболее важные физические и инженерные вопросы конструкции жестких дисков. Следующая глава будет посвящена еще более глубокому обсуждению накопителей на жестких дисках, которое ранее было доступно только в специальной технической литературе и профессиональных журналах.

Глава 4. Как устроены накопители на жестких дисках

В этой главе вы узнаете

- Различия между шаговым двигателем и приводом со звуковой катушкой
 - Каким образом обратная связь улучшает работу механизма позиционирования головок
 - Два вида информации, хранимой на жестком диске, и две системы, которые используют эту информацию, помогая компьютеру искать и восстанавливать хранимые на диске данные
 - Как устроены головки считывания-записи
 - О сходствах и различиях продольной и вертикальной записи
 - Функции электроники, смонтированной на накопителе и находящейся в контроллере жесткого диска
 - Что означает рекалибровка и как она применяется при частом включении накопителя.
-

4.1. Физические элементы накопителя

Диски вращаются, блок головок перемещается взад и вперед, а сами головки считывают и записывают данные на поверхности дисков. Эти общие вопросы конструкции дискового накопителя рассмотрены в главе 2.

Главный двигатель накопителя вращает диски с постоянной скоростью, обычно 3600 обор./мин (или 60 обор./с). Такую скорость вращения шпинделя обеспечивает специальный многополюсный двигатель постоянного тока. Вращающий момент двигателя почти идеально постоянен, поэтому диски вращаются с постоянной скоростью.

Вопрос о перемещении головок несколько сложнее, но он довольно интересен. Принцип действия головок также требует специального рассмотрения.

Для перемещения блока головок разработано много конструкций привода, но их можно классифицировать двумя основными типами: позиционеры со счислением пути рейкой и сервосистемы позиционирования с обратной связью. В приводах первого типа применяются шаговые двигатели, а второго — звуковая катушка. Поскольку системы с шаговым двигателем дешевле, медленнее и менее точны по сравнению с приводами на базе звуковой катушки, позиционеры с шаговыми двигателями применяются в дешевых накопителях сравнительно небольшой емкости.

4.1.1. Привод с шаговым двигателем

Все электрические двигатели работают по принципу притяжения и отталкивания двух магнитов. Один из магнитов укреплен на станине или статоре двигателя (неподвижной части), а второй монтируется на подвижной части (рис. 4.1). В зависимости от конструкции двигателя один из двух магнитов постоянный, но один как минимум должен быть электромагнитом, чтобы его можно было включать и выключать в точно определенные моменты времени.

Когда электрический ток проходит через обмотку возбуждения, в зазоре образуется магнитное поле. Оно заставляет ротор ориентироваться в соответствии с направлением поля. Когда направление тока в обмотке возбуждения меняется на противоположное, направление поля в зазоре также изменяется. Такое изменение заставляет

ротор ориентироваться противоположно. Если, когда ротор достигнет новой ориентации, вновь изменить направление тока, ротор будет продолжать вращение к первоначальной ориентации. Изменяя направление тока в нужные моменты времени, можно заставить ротор вращаться непрерывно, даже преодолевая некоторое трение. Можно считать, что действие двигателя напоминает раскачивание ребенка на качелях, когда вы подталкиваете качели в определенной точке.

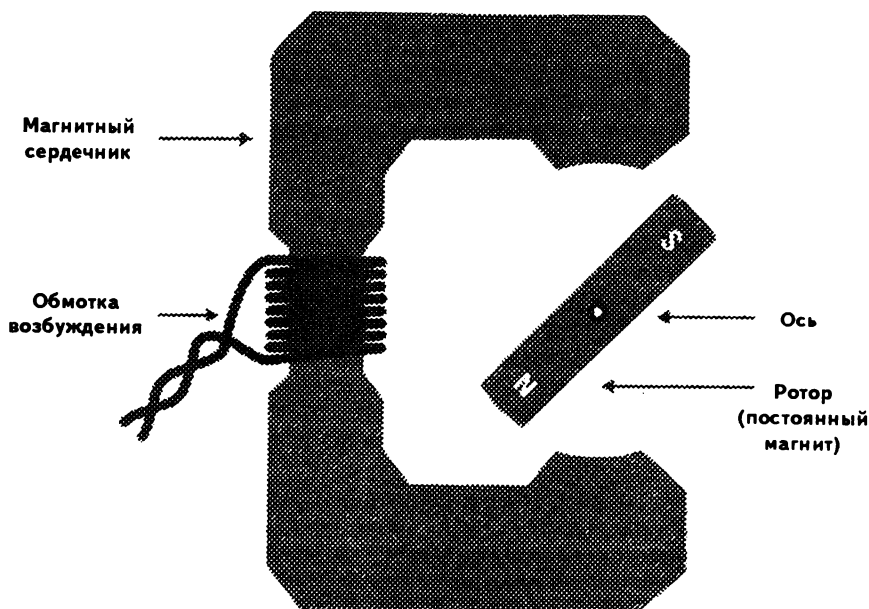
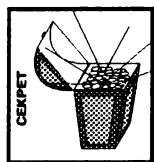


Рис. 4.1. Простой шаговый двигатель



Для плавного вращения на статоре двигателей укрепляются несколько магнитов; эти магниты "толкают" тот магнит, что смонтирован на оси. Для получения такого "хорошего" двигателя конструктор пытается слить толчки в одно максимально плавное усилие. Это особенно важно для шпиндельного двигателя жесткого диска, в котором постоянная скорость вращения оказывается важнейшим фактором.

Можно достичь эффекта действия нескольких магнитов, монтируя на статоре всего два магнита. Для этого требуются специальные конструкции магнитов и способ их размещения на статоре. Одна из возможных конструкций показана на рис. 4.2,а, а на рис. 4.2,б представлена форма двух фаз токов, которые заставляют двигатель вращаться.

На рис. 4.2 показан многополюсный электродвигатель. В этом двигателе две обмотки возбуждения намотаны на сердечнике, имеющем четыре полюса в том месте, где сердечник охватывает ротор. При вращении ротор воспринимает чередующиеся полюса двух обмоток. Такой двигатель поворачивается, когда ток вначале включает-ся в одной обмотке, а затем — в другой. После этого направление тока изменяется в первой обмотке, а затем во второй. Такая форма тока называется двухфазным возбуждением и показана на рис. 4.2,б.

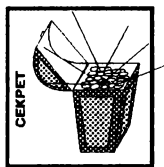
Рассмотрим, что происходит на каждом из двух этапов. На первом этапе магнит ротора находится вблизи одного из полюсов первого магнита (фаза А). На следующем этапе ротор воспринимает притяжение магнита фазы А и магнита фазы В, по-

этому ротор поворачивается в позицию между двумя соседними полюсами (именно эта позиция представлена на рис. 4.2,а). На следующем этапе поле от полюсов фазы А изменяется, а это вызывает отталкивание ротора от того полюса, к которому он вначале притягивался. Когда магнит проходит полюс фазы В, он приближается к следующему полюсу фазы А, т.е. он опять оказывается в точке между полюсами (в нашем примере это соответствует повороту на 45° по часовой стрелке). На последующих этапах двухфазного тока в обмотках возбуждения ротор должен поворачиваться на фиксированный угол при каждом изменении тока в любой обмотке.

Повторение чередующихся изменений токов в двух обмотках возбуждения вызывает непрерывное вращение многополюсного двигателя. Но есть и другой способ применения такого двигателя: просто включить ток в одной или двух обмотках и оставить его включенным. В этом случае двигатель делает часть оборота и затем останавливается, его ротор "блокируется" либо прямо перед одним из полюсов (если ток подается только в одну обмотку), либо посередине между двумя полюсами (при наличии тока в обеих обмотках). Если ток в любой обмотке изменяет направление или изменяется от выключенного к включенному (или наоборот), а затем токи опять "замирают", ось двигателя повернется в другую позицию и остановится. О применении такого двигателя в накопителях на жестких дисках сказано далее.

Плохо сконструированные двигатели вращаются толчками, т.е. они стремятся "прыгнуть" из одной позиции в другую, а при достижении ее пытаются остановиться (см. технический секрет, приведенный выше). Такая ситуация считается дефектом обычного двигателя, но для шагового двигателя именно это и нужно.

Шаговым двигателем управляют простые сигналы постоянного тока, которые заставляют ось последовательно осуществлять переход из одной позиции в другую.



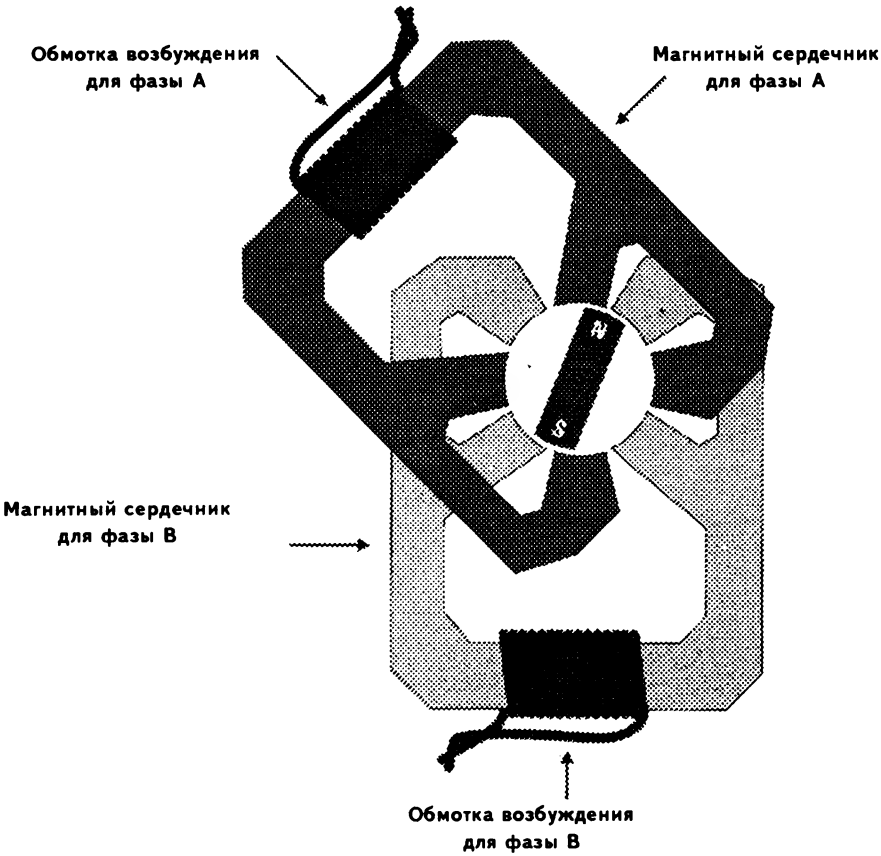
Представьте себе, что произойдет, если показанный на рис. 4.2 двигатель будет иметь очень много полюсов для каждого из магнитов. Управляя одним или обоими электромагнитами импульсами постоянного тока, можно повернуть ось двигателя в любую из очень большого числа позиций маленькими и четко определенными шагами. Здесь на каждом шаге применяется только включение и выключение магнитов, и этот способ оказывается чисто цифровым. Угол поворота оси двигателя определяется конструкцией двигателя. Размер шага совершенно не зависит от электрических сигналов; сигналы управляют только тем, когда двигатель сделает очередной шаг. Следовательно, не углубляясь в схему управления, можно считать, что шаговый двигатель всегда занимает точно определенную (угловую) позицию.

Если на оси укрепить небольшое зубчатое колесо (оно называется ведущей шестерней) и связать ведущую шестерню с большим зубчатым колесом, которое как бы развернуто (оно называется рейкой), то в процессе работы двигателя рейка будет двигаться линейными шагами. По существу, так и устроен привод позиционирования головок на базе шагового двигателя с ведущей шестерней и рейкой. Именно такой привод применялся в накопителях на жестких дисках компьютера IBM PC/XT, а затем и в большинстве дешевых накопителей.

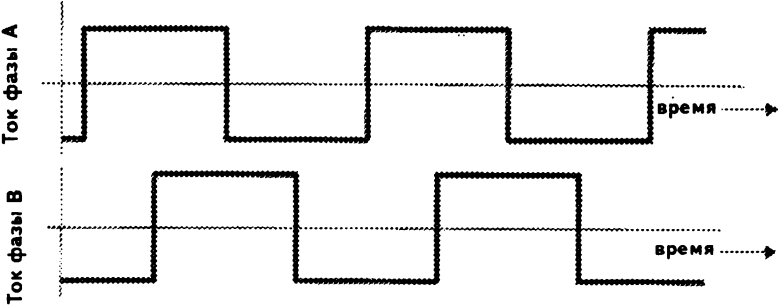
Когда шаговый двигатель соединен с ведущей шестерней и рейкой, он перемещает что-то (в нашем случае, блок головок) небольшими и четко определенными шагами по прямой (рис. 4.3). Такой привод передвигает головки в одно из множества заранее определенных мест. Точное расположение этих мест на поверхности дисков зависит от того, как сконструирован механизм (и каким образом он может изменить свою форму или размер из-за старения, износа, действия силы тяжести или какой-то внешней силы).

Данный способ перемещения головок в одну из нескольких определенных позиций иногда называют числением пути. Механизм оказывается довольно простым и дешевым и работает хорошо, чем и объясняется применение такого позиционера в

первых накопителях на жестких дисках. Принцип счисления пути применяется и сейчас в накопителях небольшой емкости.



(а)



(б)

Рис. 4.2. Многополюсный двигатель

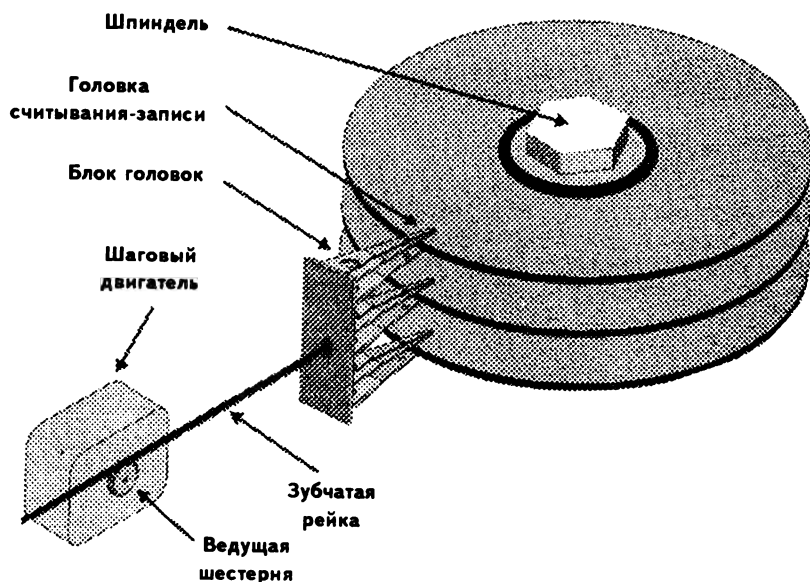


Рис. 4.3. Позиционер головок с шаговым двигателем

Однако счисление пути оказывается не наилучшим решением. Гораздо лучше приближенно перевести головки в нужную позицию, проверить, где они фактически находятся, а затем настронить с необходимой точностью. На таком подходе основаны рассматриваемые далее сервосистемы позиционирования головок.

В дисковых накопителях применялось несколько разновидностей счисления пути. На оси двигателя формировалась спиральная резьба, по которой перемещалась скользящая каретка (такая конструкция называется ходовым винтом). В другом варианте рейка заменялась большим колесом, вращение которого передавалось блоку головок с помощью металлической ленты. Все эти конструкции основаны на свойстве шагового двигателя поворачиваться на небольшие и четко определенные углы.

Мы рассмотрели самый распространенный способ построения системы позиционирования головок. Ее основой является двигатель, — типичный представитель большого класса электромоторов.

Однако существует совершенно другой способ построения двигателей, которые в этом случае обладают другими свойствами, в том числе и те, что очень удобны для построения эффективных позиционеров головок.

Несомненно, вам встречались подобные двигатели, но в таких местах, где их трудно принять за двигатели. Громкоговоритель в радиоприемнике представляет собой бумажный конус, прикрепленный к линейному электродвигателю специального вида. По-прежнему здесь есть два магнита, один из которых является постоянным магнитом, а другой электрическим. Они друг друга отталкивают. Один из магнитов (постоянный) фиксирован, а второй может перемещаться (рис. 4.4). Принцип действия громкоговорителя поясняет его поперечное сечение, представленное на рис. 4.4.

В отличие от обычного электродвигателя, который вращает ось, этот двигатель может перемещать только на короткое расстояние, пока звуковая катушка не выдвинется из постоянного магнита (или пока не упрутся во что-нибудь). Такое ограничение совершенно отлично от вращающегося двигателя, ось которого может поворачиваться на любое число оборотов, так как магнитное поле вращающейся части не выходит из сферы действия полей фиксированных магнитов.

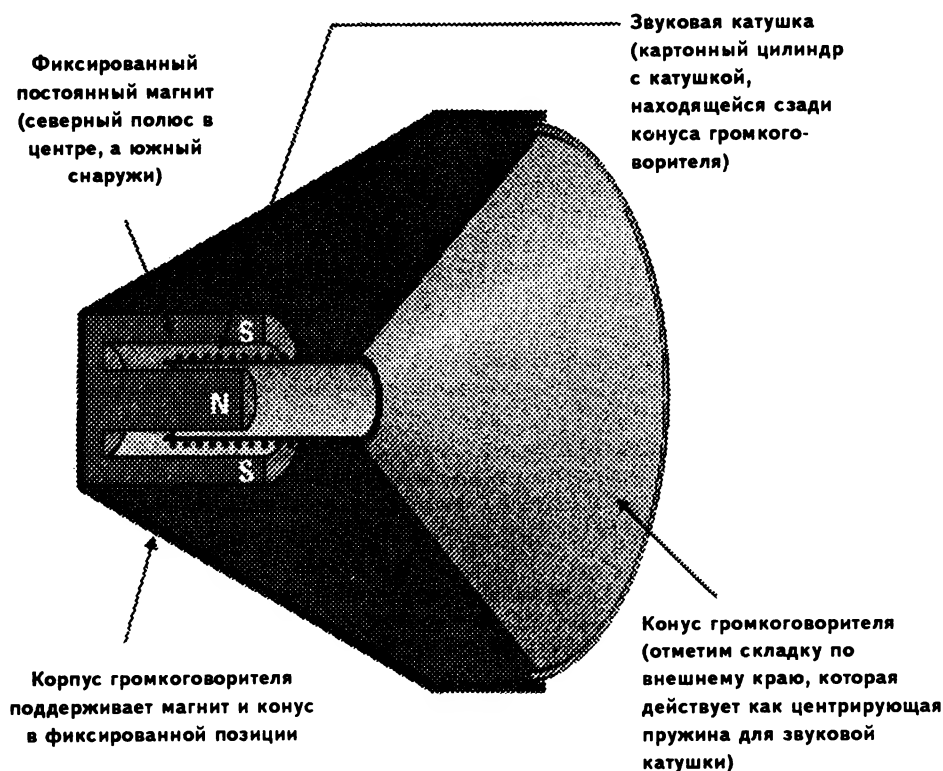
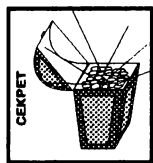


Рис. 4.4. Громкоговоритель представляет собой двигатель



Обычно в линейном электродвигателе с каждого конца рабочего диапазона движения предусматривается физический упор. Положительный ток в катушке толкает подвижную часть в одном направлении, а отрицательный ток — в другом. Чем сильнее ток, тем больше движущая сила и тем быстрее подвижная часть ускоряется в том или ином направлении.

Часто в конструкции предусматривается пружина, которая удерживает подвижную часть в середине ее диапазона движения. В громкоговорителе "пружиной" служит складка по внешнему краю конуса. Усилие такой пружины соизмеримо с усилениями, которые образуют проходящие через катушку токи.

В позиционере головок на базе двигателя со звуковой катушкой усилие пружины мало по сравнению с магнитными силами, возникающими при протекании тока в катушке. Это позволяет плавно перемещать блок головок в любом направлении для достижения нужной позиции. Но все же пружины достаточно для медленного возвращения катушки в ее "исходную" позицию при выключении тока.

Поскольку двигатель такого типа впервые применен в динамиках радиоприемников, подвижную часть (электромагнит) обычно называют звуковой катушкой. Звуковая катушка выталкивает и втягивает конус динамика, и мы слышим воспроизводимый при этом звук.

Напомним, что в шаговом двигателе ток подается постоянно, а двигатель переходит в новую позицию и останавливается. В двигателе со звуковой катушкой (по крайней мере, в двигателях позиционеров головок с относительно слабыми пружинами) подвижная часть ускоряется все время при действии тока. После начала движения единственный способ прекратить движение вблизи конца диапазона — пропу-

стить через катушку ток противоположного направления в течение времени, сравнимого с тем, когда блок головок ускорился.

Привод с шаговым двигателем, имеющий ведущую шестерню и рейку существенно отличается от привода со звуковой катушкой. В первом случае движение осуществляется шагами, причем размер шага определяется конструкцией шагового двигателя и зубчатого зацепления. Электрические сигналы определяют только, когда и в каком направлении осуществляется шаг. Такая система, по сути оказывается цифровой. В приводе со звуковой катушкой перемещение чего-то по прямой зависит от силы электрического тока, проходящего через катушку, и продолжительности действия тока. Следовательно, позиционер головок со звуковой катушкой — аналоговая система.

Примечание. Термины "аналоговый" и "цифровой" употребляются здесь как антонимы. В обычном смысле термин "аналоговый" означает аналогичный (т.е. очень похожий) на что-то. Термин "цифровой" относится к счету на пальцах (цифрах). Применительно к компьютерам и жестким дискам под аналоговыми сигналами понимают сигналы, которые могут принимать любые значения из некоторого диапазона, а цифровые сигналы принимают только вполне определенные дискретные значения. В математическом смысле аналоговые сигналы являются непрерывными переменными. Цифровые сигналы похожи на последовательные числа (1, 2, 3 и т.д.) и могут изменяться только шагами фиксированного размера. В предыдущем параграфе слова когда и в каком направлении подразумевают простые ответы типа да/нет, влево/вправо, т.е. цифровые значения. Слова же сила и продолжительность относятся к величинам, которые могут быть любыми значениями из некоторого диапазона, т.е. являются аналоговыми.



Почему постоянный магнит вращается или неподвижен

По-видимому, вы удивились почему в шаговом двигателе постоянным магнитом является ротор, а в линейном двигателе постоянный магнит фиксирован. Объяснение довольно простое.

Подключить гибкие проводники к звуковой катушке линейного двигателя не составляет труда. Обеспечить соединение при движении двигателя в пределах полного диапазона также нетрудно. Оформление звуковой катушки как подвижной части позволяет сделать ее массу небольшой, и этим достигается очень быстрое ускорение катушки.

В то же время, ось шагового двигателя может сделать много оборотов в одном направлении до перехода к повороту в другом направлении. Если бы в таких двигателях применять соединения прямо к обмотке ротора, эти проводники безнадежно запутались бы задолго до нужного числа оборотов. Поэтому гораздо проще построить шаговый двигатель с ротором из постоянного магнита, а фиксированный магнит сделать электромагнитом.

Можно также оформить оба магнита в виде электромагнитов. В двигателе постоянного тока для этого требуется переключать соединения к обмотке ротора по мере его вращения с помощью так называемого коммутатора. Во многих двигателях переменного тока для бесконтактной передачи сигналов переменного тока в ротор применяется вращающийся трансформатор. Такие двигатели в позиционерах головок жестких дисков не применяются.

Если "двигатель" со звуковой катушкой соединить с блоком головок дискового накопителя, то головки можно позиционировать на любой цилиндр, пропуская через катушку переменный ток в течение определенного временного интервала (рис. 4.5). Но если ток оказывается слишком сильным или слишком слабым или действует дольше или короче, головки передвинутся неизвестно куда. Для того, чтобы пода-

вать точно требуемый ток, в схеме управления применяется отрицательная обратная связь, которая называется также обратной связью.

Для наглядности можно сравнить работу такого позиционера с действиями водителя автомобиля. В обоих случаях задан требуемый курс (цилиндр для позиционера и полоса движения для водителя). И в обоих случаях после вывода управляемого объекта (блока головок или автомобиля) на нужный маршрут, система управления контролирует, где фактически находится объект в каждый момент времени; при необходимости она небольшими корректирующими действиями поддерживает объект на маршруте (головки на (или вблизи) центральной линии цилиндра или автомобиль в безопасном месте полосы движения).

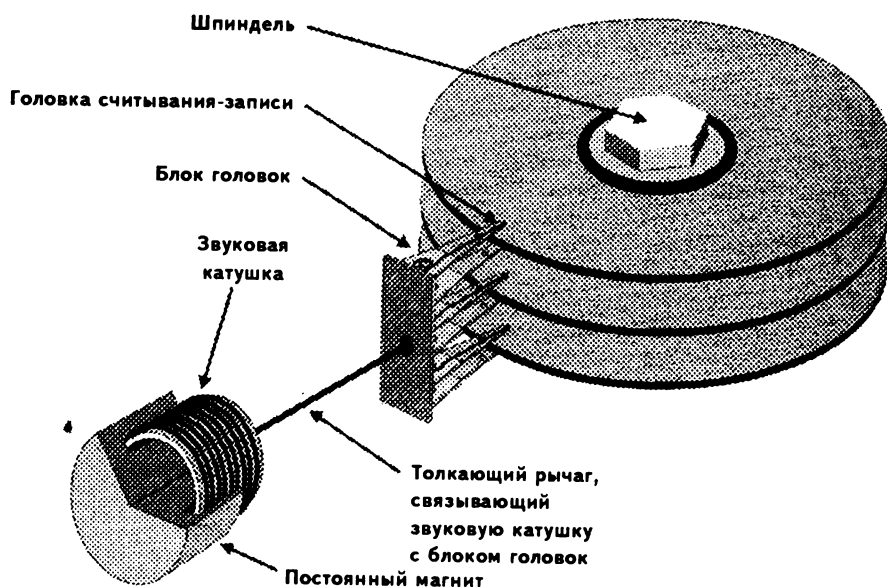


Рис. 4.5. Позиционер головок со звуковой катушкой



Новое название для старой идеи

Первое промышленное применение отрицательной обратной связи относится к системам управления, разработанным во время второй мировой войны. Такие системы были названы сервомеханизмами. Именно отсюда возник более короткий термин "серво", который сейчас применяется к подобным системам.

Однако первые случаи применения отрицательной обратной связи теряются в веках. Каждый из нас использует ее, чтобы просто не упасть или прийти в нужное место, а не блуждать хаотически "вокруг да около". Отрицательная обратная связь настолько вездесуща, что прошло много веков, прежде чем люди догадались дать ей специальное название.

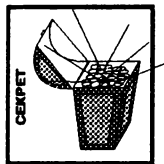
4.1.2. Преимущества двигателей со звуковой катушкой

Сервосистема имеет несколько преимуществ по сравнению с позиционером на базе шагового двигателя. Во-первых, по мере старения и износа деталей привода сервосистема самостоятельно компенсирует износ. Это объясняется тем, что система постоянно сравнивает текущую позицию головок с требуемой. Принцип сравнения помогает также компенсировать прогибы блока головок под действием силы тяжести.

Сервосистема оказывается также быстрее позиционера с шаговым двигателем. Различие в скорости объясняется тем, что в катушку можно кратковременно подать очень большой ток для более сильного толкания головок. В системе с шаговым двигателем сила тока в обмотках фиксирована.

Наконец, в сервосистеме есть встроенная защита поверхности диска. В конструкции привода предусмотрена схема парковки головок (т. е. перемещения их к шпинделю) при выключении питания. Позиционер с шаговым двигателем не предпринимает специальных действий при выключении компьютера, и головки часто просто остаются в своей последней позиции.

Отмеченное выше позволяет считать накопители с сервоприводом обладающими способностью автопарковки или самопарковки. В большинстве накопителей с шаговым двигателем парковку необходимо проводить специальной программой перед выключением компьютера.



Шаговый двигатель делает шаг и останавливается. При выключении питания он не движется. Двигатель со звуковой катушкой всегда осуществляет крошечные перемещения. При выключении питания в катушке она будет стремиться продолжить движение в том же направлении, в каком она двигалась до выключения питания. Чтобы гарантировать предсказуемое поведение двигателя со звуковой катушкой при выключении питания, машина запасает некоторую энергию. (В некоторых накопителях это электрическая энергия в виде заряда на конденсаторе, а в других — механическая энергия, запасенная вращающимися дисками.)

При выключении питания накопителя специальные схемы отключения подают эту энергию в звуковую катушку, заставляя двигатель перемещать блок головок либо по направлению к шпинделю до достижения механического упора, либо вне дисков. При выключении ПК можно услышать характерный звук (хотя в современных дисковых накопителях нужно очень внимательно прислушаться).

Можно заставить накопитель обычным образом "найти" определенную зону приземления (цилиндр, на котором данные не хранятся). Иногда после перемещения головок на позицию парковки в приводе срабатывает шпилька, фиксирующая привод в этой позиции. Такое действие (со множеством вариантов) называется автопарковкой.

Скорее всего, вы удивились, узнав, что головки паркуются около шпинделя, но для этого есть веские причины.

Для защиты головок и поверхности дисков лучше всего вывести их за пределы дисков. В некоторых накопителях реализован именно этот способ — головки отталкиваются от шпинделя до их выхода за край дисков. После этого между парами головок вставляется вилка, или клин, чтобы они не касались друг друга. Разделение пар головок, относящихся к каждому диску, позволяет быстро "втолкнуть" их в рабочую зону при включении питания.

Возможно, такой способ технически и наилучший, но его трудно реализовать. Поэтому в большинстве накопителей на жестких дисках головки позиционируются как можно ближе к шпинделю. Есть, по крайней мере, три веские причины выбора этого варианта. Во-первых, между цилиндром 0 и краем дисков очень мало места, поэтому при парковке всегда приходится учитывать риск того, что головки либо полностью выйдут за диски, либо окажутся на цилиндре 0. Напомним, что здесь хра-

нится самая важная информация — сведения о том, как обратиться ко всей остальной информации на диске.

Во-вторых — головки нужно парковать на цилиндр, который обычно не используется для хранения данных. Выбор для этого группы более коротких дорожек обоснован, так как хранение данных на более длинных дорожках оказывается более надежным. Разумеется, самые короткие дорожки находятся около шпинделя.

Наконец, представьте себе, что диски похожи на музыкальные тарелки. Если по какой-то причине диски испытывают удар, их края начинают вибрировать. Центр каждого диска жестко соединен со шпинделем и совсем не вибрирует. Если головки находятся вблизи края, вероятность касания ими вибрирующей поверхности выше, чем при позиционировании их вблизи шпинделя.

4.1.3. Позиционирование головки

Разработано несколько вариантов системы позиционирования головок в сервоприводе. В этом разделе рассмотрены шесть конструкций, которые применяются или предлагаются для применения в дисковых накопителях.

Внешние сервосистемы. Важнейшее требование к позиционеру головок с сервоуправлением заключается в том, что необходимо определять позицию головок в любой момент времени. После этого электроника может сравнить текущую позицию с требуемой и послать в привод со звуковой катушкой сигнал рассогласования для перемещения головок ближе к нужной позиции.

Для получения информации о текущей позиции к блоку головок можно подключить оптический шифратор. Одна из возможных конструкций с оптическим шифратором представлена на рис. 4.6. Здесь стеклянная пластинка со специальными метками движется между линейками светодиодов и фотодатчиков. Позиции линеек фиксированы относительно шпинделя, а стеклянная пластинка соединена с блоком головок.

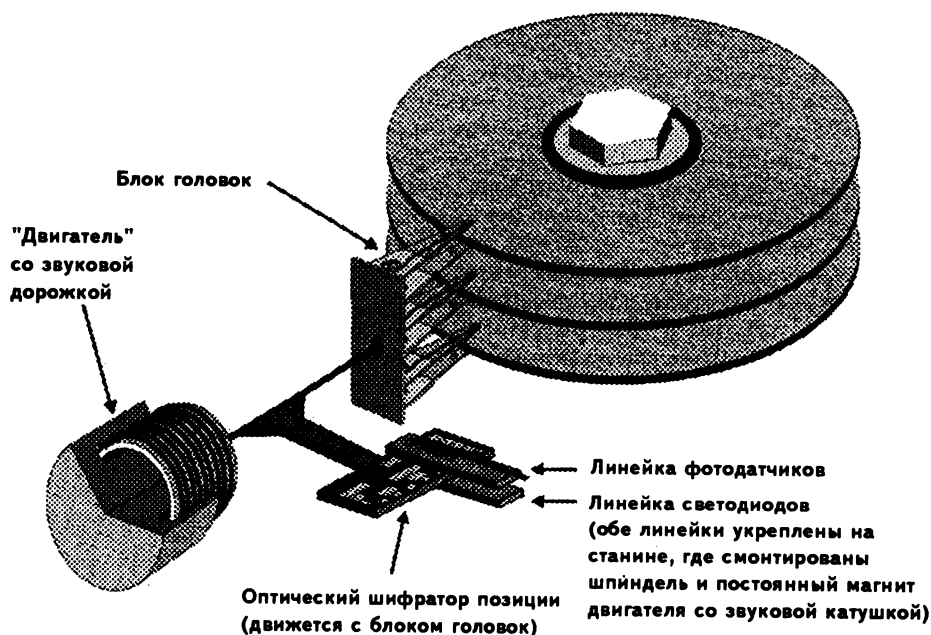


Рис. 4.6. Внешняя сервосистема позиционирования головок

Одно из достоинств данной конструкции состоит в том, что все дополнительное оборудование оказывается внешним по отношению к собственно дискам (в отличие от рассматриваемых далее конструкций). Фирма Plus Development (сейчас Quantum Corporation) для своего диска Hardcard выбрала именно такую конструкцию, что позволило оформить накопитель в виде сменной платы.

Недостаток этой системы заключается в том, что линейки светодиодов и фотодатчиков не должны смещаться относительно шпинделя, а пластинка оптического шифратора должна двигаться строго синхронно с головками. На практике полностью удовлетворить эти требования не удастся. Ни одна механическая система не может двигаться точно так, как вы ожидаете, и всегда имеется небольшие отклонения от маршрута движения. В рассматриваемых далее системах реализовано несколько оригинальных решений для компенсации этих небольших расхождений.

Давайте на минутку прервемся. Вы уже готовы перейти к рассмотрению других вариантов, но пока не представляете в деталях, как работает эта внешняя сервосистема. В частности, вы не знаете, какой узор меток наносится на стеклянную пластинку и как эти метки используются для определения позиции головок.

Применение в шифраторе кода Грея. Можно представить, что метки на пластинке — это просто черные и белые полосы, показывающие нули и единицы. В общем, так оно и есть, но это не все.

Дисковый накопитель должен знать о местонахождении блока головок со значительно большей точностью, чем расстояние между дорожками. Накопитель имеет примерно тысячу цилиндров, поэтому должно быть несколько тысяч позиций, в которых головки могут находиться в любой момент времени.

Сообщить схеме управления накопителя нужную информацию можно назначив адрес каждой возможной позиции головок. Адрес — это число от нуля до некоторого максимального значения. Адрес можно выразить двоичным числом, состоящим только из нулей и единиц. При выборе 12-битовых чисел можно представить адреса от 0 до 4095. Теперь необходимо спроектировать схему управления накопителя так, чтобы она надежно знала, в какой позиции находится блок головок в любой момент времени.

Показанная на рис. 4.6 пластинка имеет несколько параллельных столбцов из черных и белых меток (в примере число столбцов равно пяти), и для каждого столбца есть пара — светодиод-фотодатчик. Можно предположить, что для каждой позиции на пластинке соответствующий адрес закодирован двоичным числом, а затем нули и единицы этого числа записаны в различных столбцах пластинки. Такое предположение близко к истине, но не совсем.

Проблема не в том, что в примере нет 12 столбцов меток и 12 фотодатчиков; можно построить систему и с такими количеством столбцов и фотодатчиков. Более того, как показано далее, можно разработать систему, в которой не нужно регистрировать все биты каждого адреса. Реальная проблема связана с самими двоичными числами.

Приведем 21 первых двоичных чисел (их десятичные значения от 0 до 20):

0	0 0 0 0 0 1	11	0 1 0 1 1
2	0 0 0 1 0	12	0 1 1 0 0
3	0 0 0 1 1	13	0 1 1 0 1
4	0 0 1 0 0	14	0 1 1 1 0
5	0 0 1 0 1	15	0 1 1 1 1
6	0 0 1 1 0	16	1 0 0 0 0
7	0 0 1 1 1	17	1 0 0 0 1
8	0 1 0 0 0	18	1 0 0 1 0
9	0 1 0 0 1	19	1 0 0 1 1
10	0 1 0 1 0	20	1 0 1 0 0

Отметим, что при переходе от числа 7 (00111) к числу 8 (01000) одновременно изменяются четыре из пяти битов. Переход от 15 (01111) к 16 (10000) еще хуже: изменяются сразу все биты. При ближайшем рассмотрении оказывается, что в половине случаев при переходе от одного числа к соседнему изменяются минимум два бита.

Когда пластинка шифратора движется под датчиками, каждый переход от одного адреса к соседнему, требующий изменения более одного бита, может оказаться серьезной проблемой. Действительно, представим себе, что в одном из датчиков чуть-чуть нарушено выравнивание и он воспринимает изменение своего бита несколько позднее соседних датчиков. Пусть таким запаздывающим датчиком будет третий (средний); тогда при переходе от 15 (01111) к 16 (10000) появляется некоторый временной интервал, в котором выход окажется равным 10100, что соответствует позиции 20. Сервосистема посчитает, что блок головок ушел на 4-ю или 5-ю позицию (это кратковременно) и совершенно запутается. С этой проблемой разработчики сервосистем встретились уже давно и нашли элегантное ее решение.

Можно закодировать двоичные числа так, что при каждом переходе от одного числа к соседнему изменяется только один бит. Этот способ кодирования называется кодом Грея.

На первый взгляд, такое кодирование невозможно. Действительно, в половине переходов от обычного двоичного числа к соседнему изменяется более одного бита. Однако, к счастью для разработчиков сервосистем, оказалось, что двоичные числа можно всегда закодировать в коде Грея. Более того, есть много способов образования кода Грея для двоичных чисел с любым числом битов.

На рис. 4.7 приведен один из кодов Грея для пятибитовых двоичных чисел. Здесь показано 36 строк. Обратите внимание на центральные 32 строки. (Две верхние строки повторяют две нижние строки из центральных 32 строк, а нижние две строки аналогичны двум верхним строкам центральной части. Эти четыре добавленные строки, показанные серым цветом, демонстрируют, как набор повторяет сам себя, что требуется в некоторых применениях.)

Справа от каждой строки показано пятибитовое двоичное число. Эти числа не следуют в обычном численном порядке, а выстроены в порядке кода Грея. Он означает те последовательные позиции, для которых шифратор выдает их двоичные адреса. Если сервосистема, считывающая наборы черных и белых блоков, знает порядок их следования, она может надежно преобразовать любой из наборов в соответствующую позицию.

Теперь посмотрим, что произойдет при использовании на пластинке шифратора кода Грея вместо обычных двоичных чисел, представляющих адрес каждой позиции. По-прежнему выравнивание датчика может быть нарушено. Но при движении пластинки с одной позиции (соответствующей одному двоичному значению) на другую изменение с черного на белое или наоборот зафиксирует только один датчик. Таким образом, вопрос о восприятии при движении пластинки переходов несколькими датчиками снимается.

Код Грея в накопителях на жестких дисках позволяет точно сообщить о текущей позиции головок (с разрешающей способностью, равной шагу меток в столбце младшего бита). Независимо от числа невыровненных датчиков они показывают увеличение на один адрес всякий раз, когда блок головок перемещается на расстояние, равное шагу меток в столбце младшего бита, без всяких "фантомных" промежуточных позиций.

Другие надежные способы определения позиции. Есть и альтернативы использованию полного кода Грея. Как это ни удивительно, можно обойтись всего одним рядом черных и белых меток и двумя парами светодиода-фотодатчика. Одна пара датчиков позиционируется так, что она находится в середине полосы когда вторая пара находится на границе между метками.

В такой конструкции минимум оптического оборудования позволяет все же всякий раз обнаружить то, когда головка перемещается на шаг между метками, и безо-

шибочно определить, в каком направлении она движется. Если знать отправную точку и никогда не сбиваться со счета числа пройденных полосок и направлений их прохождения, можно всегда знать текущую позицию.

Такой простейший способ оптического шифрования применялся в коммерческих жестких дисках, например в некоторых накопителях Hardcard. Для его правильной работы требуется абсолютно точное слежение за сигналами от шифратора.

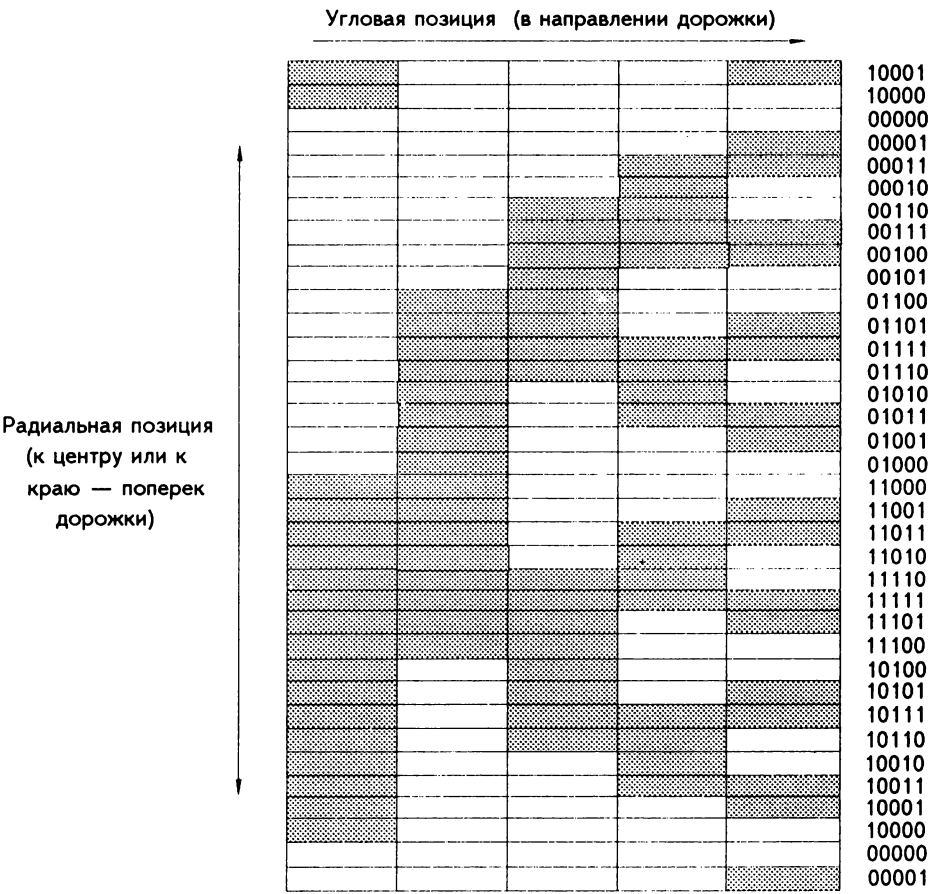


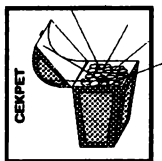
Рис. 4.7 Пример кода Грея

Промежуточное решение заключается в кодировании нескольких бит адреса каждой позиции (с помощью кода Грея), а не всех битов в адресе. Как ни странно, но можно безопасно игнорировать старшие биты, потому что головки никогда не могут мгновенно прыгнуть от одного адреса к другому, достаточно далекому от него. Всегда необходимо воспринимать каждый промежуточный адрес.

Если предположить, что перемещение более чем на 100 дорожек сразу никогда не фиксируется, для регистрации позиции можно обойтись семью младшими битами. При таком способе экономится место на пластинке шифратора (и, что более важно, поверхность диска, если сигналы записаны на диске), но возникает возможность потери позиции без всякого оповещения об этом. Чем больше битов привлекается для шифрования (начиная с младших и подключая старшие), тем надежнее определяется текущая позиция. В приведенном ранее простейшем способе используются всего два младших бита.

Сервосистемы со специальным (выделенным) диском. Внешние сервосистемы работают достаточно хорошо, но при повышении плотности цилиндров все более проявляются их недостатки. Правда, можно разместить информацию обратной связи в такой же структуре, как и данные. В этом случае даже при перекосе станины накопителя или, что вероятнее, прогибе или изгибе блока головок обратная связь точно покажет позицию головок относительно данных.

В первой и до сих пор применяемой реализации этого способа для хранения сервоинформации выделяются одна или две поверхности многодискового накопителя. Обычно в заводских условиях на верхней или нижней поверхности (а иногда на обеих) записываются специальные сервосигналы. Остальные поверхности (между верхней и нижней) предназначены для хранения данных.



Наличием специальных поверхностей объясняется следующий загадочный факт: для некоторых накопителей в справочных таблицах сообщается нечетное число головок. Поскольку каждый диск имеет верх и низ, можно полагать, что во всех дисковых накопителях должно быть четное число головок. Ответ на эту загадку состоит в том, что в таблицах не учитываются головки, выделенные для считывания сервоинформации; сообщается только о головках, предназначенных для записи и считывания пользовательских данных.

На рис. 4.8 показана блок-схема накопителя со специальным диском, в котором для хранения сервоинформации применяется только верхняя поверхность верхнего диска.

Что же записывается в качестве сервосигналов? Некоторая часть записываемых сигналов очевидна и в каждом накопителе она записывается одинаково. Однако часть информации находится под авторской защитой фирмы-производителя и сведений о такой информации обычно нет.

Схематически эти части показаны на рис. 4.9. Первая часть представляет собой шестибитный код Грея (см. рис. 4.7). Вторая часть показана серым цветом и ее содержимое неизвестно. По всей поверхности повторяются набор, состоящий из двух частей. Число повторений набора никак не связано с числом секторов на дорожке. Набор просто повторяется максимальное число раз. Всякий раз, когда набор проходит под сервоголовкой, механизм управления диском получает очередное показание местонахождения головок.

Если посмотреть на серводиск реального накопителя, он, конечно, не абсолютно точно повторяет рис. 4.9. Прежде всего, код Грея содержит гораздо больше битов. И, конечно же, информация не раскрашена на поверхности так, чтобы ее могли видеть люди; вся информация записана на магнитном покрытии поверхности. Считайте, что на рис. 4.9 черный квадрат представляет собой участок, намагниченный в прямом направлении вдоль дорожки (когда головка проходит над этим участком), а белые квадраты намагничены в противоположном направлении.

Несколько битов номера каждой позиции записаны на поверхности друг за другом. Вместо нескольких датчиков (по одному на бит), как показано для оптического индифратора на рис. 4.6, записанные биты считываются одной головкой, конструкция которой такая же, как и у головок для считывания и записи данных.

Некоторые фирмы экономят несколько битов на поверхности диска, записывая только несколько младших битов номеров радиальной позиции. Этот метод работоспособен, но требует рекалибровки (перевода головок на цилиндр 0) и последующего перехода на нужный цилиндр, когда нет уверенности о реальном местонахождении головок.

Проблема невыровненных датчиков в многодорожечной линейке датчиков, которая привела к появлению кода Грея, теперь превращается в проблему смещения блока головок немного вовнутрь или наружу во время прохождения цифрового кода позиции. По-прежнему использование кода Грея оказывается единственной гарантией того, что система управления никогда не получит искаженную картину местонахождения блока головок.

Набор сервоинформации записывается в каждой радиальной позиции, включая места на дорожке и между дорожками. Цифровые данные не могут сообщить схеме управления о местонахождении головок точнее половины расстояния между дорожками (или цилиндрами).

Отмеченные на рис. 4.9 сноской "Информация о точном позиционировании" части информации оказываются различными у каждой фирмы-производителя. После обработки цифровых данных о позиции и перемещении блока головок на нужное место, известно, что головки находятся вблизи нужного цилиндра. Информация о позиционировании помогает с максимальной точностью "навести" головки на центр этого цилиндра и возвращать их туда максимально быстро в случае отклонения.

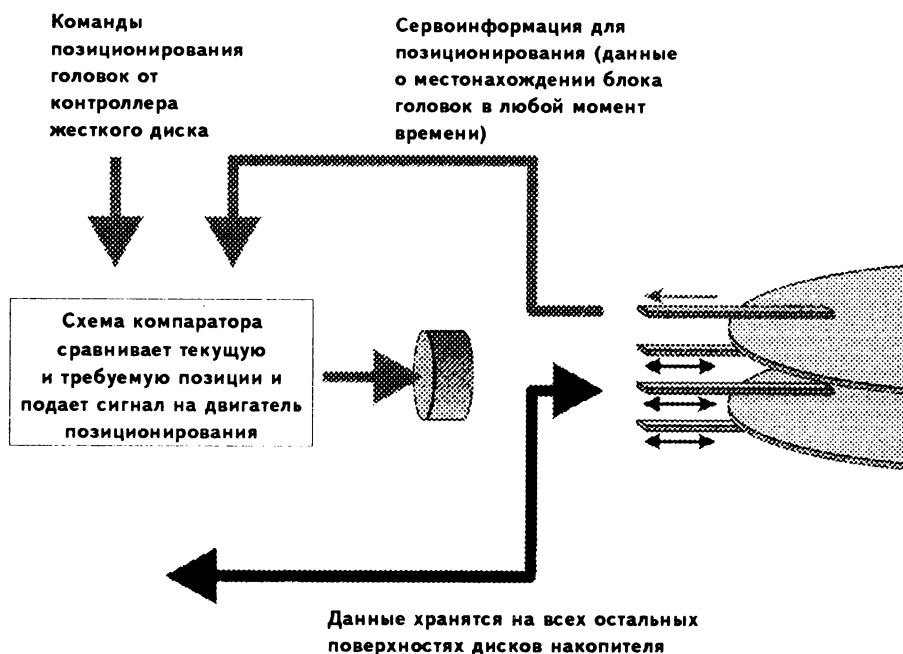


Рис. 4.8. Сервосистема со специальным диском

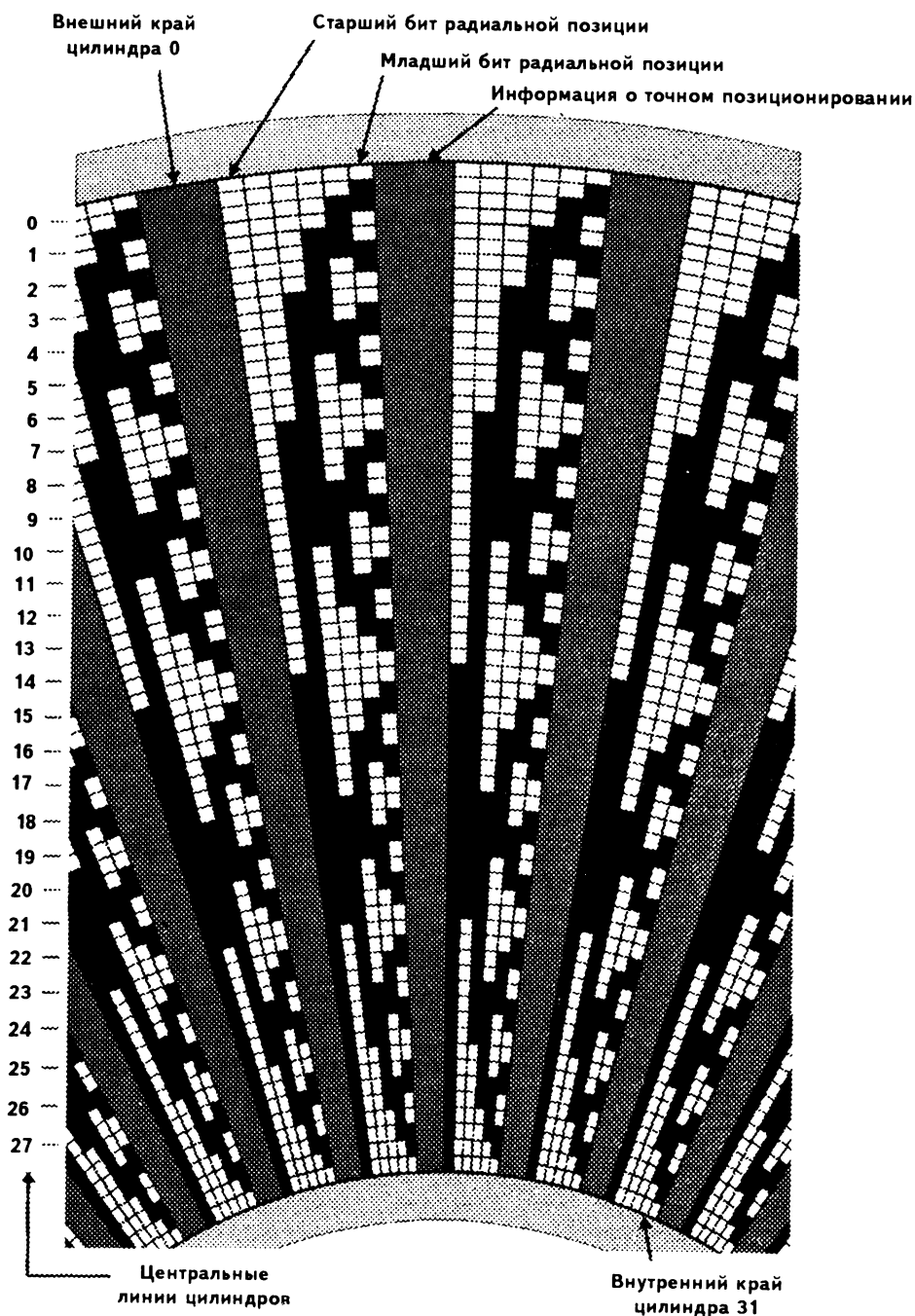


Рис. 4.9. Формат дорожек на специальном диске



Аналоговые и цифровые сигналы обратной связи

В первых сервосистемах позиционирования применялась обратная связь в виде аналоговых сигналов; т.е. представляющий позицию сигнал изменялся по величине, а не кодировал цифровое значение позиции. В первых накопителях на жестких дисках применялись аналоговые сервосигналы.

Сейчас промышленность отходит от аналогового сервопривода по тем же причинам, по каким звуковая запись осуществляется теперь, в основном, цифровыми средствами (на компакт-дисках или цифровой ленте), а не аналоговыми средствами (на пластинках или аналоговой ленте). Это объясняется тем, что цифровые системы менее подвержены помехам (т.е. оказываются более точными), а также появлением микропроцессоров, которые значительно удешевляют цифровые системы.

Сервосистема со специальным диском имеет одно важное преимущество перед остальными системами позиционирования головок и ряд серьезных недостатков. Достоинство в том, что сервоинформация всегда подается в систему управления, поэтому удар по накопителю, или смещение блока головок относительно дисков по какой-то другой причине фиксируется почти мгновенно.

Первый серьезный недостаток связан с тем, что сервоинформация занимает целую поверхность диска (или даже две поверхности), а это приводит к снижению полезной емкости диска.

Второй существенный недостаток (характерный и для внешней сервосистемы) состоит в том, что здесь предполагается точное совместное движение всех головок. В этой системе приходится полагать, что головки неподвижны относительно друг друга, а во внешней сервосистеме предполагается, что шпиндель и внешняя линейка датчиков не смещаются относительно друг друга.

Конечно, головки должны двигаться вместе, но достичь абсолютной точности здесь не удастся. Как и во всех механических системах, время от времени возникают незначительные разбросы.

Рассмотренная сервосистема гарантирует, что при запросе контроллера позиционировать головки, например на цилиндр с номером 85, сервоголовка точно позиционируется в центр дорожки 85-й сервоповерхности. Однако система не гарантирует, что все остальные головки окажутся точно в центрах дорожек соответствующих поверхностей.

Например, ошибка может появиться, если дорожки данных первоначально записывались при покупке компьютера (подробности о форматировании низкого уровня приведены в главе 8). Позже пользователь может повернуть системный блок набок. Такое изменение ориентации может вызвать некоторое скручивание или наклон блока головок. При этом все головки, которые считывают и записывают данные, несколько смещаются относительно дорожек. Такой дефект обычно называется эффектом наклонной башни.

Клиновые сервосистемы. При первой попытке справиться с эффектом наклонной башни перенесли сервоинформацию со специального диска на те же поверхности, где находятся данные. Другими словами, часть каждой поверхности хранит сервоинформацию, а на остальной поверхности находятся данные. (Переход от системы со специальным диском к клиновой системе означает, что для хранения данных используется еще одна поверхность, но часть каждой поверхности приходится выделять для сервосигналов. Выигрыш и проигрыш примерно одинаковы.)

На рис. 4.10 схематически представлена клиновая сервосистема. Как всегда, на рисунке невозможно показать реальное число цилиндров; кроме того, не все накопители имеют всего 17 секторов на дорожке. В первых конструкциях таких накопителей применялся один пакет сервоинформации для каждого оборота диска.

Поскольку в одном пакете содержится не много сервоинформации, накопитель должен ожидать нескольких оборотов до точной фиксации на нужном цилиндре.

Кроме того, нельзя узнать о возможном дрейфе головок до следующего восприятия сервоклина.

Обе указанные проблемы снижают полезность такой сервосистемы. С появлением более совершенных конструкций чистые клиновые сервосистемы не применяются.

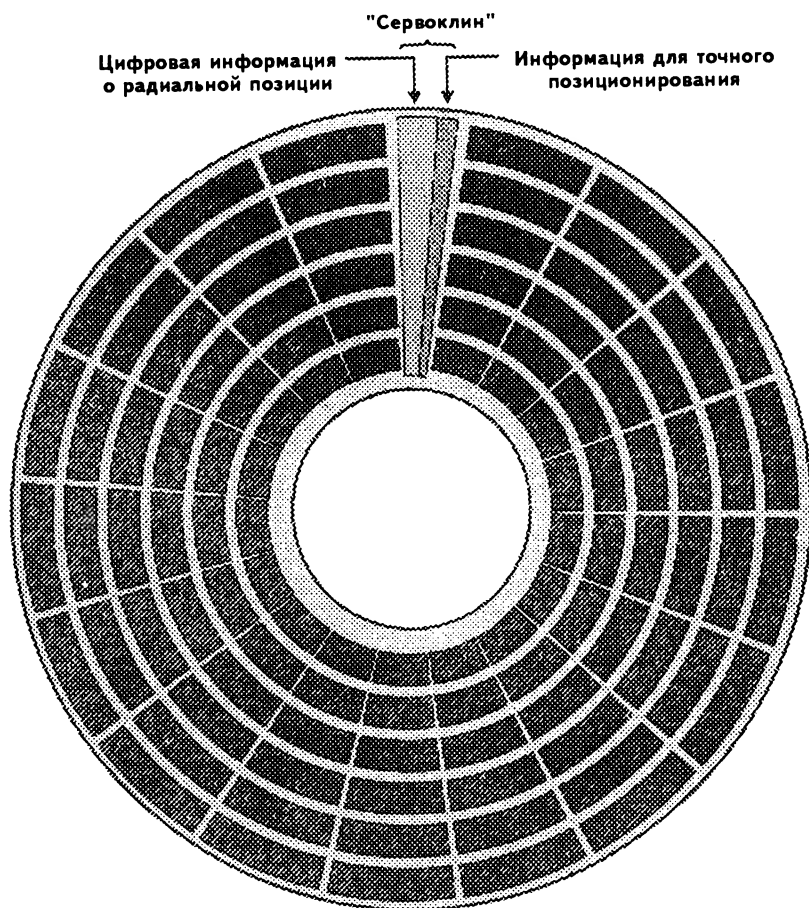
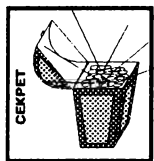


Рис. 4.10. Клиновая сервосистема

Встроенные сервосистемы. Одна из лучших конструкций называется дисковым накопителем со встроенной сервосистемой. Такой накопитель похож на рассмотренную клиновую сервосистему, но здесь пакет сервоинформации записывается на диске перед каждым сектором. На рис. 4.11 показан накопитель, имеющий всего 13 дорожек и 17 секторов на дорожке. В реальных накопителях обычно больше 17 секторов и всегда намного больше дорожек.



В некоторых накопителях в различных зонах поверхности диска имеется разное число секторов. Например, во внешней зоне число секторов может быть равно 75, в средней зоне — 50 секторов, а во внутренней зоне всего — 25 секторов. В таких накопителях клин сервоинформации приходится на целый диапазон радиальных позиций (на дорожке и вне дорожки) и для данной зоны записывается

перед каждым сектором внутри зоны. Для приведенного примера должно быть 75 пакетов сервоинформации на оборот во внешней зоне и всего 25 пакетов на оборот во внутренней зоне.

Достоинства встроенных систем очевидны. Головки никогда не могут отойти от дорожки так, чтобы электроника управления вскоре не узнала об этом факте (перед считыванием или записью следующего сектора). Проблема эффекта наклонной банши полностью снимается.

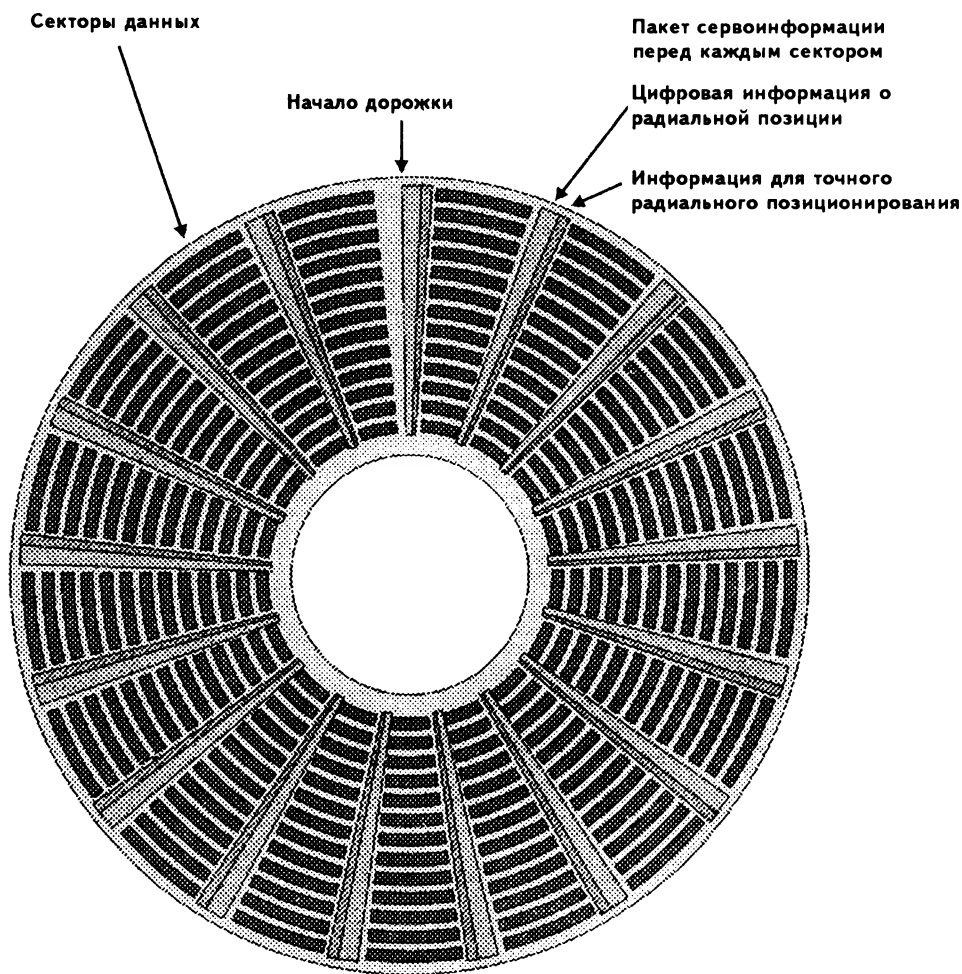


Рис. 4.11. Встроенная сервосистема

В большинстве современных дисковых накопителей применяются встроенные сервосигналы. В этом случае почти никогда не требуется повторять форматирование низкого уровня для выравнивания дорожек и тех головок, которые на них считывают и записывают данные.

Однако у встроенной сервосистемы имеется и несколько слабых мест. Одно из них — сервосигналов не так много. Наличие коротких пакетов сигналов означает,

что при отклонении головок от дорожки (или при переходе на другую дорожку) система дольше позиционирует головки на дорожку. Вторым недостатком — даже при наличии пакета сервоинформации перед каждым сектором, при записи или считывании в середине сектора невозможно узнать о том, что головки "сошли" с дорожки.

Устранить обе указанные проблемы можно при комбинированном подходе, объединяющем лучшие качества специального диска и встроенной сервосистемы.

Гибридные сервосистемы. В лучших современных конструкциях жестких дисков применяются две специальные поверхности (верхняя и нижняя) и встроенная сервоинформация перед каждым сектором на всех поверхностях с данными. Такая конструкция обеспечивает быстрый переход на любой цилиндр, привлекая сервоинформацию со специальных поверхностей. При наличии двух специальных поверхностей накопитель усредняет позиции от каждой из них, что частично компенсирует эффект наклонной башни.

Если блок головок немного прогнулся (отсюда и возник термин наклонная башня), применение двух специальных поверхностей позволяет компенсировать его последствия. Если сдвиг головок не линейен (график зависимости сдвига головки от номинальной позиции и номера головки не является прямой), приходится сделать кое-что еще.

На практике такие нелинейности встречаются. Поэтому в лучших накопителях есть две специальные поверхности и встроенная сервоинформация на каждом диске. Специализированные поверхности обеспечивают быстрое грубое позиционирование. Встроенная сервоинформация применяется для точной настройки позиции головки на той конкретной поверхности, где происходит считывание или запись.

Комбинированный подход значительно улучшает время установления (как быстро накопитель после команды перехода на новый цилиндр фактически позиционирует на нем головки). Улучшение этого показателя сокращает среднее время обращения к хранимым данным.

Возможно, самое существенное достоинство таких систем состоит в том, что они непрерывно контролируют позицию головки и даже при считывании или записи в середине сектора знают о том, не отошел ли блок головок от нужного места. В лучших накопителях эта информация применяется для подавления считывания или записи во время такого переходного перемещения. Временное подавление операции предотвращает запись на волнистые дорожки и помогает устранить неправильное считывание содержимого сектора.

Как всегда в нашей жизни, за такие преимущества приходится платить. В данном случае необходимо применять две специальные поверхности, а на остальных поверхностях — выделять участки для встроенных сервосигналов.

Вмонтированные сервосистемы. В идеальной системе сервоинформация о позиции должна поступать через те же головки, которые считывают и записывают данные, причем поступление должно происходить в то же время, когда происходит считывание или запись данных. На первый взгляд кажется, что практическое воплощение такой идеальной системы невозможно.

Однако некоторые инженеры с этим не согласны. Они предложили именно такую идеальную систему, которая получила название вмонтированной сервосистемы, где сервоинформация записывается под данными (глубже в магнитном покрытии диска). Специальные головки и электронные схемы позволяют отделить сервоинформацию от потока данных.

Такой метод лучше всех остальных сервосистем. Он обеспечивает быстрое позиционирование головок и надежное удержание их на дорожке. Система почти мгновенно фиксирует отход головок от дорожки. Кроме того, для сервоинформации не расходуется рабочие поверхности дисков.

У вмонтированных сервосистем есть только один заметный недостаток — они сложнее и, следовательно, дороже всех остальных систем. Их достоинства по срав-

нению с гибридными системами не оправдывают возросшей стоимости, поэтому в коммерческих накопителях для ПК вмонтированные сервосистемы не применяются.

4.1.4. Заголовок сектора

До сих пор мы рассматривали позиционирование головок на дорожки. Конечно, — это важно, поскольку без правильного позиционирования невозможно считать хранимые на дорожках данные (или записать на них новые). Но это только часть того процесса, в результате которого накопитель по вашему запросу отыскивает и возвращает нужные данные.

В главе 3 показано, что информация записывается на диск секторами, каждый из которых содержит 512 байт данных и имеет предшествующий заголовок сектора. Заголовок сектора содержит трехмерный адрес (головка, цилиндр и физический номер сектора).

Информация в заголовке сектора записывается так же, как и данные (например, с применением того же способа кодирования), и считывается той же электронной схемой, которая считывает данные. Эта схема совершенно отличается от тех схем, которые считывают и обрабатывают сервоинформацию. (Это отличие будет даже в клиновых, встроенных и гибридных сервосистемах, в которых одни и те же головки считывают и данные, и сервоинформацию.)

Таким образом, одни схемы поддерживают головки над дорожками с записанной информацией, а другие — отыскивают нужную вам информацию среди головок. (Еще один сервомеханизм каждого накопителя, который обеспечивает постоянство скорости вращения, рассматривается далее.)

Если вернуться к аналогии с вождением автомобиля по скоростной автострате, то механизм позиционирования головок соответствует водителю, а схемы считывания данных — штурману. Штурман смотрит на дорожные знаки и сообщает водителю, когда сменить дорогу, а водитель должен повернуть на эту дорогу и двигаться по ней до следующей команды о смене дороги.

Схемы считывания и интерпретации хранимых данных должны также обеспечивать выбор сектора с нужными данными. Когда под головкой находится именно этот сектор, схема должна либо воспринять содержимое секции данных, либо включить схему записи для того, чтобы головка записала в этой секции новые данные.

На этом мы закончим рассмотрение позиционирования головок и обратимся к тому, что делают головки при считывании или записи и как они выполняют эти операции.

4.1.5. Конструкции головок считывания-записи

Мы уже говорили о том, что головки считывания-записи представляют небольшие электромагниты, находящиеся сверху и снизу каждого диска, которые выполняют запись информации на диск и считывание с диска. Вы знаете, что при вращении диска головки находятся на расстоянии долей микрона от поверхности диска. Когда вращение прекращается, головки опускаются на диск.

Возможно, самое важное о головках считывания-записи — это то, что они всегда приобретают некоторую остаточную намагниченность. При правильной конструкции головки эта намагниченность оказывается полезной, повышая чувствительность головок при восприятии ею магнитных полей с поверхности диска. К сожалению, эта намагниченность играет и отрицательную роль.

Поскольку головки всегда являются магнитами (а не только когда компьютер использует их для записи сигналов на поверхности диска), прохождение их над диском ослабляет сигналы на его поверхности, возможно, уничтожая ("затирая") небольшие порции. Это ведет к преждевременной гибели жесткого диска и, следовательно, определяет необходимость всегда парковать головки на безопасный цилиндр при выключении двигателя.

На практике вам достаточно знать о конструкции головок считывания-записи следующие важнейшие факты: у головок есть обмотки на магнитном сердечнике, а у сердечника — небольшой зазор, через который магнитные силовые линии выходят и проникают в магнитное покрытие диска. Типичная конструкция головки, называемая индуктивной головкой считывания-записи, показана на рис. 4.12.

Но если вы хотите разобраться в тонкостях конструкции головок, понять смысл таких встречающихся в рекламе терминов, как накопители с MIG, сдвоенный MIG, тонкопленочная головка и головка с восприятием потока, следует детальнее изучить конструкции головок. Новейшее направление в области магнитных дисков, а именно вертикальная запись, также связано с особенностями конструкции и работы дисковой головки считывания-записи.

Если вы удовлетворены простейшим объяснением принципа работы магнитных головок, можно сразу перейти к разделу "Электронное обрамление накопителя". Но если вы хотите знать тонкости конструкций магнитных головок, прочитайте весь последующий материал.

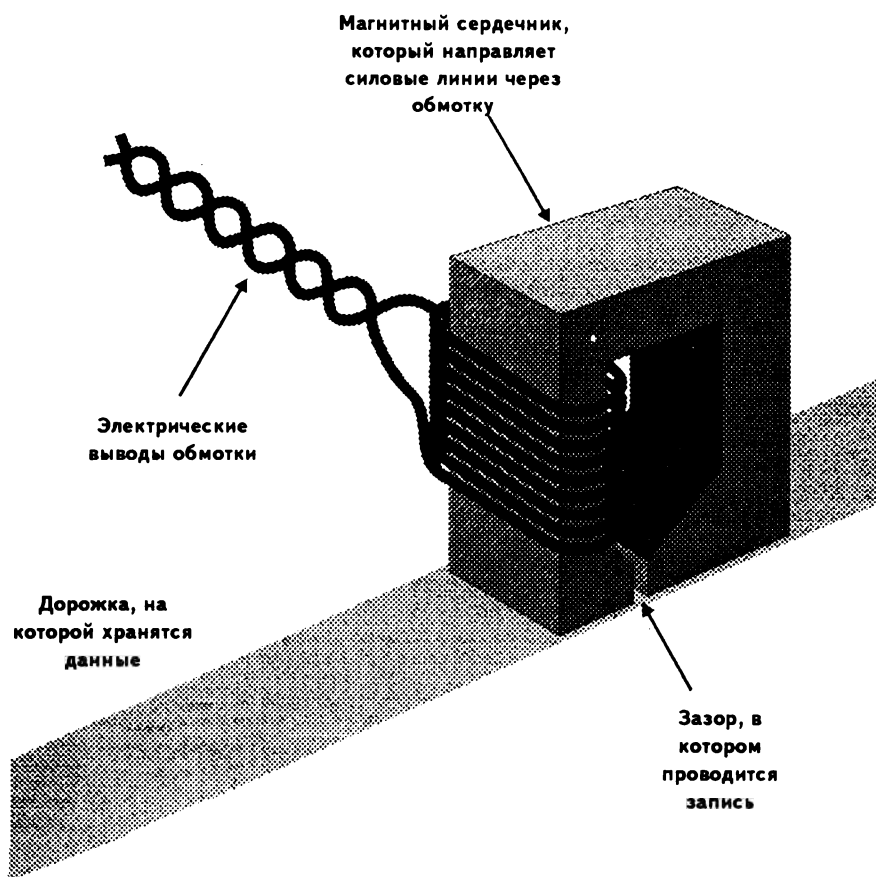


Рис. 4.12. Индуктивная головка считывания-записи

4.2. Продольная запись

В главе 2 процессы считывания и записи (см. рис. 2.4) описаны для конкретного типа головки считывания-записи и конкретного способа записи на магнитной поверхности диска. Показанная разновидность называется продольной записью с индуктивной кольцевой головкой.

Приведенный термин объясняется тем, что магнитные силовые линии, которые выходят из зазора и проникают в материал поверхности, в основном действуют параллельно поверхности — вдоль длины дорожки (отсюда возник термин "продольная"). Форма сердечника головки напоминает своеобразное кольцо (поэтому головка называется кольцевой).

Можно сконструировать совершенно другие головки. В некоторых из них применяется тонкопленочная конструкция вместо обмотки на сердечнике. В других головках магнитные силовые линии действуют перпендикулярно поверхности диска. Наконец, в третьих головках запись происходит обычным образом, а считывание опирается на совершенно другой механизм. Все упомянутые конструкции головок кратко рассматриваются в следующих разделах.

На рис. 4.12 показана увеличенная конструкция индуктивной кольцевой головки. Разработано несколько других форм головок с учетом простоты изготовления и высокой чувствительности. Однако по своей сути все кольцевые головки аналогичны показанной на рис. 4.12.

Тонкопленочные головки. Пользователям компьютеров требуются дисковые накопители все большей емкости, поэтому фирмы-производители всегда стремятся увеличить емкость своих изделий. Кроме того, заметна тенденция сокращения физических размеров накопителей. Для этого необходимо сокращать промежутки между цилиндрами и повышать плотность записи битов на дорожке.

Достижение первой цели (более близкое размещение дорожек) зависит от разработки прецизионных сервосистем, которые могут точнее позиционировать головки. Здесь же требуется создание более узких головок или, по крайней мере, головок, записывающих на более узкие дорожки. Достижение второй цели (повышение плотности записи) практически целиком зависит от улучшения конструкции головок и повышения качества поверхностей дисков.

В обоих случаях важно уменьшение размеров головок. Один из способов заключается в переходе от ферритовых сердечников с обмотками (как это показано на рис. 4.12) к тонкопленочным головкам. Есть много конструкций тонкопленочных головок; типичная конструкция представлена на рис. 4.13.

Основу конструкции составляет ферритовая пластинка. Она одновременно служит несущей для остальных компонентов головки и одной половиной магнитной цепи.

Все остальные элементы головки выполняются по технологии современных интегральных схем. Показанный на рис. 4.13 зазор образуется при осаждении на пластинку очень тонкого слоя некоторого немагнитного материала под обмоткой и под нижним концом переднего полюса.

Такая технология позволяет производить крошечные головки со сравнительно невысокой стоимостью. Тонкопленочные головки быстро превратились в промышленный стандарт.

Головки с восприятием потока. Работа всех головок записи опирается на один физический закон: создание магнитного поля при пропускании электрического тока через обмотку. (В тонкопленочных головках "обмотка" состоит всего из одного витка, но все равно он действует как обмотка.) Ток вызывает появление магнитного поля, которое концентрируется в сердечнике, сделанном из некоторого магнитного материала. выходящее из полюса сердечника поле оставляет намагниченные участки на поверхности диска, которые и хранят информацию.

Все дисковые головки записи действуют аналогично, но этого нельзя сказать о головках считывания. Большинство головок считывания являются индуктивными, но некоторые воспринимают магнитный поток. Магнитный поток — это мера магнитного поля, проходящего через конкретную область. Можно считать поток числом силовых линий или силой магнитного поля (магнитной интенсивностью), умноженной на площадь области.

Индуктивные головки воспринимают только изменение намагниченности того участка поверхности диска, который проходит под ними. Когда головка проходит над намагниченным участком, она пропускает часть поля от этого участка через свою обмотку. Если участок намагничен не так, как соседние, поле в обмотке изменяется, что вызывает появление импульса напряжения, который можно обнаружить.

Головки считывания с восприятием потока работают совершенно по-другому. Их выход зависит только от размера магнитного поля, проходящего через чувствительный элемент, а не от скорости изменения намагниченности. При конструировании чувствительного к потоку элемента используется несколько различных физических принципов. В частности, применяются магниторезистивные полоски, датчики эффекта Холла и усилители с коммутацией потока.

Эти датчики применяются только для считывания магнитного поля, а не для его образования, поэтому их можно объединить с обычной индуктивной головкой записи. Такой гибрид монтируется на ползунке и присоединяется к рычагу блока головок так же, как обычная индуктивная кольцевая головка считывания-записи. (Принцип действия ползунка рассмотрен далее во врезке "Крах головки — реальна ли проблема?")

Довольно экзотические головки с восприятием потока представляют интерес и применяются в некоторых других видах магнитной записи, но в накопителях на жестких дисках для ПК они пока не используются.

Несмотря на совершенно другую технологию производства, работа головки с восприятием потока во многом похожа на действие кольцевой головки. В частности, информация записывается по участкам дорожки, намагниченность которых ориентируется вдоль дорожки (назад или вперед). Такой метод называется продольной записью.

Ограничение на плотность битов при продольной записи. Другой способ, в котором намагниченность участков на поверхности диска направлена вверх или вниз (т.е. внутрь поверхности или наружу), называется перпендикулярной или вертикальной записью. Отличия этого способа от продольной записи рассматриваются в следующем разделе.

На рис. 4.14 показано увеличенное поперечное сечение диска вдоль дорожки, на которую сделана продольная запись. Отметим, что стрелки, показывающие намагниченность, направлены влево или вправо. Отметим также, что границы участков с разной намагниченностью очень неровные.

Неровность краев участков объясняется физическими причинами и имеет одно очень важное следствие. Нельзя сделать продольно записанный участок произвольно малым. Если все же попытаться сделать это, участок просто исчезнет. Такая ситуация приводит к ограничению числа битов на единицу длины дорожки, которое можно записать на диск.

Критические размеры головок считывания-записи с продольной записью. Еще одно ограничение числа битов, которое можно записать на единицу длины дорожки, связано с конструкцией головки. Минимальная длина магнитного триггера определяется некоторыми критическими размерами головки. На рис. 4.15 представлены боковой вид кольцевой головки для продольной записи и поперечное сечение части диска. На рисунке показаны три места, в которых размер оказывает критическое влияние на производительность головки.

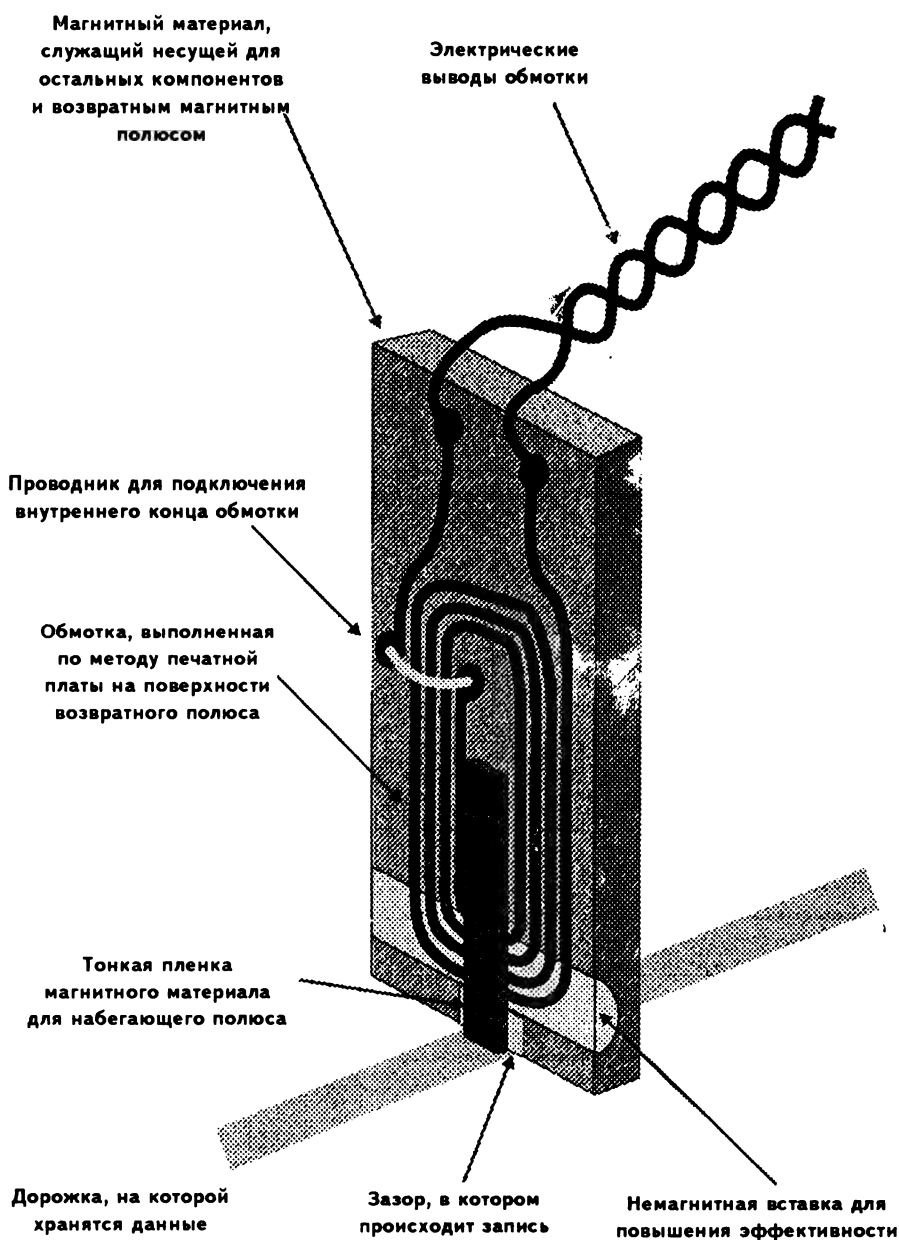


Рис. 4.13. Тонкопленочная (индукционная) головка

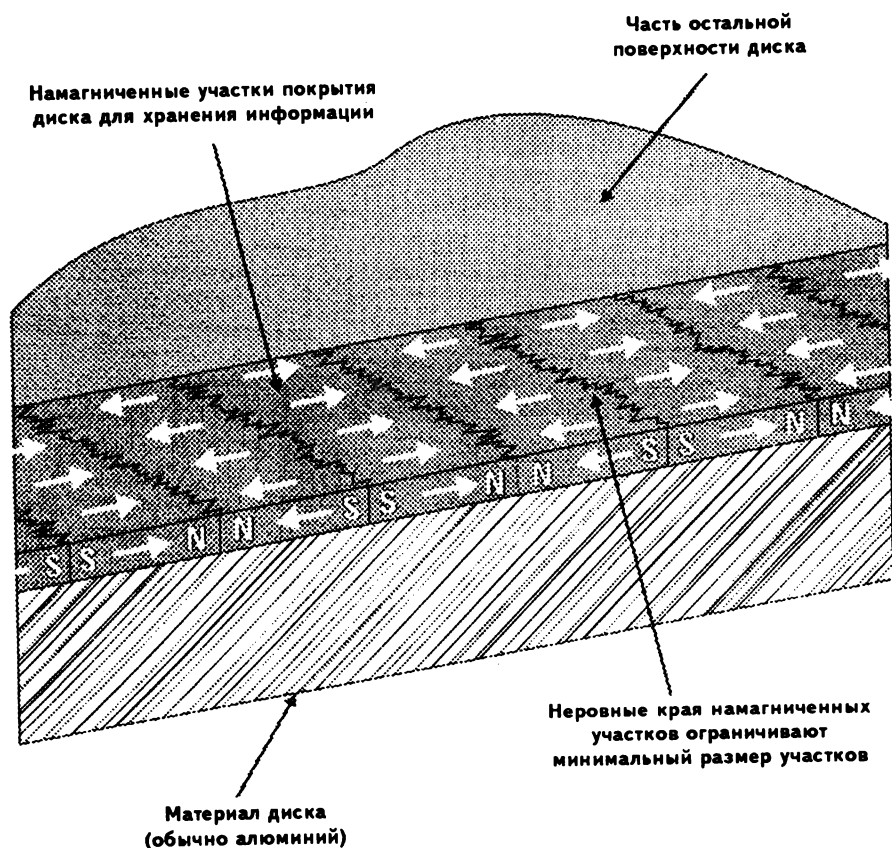


Рис. 4.14. Продольная запись

Самое сильное ограничение на размер записываемого участка оказывает размер промежутка. В нем силовые линии поля выходят из сердечника и проходят через близлежащее покрытие, которое и определяет намагниченный участок покрытия.

Другим очень важным расстоянием является высота полета головки над поверхностью. Магнитное поле вокруг зазора очень быстро рассеивается и уменьшается при удалении от зазора. Поскольку имеет значение расстояние от зазора до намагничивающегося покрытия диска, приходится учитывать воздушный промежуток (фактическая высота полета головки над диском) и толщину защитного или связующего покрытия диска.

Наконец, критичным оказывается и толщина полюсов около зазора. Если полюса в зазоре слишком широкие, значительная часть магнитных силовых линий просто "перескочит" через зазор. Они не проходят через поверхность диска и не намагничивают его. Если же сделать полюса очень узкими, силовые линии так сойдутся в наконечниках полюсов, что материал сердечника достигнет максимальной намагниченности. Такое явление называется насыщением. После достижения насыщения сердечник не сможет больше концентрировать силовые линии.

Эффективная толщина полюсов равна либо их фактической толщине в наконечнике, либо толщине в точке, где материал сердечника насыщается. В местах насыщения силовые линии начинают выходить из наконечника и зазор как бы становится шире, чем он есть на самом деле.

Приведенная информация может показаться совершенно бесполезной, но это не так. Рассмотренная картина объясняет основную причину применения плавающих головок. Можно подумать, что головки плавают для того, чтобы защитить поверхность от стирания. Частью это так, но гораздо важнее защитить головки от износа. Если вначале толщина полюсов сделана равной оптимальному значению, то любой износ сразу же вызовет увеличение эффективной ширины зазора и снижение плотности битов. Подобная ситуация в корне отличается от конструкции головок для вертикальной записи.

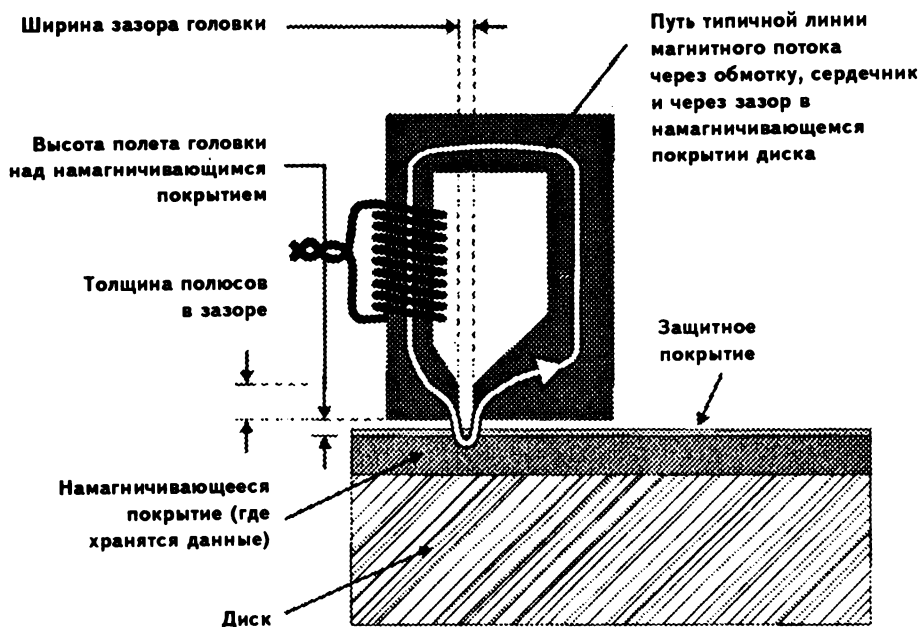
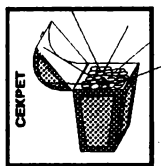


Рис. 4.15. Критические размеры при продольной записи

На рис. 4.14 не показан еще один критический размер головки считывания-записи — ширина полюсов в поперечном направлении дорожки. Именно этот размер определяет ширину дорожки, на которой головка осуществляет запись на поверхности диска. (Из-за рассеяния полей на концах зазора ширина дорожки несколько больше ширины головки. Более того, ради безопасности производители накопителей обычно оставляют между соседними дорожками промежуток минимум в половину ширины дорожки. Таким образом, среднее расстояние между дорожками примерно вдвое шире полюсов.)

МIG-головки. Чтобы сузить эффективный размер зазора и защитить наконечники полюсов от насыщения, производители покрывают внутреннюю поверхность кольца с одной или обеих сторон намагничивающейся металлической пленкой. Такая головка называется MIG-головкой (Metal In Gap — Металл в зазоре) или двойной

MIG-головкой (в случае двух слоев металла в зазоре). Этот прием оказывается полезным потому, что имеются металлические пленки, которые обеспечивают более высокие уровни намагниченности, чем ферритовые материалы, применяемые в обычных кольцевых сердечниках головок.



Интересно, хотя и не столь важно, отметить, что зазор сердечника головки не пустой. Обычно зазор заполнен твердым немагнитным материалом, который не только поддерживает постоянный размер зазора, но и обеспечивает отсутствие острых краев головки, которые могут поцарапать поверхность при "приземлении" головки, которое происходит при каждой остановке вращения дисков.

Производители делают головки (и механизм их поддержания и крепления к приводу) максимально легкими и жесткими. В этом случае блок головок движется быстро и легко, помещая головки в определенное место для каждой позиции привода.

4.3. Вертикальная запись

Наверно, было бы неплохо повысить емкость жестких дисков примерно в 10 раз, сохранив их столь же дешевыми, простыми в изготовлении и надежными, как и современные накопители. Именно этого позволяет достичь вертикальная, или перпендикулярная запись.

Идея вертикальной записи несложна: нужно просто выстроить намагниченные участки так, чтобы крошечные магниты в намагничивающемся покрытии "стояли" вертикально, показывая вовнутрь или наружу поверхности диска. Есть несколько способов для достижения такого результата. Один из самых распространенных и простых способов, который реализуемый при незначительной модернизации технологии, представлен на рис. 4.16.

При сравнении рис. 4.16 и рис. 4.14. видны три существенных различия. Во-первых, белые стрелки, показывающие направление намагниченности каждого участка, направлены по вертикали (что, собственно, и объясняет название этого способа записи). Поверхность дорожки на каждом участке заполнена знаками N и S, обозначающими острие или хвост стрелки в данной точке поверхности.

Во-вторых, на рис. 4.16 появилось еще одно покрытие. Это внутреннее покрытие, сделанное из магнитного, но не намагничивающегося материала. Другими словами, этот слой просто проводит магнитные силовые линии аналогично материалу сердечника головки. При записи этот слой служит важнейшей частью магнитной цепи головки. В остальное время проходящие через этот слой поля образуют замкнутые петли через магниты, образованные в противоположных направлениях (вверх и вниз) двух соседних участков намагничивающегося покрытия поверхности диска.

В-третьих, — это очень ровные границы между соседними участками с противоположной намагниченностью. Суть этого различия можно понять, если представить себе, что на столе лежат два магнита и вы пытаетесь сдвинуть их. Если сближать северные полюса, магниты отталкиваются. При этом возникает неустойчивая ситуация, а именно: создается продольная запись в намагничивающемся покрытии поверхности диска. Но если положить магниты так, чтобы северный полюс одного находился рядом с южным полюсом другого, магниты притягиваются. Такое притяжение образует очень устойчивую ситуацию и вертикальная запись создается в намагничивающемся покрытии поверхности диска.

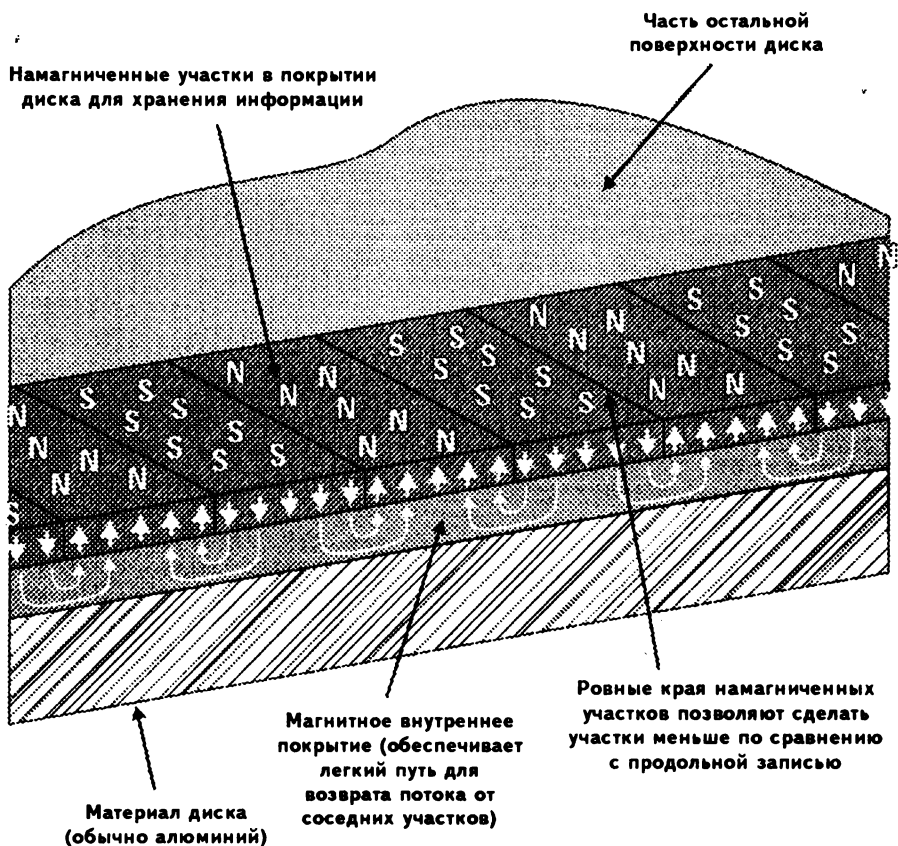


Рис. 4.16. Перпендикулярная (вертикальная) запись



Довольно легко запутаться

Сильное сходство дисков, предназначенных для продольной и вертикальной записи, породило несколько удивительных примеров путаницы на предприятиях по производству дисков. Если вынуть диски и внимательно рассмотреть их, то буквально ничего нельзя заметить, что отличало бы типы дисков. Внутренний слой, используемый при вертикальной записи, полностью закрыт и невидим. Поверхностные слои (намагничивающийся и защитный) в обоих типах дисков совершенно одинаковы.

Единственный надежный способ узнать тип диска — вставить его в накопитель и проверить. Если попытаться записать некоторые данные, используя диск с продольной записью и головку с вертикальной записью, то при считывании данных будут получены очень слабые сигналы. Это объясняется тем, что при отсутствии внутреннего слоя диск не может эффективно возвращать силовые линии от переднего (узкого) полюса головки к возвратному (широкому), поэтому записывается очень слабый сигнал.

Если же попытаться записать данные, используя головку для продольной записи и диск, рассчитанный на вертикальную запись, то при считывании надежного сигнала тоже не будет. В этом случае внутренний слой "захватывает" все силовые линии от намагниченных участков поверхности диска, которые должны были бы возвращаться через воздух над диском. Головка захватывает и возвращает через сердечник и обмотку только те силовые линии, которые проходят через воздух (см. рис. 2.4).

Любая фирма, производящая оба типа накопителей (с продольной и вертикальной записью), должна поддерживать строгий порядок на складе дисков!

В результате при вертикальной записи намагниченные участки можно сделать намного меньшими по сравнению с продольной записью. Конечно, для этого, во-первых, требуются головки, критические размеры которых меньше достижимых по современной технологии, и, во-вторых, высота полета головок должна быть меньше, чем в современных накопителях. Поскольку технология производства головок постоянно совершенствуется, вскоре вертикальная запись будет доминировать в дисковых накопителях.

Каковы же критические размеры головки для вертикальной записи? Чем они отличаются от соответствующих размеров головки для продольной записи?

Критические размеры головок считывания-записи с вертикальной записью. В головке для вертикальной записи критичен размер одного из полюсов, а размер второго не критичен (рис. 4.17). Собственно запись осуществляет очень узкий полюс головки. Широкий полюс служит просто для замыкания магнитных силовых линий. Возвратная головка довольно широка и силовые линии легко проходят значительное расстояние во внутреннем слое, поэтому плотность линий при входе в возвратный полюс настолько мала, что она не оказывает заметного влияния на намагничивающееся покрытие.

Наверняка, вы подумаете, что силовые линии возвращаются через соседний участок далеко не всегда. Но оказывается, что точный путь возврата силовых линий не столь важен, так как они всегда найдут подходящее место. Длина силовых линий не играет роли, важной оказывается только их плотность (число линий на квадратный сантиметр).

Для этого типа головки еще двумя критическими размерами являются высота полета головки и ширина наконечника полюса (она определяет ширину дорожки.) Толщина наконечника возвратного полюса не существенна.

До тех пор, пока тонкий наконечник имеет почти постоянную форму поперечного сечения, его длина не является критическим параметром. Даже при износе наконечник сохраняет неизменными поперечное сечение и форму. Следовательно, головка с вертикальной записью может безопасно работать в контакте с поверхностью диска! Такое невозможно с головкой для продольной записи.



Крах головки — реальна ли проблема?

Многие слышали ужасных историй о крахах магнитных головок жестких дисков. Несколько лет назад крах головок был серьезной проблемой, но теперь она в основном решена.

Если головка сильно ударяет по поверхности диска, поверхность (и, возможно, головка) повреждается. Конечно, если часть намагничивающегося покрытия выбита с диска, данные около этого участка теряются.

Современные конструкции накопителей и технологический процесс почти устранили крах головки как причину гибели жесткого диска. Фактически мягкий "крах" головки происходит при каждом выключении компьютера. После замедления диска поток воздуха, поддерживающий головки, исчезает, головки мягко "приземляются" на поверхность и остаются на ней до полной остановки вращения дисков.

Головки современных дисковых накопителей всегда монтируются на специальном ползунке. Конструкция ползунка позволяет даже слабому потоку воздуха поднять головку над поверхностью и скользить по поверхности без заметного износа покры-

тия, когда скорости вращения недостаточно для подъема головок. Более того, в конструкциях современных накопителей часто предусматривается специальное защитное покрытие, которое служит своеобразной смазкой при его движении ползунка по поверхности диска.

Современные накопители чаще выходят из строя из-за биения шпинделя или отказа электронной схемы, чем от разрушающего краха головки.

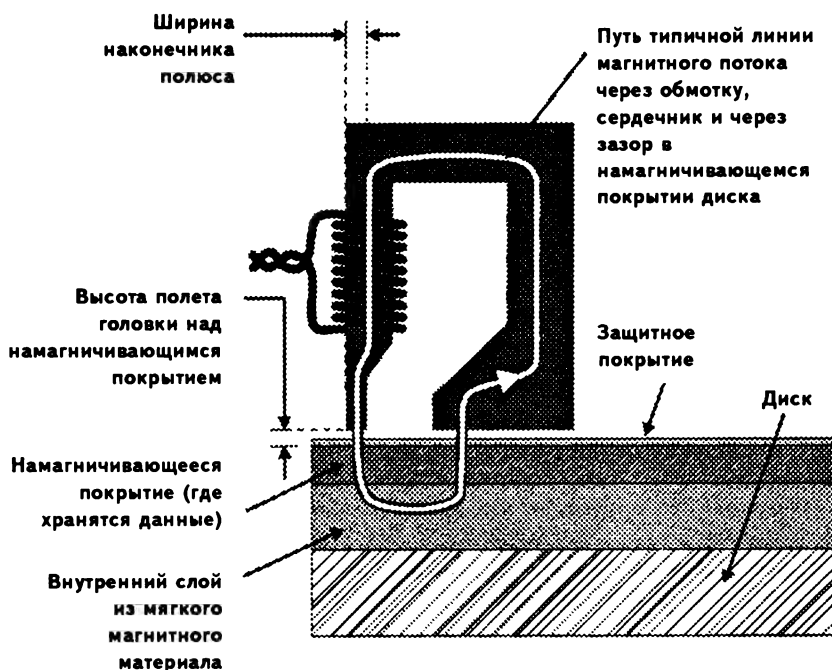


Рис. 4.17. Критические размеры при вертикальной записи

В головке для вертикальной записи умеренный износ наконечника полюса не играет заметной роли. С другой стороны, при продольной записи любой износ, уменьшающий толщину головки вблизи наконечника полюса, сильно снижает производительность головки. Конечно, если будет возможность безопасной работы при контакте головки с поверхностью диска, то высоту полета головки можно уменьшить до толщины защитного покрытия поверхности диска.

Перспектива. Сейчас тонкопленочные головки производятся по технологии интегральных схем, поэтому следует ожидать такого же прогресса в сокращении размеров головок, как и в повышении плотности элементов в интегральных схемах. Значительное уменьшение размеров головок по сравнению с современными определяется только временем. При совмещении сокращения размеров с минимально возможной высотой полета появятся более узкие дорожки и повысится плотность битов на дорожке.

Как далеко пойдет этот прогресс? В ближайшие годы вертикальная контактная запись позволит значительно повысить плотность битов на дорожке. Для этих головок допустим износ, и в будущем появятся накопители, которые допускают или даже требуют периодической замены головок!

Плотность дорожек также значительно увеличится благодаря новым магнитным материалам и конструкциям головок. К сожалению, создать технологию производства позиционеров головок, которые в 10 раз точнее современных, довольно сложно.

Произведение плотности дорожек и плотности битов на дорожке определяет поверхностную плотность записи. В лучших современных накопителях она составляет 80 тыс. битов на квадратный миллиметр. Можно предположить, что в ближайшие годы поверхностная плотность увеличится в пять раз, а в следующем десятилетии — более чем в двадцать раз.

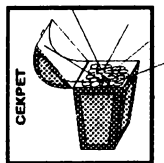
4.4. Что происходит в электронике накопителя

В начале главы 2 было дано общее описание электронной схемной платы, которая подключается к каждому накопителю на жестких дисках. Детальные функции схемной платы зависят от типа накопителя. Для всех накопителей требуются схема управления шпиндельным двигателем, схема позиционирования головок, а также схемы коммутаторов и усилителей сигналов. Кроме того, для некоторых накопителей требуются специальные интерфейсные схемы.

Фирмы — производители накопителей оформляют эти общие элементы несколькими способами; некоторые из них рассмотрены в следующих разделах.

4.4.1. Схема управления шпиндельным двигателем

Во всех накопителях для ПК на плате имеется схема для управления шпиндельным двигателем (который вращает диски с постоянной скоростью). Для поддержания нужной скорости вращения применяются три способа. Простейшее решение состоит в применении синхронного двигателя. Несколько более сложный способ реализован в накопителях с фазовой подстройкой частоты. В наиболее совершенном способе применяется отрицательная обратная связь для регулирования не просто скорости вращения, а фактической скорости, с которой секторы проходят под головками.



В некоторых накопителях применяются синхронные шпиндельные двигатели, которые вращаются синхронно с переменным электрическим сигналом (как в электрических часах, в которых для поддержания точного времени применяется сеть переменного тока). На схемной плате есть генератор с кварцевой стабилизацией частоты (как в кварцевых часах) и усилитель для подачи сигнала генератора на двигатель.

Во втором способе шпиндельный двигатель встроен в схему управления. Он образует часть генератора, который в данном случае называется схемой фазовой подстройки частоты. По-прежнему частотой управляет кварц. Единственное существенное различие состоит в том, что здесь генератор не работает, пока двигатель не вращается.

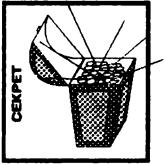
Третий способ похож на второй, но здесь вместо кварца, определяющего частоту генератора, контролируется скорость, с какой секторы проходят под головкой считывания-записи. Это позволяет несколько изменять скорость вращения двигателя, поддерживая постоянную скорость прохождения секторов, а не просто постоянную скорость вращения в оборотах в минуту.

Нетрудно обнаружить сходство этих способов и рассмотренных ранее способов позиционирования головок. Действительно, первый способ аналогичен позиционерам головок с шаговым двигателем и счислением пути. Второй способ похож на внешнюю сервосистему, в которой применяется обратная связь, но не от фактических данных на диске. Наконец, третий способ похож на наиболее совершенные позиционеры, в которых используется информация о фактических данных (или о связанных с ними сервопакетах) на дисках, которые в данный момент проходят под головками.

4.4.2. Схема управления позиционированием головок

Функции встроенной в накопитель электронная схема для управления приводом позиционирования головок зависят от двух факторов. Первый фактор — это тип используемого в накопителе механизма позиционирования (шаговый двигатель или сервосистема). Второй фактор — это тип интерфейса накопителя: (ST412/ST506, ESDI, SCSI, IDE/ATA или плата жесткого диска).

Обычно не обязательно (но лучше!) знать, какой тип схемы позиционирования головок имеется в вашем накопителе. Если применяется позиционер с сервосистемой, его можно вывести из строя, применив неподходящую программу парковки головок. При выполнении такой программы вы услышите "банг-банг-банг" до тех пор, пока не выключите компьютер. Лучше всего в этой ситуации побыстрее его выключить, чтобы не испортить накопитель.



Контроллеры ST412/ST506 (применяемые в накопителях MFM и RLL) просто командуют накопителю сделать шаги внутрь или наружу на некоторое число цилиндров. В отличие от контроллеров ESDI и SCSI они не определяют номер целевого цилиндра.

Для контроллеров ST412/ST506 наиболее подходят накопители с шаговым двигателем. Такие двигатели делают шаг при получении импульса тока и не знают, на каком цилиндре находятся головки.

На плате накопителя имеется буфер, который может хранить несколько команд на произведение шага, но слежения за абсолютным номером цилиндра нет. Если, например, скомандовать накопителю с шаговым двигателем перейти на 4000 цилиндров, он послушно перейдет на максимальный цилиндр, а затем несколько тысяч раз будет толкать головки в некоторый физический упор, не замечая, что фактически они не движутся.

В накопителе с сервосистемой ситуация совершенно иная. Если по-прежнему говорить о накопителе с интерфейсом ST412/ST506, контроллер командует ему сделать шаг вовнутрь или наружу на один цилиндр. И вновь электроника накопителя сохраняет несколько таких в буфере до тех пор, пока не придет время выполнять их.

В отличие от накопителей с шаговыми двигателями накопители с сервосистемой должны следить за тем, куда (на какой номер цилиндра) они должны перемещать головки. Такая информация необходима потому, что единственный способ правильного позиционирования головок заключается в том, чтобы считать специальные сервосигналы, посмотреть, о каком номере цилиндра сообщают сигналы, а затем сравнить его с нужным номером цилиндра.

Если скомандовать такому накопителю перейти на 4000 цилиндров, он вычислит огромный номер отыскиваемого цилиндра, а затем подаст в звуковую катушку ток соответствующего направления и будет проверять по сервосигналам, позиционированы ли головки на нужную позицию. Ни в одном дисковом накопителе для персональных компьютеров нет такого количества цилиндров, поэтому позиционер не найдет того, что нужно. А что произойдет потом?

Схема достаточна "умна", чтобы заметить, что она не может перейти в нужную позицию даже при подаче в звуковую катушку мощного продолжительного тока. Заметив это, она попытается повторить операцию. Прежде всего она изменит направление тока и будет поддерживать его до восприятия цилиндра 0 на внешнем краю диска. После этого она вновь изменит ток и возвратится к поиску заданного цилиндра. Конечно, и в этом случае обеспечена неудача.

Накопитель заикнется на этих операциях навсегда, либо до своего разрушения. К сожалению, некоторые "дубовые" программы парковки головок заставляют накопитель действовать именно таким образом.

Поскольку контроллеры ST412/ST506 (применяемые с накопителями MFM и RLL) командуют накопителю сделать шаг внутрь или наружу на один цилиндр несколько раз, они хорошо согласуются с позиционером головок на базе шагового дви-

гателя. Однако в более производительных накопителях MFM и RLL требуется дополнительная схема для работы позиционера головок с сервосистемой.

Все накопители ESDI должны иметь такую схему, так как их контроллеры обычно посылают в накопитель команду перехода на конкретный цилиндр по его номеру. Такие накопители относятся к разряду высокопроизводительных и имеют позиционеры головок с сервосистемой, поэтому в них обязательно должна быть встроена возможность позиционирования по абсолютному номеру цилиндра.

Накопители IDE/ATA и платы жестких дисков, по существу, представляют собой комбинацию накопителя (обычно RLL или ERL) и платы контроллера. В них также имеется схема позиционирования по абсолютному номеру цилиндра.

Накопители SCSI (и некоторые самые современные накопители IDE) воспринимают команды поиска логического адреса блока (Logical Block Address LBA). В накопитель приходится встраивать еще больше электронных схем, так как он должен вначале вычислить, какому физическому адресу (головка, цилиндр и физический номер сектора) соответствует LBA, а затем с помощью сервосистемы перейти в нужное место.

4.4.3. Коммутаторы и усилители

Наконец, на электронной плате накопителя должны быть схемы, подключенные к головкам считывания-записи. В эти схемы входят набор коммутаторов для выбора нужной головки и задания режима считывания или записи; усилитель записи, который получает сигналы от контроллера и превращает их в достаточно сильные для записи информации на поверхности диска; усилитель считывания, который растягивает и фильтрует сигналы от головок при считывании данных (рис. 4.18). Помимо этих схем в накопителях ESDI, SCSI и IDE/ATA, а также на платах жестких дисков имеется схема сепаратора (разделителя) сигналов синхронизации и данных.

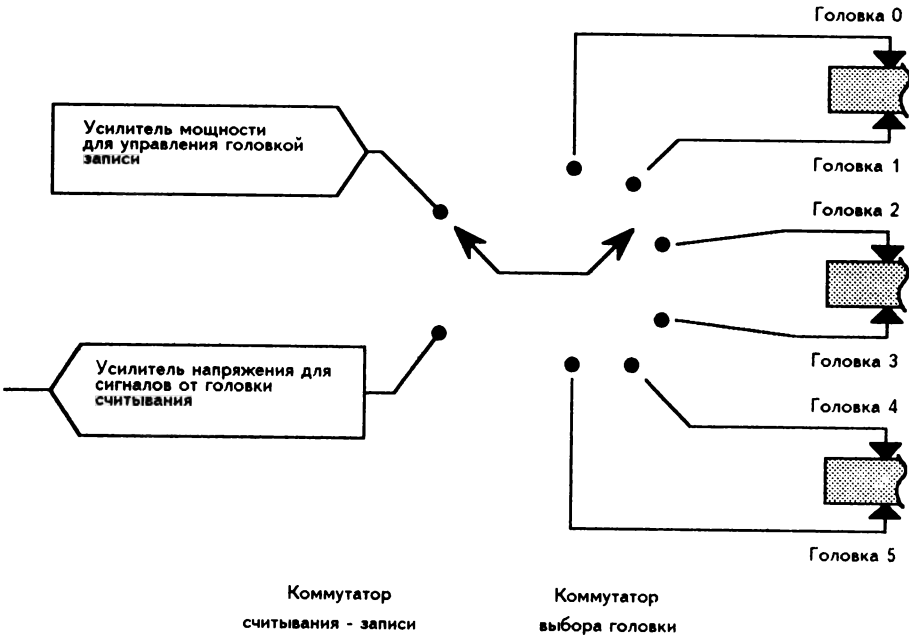


Рис. 4.18. Блок-схема коммутаторов головок считывания записи

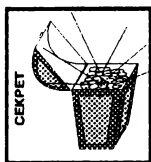


Схема сепаратора синхронизации и данных воспринимает аналоговые сигналы от головок (после их усиления и фильтрации) и преобразует эти импульсы в двоичные цифровые данные. Если в накопителе применяется кодирование MFM, схема определяет, какие импульсы являются сигналами синхронизации, а какие – сигналами данных, а затем разделяет их (отсюда и название схемы). При использовании кодирования RLL задача схемы несколько усложняется, но она по-прежнему называется сепаратором синхронизации и данных.

Для накопителей ESDI это все, но в накопителях SCSI и IDE, а также на платах жестких дисков имеется еще один элемент.

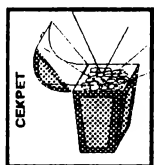
4.4.4. Интерфейсная электроника

Накопители IDE/ATA и платы жестких дисков, по существу, объединяют законченный контроллер и жесткий диск. Базовая конструкция накопителей аналогична накопителям RLL. В любом случае в этих накопителях должна быть схема, которая, помимо выполнения ранее рассмотренных задач, связана с шиной ввода-вывода компьютера.

Накопители SCSI содержат полную схему контроллера и накопитель, а также схему интерфейса SCSI, которая придает им достаточный "интеллект" для общения с главным адаптером SCSI и управления шиной SCSI.

4.5. Что происходит в электронике контроллера

Накопители MFM, RLL и ESDI подключаются к платам контроллеров, которые вставляются в разъемы шины ввода-вывода ПК. Платы контроллеров воспринимают команды от компьютера и преобразуют их на "язык", понятный накопителям, а затем реализуют обратное взаимодействие накопителей с компьютером.



Платы контроллеров упаковывают поступающие от компьютера данные, добавляя к ним байты кода обнаружения и исправления ошибок ECC. В случае интерфейсных контроллеров ST412/ST506 (для накопителей MFM и RLL) они должны кодировать байты данных и ECC в соответствии с применяемым способом кодирования данных (MFM или RLL). Накопители ESDI реализуют эти функции самостоятельно.

Контроллер ST412/ST506 должен разделить сигналы синхронизации и данных, т.е. выделить фактические двоичные данные из последовательности импульсов от головок накопителя. Системы ESDI осуществляют такое разделение в накопителе. Любой из этих контроллеров (ST412/ST506 или ESDI) должен затем вычислить ECC и сравнить результат со считанным, при расхождении попытаться повторно считать данные и передать данные в компьютер (с удаленными байтами ECC и, возможно, исправленными благодаря ECC ошибками). Наконец, контроллер должен послать в компьютер сообщение о степени доверия переданным данным.

Самое важное, что нужно знать о работе плат контроллеров диска, состоит в том, что они функционируют уникальным образом. Такая уникальность означает, во-первых, что для накопителя необходимо выбирать соответствующий контроллер. Накопители MFM обычно не работают при подключении к контроллерам RLL (даже несмотря на то, что в обоих применяется интерфейс ST412/ST506) и никакой контроллер нельзя использовать с накопителем ESDI или наоборот.

Уже отмечалось, что невозможно форматировать накопитель и заполнить его данными одним контроллером и ожидать, что другой контроллер (даже того же типа) сможет считать данные. Только два контроллера одной модели с одинаковым номером ревизии, выпущенные фирмой примерно в одно и то же время, могут обеспечить возможность замены.

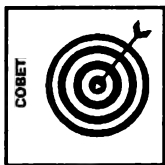
Многие полагают, что контроллер SCSI (который точнее называть главным адаптером SCSI) совершенно отличен от других. Контроллер диска, как таковой, находится в накопителе SCSI вместе со схемой подчиненного интерфейса SCSI.

Конечно, для накопителей с контроллерами, смонтированными непосредственно в самом накопителе (сюда относятся все накопители IDE/ATA, платы жестких дисков и накопители SCSI), проблемы неподходящего контроллера не существует.

4.6. Рекалибровка и странные ночные звуки

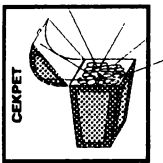
Иногда компьютер издает пугающие шаркающие звуки. Понятно, что они возникают в накопителе, но что там происходит и почему? Нужно ли вам беспокоиться?

Некоторые шумные накопители всегда издают странные звуки. Если они работают хорошо, можно полагать, что они будут продолжать работать хорошо примерно столько же времени, сколько и другой аналогичный накопитель. Вы можете спокойно игнорировать странные звуки таких накопителей. Беспокоиться нужно тогда, когда накопитель работал бесшумно и вдруг начинает шуметь. Конечно, такие звуки пугают людей.



Естественно, вас испугают странные звуки, когда вы замечаете, что их источником является накопитель и что они не похожи на обычные. Многие дисковые накопители издают особенные звуки. Изучите обычные "хорошие" звуки и прислушайтесь к любым звукам, отличающимся от обычных.

Есть две основные причины появления странных звуков, но в обоих случаях накопитель осуществляет рекалибровку. Звуки означают, что накопитель "запутался" в поступивших командах. Если некоторые секторы находятся в таких местах диска, где невозможно надежно хранить информацию, возможно, достаточно отметить такие секторы как плохие (дефектные), а затем избегать записывать в них данные. Конечно, если в такие секторы уже записаны некоторые данные, такого приема недостаточно.



Первая основная причина звуков рекалибровки состоит в том, что накопитель потерял "чувство реальности", в частности не знает, на каком цилиндре находятся его головки. Возможно, что это накопитель с сервосистемой и сигналы сервоголовок сообщают электронике накопителя о нахождении не на той дорожке, на которой они должны быть. Возможно, это накопитель с шаговым двигателем и сигналы, считанные от заголовков секторов на дорожке данных, сообщают об этом же. Такая ситуация может возникнуть даже тогда, когда накопитель не считывает и не записывает информации (т.е. вы наблюдаете стимул DOS или прикладная программа ожидает вашего ввода).

Чтобы определить, что с дорожкой, электроника накопителя сначала возвращается на внешнюю дорожку (цилиндр 0) и начинает действовать оттуда. Такой прием наиболее подходит для накопителя с шаговым двигателем, в котором самый очевидный способ найти цилиндр, например номер 12, состоит в том, чтобы перейти на цилиндр номер 0 и сделать 12 шагов к шпинделю. Этот же прием используется и в накопителях с сервосистемой для подтверждения поиска цилиндра.

Вторая причина появления странных звуков связана с попытками накопителя считать плохой сектор. Возможно, он не может найти заголовок сектора с правильным адресом или значение CRC для заголовка сектора не согласуется с содержимым заголовка. И в этом случае накопитель не уверен в правильном цилиндре, поэтому он попытается повторно считать сектор, возвращаясь к началу диска (цилиндру 0) и переходя на нужную дорожку для поиска заданного сектора. Накопитель использует этот способ для того, чтобы убедиться, что он не может считать сектор просто из-за обращения к неправильному месту.

Некоторые современные накопители IDE пытаются вновь и вновь считать плохой сектор, приближаясь к цилиндру сначала с одной стороны, а затем с другой, или применяя другие способы небольшого "дрожания" головок и пытаясь все же считать сектор.

Можно услышать звуки и при первой подготовке жесткого диска. При форматировании низкого и высокого уровня компьютер тщательно проверяет каждый сектор диска для надежного хранения и считывания информации. При обнаружении дефектного пятна программа заставляет контроллер повторять проверку. Только после многих неудач это место отмечается как дефектное, и DOS никогда не будет хранить в нем информацию.

Вы наверняка услышите звуки при выполнении программы тестирования поверхности диска. Они возникают потому, что программа тестирует даже те места, которые ранее были объявлены плохими, на случай пропадания дефекта.

Самое плохое — услышать звуки, когда обычная программа DOS пытается считать или записать информацию на диск. Именно в это время вы можете потерять часть данных, если оказались невнимательным (и, конечно, невезучим) человеком.

Если вы услышите звуки при выполнении обычной программы DOS, необходимо сразу же зарезервировать активные файлы на гибкий диск или ленту. Если DOS не проводит резервирования, сообщая о невозможности найти сектор или об ошибке при считывании, необходимо провести резервирование с помощью специальной программы, например SpinRite.

Заключение

- Шаговые двигатели являются простыми цифровыми приводами. Они применяются в дешевых дисковых накопителях малой емкости для позиционирования головок по способу счисления пути.
- В более дорогих дисковых накопителях применяется отрицательная обратная связь для постоянной коррекции позиции головок и выравнивания их по центрам дорожек. Такие приводы также называются сервосистемами.
- Двигатели со звуковой катушкой могут выталкивать или втягивать с разным усилием и их можно использовать для разработки очень быстро реагирующих механизмов позиционирования головок. Они не могут "навести" головки в нужное место без помощи сервосистемы, которая контролирует текущее положение головок.
- В позиционерах головок накопителей на жестких дисках применяются несколько типов сервосистем. Наиболее популярны сейчас встроенные и гибридные сервосистемы. В последней объединены достоинства специального диска и встроенной сервосистемы.
- Большинство головок считывания-записи являются разновидностями индуктивной кольцевой головки и рассчитаны на продольную запись. Новейшие технологии позволяют создать головки, которые записывают и считывают больше битов на единицу длины дорожки и работают с более узкими дорожками.

- Новым способом хранения данных на намагничивающейся поверхности диска является вертикальная (перпендикулярная) запись. Потенциально он обещает значительное повышение плотности хранения данных. Накопители на жестких дисках с вертикальной записью только появляются на рынке, но, по-видимому, их ожидает быстрое развитие в ближайшем будущем.
- Электроника, смонтированная на накопителе, или на отдельной плате контроллера, реализует несколько функций: управление скоростью вращения дисков; перемещение блока головок на нужный цилиндр; объединение сигналов данных и синхронизации (кодирование) и добавление байтов ECC до сохранения данных на диске.
- Другие электронные схемы решают противоположные двум последним функциям задачи при считывании данных; они разделяют сигналы данных и синхронизации и при необходимости восстанавливают искаженные данные с привлечением записанной информации ECC.
- В накопителях с различным интерфейсом (ST412/ST506, ESDI, IDE/ATA и SCSI) и платах жестких дисков (которые являются разновидностью накопителя IDE/ATA) функции распределены по-разному, но во всех накопителях эти функции должны быть реализованы.
- Одной из причин отказа накопителя является "забывание" им цилиндра, на который позиционирован блок головок. При возникновении такой ситуации накопитель осуществляет рекалибровку, перемещая головки на цилиндр 0, а затем на нужный цилиндр. Если ваш накопитель начинает проводить рекалибровку часто и хаотически, — это возможный признак скорой его гибели. Обратите внимание на это предупреждение! Второго вы просто можете не получить.

В следующей главе показано, как заставить диск работать быстрее. Кэширование диска может быть довольно сложным, но не бойтесь применять этот очень полезный прием. В следующей главе затрагивается и вопрос о кэшировании памяти, т.е. еще об одном приеме повышения производительности ПК.)

Глава 5. Кэширование диска

В этой главе вы узнаете

- Что такое кэширование диска
- Отличие и сходство кэширования диска и кэширования памяти
- Преимущества пользователей ПК от кэширования диска,
- Сходство и отличие шести методов кэширования диска
- Когда применять дисковый контроллер с аппаратным кэшированием
- Как правильно задать число буферов BUFFERS в файле конфигурирования CONFIG.SYS.

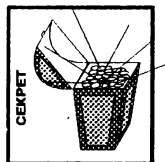
5.1. Назначение дискового кэша

Дисковая кэш-память (или просто кэш) в самом общем виде представляет хранилище данных, считанных с диска или подлежащих записи на диск. Главная цель дискового кэша — ускорить операции считывания и записи. В этом отношении дисковый кэш напоминает кэш основной памяти, который применяется в некоторых компьютерах для хранения информации и находится между микросхемой центрального процессора (ЦП) и основным ЗУПВ. Аналогичные по назначению, эти два типа кэша имеют и существенные различия (рис. 5.1). Далее рассматривается только дисковый кэш.

Различие между кэшированием диска и памяти состоит в следующем. Дисковый кэш — это промежуточное хранилище (ЗУПВ) данных, передаваемых между основным ЗУПВ ПК и дисковым накопителем. Кэш память — это временное хранилище (очень быстрое ЗУПВ) данных, находящееся между ЦП (микросхема 8086/8088, 80286, 80386 или i486) и основным ЗУПВ. Персональный компьютер может иметь оба вида кэша (в самых быстрых компьютерах), один вид или ни одного.

Любой компьютер — это, в сущности, преобразователь информации. Чтобы ПК обрабатывал информацию, в его основном ЗУПВ должны находиться программа обработки и сами обрабатываемые данные. При выключении питания вся находившаяся в ЗУПВ информация теряется. Следовательно, необходимо хранить программы и информацию на жестком диске или некотором другом энергонезависимом носителе. Конечно, поскольку обрабатывается только информация, находящаяся в ЗУПВ, требуется загружать все с диска в ЗУПВ и обратно.

DOS считывает информацию с диска и записывает ее на диск с привлечением так называемых дисковых буферов DOS (о них мы поговорим позже), т.е. специальных областей ЗУПВ, предназначенных только для таких операций. Прежде всего познанимся с аппаратными и программными аспектами кэширования диска. При этом важно помнить, что ПК запрашивает информацию с диска и посылает ее в дисковый накопитель только целым сектором.



MS DOS и PC DOS 3.3 могут выдать команду контроллеру жесткого диска считать или записать одновременно несколько секторов. Однако в большинстве программ такая возможность не используется. Более того, некоторые контроллеры не могут правильно обработать такие запросы при возникновении даже исправимой ошибки.

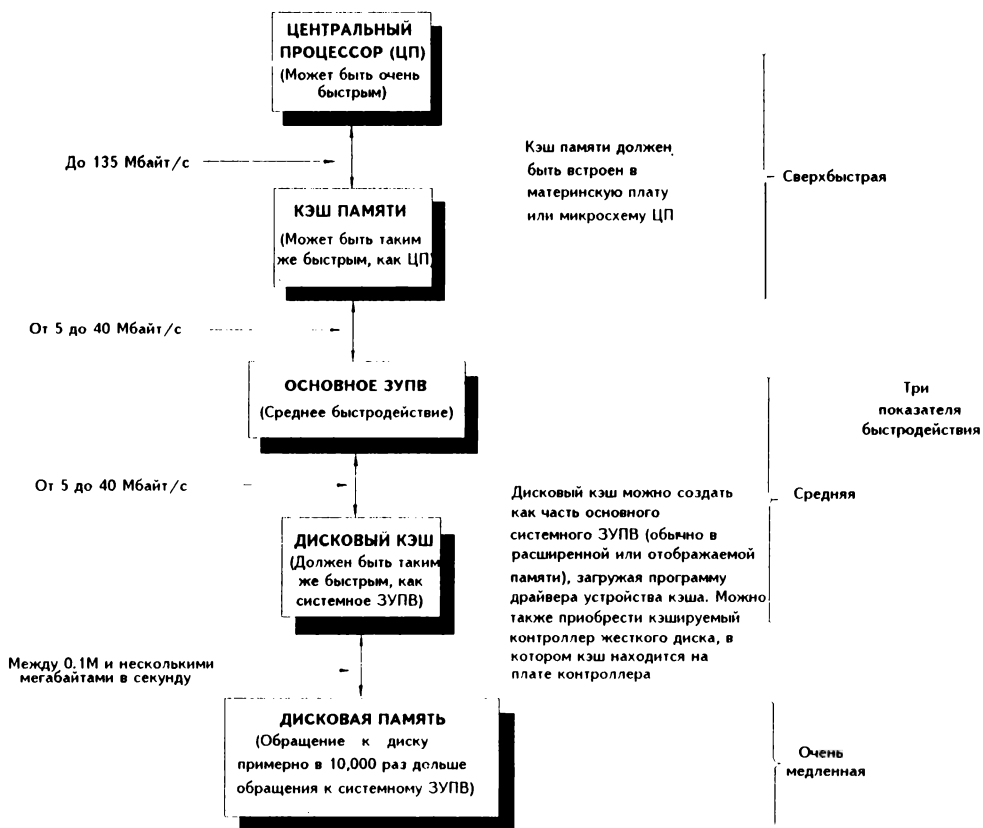


Рис.5.1. Кэш памяти в сравнении с дисковым кэшем; показаны типичные быстродействия и скорости передачи данных между ними

Нужно помнить и о том, что дисковый накопитель имеет движущиеся части, а это означает, что накопитель оперирует данными медленнее, чем ЦП обменивается ими с ЗУПВ. (Приближенный коэффициент быстродействия составляет 1000:1 и может доходить до нескольких десятков тысяч к одному.)

5.2. Виды кэша

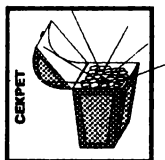
Аппаратный дисковый кэш представляет блок ЗУПВ в контроллере диска, где можно временно хранить информацию до передачи ее в накопитель или в компьютер. Программный кэш — это программа, которая использует часть основного ЗУПВ компьютера для имитации аппаратного дискового кэша.

Нужно отчетливо понимать, что аппаратный и программный кэши должны как-то управлять использованием их памяти. В случае аппаратного кэша эту функцию реализует отдельный небольшой компьютер в составе контроллера диска, а в программном кэше для этого привлекается резидентная программа, находящаяся в основной памяти компьютера.

В менеджерах дискового кэша (аппаратного или программного) применяется несколько стратегий управления; наиболее распространенные рассматриваются в следующих разделах.

5.2.1. Буфер дорожки

Буфер дорожки — это блок ЗУПВ, емкость которого равна емкости целой дорожки диска. Когда DOS запрашивает с дискового накопителя сектор информации, контроллер кэша считывает не только запрошенный сектор, но и остальную часть дорожки, "надеясь" на то, что DOS запросит и следующий сектор с этой же дорожки (что чаще всего и происходит). В такой ситуации контроллер сразу же передает данные, не ожидая, когда нужный сектор подойдет к головке.



Наличие буфера дорожки в контроллере диска (как разновидности аппаратного кэша) может существенно смягчить влияние неправильного коэффициента чередования накопителя. Однако для достижения оптимальной производительности с некоторыми контроллерами дисков необходимо согласовать коэффициент чередования и буфер дорожки.

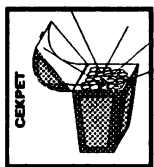
Поскольку в типичном накопителе с кодированием MFM есть 17 секторов по 512 байт, для буфера дорожки требуется ЗУПВ всего на 8 Кбайт. В накопителе с кодированием RLL емкость буфера составляет около 13 К байт, а в накопителях с интерфейсом ESDI или SCSI может достигать до 38 Кбайт. Сейчас ЗУПВ такой емкости относительно дешево, поэтому буфер дорожки применяется в большинстве контроллеров диска. В накопителях с интерфейсом IDE/AT и SCSI, в которых контроллер встроен в накопитель, всегда имеется буфер дорожки.

5.2.2. Кэш считывания

В этой разновидности дискового кэша применяется ЗУПВ большей емкости, и оно используется более эффективно. В таком кэше емкость ЗУПВ достаточна для хранения данных с нескольких дорожек и обычно варьируется от 32 К до 16 Мбайт.

Когда компьютер запрашивает считывание сектора информации с диска, запрос направляется в контроллер кэша. Контроллер прежде всего проверяет, не хранится ли запрошенная информация в кэше. Если она находится там, контроллер возвращает запрошенную информацию в ЦП и на этом его работа закончена. Если же информации нет, контроллер кэша получает данные с диска и передает их в ЦП; кроме того, он сохраняет копию данных и фиксирует, откуда с диска поступили данные в

кэш. Этот процесс продолжается до заполнения памяти дискового кэша. По мере заполнения кэша считывания вероятность того, что DOS запросит секторы, уже находящиеся в кэше, заметно возрастает, а это повышает эффективное быстродействие дискового накопителя.



Обнаружение отыскиваемых данных в кэше называется попаданием, а коэффициент попаданий служит мерой эффективности кэша. При благоприятных обстоятельствах коэффициент попаданий в большом кэше может достигать и более 90%. При попадании кэш возвращает данные почти без задержки, но в случае промаха требуется фактическое обращение к диску. При коэффициенте попаданий 90% (и коэффициенте промахов соответственно 10%) накопитель работает как бы в десять раз быстрее обычного.

Конечно, такие благоприятные обстоятельства складываются не все время и эффективность кэша снижается. Но если кэш хотя бы удваивает скорость накопителя, и лишь иногда повышает его скорость в десять раз, то и тогда применение кэша оказывается оправданным.

Когда у контроллера запрашивается сектор, содержимого которого в кэше нет, контроллер должен обратиться к диску и считать сектор. Если в этот момент времени память кэша полностью заполнена, контроллер уничтожает наиболее давно не используемый элемент в кэше и заменяет его содержимым сектора или дорожки, которые только что считаны с диска. (Вы, как пользователь, совершенно не касаетесь стратегии замены в кэше, так она выбрана производителем.)

Разумеется, сделать "правильный" выбор заменяемого элемента кэша нелегко. Наиболее широко применяется выбор того элемента кэша, к которому дольше всего не было обращений, но этот способ не единственный. Удаляемый элемент не обязательно является тем, который помещен в память кэша давнее остальных, т.е. не самый "старый". Если к такому "старому" элементу не было обращения недавнее, чем последнее обращение к некоторому другому элементу, лучше всего было бы удалить этот самый старый элемент, а не тот, к которому дольше всего не было обращений; возможно, этот элемент вскоре понадобится.

В некоторых контроллерах кэша удаляется самый старый элемент, а не тот, к которому дольше всего не было обращений, но этот способ не получил широкого распространения. Во многих контроллерах не ведется точной регистрации обращений к каждому элементу кэша; в лучшем случае, они реализуют приближенный вариант способа замены элемента, к которому дольше всего не было обращений.

Иногда встречается термин "Вспомогательный кэш считывания". Здесь при запросе сектора с диска проводится параллельное обращение к кэшу и диску и в ЦП информация возвращается оттуда, где она быстрее найдена. Диск намного медленнее ЗУПВ, и отличие такой организации от стандартного обращения к диску только при отсутствии информации в кэше оказывается незначительным. При кэшировании памяти различие в скорости основной памяти от памяти дискового кэша оказывается не столь существенным, поэтому отличие способов может оказаться важным.

Еще одна разновидность кэша — кэш с опережающим считыванием, в котором с диска всегда считывается целая дорожка информации. В этом варианте объединяются некоторые достоинства буфера дорожки и стандартного кэша считывания и он применяется в аппаратном дисковом кэше, реализованном в контроллере диска. Как и в случае буфера дорожки, кэш с опережающим считыванием не влияет на скорость записи; он ускоряет только считывание с диска.

5.2.3. Кэш со сквозной записью

Кэш со сквозной записью ускоряет запись на диск за счет выделения части памяти кэша для хранения последней записанной информации. Остальная часть памяти кэша применяется для кэширования считывания и действует так же, как только что рассмотренные кэши считывания.

Различие между кэшем со сквозной записью и кэшем считывания проявляется, когда ЦП требуется записать некоторую информацию на диск. Кэш со сквозной записью перехватывает эти запросы, а также запросы считывания.

Прежде всего контроллер кэша проверяет, не находятся ли посылаемые сейчас на диск данные в кэше и не записаны ли они на то же самое место диска. При положительном результате проверки контроллер сообщает ЦП об окончании записи информации, даже если он фактически не проводил никакой записи. В этом случае ЦП продолжает свои действия без задержки на фактическую операцию записи.

Если посылаемая на диск информация раньше на диск не записывалась (и ее, следовательно, нет в кэше), контроллер должен сделать фактическую запись. Одновременно с этим контроллер помещает копию данных (и место, куда она записывается) в кэш в надежде, что вскоре ему придется повторить такую запись. В такой ситуации ЦП должен ожидать окончания записи на диск, но если в будущем ему придется повторить запись той же информации (и если копия ее в кэше не уничтожена), то задержки на запись не будет.

Кэш со сквозной записью (и считыванием) получил наибольшее распространение и для большинства людей это наиболее подходящий вариант. Имеются и другие способы реализации кэша, которые подходят в конкретных ситуациях.

5.2.4. Улучшенный кэш со сквозной записью

Под этой разновидностью кэша понимают кэширующий контроллер диска, наподобие уже рассмотренных, но либо с ЗУПВ большей емкости, либо с "лучшим" методом управления кэшем.

Одно из улучшений стандартного кэша со сквозной записью состоит в сортировке всех ожидающих записей так, чтобы накопитель смог выполнить команды за минимальное время. Никакие запросы записи на диск искусственно не задерживаются, но, поскольку ЦП может формировать запросы записи (и подавать данные в кэш) намного быстрее, чем их может воспринять диск, может образоваться очередь запросов, ожидающих обработки диском. Если теперь рассортировать запросы в соответствии с местонахождением на диске, накопитель удовлетворит все запросы за минимальное время.

Этот вариант напоминает рассматриваемый далее элеваторный кэш (и имеет значительную часть его достоинств), но без опасности отложенной записи. (Когда кэш откладывает запись информации на диск более чем на незначительный временной интервал, вы рискуете потерять часть данных, а в худшем случае — разрушить целостность всей базы данных (подробнее в следующем пункте).

В популярной программе Super PC-Kwik сортировка операций записи называется PowerWrite. Фирма PC-Kwik специально подчеркивает, что PowerWrite не является вариантом отложенной записи, а более эффективным вариантом сквозной записи.

Еще одно тонкое, но существенное улучшение стратегии записи, применяется в некоторых коммерческих программах кэширования. Здесь не только сортируются ожидающие операции записи, но они откладываются (конечно, на короткое время) до тех пор, пока система не будет занята считыванием с диска или пока не истечет заданный временной интервал (небольшая доля секунды). Такая задержка часто позволяет использовать диск для считывания нужной ЦП информации, но и не задерживает запись информации на такой интервал, при котором велика вероятность потери данных.

5.2.5. Кэш с отложенной записью

Это совершенно другая разновидность кэширующего контроллера диска. Такой кэш отличается от кэша со сквозной записью одним существенным аспектом: кэш с отложенной записью не записывает информацию на диск при ее получении; запись проводится позже при готовности контроллера кэша и задержка может составлять несколько секунд.

Кэш сразу же сообщает ЦП о том, что операция записи закончена, поэтому компьютер может продолжать свои действия намного быстрее по сравнению с ожиданием фактической записи данных на диск, и это, конечно, хорошо. Однако иногда данные вообще не достигают накопителя, а это, без сомнения, очень плохо.

Для некоторых пользователей кэш с отложенной записью оказывается идеальным. Но, с другой стороны, эта разновидность кэша может быть настолько опасной, что другие пользователи не хотят и слышать о ней.

Когда ЦП запрашивает информацию с диска, такой контроллер действует так же, как и другие кэширующие контроллеры с ЗУПВ большой емкости. Он проверяет, не считан ли уже запрошенный ЦП сектор. Если сектор уже в контроллере, он сразу же передается в ЦП. Если же сектора нет, контроллер считывает либо один нужный сектор, либо всю дорожку с нужным сектором, помещая считанные данные в кэш и надеясь, что они вскоре понадобятся процессору.

Различие проявляется, когда ЦП просит кэш записать информацию на диск. Кэш с отложенной записью (как подчеркивает ее название) просто воспринимает информацию и помещает ее в свою память — на некоторое время. Запись на диск проводится позже.

Такой вариант обеспечивает максимальную производительность по сравнению со всеми другими кэширующими контроллерами дисков. Он ускоряет даже первую операцию записи в то время, как остальные кэширующие контроллеры не улучшают производительность до тех пор, пока ЦП не запрашивает повторного считывания или записи того, что было недавно считано или записано (и, разумеется, сохранено).

Контроллеры данного типа оказываются самыми быстрыми! Однако у них есть и существенный недостаток: возможность потери данных при отказе питания, когда кэш еще не записал данные из ЗУПВ на диск, или при "зависании" компьютера из-за сбоев в выполняемой программе. Если подлежащие записи на диск данные были платёжными документами, вас ожидают серьезные неприятности.

Есть несколько способов заставить контроллер очистить буферы (т.е. записать информацию из кэша на диск). Иногда очистка буферов проводится автоматически при выходе из приложения и возврате в DOS. В ряде случаев для инициирования очистки приходится нажимать специальную комбинацию клавиш. В противном случае контроллер ожидает заполнения буфера или освобождения накопителя в течение заданного временного интервала до записи на диск кэшированной информации.

В последних версиях MS DOS и Windows фирма Microsoft предоставляет программу кэширования диска SMARTDrive. Эта программа была не особенно хорошей, но с версии 4.0, (которая начала поставляться с Windows 3.1), программу SMARTDrive значительно улучшили.

Основной недостаток SMARTDrive 4.0 состоял в том, что она либо кэшировала накопитель (считывание и отложенная запись), либо не кэшировала совсем. Более тонкого управления программой просто не было, а для многих пользователей ни один из этих вариантов не подходил. Начиная с версии 4.1, которая поставляется с DOS 6, программа SMARTDrive стала конкурентоспособной и сейчас обеспечивает такие же гибкость и производительность, как и лучшие программы кэширования диска.

При выборе кэша с отложенной записью настоятельно рекомендуется приобрести и применять бесперебойный источник питания (Uninterruptible Power Supply — UPS). По крайней мере он поможет предотвратить потерю данных в кэше (еще не записанных на диск) из-за аварийного выключения питания.

При использовании программного кэша с отложенной записью необходимо помнить и о зависании компьютера. Если компьютер подвержен частым зависаниям (например, из-за ненадежных компонентов или при работе с несовместимым набором программ), такую программу применять не рекомендуется.

5.2.6. Элеваторный кэш

Элеваторный кэш — наиболее изящная разновидность кэша с отложенной записью. Он не только максимально ускоряет дисковые операции, но и позволяет накопителю работать менее напряженно.

Контроллер с элеваторным кэшированием организует все накопленные в кэше (и пока не записанные на диск) данные в порядке цилиндров. После этого он записывает информации на диск, перемещая блок головок с внешнего края диска к внутреннему только один раз и записывает данные в том порядке, в каком они размещаются в накопителе, а не в том, в каком их передавал в контроллер ЦП. Такой способ оказывается наиболее быстрым, но здесь приходится помнить о возможности потери данных.

5.3. Выбор опций для программы кэширования

Подготовка или настройка программы кэширования диска может оказаться сложной задачей, поскольку лучшие программы имеют многочисленные опции. Как узнать, какие опции будут для вас лучшими?

Вы ничего не узнаете, не проделав множества экспериментов. Можно принять в программе настройку по умолчанию и, очень вероятно, вы от этого только выиграете. Но если вы хотите действительно настроить систему и заставить ее функционировать с максимальной производительностью, у вас нет выбора, кроме произведения множества экспериментов до обнаружения того способа, который дает наилучшие результаты.

Проводите эксперименты с теми программами, в которыми вы обычно работаете. Иногда производители кэша поставляют тестовую программу, которая позволяет оценить их продукт. Другие тестирующие программы поставляются с различными пакетами утилит. Не принимайте на веру рекомендации тестовых программ по настройке вашего кэша. Эти рекомендации просто показывают, насколько хорошо текущую настройку ускоряют тестовые программы, но ваша повседневная работа не обязательно ускорится.

5.4. Различные организации кэша

Сущность любого кэша — предоставить пул повторно используемой памяти для временного хранения информации. До сих пор при рассмотрении различных способов кэширования мы касались тех моментов, когда элементы сохранялись в кэше и когда они копировались из кэша на диск для постоянного хранения.

Имеется и еще одно различие в организации кэша — как контроллер кэша определяет, какие части его памяти использовать для сохранения кэшируемого элемента.

5.4.1. Полностью ассоциативный кэш

Полностью ассоциативный кэш — это лучшая организация кэша. Название подчеркивает то, что любой сохраняемый элемент можно поместить в любое место кэша вместе с регистрацией того полного адреса диска, откуда взята информация или куда она должна быть отправлена.

Такой кэш оказывается лучшим, но и одновременно самым дорогим. Контроллер кэша должен анализировать каждую ячейку при считывании или записи, проверяя, не находится ли уже элемент в кэше. Определение оптимального уничтожаемого элемента при заполнении памяти кэша также требует выполнения определенных операций. Поэтому, полностью ассоциативный кэш обычно реализуется аппаратно, что обеспечивает одновременный параллельный аппаратный опрос всех ячеек. Пока этот способ оказывается для персональных компьютеров слишком дорогим.

5.4.2. Кэш с прямым отображением

Противоположным оказывается кэш с прямым отображением, в котором каждой дисковой ячейке назначается уникальная ячейка в кэше. Конечно, поскольку память кэша намного меньше емкости дискового накопителя, каждой ячейке кэша назначаются много ячеек диска. (Математики называют такую ситуацию многозначным отображением.)

В кэше с прямым отображением имеется очевидный и простой способ отображения дисковых ячеек на ячейки кэша: последовательные ячейки на диске назначаются последовательным ячейкам кэша до исчерпания ячеек кэша. Следующая дисковая ячейка назначается первой ячейке кэша, и процесс повторяется до назначения всех дисковых ячеек ячейкам кэша. Дисковые ячейки, назначенные каждой ячейке кэша, максимально удалены друг от друга, а такое отображение в большинстве случаев оказывается оптимальным.

В каждой операции считывания или записи информации сектора контроллер кэша должен проверять, не кэширован ли элемент, только в одном месте. Если элемент не кэширован, контроллер сразу знает, какой элемент в кэше нужно замещать новым. Кэш с прямым отображением является простейшей разновидностью кэша, но одновременно и наименее эффективным.

5.4.3. Ассоциативный по множеству кэш

Ассоциативный по множеству кэш (по-видимому, вы слышали названия двунаправленный и четырехнаправленный ассоциативный по множеству кэш), в сущности, представляет большое количество очень малых полностью ассоциативных кэшей. Дисковое пространство разделяется на большое количество одинаковых множеств, и каждой дисковой ячейке назначается один из этих малых мини-кэшей. Например, четырехнаправленный ассоциативный по множеству кэш хранит четыре элемента данных (и регистрирует, где они находятся на диске) в каждом из своих крошечных полностью ассоциативных кэшей.

Отображение дисковых ячеек на кэши осуществляется аналогично способу, используемому при прямом отображении. По-прежнему различные ячейки на диске отображены на тот же самый мини-кэш, будут максимально разнесенными.

Когда в четырехнаправленном с отображением по множеству кэше требуется считать или записать сектор, он должен проверить только в четырех местах, не находится ли уже этот элемент в кэше. Когда эти четыре ячейки уже заполнены другими данными, для уничтожения приходится выбирать только среди четырех ячеек. Такая организация намного проще полностью ассоциативного кэша, но сложнее кэша с прямым отображением. Конечно, стоимость такого кэша находится между двумя крайностями.

Однонаправленный ассоциативный по множеству кэш является просто другим названием кэша с прямым отображением. С другой стороны, *n*-направленный ассоциативный кэш, в котором *n* равно количеству элементов, которое можно хранить во всем кэше, является другим описанием полностью ассоциативного кэша.

5.4.4. Как узнать тип кэша

Как определить, какой у вас тип кэша? По-видимому, вам это сделать не удастся. Производители кэшей выбирают представляющийся им наилучшим способ и не общаются детали своего способа. (Исключение составляют выбираемые пользователем параметры для разрешения или запрещения некоторых опций. Производители описывают эти параметры так, чтобы пользователи знали, что они делают.)

Каждый из производителей кэшей предоставляет сведения, доказывающие, что их способ организации наилучший. Очевидно, это так и есть, но только для ситуаций, использованных при тестировании. Огромное число предложений на рынке показывает, что "лучшее" решение для вашего компьютера зависит от конкретной системы и способов ее использования.

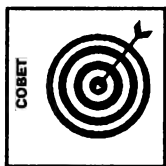
5.5. Буферирующие и кэширующие контроллеры

Некоторые производители контроллеров дисков могут утверждать, что одно из их изделий имеет кэш, когда на самом деле это более или менее большой буфер: ЗУПВ значительной емкости, которое контроллер может заполнять с одного конца и опустошать с другого. Такой буфер может быть доступным и в середине, но все-таки это просто один блок данных. Если в буфере нет запрошенных вами данных, контроллер заполняет весь блок новыми данными.

Функционируя как сверхбуфер дорожки, такой буфер может быть намного больше буфера дорожки, но не обеспечивает значительного выигрыша в скорости. Его основной недостаток связан с отсутствием гибкости, не позволяющей уследить за тем, как DOS "мечется" по дисковому накопителю.

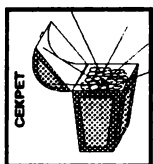
5.5.1. Программа кэширования или аппаратный кэш

Контроллеры диска с аппаратным кэшированием повышают быстродействие, но они довольно дороги. Программный дисковый кэш оказывается намного дешевле и часто действует одинаково хорошо. Программа привлекает неиспользуемое ЗУПВ персонального компьютера аналогично тому, как кэширующие контроллеры используют свое ЗУПВ.



Программа кэширования диска может хранить информацию в расширенной памяти (в компьютерах AT и выше) или в отображаемой памяти. Не рекомендуется хранить информацию в системном ЗУПВ (нижние 640 Кбайт); эту память необходимо оставить для других целей.

Применяя программу кэширования диска, например Super PC-Kwik фирмы PC-Kwik, вы получите много достоинств, кроме аппаратного кэша с отложенной записью. Последняя версия программы SMARTDrive фирмы Microsoft обеспечивает и кэширование с отложенной записью. Стоимость таких программ составляет только часть стоимости аппаратного кэша, даже с учетом приобретения дополнительного ЗУПВ.



Не удивляйтесь, узнав, что эффективность программного кэша может быть такой же, как и у аппаратного. На самом деле, в контроллерах дисков с аппаратным кэшированием есть отдельный процессор для управления кэшем, а программа кэширования должна использовать для этого ЦП персонального компьютера.

Причина успешной конкуренции программ кэширования с аппаратным кэшем состоит в том, что их ограничивает быстроедействие шины ввода-вывода. Если у вас нет компьютера с быстрой (32-битовой или шире) шиной ввода-вывода, вы можете обойтись программой кэширования диска. Быстрая 32-битовая шина относится к компьютеру с MCA (Micro Channel Architecture) или с EISA (Extended Industry Standard Architecture), к компьютеру со специальной "локальной" шиной или к особому разьему и соответствующему контроллеру диска.

Для достижения абсолютно максимальной производительности можно применить программное и аппаратное кэширование! (Однако выигрыш в быстродействии будет всего на несколько процентов выше по сравнению с тем, когда применяется что-то одно.)

Расширенная и отображаемая память

Многие люди путаются с определениями расширенной (extended) и отображаемой (expanded) памяти в персональных компьютерах. Отображаемая память разработана для компьютеров с процессорами 8088/8086. Ее назначение состояло в том, чтобы преодолеть известный барьер DOS в 640 Кбайт. Благодаря страничной организации она позволяла памяти до 16 Мбайт (правда, небольшими областями) появляться в адресном пространстве 1 Мбайт процессора.

С другой стороны, расширенная память имеется только в компьютерах с процессорами 80286, 386 или 486. Эта память имеет реальные физические адреса выше 1 Мбайт, максимально доступного в первых ПК. Проблема этой памяти при работе с DOS заключается в том, что DOS не проектировалась с расчетом на расширенную память. Поэтому для использования расширенной памяти DOS программами требуются специальные приемы.

У контроллера диска с аппаратным кэшированием есть одно важное преимущество по сравнению с программой кэширования. Если в кэше есть информация, еще не записанная на диск и если компьютер зависает (т.е. совершенно запутывается в том, что делать дальше), программа кэширования не сможет закончить запись данных на диск, а контроллер сможет сделать это.

5.6. Дисковые буферы DOS

Рассмотрим теперь буферы, которые имеются в каждом персональном компьютере, — дисковые буферы DOS. Если в вашем файле CONFIG.SYS нет строки `BUFFERS = n,m`, где n,m — некоторое число, то DOS назначает принимаемое по умолчанию число буферов.

В ранних версиях DOS это принимаемое по умолчанию число составляло всего 2 для компьютеров PC и XT и только 3 для компьютеров AT и с процессорами 386 и 486. Несмотря на то, что такого числа буферов было достаточно для ПК с емкостью ЗУПВ 16 Кбайт и даже с емкостью 640 Кбайт, сейчас такое число буферов стало неприемлемым.

Небольшое число дисковых буферов DOS имеет смысл только при наличии большого кэша в контроллере диска или при использовании программы кэширования диска. В этой ситуации нужно следовать рекомендациям фирмы — производителя кэша, а в остальных случаях вам придется задать свое число буферов в операционной системе `BUFFERS`.

В более поздних версиях DOS число буферов увеличено. Принимаемые по умолчанию значения для MS DOS 5.0 и рекомендуемые фирмой Microsoft приведены в табл. 5.1 и 5.2. Отметим, что значения по умолчанию зависят только от емкости ЗУПВ, а рекомендуемые значения зависят от емкости (или размера) жесткого диска.

Таблица 5.1. Принимаемое по умолчанию количество буферов в MS DOS 5.0

Конфигурация ЗУПВ ПК	Буферы	Количество байтов ЗУПВ
Менее 128 К с накопителем 360 К	2	—
Менее 128 К и накопитель больше 360 К	3	—
От 128 К до 255 К	5	2672
От 256 К до 511 К	10	5328
От 512 К до 640 К	15	7984

Таблица 5.2. Рекомендуемые Microsoft значения для BIFFERS = n, m

Размер жесткого диска	Количество буферов DOS	Количество дважды кэшируемых буферов
(n) кэша (m)		
Менее 40 Мбайт	20	8
От 40 М до 79 Мбайт	30	8
От 80 М до 119 Мбайт	40	8
Более 120 Мбайт	50	8

5.7. Выбор количество буферов в операторе BUFFERS

Кажется, вы готовы задать правильное число в операторе BIFFERS= файла CONFIG.SYS. Единого правильного ответа для всех персональных компьютеров нет, но ход рассуждений всегда примерно одинаков. Для эффективной работы компьютера желательно иметь достаточное количество буферов DOS. Но, с другой стороны, для буферов расходуется ценное ЗУПВ в области 640 Кбайт. Приходится также учитывать особые требования прикладных программ к числу дисковых буферов DOS. Ну, и, конечно, ваш выбор должен быть оправданным. Для выбора количества буферов DOS имеются общие рекомендации, которые рассматриваются дальше.

DOS не может функционировать, не имея минимум одного дискового буфера, хотя при наличии только одного буфера его придется все время использовать повторно, а такая работа окажется малоэффективной.

Допустимые значения для чисел в операторе BIFFERS файла CONFIG.SYS составляют от 1 до 99 для общего числа буферов и от 1 до 8 для вторичного кэша. Почти всегда выбор максимального значения 99 будет ошибочным, так как для буферов будет отведено около 50 Кбайт нижней памяти (первые 640 Кбайт).

При использовании другого способа кэширования диска (программный кэш или кэширующий контроллер диска) можно задать небольшое количество дисковых буферов DOS. Например, хорошим выбором окажется оператор BIFFERS=3. В этом же случае можно оставить минимальное количество буферов для вторичного кэша

DOS. Поскольку принимаемое по умолчанию значение всегда равно 1, можно ничего не определять, а правильный выбор получается автоматически.

Если по какой-то причине вы не применяете программу кэширования диска или кэширующий контроллер диска, придется увеличить как общее количество буферов, так и количество буферов вторичного кэша (см. табл. 5.2).

Некоторые программы требуют задания определенного минимального числа в строке `BUFFERS=`, но большинство программ об этом не заботится. Они полагают, что вы правильно определили число дисковых буферов DOS и настроили другие средства кэширования диска. Вам редко придется учитывать конкретные требования программ.

При работе с DOS 5.0 приходится учитывать некоторые специальные особенности. Если вы применяете MS DOS или PC DOS, определили `DOS=HIGH` и загрузили драйвер `HIMEM.SYS`, дисковые буферы DOS размещаются в области высокой памяти (HMA), которая находится сразу над 1 Мбайт в адресном пространстве памяти. Дисковые буферы DOS размещаются там, если для них достаточно места. В эту область помещаются до 35-40 буферов. Помните, что при определении слишком большого количества буферов все они окажутся в нижней памяти. При использовании DR DOS 6 можно загрузить часть или все дисковые буферы DOS либо в HMA, либо в любые доступные блоки верхней памяти (UMB).

При задании параметров программы кэширования диска единственный способ определить лучшее количество дисковых буферов DOS в конкретной ситуации заключается в проведении тщательного (и продолжительного) теста. Если вы не хотите проводить такой тест, можно воспользоваться рекомендациями Microsoft (или Digital Research при работе с DR DOS) или рекомендациями производителя программы кэширования диска.

Заключение

- Кэш — это повторно используемый блок памяти для временного хранения информации. Главное назначение кэша — ускорить передачу информации между устройствами с различным быстродействием.
- Дисковый кэш является удивительным добавлением для любого ПК. Его можно организовать в любом компьютере; для этого достаточно ввести соответствующую строку в файл `CONFIG.SYS` или `AUTOEXEC.BAT` для загрузки программы кэширования диска.
- Имеются также аппаратные кэширующие контроллеры диска, и иногда (хотя и не часто) их выбор оказывается наилучшим.
- Есть много разновидностей кэша. Их различия заключаются в выборе сохраняемой информации и передаче сохраненной информации получателю сразу же или позже.
- Наибольшее распространение получил дисковый кэш считывания и сквозной записи. Наглядным примером здесь служит программа PC-Kwik.
- Применяется также кэш считывания и отложенной записи, например программа SMARTDrive фирмы Microsoft.
- Другое различие касается способа выбора того места, в которое помещается новый элемент данных. Есть полностью ассоциативный кэш, ассоциативный по множеству кэш и кэш с прямым отображением.
- Оператор `BUFFERS=` в файле `CONFIG.SYS` управляет количеством создаваемых дисковых буферов DOS и буферов вторичного кэша. Вероятно, лучше всего оставить это количество небольшим (например, 3 и 1), а затем применить программу кэширования диска. Если это невозможно, воспользуйтесь рекомендациями Microsoft или (при наличии времени) выполните некоторые тесты и подберите оптимальные для вашего компьютера числа.

Теперь вы знаете все о структуре и работе жестких дисков и готовы к рассмотрению того, как диски взаимодействуют с MS DOS

•

Глава 6. Жесткие диски в DOS

В этой главе вы узнаете

- Что такое DOS и зачем она нужна
 - Как DOS работает с дисковыми накопителями и в чем они отличаются от других устройств, например клавиатуры и принтеров
 - О нескольких уровнях (слоях) программ, имеющихся в вашем ПК и как эти программы работают
 - Как работает механизм прерываний ПК и как он помогает уровням программ выполнять их работу
 - Как резидентная программа или драйвер устройства может изменить работу ПК и почему вам часто требуется применять их
 - Как DOS поддерживает множество разнообразных жестких дисков, включая те, которых не было при появлении DOS
 - Очень интересные способы, с помощью которых разработчики дисков расширили принцип преобразования секторов в накопителях IDE и SCSI.
-

6.1. Введение в DOS

Что такое DOS? Зачем она нужна мне? Что она делает для меня? Как она делает это? Это все важные вопросы, на которые требуются понятные ответы. Эта книга не рассчитана на подробное изложение DOS, но в этой главе автор дает полезные ответы на поставленные выше вопросы.

Вы купили свой компьютер для выполнения некоторой работы или, возможно, для решения многих задач. Вы купили программы, называемые прикладными программами или приложениями, которые помогают вам выполнить эту работу. Примерами приложений являются такие программы, как WordPerfect, Excel или Lotus 1-2-3. Зачем же вам все-таки нужна DOS?

Разработчики прикладных программ не знают и не хотят знать все детали о имеющихся у вас аппаратных средствах. Вместо этого они хотят написать программы для гипотетического "стандартного" ПК. Более того, они не хотят знать ничего о том, какие другие программы могут выполняться в вашем компьютере вместе с их приложениями.

Аббревиатура DOS (Disk Operating System) означает дисковая операционная система. Это название программы, но фактически совокупности программ. Основная задача этих программ заключается в том, чтобы управлять ресурсами ПК для вас и для всех других прикладных программ, которые вы выбрали для выполнения. Такая возможность позволяет разработчикам этих прикладных программ сконцентрировать усилия на выполнение своей работы, не заботясь о том, как управлять аппаратными средствами компьютера, и о том, какие другие программы могут выполняться на этом же компьютере в то же время.

Разделение программирования для компьютера на создание операционной системы и разработку приложений действует прекрасно. Конечно, разработчик прикладных программ должен кое-что знать о диапазоне аппаратных возможностей, а в некоторых случаях даже написать конкретный драйвер для конкретных классов аппаратных средств. Обычно это относится к дисплеям и принтерам.

Наибольшее разделение задач между разработчиками операционной системы и прикладными программистами имеет место в альтернативных операционных системах и в некоторых расширениях DOS Windows, GeoWork, OS/2, и Windows NT

являются примерами программных сред (аппаратного и программного контекста, в которых выполняется программа), обеспечивающих максимальную независимость от устройств. Это значит, что программы, написанные для работы с одной из этих сред, не обязательно рассчитаны на знание устройств, подключенных к тому ПК, на которых эти программы и будут выполняться. Приложения могут запросить информацию от среды, например какая разрешающая способность экрана или принтера, а затем подстроить свой выход под полученные значения.

Обычно люди говорят о DOS как о единой сущности. Фактически же DOS состоит из нескольких уровней, каждый из которых решает свои задачи. Очень удобно разделить операционную систему на две части: BIOS и собственно DOS. (Да, DOS — это название одной части гораздо большего целого, которое тоже называется DOS. Автор больше не может прояснить эту ситуацию, считая это просто путающим фактом. По крайней мере, автор попытается четко различать эти два смысла в последующем материале).

Первый набор программ, называемый базовой системой ввода-вывода или BIOS (Basic Input/Output System), учитывает знание разработчиком точных аппаратных средств вашего ПК и команд для управления ими. Вторая часть DOS определяет стратегию управления ресурсами компьютера, в частности дисковыми накопителями.

Любой ПК имеет много общего со всеми остальными компьютерами, поэтому стандартными компонентами аппаратных средств может управлять стандартный набор процедур BIOS. Но многие ПК имеют и относительно необычные аппаратные средства, поэтому для управления такими средствами требуются специализированные добавления в BIOS.

Во всех ПК применяется стандартная стратегия для управления файлами. Вот почему эту функцию выполняет высший уровень DOS (именно он и называется DOS). DOS не нужно модифицировать при изменении аппаратных средств компьютера. Однако может потребоваться дополнить ее, если вы подключились к сети или приобрели накопитель CD ROM или какие-то другие новые аппаратные средства.

Подробности того, как DOS и BIOS работают с аппаратными средствами и как они выполняют свои функции, будут проявляться по мере изучения материалов этой главы и глав 7–10. Наконец, вам нужна DOS потому, что она требуется прикладным программам, а для вас это очень удобно. Фирмы будут разрабатывать больше хороших программ, отделяя разработку приложений от операционной системы, которой занимаются отдельные коллективы специалистов.

6.2. Взаимосвязь между жесткими дисками и DOS

Вы удивляетесь, почему в этой книге DOS так упорно связывается с жесткими дисками. Ответ заключается в том, что точное знание работы жесткого диска влияет на множество программ, которые применяются для обращения к нему, а все эти программы в ПК образуют часть того, что мы называем DOS. По крайней мере это справедливо для большинства ПК за исключением тех ПК, которые работают только с OS/2, UNIX или другими операционными системами.

Все IBM PC или совместимые компьютеры (которые в этой книге обозначаются ПК) могут выполнять MS DOS — дисковую операционную систему фирмы Microsoft. DOS поставляется в разнообразных формах. Кроме оригинальной PC DOS от фирмы IBM (которая в сущности является MS DOS с другим названием) и чистой MS DOS, которую можно приобрести прямо у фирмы Microsoft, имеется множество специализированных версий MS DOS (например, Compaq DOS), которые модифицированы для работы с аппаратными средствами, несколько отличающимися от разработок фирмы IBM. Фирма Digital Research (подразделение фирмы Novell) выпускает систему DR DOS, функционально почти аналогичную MS DOS. Объеди-

няя все эти разновидности DOS, можно сказать, что большинство ПК работают с DOS каждый день.

Для выполнения DOS компьютер должен быть спроектирован на базе микропроцессора 80x86 фирмы Intel (он называется также центральным процессором, или ЦП) и должен удовлетворять добавлениям фирмы Microsoft к разработке фирмы Intel. Более того, чтобы быть совместимым с IBM PC, компьютер должен удовлетворять определенным аспектам разработки семейств IBM PC PS/1 или PS/2. Таким образом, название ПК означает гораздо больше, чем то, что компьютер просто использует микропроцессор 80x86.

Примечание. Обозначение 80x86 относится к микропроцессорам 8088, 8086, 80186, 80286, 386 и 486, а также другим членам этого семейства. (Можно эмулировать ПК на другом компьютере и выполнять программы DOS. Например, компьютер Amiga может иметь плату PC Bridge, а компьютер Macintosh или NeXT может выполнять программу эмулятора Soft PC.)

Некоторые используют на своих ПК другие операционные системы. В научных учреждениях часто применяется система UNIX, в больших коммерческих фирмах иногда применяется OS/2. В наиболее популярной сети Novell NetWare применяется специальная операционная система на файловых серверах (а DOS как обычная операционная система применяется на рабочих станциях). Фирма Microsoft недавно объявила новую операционную систему Windows NT для ПК. (В главах 9 и 10 вы найдете краткое описание, чем OS/2, Windows NT, UNIX и NetWare отличаются от DOS при работе с жесткими дисками.)

Большинство из нас применяют MS DOS, поэтому в этой книге упор делается на то, как именно она работает с жесткими дисками. Кроме того, программа SpinRite и другие программы изменения коэффициента чередования секторов работают только на ПК в среде DOS. (Чередование секторов и изменение коэффициента чередования рассмотрены в главе 3.)

Общая информация о жестких дисках в предыдущих главах применима к любым жестким дискам в любом компьютере. Теперь же мы подробнее коснемся жестких дисков в ПК, работающих с DOS. Поиск неисправностей (глава 11), в основном применим ко всем жестким дискам, но сообщения об ошибках относятся только к компьютерам с DOS.

Достаточно общих вопросов и пора детально рассмотреть, как DOS управляет жесткими дисками в ПК.

6.3. Блочные и символьные устройства DOS

Прежде всего необходимо разобраться в том, как DOS управляет передачей информации из одного места ПК в другое. Иногда информация поступает с клавиатуры, в другое время — с дискового накопителя. Иногда получателем информации является экран или принтер, а в другое время — дисковый или ленточный накопитель.

DOS обращается со всеми возможными источниками и получателями информации довольно единообразно, считая их просто устройствами. DOS классифицирует все устройства на два подкласса: символьные и блочные устройства. После этого DOS работает с символьным устройством одним способом, а с блочным другим способом.

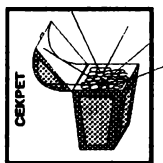
Под символьным устройством понимается любое устройство, которое может воспринять или передать одновременно только один символ информации. Примерами символьных устройств являются экран и клавиатура. Вы нажимаете одну клавишу, и компьютер получает одно клавишное действие. Это же клавишное действие затем по-

сылается на экран для вывода, информируя вас о том, что компьютер зафиксировал ваше действие. К символьным устройствам относятся также мышь, модем и принтер.

Блочные устройства оперируют информацией блоками некоторого фиксированного размера, содержащими много символов. Дисковые и ленточные накопители являются блочными устройствами. Любое блочное устройство должно иметь буфер в ЗУПВ, где компьютер может собрать передаваемую устройству информацию или где устройство может поместить текущий блок своей информации.

По аналогии можно сказать, что дисковый накопитель похож на огромное хранилище данных. Буфер похож на погрузо-разгрузочную площадку, где можно собрать пакеты данных для погрузки на грузовик и отправки и где данные можно разгрузить с прибывающих грузовиков до размещения внутри хранилища.

Необходимо иметь хотя бы несколько дисковых буферов DOS. DOS позволяет создать любое число буферов от 2 до 99. В главе 5 подробно рассмотрено, какое количество дисковых буферов принимается по умолчанию в различных ситуациях и как определить лучшее количество буферов для вашей системы.



В DOS дисковый буфер обычно содержит 512 байт данных, соответствующих одному дисковому сектору. Каждый буфер фактически занимает несколько больше памяти, чем 512 байт, а именно — 528 байт. DOS использует дополнительные 16 байт для управления буфером.

Некоторые производители разработали специальные программы, позволяющие пользователям старых версий DOS (до версии 3.3) обращаться к большим жестким дискам (емкостью больше 32 Мбайт). В некоторых случаях эти специальные программы представляют собой адаптированные версии DOS, а другие являются программами организации разделов третьих фирм. Когда одна из специальных программ загружается в память, она для достижения своей цели фактически модифицирует то, как жесткий диск представляется DOS. Иногда такая модификация осуществляется увеличением кажущегося размера (логического размера) сектора жесткого диска.

Имеются подключаемые к ПК периферийные устройства, которые похожи на жесткий диск, но используют большие секторы. Например, в некоторых оптических дисковых накопителях с записью применяются 512- или 1024-байтовые секторы. Это фактически физическое увеличение размера сектора, а не просто удобная фикция, как это было в случае упомянутых выше программ.

Когда у вас есть блочное устройство, которое использует размер физического или логического сектора больше 512 байт, дисковые буферы DOS увеличиваются так, что они по-прежнему могут хранить один (логический) сектор данных.

Когда ваша программа хочет считать некоторую информацию с диска, даже если ей нужно считать всего несколько байтов, она должна просить DOS считать в дисковый буфер целый сектор, содержащий эти байты. После этого программа может выделить нужную информацию из буфера.

Аналогично, если ваша программа хочет записать некоторую информацию на диск, она фактически записывает ее в дисковый буфер. Обычно содержимое буфера записывается в сектор на диске после заполнения буфера. Если программе требуется дальнейшая запись, она повторяет рассмотренный процесс, заполняя буфер и записывая его на диск сектор за сектором.

Когда программа заканчивает запись в буфер последней информации, то, даже если она не заполняет буфер полностью, программа должна как-то осуществить запись буфера на диск. Для этого она сообщает DOS об окончании работы с файлом (т.е. "закрывает" файл) и DOS записывает последний буфер на диск независимо от его содержимого.

Наглядно можно представить себе, что символьные устройства получают или отправляют информацию "в розницу", а блочные устройства оперируют информацией "оптом" (рис. 6.1).

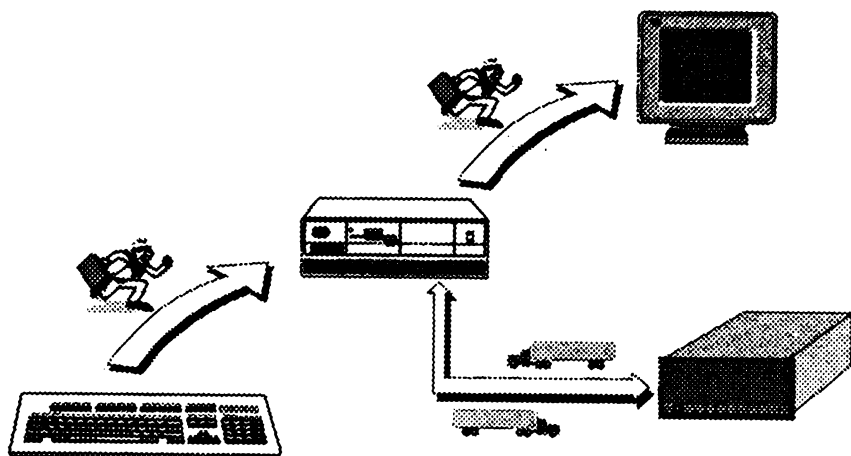
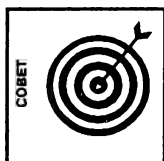


Рис. 6.1. Отличия символьных и блочных устройств DOS



Малонизвестный и скрытый риск потери секретов

Из-за принятого способа записи информации на диск очень вероятно, что последний буфер будет содержать не только информацию программы. Ясно, что в начале буфера находится последний вывод программы, но в его конце содержится то, что записывалось на диск или считывалось с диска последний раз.

В результате почти любой файл на диске, который не точно заполняет последний выделенный ему сектор, в конце содержит какую-то ненужную информацию. Более того, так как DOS распределяет пространство на диске единицами, которые кратны сектору, в конце файла могут оказаться секторы, в которые программа совсем не записывала. О содержимом таких секторов приходится только гадать.

Обычно этот факт не вызывает никаких проблем, поскольку DOS фиксирует в каталоге диска размер файла с точностью до байта. Большинство программ использует эту информацию для считывания только нужной информации. Но это означает только, что они достаточно осведомлены для игнорирования "мусора". DOS как минимум, передаст весь последний сектор данных файла и, возможно, остальные секторы этой единицы распределения; только использующая эту информацию программа решает, что она должна игнорировать.

Имеются дисковые утилиты, позволяющие посмотреть, что в действительности хранится на диске. Они могут показать весь тот "мусор", который находится за концом каждого вашего файла. (Одной из таких утилит является Disk Editor из Утилит Нортон.) Такая возможность оказывается важной, если вам нужно защитить свою информацию. Иногда вы можете передать файл кому-то еще, не зная о наличии в его конце конфиденциальной информации.

Если это обстоятельство важно для вас и особенно если вы обнаружили, что кто-то пытался посмотреть диск в ваше отсутствие, есть простое решение. Нужно выполнить программу WipeInfo (также из утилит Нортон) и с помощью опции /K заставить ее "Уничтожить пропавшее пространство в конце файлов". Можно также задать опцию /E для уничтожения ("затирания") стертых мест на диске. (Эта опция делает невозможным восстановление файлов, удаленных до выполнения WipeInfo.)

Некоторые базы данных уязвимы еще в одном отношении: несущественные данные можно вставить в середину файла так, что он оказывается полностью невидимым другим программам, кроме создавшего его приложения, но его все же можно посмотреть с помощью утилиты просмотра диска.

Такой "мусор" можно обнаружить также, когда файл передается через модем из одного компьютера в другой. Часто информация посылается блоками и точная длина файла не сообщается; в результате принимающий ПК фиксирует длину с округлением до конца последнего блока. Размер блока варьируется от менее 100 байт до нескольких килобайтов. Некоторые коммуникационные программы дополняют конец последнего блока безвредным "символом-заполнителем", а другие посылают то, что находится в дисковом буфере за концом файла. Во втором случае конец файла, сохраненный в принимающем ПК, может содержать вполне видимую и, возможно, путывающую информацию.

6.4. Действия DOS

Напомним, что в ПК есть несколько уровней программ. Вы общаетесь только с высшим уровнем, но программы этого уровня не могут действовать без помощи программ нижних уровней.

Далее подробно рассматриваются такие программы. В частности, вы познакомитесь с программами нижнего уровня, (процедурами BIOS), важным средством взаимодействия между программами (прерываниями), и интересными резидентными программами. Понимание этих аспектов функционирования ПК важно для понимания взаимодействия DOS с жесткими дисками.

6.4.1. Уровни программ в ПК

Некоторые программы, управляющие работой ПК, встроены в аппаратные средства. (Обычно такие программы называются аппаратно-программными средствами — *firmware*.) Эти программы содержатся в микросхемах ПЗУ, расположенных на материнской плате ПК и, возможно, на съемных платах.

Другие программы считываются в ЗУПВ с одного или нескольких дисковых накопителей. При загрузке компьютера с диска вы загружаете несколько уровней программ в ЗУПВ своего ПК.

На рис. 6.2 в упрощенном виде представлены некоторые уровни. Этот рисунок показывает фактическое распределение памяти в реальном ПК, работающем с MS DOS 5.0. Отметим, что нижние 64 Кбайт даны в увеличенном масштабе, чтобы подробнее показать отдельные уровни в нижней части адресного пространства памяти ЦП.

На рис. 6.2 представлен только первый мегабайт адресного пространства памяти ЦП. Он соответствует всей памяти компьютеров IBM ПК и XT. В компьютерах AT и последующих моделях есть расширенная память, но DOS обычно не может использовать ее. Первые 640 Кбайт первого мегабайта обычно называются нижней памятью или системным ЗУПВ. (Программы с расширением DOS, например Windows 3.x, Lotus 1-2-3 3.x и AutoCad 10, а также драйверы устройств VDISK и RAMDRIVE, поставляемые с ПК DOS и MS DOS могут использовать расширенную память. Начиная с версии 5.0, DOS включает драйвер EMM386, который позволяет преобразовывать расширенную память в имитируемую отображаемую память или в верхнюю память.)

На рис. 6.2 может быть представлен любой ПК. Шахматный узор в верхнем конце адресного пространства памяти показывает встроенные программы, находящиеся в ROM BIOS на материнской плате. Еще один такой узор от адреса C0000h до адреса E0000h означает, что в этой области могут находиться дополнительные процедуры ROM BIOS. Они размещаются в ПЗУ на съемных платах.

Затененная область от адреса A0000h до C0000h показывает, что она используется для ЗУПВ видеоплат, в котором хранится информация об изображении. Обычно занята не вся эта область, но потенциально ее всю можно использовать для видеоплат.

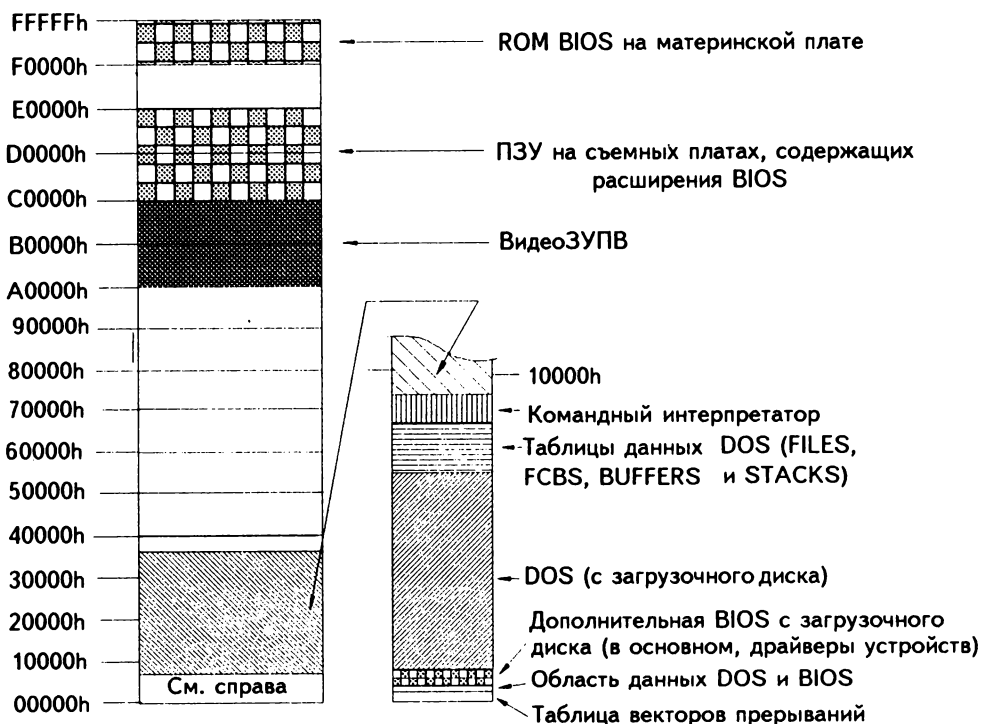


Рис. 6.2. Уровни программ в области памяти типичного ПК, работающего с MS DOS 5.0

Области в нижней памяти (ниже адреса 40000h) — это те уровни, которые загружаются при запуске ПК и выполняют вашу первую прикладную программу. Конечно, занимаемая прикладной программой область зависит от конкретной программы. Пустое пространство выше прикладной программы и ниже видеоЗУПВ означает свободное ЗУПВ. (Самая верхняя часть этой области занята транзитной частью COMMAND.COM, но ее может "взять" любая прикладная программа, желающая использовать это пространство. В случае перезаписи содержимое этой части восстанавливает резидентная часть COMMAND.COM при возврате к стимулу DOS.)

Ради максимального упрощения примера на загрузочном диске нет файлов CONFIG.SYS и AUTOEXEC.BAT, поэтому рис. 6.2 показывает простейшую картину для ПК, работающего с DOS. Этот рисунок пригодится нам при обсуждении далее прерываний. Усложнение из-за введения хотя бы одной резидентной программы также обсуждается далее.

Далее вы узнаете о назначении области, отмеченной как "Таблица векторов прерываний". Область данных DOS и BIOS представляет служебную память, в которой DOS и BIOS содержат информацию о конфигурации ПК и режимах работы. Область "Дополнительная BIOS с загрузочного диска" содержит дополнительные процедуры BIOS, загружаемые из скрытого системного файла на любой загрузочной дискете. Область DOS загружается из другого скрытого системного файла. Командным интерпретатором обычно служит программа COMMAND.COM, о которой знает большинство людей, так как она представлена видимым файлом на загрузочных дисках DOS.

(Замечание о рис. 6.2 — на нем предполагается отсутствие специальных программ управления памятью, даже поставляемых с DOS; при наличии таких программ DOS занимает меньше нижней памяти, чем показано на рисунке.)

Необходимо обратиться к одной из книг, посвященных DOS, использованию памяти и управлению в ПК, чтобы подробнее разобраться в тех уровнях программ, которые создаются в нижней памяти. Здесь нам достаточно или почти достаточно просто знать о существовании этих уровней. Гораздо важнее, как эти уровни используются, когда вы или ваши программы обращаются к информации на диске. (Одной из таких книг является монография Memory Management for All of Us, опубликованная издательством Sams в 1992 г.)

При загрузке документа в текстовый процессор, например, вы просто нажимаете несколько клавиш и сообщаете имя файла. Ни вы, ни текстовый процессор не должны точно знать, как загружается документ; программа просто предлагает другой программе на нижнем уровне выполнить загрузку документа.

Текстовый процессор передает ваш запрос DOS (здесь DOS означает только высший уровень большей сущности, называемой DOS, которая включает в себя BIOS). DOS "понимает" запросы в терминах имен файлов. Она преобразует ваш запрос в последовательность более детальных запросов для BIOS, которая и представляет собой совокупность программ нижнего уровня. DOS должна сформулировать каждый из этих запросов для BIOS в терминах фактических участков диска, где хранятся данные файла (конкретные значения цилиндра, головки и сектора).

До выдачи запросов DOS вначале должна определить, где на диске хранится файл. Для этого она должна запросить у BIOS копии таблицы размещения файлов (File Allocation Table — FAT) и одного или нескольких каталогов. Обращаясь к этим таблицам, DOS может найти файл на диске.

BIOS преобразует эти запросы нижнего уровня в последовательность еще более подробных запросов для контроллера жесткого диска. Например, для считывания одного сектора BIOS предлагает контроллеру позиционировать блок головок считывания-записи на нужный цилиндр, включить нужную головку и, наконец, зафиксировать подход нужного сектора и считать его. На рис. 6.3–6.6 показаны некоторые из многочисленных этапов рассмотренного процесса (рис. 6.3 и 6.4, а также 6.5 и 6.6 нужно рассматривать попарно.)

Рассмотренные уровни позволяют разработчикам некоторого уровня сосредоточиться именно на конкретном уровне работы ПК, а это дает значительные преимущества и им и нам. Если со временем вы захотите изменить действия ПК на одном уровне, нужно заменить только программу, которая работает именно на этом уровне, например, при покупке нового текстового процессора или замене драйвера принтера.

6.4.2. Прерывания

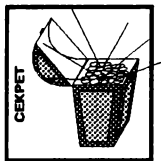
Рассмотренная выше концепция многоуровневого программирования не действует без стандартизованного способа передачи сообщений между различными уровнями. ПК осуществляет такое взаимодействие с помощью прерываний.

Прерывания — это события, которые заставляют выполняемую программу приостановить свои действия, пока другая программа не обработает это событие. По окончании программы обработки события она возвращает управление ПК первой программе, которая возобновляет свои действия.

Некоторые прерывания вызывают аппаратные события. Например, контроллер жесткого диска может сообщить программе BIOS о том, что он считал сектор и данные готовы для обработки. При других обстоятельствах контроллер должен сообщить программе BIOS о том, что прочитать запрошенный сектор невозможно.

Другие прерывания вызывают программы. Например, DOS выдает прерывание всякий раз, когда BIOS должна для нее что-то сделать. Многие прикладные программы формируют прерывание, когда DOS должна что-то сделать для программы.

В микропроцессорах фирмы Intel реализована простая и надежная процедура прерываний. Когда возникает аппаратное или программное прерывание, микропроцессор вызывает подпрограмму, обрабатывающую конкретное событие. Механизм прерываний кажется несколько "окольным", но тем не менее он оказывается очень гибким и мощным.



Микропроцессоры 80x86 резервируют первые 1,024 байта ЗУПВ для таблицы, называемой таблицей векторов прерываний. Таблица содержит 256 адресов, по четыре байта на адрес. Первые два байта хранят значение сегмента, а следующие два — смещение в сегменте. Хранимые адреса представляют собой адреса мини-программ, называемых процедурами обслуживания прерывания (Interrupt Service Routine — ISR), которые вызываются для обработки событий типов 0–255.

Главное назначение DOS — предоставить программам разнообразные "услуги" или обслуживание. Программы запрашивают обслуживание, вызывая то или иное прерывание. С этой точки зрения DOS — это просто огромный набор ISR.

Когда возникает событие определенного типа, микропроцессор прерывает свои операции и обращается к таблице векторов прерываний по адресу, связанному с конкретным типом прерывания. После этого микропроцессор выполняет по этому адресу программу, которая знает, что делать для этого события. После окончания, программа сообщает микропроцессору о возобновлении прерванных действий.

Схематическое представление этого процесса показано на рис. 6.7. Оно опирается на многоуровневую модель, приведенную на рис. 6.2, и использует фактические адреса и размеры каждого уровня для описанного ранее ПК.

При первой загрузке компьютера программа запуска из ROM BIOS на материнской плате заполняет некоторые элементы таблицы векторов прерываний, заставляя их адресовать процедуры обслуживания прерываний, находящиеся в ROM BIOS на материнской плате. Позднее, при загрузке DOS с диска, она заполняет другие элементы (и изменяет часть элементов, заполненных программой запуска). Другие программы могут в любое время изменить элементы в таблице векторов прерываний. Часто такой прием является частью очень гибкого способа адаптации ПК. (Эта адаптация начинается при загрузке DOS и обработке файла конфигурирования CONFIG.SYS. Она продолжается при загрузке резидентных программ и может происходить в середине выполнения прикладной программы.)

Два класса программ изменяют содержимое таблицы векторов прерываний — резидентные программы и драйверы устройств.

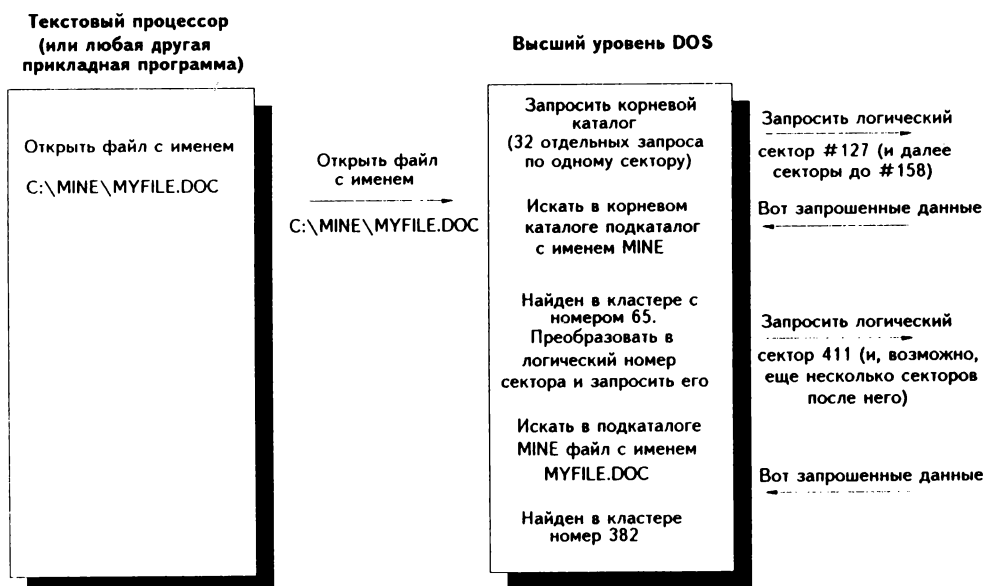


Рис. 6.3. Некоторые этапы обработки запроса файла (первая часть)

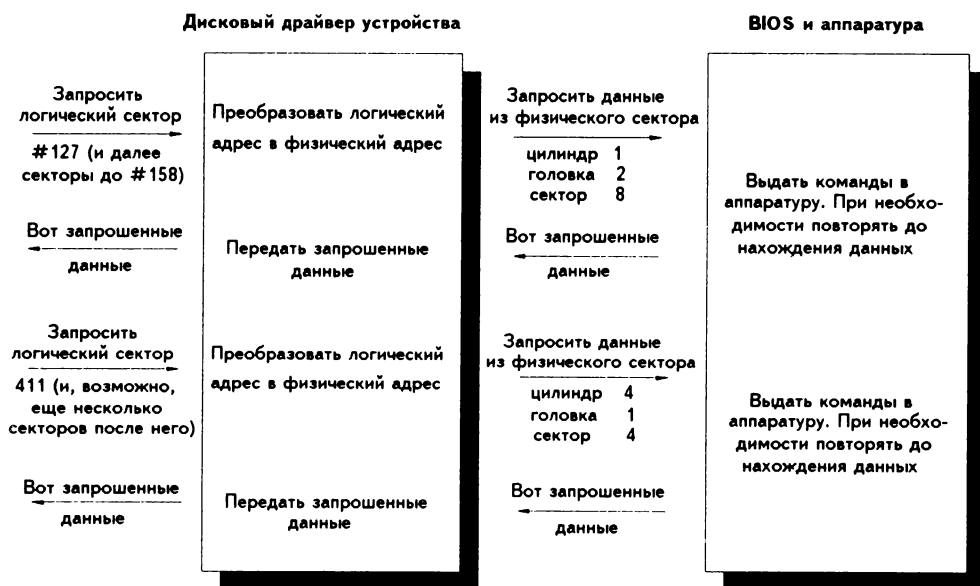
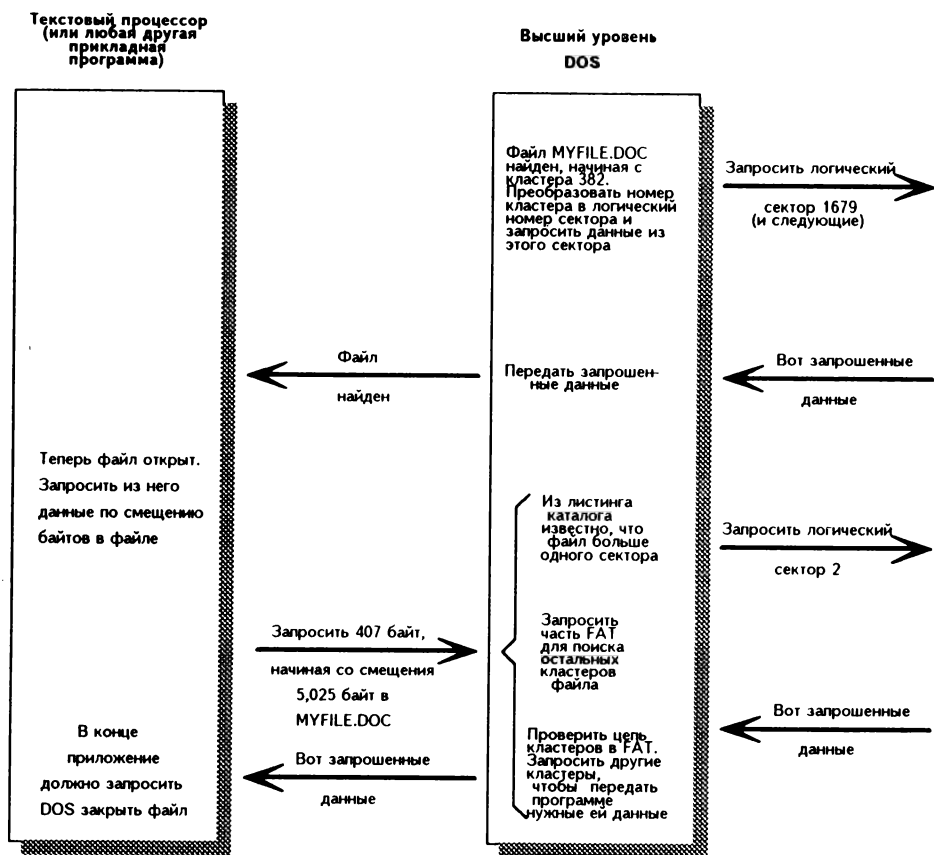
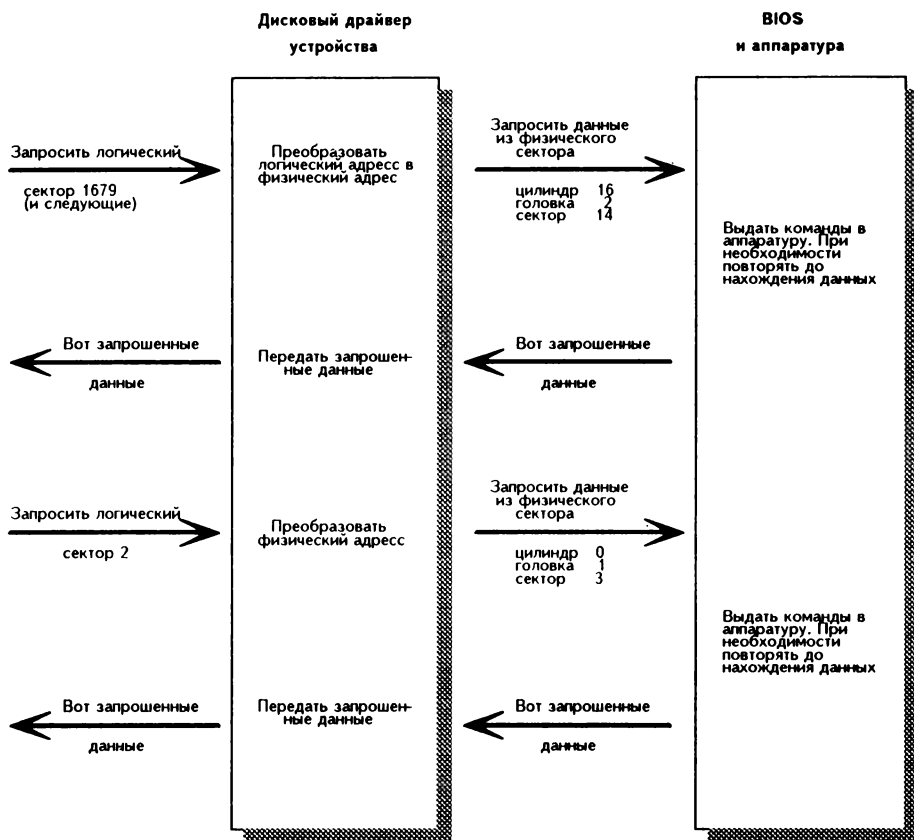


Рис. 6.4. Некоторые этапы обработки запроса файла (вторая часть)



Действие продолжается с рис.6.3 и 6.4 к рис.6.6

Рис. 6.5. Некоторые этапы обработки запроса файла (третья часть)



Действие продолжается с рис.6.3 - 6.5

Рис. 6.6. Некоторые этапы обработки запроса файла (четвертая часть)

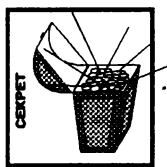
6.4.3. Резидентные программы

Резидентную программу (Terminate-and-Stay-Resident TSR) можно выполнить в ответ на стимул DOS, с помощью строки в файле AUTOEXEC.BAT (или любом другом пакетном файле) или с помощью строки INSTALL в файле CONFIG.SYS (последний способ возможен только в DOS 5.0). Названия таких программ объясняются их последним действием при загрузке. Они сообщают DOS о том, что работоспособны и управление ПК можно передать другой программе, но они не освобож-

дают занимаемую ими память. Следовательно, они заканчиваются (прекращают) управление компьютером, но остаются резидентными (остаются в памяти). Значимость такого действия будет понятнее дальше.

Каждая резидентная программа должна выполнить четыре действия. Во-первых, она должна скопировать в какое-то место внутри себя минимум один элемент таблицы векторов прерываний. Затем она должна поместить в этот элемент таблицы (или в несколько элементов, если она "цепляется" к нескольким прерываниям) адрес некоторой содержащейся в ней мини-программы. Непосредственно перед "засыпанием" резидентная программа должна сообщить DOS, какие части занятой ею памяти можно использовать повторно другими программами, а какие ей нужны. Последнее действие резидентной программы — сообщить DOS "Я закончилась,¹ но не выбрасывай меня — я буду работать позже".

Изменения, проведенные резидентной программой в таблице векторов прерываний, являются ее единственным шансом вновь получить управление компьютером. Без инициализации никакая резидентная программа не сможет сделать для вас ничего полезного.



До первого выполнения резидентной программы таблица векторов прерываний содержит элементы, показывающие на различные ISR. При первом выполнении резидентная программа анализирует конкретные элементы в таблице векторов прерываний, а затем копирует некоторые из этих чисел в безопасное место внутри себя. Затем резидентная программа помещает в эти элементы таблицы векторов прерываний другие адреса. Эти адреса служат указателями других мест внутри резидентной программы. В каждом таком определенном месте резидентная программа имеет мини-программу.

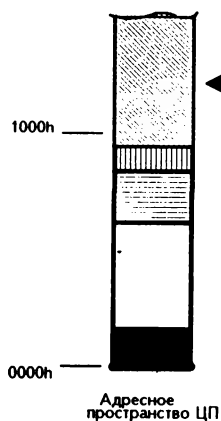
Наконец, резидентная программа выдает программное прерывание, которое сообщает DOS о том, что она закончилась, но "просит" DOS оставить ее в памяти и не записывать на то место другой программы. Как упоминалось выше, именно отсюда появилось название "закончиться и остаться резидентной".

Теперь, когда возникает одно из событий, к которому "прицепилась" резидентная программа, ПК прекращает текущие действия. ЦП оставляет в памяти указатель, показывающий, где он работал. После этого он вызывает программу, адресуемую элементом таблицы векторов прерываний, который соответствует возникшему событию, ожидая, что она и будет нужной для обработки данного события программой. Если резидентная программа заменила первоначальные числа в выбранном элементе таблицы векторов прерываний, ЦП не вызывает исходную ISR, а выполняет соответствующую мини-программу в резидентной программе.

Как и любая другая ISR, мини-программа выполняет то, на что она рассчитана. По окончании ISR ведет себя несколько по-другому. Обычная ISR заканчивается, сообщая ЦП о возврате к тем действиям, которые он выполнял до появления прерывающего события. Резидентная программа также может осуществить такое действие (если она полностью обработала вызвавшее прерывание событие), но может вызвать ту ISR, адрес из таблицы векторов прерываний которой хранится внутри резидентной программы и был заменен в таблице векторов прерываний ее адресом.

Теперь исходная ISR может выполнить то, что она обычно и делает. Когда она заканчивается, она предлагает резидентной программе вернуться к своим действиям, а затем резидентная программа возобновляет выполнение основной программы.

Такая ситуация возникает, когда резидентная программа "прицепляет" себя между обычной ISR и основной программой. Возможно, что таким образом сцепляются несколько резидентных программ. Такой способ, называемый сцеплением прерываний, оказывается мощным средством заставить ПК выполнять полезные действия.

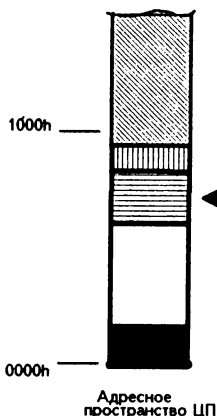


Предположим, что для появления прерывания ЦП выполняет программу пользователя. Следовательно, регистры сегмента кода (CS) и указателя команды (IP) дают текущий адрес программы где-то в программе пользователя адресного пространства (ЦП) (На рисунке принято, что CS : IP содержит 0F36 : 4257, что соответствует линейному адресу 135B7h или 77.43K, как показано стрелкой)

Когда возникает прерывание, ЦП вначале сохраняет текущий адрес программы и содержимое регистра FLAGS в стеке. Затем он переходит по адресу, содержащемуся в таблице векторов прерывания для данного типа (номера) прерывания. В показанном примере принято, что прерывание типа 9 (вызванное символом от клавиатуры) вызывает переход к CS : IP = 0D99 : 0045, что соответствует линейному адресу 0D9D5h или 54.46K, как показано стрелкой

A	F000 : EF6F
9	0D99 : 0045
8	0D99 : 003C
7	F000 : EF6F
6	F000 : EF6F
5	F000 : FF54
4	0070 : 06F4
3	0070 : 06F4
2	0D99 : 0016
1	0070 : 06F4
0	002A : 91FB

Содержимое таблицы векторов прерывания



Теперь текущий адрес программы показывает процедуру обслуживания прерывания (ISR), находящуюся в этом примере сразу ниже командного интерпретатора. Эту ISR загрузила DOS для обслуживания стандартной консоли (клавиатура и экран)

Когда ISR заканчивает свои действия, она выполняет команду IRET (возврат из прерывания). Она заставляет ЦП считать сохраненные значения "текущего адреса программы" и FLAGS. После этого ЦП возобновляет программу пользователя

Рис.6.7. Как векторы прерываний и процедуры обслуживания прерываний применяются для обработки событий

Объяснение может показаться сложнее, чем есть на самом деле. На рис. 6.8 и 6.9 показано, как одна резидентная программа "прицепляется" к одному прерыванию. Реальные резидентные программы часто "прицепляются" к нескольким прерываниям, и в памяти одновременно могут находиться несколько резидентных программ. Такую ситуацию очень сложно представить на рисунке. Однако ПК легко справляется и с такими сложными ситуациями.

6.4.4. Драйверы устройств

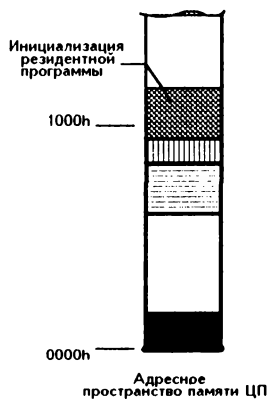
Еще один вид программ, изменяющих таблицу векторов прерываний, — это драйверы устройств. Драйвер устройства — это программа, которая воспринимает от DOS стандартные команды на передачу информации в устройство или из него и преобразует эти стандартные команды в ту форму, которая требуется для конкретного устройства. Такая программа загружается с помощью строки `DEVICE` или `DEVICENIGH` в файле `CONFIG.SYS` или маскируется как резидентная и загружается так же, как любая резидентная программа. (Например, строкой `INSTALL` в файле `CONFIG.SYS`, строкой в файле `AUTOEXEC.BAT`, строкой в некотором другом пакетном файле или прямым выполнением на стимул DOS.). Основное различие между драйверами устройств и обычными резидентными программами состоит в том, что, помимо изменения элементов в таблице векторов прерываний драйверы устройств самостоятельно превращаются в часть операционной системы.

DOS поддерживает так называемую цепь драйверов устройств. Когда ей требуется послать в устройство запрос на обслуживание, запрос проходит по цепи драйверов устройств. Первое устройство в цепи, которое распознает запрос DOS, воспринимает и выполняет его. Все сообщения об успехе или неудаче посылаются обратно по той же цепи.

Такой способ позволяет встраивать в цепь новые драйверы устройств и, если они оказываются впереди любых аналогичных драйверов, установленных ранее, только новые драйверы получают сообщения от DOS. По существу, новые драйверы устройств заменяют старые аналогичные драйверы.

Именно таким образом, например, драйвер `ANSI.SYS` может заменить принимаемую по умолчанию процедуру обработки консоли своей, более гибкой программой. Консоль является символьным устройством. Но таким же способом можно добавить и новый драйвер блочного устройства. Можно также, хотя это труднее, заменить принятый по умолчанию драйвер блочного устройства DOS. В конце главы показано, почему иногда очень важно заменить принятый по умолчанию драйвер блочного устройства DOS, и подробнее рассмотрено, как осуществить такую замену.

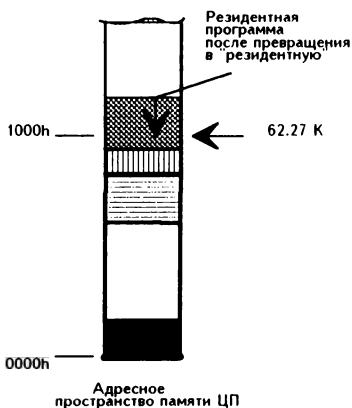
Фирма Intel предусмотрела до 256 типов прерываний. Поскольку ни один ПК не реагирует на столько событий (и потому, что в некоторых случаях многие события группируются в одно прерывание), часть таблицы векторов прерываний не используется. DOS помещает в это иначе "пропадающее пространство" адресные указатели таблиц информации, а не программ обработки прерываний.



Вначале резидентная программа действует так же, как любая другая программа. Загруженная выше командного интерпретатора, она может занимать значительную область памяти. При инициализации она сохраняет внутри себя адрес(а) одной или нескольких ISR из таблицы векторов прерываний. В этом примере сохраняется только адрес INT 9. После этого она заменяет это значение (или каждое из значений) указателем на себя, освобождает максимум памяти и сообщает командному интерпретатору, что она хочет "окончиться и остаться резидентной" (отсюда появилось название TSR - Terminate and Stay Resident)

A	F000 : EF6F
9	0D99 : 0045
8	0D99 : 003C
7	F000 : EF6F
6	F000 : EF6F
5	F000 : FF54
4	0070 : 06F4
3	0070 : 06F4
2	0D99 : 0016
1	0070 : 06F4
0	002A : 91FB

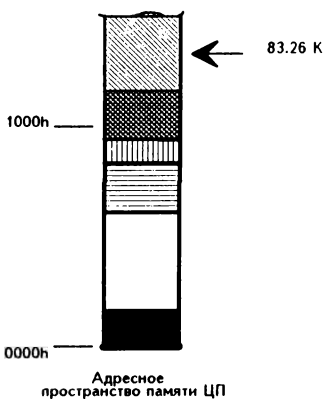
Элементы таблицы векторов прерываний до загрузки резидентной программы



Приложение пользователя загружается так же, как и прежде, но только выше в память, чем без резидентной программы. Отметим новый адрес для INT 9 (0F40 : 04EC = 62.27 K) в таблице векторов прерываний. Это точка входа резидентной программы

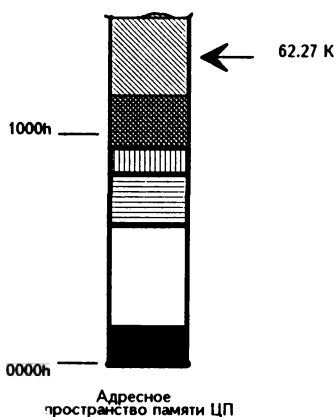
A	F000 : EF6F
9	0D99 : 0045
8	0D99 : 003C
7	F000 : EF6F
6	F000 : EF6F
5	F000 : FF54
4	0070 : 06F4
3	0070 : 06F4
2	0D99 : 0016
1	0070 : 06F4
0	002A : 91FB

Элементы таблицы векторов прерываний после загрузки резидентной программы

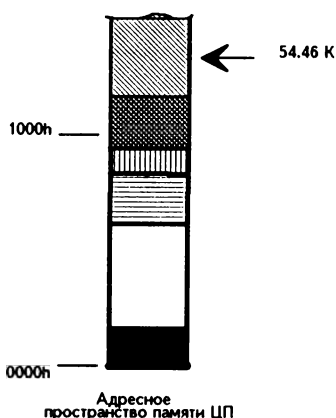


Рассмотрим, что происходит при возникновении прерывания клавиатуры. Пусть программа пользователя выполняется, например, по адресу 10AB : 4257. (Это тот же адрес, что и в предыдущем примере, но значение сегмента увеличено на размер памяти, занятой резидентной программой). Стрелка слева показывает текущий адрес программы 83.26 K.

Рис.6.8. Резидентная программа "прицепляется" к прерыванию для обработки только определенных событий, а другие события обрабатываются первоначальной процедурой обслуживания прерывания (первая часть)

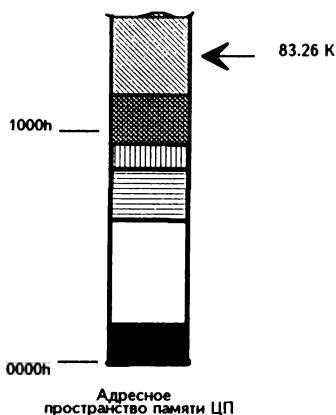


Предположим, что нажата клавиша. Возникает прерывание типа 9. ЦП прекращает свои действия, сохраняет текущий адрес программы и содержимое регистра EFLAGS, а затем обращается к таблице векторов прерываний для определения дальнейших действий. Так как в этой ячейке резидентная программа оставила свою точку входа, ЦП для обработки события вызывает резидентную программу. (Это означает, что новым значением текущего адреса программы будет 0F40 : 04EC или 62.27 K, как показано стрелкой слева)



Рассмотрим две ситуации. В первой ситуации резидентная программа видит, что вызвавшее прерывание клавишное действие является "горячей клавишей", которая активизирует эту резидентную программу. Поэтому она выводит окно или производит какие-то свои другие действия. По окончании этих действий программа выполняет команду IRET, заставляя ЦП возобновить выполнение приложения пользователя с той точки, в которой возникло прерывание

Во второй ситуации резидентная программа видит, что клавишное действие не является ее горячей клавишей. Здесь она должна передать управление ПК первоначальной ISR обработки ввода с консоли. Для этого она вызывает ISR по исходному адресу (54.46 K), который резидентная программа сохранила внутри себя (см. стрелку слева в середине карты памяти)

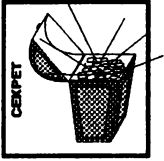


Когда ISR заканчивает свои действия, она выполняет команду IRET. Это вызывает возврат управления ПК резидентной программе, которая сразу же возвращает управление основной программе (по адресу 10AB : 4257 или 83.26 K)

Рис.6.9. Резидентная программа "прицепляется" к прерыванию для обработки только определенных событий, а другие события обрабатываются первоначальной процедурой обслуживания прерывания (вторая часть)

6.4.5. Таблицы параметров жесткого диска

DOS взаимодействует с разнообразными дисковыми накопителями, привлекая хранимые параметры, описывающие ключевые характеристики накопителей. Основная таблица параметров называется таблицей параметров жесткого диска (HDPT — Hard Disk Parameter Table). Большинство контроллеров диска ПК может управлять одним или двумя накопителями, поэтому в DOS предусмотрены две таблицы параметров жесткого диска. (Начиная с версии 5, DOS допускает до семи физических накопителей, хотя пока этой возможностью пользуются редко.)



Вы можете подумать, что в вашем ПК больше двух дисковых накопителей. Например, они могут называться С:, D: и E:. Эти накопители называются томами, или логическими накопителями. Они действуют так, как будто это отдельные дисковые накопители, но фактически они обычно являются разными разделами одного, максимум двух физических накопителей. Конечно, DOS знает, какие тома принадлежат каждому физическому накопителю.

Кроме таблиц параметров жесткого диска, каждая из которых описывает один физический накопитель, DOS поддерживает для каждого тома еще две таблицы (блок параметров накопителя и блок параметров BIOS). Эти таблицы подробнее рассмотрены в главе 10.

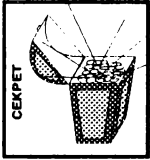
Разбиение жесткого диска на тома является одним из этапов подготовки его к работе. Подробнее об этом говорится в главе 9, а в главе 7 показаны причины создания таких томов.

DOS использует прерывания 41h и 46h для размещения в элементах таблицы векторов прерываний, которые обычно адресуют ISR, указателей первой и второй таблиц параметров жесткого диска. Если оба накопителя имеют одинаковые параметры, DOS обходится только одной таблицей параметров, помещая в оба элемента таблицы векторов прерываний одни и те же числа (рис.6.10).

Если ПК имеет только один физический накопитель, элемент INT 46h таблицы векторов прерываний показывает на одну HDPT в ROM BIOS на материнской плате; какая это таблица, не имеет значения, так как элементы КМОП-ЗУПВ конфигурации покажут отсутствие второго накопителя.

Шестнадцатеричное число 41h равно десятичному числу 65, а 46h — десятичному числу 70. Следовательно, два элемента в таблице векторов прерываний, которые показывают на таблицы HDPT, находятся в позициях с номерами 65 и 70. Поскольку для задания адреса программы ISR требуются четыре байта, фактические адреса указателей равны 00104h (десятичное число 260) и 00118h (десятичное число 280). Фактически каждый из этих адресов показывает только на первый из четырех байтов, необходимых для определения адреса соответствующей таблицы. Следующих три адреса содержат остальную часть адреса таблицы.

В таблице параметров жесткого диска DOS хранит числа цилиндров и головок (или поверхностей) для каждого накопителя, а также числа, показывающие, какую максимальную длину ошибки могут исправить коды ECC (они хранятся в секторе вместе с данными) и сколько раз повторять считывание сектора, который правильно с первого раза не считан. DOS также хранит некоторые значения тайм-аутов (они требуются для управления повторением некоторых операций контроллера), а также несколько довольно загадочных чисел: цилиндр, с которого начинается уменьшенный ток записи, и цилиндр, с которого начинается предкомпенсация записи.



Длина каждой таблицы параметров составляет 16 байт. Назначение каждого байта показано на рис. 6.11. Не все контроллеры жестких дисков используют все числа в этой структуре, но так как DOS не знает, с каким контроллером она работает, она заполняет все параметры.

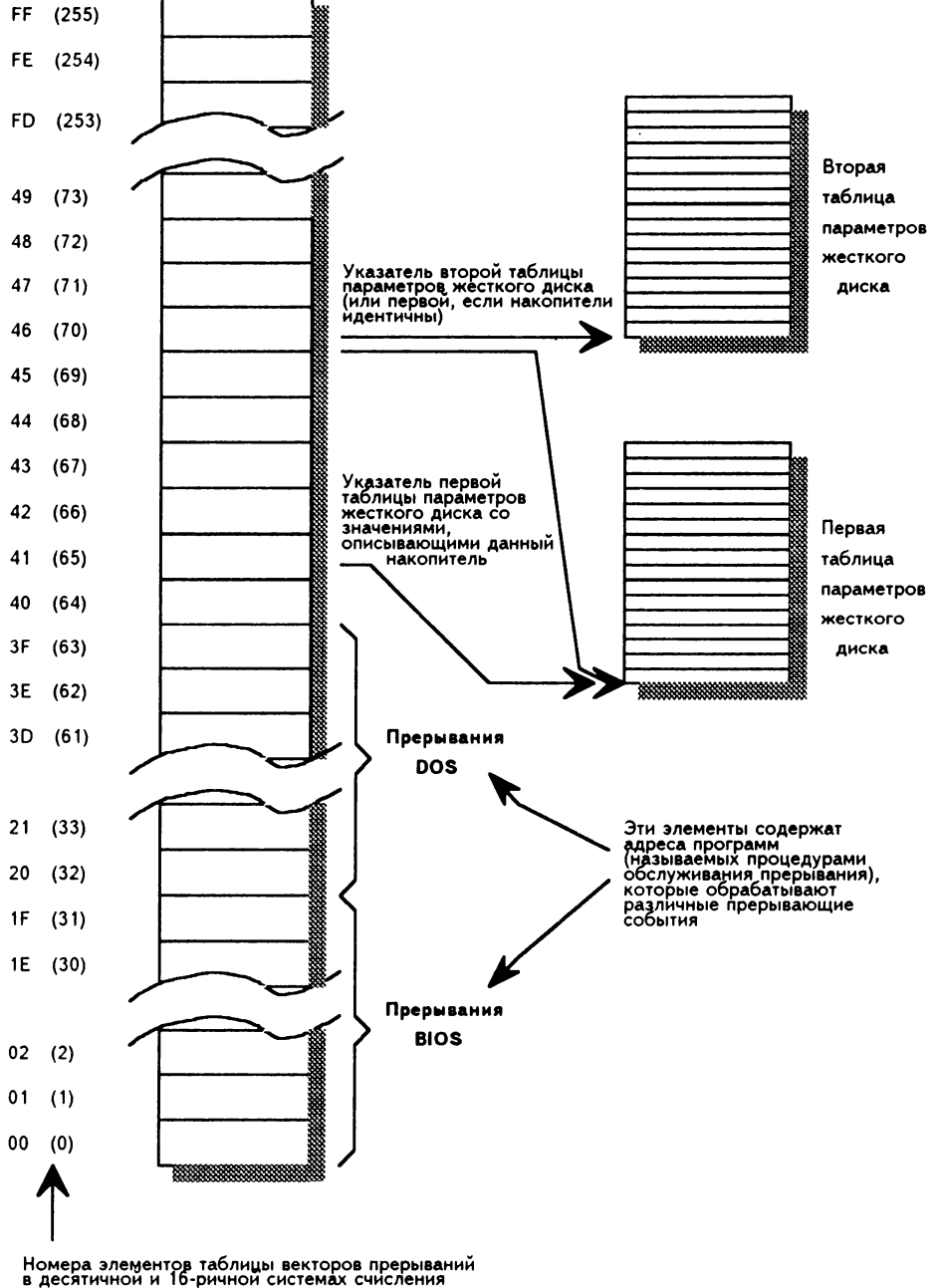


Рис. 6.10. Некоторые числа, хранимые в таблице векторов прерываний, показывают не на процедуры обслуживания прерываний, а на таблицы параметров жестких дисков

Таблицы параметров жестких дисков можно разместить почти в любом месте первого мегабайта адресного пространства памяти ЦП. Любая программа, которая хочет переместить таблицы, должна только изменить их указатели в таблице векторов прерываний; DOS от этого не пострадает.

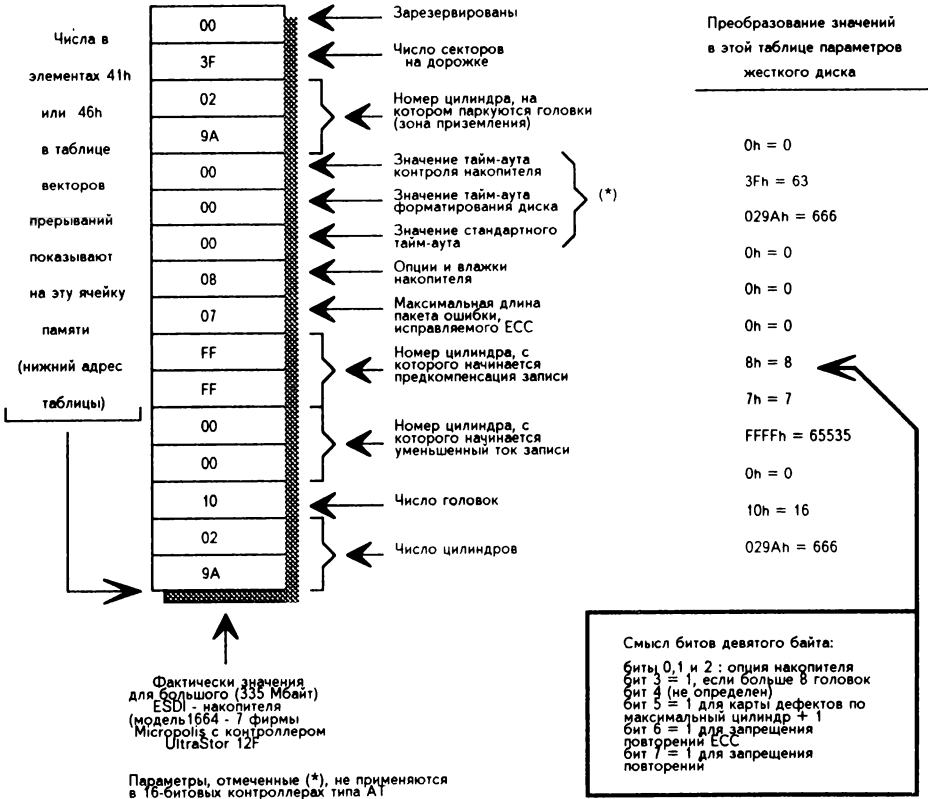


Рис.6.11. Структура таблицы параметров жесткого диска

16-ричное представление

В этой книге принято, что число, заканчивающееся буквой h, представлено в шестнадцатеричной системе. Иногда такие числа обозначаются завершающей буквой H или начальными символами 0x. (Если в каком-либо числе вы встречаете буквы от A до F, это число является 16-ричным. Ситуация усложняется, когда 16-ричное число состоит из одних букв, т.е. его можно спутать со словом. В этом случае первой цифрой числа указывается ноль. Но в этой книге все 16-ричные числа заканчиваются буквой h.)

Важно отметить, что без легкого изменения элементов HDPT было бы нельзя применять в ПК многочисленные типы жестких дисков. В следующем разделе рассмотрены ситуации, когда таблица параметров жесткого диска перемещается в новое место или когда создается новая таблица и превращается в активную путем помещения указателя на нее в таблицу векторов прерываний.

6.5. Определение типа накопителя

В вашем ПК может применяться один из нескольких способов контроля того, что таблицы параметров жестких дисков DOS содержат правильные числа. В следующих пунктах последовательно рассмотрены наиболее популярные способы.

6.5.1. Первый ХТ и аналогичные платы старых контроллеров

Контроллер жесткого диска ХТ имел несколько перемычек, с помощью которых определялись типы первого и второго подключенных накопителей. Было четыре позиции для перемычки первого накопителя и четыре для второго. Очевидно, в этом способе число выборов довольно ограничено.

При загрузке ХТ или клона ХТ программа начальной загрузки, получавшая управление, проводила много действий, включая и считывание перемычек. Полученная информация сообщала программе, какие числа поместить в таблицу векторов прерываний для адресации правильных таблиц параметров жесткого диска.

6.5.2. Платы новых автоконфигурирующих контроллеров ХТ

Большинство современных контроллеров диска ХТ работает с множеством типов накопителей. В процессе форматирования низкого уровня (глава 8) вы сообщаете контроллеру параметры накопителя (количество головок, цилиндров и секторов на дорожке), а также номера цилиндров, с которых начинается предкомпенсация записи и уменьшенный ток записи. Такие контроллеры фиксируют эту информацию в самом накопителе.

При загрузке ПК такой контроллер получает управление компьютером. После этого он считывает параметры первого накопителя из самого накопителя, создает таблицу параметров первого накопителя и помещает ее адрес в таблицу векторов прерываний; при наличии второго накопителя этот процесс повторяется. Возможность автоматически конфигурировать себя для правильной работы с тем накопителем, к которому они подключены, объясняет название — автоконфигурирующие контроллеры жесткого диска.

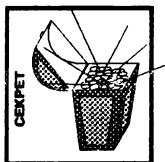
Куда контроллер помещает таблицы HDPT? Можно разместить HDPT в неиспользуемой части таблицы векторов прерываний, можно также поместить HDPT в ЗУПВ на плате контроллера, которое отображается на адресное пространство памяти ЦП в том же общем диапазоне адресов, что и ROM BIOS на плате контроллера.

Первая из приведенных возможностей проблематична. Иногда та область таблицы векторов прерываний, которую производитель контроллера считает свободной, в вашем ПК уже занята какими-то аппаратными средствами. В этом случае возникает серьезный конфликт. Второй выбор подходит для компьютеров класса XT, но не годится для новых машин (типа AT и с процессорами 386 и 486).

По-видимому, самый распространенный конфликт возникает между платой отображаемой памяти (EMS) и вторым физическим накопителем. Если вы заметите, что ваш второй жесткий диск становится "пьяным" сразу после загрузки программы менеджера отображаемой памяти (EMM), — у вас возник именно этот конфликт.

6.5.3. Компьютеры AT и старшие модели

Если у вас есть компьютер AT или старшая модель, то в нем есть так называемое КМОП-ЗУПВ конфигурации, которое хранит сведения о ПК, включая количество и тип дисковых накопителей (на гибких и жестких дисках), имеющихся в ПК.



КМОП-ЗУПВ выполнено по технологии, в которой для питания микросхем требуется незначительная мощность. Как и другие ЗУПВ, оно теряло бы свое содержимое при выключении ПК, но для его питания применяется небольшая батарея.

КМОП-ЗУПВ хранит дату и время дня, емкость памяти (ЗУПВ), тип дисплея и некоторую другую информацию о конфигурации ПК. Кварцевый генератор, также питаемый от батареи, постоянно обновляет время и дату. Вот почему компьютеры AT в отличие от XT (не имеющих КМОП-ЗУПВ) "просыпаются", зная правильные время и дату.

При включении AT он может "пожаловаться", что значения в КМОП-ЗУПВ неправильные: неправильные время или дата или неправильная информация о конфигурации (которая включает в себя и тип имеющихся накопителей).

Как компьютер узнает о неправильных времени или дате? А если сам компьютер следит за имеющимся оборудованием, как он узнает, что информация неправильная? Ответ заключается в том, что компьютер хранит контрольную сумму с информацией о конфигурации и еще одну контрольную сумму с временем и датой. При включении компьютера он вновь вычисляет контрольные суммы и сравнивает их с хранимыми значениями. Если суммы не совпадают, компьютер делает вывод, что возник сбой информации, хранимой в КМОП-ЗУПВ конфигурации.

Если испортились время или дата и вы работаете с MS DOS (или PC DOS) версии 5, ошибка поправима. На стимул DOS напечатайте DATE и нажмите клавишу Enter. Отвечайте на запрос ввода даты. Повторите операцию с командой TIME.

Если вы работаете со старой версией DOS, то заметите, что приведенные действия устанавливают только часы DOS, но не часы реального времени. Другими словами, ваш ПК знает правильные время и дату только до следующей перезагрузки.

В этом случае или при неправильной контрольной сумме конфигурации, вам придется выполнить программу настройки для исправления этих значений. В некоторых компьютерах для этого достаточно нажать определенную комбинацию горячих клавиш, в процессе загрузки или в любой момент времени. В других компьютерах (например, в IBM PC и PS/2) программа настройки находится на специальной дискете.

Если данные в КМОП-ЗУПВ хронически оказываются неправильными, может быть, придется заменить батарею или саму микросхему КМОП-ЗУПВ.

Для экономии емкости КМОП-ЗУПВ в нем обычно для каждого накопителя хранится всего одно количество, называемое типом накопителя. Компьютер хранит фактические параметры этого типа накопителя в таблице типа накопителя AT, находящейся в микросхемах ROM BIOS на материнской плате ПК.

В некоторых BIOS предусматривается, что пользователь сам определяет тип накопителя. Ввод этого номера, обычно 46 или 47, показывает BIOS, что вы хотите сообщ-

щить ему о нестандартном накопителе. После этого вы вводите физические параметры данного накопителя, и они сохраняются в КМОП-ЗУПВ вместе с номером типа накопителя.

6.5.4. Таблица типа накопителя AT

Таблица типа накопителя AT в ROM BIOS на материнской плате представляет совокупность многих таблиц параметров жестких дисков, по одной для каждого типа накопителя, о котором AT имеет информацию. Такой способ позволяет программе запуска просто посмотреть в КМОП-ЗУПВ, какие типы накопителей установлены (по номеру типа), а затем поместить в таблицу векторов прерываний адреса соответствующих элементов таблицы типа накопителя AT. После этого таблицы параметров жесткого диска готовы к работе.



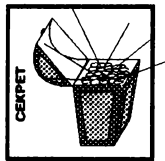
В первом компьютере AT предполагалось, что все накопители на жестких дисках известны и включены в многие таблицы HDPT в ROM BIOS. Конечно, такое предположение было нелепым.

В первых моделях компьютера AT сравнительно небольшая таблица содержала всего 14 типов накопителей. (Однако это было гораздо больше по сравнению с четырьмя типами в контроллере XT.) Позднее фирма IBM существенно расширила таблицу типа накопителя в ROM BIOS, и в некоторых моделях количество типов доходит до сотни, но все равно этого недостаточно.

А что если ваш накопитель не указан в главной HDPT в ROM BIOS на материнской плате? У вас есть несколько вариантов. Можно, например, заменить микросхему ROM BIOS новой версией, в которой таблица типа накопителя имеет один новый элемент, описывающий ваш накопитель. Конечно, это — выход из положения, но вам придется приобретать ROM BIOS при каждой замене накопителя, и такая перспектива малоприятна.

В некоторых контроллерах дисков, особенно для RLL-накопителей, реализован более удобный способ преодоления ограниченного количества типов накопителей в таблице типа AT. Этот способ такой же, как и применяемый в автоконфигурирующих контроллерах XT, рассмотренных в предыдущем разделе. Такие контроллеры полагают, что вы сообщите программе настройки тип установленного накопителя 1 (минимальную емкость).

Программа запуска BIOS вначале помещает в таблицу векторов прерываний указатели HDPT в ROM BIOS для типа накопителя 1. Позднее в процессе загрузки при так называемом сканировании съемного ПЗУ управление ПК получает программа расширения BIOS контроллера диска. При этом она считывает фактические параметры из самого накопителя, создает в ЗУПВ новые таблицы параметров жесткого диска и изменяет указатели для адресации новых таблиц.



Вскоре после включения питания ПК программа запуска проверяет наличие специальной видеоплаты. Обычно она проверяет по адресу C0000h наличие съемного ПЗУ. Обычно такое ПЗУ называется видеорасширением ROM BIOS, так как оно содержит программы, являющиеся дополнением BIOS (иногда это ПЗУ кратко называют видеоПЗУ).

Если программа запуска обнаруживает видеорасширение ROM BIOS, выполняет программа инициализации из этого ПЗУ. Она обеспечивает для BIOS правильный вывод на экран. (Платы всех адаптеров EGA и VGA для ПК имеют ПЗУ видеорасширения BIOS.)

Затем программа запуска определяет, сколько ЗУПВ имеет ПК, какие дисковые накопители, какой дисплей и другую информацию. Программа бегло тестирует обнаруженные компоненты — это называется самоконтролем при включении питания (POST — Power-On Self-Test).

По окончании POST программа запуска проверяет еще несколько областей памяти в поисках дополнительных микросхем ROM BIOS, содержащих расширения основного ROM BIOS. (Возможные области являются кратными 2К, начиная либо с C0800h, либо после видеоПЗУ, если оно было найдено ранее, и кончая адресом E0000h.)

При обнаружении каждого расширения ROM BIOS программа запуска вызывает программу инициализации этого ROM BIOS, которая всегда находится в стандартном месте. Эта программа может выполнить любые требуемые операции. По окончании она должна вернуть управление программе запуска, чтобы та продолжила поиск расширений и, в конце концов, загрузила операционную систему.

Физически микросхемы ПЗУ расширения BIOS обычно размещаются на съемных платах. Обычно процедура инициализации каждого съемного ПЗУ показывает свое присутствие сообщением на экране, а затем выполняет операции доступа к своим специальным аппаратным средствам.

Каждый компьютер XT имеет съемное ПЗУ для работы ПК с жестким диском. Это объясняется тем, что при разработке первого IBM ПК жесткий диск не был стандартным устройством компьютера. В компьютерах AT и всех последующих моделях поддержка жесткого диска включена в основное ROM BIOS на материнской плате. Но даже в этих ПК можно встретить контроллер жесткого диска со съемным ПЗУ.

Минимальное содержимое съемного ПЗУ является просто расширением основной таблицы типа накопителя AT. Более приятным и намного более гибким оказывается автоконфигурирующий контроллер, который приспособляется к накопителю почти с любыми параметрами, записывая параметры в сам накопитель.

Обнаружить наличие автоконфигурирующего контроллера можно по сообщениям, появляющимся на экране в процессе загрузки. Только очень немногие съемные ПЗУ не показывают своего присутствия при инициализации своих аппаратных средств и подключении к BIOS.

Необходимо знать об одной особенности автоконфигурирующих контроллеров: некоторые из них помещают HDPT в неподходящее место. Выбор неиспользуемой области таблицы векторов прерываний и использование ЗУПВ в том же диапазоне адресов, что и съемное ПЗУ, нельзя признать удачным.

Материал из раздела 6.5.2 применим и здесь. Поскольку ROM BIOS на материнской плате поддерживает жесткие диски, обращаться к съемному ПЗУ приходится только в процессе загрузки. После этого в хорошем компьютере (с процессором 386 и выше) можно переотобразить расширенную память на область съемного ПЗУ и обеспечить больше ЗУПВ в верхней памяти. Но этого нельзя сделать, если там находится HDPT. Лучше всего вначале загрузить драйвер MOVEHDD.SYS, который переместит HDPT в более подходящее место.

6.5.5. Жесткие диски на съемных платах

В жестком диске на съемной плате контроллер смонтирован на самой плате. Такие контроллеры знают все о подключенных накопителях и им ничего специального сообщать не нужно.

Необходимо информировать ПК об установке жесткого диска, но, как и в случае автоконфигурирующих контроллеров AT, можно немного "соврать". Вы просто сообщаете, что у вас есть накопитель типа 0. При загрузке плата жесткого диска определит истинный накопитель и создаст для него подходящую HDPT.

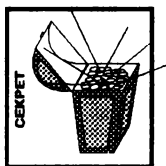
6.5.6. ESDI-накопители

Накопители и контроллеры ESDI намного "умнее" накопителей MFM и RLL а также контроллеров ST412/ST506. Накопители сами хранят свои параметры и готовы отвечать на стандартный запрос контроллера о параметрах.

Таким образом, установка накопителя ESDI проводится очень просто. Вы просто сообщаете о наличии накопителя типа 1 или типа 0. (Тип 0 означает, что накопитель не установлен.) Выбор зависит от имеющегося контроллера ESDI и содержится в прилагаемой к нему инструкции.

6.5.7. Накопители IDE/ATA

Сейчас для ПК наиболее популярны накопители IDE, но точнее их называть накопителями IDE/ATA. Это название означает Интегрированная электроника накопителя с соединением AT (подробнее см. в главе 3).



Строго говоря, любой накопитель со своим контроллером является IDE. К ним относятся все накопители SCSI, все жесткие диски на съемных платах и некоторые накопители ESDI (например, устанавливаемые в некоторые модели PS/2). К ним же относятся накопители для совершенно других компьютеров, например Macintosh. Однако эти накопители большинство людей не относят к накопителям IDE.

При добавлении букв ATA подразумевается накопитель для ПК, который имеет 40-контактное подключение к стандартной шине ввода-вывода AT через разъем ATA. Этот разъем и передаваемые сигналы определены в промышленном стандарте, называемом спецификацией соединения AT. Только накопители, рассчитанные на этот интерфейс, можно называть накопителями IDE/ATA.

В таких накопителях, как и в накопителях на съемных платах, контроллер встроен в накопитель. В отличие от накопителей на съемных платах, накопители IDE/ATA монтируются в стандартном посадочном месте. Имеются два способа подключения их к шине ввода-вывода ПК: через специальный 40-контактный разъем или через переходную плату, которая вставляется в шину и имеет 40-контактный разъем (на переходной плате может также находиться контроллер гибкого диска).

Пока вроде нет ничего особенного, но подождите. В таких накопителях есть одна необычная особенность, которая легко может запутать вас, если о ней не знать.

Накопители IDE/ATA — это настоящие "хамелеоны". Они могут подстроиться почти под любые предлагаемые параметры. (Конечно, накопитель IDE/ATA имеет определенную максимальную емкость, но ее можно распределить по цилиндрам, головкам и секторам на дорожке почти любым способом.)

При установке накопителя IDE/ATA можно выбрать любой тип накопителя, если его емкость не больше емкости IDE/ATA. Вы вводите номер типа накопителя в КМОП-ЗУПВ конфигурации. При следующей загрузке компьютера программа запуска помещает в элементы 41h (и 46h при наличии двух накопителей) таблицы векторов прерываний указатели на таблицы HDPT для указанных типов накопителей.

Когда накопитель IDE/ATA начинает работать, он проверяет эти указатели и обращается к соответствующей HDPT. После этого он настраивает отображение секторов (или преобразование секторов) так, чтобы выглядеть накопителем того типа, о котором вы сообщили. (Отображение секторов подробно рассматривается в главе 7; а также дальше, в материале по накопителям SCSI.)

Такая ситуация представляется замечательной. Вы не должны знать истинные параметры установленного накопителя IDE/ATA. Все что вам нужно знать — максимальная емкость.

Но эта же ситуация оказывается и ужасной, особенно, если вы забыли тип накопителя, а содержимое КМОП-ЗУПВ испортилось. В отличие от других типов нако-

пителей, накопитель IDE/ATA не может помочь вам узнать, какие параметры были у него при последнем использовании. Если при загрузке компьютера элемент КМОП-ЗУПВ для жесткого диска изменился и показывает другой тип накопителя, накопитель IDE/ATA "бодро" посчитает, что вам требуется другое отображение секторов и послушно выполнит "ваше желание".

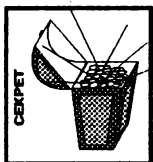
В результате вы не сможете обратиться к данным на этом диске до определения того типа, который действовал при размещении данных. В главе 11 показано, как выйти из этого затруднительного положения.



Для защиты лучше всего записать тот тип накопителя, который вы выбрали при установке накопителя. Запишите его на листочке и приклейте к самому накопителю. Запишите эту информацию в перечень деталей вашего ПК и держите его в легкодоступном месте.

6.5.8. SCSI-накопители

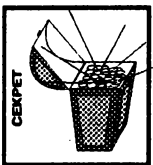
Накопитель SCSI — это накопитель IDE (в широком смысле это термина) плюс интерфейс ведомого SCSI. Накопители SCSI подключаются к ПК с помощью трех типов главных адаптеров SCSI, каждый из которых выглядит для DOS несколько по-другому.



Рассматриваемые далее типы характерны для всех съемных плат для управления блоковым устройством DOS, а не только для главных адаптеров SCSI. Примерами других плат могут служить контроллеры оптических дисков и некоторых ленточных накопителей.

Интерфейс SCSI, как и высший уровень DOS, считает жесткий диск просто большим количеством секторов. При этом для указания позиции в накопителе требуется только одно количество — логический адрес блока (LBA — Logical Block Address). (Это количество эквивалентно логическому номеру сектора в DOS, но, возможно, со смещением, учитывающим неиспользуемую DOS область диска.)

DOS и BIOS взаимодействуют с жестким диском на физическом уровне, привлекая для определения позиции на диске номера цилиндра, головки и физического сектора. Три типа главных адаптеров SCSI различаются, в основном, преобразованием адресной информации о жестком диске от BIOS в стандарт SCSI.



Поскольку совместимый по регистрам главный адаптер SCSI выглядит стандартным контроллером жесткого диска AT, компьютер взаимодействует с ним командами контроллера MFM. Однако адаптер отличается от стандартных контроллеров AT тем, что он не должен превращать управляемые накопители в накопители, соответствующие таблице типа накопителя AT. Для этого он должен быть автоконфигурирующим контроллером и создавать свою таблицу параметров жесткого диска в ЗУПВ в процессе сканирования съемного ПЗУ при загрузке, а не использовать элементы таблицы типа накопителя AT в ROM BIOS на материнской плате.

Большинство главных адаптеров SCSI не совместимо по регистрам, но все они совместимы по INT 13h. Эти контроллеры имеют съемное ПЗУ, которое переадресует элемент прерывания 13h в таблице векторов прерываний на программу внутри самого себя. Следовательно, контроллер совсем не использует программу BIOS из ПЗУ на материнской плате. (Вектор прерывания INT 13h хранится в двадцатой позиции таблицы, а в первой позиции хранится вектор INT 0.)

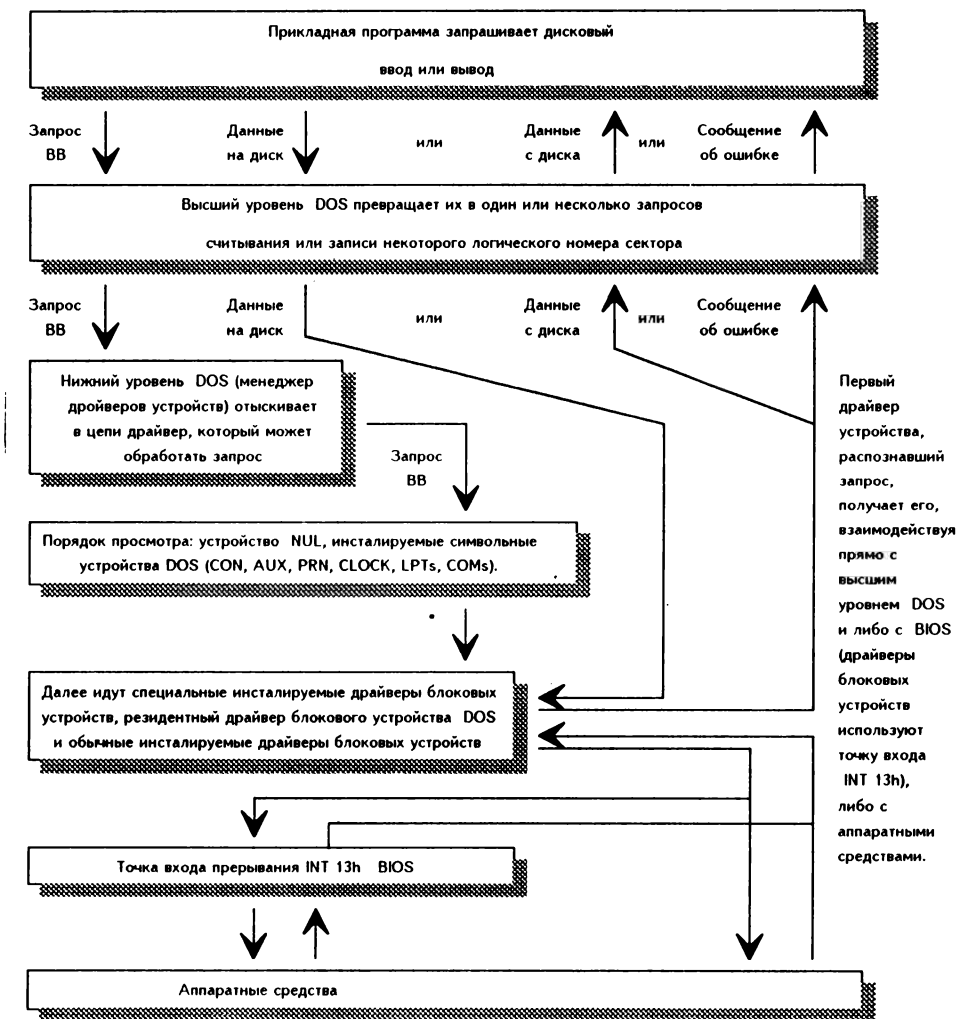
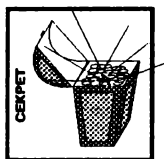


Рис. 6.12. Как устанавливаемый драйвер блочного устройства, предназначенный для замены блочного драйвера DOS, вставляет себя в DOS



Совместимый по INT 13h главный адаптер SCSI представляет компьютеру фиктивный накопитель, имеющий некоторые удобные количества цилиндров, головок и физических секторов на дорожке. Он преобразует все запросы INT 13h в логический номер сектора и посылает преобразованные запросы в ведомый SCSI-адаптер. Ведомый адаптер, в свою очередь, преобразует логические адреса блоков в номера реальных цилиндра, головки и физического сектора с учетом настоящих параметров накопителя. (Подробный пример преобразования рассмотрен в главе 7.)

Третий тип главных адаптеров SCSI не совместим со стандартным контроллером жесткого диска AT ни по регистрам, ни по INT 13h. Для работы с таким адаптером требуется устанавливаемый драйвер устройства. Программа драйвера может находиться в съемном ПЗУ главного адаптера или загружаться в ЗУПВ с загрузочного диска. Цель любого устанавливаемого драйвера устройства состоит в том, чтобы заменить нижний уровень DOS. В этом отношении драйвер устройства напоминает многие резидентные программы, но он "подцепляется" к DOS по-другому.

Главный адаптер SCSI не единственное устройство, которое требует применения устанавливаемого драйвера устройства. Например, драйвер необходим для работы с накопителем CD ROM. Отличие состоит в том, что устанавливаемый драйвер устройства для главного адаптера SCSI должен заменять принимаемый по умолчанию стандартный драйвер блокового устройства DOS (т.е. тот драйвер, который DOS обычно использует для обращения к накопителю C). Имеются некоторые тонкости обращения DOS к диску, что несколько усложняет разработку такого драйвера.

Независимо от имеющегося у вас типа главного адаптера SCSI установка жесткого диска проводится аналогично. Обычно вы сообщаете программе настройки, которая записывает в КМОП-ЗУПВ конфигурации, что у вас нет жесткого диска. (При добавлении накопителя SCSI к ПК, имеющему другой накопитель, можно оставить настройки в КМОП-ЗУПВ пустой.)

ROM BIOS главного адаптера или устанавливаемый драйвер устройства запрашивают у накопителя SCSI его емкость. Главный адаптер сообщает BIOS о наличии жесткого диска с 64 головками и 32 секторами на дорожке. Это соответствует емкости цилиндра 1 Мбайт, поэтому главный адаптер сообщает о емкости накопителя как о количестве цилиндров. (Такие сообщения главного адаптера нужны для того, чтобы поместить значения в таблицу параметров жесткого диска и адресовать ее через соответствующий вектор прерывания.)

Когда DOS не следует своим правилам

DOS имеет два прерывания для обращения к диску на логическом уровне: INT 25h (считывание сектора) и INT 26h (запись сектора). Как ни странно, DOS не использует их. Однако любая другая программа может применить эти прерывания и получить такой же результат, какой получает и DOS по другому способу.

Высший уровень DOS (или внешняя программа) может обратиться к позиции на диске, определяя только логический номер сектора. Резидентный драйвер блокового устройства DOS обычно преобразует такие запросы в запросы с указанием реальных физических адресов (номеров цилиндра, головки и физического сектора). В конце концов, для фактического обращения к диску этот драйвер вызывает BIOS через прерывание INT 13h.

DOS обращается к драйверу абсолютным вызовом по известному ей адресу. Изменение любого элемента в таблице векторов прерываний (например, для INT 25h или INT 26h) не разрешает вам помешать этому действию.

На нижнем уровне DOS поддерживается связанный список драйверов устройств. DOS хранит адрес только первого драйвера, а каждый драйвер адресует следующий в цепи. Имеется стандартный способ вставлять символьные устройства в начало цепи, а блоковые устройства в ее конец. Однако нет стандартного способа вставить блоковое устройство в начало цепи.

DOS при поиске драйвера устройства, способного выполнить нужную ей задачу, просматривает цепь по порядку, поэтому инсталлированное символьное устройство может легко заменить резидентное, а инсталлированное блоковое устройство обычно не может. Тот, кому требуется заменить резидентный драйвер блокового устройства DOS, должен предпринять довольно нестандартные действия.

Производители главных адаптеров SCSI третьего типа должны разрабатывать инсталлируемые драйверы устройств, которые "вставляют" себя впереди обычного резидентного драйвера блокового устройства DOS (рис. 6.12). Обычно они также "прицепляются" к прерыванию INT 13h, чтобы реагировать на дисковые запросы и на этом уровне.

Поскольку контроллер SCSI считает диск одномерной цепочкой секторов, его специальный драйвер блокового устройства решает простую задачу на уровне INT 25h и INT 26h. Для получения логического адреса блока он просто прибавляет некоторое смещение к логическому номеру сектор DOS. Когда он перехватывает вызов INT 13h, контроллер должен выполнить задачу, обратную применительно к резидентному блоковому устройству DOS, преобразуя трехмерный дисковый адрес в одномерный.

Заключение

- MS DOS — это операционная система для ПК (ее "родственниками" являются системы PC DOS и DR DOS). DOS образует несколько программных уровней между аппаратными средствами и прикладными программами. Ее задача — управлять доступом к аппаратным средствам всех выполняемых программ и "активизировать" аппаратные средства по запросам программ.
- То, что обычно называют DOS, можно разделить на две части — BIOS и "настоящая" DOS. Первая часть, находящаяся в ПЗУ или считываемая из дисковых файлов, взаимодействует с аппаратными средствами. "Настоящая" DOS представляет данные файлами и управляет использованием диска и других устройств.
- Все источники и получатели данных DOS считает устройствами и управляет ими двумя способами: один — для символьных и второй — для блоковых устройств.
- Символьные устройства оперируют байтами (символами). Примерами символьных устройств служат клавиатура, экран, мышь, модем и принтер.
- Блоковые устройства оперируют блоками информации, обычно состоящими из 512 байт. Примерами блоковых устройств являются все дисковые накопители, ленточные накопители, накопители CD ROM и др.
- Программы в ПК разделены на уровни, причем нижние уровни поддерживают верхние. Прикладные программы вызывают DOS для решения своих разнообразных задач. В свою очередь, DOS вызывает BIOS, которая и взаимодействует с аппаратными устройствами.
- В эту структуру DOS могут вмешаться драйверы устройств и резидентные программы и изменить действия DOS для программ.
- Основным механизмом взаимодействия между программными уровнями в ПК служит прерывание. Второй способ обеспечивается с помощью драйверов устройств.
- DOS использует также разнообразные таблицы данных. Важнейшей таблицей для операций на диске является таблица параметров жесткого диска (HDPT). Имеется одна HDPT для каждого физического накопителя, содержащая важную информацию о накопителе (количество цилиндров, головок и секторов на дорожке).
- В первых компьютерах IBM PC/XT и PC/AT микросхемы ROM BIOS жесткого диска содержали информацию о всех типах накопителей, которые можно

было установить в ПК. В современных ПК применяются несколько способов получения информации об установленных жестких дисках.

- Лучше всего поместить параметры диска в сам накопитель и заставить контроллер считать их во время загрузки. Этот способ позволяет применять любой накопитель, интерфейс которого совместим с контроллером.
- Накопители IDE/ATA являются "хамелеонами". Они представляют себя имеющими те параметры, которые вы пожелаете. Иногда это прекрасно, но может оказаться большой бедой, если вы забыли сообщаемый тип накопителя, — данные восстановить очень трудно.

Зная, как работает с дисками DOS, вы готовы познакомиться с тем, как эти способы ограничивают емкость диска. В следующей главе рассмотрены такие ограничения и приемы их преодоления.

Глава 7. Преодоление барьеров

В этой главе вы узнаете

- Каким образом особенности DOS и первого ПК ограничивают максимальную емкость дисков
 - Какие ограничения относятся к логическим накопителям (буквам накопителей), а какие — к физическим
 - Как обойти ограничения при работе в версиях DOS до 3.3
 - В DOS 3.3 реализован лучший способ преодоления этих ограничений
 - В DOS 4 появился еще более лучший способ преодоления ограничений и его применение
 - Про сообщения об ошибках, которые выдают контроллер диска и BIOS, когда обращение к жесткому диску безуспешно; как и почему DOS подавляет некоторые из этих сообщений
 - Как "умная" программа может обойти DOS и перехватить все сообщения о дисковых ошибках и почему это важно для вас.
-

7.1. Логические ограничения DOS на емкость диска

До версии 4 DOS не могла работать с томом жесткого диска больше 32 Мбайт. (Дисковый том в данном контексте подразумевает то, что для DOS выглядит дисковым накопителем, т.е. сущность, которой DOS назначает букву накопителя.)

На высшем уровне DOS считает дисковый накопитель просто длинной цепочкой секторов. Она нумерует секторы, начиная с единицы. Это логическое представление дискового накопителя. Позже, когда DOS готова скомандовать аппаратным средствам что-то считать или записать, она должна преобразовать логический номер каждого сектора в реальный физический адрес (головку, цилиндр и физический номер сектора).

Высший уровень DOS имеет встроенное ограничение (или предел) на число, представляющее логический номер сектора. Это ограничение определяет максимальную емкость (или размер) любого логического тома, с которым может работать DOS. Процедуры нижнего уровня (в BIOS) имеют другой набор ограничений, которые совместно определяют другую максимальную емкость для любого физического накопителя, подключенного к PC. Не требуется, чтобы эти две максимальные емкости диска были одинаковыми, и, в общем, они не совпадают. В этом разделе показано, откуда вытекает ограничение DOS и какую максимальную емкость оно подразумевает. В следующем разделе речь пойдет об ограничениях BIOS и их влиянии.

Во всех версиях DOS до 4.0 логический номер сектора должен помещаться в 16-битовое число. Это означает, что в адресуемой DOS части дискового накопителя (одном дисковом томе) не может быть больше 65,536 (2^{16}) секторов.

В DOS стандартный размер всех секторов составляет 512 байт. Этот размер, умноженный на максимальное число секторов, дает 32 Мбайт, т.е. максимальную емкость, которую DOS могла адресовать в одном дисковом томе до DOS 4.0.

В DOS 4.0 верхнее ограничение на номер сектора увеличено. Теперь вместо 16-битового числа номер сектора представляется 32-битовым числом (до 4,294,967,296). Поэтому максимальная емкость диска со стандартными 512-байтовыми секторами со-

ставляет 2 Тбайт (примерно 2 миллиона миллионов байт). По крайней мере сейчас такая емкость кажется достаточной!

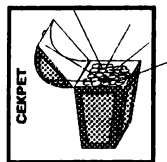
Ограничение 32-битового номера сектора определяет максимальную логическую емкость диска (или, точнее, логического тома). Другой набор ограничений, называемых физическими ограничениями, более серьезен и не зависит от применяемой версии DOS. В следующем разделе показано, что ограничение на емкость диска, определяемое официальными ограничениями BIOS на физическую адресацию, составляет 504 Мбайт. Есть широко распространенный способ увеличения этой емкости, но даже при этом имеется ограничение на емкость любого жесткого диска PC — 7,875 Гбайт. В любом случае это намного меньшее ограничение, чем логическое ограничение емкости дискового тома для новых версий DOS (DOS 4.0)

Задолго до появления DOS 4.0 конструкторы захотели превысить барьер емкости в 32 Мбайт и разработали для этого несколько способов. Применение нестандартных способов может привести к неприятным последствиям (раздел 7.3).

7.2. Физические ограничения BIOS на емкость диска

Когда программа хочет считать некоторую информацию с диска или записать ее на диск, этот запрос, в конце концов, проходит вниз до физического уровня. Он начинается как запрос файла, который DOS преобразует в несколько обращений к конкретным кластерам или логическим секторам в накопителе. Наконец, до того, как скомандовать аппаратуре провести требуемые считывание или запись, каждый из логических адресов должен превратиться в конкретные цилиндр и головку (они определяют дорожку) и в конкретный сектор на этой дорожке.

Такое преобразование осуществляет драйвер блочного устройства. Имеется принимаемый по умолчанию драйвер блочного устройства, который является частью кода BIOS, загружаемого DOS с диска, но можно его заменить или модифицировать его действия с помощью устанавливаемого драйвера устройства, загружаемого при выполнении файла CONFIG.SYS или позже.



Это преобразование осуществляет высший уровень DOS, вызывая драйвер устройства (резидентный или заменяющий). В любом случае вызванная программа преобразует запрос в физический адрес, а затем вызывает BIOS через подфункцию 5 или 6 прерывания INT 13h (рис. 6.12).

Когда программа вызывает обслуживание прерывания, номер прерывания определяет, какая процедура обслуживания прерывания ISR будет обрабатывать запрос. Чтобы обеспечить больше обслуживаний без риска нехватки номеров прерываний, общепринято разрабатывать процедуры обслуживания прерываний, которые выполняют большее число различных действий. Такие различные действия называются подфункциями ISR. Вызывающая ISR программа показывает нужную подфункцию, помещая номер этой подфункции в регистр AH центрального процессора до выдачи прерывания.

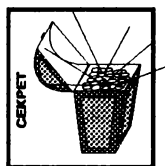
Числа, которые драйвер устройства привлекает для преобразования, являются параметрами накопителя. Драйвер устройства узнает эти параметры во время загрузки из другого вызова INT 13h с подфункцией 8.

После преобразования местонахождения в фактический физический адрес запрос передачи данных должен поступить в BIOS, которая, в свою очередь, посылает запрос в схемы контроллера диска.

Вызов прерывания, который DOS использует для передачи информации BIOS (INT 13h), имеет несколько жестких ограничений на величину чисел, допустимых для компонентов физического адреса. Для определения цилиндра допускаются

только 10-битовые числа, головки — 4-битовые и сектора — 6-битовые числа. Такое ограничение означает, что можно иметь номера цилиндров от 0 до 1023, головок — от 0 до 15 и секторов — от 1 до 63. Все сверх этого может существовать, но невозможно через BIOS передать такую информацию контроллеру диска.

Если перемножить приведенные числа (1024 цилиндра на 16 головок на 63 сектора и на 512 байт в секторе), окажется, что общая максимальная емкость любого физического накопителя на жестких дисках в полностью стандартном ПК составляет 504 Мбайт.



Фактически только предполагается, что номер головки должен находиться в диапазоне от 0 до 15. Те места в DOS и BIOS, которые хранят реальный максимальный номер головки, допускают числа до 255. Ничто в DOS и BIOS не контролирует, является ли номер головки меньше 16; в документации только сказано, каким он предполагается. Это важная лазейка в, казалось бы, непроницаемом наборе ограничений емкости. Об этой лазейке и других способах преодоления ограничений емкости DOS (логических) и BIOS (физических) речь пойдет дальше.

Даже если допустить максимум в 256 головок, то максимальная емкость жесткого диска ПК составит 7.875 Гбайт (примерно 8,455 млн байт). Это не слишком большое значение для современных накопителей, устанавливаемых в ПК. Все больше пользователей столкнутся с этим ограничением в ближайшем будущем.

Для первых ПК ограничения BIOS на емкость жесткого диска не казались особенно серьезными, но с появлением новых накопителей они становятся неприемлемыми. Несмотря на возможность преодолеть ограничения несколькими интересными способами, такие действия имеют серьезные последствия, о которых сказано в разделе 7.4. Перед обсуждением способов преодоления физических ограничений вернемся к логическим ограничениям и рассмотрим некоторые способы их преодоления.

7.3. Преодоление логических ограничений

В разделе 7.1 показано, что DOS ограничивает логическую емкость накопителей до 32 Мбайт. Начиная с DOS 3.3 можно использовать накопители большей емкости, но только разделяя их на несколько логических томов, емкость каждого из которых не превышает 32 Мбайт. Начиная с DOS 4.0 можно использовать дисковый накопитель почти любой емкости как единый логический том или разбить его на несколько томов по желанию пользователя.

Такие усовершенствования в DOS были реакцией на нужды пользователей и хорошие идеи решения проблемы емкости нескольких фирм, выпускающих съемные платы и программы-утилиты. В этом разделе рассмотрены первые приемы работы с жесткими дисками большой емкости.

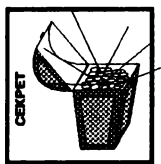
7.3.1. Большие секторы

До DOS 4.0 нельзя было иметь больше 65,536 секторов в адресуемом DOS дисковом томе. Один из приемов работы с диском большей емкости, чем 32 Мбайт без нарушения этого ограничения, состоит в том, чтобы просто сделать каждый сектор больше 512 байт. DOS допускает это, но требует некоторой помощи. Размер сектора может меняться, но принято использовать размеры, кратные 512 байтам. Обычно используют секторы с размером 1,024 байта, 1,536 байт, 2,048 байт и т.д. до 8,192 байта.

Одно из серьезных последствий этого приема заключается в потенциальном снижении эффективности использования дискового пространства. Любой файл занимает минимум один сектор и, даже несколько секторов (если кластер состоит из не-

скольких секторов). Это справедливо независимо от того, насколько малым является файл. При хранении на диске с большими секторами множества крошечных файлов пропадает значительное дисковое пространство. Пространство в каждом кластере за концом файла называется пропадающим пространством. Большие секторы (и, следовательно, большие кластеры) увеличивают пропадающее пространство. Подробнее о размере кластеров и вызываемой ими неэффективности сказано в главе 10.

Второе важное последствие использования сектора, размер которого больше стандартного, связано с тем, что вам потребуется не-DOS программа для задания выбранного нестандартного размера сектора. Совместимость вашего компьютера несколько ухудшается и это может вызвать проблемы.



Инженеры из фирмы Wyse Technology нашли способ работы DOS с дисковыми накопителями больше 32 Мбайт в их системе MS DOS 3.21. Этот способ просто позволяет программе FORMAT записывать большие секторы (или, точнее, то, что выглядит для DOS большими секторами). Они применили этот же способ в версии 3.3. Начиная с Wyse MS DOS 4.0, они перешли на способ фирм IBM и Microsoft работы с большими дисковыми разделами с использованием только стандартных 512-байтовых секторов.

Размер сектора хранится в загрузочной записи любого диска, отформатированного последними версиями DOS. Поэтому любая версия Wyse MS DOS после 3.21 может распознавать больший размер сектора фирмы Wyse для работы с большими дисками. Версии MS DOS других производителей могут не работать с такими дисками.

Фактические физические секторы создает контроллер жесткого диска, когда в процессе форматирования низкого уровня он помещает в секторы точно 512 байт данных. Секторы большего размера являются просто логическими секторами, создаваемыми из меньших физических секторов специальной версией программы FORMAT системы Wyse DOS, когда она осуществляет форматирование высокого уровня.

Фирма Compaq, начиная со своей MS DOS 3.31, разработала специальный способ работы с большими дисковыми разделами. Позднее она сообщила, что этот способ разработан в тесном сотрудничестве с фирмой Microsoft. Именно этот способ использован фирмами IBM и Microsoft в версии 4.0 PC DOS и MS DOS. (Это было приятной новостью для пользователей компьютеров фирмы Compaq, так как не требовало ничего специального для дисков при полной совместимости с новыми версиями DOS.)

Если вы применяете специальный драйвер для модификации DOS, устанавливая больший размер сектора, вы идете на определенный риск. При потере программы драйвера вы не сможете обратиться к данным на жестком диске. Кроме того, любое отступление от DOS потенциально вызывает неприятное взаимодействие с некоторой прикладной программой DOS, которая ожидает, что вашей операционной системой является чистая DOS.

7.3.2. Дисковые разделы

Еще один распространенный способ преодоления ограничения в 32 Мбайт заключается в разбиении большого накопителя на несколько логических томов, емкость каждого из которых меньше 32 Мбайт. DOS 3.3 позволяет это сделать по способу фирм IBM и Microsoft. До этого для преодоления ограничения требовались дополнительные программы, расширяющие базовые процедуры DOS. Подробности создания логических накопителей в DOS 3.3+ рассматриваются в главе 10.

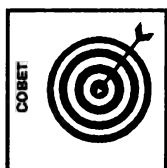
Каковы последствия использования логических накопителей? Если ваш дисковый накопитель первоначально является одним томом, то при добавлении на него файлов они сохранялись в любом доступном месте. Но после организации разделов границы оказываются не гибкими. Следовательно, при заполнении одного раздела нельзя

автоматически "занять" пространство у другого раздела. А это означает, что необходимо тщательно управлять тем, что направляется в каждый раздел.

Ограничения, накладываемые дисковыми разделами фиксированной емкости, не обязательно плохие. Во-первых, они заставляют вас лучше организовать файлы. Во-вторых, они повышают скорость работы диска, так как файлы, а также таблицы FAT и каталоги, относящиеся к этим файлам, содержатся в разделе, занимающем относительно мало цилиндров. Это позволяет DOS обращаться к файлам быстрее.

Организация разделов не годится, если, например, вы приобрели большой жесткий диск, чтобы иметь место для размещения файлов больше 32 Мбайт. В этом случае имеет смысл способ с большими секторами или использование одного огромного раздела в DOS 4.0. В любом способе можно адресовать весь диск как один раздел.

При организации разделов с помощью программ третьих фирм обычно можно обращаться к первому разделу любого физического накопителя, применяя только DOS. К остальному пространству можно обращаться только при наличии специального драйвера, установленного "поверх" DOS. При организации разделов с помощью DOS 3.3 эта проблема не возникает.



Среди популярных программ организации разделов отметим Disk Manager фирмы OnTrack Software, SpeedStor фирмы Storage Dimensions, драйвер EVDR.SYS фирмы Priam и Vfeature Deluxe фирмы Golden Bow Software. Последняя программа позволяет изменять размер сектора и даже превратить два физических накопителя (логически) в один большой накопитель. Это пример образования в ПК дисковых томов больше 8 Гбайт.

DOS 3.3 "прожила" длительное время и большинство ее ошибок исправлено. Поэтому нерационально применять для организации разделов на новом жестком диске программы третьих фирм. Они превращают систему в зависящую от этих драйверов. Если у вас есть DOS 3.3, а в системе применяется один из таких драйверов, целесообразно внести некоторые изменения.

Независимо от способа организации дисковых разделов необходимо осуществить это до форматирования (высокого уровня) дисковых томов. Подробности организации разделов рассмотрены в главе 9.

7.4. Преодоление физических ограничений

Вначале самым серьезным ограничением емкости тома DOS было логическое ограничение. Когда его в DOS 4.0 подняли с 32 Мбайт до 2 Тбайт, оно перестало быть проблемой. Теперь важнее другое ограничение, которое накладывается BIOS и физическими параметрами подключенного к PC жесткого диска.

Преодоление этого ограничения оказалось несколько труднее преодоления логического ограничения. В этом разделе рассмотрены несколько эффективных способов, разработанных для решения этой проблемы.

7.4.1. Чисто программный способ

В программе организации разделов Disk Manager фирмы OnTrack Software имеется компонент XBIOS, который можно использовать с любым 16-битовым контроллером жесткого диска (он вставляется в два слота шины ввода-вывода ISA или EISA). Применение XBIOS расширяет возможности Disk Manager работать с большими жесткими дисками.

XBIOS — это программа, которая добавляется к DOS. Disk Manager знает, как найти ее и использовать вместо программы прерывания 13h BIOS для считывания с жесткого диска или записи на него. Фирма IBM разработала первый интерфейс

жесткого диска для PC/XT на базе контроллера фирмы Western Digital, но BIOS не поддерживает весь диапазон параметров диска, который обеспечивает контроллер.

Контроллер фирмы Western Digital допускает до 12 бит для номера цилиндра вместо 10 бит, разрешенных для INT 13h. По существу, программа XBIOS изменяет ограничение BIOS, допускающее номера цилиндров только от 0 до 1023, и расширяет диапазон до 0 — 4095; она не снимает ограничений BIOS (INT 13h) на диапазон номеров головок и секторов.

Такое решение годится только при организации разделов с помощью программы Disk Manager. Оно неудобно в том отношении, что программу приходится считывать с диска. Но после того, как она считана и привязана к DOS и Disk Manager, программа изменяет операцию DOS считывания с диска. Инженеры фирмы OnTrack нашли оригинальное решение считать с диска одним способом для нахождения XBIOS, а затем другим способом с того же самого диска.

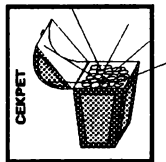
Как часто случается с чем-то оригинальным, некоторые другие программы оказываются несовместимыми с этим чем-то. Вскоре после поставки Windows 3.0 пользователи сообщили о том, что они теряли все данные на жестких дисках. Проблема объяснялась неудачным взаимодействием программы фирмы OnTrack и версией драйвера SMARTDRV.SYS из Windows 3.0.

Теперь эта проблема решена. Так или иначе, поставщики справляются с несовместимостью, разрабатывая "заплаты" для той или другой программы. Более надежный способ преодоления ограничений заключается в применении специального контроллера жесткого диска.

7.4.2. Отображение секторов

Напомним, что на высшем уровне DOS считает диск просто длинной цепочкой секторов, в которых она может хранить данные. Когда она готова запросить BIOS скомандовать накопителю выполнить фактическое обращение к диску, она преобразует логический номер сектора в трехмерный набор чисел адресуемых цилиндра, головки и физического сектора.

Контроллер жесткого диска может выполнять нечто аналогичное только в обратном порядке. Он может взять все дисковое пространство физического накопителя и преобразовать его в один длинный список секторов. После этого он может преобразовать элемент этого списка в фиктивный адрес цилиндра, головки и сектора, который он представляет DOS. Такой прием называется преобразованием головки и сектора или отображением секторов.

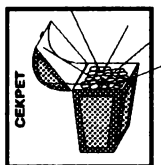


Предположим, что у вас есть накопитель, который физически имеет 7 головок, 1,781 цилиндр и 54 сектора на дорожке. (Это параметры накопителя ESDI 1664-7 фирмы Micropolis.) В результате умножения этих чисел получается, что накопитель имеет 673,218 секторов. (Если каждый сектор содержит 512 байт, то форматированная емкость этого накопителя составляет 328,7 Мбайт.)

Предположим теперь, что контроллер диска при взаимодействии с BIOS полагает, что максимально допустимые BIOS значения равны 16 головкам и 63 секторам на дорожке. Тогда в цилиндре будет $16 \cdot 63 = 1,008$ секторов и такой фиктивный накопитель будет иметь только 667 цилиндров, что удовлетворяет ограничению BIOS в 1,024 цилиндра.

Отображение действует следующим образом. Первые 54 сектора помещаются на первую дорожку (цилиндр 0, головка 0). Следующие 9 секторов, которые, по мнению DOS, находятся на той же дорожке, помещаются на следующую дорожку (цилиндр 0, головка 1). То, что DOS считает второй дорожкой (ее цилиндр 0, головка 1), начинается на цилиндре 0, головке 1 (первые 45 секторов) и заканчивается 18 секторами на цилиндре 0, головке 2 и т.д. (рис. 7.1).

Отображение (или преобразование) секторов появилось в контроллерах жестких дисков в другом контексте до преодоления физических ограничений. Некоторые ранее разработанные программы для PC "знали", что все дисковые накопители имеют 17 секторов на дорожке (как в накопителях MFM). Такие программы не работали с накопителями RLL, имеющими 26 секторов на дорожке. Одно из решений и состояло во введении в контроллер отображения секторов.

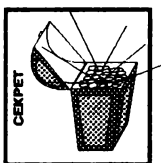


При этом контроллер "обманывал" DOS в другом направлении по сравнению с рассмотренным выше примером: контроллер представлял DOS фиктивный накопитель MFM с 17 секторами на дорожке, т.е. имеющим примерно в 1,5 раза больше дорожек, чем в реальном накопителе. Для этого можно представить, что накопитель имеет как бы в 1,5 раза больше головок или в 1,5 раза больше цилиндров. Пока оба эти числа остаются ниже ограничения BIOS, неважно, какую "ложь" выбирают разработчики для контроллера.

Достоинство такого "лживого" контроллера состоит в том, что жесткий диск выглядит более стандартным диском в части адресации DOS и BIOS, а недостаток — в том, что DOS, BIOS и другие программы не знают параметров реального накопителя. Для большинства операций DOS и приложений DOS это не играет роли, но усложняет программы изменения коэффициента чередования накопителя. Такие программы должны точно знать, какие секторы образуют физическую дорожку реального накопителя. Не зная реальных параметров, эти программы не могут обеспечить всех тех преимуществ, для достижения которых они разработаны.

7.4.3. Другие способы отображения в контроллере

Иногда отображение в контроллерах вводится по другим причинам. Одна из них состоит в том, чтобы представить накопитель имеющим меньше дефектов, чем на самом деле, а вторая — в желании поместить больше данных в меньшее пространство.



Типичным современным контроллером ESDI является UltraStor 12F и его кэширующий вариант 12C (он хорошо подходит для упомянутого выше накопителя фирмы Micropolis). При проведении форматирования низкого уровня контроллер предлагает множество опций.

Можно, например, запросить Head Skewing (Перекося головки), которая корректирует номера секторов при переходе с одних дорожки или цилиндра к соседним, обеспечивая время на коммутацию головок или перемещения головок с цилиндра на цилиндр (о перекося головки см. в главе 3, рис. 3.3).

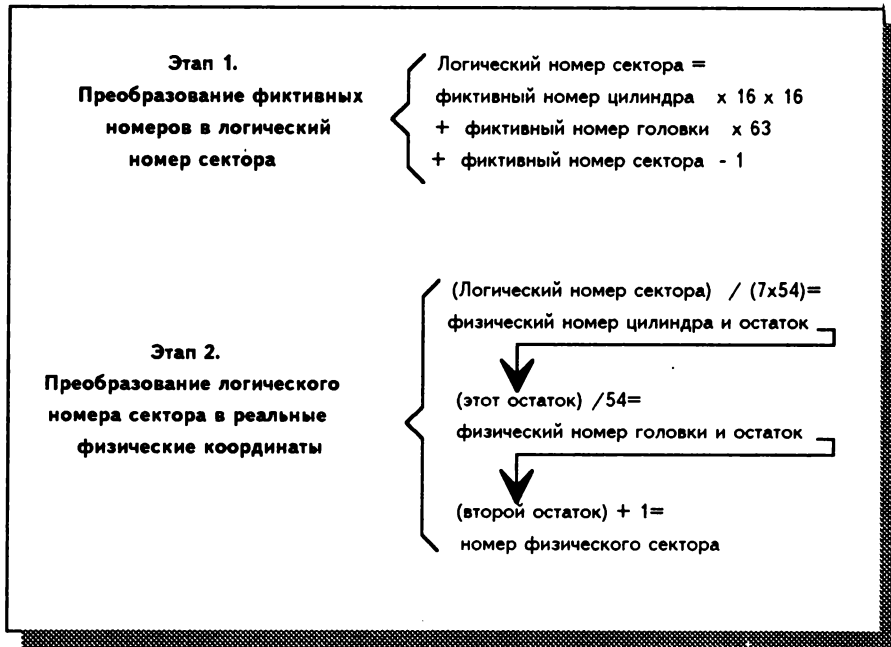
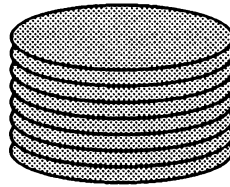
Еще одна опция Spare Sector (Запасной сектор) означает, что контроллер форматирует дорожки накопителя с 55 секторами, но, если все секторы хорошие, он использует только первые 54. Если какой-то сектор на дорожке плохой, он отмечается как дефектный и игнорируется. После этого физические номера остальных секторов на этой дорожке уменьшаются на единицу (предполагая коэффициент чередования секторов 1:1).

Контроллер предлагает также четыре способа Sector Mapping (Отображение секторов): представить 17, 32 или 63 сектора на дорожке или реальное число, которое для накопителя 1664-7 составляет 54 сектора на дорожке.

Причина предложения 17 секторов очевидна: превратить накопитель в обычный накопитель MFM. Причина образования 63-х секторов также ясна: 63 — максимальное число секторов на дорожке, встроенное в INT 13h. Отсутствие преобразования обеспечивает программам доступ к реальному физическому накопителю. Несколько странным выглядит вариант с 32 секторами на дорожке. Он объясняется тем, что некоторые версии DOS имеют программу FDISK, которая не работает более, чем с 32 секторами на дорожке, и эта опция допускает максимальное число секторов, на которое рассчитаны такие версии DOS.

Моделируемый накопитель

число цилиндров	667
число головок	16
число секторов на дорожке	63



Физическая реальность

Число цилиндров	1781
Число головок	7
Число секторов на дорожке	54

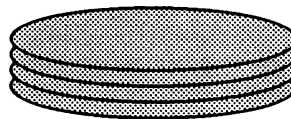


Рис. 7.1. Пример преобразования головки и сектора

Для накопителей с емкостью больше 504 Мбайт необходимо использовать с DOS следующую опцию Track Mapping (отображение дорожки). Ограничение в 504 Мбайт накладывает официальный интерфейс BIOS INT 13h. В этой опции применяется лазейка в ограничении BIOS емкости диска, т.е. искусственно увеличивается максимальный номер головки с сохранением номеров сектора и цилиндра в пределах даже для накопителей очень большой емкости (хотя это возможно только для накопителей, емкость которых не превышает 8,875 Гбайт).

Наконец, контроллер UltraStor 12F предлагает опцию 1,024 Cylinder Truncation (Усечение цилиндра), которая просто заставляет контроллер игнорировать цилиндры

после номера 1,023. Применение этой опции позволяет контроллеру предоставить BIOS допустимый накопитель, хотя и с уменьшением фактической емкости накопителя.

Многие современные контроллеры предлагают такие же возможности в различных вариантах. Некоторые из них рассчитаны на накопители большой емкости, другие проводят преобразование как побочную операцию, а многие модели должны представлять накопитель с параметрами, отличающимися от реальных, просто потому, что накопитель нельзя точно описать одним набором параметров. (В таких накопителях применяется разделение на зоны, т.е. диапазоны номеров цилиндров, имеющих различное число секторов на дорожке.)

Интенсивное отображение секторов в силу многих причин применяется в накопителях SCSI- IDE/ATA. Все такие современные накопители рассчитаны на то, чтобы выглядеть совершенными, данных в меньшее пространство.

Как такие накопители создают иллюзию своего совершенства? Для этого в каждом секторе имеется специальный дополнительный указатель. Если данный сектор хороший, указатель не используется, а если сектор имеет дефект, указатель показывает, где найти заменяющий сектор. В таких накопителях имеются запасные дорожки и могут быть запасные секторы на каждой дорожке. Для DOS накопители выглядят совершенными, и программа CHKDSK никогда не сообщает о плохих секторах.

Иногда (в большинстве накопителей IDE/ATA) такой способ применяется только для дефектов, обнаруженных при первом тестировании диска в процессе производства. Другие накопители (обычно SCSI) могут проводить так называемый "горячий ремонт", т.е. могут переназначать секторы "на лету", когда контроллер обнаруживает плохой сектор.

В любом накопителе с зонной организацией применяется преобразование секторов. Про этот способ в главах 2 и 3 говорилось, что он применяется в накопителях SCSI и IDE и иногда в платах жестких дисков для достижения максимальной емкости. Суть этого способа в том, что количество секторов на дорожке непостоянно на всех цилиндрах, а это требует, чтобы контроллер "обманывал" DOS, так как все в DOS рассчитано на постоянное количество секторов на дорожке для любого конкретного накопителя. Следовательно, в таких накопителях контроллер должен всегда преобразовать головку и сектор.

7.5. RAID и другие способы

Преобразование секторов позволяет использовать диск, имеющий слишком много цилиндров или секторов на дорожке, но оно не помогает, если емкость диска превышает ограничение BIOS в 504 М или 8.875 Гбайт (число зависит от того, разрешено числу головок превышать 16 или нет). Вариант с XBIOS разрешает применять диски, имеющие до 4,096 цилиндров, что повышает емкость до 2 Г или 32 Гбайт (по-прежнему в зависимости от числа головок). Другие фирмы разработали различные способы, напоминающие XBIOS, обеспечивающие другие пределы, но во всех из них диапазон емкости примерно один и тот же.

Как же применять накопитель большей емкости? Как быть, если вам нужен том, содержащий десятки гигабайтов и более? Ведь нельзя обмануть ПК и заставить его работать с таким большим диском; действительно, с одним таким накопителем ПК не справится. Но можно подключить к ПК несколько накопителей и достичь практически любой требуемой емкости. Некоторые способы рассчитаны именно на то, чтобы создать эффект одного большого жесткого диска, а другие ориентируются на улучшение дисковой подсистемы в ПК.

Уже начиная с PC/XT, подключение к одному контроллеру одного или двух накопителей в ПК стало стандартом. Со временем в ПК появилось два контроллера и максимальное число накопителей увеличилось до четырех.

Применение двух аналогичных контроллеров долго было возможным, но редко используемым вариантом. Некоторые контроллеры MFM, RLL и ESDI допускали наличие в ПК еще одного контроллера. Только один из контроллеров мог иметь активную съемную плату ПЗУ, а обоим контроллерам приходилось назначать различные адреса портов.

Появление главных адаптеров и накопителей SCSI открыло новые возможности. Каждый такой адаптер допускает подключение до семи ведомых устройств SCSI, каждое из которых, в принципе, может обслуживать до восьми логических устройств. Таким образом, теоретически один главный адаптер SCSI может поддерживать до 56 накопителей на жестких дисках.

Такую возможность нельзя реализовать с большинством имеющихся на рынке накопителей SCSI, так как каждый из них имеет встроенный интерфейс SCSI ведомого устройства. Но если кто-то разработает подходящий интерфейс SCSI ведомого для нескольких накопителей, можно подключить к нему до восьми накопителей IDE/ATA и до семи интерфейсных устройств к одному главному адаптеру SCSI. Более того, в одном ПК можно иметь несколько главных адаптеров SCSI. Кажется, что теперь реального предела емкости нет, за исключением нехватки уровней запросов прерываний IRQ и слотов в компьютере.

Решение с несколькими накопителями позволяет иметь в ПК очень большую суммарную емкость, но за счет создания отдельных запоминающих устройств. По крайней мере на аппаратном уровне для доступа к данным потребуется определить четыре параметра — номера накопителя, цилиндра, головки и физического сектора.

До версии 5 DOS могла работать только с двумя накопителями на жестких дисках, а теперь она может поддерживать до семи физических накопителей. (Конечно, эти накопители можно разделить максимум на 24 отдельных логических накопителя с буквами от C до Z.) При наличии в ПК более 50 накопителей DOS не сможет назначить всем им буквы.

Решение этой проблемы и создание очень больших логических томов DOS состоит в том, чтобы считать несколько физических накопителей одним логическим томом. До недавнего времени для этого требовались программы драйверов третьих фирм (например, драйвер VFeature Deluxe фирмы Golden Bow Software). Несомненно, одно из удачных решений этой проблемы для много гигабайтных накопителей получит широкое распространение.

Еще один подход, обещающий некоторые дополнительные преимущества, связан с методом RAID (Redundant Array of Inexpensive Disks — Избыточный набор дешевых дисков). Он опирается на то, что для памяти сверхбольшой емкости удобнее и дешевле применить несколько меньших накопителей, чем пытаться создать один накопитель огромной емкости. Более того, при тщательном построении системы из отдельных накопителей можно добиться, что система окажется надежнее любого из ее компонентов. (Такое утверждение представляется необычным, так как чаще при объединении частей целое менее надежно любой части.)

Формально определены пять уровней RAID, различаемых по сложности и предлагаемыми возможностями. Один из способов, принятый в нескольких уровнях, заключается в хранении каждого байта в нескольких накопителях. При таком способе хранения информации и применении достаточно "умных" кодов с исправлением ошибок становится возможным с достаточно высокой надежностью восстановить всю информацию, даже если один из накопителей полностью выйдет из строя.

Некоторые реализации RAID позволяют осуществить "горячий обмен" — заменить поломанный накопитель, когда все остальные продолжают работать. При правильной реализации и эксплуатации такие способы обещают подсистемы памяти с бесконечным средним временем между отказами.

Таким образом, современное (после DOS 4) ограничение емкости логического тома (2 Тбайт) кажется уже не столь фантастическим. Конечно, со временем появятся ПК с логическими томами DOS примерно такой емкости. Но остановимся и поразмыслим. Сейчас имеются накопители емкостью 1 Гбайт (и они быстро становятся общепринятыми). А всего 13 лет назад в первом ПК емкость накопителя на односто-

роннем гибком диске составляла всего 160 Кбайт. Если прогресс будет продолжаться с такой же скоростью, то менее чем через 10 лет придется преодолевать предел логического тома в 2 Тбайт.

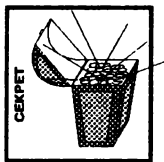
7.6. Сообщения об ошибках

Раньше мы рассмотрели несколько уровней программ в ПК. Запросы дисковых операций проходили с верхнего уровня к нижнему — к командам аппаратным средствам. На любом этапе этого процесса запрашиваемый модуль может реагировать некоторым сообщением запросчику (например, сообщая об успехе или неудаче или поясняя, почему невозможно выполнить запрошенное). Такие сообщения возвращаются по той же цепи в обратном порядке (рис. 7.2).

Любой уровень может сформировать сообщение об ошибке, если он не понимает поступившего сверху запроса. Сообщение об ошибке выдается также в случае, если аппаратура работает неправильно или если конкретный уровень не может выполнить задачу по какой-то другой причине.

Каждый программный уровень может реагировать на сообщение об ошибке несколькими способами: игнорировать сообщение, повторить запрос в надежде на успех или предпринять что-либо другое.

Нас особенно интересуют запросы накопителю или контроллеру считать или записать некоторые данные на диск. Существует множество причин того, что эти запросы выполняться не удастся. О каждой причине контроллер посылает сообщение об ошибке в BIOS, она передает его DOS, а уже DOS посылает его (или не посылает) пользователю или его прикладной программе.



В накопитель может поступить запрос позиционировать головки на конкретный цилиндр ("отыскать" этот цилиндр). Контроллер пытается выполнить это действие, а затем обнаруживает, что оно безуспешно. Обычная процедура восстановления из такой ошибки состоит в том, чтобы рекалибровать накопитель (найти цилиндр 0), а затем попытаться вновь найти нужный цилиндр.

В контроллер может поступить запрос считать конкретный сектор. В этой операции возможны два вида ошибок. Первая связана с невозможностью найти заголовок сектора или (при его отыскании) неправильным циклическим избыточным контролем. Сообщение об этом означает, что невозможно уверенно определить местонахождение на диске. По-прежнему подходящей реакцией оказывается рекалибровка и повторная попытка.

Второй вид ошибки при считывании возникает, когда контроллер нашел заголовок сектора, но считываемые данные несовместимы с кодами исправления ошибки (ЕСС). Далее нужно учитывать, достаточно ли ЕСС для исправления данных. В любом случае простейшим вариантом будет повторная попытка считывания. Когда ЕСС достаточно для исправления ошибки, иногда проще и быстрее исправить данные. Если контроллер использует ЕСС, он должен передать исправленные данные с сообщением о привлечении ЕСС.

Программа, получающая сообщение об ошибке, выбирает повтор операции. Когда прекратить эти попытки? Чаше BIOS просит контроллер жесткого диска повторить несколько раз считывание сектора, который он не может найти. При неудаче BIOS признает поражение и сообщает DOS о невозможности найти сектор. Теперь DOS может просить BIOS повторить обращение к контроллеру и начинается второй круг попыток. Поскольку DOS запрашивает BIOS несколько раз, а BIOS несколько раз обращается к контроллеру, общее число попыток очень быстро растет. Сотни попыток не являются необычным числом.

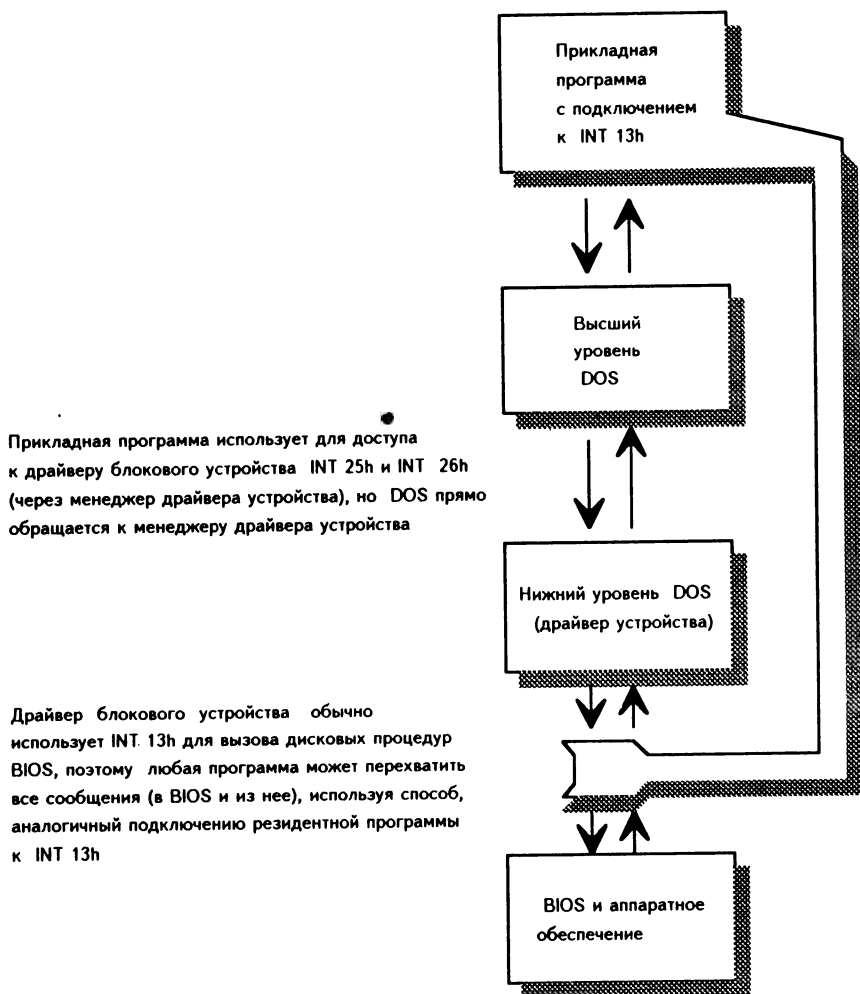


Рис. 7.2. Подключение прерывания к перехвату сообщений об ошибках

Наконец, DOS решаете признать поражение. После этого возникают две ситуации. Чаще всего DOS сообщает о поражении коротким сообщением и предоставляет выбор Abort, Retry or Ignore (отмена, повторение или игнорирование) или Abort, Retry, Ignore or Fail (отмена, повторение, игнорирование или продолжение с

ошибкой) — точная фраза зависит от версии DOS. Если запрос сектора данных был от современной сложной прикладной программы, она может перехватить сообщение DOS о поражении и выполнить то, что посчитали нужным разработчики программы. Они могут даже повторно запросить DOS, что вызывает дополнительные сотни попыток.

Имеется один случай, когда DOS ни прилежна, ни информативна, — ECC оказалось достаточно для восстановления первоначально неправильно считанных данных. Повторные попытки при этом не инициируются. DOS просто воспринимает восстановленные данные как "истину в последней инстанции", не информируя приложение о возникшей ранее ошибке. Эта ситуация самая печальная. Она скрывает от вас надвигающуюся гибель, сообщая только о наступившей катастрофе. К счастью, даже приложение DOS может оказаться "впереди DOS" и перехватить сообщения об ошибке от контроллера диска до их маскировки.

Перехват осуществляется довольно просто — в приложение DOS вводится код, который загружает адрес внутри себя в таблицу векторов прерываний для прерывания INT 13h. Когда в дальнейшем DOS просит BIOS обратиться к дисковому накопителю, фактически она обращается к специальному коду внутри приложения.

DOS не знает о перехвате; она просто выдает вызов INT 13h. Вместо прежней процедуры обслуживания прерывания INT 13h процессор находит адрес кода мини-программы в приложении и вызывает в качестве процедуры обслуживания прерывания прикладную программу. Мини-программа не обязательно должна знать, как проводить любое реальное обращение к диску. Вместо выполнения запрошенной задачи программа должна просто вызвать BIOS по тому адресу, который ранее находился в данных ячейках таблицы векторов прерываний.

Зачем же вводить программу, если она не делает ничего, кроме передачи запроса? Но при этом программа оказывается на пути, по которому система возвращает все сообщения программам высшего уровня о результатах запроса. Программа получает сообщения раньше DOS, которая, возможно, проигнорирует их (см. рис. 6.12).

Программы изменения коэффициента чередования секторов типа SpinRite "цепляются" к INT 13h именно таким образом. После этого они "видят" любое место на диске, где происходит "борьба" за получение данных. Программы могут переместить восстановленные данные в некоторое безопасное место до того, когда проблема окажется непреодолимой.

Заключение

- Особенность BIOS первого IBM PC, скопированная в последующих ПК, ограничивает для жесткого диска количество головок, цилиндров и секторов на дорожке. Эти "официальные" ограничения означают, что емкость подключаемого к ПК жесткого диска не может быть больше 504 Мбайт.
- Широко применяемый "обман" допускает число головок до 255, что повышает ограничение емкости накопителя до 8,875 Гбайт.
- Помимо физических ограничений DOS накладывает логические ограничения. До версии 3.3 дисковый том (буква накопителя) не мог содержать более 32 Мбайт. Это ограничение в DOS 4.0 увеличивает до 2 Тбайт (примерно 2,2 миллиона миллионов байт).
- До DOS 3.3 некоторые производители компьютеров модернизировали DOS, чтобы допустить логические дисковые тома больше 32 Мбайт. Для этого некоторые программы FORMAT создают логические секторы больше 512 байт (физические секторы остаются 512-байтовыми). В DOS 3.31 фирмы Compaq реализован другой способ, который в дальнейшем был принят в MS DOS 4.0 и последующих версиях.
- DOS 3.3 допускает организацию на большом физическом накопителе множества логических томов емкостью до 32 Мбайт. Для большинства людей такое решение оказалось вполне приемлемым.

- До DOS 3.3 программы организации разделов третьих фирм разрешали пользователям создавать логические тома после C, большие 32 Мбайт. С появлением DOS 4.0 эти решения устарели. Однако во многих ПК до сих пор инсталлированы драйверы третьих фирм, и для их ликвидации владельцам ПК придется заново организовать разделы на жестком диске.
- Накопители, имеющие больше 1,024 цилиндров, можно полностью использовать в ПК только после преодоления физического ограничения. Для такого "преодоления" можно расширить BIOS или применить контроллер диска с преобразованием секторов. Второй способ предпочтительнее, так как он невидим для большинства программ.
- К одному ПК можно подключить несколько накопителей, — обычно два, но допускается и больше. Новые идеи RAID позволяют применять больше накопителей с повышенной надежностью.
- Большинство программ "заставляет" аппаратные средства работать, передавая сообщения запросов через несколько уровней программ. Сообщения о реакциях возвращаются по той же цепи. DOS фильтрует некоторые сообщения об ошибках, и пользователи не видят более того, что, по мнению DOS, они должны видеть. Такая фильтрация неблагоприятна, так как она может привести к потере данных, если вы не знаете о надвигающейся гибели накопителя до того, когда можно переслать данные в безопасное место.
- Любая программа DOS, которая хочет "видеть" эти сообщения, может получить их, "цепляясь" к прерыванию 13h. С помощью таких программ вы можете защитить себя от потери данных в случае преждевременной гибели накопителя.

Теперь вы знаете, как MS DOS работает с жесткими дисками. В следующих трех главах вы узнаете, как подготовить жесткий диск для DOS, а также некоторых других операционных систем.

Глава 8. Подготовка жесткого диска к хранению информации.

Этап 1 — физическое форматирование

В этой главе вы узнаете

- Почему жесткий диск бесполезен, если на нем отсутствует важная информация, записанная при форматировании диска
 - Три этапа или ступени форматирования
 - Как проходит первый этап, называемый форматированием низкого уровня и чем отличаются контроллеры диска.
-

8.1. Что такое форматирование диска?

Теперь вы знаете, как работают жесткие диски и как с ними работает MS DOS. Прежде чем любой жесткий диск сможет работать с DOS, его необходимо подготовить. При поступлении с предприятия жесткий диск не может хранить никакой информации, так как на поверхности дисков нет организующих магнитных отметок. Нанесение таких организующих отметок на поверхность диска будет первым из трех этапов того, что необходимо предпринять, чтобы DOS могла использовать диск — этот процесс называется форматированием диска. В этой главе рассмотрен весь процесс форматирования и подробно описан первый его этап.

Напомним, что контроллер диска отыскивает секторы информации. Он ожидает, что каждый сектор состоит из заголовка сектора, содержащего адресную информацию и код избыточного циклического контроля (CRC), а также области данных и соответствующих кодов исправления ошибки (ECC). Контроллер диска ожидает, что эта информация закодирована по принятому способу кодирования данных (например, MFM, RLL или ARRL). Но до того как накопитель "представили" контроллеру, на диске нет никакой такой информации.

Первая операция при подготовке жесткого диска — заставить контроллер записать "скелет" тех данных, с которыми он будет работать. Эта операция называется физическим форматированием (или форматированием низкого уровня). После создания в накопителе структуры для хранения информации необходимо организовать разделы и провести логическое форматирование (или форматирование высокого уровня), чтобы диск работал с DOS.

8.2. Трехэтапный процесс (и его аналогия)

Для понимания трехступенчатого процесса подготовки жесткого диска рассмотрим такую аналогию. Новый жесткий диск похож на новую стоянку автомашин — диск и стоянка имеют потенциально много места для хранения. На стоянке прежде всего нужно провести линии, показывающие клиентам, где можно ставить автомашины (аналогично форматированию низкого уровня диска). После этого на стоянке нужно возвести ограждения, отделяющие различные группы водителей (например, клиентов и работников стоянки), а это похоже на организацию дисковых разделов.

Затем необходимы служащие (как DOS "паркует" ваши файлы на диске) и помещение для хранения ключей. Вам потребуется каталог фамилий владельцев и номеров мест, где находится каждая автомашинна, а это похоже на каталог диска, создаваемый при форматировании высокого уровня. Теперь вы знаете, какие места пусты и где найти автомашину каждого водителя.

Конечно, все автомашины примерно одинакового размера, а файлы на диске могут быть большими и маленькими. Чтобы учесть это в нашей аналогии, предположим, что иногда сразу прибывает группа автомашин и руководитель группы решает, когда стоянку покинет вся группа. Теперь в каталог достаточно занести только номер места стоянки руководителя группы. Другая таблица будет показывать, какие места пусты, а какие заняты автомашинами группы. Наглядное представление о приведенной аналогии дает рис. 8.1.

8.3. Три способа физического форматирования

Существует три способа форматирования низкого уровня на жестком диске, но все они заканчиваются одинаково. Первый — воспользоваться специальной программой, поставляемой на дискете производителем контроллера. Второй способ — выполнить программу из ПЗУ расширения BIOS в контроллере или в ROM BIOS материнской платы. Третий способ — использовать программу третьей фирмы (которая, в свою очередь, применяет программу ROM BIOS).

Некоторые контроллеры, поставляемые с диском, имеют программу форматирования низкого уровня. Фирма IBM предоставляет такую программу на дискете Улучшенной диагностики, которая прилагается к Техническому справочному руководству, и на Справочной дискете, поставляемой с каждой моделью PS/2. (Для доступа к этим специальным опциям меню Справочной дискеты следует нажать клавиши Ctrl+A.)

Другие контроллеры используют программу, которая постоянно находится в ПЗУ на плате расширения BIOS. Обычно для выполнения этой программы привлекается отладчик DOS DEBUG. На стимул DEBUG (дефис) напечатайте G=C800:5 и нажмите клавишу Enter. Если ваш контроллер диска имеет ПЗУ по адресу, отличающемуся от C8000h, подставьте этот адрес (опуская последний нуль). Иногда последнее число не равно 5, и в этом случае программа ничего не делает. Если после ввода предыдущей команды на экране просто появляется стимул, нажмите клавишу Q, а затем Enter. Вы выходите из DEBUG и возвращаетесь к стимулу DOS. За инструкциями, что делать дальше, обратитесь к производителю контроллера.

Можно также воспользоваться программой третьей фирмы, например Disk Manager или SpeedStor. Они поставляются с простыми инструкциями. Отметим, однако, что при простейшем способе выполнения этих программ действий будет больше, чем вам требуется. Например, выполняйте Disk Manager в ручном (Manual) режиме, чтобы программа не провела всех действий по подготовке диска. (В главе 9 показано, почему нельзя разрешать этим программам выполнять всю работу.)

В каждом из трех способов формируется запрос контроллеру на размещение на диске наборов физического формата. В ответ на запрос контроллер перемещает блок головок в внутреннего цилиндра на внешний по одному цилиндру. Для каждого цилиндра контроллер поочередно коммутирует головки, проходя по всем головкам до передвижения к следующему цилиндру. Для каждого цилиндра и головки контроллер записывает полную дорожку информации в том формате, какой ему потребуется для считывания или записи дополнительных данных.

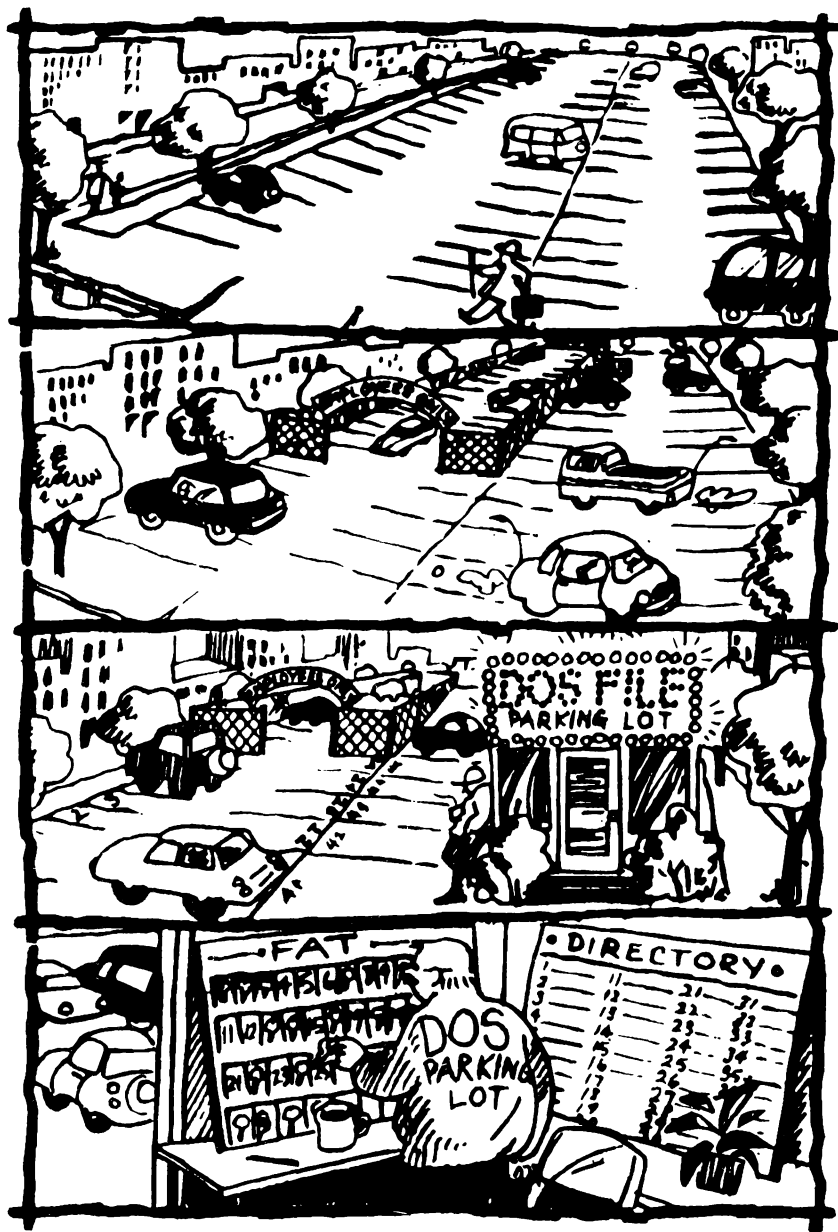
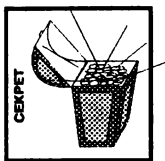


Рис. 8.1.Хранение информации на жестком диске весьма напоминает способ парковки автомобилей на стоянке



Начиная форматирование с внутреннего цилиндра, контроллер отводит максимум времени операциям до перехода к внешним цилиндрам, в которые DOS помещает самые критические данные накопителя — таблицы размещения файлов и корневой каталог. Без этой информации нельзя получить никаких данных. Если вам не требуется форматирование низкого уровня, можно остановить процесс и сохранить часть данных. (Повторение форматирования низкого уровня уничтожает все данные.)

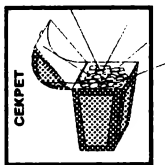
Первые контроллеры жестких дисков ПК выполняли форматирование низкого уровня абсолютно противоположным образом, сразу же уничтожая наиболее критические данные. Производители контроллеров изменили способ форматирования после того, как от него пострадало (вплоть до сердечных приступов) множество пользователей. Проверьте, есть ли в вашем ПК современный контроллер диска.

Форматирование низкого уровня проводится по дорожкам. Обычно можно заставить контроллер выполнить его для конкретной дорожки и иногда для всего накопителя. Некоторые программы третьих фирм (например, Disk Manager) позволяют также проводить форматирование для диапазона дорожек. Это просто означает, что они повторяют команду контроллеру форматировать одну дорожку для каждой головки до форматирования указанного диапазона цилиндров.

Важно отметить, что только во время форматирования низкого уровня контроллер записывает заголовки секторов. Он также записывает секции данных секторов (заполняя их фиктивными данными и соответствующими кодами ECC). Конечно, заголовки секторов считываются, чтобы определить, где считывать или записывать данные, но больше запись в заголовки не проводится.

8.3.1. Только контроллер проводит форматирование

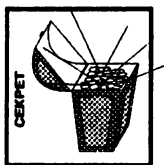
На низшем уровне диск может форматироваться только контроллером, поэтому способ форматирования определяет схема контроллера. Каждый контроллер осуществляет форматирование по-разному. (Если контроллер не может выполнить форматирование низкого уровня, никакая программа не заставит его сделать это.)



С другой стороны, контроллеры гибких дисков всех ПК осуществляют форматирование низкого уровня одинаково (для дисков одного диаметра и емкости) частично потому, что при разработке контроллеров применяется только один стандартный набор микросхем. Но более важно, что это отражает наше требование использовать диск, сформатированный на одном ПК, в любом другом компьютере. Поскольку жесткие диски несменные, такое требование к ним не предъявляется.

Некоторые жесткие диски являются сменными. Они работают только в тех ПК, которые имеют одни и те же специальные аппаратные средства. Нельзя, например, взять сменный диск AdPack фирмы Tandon, вставить его в систему MegaDrive и ожидать, что система заработает.

Различные контроллеры иногда помещают номера головки, цилиндра и физического сектора в заголовки секторов в разном порядке. Они также могут различаться способом вычисления ECC, местом расположения флажка плохого сектора и другой информацией в заголовке сектора. Наконец, даже контроллеры одного и того же номинального типа (например, RLL-контроллеры) могут отличаться тонкостями кодирования данных. Вследствие этой индивидуальности необходимо проводить форматирование низкого уровня накопителя при подключении к нему нового контроллера.

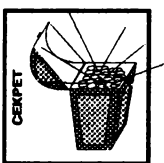


Вам придется повторять форматирование низкого уровня при каждом подключении накопителя к контроллеру жесткого диска нового типа. Разумеется, все контроллеры, например накопителей Tandon AdPack, должны быть достаточно совместимыми, чтобы можно было заменять пакеты дисков AdPack. Возможность пересылать данные так же, как в месте продажи сменного накопителя, накладывает серьезное ограничение на проектирование таких контроллеров. Другие производители, не связанные этим ограничением, изменяют форматующие наборы в каждой модели, а иногда даже чаще.

Контроллеры жестких дисков PerStor необычны во многих отношениях. Например, они являются исключением из только что сформулированного правила. Далее, они хранят информацию в два раза плотнее обычных накопителей MFM (даже плотнее, чем в накопителях RLL). Они нумеруют физические секторы с нуля (вместо ожидаемой BIOS единицы), а затем скрывают этот факт от BIOS. Другие приемы позволяют контроллеру PerStor распознать формат низкого уровня накопителя, форматированного любым другим контроллером PerStor. Можно даже использовать накопитель, форматированный контроллером PerStor XT, с контроллером PerStor AT.



Помните, что форматирование низкого уровня жесткого диска разрушает все ранее записанные данные. Большинство самых опасных компьютерных вирусов заставляет контроллер диска повторять форматирование низкого уровня на одном или всех накопителях.



Некоторые специалисты по восстановлению данных говорят о том, что могут обнаружить остаточную намагниченность на поверхности диска — могут (по крайней мере иногда) восстановить данные даже после перезаписи, вызванной форматированием низкого уровня или копированием одного файла на другой. Такая возможность вызвала озабоченность правительства относительно национальной безопасности, и тогда установили, что файлы нужно стирать, записывая много раз различные (определенные заранее) наборы данных. В обычных же условиях можно считать, что после перезаписи файла или диска информация исчезает.

В общем, DOS не разрушает данные при форматировании высокого уровня или логического форматирования (правда, некоторые версии DOS все же разрушают данные). Именно это позволяет создать утилиты для восстановления данных после форматирования (подробнее в главе 11).

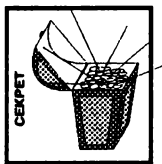
Программы изменения коэффициента чередования секторов, например SpinRite, позволяют повторить форматирование низкого уровня накопителя, не разрушая хранимых данных. Это не то же самое, что отмена форматирования низкого уровня. Такие программы "освежают" информацию в заголовках секторов, не требуя резервирования данных с диска, повторения проведения организации разделов, форматирования высокого уровня и возвращения зарезервированных данных.

В сущности, программы изменения коэффициента чередования секторов проводят это, образуя копию данных поочередно для одной дорожки. Затем они просят контроллер реформатировать одну эту дорожку. Они проверяют вновь сформатированную дорожку, а затем возвращают на нее первоначальные данные. Фактически такой прием не будет исключением из того правила, что форматирование низкого уровня разрушает данные. Просто программы достаточно долго хранят копию и возвращают ее после окончания реформатирования. Возвращаемые данные включают информацию высокого уровня DOS, которая позволяет избежать реформатирования и цикла резервирования-возвращения.

8.3.2. Некоторые нестандартные форматы

Несмотря на то, что почти каждый контроллер жесткого диска имеет свои тонкости форматирования низкого уровня, некоторые оказываются "таинственные" остальные. Несколько примеров рассмотрено дальше.

Комбинации диска и контроллера. Некоторые производители, поставляющие контроллеры и накопители в комплекте, наносят формат низкого уровня на предприятии и делают все от них зависящее, чтобы предотвратить повторение форматирования низкого уровня. Для таких комбинаций контроллера и накопителя не играет роли, что программа типа SpeedStor запрашивает о форматировании низкого уровня при подготовке диска к хранению информации или программа типа SpinRite запрашивает о форматировании как средстве повышения скорости переноса данных или улучшения целостности уже хранимых данных; контроллер просто отказывается проводить форматирование низкого уровня. Следовательно, такое форматирование осуществить невозможно.



В семействе Hardcard фирмы Quantum реализовано два приема, которые привели к запрету другим программам изменять заводское форматирование низкого уровня. Один из них — это зонная запись, т.е. количество секторов на внешних дорожках больше, чем на внутренних, а второй — отображение запасного сектора, с помощью которого фирма заменяет дефектные секторы некоторыми неиспользуемыми секторами, чтобы накопитель не имел дефектов (по крайней мере при поставке). Инженеры фирмы Quantum использовали эти приемы для предотвращения форматирования низкого уровня кем-то другим.

В большинстве накопителей IDE и SCSI реализованы такие же приемы. Они часто не разрешают никаким другим программам повторять форматирование низкого уровня.

Еще один интересный и странный прием применила фирма Toshiba в своих laptop. Этот прием называется мультиплексирование головок и предназначен для размещения большого объема данных в малых накопителях, но с представлением накопителя как стандартного накопителя MFM. Фирма Toshiba смогла разместить на дорожке 34 сектора. Такой прием удваивает объем данных на каждой дорожке, но если BIOS узнает о том, что это сделано, она не воспринимает накопитель как стандартный.

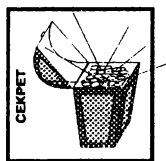
Инженеры фирмы Toshiba разработали специальный контроллер. Когда BIOS командует контроллеру считать сектор с конкретным номером для конкретного номера дорожки, контроллер преобразует запрошенное местоположение в реальное место, используя следующее правило преобразования. Прежде всего, для получения номера реального сектора удваивается номер запрошенного сектора и прибавляется единица, если номер головки нечетный. После этого для получения реального номера головки запрошенный номер головки делится на два с отбрасыванием остатка. Если, например, контроллеру задано считать сектор 5 для головки 0, он фактически считает сектор 10 для головки 0. Если задано считать сектор 15 для головки 1, фактически считывается сектор 31 для головки 0.

Самое непривычное в таком преобразовании состоит в том, что логические головки 0 и 1 вместе используют верхнюю поверхность, а не каждая головка работает со своей поверхностью. Логическая головка 0 использует каждый четный сектор, а логическая головка 1 — остальные секторы. В результате накопитель имеет только 17 секторов на дорожке, как обычный накопитель MFM.

Конечно, для каждой поверхности имеется только одна физическая головка. Фирма Toshiba просто позволила каждой головке иногда иметь логический номер n , а иногда логический номер $n+1$. Тогда форматирование низкого уровня для дорожки на верхней поверхности осуществляется только для обеих логических головок 0 и 1. Фирма Toshiba полагает, что такой необычный прием заставит пользователей не при-

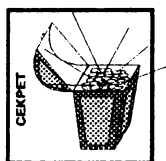
менять другие программы для форматирования низкого уровня накопителей их laptop. К сожалению, он не запрещает программам проводить форматирование низкого уровня и не сообщает правильного формата. Пользователи, которые хотят осуществлять форматирование низкого уровня laptop фирмы Toshiba, должны знать об этом и не делать ничего опасного.

Автоконфигурирующие контроллеры жесткого диска. В некоторых автоконфигурирующих контроллерах жесткого диска применяется менее сложный способ форматирования. Они помещают параметры накопителя в специальное место в накопителе, где-то в обычной области данных, но "невидимо". Например, в дополнение к обычным секторам данных они могут поместить параметры в специальный крошечный сектор на одной из дорожек данных. Такие контроллеры допускают запрос о форматировании низкого уровня любой дорожки. Если неосторожно задать дорожку, содержащую таблицу параметров диска, таблицу можно разрушить, и весь накопитель будет недоступным.



Несколько разработчиков дисковых утилит обнаружили эту особенность контроллера уничтожать сведения о параметрах диска. Автор программы SpinRite открыл ее в ходе разработки программы и в дальнейшем учел особенности работы с такими контроллерами. Теперь программу SpinRite можно безопасно применять с любыми контроллерами.

Почти все автоконфигурирующие контроллеры помещают информацию о параметрах диска на специальный цилиндр, к которому может обращаться только контроллер и только для этого. Такие контроллеры не допускают разрушения таблицы параметров диска независимо от запрошенных действий.



Возможно, самый сложный прием использует в некоторых своих контроллерах фирма Western Digital. Инженеры этой фирмы заметили, что часть главной загрузочной записи всегда остается пустой. Этот сектор, называемый также таблицей разделов, занимает первый сектор диска. (Подробнее об этом секторе в главе 9.) Фирма Western Digital решила сделать это неиспользуемое пространство полезным.

Контроллеры, в которых реализован рассматриваемый прием, помещают параметры накопителя в середину пустого места главной загрузочной записи. После этого контроллер прячет эту информацию. Если считать или записать этот сектор, контроллер покажет, что данная область хранит нули. Только контроллер может считать и изменить хранимые там числа и делает это он только для своих целей, а не для пользователя. До тех пор пока какая-нибудь программа не попытается сохранить там информацию, вы никогда не узнаете о специальных операциях контроллера.

Заключение

- Форматирование означает подготовку диска к работе. Все диски необходимо форматировать, прежде чем DOS сможет сохранять и считывать информацию.
- Полное форматирование жесткого диска состоит из трех этапов. Первый этап называется физическим форматированием, или форматированием низкого уровня. Второй этап называется организацией разделов, а третий — логическим форматированием, или форматированием высокого уровня.
- Осуществить форматирование низкого уровня можно тремя способами: программой подготовки диска, поставляемой производителем накопителя, аналогичной программой, встроенной в ПЗУ расширения BIOS на плате контроллера или материнской плате ПК, или программами подготовки диска третьих фирм.

- Во всех трех способах используется один и тот же процесс — возможность контроллера проводить форматирование низкого уровня.
- Некоторые жесткие диски поставляются с уже записанным форматом низкого уровня. Такие накопители обязательно поставляются с соответствующим контроллером. Примерами служат платы жестких дисков, накопители SCSI и некоторые комплекты накопителя (ESDI, RLL или MFM) и контроллера.
- Форматирование некоторых заранее сформатированных дисков может оказаться разрушительным, если пользователь (или программа) пытается повторить форматирование низкого уровня. Большинство заранее сформатированных дисков защищает себя и отказывается проводить форматирование низкого уровня за исключением подачи специальных команд, о которых знают только специалисты фирмы-производителя.

В этой главе рассмотрен первый этап подготовки нового накопителя для работы с DOS. В следующей главе речь пойдет о втором этапе, который называется организацией разделов.

Глава 9. Подготовка жесткого диска к хранению информации.

Этап 2 — организация разделов

В этой главе вы узнаете

- Что такое организация разделов и зачем нужны разделы
 - На какие области жесткий диск разделяется при организации разделов
 - Как DOS следит за разделами жесткого диска
 - Что такое расширенная таблица разделов и когда она применяется
 - Как узнать, какие разделы и логические тома имеет накопитель
 - Как изменить организацию разделов
 - Важное предупреждение о слишком простой ошибке при использовании команды DOS FDISK
 - Что делают типичные программы организации разделов третьих фирм
 - Почему нецелесообразно применять программы организации разделов третьих фирм,
 - Как поместить на жесткий диск информацию с личной идентификацией там, где ее почти никто не найдет, но вы всегда сможете легко доказать, что компьютер ваш.
-

9.1. Зачем на жестком диске нужны разделы

После проведения форматирования низкого уровня жесткий диск готов к хранению информации. Другими словами, контроллер, который поместил на диск формат низкого уровня, может теперь найти любое указанное место и записать туда переданные данные. Он может также по команде найти и вернуть эти данные.

Контроллеру достаточно формата низкого уровня, но этого далеко недостаточно для DOS. До сохранения файлов в компьютере с MS DOS необходимо организовать на диске разделы и провести форматирование высокого уровня. Возвращаясь к аналогии со стоянкой автомашин, можно сказать, что определены места для автомашин, но нужно построить какие-то ограждения и будку для обслуживающего персонала.

На гибких дисках разделов нет. Разработчики DOS могли не вводить разделов и на жестких дисках, но они потребовались из-за высокой стоимости жестких дисков в пору "молодости" ПК и DOS. Разработчики DOS чувствовали, что нужно разрешить пользователям устанавливать на ПК несколько операционных систем, включая DOS фирм IBM и Microsoft, P-system фирмы Softech и CP/M-86 фирмы Digital Research. Для этого они разрешили каждой операционной системе использовать часть (или раздел) очень дорогого жесткого диска.

Сейчас стоимость хранения 1 Мбайт на жестком диске составляет примерно 1% стоимости в пору "молодости" DOS. Выделение всего накопителя только одной операционной системе теперь выглядит менее экстравагантно, чем раньше, но конструкция ПК требует организации разделов на жестком диске ради совместимости с прежними версиями DOS и BIOS.

9.2. Разделение диска

Организация разделов означает деление на части. Почти все дисковое пространство поделено на разделы, в которых можно хранить файлы информации. Однако ваш компьютер не знает об этом и не может использовать эти части без специальной "вводной" секции на диске. Такая секция, она находится в самом начале диска (цилиндр 0, головка 0, физический сектор 1), содержит так называемую главную загрузочную запись (Master Boot Record — MBR). Эта односекторная программа часто называется таблицей разделов по имени находящейся в ней таблицы данных.

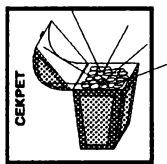
В самом конце диска обычно имеются одна или две другие специальные части — диагностический цилиндр и, в некоторых контроллерах, секретный цилиндр для хранения параметров накопителя.

На рис. 9.1 показаны три основные секции, которые создаются при организации разделов (главная загрузочная запись, средняя часть и диагностический цилиндр плюс секретный цилиндр, если он есть). На рисунке также показано, как можно делить среднюю часть.

9.2.1. Главная загрузочная запись и таблица разделов

Самый первый сектор накопителя (цилиндр 0, головка 0, физический сектор 1) называется главной загрузочной записью, или таблицей разделов. Этот сектор называется таблицей разделов потому, что он содержит таблицу данных, которая определяет области (разделы), на которые разделено почти все дисковое пространство. Сектор также называется главной загрузочной записью, потому что он содержит программу, которая считывает эту таблицу, переходит к правильному разделу и загружает из самого начала этого раздела другую загрузочную запись.

Если это раздел является загрузочным разделом DOS, программная часть загрузочного сектора в начале раздела идентична программной части загрузочного сектора любого диска или дискеты DOS. Если же это, например, раздел UNIX, то загрузочный сектор содержит стандартную программу загрузки UNIX.



Главная загрузочная запись занимает всего один сектор (цилиндр 0, головка 0, сектор 1). DOS 2.x выделяет для главной загрузочной записи только этот один сектор; первый раздел диска в этой версии DOS начинается со следующего сектора (цилиндр 0, головка 0, сектор 2). Начиная с DOS 3.0, с учетом все большей емкости жестких дисков и стремления к простоте DOS "бесполезно расходует" всю первую дорожку. Другими словами, первый раздел на дисках с современными версиями DOS начинается с цилиндра 0, головки 1 и сектора 1.

Только некоторые контроллеры жестких дисков используют это пропадающее пространство. Например, контроллеры фирмы Western Digital хранят здесь параметры накопителя. Некоторые антивирусные программы помещают копию MBR в следующий сектор этой первой дорожки и позднее (при "заражении" основной MBR) могут скопировать ее оттуда. А некоторые вирусы копируют MBR в сектор где-то в середине дорожки или помещают туда свой код.

Если части этой дорожки не используются, туда можно поместить все что угодно, но вначале следует проверить, что не занята ли она. Остаток дорожки не используется при соблюдении следующих условий:

- ✓ контроллер не использует остаток дорожки;
- ✓ вы не используете те части дорожки, которые нужны некоторым применяемым вами антивирусным программам;
- ✓ вы не используете те части дорожки, которые перезаписываются тем вирусом, который может "подхватить" ваш компьютер.

Что же можно поместить на эту дорожку? Некоторые предложения приведены в конце главы.

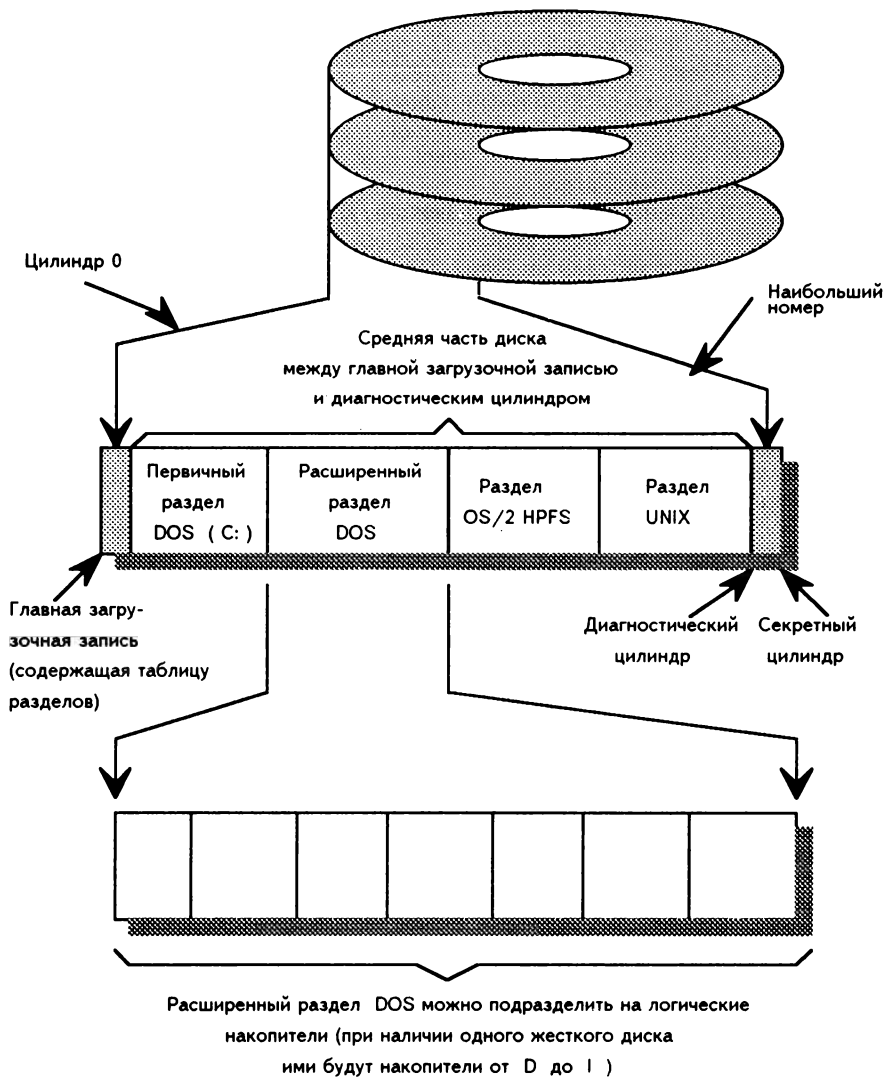


Рис. 9.1. Области, создаваемые на диске при организации разделов

При наличии двух накопителей на жестких дисках главная загрузочная запись на первом из них считывает таблицу данных для обоих накопителей. Оба накопителя имеют полную программу MBR и таблицу разделов. Это значит, что вы даже не заметите, что программа MBR (но не таблица разделов) испорчена в первом секторе второго физического накопителя, если не посмотрите его таким средством анализа диска, как программа Disk Editor из утилит Нортон. С другой стороны, такое повреждение не скажется на вашей работе, если вы в дальнейшем не попытаетесь использовать этот жесткий диск как первый физический накопитель в ПК.

Можно вывести содержимое первого сектора с помощью программы DOS DEBUG или другой программы. Особенно удобно пользоваться программой Disk Editor из утилит Нортон (версия 5+). Она автоматически запускается в режиме только считывания, и у вас меньше шансов повредить MBR при просмотре. При включении защиты только считывания программой Disk Editor можно записать MBR в файл, а при повреждении MBR восстановить ее из файла.

Вывод программы DEBUG любого файла или области памяти имеет вид, показанный на рис. 9.2. Это фактическое содержимое MBR, созданной DOS 5 на основном жестком диске (модель 1598-15 накопителя SCSI емкостью 1 Гбайт фирмы Micropolis) компьютера автора.

Относительный адрес	16-ричные байты данных	Данные в коде ASCII
0000	FA 33 C0 8E D0 BC 00 7C - 8B F4 50 07 50 1F FB FC	.3..... ..P.P...
0010	BF 00 06 B9 00 01 F2 A5 - EA 1D 06 00 00 BE BE 07	...<...<...u...
0020	B3 04 80 3C 80 74 0E 80 - 3C 00 75 1C 83 C6 10 FE	..u...<...L...<...
0030	CB 75 EF CD 18 8B 14 8B - 4C 02 8B FE 83 C6 10 FE	..t...<...t...<...
0040	CB 74 1A 80 3C 00 74 F4 - BE 8B 06 AC 3C 00 74 0B	V.....s...3...
0050	56 B8 07 00 B4 0E CD 10 - 5E EB F0 EB FE BF 05 00	W...s...3...
0060	BB 00 7C BB 01 02 57 CD - 13 5F 73 0C 33 C0 CD 13	0u..... ...=
0070	4F 75 ED BE A3 06 EB D3 - BE C2 06 BF FE 7D 81 3D	U...u..... ...Inval
0080	55 AA 75 C7 8B F5 EA 00 - 7C 00 00 49 6E 76 61 6C	id partition tab
0090	69 64 20 70 61 72 74 69 - 74 69 6F 6E 20 74 61 62	le Error loading
00A0	A0 6C 00 45 72 72 6F 72 - 20 6C 6F 61 64 69 6E 67	operating syste
00B0	20 6F 70 65 72 61 74 69 - 6E 67 20 73 79 73 74 65	m.Missing operat
00C0	60 00 4D 69 73 73 69 6E - 67 20 6F 70 65 72 61 74	ing system.....
00D0	69 6E 67 20 73 79 73 74 - 65 6D 00 00 00 00 00 00	
00E0	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
00F0	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0100	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0110	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0120	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0130	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0140	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0150	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0160	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0170	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0180	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
0190	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
01A0	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
01B0	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	
01C0	01 00 04 3F 20 1E 20 00 - 00 00 E0 F7 00 00 80 00	?
01D0	C1 DE A0 3F E0 DE 00 F0 - 1E 00 00 08 00 00 00 00	?
01E0	00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00	?
01F0	01 1F 05 3F E0 DD 00 F8 - 00 00 00 F8 1D 00 55 AA	?.....U.

Рис. 9.2. Типичная главная загрузочная запись (MBR). Содержимое сектора диска показано в формате программы DEBUG. Данные таблицы разделов находятся ближе к концу; программа занимает примерно первую половину сектора

Колонки на рисунке сведены в три группы. Первая показывает 16-ричные адреса. Вторая группа показывает также 16-ричные фактические байты, хранимые по этим адресам. Наконец, в третьей части показаны те же самые данные в виде символов кода ASCII.

Вторая и третья части содержат по 16 байт данных в строке. Поэтому вторая часть состоит из 16 подколонок, каждая из которых содержит один байт. В третьей части все значения, соответствующие непечатаемым символам, показаны точками. Адрес в первой части каждой строки равен числу байтов, находящихся в секторе до первого байта данных, показанного в этой строке.

На рисунке показаны 32 строки, и адреса содержат числа от нуля до 01F0h. Если прибавить число байтов последней строки перед последним байтом (15 или Fh), получится адрес 1FFh (десятичное число 511) для последнего адреса сектора. Адреса от нуля до 511 означают, что показаны 512 байт, что равно размеру сектора.

Отметим, что программа MBR содержится в первых 14 строках (меньше половины сектора) и что последние 80 байт (пять строк) заполнены текстами сообщений, которые программа выводит на экран. Данные таблицы разделов содержатся в последних пяти строках рисунка, начиная с адреса 01BEh. Почти половина сектора от адреса 1DAh до 1BDh, т.е. 227 байт, не используется и заполнена нулями.

Последних два байта в секторе (со значениями 55h AAh по адресам 1FEh и 1FFh) образуют "сигнатуру", которая помогает программе загрузки (это программа в ROM BIOS на материнской плате, которая запускает компьютер) проверить правильность загрузочной записи.

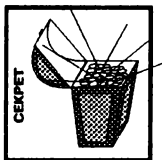
Большая часть диска находится между главной загрузочной записью и диагностическим цилиндром. Это пространство можно использовать как один раздел или разбить его на два-четыре раздела. Таблица разделов показывает начальный и конечный адреса (цилиндр, головка и физический сектор) для каждого раздела. Кроме того, для каждого раздела она показывает операционную систему, которая образует в разделе формат высокого уровня. Наконец, таблица показывает, какой из разделов является активным (загрузочным).

На рис. 9.3 подробно показано содержимое загрузочной записи. Этот пример взят из второго жесткого диска (накопитель ESDI емкостью 335 Мбайт фирмы Micropolis) компьютера автора.

Задача читателю

Разобравшись, как читать данные главной загрузочной записи, проверьте правильно ли вы понимаете изложенное, набросав, как выглядит таблица разделов с рис. 9.2, если ее представить в формате рис. 9.3. Проверьте, имеют ли ваши числа смысл для накопителя. SCSI (посмотрите в главе 3, какими должны быть параметры накопителя).

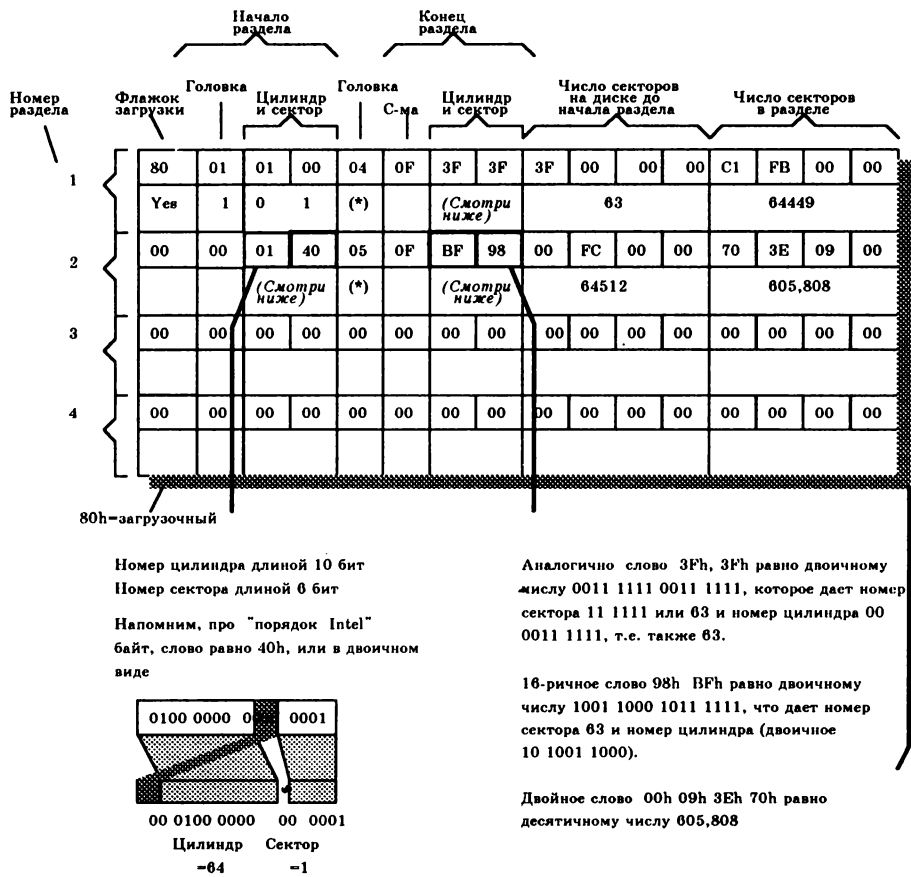
Предупреждение. В этой таблице разделов имеется одно нестандартное для DOS значение. Ответ на задачу и значение нестандартного элемента приведены в конце этой главы.



Номера сектора и цилиндра для начала и конца каждого раздела хранятся в соответствии с ограничением INT 13h на диапазон этих номеров. (Напомним, что INT 13h — это прерывание BIOS, которое управляет дисковыми операциями считывания и записи. Оно ограничивает номера цилиндров в диапазоне от 0 до 1,023 и секторов от 1 до 63. Кроме того, по крайней мере официально оно ограничивает номера головки от 0 до 15.

Таким образом, это одно из мест, где BIOS накладывает ограничение на емкость диска в 504 Мбайт. Напомним, что это ограничение вытекает из 512-байтовых секторов и "официального ограничения" INT 13h иметь не более 1,024 цилиндров, 63 секторов на дорожке и 16 головок. Поскольку в таблице разделов и в INT 13h номера головки могут быть любыми числами вплоть до 255, ограничение возрастает до

8 Гбайт. Превысить это ограничение можно, только "нарушив правила" процедуры INT 13h BIOS.



(*) DOS различает такие значения системного байта

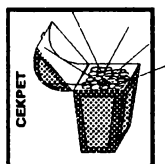
- 0 — необычный раздел
- 1 — первичный раздел DOS с 12-битовой FAT
- 4 — первичный раздел DOS с 16-битовой FAT
- 5 — расширенный раздел DOS
- 6 — "огромный" раздел DOS (свыше 32 Мбайт)

Рис. 9.3. Структура таблицы разделов в главной загрузочной записи

Общее число секторов и число секторов от начала диска до начала каждого раздела хранятся в четырехбайтовых ячейках (по 32 бита). Это условие совместимо с DOS 4.0 для больших разделов жесткого диска; 32 бита для номера сектора позволяют иметь 2^{32} секторов. Если каждый сектор содержит 512 байт, емкость накопителя может достигать до 2 Тбайт, т.е. в 4,000 раз больше, чем максимальная емкость, допустимая в BIOS максимальными номерами сектора, головки и цилиндра.

9.2.2. Диагностический цилиндр

Диагностический цилиндр (или, по терминологии фирмы IBM, инженерный цилиндр потребителя) — это один полный цилиндр (все головки на дорожке). Обычно это последний, ближайший к шпинделю цилиндр. Фирма IBM сообщает, что этот цилиндр предназначен для проведения тестов без искажения данных, хранимых в остальной части диска.



Некоторые контроллеры не резервируют диагностический цилиндр. Этот цилиндр был стандартной частью конструкции жесткого диска фирмы IBM, поэтому многие утилиты рассчитаны на него. Например, программа SpinRite проверяет наличие зарезервированного диагностического цилиндра в конкретном накопителе и использует его. Она также извещает об отсутствии такого цилиндра и отказывается выполнять некоторые операции, предотвращая случайную перезапись данных в накопителе.

9.2.3. Секретный цилиндр

Последней частью, выделяемой системой на диске, является так называемый секретный цилиндр. Он есть не во всех накопителях, но при наличии автоконфигурирующего контроллера или накопителей ESDI, IDE/ATA или SCSI он обычно есть; в этот цилиндр контроллер помещает параметры накопителя и некоторую другую полезную информацию. Обычно этот цилиндр расположен еще ближе к шпинделю, чем диагностический.

9.3. Где хранятся файлы

Средняя часть диска (которая равна почти всей его емкости) предназначена для хранения информации пользователя. Остальные области до и после нее являются служебными, но информация в первой секции (главная загрузочная запись) абсолютно необходима для доступа к данным, хранимым в остальной части диска.

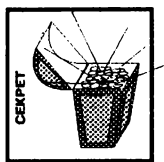
9.3.1. Активный или загрузочный раздел

Из двух, трех или четырех разделов, на которые разбита средняя часть диска, загрузить можно только из одного. Первый байт в каждой строке таблицы разделов называется флажком загрузки. Только одному из этих четырех байтов разрешается быть ненулевым. Этот единственный байт содержит 80h, показывая активный раздел. При загрузке компьютера этот флажок сообщает программе MBR, к какому разделу обращаться за программой загрузочного сектора.

9.3.2. Первичный раздел DOS

Активный раздел может создать и форматировать любая операционная система, какая есть в ПК. Если этот раздел форматирует DOS, он называется первичным разделом DOS. (Метки в разделах помещаются программой DOS FDISK или соответствующей программой другой операционной системы.) В DOS 3.2 все остальные раз-

дела игнорируются. DOS 3.3 после загрузки из одного раздела может обращаться к данным еще в одном разделе, если его отметить как "расширенный раздел DOS". Сам расширенный раздел DOS не может быть загрузочным.



Все версии DOS до 4.0 могли работать только с логическим томом емкостью не более 32 Мбайт. (Логический том — это то, что DOS идентифицирует буквой накопителя.) Напомним, что это ограничение связано с представлением логического номера сектора 16-битовым числом и размером сектора 512 байт.

DOS 3.3 ограничена логическим томом 32 Мбайт, но она может иметь несколько томов в физическом накопителе. Можно иметь один том, являющийся первичным разделом DOS (именно он является загрузочным), и любое число в одном расширенном разделе DOS.

Для каждого раздела в таблице разделов имеется системный байт, показывающий тип раздела. Имеются четыре возможных значения этого байта для любого раздела, к которому может обращаться DOS. Значения 1 или 4 сообщают, что это обычный раздел DOS; только такие разделы создавались DOS 3.2. (Два разных значения показывают размер элементов в таблице FAT — глава 10.) Значение 5 показывает расширенный раздел DOS, который может создать только DOS 3.3. Значение отведено для "огромного" раздела DOS, потенциально загрузочного раздела, создаваемого DOS 4.0, с емкостью более 32 Мбайт. Значения системного байта помещает в таблицу разделов программа FDISK. Программы организации разделов третьих фирм или не-DOS операционные системы, например OS/2 или UNIX, помещают в системные байты другие значения.

При использовании для организации разделов программ третьих фирм помните, что они могут помещать в системные байты специальные не-DOS значения для всех разделов, кроме первичного раздела DOS; для обращения к данным из этих специальных разделов требуется особый устанавливаемый драйвер устройства, который обычно поставляется с программой организации разделов. (Например, Disk Manager использует драйвер устройства DMDRVR.BIN для доступа к логическим накопителям, создаваемым в расширенном разделе DOS.) Без такого драйвера сама DOS не может получить никаких данных, хранимых в таком разделе.

Если для форматирования раздела применяется операционная система, отличная от DOS, она помещает в системный байт свое идентифицирующее значение. В общем, это значение делает раздел "отсутствующим" для всех систем, кроме данной. (Некоторые операционные системы, которые могут считывать файлы из разделов других систем, являются исключениями — об этом будет сказано дальше.)

9.3.3. Расширенный раздел DOS

DOS рассматривает расширенный раздел DOS как отдельный диск или совокупность отдельных дисков. Его элемент в таблице разделов (в главной загрузочной записи) выглядит так же, как для любого раздела, но DOS использует его по-другому. Поскольку этот раздел выглядит как другой диск, можно образовать расширенный раздел DOS любой емкости (вплоть до 504 М или 8 Гбайт), даже если вы работаете с DOS 3.3. Элементы в таблице разделов служат указателями этой области.

Обычно программа главной загрузочной записи считывает первый сектор в активном разделе и ожидает, что она является правильной программой загрузочного сектора. Для расширенного раздела DOS программа главной загрузочной записи переходит к соответствующей ячейке, но вместо поиска и загрузки загрузочного сектора она отыскивает другую таблицу разделов. Эта вторичная, или расширенная, таблица разделов показывает дальнейшее разбиение расширенного раздела DOS на логические тома DOS. При работе с DOS 3.3 эти логические тома должны быть не больше 32 Мбайт.

Таблица разделов в начале каждого логического тома DOS внутри расширенного раздела DOS аналогична таблице разделов главной загрузочной записи, но с двумя

исключениями: она не содержит программу главной загрузочной записи и ее таблица данных имеет только один или два элемента. Первый элемент сообщает емкость текущего доступного DOS подраздела. Второй элемент показывает начало следующего логического тома DOS (рис. 9.4). Третий и четвертый элементы заполнены нулями. Таблица разделов для последнего логического тома в расширенном разделе DOS имеет нули и во втором элементе.

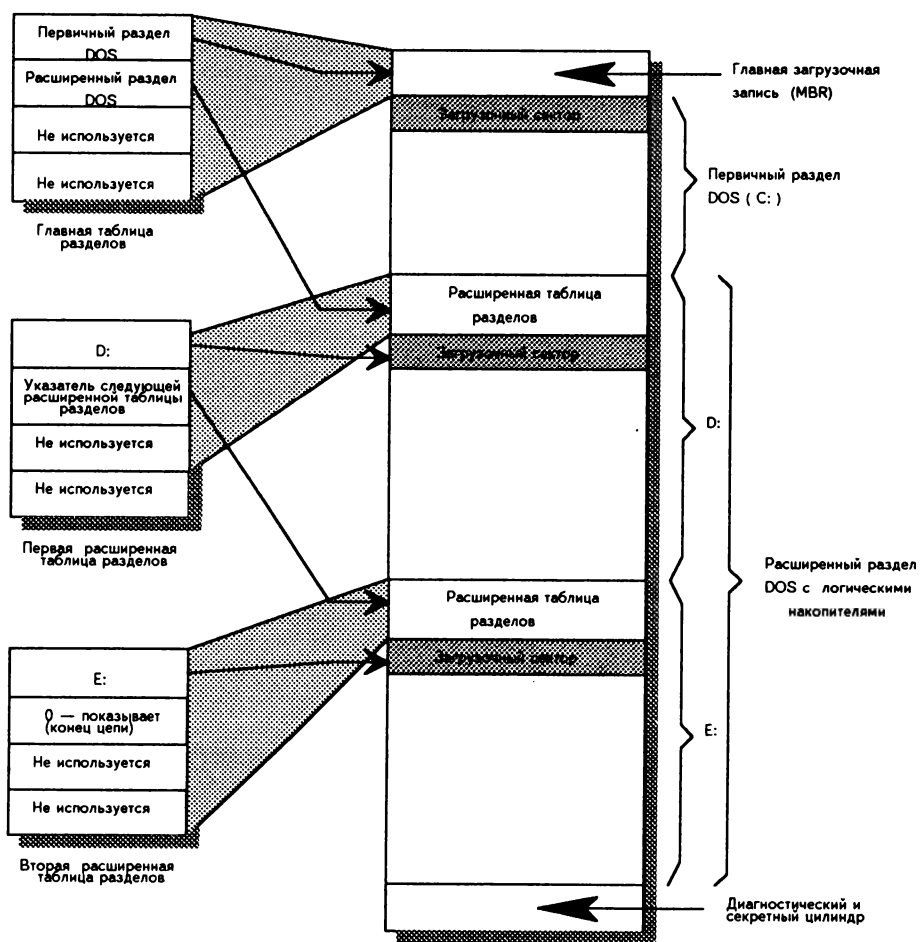


Рис. 9.4. Как главная и расширенная таблицы разделов разбивают диск на логические тома (буквы накопителей)

9.3.4. Другие виды разделов

Помимо первичного раздела DOS и расширенного раздела DOS, на жестком диске могут быть еще один или два раздела. Эти разделы предназначены для использования другой операционной системой.

При загрузке DOS другие разделы невидимы. При загрузке другой операционной системы она обращается к своему разделу и не обращается к разделам DOS.

9.3.5. Не-DOS операционные системы, которые могут использовать первичный и расширенный разделы DOS

К первичным и расширенным разделам DOS могут обращаться несколько других операционных систем. Одной из них является система OS/2, а второй объявленная, но пока не поставляемая система Windows NT, очень похожая на OS/2.

Систему OS/2 фактически можно поместить в логический том DOS — в первичный раздел или один из логических томов внутри расширенного раздела. Но OS/2 можно также применить для реформатирования одного из логических томов DOS для файловой системы OS/2, называемой HPFS (High Performance File System — высокоэффективная файловая система). Пока OS/2 использует файловую систему с FAT, ее раздел и все ее файлы доступны DOS; кроме того, все файлы DOS доступны OS/2.

При создании логических накопителей с HPFS они будут доступны только OS/2. DOS 5 может обращаться ко всем логическим накопителям с системой FAT даже в расширенном разделе DOS. Предыдущие версии DOS могли работать только с томами, использующими FAT, которые расположены перед первым томом HPFS.

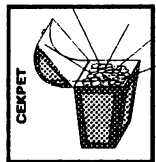
Windows NT будет поддерживать три файловые системы. Ее можно устанавливать и применять в томе с FAT или в томах, форматированных с ее системой NTFS. Windows NT сможет считывать все файлы в любых томах с FAT, HPFS или NTFS.

9.4. Таблица разделов

В DOS имеется программа FDISK для создания, просмотра и изменения таблиц разделов. Кроме работы с разделами DOS, программа FDISK может зарезервировать место для разделов других операционных систем, но больше с этими разделами она сделать ничего не может, так как рассчитана только на DOS.

С помощью программы FDISK можно отметить любой из четырех разделов в главной загрузочной записи (кроме расширенного раздела DOS, если он есть) как активный или загрузочный. Таким образом, программа FDISK может "выключить" DOS и "включить" UNIX или другую операционную систему, находящуюся в одном из других разделов.

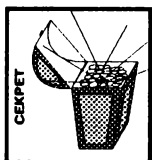
При выполнении FDISK на новом диске до внесения любых изменений необходимо посмотреть, какую информацию о разделах может найти программа. Если на диске проведено форматирование низкого уровня, FDISK не найдет главной загрузочной записи/таблицы разделов. В этом случае следует запросить FDISK создать ее.



Применяя программу FDISK, можно по-разному повредить диск. Например, можно неправильно выбрать правильный диск для работы FDISK или разрешить FDISK изменить или даже вновь подтвердить уже имеющуюся в таблице разделов информацию.

О дополнительных опасностях вы прочтаете в следующем предупреждении. Однако есть простой прием для того, чтобы избежать хотя бы одной проблемы: если вы собираетесь применить программу FDISK для вновь подключенного диска, сначала отсоедините имеющийся жесткий диск. Ясно, что при этом FDISK не сможет разрушить имеющиеся данные. После окончания работы с FDISK следует подключить первый и новый диски, осуществить перезагрузку и продолжать свою работу.

При наличии в ПК более одного физического жесткого диска программа FDISK имеет в основном меню опцию для выбора накопителя. Если не выбрать эту опцию, FDISK обращается к первому диску (который она называет Накопитель 0 — обычно накопитель C). Обязательно попросите FDISK показать найденную информацию — она поможет вам выбрать правильный накопитель.



Будьте очень внимательны при работе с диском, который уже содержит главную загрузочную запись. Изменение чисел в таблице разделов приведет как бы к исчезновению всех данных. Можно восстановить данные (или их большую часть), если восстановить значения в таблице разделов, но эти значения нужно знать. Если вы изменяете эти числа программой, допускающей прямое редактирование диска, вы будете изменять только числа в таблице разделов. Потом можно отменить ваши действия, просто возвращая первоначальные числа. Если вы изменяете таблицу разделов с помощью программы FDISK, она обычно стирает самую критическую часть информации в создаваемом разделе, препятствуя легкому восстановлению (обратите внимание на "Важное предупреждение", помещенное ниже).

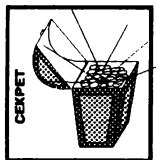
Интересным примером того, как другие операционные системы, используют разделы, и того, что вы можете встретить в ПК, которые обычно работают как машины DOS, является Boot Manager (Менеджер загрузки) системы OS/2. Эта программа, появившаяся в OS/2 2.0, сама занимает раздел 1 Мбайт. Этот раздел отмечается в таблице разделов как активный. При загрузке компьютера с установленным Boot Manager машиной управляет эта программа.

Привлекая внутреннюю таблицу данных, Boot Manager выводит меню, показывающее, какие операционные системы установлены в других разделах, и позволяющее вам выбрать одну из них для управления ПК. После выбора (или истечения тайм-аута, после которого принимается выбор по умолчанию) Boot Manager переходит к указанному разделу, загружает и выполняет его загрузочный сектор.

9.4.1. Этапы организации разделов на новом диске

Предполагая, что загрузочный накопитель C отсутствует, необходимо вначале загрузить компьютер с дискеты DOS в накопителе A. Держите дискету DOS в удобном месте — придется перезагружать компьютер с этой дискеты несколько раз и даже многократно, прежде чем компьютер будет готов к загрузке с жесткого диска.

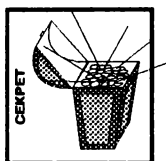
Если емкость жесткого диска не более 32 Мбайт, ваша работа оказывается простой. Вначале вы выполняете программу FDISK, разрешая ей использовать весь диск для DOS, а затем переходите к форматированию высокого уровня, рассмотренному в главе 10.



В этом случае программа FDISK сразу выполняет несколько операций. Во-первых, она создает главную загрузочную запись с таблицей разделов. Затем она отмечает начальную и конечную точки первого раздела как начало и конец дискового пространства (за исключением сектора или дорожки для главной загрузочной записи и диагностического цилиндра в конце). FDISK отмечает этот раздел как принадлежащий DOS и делает его активным. Остальных три элемента в таблице разделов заполняются нулями, показывающими, что они не используются.



Важное предупреждение. В работе FDISK присутствует малоизвестный момент, который может привести к потере данных. Помните о нем — это когда-нибудь спасет ваши данные! Самый неприятный момент в работе FDISK — она уничтожает примерно первую половину цилиндров каждого создаваемого ею раздела. Это действие очень затрудняет отмену операций FDISK.



Это действие вызвано тем, что программа FDISK стремится, чтобы раздел не имел формата высокого уровня. Если бы она просто помещала числа в таблицу разделов, ее операции можно было бы отменить, но вы бы оставались с "мусором" в новом разделе, который, по мнению разработчиков DOS, может иногда мешать.

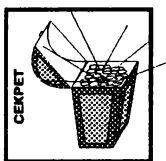
Как полностью использовать жесткий диск. Можно не отводить весь диск для DOS, а оставить его часть для другой операционной системы; собственно, это и было причиной появления разделов и программы FDISK. Однако для накопителя с емкостью менее 32 Мбайт такое решение непрактично. Современные версии большинства операционных систем требуют намного больше, чем половина диска, чтобы сделать что-либо полезное. Кроме того, сейчас большинство пользователей DOS считают емкость меньше 16 Мбайт недостаточной для DOS и файлов данных приложений DOS.

При емкости диска больше, чем 32 Мбайт, у вас есть три варианта в зависимости от применяемой версии DOS.

1. Для DOS 3.2 единственным способом использования всего диска являются программы организации разделов третьих фирм. Другие решения состоят в переходе к более новой версии DOS или игнорировании емкости диска сверх 32 Мбайт (но кто этого хочет?). Программы организации разделов рассматриваются в следующем разделе главы.
2. DOS 3.3 позволяет использовать диск почти любой емкости, разбивая его на области не более 32 Мбайт.
3. С DOS 4.0 можно использовать диск любой емкости (до 500 Мбайт) как один большой раздел или разделить его. Перед решением разбить большой диск в DOS 4.0 посмотрите материал главы 7.

При выборе одного большого раздела в DOS 4.0 процесс оказывается таким же, как и для диска малой емкости (до 32 Мбайт) в предыдущих версиях DOS. Позвольте программе FDISK работать с принимаемыми по умолчанию опциями, а затем переходите к форматированию высокого уровня.

Организация более одного тома в одном накопителе. Выбор разбиения диска проходит в несколько этапов. Сначала нужно выполнить FDISK и запросить ее создать первичный раздел DOS, емкость которого меньше емкости диска. Команда FDISK DOS 3.3 не позволяет задать емкость раздела больше 32 Мбайт. Затем нужно образовать расширенный раздел DOS, назначая ему оставшуюся емкость диска, если вы хотите использовать весь диск с DOS. Выполните это действие, даже если вы хотите иметь несколько меньших разделов.



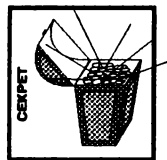
На этих этапах программа FDISK вновь создает главную загрузочную запись и ее таблицу разделов. Она заполняет первый элемент для первичного раздела DOS с выбранной вами емкостью этого раздела и отмечает этот раздел как принадлежащий DOS. После этого FDISK заполняет второй элемент таблицы разделов для расширенного раздела DOS. В остальные два элемента таблицы разделов помещаются нули.

Программа FDISK ранних версий DOS отмечает созданный ею первичный раздел DOS как активный, разрешая загрузку из этого раздела после форматирования высокого уровня. Новые версии DOS автоматически этого не делают. Необходимо дать команду FDISK отметить активный раздел как отдельный этап, если вы не выделяете весь диск для одного раздела DOS.

Не нужно создавать первичный раздел DOS, если на жестком диске образуется расширенный раздел DOS. Помните, однако, что нельзя загрузить ПК, используя DOS, из расширенного раздела DOS. (Можно загрузить OS/2 2.0 с логического накопителя в расширенном разделе DOS.) Если вы планируете применять диск как второй диск в ПК, целесообразно создать на нем только расширенный раздел DOS. После этого его буквы накопителей будут следовать после букв раздела DOS и логических накопителей в расширенном разделе DOS первого диска.

В некоторых версиях DOS программа FDISK спрашивает, сколько цилиндров выделять каждому создаваемому разделу. В других версиях она спрашивает емкость разделов в мегабайтах. Если ваша версия DOS относится к последним, вам труднее будет точно запросить запрашиваемую емкость, так как за исключением первого раздела программа FDISK всегда выделяет каждому создаваемому разделу целое число цилиндров. Первый раздел начинается сразу после загрузочной записи либо в следующем секторе (DOS 2.x), либо с начала следующей дорожки (DOS 3.0). Первый раздел, как и все остальные, заканчивается в конце некоторого цилиндра.

Теперь также с помощью программы FDISK необходимо создать логические накопители DOS внутри расширенного раздела DOS. Программа FDISK в DOS 3.3 ограничивает емкость каждого из этих накопителей 32 Мбайт. В DOS 4.0 можно определять для накопителей любую емкость. Разрешается создавать столько логических накопителей (любой емкости), сколько поместится на диск. Та часть расширенного раздела DOS, которая не распределена логическому накопителю DOS, остается недоступной для использования DOS.



На этом этапе FDISK создает таблицы разделов внутри расширенного раздела DOS, по одной для каждого подраздела или логического диска DOS. Каждая из этих таблиц имеет одну строку для данного подраздела, и все таблицы, кроме последней, имеют вторую строку, которая указывает на следующую таблицу расширенного раздела. Эта вторая строка показывает емкость раздела как общую емкость расширенного раздела, еще не занятую этим и предыдущими логическими дисками DOS. Последних два элемента (три в таблице последнего подраздела) заполняются нулями.

После создания главной загрузочной записи, включая таблицу разделов, можно форматировать первичный раздел DOS. При наличии расширенного раздела DOS после образования в нем логических накопителей DOS может форматировать каждый из них. Для программы DOS FORMAT каждый из разделов или логических дисков DOS выглядит почти так же, как гибкий диск. Единственное различие состоит в том, что FORMAT знает о несменном диске и его емкости.

9.4.2. Программы организации разделов третьих фирм

Если у вас накопитель емкостью больше 32 Мбайт и DOS 3.2, вам придется перейти на новую версию DOS, воспользоваться программами организации разделов третьих фирм или игнорировать объем накопителя, превышающий 32 Мбайт.

Программы третьих фирм могут управлять всеми тремя частями процесса: форматированием низкого уровня, организацией разделов и форматированием высокого уровня. Такие программы разрабатываются несколькими фирмами; с особенностями работы программы необходимо ознакомиться по документации. В режиме по умол-

чанию работать с программой очень просто, но для лучшего управления процессом следует использовать интерактивный режим. Предыдущий материал должен помочь разобраться в организации разделов и правильном выборе.

Важное напоминание. Программы организации разделов третьих фирм обеспечивают несовместимое с DOS решение проблемы, которую не могли решить версии DOS до 3.3, превращая диск в полностью несовместимый с процедурами DOS. Для доступа ко всему, кроме первого раздела, создаваемого этими программами (который часто невелик), необходимо установить через файл CONFIG.SYS драйвер устройства. Загрузка с копии дистрибутивного диска DOS не помогает.

Программы сами устанавливают требуемый драйвер устройства, но обязательно нужно иметь резервные копии данных на другом носителе, а не на жестком диске. Резервы должны обеспечивать загрузку и содержать все требуемые драйверы устройств с конфигурацией файлов под накопитель на гибком диске. Проверьте безопасную загрузочную дискету после ее создания, чтобы убедиться, что доступ к жесткому диску возможен.

9.4.3. Рекомендация

При переходе к DOS 3.3 не пользуйтесь программами организации разделов третьих фирм. При установке новой версии DOS заодно модернизируйте и разделы диска. Для этого необходимо вначале зарезервировать все данные. Затем загрузитесь с нового диска DOS и воспользуйтесь программой FDISK новой версии DOS, которая сделает все данные недоступными. Удалите все расширенные разделы DOS и первичный раздел DOS. (Для удаления разделов вам придется воспользоваться той программой, которая создавала их.) Вновь создайте разделы (первичный раздел DOS и расширенный раздел DOS) с выбранными емкостями. После этого выполните программу FORMAT (также из новой версии DOS). На этом последнем этапе готовятся разделы и устанавливается новая DOS на диске C.

При модернизации помните о следующем.

- ✓ Командой для загрузочного первичного раздела DOS является FORMAT C: /S /V. Для других разделов требуется команда FORMAT d: /V, где *d* заменяется подходящей буквой накопителя.
- ✓ Проверьте модернизацию всех файлов DOS, копируя их с гибкого диска на жесткий диск (воспользуйтесь командой COPY *.* и обеспечьте копирование в подкаталог C:\DOS). Затем верните все файлы данных и программ.
- ✓ При возвращении данных и программ можно поместить их в первоначальные каталоги или реорганизовать жесткий диск.
- ✓ Остерегайтесь "ленивой" программы installations, полученной от дилера или приятеля, которая будет продолжать программу организации разделов после перехода к DOS 3.3. Попросите специалиста удалить ненужные программы или удалите их самостоятельно.

9.5. Предложение (только для смелых читателей!)



Если вы не работаете с DOS 2.x, на жестком диске будет пространство между главной загрузочной записью и началом первого раздела. (Обычно это вся первая дорожка, кроме первого сектора.)

Некоторые контроллеры диска используют все это пропадающее пространство или его часть. Какие-то антивирусные программы используют один сектор, а какие-то компьютерные вирусы записываются в один из секторов этой области. Кроме таких участков остальное пространство доступно вам для любого применения (конечно, кроме хранения обычных файлов DOS, так как DOS не "видит" это пространство). Автор предлагает новое применение для этого пространства.

Иногда компьютеры воруют, а иногда украденный компьютер полиция даже находит. Но если владелец не может доказать право на свою собственность, компьютер не возвращают. Большой жесткий диск можно вынуть из ПК, поэтому выгравированный лицензионный номер накопителя внутри системного блока ПК не помогает идентифицировать накопитель.

Но если на украденном у вас компьютере вор не успел провести форматирование низкого уровня (даже если раздел DOS сформатирован для удаления всех программ и другой информации), можно доказать, что диск ваш, если ваши фамилия, адрес, номер телефона и другая идентифицирующая информация записаны в один или несколько секторов пропадающего пространства на первой дорожке. (Можно записать несколько копий такой информации; если один из секторов будет перезаписан вирусом или антивирусом, остальные копии останутся для идентификации.)

Такая небольшая мера предосторожности поможет когда-нибудь вернуть ценный жесткий диск. Но если вы решили принять это предложение, будьте очень внимательными. Проверьте и дважды проверьте свое намерение до его осуществления. Воспользуйтесь программой Disk Editor из утилит Нортон для просмотра каждого сектора первой дорожки жесткого диска. Убедитесь, что дорожка совершенно пуста, а если нет, проверьте, какие секторы заняты или свободны. После этого поместите идентификационную информацию только в свободные секторы.

9.6. Еще одна рекомендация

Многие пользователи, приобретая новый компьютер, новый жесткий диск или новую версию DOS, восхищаются возможностью форматировать жесткий диск как один большой том DOS. Они так и поступают. И только со временем начинают понимать, что такое решение не только необязательно, но и вредно.

В главе 1 показано, что приходится учитывать нечто большее, чем просто максимальную емкость логического тома, допустимого DOS и BIOS. Часто намного важнее разбить жесткий диск на несколько логических накопителей просто ради собственного удобства.

DOS, конечно, справится с тысячей каталогов и более чем 21,000 файлов, но многие пользователи получают удовольствие от такой структуры каталогов. Даже DOS замедляет свои действия, "продираясь" через все эти подкаталоги к нужному файлу. (Наверное, вы недоверчиво ухмыльнулись, увидев такие числа. Но это реальные числа, которые автор взял для накопителя 1 Гбайт своего основного ПК. Если у вас появился диск большей емкости, чем прежний, вы вскоре удивитесь количеству файлов в накопителе.)

Более того, в главе 10 показано, что при большой емкости логических томов значительная часть диска пропадает. Нужно иметь очень веские доводы, чтобы вводить емкость свыше 128 Мбайт.

Все эти замечания, конечно, не означают, что вы сделаете явную ошибку, превратив жесткий диск в один большой логический том (накопитель C и больше нет накопителей с буквами). Автор просто советует перед этим продумать альтернативы и удобные способы работы на компьютере. Возможно, вы найдете намного лучший способ организации разделов на жестком диске.

Ответ на задачу

Здесь приведено решение поставленной ранее задачи. Накопитель SCSI емкостью 1 Гбайт, чья MBR приведена на рис. 9.2, разбит на три части. Сырая емкость накопителя составляет около 1 Гбайт, а форматированная — 991 Мбайт.

Первая часть — это первичный раздел DOS (значение системного байта равно 4). Как обычно, он начинается с головки 1, цилиндра 0 и сектора 1, а кончается на головке 63, цилиндре 30 и секторе 32. Напомним, что это накопитель SCSI и, как все такие накопители, он должен преобразовывать секторы. Как типичный накопитель SCSI, он выглядит для DOS накопителем, имеющим емкость цилиндра 1 Мбайт (головки от 0 до 63 и секторы от 1 до 32). Количество секторов в этом разделе равно 63,456 и есть 32 сектора до его начала.

Второй раздел в таблице загрузочный. Он невелик и находится около конца накопителя. Значение системного байта равно 0Ah, что для DOS считается недопустимым. Этот раздел содержит Boot Manager системы OS/2. Он начинается с головки 0, цилиндра 990 и сектора 1 и продолжается до головки 63, цилиндра 990 и сектора 32. Общее количество секторов в разделе составляет 2,048 (1 Мбайт), а до его начала имеется 2,027,520 секторов.

Этот дисковый накопитель вначале выполняет программу Boot Manager. После этого можно загрузить DOS или OS/2. Ваше удивление — а где же на диске система OS/2 понятно (вы знаете, что DOS находится в первичном разделе DOS, которым является накопитель C). В данном примере системные файлы OS/2 установлены в логическом накопителе, находящемся внутри расширенного раздела DOS. Этот логический накопитель можно форматировать либо с файловой структурой HPFS системы OS/2 (и в этом случае к нему может обращаться только OS/2), либо со структурой FAT (в этом случае к файлам в этом логическом накопителе могут обращаться DOS и OS/2).

Последняя строка (третья строка таблицы разделов заполнена нулями и не используется) в таблице описывает остальную часть диска. Это расширенный раздел DOS (значение системного байта равно 5). Он начинается с головки 0, цилиндра 31 и сектора 1, а заканчивается на головке 63, цилиндре 989 и секторе 32. Этот раздел содержит 1,964,032 сектора и до его начала имеется 63,488 секторов.

Для контроля можно посмотреть, что сумма байтов в каждом разделе и байтов до него равна количеству байтов до следующего раздела. Конечно, нужно пройти через разделы в порядке их размещения на диске, а не в порядке таблицы разделов. В данном случае порядком будет первая строка, затем последняя строка и, наконец, последняя строка.

Заключение

- На жестких дисках необходимо создавать разделы, так как в пору "молодости" ПК жесткие диски были дорогими и никто не знал, какая операционная система будет доминирующей. Сейчас, даже несмотря на изменчивую ситуацию, приходится создавать разделы, потому что они "глубоко въелись" в конструкцию ПК.
- DOS создает и использует два вида разделов: первичный раздел DOS, который может иметь только один логический том, и расширенный раздел DOS, который может содержать несколько логических томов (и даже некоторые тома системы OS/2). Для поддержки расширенного раздела DOS требуется версия 3.3.
- Все версии DOS до 4.0 (за некоторыми исключениями, например Compaq DOS 3.31) не могли использовать логический накопитель больше 32 Мбайт. Новые версии допускают почти произвольно большие разделы (вплоть до ограничения BIOS в 504 М или 8.875 Гбайт).
- Программа DOS FDISK позволяет посмотреть имеющиеся на диске разделы и логические накопители и с ее помощью их можно изменить. Помните, что про-

грамма FDISK очень опасна, поэтому до ее применения прочитайте все приведенные в главе предупреждения о ней!

- Для DOS 3.3 не требуются программы организации разделов третьих фирм (например, Disk Manager или SpeedStor), настоятельно рекомендуется не применять такие программы.

В этой и предшествующей главах вы изучили первых два этапа подготовки нового накопителя для работы с DOS. В следующей главе вы познакомитесь с заключительным этапом, который называется логическим форматированием, или форматированием высокого уровня.

Глава 10. Подготовка жесткого диска к хранению информации.

Этап 3 — логическое форматирование

В этой главе вы узнаете

- Все о третьем заключительном этапе подготовки жесткого диска для работы с DOS
 - Как DOS организует информацию в логическом томе DOS
 - Что делает программа DOS FORMAT при подготовке тома
 - Чем отличается формат DOS от форматов других операционных систем.
-

10.1. Что такое логическое форматирование?

После форматирования низкого уровня и организации разделов вы готовы к форматированию высокого уровня. Продолжая аналогию со стоянкой автомашин, скажем, что теперь вы разметили места стоянки (контроллер записал секторы на диск) и возвели ограждения (таблицы разделов, созданные программой FDISK). Теперь остается организовать сервисную структуру, чтобы автомашины (файлы) могли парковаться автоматически. Эту последнюю структуру создает программа DOS FORMAT. Дальше в этой главе рассматриваются функции этой программы и таблица размещения файлов (File Allocation Table — FAT).

10.2. Логическая организация раздела DOS

В предыдущих главах показана подготовка диска путем форматирования низкого уровня и разбиения диска на главную загрузочную запись, один или несколько разделов (из которых DOS может видеть только один первичный раздел DOS и один расширенный раздел DOS) и диагностический цилиндр (плюс, возможно, секретный цилиндр). После этого необходимо использовать программу DOS FORMAT для создания нескольких областей внутри раздела DOS и внутри каждого логического диска DOS в расширенном разделе (если вы определили логические накопители). Только после этого диск готов к работе с DOS.

DOS разбивает каждый диск на четыре области. Она же разбивает каждый первичный раздел DOS и каждый логический том на жестком диске на те же четыре области. Этими областями являются загрузочный сектор, таблица размещения файлов, корневой каталог и область данных (рис. 10.1).

На этом высшем уровне DOS считает все дисковое пространство в каждом из накопителей единым длинным списком логических секторов. Только когда DOS посылает команды нижним уровням и, в конце концов, BIOS, такое представление преобразуется в реальные физические адреса.

10.2.1. Загрузочная запись DOS

Загрузочная запись DOS — это первый сектор в логическом томе DOS. Она одинакова во всех томах, отформатированных одной версией DOS. Загрузочная запись DOS занимает точно один сектор и содержит программу, выполняющую важную часть загрузки, поэтому ее часто называют загрузочным сектором (рис. 10.2).

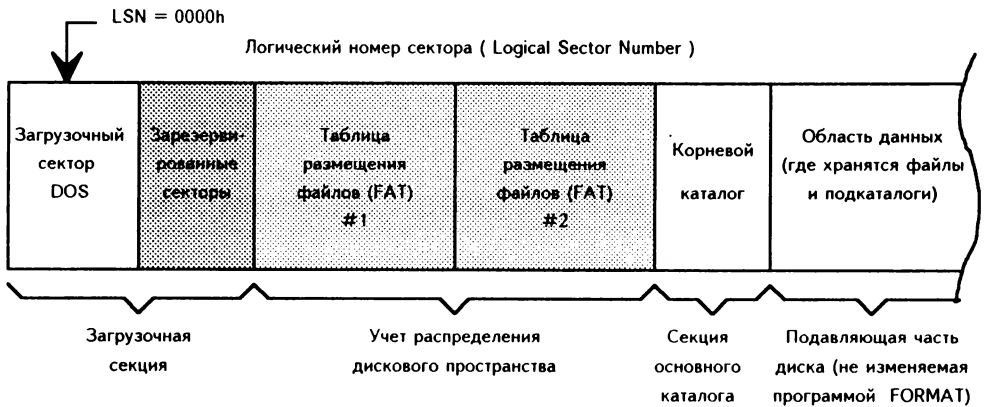
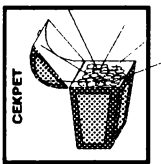


Рис. 10.1. Четыре важных области внутри любого логического накопителя DOS

Загрузочная запись DOS выполняет две функции. Она содержит программу, которая загружает операционную систему, что нужно для компьютера (отсюда и название "загрузочная запись"), и таблицу данных с критической информацией об этом разделе DOS. Далее рассмотрим смысл всех чисел, хранимых в таблице данных загрузочной записи.

Структура всех загрузочных записей, создаваемых одной версией DOS, одинакова. Они содержат данные, описывающие диск, на котором находятся записи. Конечно, эти числа должны отличаться на дисках разной емкости и дисках, способы обращения к которым со стороны DOS и BIOS отличаются. Но, во всем остальном, кроме различия в содержимом таблицы данных, загрузочная запись, например на дискете ПК DOS 3.3, аналогична загрузочной записи в первичном разделе DOS жесткого диска. В этом отношении загрузочная запись DOS одинакова в любом из логических накопителей, созданных в расширенном разделе DOS при условии, что эти логические накопители отформатированы ПК DOS 3.3.



На дискете DOS загрузочная запись имеет такой физический адрес: цилиндр 0, головка 0, сектор 1. На жестких дисках DOS 2 загрузочная запись первого раздела находится по адресу: цилиндр 0, головка 0, сектор 2. (Цилиндр 0, головку 0, сектор 1 занимает главная загрузочная запись, называемая таблицей разделов.) В версиях DOS 3.0, где занят только первый сектор дорожки, содержащий

главную загрузочную запись, загрузочная запись находится в цилиндре 0, головке 1, секторе 1.

Аналогично в расширенном разделе DOS (только в DOS 3.3) каждый логический том начинается с головки 1, сектора 1, цилиндра 1 данной области диска. (Все разделы, включая расширенный раздел DOS, и все логические накопители начинаются на границе цилиндра.) Расширенная таблица разделов находится в головке 0, секторе 1 данного цилиндра, а остальные секторы дорожки пропускаются (головка 0, секторы 2 и т.д.).

Относительный адрес	16-ричные байты данных	Данные в коде ASCII
0000	EB 3C 90 4D 53 44 4F 53 - 35 2E 30 00 02 04 01 00	< .MSDOS5.0 . . .
0010	02 00 02 E0 F7 E8 3E 00 - 20 00 40 00 20 00 00 00	> @ NO NA
0020	06 00 00 00 80 08 25 09 - 17 00 00 4E 4F 20 4E 41	ME . . . x 6.7 V
0030	4D 45 20 20 20 20 46 41 - 54 31 36 20 20 20 FA 33	S > FAT16
0040	C0 8E D0 BC 00 7C 16 07 - BB 78 00 36 C5 37 1E 56	ry 3 9 t >
0050	16 53 8E 3E 7C B9 0B 00 - FC F3 A4 06 1F C6 45 FE	. M . G . >
0060	0F 8B 0E 18 7C 88 4D F9 - 89 47 02 C7 07 3E 7C E8	. . & . R
0070	CD 13 72 79 33 C0 39 06 - 13 7C 74 08 8B 0E 13 7C	
0080	89 0E 20 7C A0 10 7C F7 - 26 16 7C 03 06 1C 7C 13	
0090	16 1E 7C 03 06 0E 7C F3 - D2 00 A3 20 7C B9 16 52	
00A0	7C A3 49 7C 89 16 4B 7C - D8 20 00 F7 26 11 7C 88	
00B0	1E 0B 7C 03 C3 48 F7 F3 - 01 06 49 7C 83 16 4B 7C	
00C0	00 BB 00 05 8B 16 52 7C - A1 50 7C E8 92 00 71 1D	
00D0	B0 01 E8 AC 00 72 16 8B - FB B9 0B 00 BE E6 7D F3	
00E0	E8 5F 00 33 C0 7F 20 B9 0B - 00 F3 A6 74 18 B9 9F 7D	
0100	58 58 58 E8 E8 8B 47 1A - 1F 8F 04 8F 44 02 CD 19	u D .
0110	F7 E3 03 06 49 7C 13 16 - 4B 6B 00 00 7C 32 FF	XXX 3 . G HH 2
0120	50 52 51 E8 3A 00 72 D8 - B0 01 E8 54 00 59 53 58	PRO . r k .
0130	72 BB 05 01 00 83 D2 00 - 03 1E 0B 7C E2 E2 8A 2E	r . \$. . k .
0140	15 7C 8A 16 24 7C 8B 1E - 49 7C 1A 4B 7C EA 00 00	P . \$. . k .
0150	70 00 AC 0A C0 74 29 B4 - 0E BB 07 00 CD 10 EB F2	3 . s . 6 % . M . 0
0160	3B D6 16 7C 73 19 F7 36 - 18 7C FE C2 88 16 4F 7C	\$. M . 6 % . 60
0170	C3 D2 F3 36 1A 7C 88 16 - 06 D2 E8 0A 36 4F 7C 8B	Non . . . % . disk
0180	CA 86 E9 8A 16 24 7C 8A - 36 25 7C CD 13 C3 0D 0A	or disk error disk
0190	4E 6F 6E 2D 53 79 73 74 - 65 6D 20 64 69 73 6B 20	replace any key when re
01A0	6F 72 2D 64 69 73 6B 20 - 65 72 72 6F 72 0D 0A 52	adv IO press
01B0	6F 70 6C 61 23 65 20 61 - 6E 64 20 70 72 65 73 73	SMSDOS SYS . . U .
01C0	61 64 79 0D 0A 00 49 4F - 20 20 20 20 20 20 55 59	
01D0	53 4D 53 44 4F 53 20 20 - 20 53 59 53 00 80 55 AA	

Рис. 10.2. Загрузочная запись DOS. Этот пример взят из логического накопителя C (первичный раздел DOS) жесткого диска, его главная загрузочная запись показана на рис. 9.2.

Загрузочная запись DOS начинается командой перехода. После нее находится таблица данных, большая часть значений которой копируется в область ЗУПВ, называемую блоком параметров BIOS для этого накопителя. Наконец, есть фактическая программа загрузки, которая привлекает числа из таблицы для поиска системных файлов в разделе, если компьютер загружается из этого раздела. Конечно, загрузочным может быть только первичный раздел DOS (который является накопителем C). Следовательно, ни в одном из других разделов или логических томов программа загрузки не нужна. Но все разделы DOS и логические тома ради простоты имеют одинаковую загрузочную запись DOS.

Каждая загрузочная запись DOS заканчивается двумя байтами 55h AAh. Они называются сигнатурой и применяются в нескольких структурах данных DOS. Например, главная загрузочная запись также заканчивается такой же сигнатурой.

10.2.2. Область данных

Несмотря на то, что область данных не является следующей частью раздела, нужно рассмотреть ее следующей. После изучения способа хранения данных проще разобраться с тем, зачем нужны две другие секции и как они организованы.

На одном из своих уровней DOS следит за файлом. На уровне, который немного ниже этого, она следит либо за символом (когда файл поступает из символического устройства или передается в него), либо за сектором (когда он поступает от блочного устройства, например дискового накопителя, или передается в него).

Можно подумать, что, поскольку DOS следит за сектором данных при работе с дисковым накопителем, она использует эту же единицу и для слежения за ним при появлении. Однако это не так. Вместо сектора DOS использует другую единицу, называемую кластером или единицей распределения.

Кластеры. Это небольшая группа секторов. Кластер может состоять из любого числа секторов (включая один), равного степени двух. Количество секторов в кластере зависит от емкости дискового накопителя и версии DOS. Почему DOS использует кластеры и как она определяет размер кластера?

Каждый участок диска, на котором можно сохранить данные, находится в одном из трех состояний: свободен, занят файлом или подкаталогом, поврежден и не годится для хранения данных. DOS должна следить за состоянием каждого сектора в области данных каждого накопителя DOS. Далее, DOS должна следить за тем, какие файлы "владеют" каждым используемым участком диска. Если файл не помещается в один участок, его приходится хранить частями. Следовательно, DOS должна "уметь" связывать нужные участки в файл, когда программа запрашивает данные.

Система постоянно регистрирует эту информацию на диске с файлами, расположение которых она описывает. Для ускорения доступа к файлам она хранит большую часть информации в ЗУПВ для диска, к которому сейчас обращается DOS.

Однако хранение этой информации в ЗУПВ "съедает" значительную область первых 640 Кбайт системного ЗУПВ, поэтому разработчики DOS пошли на компромисс и разбили информацию о дисковых файлах на две части — информацию о каталоге файлов, которая дает о файлах все, но только "намекает" на то, где они расположены, и информацию о распределении дискового пространства, которая сообщает все о состоянии каждого участка области данных диска.

В аналогии со стоянкой автомашин информация о каталоге файлов эквивалентна списку, где находится каждая машина и приведен ее внешний вид. Информация о распределении дискового пространства эквивалентна списку мест стоянок — какие заняты, свободны или повреждены, а для групп автомашин — какие машины входят в группу.

На диске часть информации о каталоге хранится в корневом каталоге, а остальная — в подкаталогах. Информация о распределении дискового пространства хранится в таблице размещения файлов (FAT).

При обращении к диску для получения одного файла DOS должна хранить в ЗУПВ информацию о каталоге только для одного файла или, максимум, о нескольких файлах в одном подкаталоге. Чтобы знать, где на диске найти файл и куда она может поместить дополнительную информацию для этого или нового файла, DOS должна иметь в ЗУПВ информацию о распределении дискового пространства для большей части или всего диска.

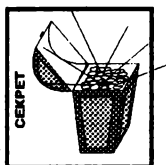
Если вы получаете информацию или записываете ее только в один файл, DOS выполняет операции очень быстро. Обращение к новому файлу длится несколько дольше, так как DOS должна считать с диска информацию о каталоге для этого файла.

Упрощая информацию о распределении дискового пространства и отделяя ее от подробностей о файлах, занимающих это пространство, разработчики DOS минимизировали объем ЗУПВ для DOS при обработке файлов или их создании. Когда DOS

требуется узнать имя файла, размер, время создания или другие факты, она ищет эту информацию в другом месте.

Но даже сама информация о распределении пространства требует хранения в ЗУПВ многих данных. Если бы DOS следила за состоянием каждого отдельного сектора жесткого диска очень большой емкости, все ЗУПВ пришлось бы отвести только для этой информации, а места для приложений не осталось бы. Именно здесь на помощь приходят кластеры.

Кластер — это некоторое удобное число секторов с последовательными номерами. Разбивая дисковое пространство на эти большие единицы, DOS должна следить за меньшим числом единиц. Сокращая максимальный номер кластера, DOS также экономит пространство (подробности в разделе 10.2.3).



Версии DOS 1 и 2 разбивали область данных любого диска не более чем на 4,000 единиц хранения, или кластеров. Кластеры нумеруются, начиная с 2. Только здесь DOS нумерует элементы, начиная не с 0, не с 1, а именно с 2. Фактически в FAT есть элементы или по крайней мере места для этих элементов для кластеров 0 и 1. Они используются для специального сигнатурного байта, называемого байтом дескриптора носителя, и заполняющих байт. Если посмотреть на любой жесткий диск, то окажется, что первый байт в каждой FAT содержит F8h, а следующих два-три байта (в зависимости от 12- или 16-битовой FAT) содержат FFh. Значение F8h является байтом дескриптора носителя для жесткого диска. Для гибких дисков разной емкости применяются другие значения. Байты FFh служат заполнителями.

Начиная с DOS 3, число возможных кластеров стало больше 65,000. В версии DOS 4.0 максимальный номер кластера такой же, но кластеры теперь называются единицами распределения.

Необходимость следить за 4,000 или даже 65,000 кластеров значительно упрощает задачу DOS по сравнению со слежением за каждым сектором. Напомним, что до версии DOS 4 на одном дисковом накопителе разрешалось иметь 65,536 секторов — отсюда возникло известное логическое ограничение в 32 Мбайт. Начиная с версии DOS 4, максимальный номер увеличен по одному правилу до 4 млрд (это новое логическое ограничение DOS), а по другому правилу — до 1 млн. (это ограничение INT 13h).

Пропадающее пространство. DOS значительно экономит ценное пространство ЗУПВ, следя за распределением дискового пространства файлам группами секторов, называемыми кластерами. Но при этом появляется один недостаток. Не распределяя дисковое пространство байтами, DOS должна расходовать часть его впустую. Каждому файлу, независимо от его размера, приходится выделять участок диска, представляющий целое число кластеров.

Пространство в конце последнего кластера, отведенного файлу, которое не заполнено данными файла, называется пропавшим (пропадающим) пространством. У некоторых файлов такое пространство довольно велико, например однобайтный файл впустую расходует весь выделенный ему кластер, кроме одного байта. У других файлов пропавшее пространство меньше или его нет совсем, т.е. файл полностью занимает данными все выделенные ему кластеры.

В среднем, когда размеры файлов на диске распределены более или менее случайно, на каждый файл пропадает примерно половина кластера. Если, например, на диске размер кластера составляет четыре сектора (2,048 байт), то теряется столько килобайт, сколько файлов находится на диске.

Конечно, такая величина кажется большой, но она экономит для DOS ЗУПВ и время. Предположим, что средний размер файлов на жестком диске 50 Кбайт, тогда для тысячи файлов теряется 1 Мбайт, а занимают они 50 Мбайт. Другими словами, пропадает около 2% дискового пространства.

Пропавшее пространство может быть и большим, и очень большим. Если ваш большой диск организован как один том или если вы создали логические тома с неподходящей емкостью, размер кластера будет намного больше 2 Кбайт. В этом случае пропавшее пространство также намного увеличивается.

Дальше показано, как DOS выбирает размер кластера и как повлиять на этот выбор. Эти действия определяют размер пропадающего дискового пространства.

10.2.3. Таблица размещения файлов

DOS помещает в таблицу размещения файлов (FAT) информацию о состоянии всех кластеров в разделе DOS или логическом накопителе DOS. FAT — это просто таблицу чисел — по одному числу на кластер, которые сообщают состояние этих кластеров. Одно специальное значение будет у кластеров, доступных для хранения файлов, а другое значение показывает поврежденный кластер, который система не должна использовать. Еще одно значение у файла, который полностью помещается внутри кластера, или файла, который занимает несколько кластеров и заканчивается в этом кластере. Несколько значений зарезервировано (а вдруг появятся новые блестящие идеи о дополнительных возможностях FAT!). Любое другое допустимое число указывает на кластер, в котором продолжается файл, содержащийся в данном кластере (рис. 10.3 и 10.4).

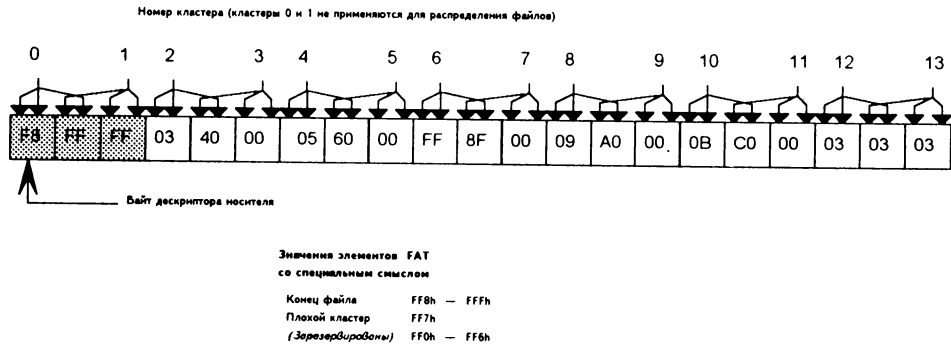
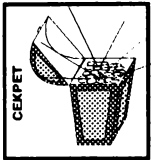


Рис. 10.3. 12-битовая таблица размещения файлов (FAT)

FAT содержит цепи кластеров, показывающие, куда система поместила каждый файл. Она также показывает, какое пространство доступно DOS для записи новой информации. Наконец, FAT фиксирует поврежденные участки на диске, запрещая системе сохранять там данные.

Наверное, вы видели, что команда DOS CHKDSK выводит сообщение об обнаружении нескольких потерянных кластеров в нескольких цепях. О том, как цепь кластеров может оказаться потерянной, сказано далее, в разделе 10.2.8.



Несколько дополнительных чисел в начале первого сектора FAT отмечают начало FAT и показывают тип диска. Первый из этих байтов называется байтом дескриптора носителя. Другие два или три специальных байта заполняются единицами (и имеют 16-ричное значение FFh).

Байт дескриптора носителя содержится также в таблице данных загрузочной записи DOS. К сожалению, этот байт не дает много полезной информации. Но если он не содержит F8h, то диск не является жестким диском.

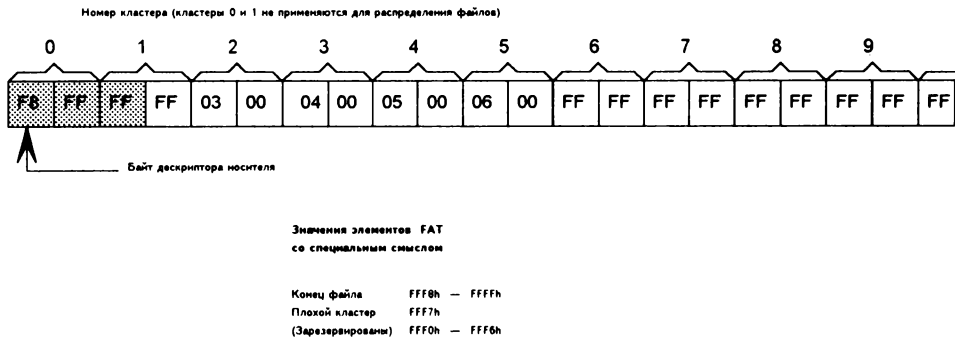


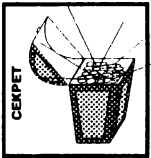
Рис. 10.4. 16-битовая таблица размещения файлов (FAT)

Два типа FAT. Есть два типа таблиц FAT — 12- и 16-битовые. До версии DOS 3 существовали только 12-битовые FAT, а в последующих версиях применяются оба типа таблиц. Различие относится к длине чисел, хранимых в FAT для каждого кластера. Обычно вы не нужно знать, какие элементы FAT у вашего диска.



Есть одно важное исключение из приведенного выше утверждения. Вам нужно знать размер элементов FAT при переходе от 12-битовой FAT к новой версии DOS, которая использует 16-битовую FAT. В этом случае вам придется до модернизации зарезервировать данные, иначе обратиться к ним невозможно.

Если для просмотра FAT жесткого диска вы пользуетесь утилитой, подготовьтесь к пониманию увиденного. Некоторые утилиты, например Disk Editor из утилит Нортона (версия 5+), позволяют посмотреть выводимую информацию в нескольких форматах.



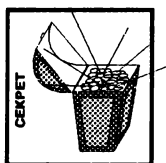
Утилита Disk Editor может показать FAT в терминах, которые легко понять, прочитав описание в данном разделе. Она также позволяет посмотреть, что фактически записано на диске. Этот формат расшифровать несколько сложнее.

Если логический том, чью FAT вы смотрите, имеет такую емкость, которая заставила программу DOS FORMAT создать 16-битовую FAT, задача оказывается несложной. Нужно просто знать, что элемент каждого кластера занимает два байта и байты расположены на диске в "порядке Intel", т.е. младший байт первый. (Есть два основных стандарта на хранение многобайтовых

чисел в последовательных ячейках памяти. "Порядок Intel" означает, что младший байт помещается по нижнему (меньшему) адресу памяти, а "порядок Motorola" помещает туда старший байт. В семействе микропроцессоров 80x86, которое применяется в ПК, принят первый порядок, а в микропроцессорах семейства 68x00 фирмы Motorola, которые применяются в компьютерах Macintosh, принят второй порядок.)

Если посмотреть 12-битовую FAT, картина более запутана. Внимательно посмотрите на рис. 10.3. Вы видите, что три тетрады (полубайта), образующие элемент для кластера, не расположены на диске последовательно. Слава Богу (и Питеру Норпону), что Disk Editor знает о всех этих тонкостях и выводит любую FAT (12- или 16-битовую) в понятном формате.

Приятно узнать, что создает FAT и управляет ими сама DOS. Вам почти никогда не придется задумываться о FAT. Это просто внутреннее "приспособление" DOS, помогающее ей работать, не беспокоя понапрасну пользователей.



Во всех версиях DOS до 3.0 в FAT применялись исключительно 12-битовые числа, что ограничивало число кластеров не более 4,000. Возведение 2 в степень 12 дает 4096. Вычитание специальных значений, которые нельзя использовать для указания других кластеров, дает максимум 4078 кластеров. Для гибких дисков во всех версиях DOS применяются 12-битовые элементы FAT.

Когда DOS 3 проводит форматирование высокого уровня, она анализирует емкость диска (общее число секторов в логическом томе). После этого она решает, нельзя ли оставить 4000 кластеров, не увеличивая размер кластера до слишком большого. Если это возможно, DOS образует FAT с 12-битовыми числами, а если нельзя, применяет 16-битовую FAT. Большой размер элементов FAT означает, что можно иметь 65518 кластеров.

В предыдущем абзаце вы, должно быть, заметили довольно расплывчатый термин "слишком большой". Точное правило выбора размера кластера и размера элементов FAT не документировано. Более того, DOS не позволяет вам сообщить о своем предпочтении. Программа FORMAT делает то, что встроено в нее, а вам остается смириться с ее выбором. Вы можете немного повлиять на выбор, тщательно определяя емкости разделов. Любой раздел менее 16 Мбайт получает 16-битовую FAT и минимальный размер кластера. У меньших разделов, по-видимому, будет 16-битовая FAT; размер кластера будет по выбору DOS не "слишком большим", но не по вашему выбору.

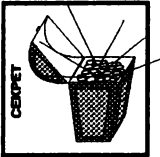
Отметим, что для разделов более 32 Мбайт кластеры обязательно будут больше одного сектора. Если раздел очень большой, кластеры также становятся большими. Например, для раздела 260 Мбайт размер кластера составит 8 Кбайт. В табл. 10.1 приведены взаимосвязи между емкостью тома, размером кластера, размером элемента FAT и версией DOS.

Таблица 10.1. Размеры кластеров для разных дисков или логических накопителей и версий DOS

Емкость тома (Мбайт)	Размер кластеров (Кбайт)	Размер элемента FAT (биты)	Версия DOS
До 32	4	12	2.x
До 8	2	12	3.0+
8-15	4	12	3.0+
16-32	2	16	3.x
16-127	2	16	4.0+

128–255	4	16	4.0+
256–511	8	16	4.0+
512–1023	16	16	4.0+

Какая у вас FAT? Независимо от типа FAT, она занимает только малую часть описываемого диска. Поскольку FAT важна для правильного доступа к файлам на диске и имеет небольшой размер, DOS для большей надежности поддерживает на многих дисках две копии FAT.



Для 12-битовой FAT в каждые три байта DOS помещает два числа о состоянии кластеров, так как три байта составляют 24 бита. Следовательно, в DOS 1 или 2 размер FAT не может быть больше 6 Кбайт (4000 элементов по 1,5 байта каждый). В DOS 3.0 максимальный размер FAT больше, так как имеется больше кластеров (и больше элементов в FAT) и каждый элемент FAT занимает два байта. Для новых версий DOS максимальный размер FAT составляет 128 Кбайт.

Легко разрешить DOS хранить FAT размером 6 Кбайт в ЗУПВ, но хранить там все 128 К очень накладно. Поэтому в DOS 3.0 принято обоснованное решение, какую часть FAT хранить в ЗУПВ, а какую — считывать с диска при необходимости.

Когда DOS распределяет файлу новый кластер или отмечает, что кластер освобожден (так как файл удален), она модифицирует обе копии FAT. Если DOS получает ошибку при считывании первой FAT, она пытается считать вторую FAT. В случае успеха DOS пользуется полученной информацией, но не сообщает об использовании второй FAT и не пытается исправить поврежденную первую FAT.

Программа DOS CHKDSK позволяет обнаружить расхождения между двумя копиями FAT. Кроме этих двух применений, DOS игнорирует вторую FAT, а это диктует регулярно выполнять CHKDSK для выявления потенциальных ошибок.

DOS никогда не исправляет обнаруженные в FAT ошибки. Как максимум, она сообщает вам об этом, а затем оставляет все как есть. Необходимо исправить с помощью специальной утилиты прежде, чем DOS прекратит работать. Программа CHKDSK исправляет некоторые ошибки в FAT. К сожалению, ее способ исправления ошибок далек от совершенства.

10.2.4. Корневой каталог

Корневой каталог, т.е. упорядоченная таблица сведений о файлах, является последней областью раздела DOS или логического накопителя, который создает программа FORMAT. Именно здесь DOS начинает сохранять информацию о каталоге файлов, связывая имя файла с его расположением на диске.

Каждый элемент каталога состоит из 32 байт. Первые 11 байт содержат имя файла (8 байт) и расширение (3 байта). В следующем байте хранятся атрибуты файла. Десять байтов зарезервированы, т.е. фирма Microsoft не рекомендует использовать их. Обычно они заполняются нулями. После этого расположены время и дата, когда программа открыла файл с разрешением на изменение его содержимого (2 байта для времени и 2 байта для даты). Следующим расположен номер кластера, с которого DOS начинает сохранение содержимого файла (2 байта). Последние 4 байта отведены под размер файла в байтах (рис. 10.5).

На рис. 10.5 каждый байт представлен прямоугольником, а каждый элемент каталога — одной строкой прямоугольников. Внутри каждого прямоугольника снизу показано 16-ричное число. Когда оно является символом кода ASCII, выше числа приводится этот символ.

Байт атрибутов DOS интерпретирует по битам. Рис. 10.5 поясняет смысл каждого из шести используемых DOS атрибутов. Некоторые сетевые операционные системы используют один или оба из оставшихся битов.

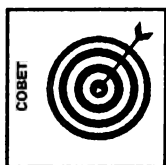
Дата и время последнего изменения файла хранятся в двухбайтовых числах. На рис. 10.5 показан формат этих полей. Старший бит (MSB) находится справа, а младший (LSB) — слева (для байтов атрибутов, времени и даты).

Отметим, что элемент каталога показывает только, где начинается файл, а не где находится весь файл. Для поиска остальной части файла DOS обращается к FAT, отправляясь от кластера, где начинается файл (и где он кончается, если файл небольшой), и прослеживая цепь кластеров до достижения конца файла.

Корневой каталог имеет фиксированный размер, зависящий от версии DOS и емкости и типа дискового накопителя, обычно для жесткого диска это 512 элементов.

10.2.5. Подкаталоги

При наличии более 512 файлов (или такого их числа, которое помещается в корневой каталог) необходимо хранить "каталогизирующую" информацию в подкаталоге, идея которого реализована в DOS 2.0. Подкаталог — это просто специальный файл, т.е. хотя он и хранится как файл, но его содержимое интерпретируется как расширение корневого каталога.



Целесообразность хранения всего, что можно, вне корневого каталога объясняется тем, что при попытке вместить в корневой каталог больше файлов, чем в него помещается, DOS сообщает о том, что это сделать нельзя. Однако ее сообщение слегка запутанное — на диске нет больше места (а не отсутствует место в корневом каталоге). Помните, что среди файлов в корневом каталоге могут быть скрытые системные файлы (если это загрузочный диск) и метка тома (если назначено имя этому дисковому тому). Придерживайтесь такого правила: если что-то не должно быть в корневом каталоге, это что-то нужно оттуда убрать.

Подкаталоги первого уровня показаны как файлы в корневом каталоге. Они содержат листинги других файлов и, возможно, других подкаталогов второго уровня и т.д. На рис. 10.6 показаны два представления структуры каталога и подкаталогов гипотетического жесткого диска.

В верхней части рис. 10.6 показано то, что можно увидеть при выполнении команды DOS TREE (DOS 5); внизу — несколько причудливое представление каталогов как комнат в хранилище данных. Символ наклонной черты / для DOS является именем корневого каталога.

Глубина дерева каталогов может быть практически любой. Ограничение состоит в том, что длина маршрута (или пути) от корневого каталога до любого файла не должна превышать 64 символов с учетом символов /, отделяющих имена каталогов, и последнего символа / в конце маршрута.

10.2.6. Метка тома

Еще одним специальным "файлом", который можно найти в корневом каталоге, является метка тома, используемая для хранения имени, ассоциированного с данным дисковым разделом или логическим томом. Система хранит метку как файл, но в нее включен специальный атрибут тома, а размер и номер кластера равны нулю.

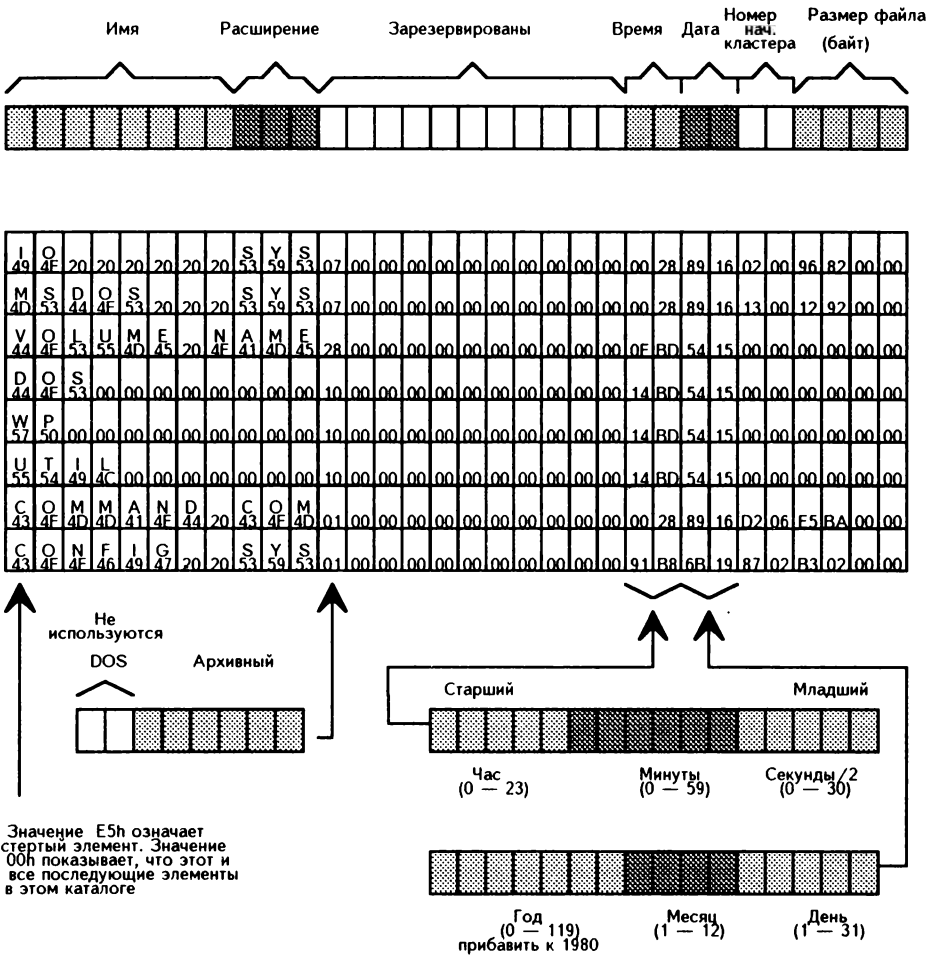


Рис. 10.5. Структура (корневого) каталога DOS



Именовывать разделы или тома не обязательно, но рекомендуется пользоваться метками. Начиная с версии DOS 3.2, программа FORMAT имеет очень ценную возможность: если запросить реформатирование жесткого диска, имеющего метку тома, программа откажется делать это до сообщения ей метки тома. Это свойство помогает предотвратить случайное уничтожение данных в накопителе С. (Программа помогает, но не запрещает уничтожение данных. Пользуйтесь программой FORMAT очень осторожно и не позволяйте этой простой предосторожности успокоить вас и не делать хороших резервных копий.и)

10.2.7. Вывод по структурным частям

Теперь вы знаете все элементы, делающие диск пригодным для DOS. По аналогии со стоянкой автомашин линии на мостовой похожи на форматирование низкого уровня. Ограждения напоминают разделы. Сервисная структура аналогична FAT и каталогам. Метка дискового тома соответствует знаку при въезде.

10.2.8. Как кластеры "теряются"

DOS модифицирует информацию в FAT и каталоге независимо друг от друга. Обычно DOS обновляет FAT при каждом добавлении новой информации в дисковый файл, а каталог — по окончании работы с файлом.

Если иногда программа прерывается при записи файла на диск, элементы FAT показывают, где на диске найти данные файла, но пока нет элемента каталога для файла, показывающего, где начинается файл. Такие каталоги, отмеченные в FAT для этого файла, принято называть потерянной цепью.

Самый распространенный способ образования потерянных цепей связан с ситуацией, когда программа "запуталась" и единственным средством прервать ее кажется перезагрузка компьютера. При этом компьютер прекращает свои действия, но программа "не приводит себя в порядок" и неизбежным следствием оказываются потерянные цепи. Это одна из причин того, что не рекомендуется прекращать программы выключением или перезагрузкой компьютера.

Проще всего обнаружить потерянные цепи, выполнив программу DOS CHKDSK без всяких ключей командной строки. Напечатайте CHKDSK C:, но не печатайте, хотя бы в первый раз, CHKDSK C: /F.

Если программа CHKDSK сообщает только о потерянных кластерах, ее целесообразно повторить для этого же диска, определив теперь опцию /F. При появлении вопроса о превращении потерянных цепей в файлы отвечайте Yes (Да). По окончании CHKDSK в корневом каталоге появятся один или несколько файлов с именами FILE0000.CHK, FILE0001.CHK и т.д. Посмотрите содержимое этих файлов.



Превращая потерянную цепь в файл, не считайте, что вы получите полный и точный файл. В частности, после восстановления программы, не выполняйте этой программы. "Обрывки" могут оказаться сильно поврежденными. Замените программу из резерва или с диспетчера электронного диска. Когда восстановлен файл электронного бланка, текстового процессора или базы данных, его можно рассмотреть подробно, пользуясь создавшим файл приложением. Часто обнаруживается значительная часть потерянных данных, но некоторые секторы информации отсутствуют, а в других находятся совершенно другие данные, взятые из какого-то другого (стертого или потерянного) файла.

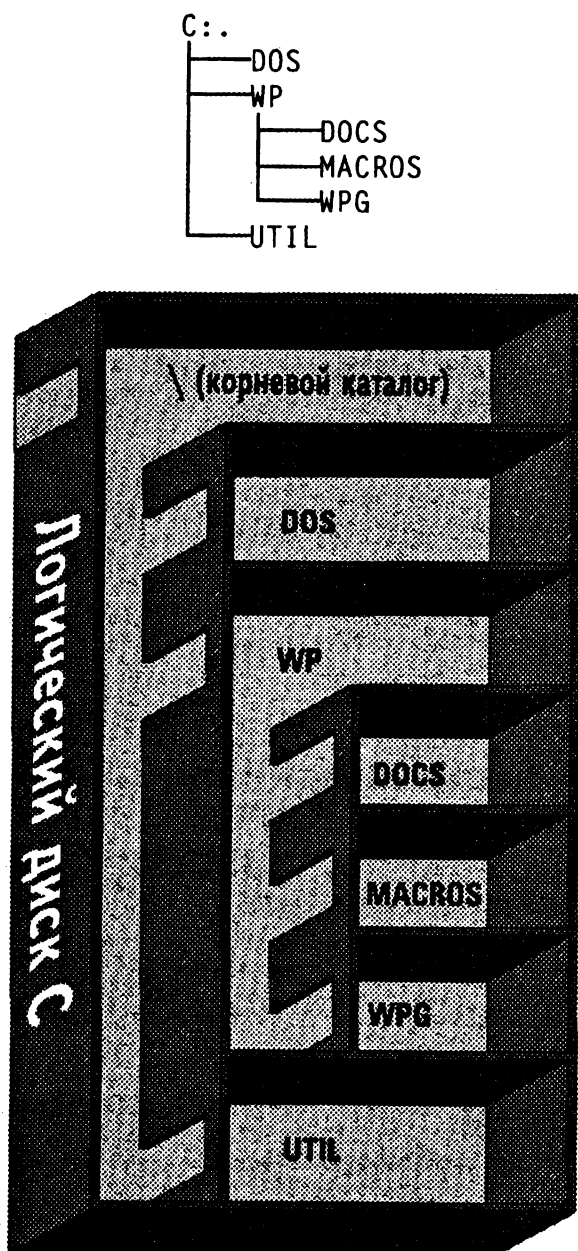


Рис. 10.6. Два представления дерева каталогов

10.2.9. Другие способы повреждения данных

Если файл, в который записываются данные, уже существует, то элемент каталога также существует. Однако длина файла в элементе каталог не корректируется до окончания операций DOS с файлом (когда добавляющая в файл прикладная программа "закрывает" файл). Если что-то прерывает программу до закрытия файла, в структуре каталог — FAT возникает ошибка еще одного вида. Программа CHKDSK замечает расхождение и сообщает о неправильной длине файла. К сожалению, она исправляет эту ошибку не так, как бы вам хотелось. (Она просто "обрубает" лишнюю длину, т.е. самые последние сохраненные данные.)

Для предотвращения этой "любезности" программы CHKDSK старайтесь не выполнять ее с опцией /F до просмотра ошибок на диске. Если программа находит файлы с неправильной длиной, можно воспользоваться другой утилитой, чтобы устранить ошибки. Для таких операций наиболее удобны пакеты утилит Нортон и PC Tools (фирмы Central Point Software).

Последний вид ошибок, которые обнаруживает программа CHKDSK, — это файлы, связанные по некоторому кластеру. Такое сообщение означает, что цепи кластеров двух файлов содержат общее число, т.е. оба файла претендуют на часть их данных, хранящихся в одном и том же месте.



Причина появления связанных файлов не вполне очевидна. Ясно только, что DOS или программа, которая обходит DOS, запуталась и поместила в FAT неправильную информацию. Лучше всего стереть оба файла и восстановить их с резервных копий. Но перед этим целесообразно узнать "вторую точку зрения" некоторой другой утилиты. Запутавшейся программой может оказаться сама CHKDSK. Целесообразно осуществить перезагрузку компьютера, чтобы новая утилита получила с диска свежую копию FAT.

10.3. Другие операции программы FORMAT

Программа DOS FORMAT выполняет больше операций, чем просто распределение четырех областей логического тома DOS. Она также должна инициализировать значения в трех из этих областей и выполнить хотя бы беглый контроль четвертой области. Наконец, и к сожалению, она может выполнить еще одну операцию (в зависимости от марки и версии DOS), которая может отнять у вас целый день! Внимательно прочитайте следующий материал, чтобы случайно не сделать ошибки.

10.3.1. Инициализация диска

В загрузочный сектор помещаются копия стандартной программы загрузки DOS и таблица данных для этого раздела или логического тома. В первый байт FAT записывается дескриптор носителя; а следующих два или три байта заполняются единицами. Остальные элементы FAT заполняются нулями.

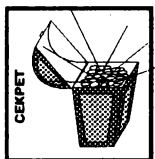
10.3.2. Простой тест диска

После этих операций программа FORMAT считывает все пространство данных. Она отмечает в FAT все кластеры, в которых обнаруживаются один или несколько плохих секторов (их не может считать сама программа FORMAT или они отмечены как плохие программой форматирования низкого уровня). На загрузочный диск копируются системные файлы и для них заполняются элементы корневого каталога и FAT.

Большинство версий DOS при выполнении программы FORMAT ничего не записывает в область данных (кроме системных файлов для загрузочного диска). В главе 11 приведены те версии DOS, в которых программа FORMAT проводит запись в область данных.

10.3.3. Интерпретация таблицы данных загрузочной записи DOS

Только теперь вы сможете разобраться во всех числах, содержащихся в таблице данных загрузочной записи DOS. От 13 до 50 таких чисел (в зависимости от версии DOS) копируются в блок параметров BIOS для накопителя. Все они доступны программам DOS через вызовы прерываний (рис. 10.7).



Область данных в загрузочном секторе на рис. 10.7 содержит следующую информацию.

- ✓ Первые восемь байтов таблицы данных (начиная с четвертого байта в секторе) содержат название производителя комплектного оборудования (OEM) и номер версии DOS для программы FORMAT, создавшей этот загрузочный сектор. Это поле не входит в блок параметров BIOS.
- ✓ Следующее поле (два байта) содержит число байтов в секторе (максимальное значение 65,536). Жесткие диски всех ПК используют 512 байт в физическом секторе; другой размер логического сектора обычно кратен 512 байт.
- ✓ Третье однобайтовое поле показывает число секторов в кластере. Произведение этого и числа и предыдущего дает число байтов в кластере.
- ✓ Четвертое поле (два байта) определяет число зарезервированных секторов, а фактически равно числу секторов в этом разделе до начала первой копии FAT. Поскольку логические секторы начинаются с нуля, число зарезервированных секторов равно первому сектору в первой FAT. Обычно оно равно единице, показывая, что перед FAT находится только загрузочный сектор, хотя разработчики DOS допускают формат с большим числом зарезервированных секторов между загрузочным сектором и первым сектором FAT.
- ✓ Следующих два поля (один и два байта соответственно) показывают число FAT и число элементов в корневом каталоге.
- ✓ Поле общего числа секторов в томе применяется только для томов DOS, емкость которых меньше 32 Мбайт. Это двухбайтное число не может представить число секторов в большем томе (предполагая размер логического сектора 512 байт). Для логических томов DOS, емкость которых больше 32 Мбайт, в этом поле содержится нуль. Фактическая емкость таких "огромных" разделов помещается в другое поле, имеющееся только в загрузочных записях, создаваемых DOS 4.0+.
- ✓ В следующем поле находится байт дескриптора носителя; для жесткого диска он равен F8h.
- ✓ Следующее поле показывает число секторов в FAT.
- ✓ Таблица данных в DOS 2 заканчивается еще одним полем, содержащим число секторов на дорожке. В DOS 2.x — это однобайтное число, поэтому значение в этом поле не может превышать 255.
- ✓ DOS 3 расширила поле количества секторов на дорожке и добавила два новых поля — количество головок и количество скрытых секторов. Количество головок равно количеству поверхностей дисков, на которых записываются данные. Количество скрытых секторов равно общему количеству физических секторов в нако-

пителе до начала данного раздела. (Отметим различие между зарезервированными секторами и скрытыми. Первое показывает, сколько секторов внутри этого раздела нужно пропустить для получения начала первой FAT. Второе число показывает, сколько секторов в накопителе нужно пропустить для получения начала этого раздела.)

- ✓ В DOS 4 поле количество скрытых секторов увеличено с двух до четырех байтов, что допускает наличие гигантских дисков! В версии 4 добавлено также четырехбайтное поле общего количества секторов для логических томов DOS, емкость которых больше 32 Мбайт.
- ✓ В DOS 5 добавлено еще несколько полей: физический адрес BIOS для диска, содержащего этот раздел или логический накопитель, расширенная сигнатура загрузочной записи, порядковый номер тома, имя тома и цепочка идентификации файловой системы.

Физический адрес BIOS для диска вносит некоторую путаницу. Обычное значение для первого физического жесткого диска равно 80h. Второй жесткий диск имеет физический адрес 81h. Начиная с DOS 5.0, можно иметь до семи физических жестких дисков с адресами до 86h.

DOS называет том внутри первичного раздела DOS первым физическим накопителем C. Если имеется первичный раздел на втором физическом накопителе, он называется D. Этот процесс продолжается по всем физическим дискам, если только они имеют первичные разделы DOS.

Логические тома внутри расширенного раздела DOS на первом накопителе получают следующие буквы накопителей. (При наличии двух физических накопителей, имеющих первичные разделы DOS, буквы накопителей начинаются с E.) Затем буквы накопителей назначаются логическим томам внутри расширенного раздела DOS на следующем физическом накопителе. И вновь этот процесс продолжается до назначения букв логическим накопителям (или до достижения Z).

Обычно загрузочные секторы в логических томах внутри расширенного раздела имеют такой же физический адрес BIOS, как и первичный раздел DOS на том же самом накопителе.

10.3.4. Отличие форматирования гибких дисков

Обычно программа DOS FORMAT выполняет для гибких дисков больше операций, чем для жестких. В частности, она должна делать больше при первом форматировании дискеты. На жестком диске до выполнения программы FORMAT необходимо провести форматирование низкого уровня и организовать разделы. На гибких дисках все операции выполняет программа FORMAT. (Другими словами, она осуществляет форматирование низкого и высокого уровня. На гибких дисках разделов нет, поэтому создавать таблицу разделов не нужно.)

Начиная с версии DOS 5, пользоваться командой FORMAT стало несколько безопаснее. Режимом по умолчанию, если форматруемый диск уже отформатирован, является безопасное форматирование: FORMAT пропускает этап форматирования низкого уровня и до инициализации FAT и корневого каталога сохраняет их содержимое на диске. Следовательно, можно восстановить файлы на диске, если в секторы восстанавливаемого файла информация не сохранялась. После перезаписи данных восстановление файла невозможно.

Эти улучшения безопасности FORMAT применимы теперь к гибким дискам так же, как к жестким дискам. При форматировании ранее отформатированного гибкого диска в DOS 5.0+ программа FORMAT делает то же, что и для жесткого диска.

10.4. Альтернативы форматированию высокого уровня DOS

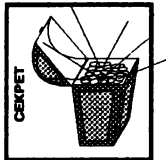
В главах 8 и 9 рассмотрены форматирование низкого уровня и организация разделов для дисков, используемых DOS и другими операционными системами. Формат низкого уровня наносит контроллер, поэтому операционная система изменить его не может. BIOS считывает таблицу разделов, поэтому она должна быть одинаковой, независимо от установленной на диске операционной системы. Формат высокого уровня проводит внутри каждого раздела операционная система, использующая этот раздел, поэтому его форма варьируется в различных операционных системах.

Для одного ПК сейчас популярны операционные системы UNIX и OS/2. Во многих локальных сетях применяется система NetWare фирмы Novell, которая работает как специальная операционная система в файловом сервере. Фирма Novell предоставляет операционную систему только файловому серверу, а в ПК рабочих станций применяется DOS или OS/2. Фирма Microsoft объявила новую операционную систему Windows NT, которая будет конкурировать с системами UNIX, OS/2 и NetWare.

Все эти операционные системы "понимают" и применяют такой же способ организации разделов, как и DOS, поэтому они могут разделять жесткий диск с DOS (см. главу 9). Но любая из них может быть единственной операционной системой на вашем ПК.

10.4.1. Система UNIX

Система UNIX осуществляет форматирование высокого уровня совсем не так, как DOS, поэтому раздел UNIX совершенно "непонятен" DOS. Это правда, несмотря на то, что разработчики DOS многое позаимствовали из UNIX. При внимательном рассмотрении окажется, что UNIX обращается с жестким диском так же, как DOS, но все же эти системы не могут считывать разделы друг друга.



Приведем несколько отличий UNIX (и Xenix) от DOS.

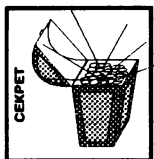
- ✓ UNIX определяет области на диске в терминах блоков и групп. Эти области аналогичны секторам и кластерам, но UNIX контролирует их на ошибки ЕСС в каждой операции считывания или записи. При появлении ошибки данные автоматически перемещаются в новое место. Следовательно, при форматировании раздела приходится выделять запасные дорожки. Выполняя программу форматирования, система UNIX предлагает число запасных дорожек с учетом емкости раздела.
- ✓ UNIX отдельно следит за дисковым пространством, занятым файлами каждого пользователя системы. Каждый пользователь ограничен числом файлов и числом мегабайтов независимо от других пользователей.
- ✓ UNIX определяет для файла больше атрибутов, например различные разрешения и привязки нескольких имен файлов к одним и тем же данным.
- ✓ UNIX определяет для файлов "грязный" бит. Если файл почему-то неправильно записан на диск или его элемент каталога неправильно модифицирован, система отмечает этот факт и требует при следующем запуске системы выполнить эквивалент программы CHKDSK.

Смещение от начала загрузочного сектора DOS		Смещение от начала блока параметров BIOS		Размер BPB в разных версиях DOS		Значения из рис.10.2	
						ASCII или 16-ричные	Десятичное
00h						EBh	
01h						3Ch	
02h						90h	
03h						M	
04h						C	
05h						0	
06h						0	
07h						S	
08h						S	
09h						0	
0Ah						0	
0Bh						00h	
0Ch	00h					02h	512
0Dh	01h					04h	4
0Eh	02h					01h	
0Fh	03h					00h	1
10h	04h					02h	2
11h	05h					00h	
12h	06h					00h	
13h	07h					02h	512
14h	08h					00h	
15h	09h					00h	
16h	0Ah					00h	63,456
17h	0Bh					EBh	Жесткий диск
18h	0Ch					3Eh	
19h	0Dh					00h	62
1Ah	0Eh					20h	
1Bh	0Fh					00h	32
1Ch	10h					40h	
1Dh	11h					00h	64
1Eh	12h					00h	
1Fh	13h					00h	32
20h	14h					00h	
21h	15h					00h	
22h	16h					00h	
23h	17h					00h	
24h	18h					00h	
25h	19h					80h	
26h	1Ah					00h	Первый накопитель
27h	1Bh					29h	
28h	1Ch					09h	
29h	1Dh					17h	2043
2Ah	1Eh					00h	
2Bh	1Fh					00h	
2Ch	20h					N	
2Dh	21h					0	
2Eh	22h					N	
2Fh	23h					A	
30h	24h					M	
31h	25h					E	
32h	26h						
33h	27h						
34h	28h						
35h	29h						
36h	2Ah						
37h	2Bh					F	
38h	2Ch					A	
39h	2Dh					I	
3Ah	2Eh					B	
3Bh	2Fh						
3Ch	30h						
3Dh	31h						
3Eh	32h						

Рис. 10.7. Начало загрузочного сектора DOS, показывающее находящиеся там данные и элементы, которые копируются в блок параметров BIOS

10.4.2. OS/2 и HPFS

OS/2 работает как-то странно. Она может заставить свои диски выглядеть как диски DOS, что позволяет ей разделять такие диски или разделы с DOS. Она может также распределять дисковое пространство, пользуясь высокопроизводительной файловой системой (HPFS), которая совершенно несовместима с FAT для распределения дискового пространства. Можно организовать некоторые накопители с форматом FAT, а другие — с форматом HPFS, что позволяет DOS обращаться к первым накопителям при перезагрузке компьютера с DOS.



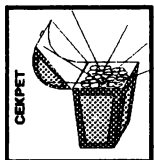
DOS "выросла" из систем CP/M и UNIX, и в каждую версию ее включалось все больше возможностей UNIX. Неудивительно, что и "дети" DOS — система OS/2 — имеет много общего с UNIX. Далее приведены отличия раздела HPFS от обычного раздела DOS.

- ✓ Раздел HPFS не имеет таблиц FAT. Информация о каталоге "разбросана" по нескольким местам диска, поближе к расположению файлов. При этом уменьшаются перемещения головок между файлами и описывающими их элементами каталога. Новый формат каталога позволяет применять длинные имена файлов (до 255 символов).
- ✓ Файловая система HPFS распределяет дисковое пространство секторами (больше нет многосекторных кластеров с их результирующей неэффективностью). К сожалению, средства контроля целостности данных и перемещения их в безопасное место отсутствуют.
- ✓ Раздел HPFS может быть огромным, т.е. его максимальная логическая емкость может быть действительно большой. OS/2 использует BIOS, поэтому INT 13h ограничивает раздел до максимальной физической емкости в 504 М или 8 Гбайт в зависимости от выбранного способа "обмана" (подробнее см. главу 7).
- ✓ OS/2 значительно расширяет понятие атрибутов файла. Система HPFS интенсивно использует кэширование диска, включая кэширование с отложенной записью, но это средство оказывается рискованным (см. главу 5).

Если выключить питание (в случае аварии сети или "зависания" компьютера) до записи информации на диск, файлы не обновляются правильно. Для защиты от катастрофы OS/2 следит за состоянием дискового кэша и извещает вас об ошибке (или несчастье) при следующей загрузке OS/2.

10.4.3. Windows NT и NTFS

Windows NT расширяет введенное в OS/2 понятие "родства" с файловыми системами старых операционных систем. Windows NT сможет считывать и записывать файлы в логическом томе с FAT, логическом томе HPFS или в своих новых томах NTFS.

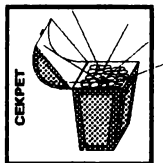


Файловая система NTFS включает большинство достоинств HPFS OS/2, файловой системы UNIX и обладает новыми возможностями. Система NTFS рассчитана на обеспечение максимальной целостности файлов, поэтому в ней имеются средство горячего ремонта UNIX и модель транзакций, обеспечивающие либо запись всего файла, либо ничего. Конечно, оценить практические возможности продукта можно только после его появления на рынке.

10.4.4. Система NetWare

Система NetWare фирмы Novell в силу нескольких причин использует для файловых серверов специальный формат. Наличие особого формата позволило фирме Novell оптимизировать использование диска для специальных целей и обеспечить защиту файлов.

NetWare при обращении к файлам, находящимся в серверах, требует паролей и прав доступа. Такая защита оправдана тем, что не разрешает загрузить ПК сервера с гибкого диска DOS, а затем воспользоваться стандартными командами DOS для считывания или копирования файлов жесткого диска. Благодаря уникальному форматированию высокого уровня системы NetWare для файлового сервера невозможно провести загрузку с диска DOS для доступа к файлам.



Система NetWare известна своей программой COMPSURF. Стандартная процедура инициализации NetWare проводит вас через все этапы подготовки жесткого диска, — от форматирования низкого уровня до форматирования высокого уровня с организацией разделов между ними. Эта процедура дает повод некоторым считать, что фирма Novell осуществляет все уровни подготовки диска специальным образом.

Фактически это не так. Система NetWare просто скрывает от вас большинство этапов. Конечно, вначале она просит контроллер выполнить обычное форматирование низкого уровня. Затем программа COMPSURF проверяет диск как расширение обычных тестов контроллера, после форматирования низкого уровня. Большое число тестов гарантирует NetWare обнаружение всех плохих участков в жестких дисках сервера. Программа COMPSURF организует разделы для NetWare аналогично программе SpinRite для логического тома DOS, по крайней мере в части тестирования целостности диска. Основное отличие связано с тем, что COMPSURF осуществляет это до помещения данных на диск.

Наконец, начиная с версии NetWare 3.x, фирма Novell заменила программу COMPSURF новой процедурой Surface Test (в меню Disk Options программы INSTALL). Эта процедура позволяет пользователю выполнить неразрушающий тест поверхности в любое время. Единственное требование — сначала диск необходимо отсоединить от сети.

После окончания тестирования вновь форматированного диска процедура инсталлирования NetWare создает стандартную таблицу разделов и проводит форматирование высокого уровня — это единственная уникальная для NetWare часть.

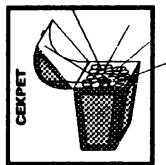
10.5. Как найти дефектные участки

В процессе форматирования низкого уровня контроллер диска, по крайней мере бегло, проверяет каждый сектор, придавая вам уверенность в надежном хранении данных. Несмотря на то, что контроллер устанавливает специальный флажок в заголовке плохого сектора, случайно плохой сектор может пройти тесты контроллера и отмечаться как хороший.

Некоторые контроллеры просто проверяют возможность считывания заголовков без ошибок CRC. Они даже не проверяют правильность информации в заголовке сектора и совсем не контролируют область данных сектора.

Программа форматирования низкого уровня обычно позволяет ввести номера плохой дорожки и плохого сектора. При вводе плохих дорожек контроллер устанавливает флажок плохого сектора в каждом секторе указанных вами дорожек, независимо от результатов контроля. Аналогично при вводе номеров плохих секторов эти секторы отмечаются как плохие, независимо от результата теста контроллера.

Программа форматирования высокого уровня DOS (FORMAT) повторно проверяет всю область данных в каждом форматлируемом разделе DOS или логическом накопителе DOS. (Этот тест несколько лучше, но все равно он сводится к считыванию и не выявляет многих пограничных секторов.) После этого программа FORMAT записывает в FAT специальное значение для каждого кластера, содержащего один или несколько плохих секторов. Отметим, что поскольку DOS не может следить за единицей хранения, меньшей кластера, целые кластеры отмечаются плохими и становятся недоступными для хранения данных.



Нужно помнить еще об одном интересном моменте. Что если заголовков сектора, из которого вы хотите считать, оказывается плохим? В этом случае невозможно считать, что это за сектор. А как же узнать, какой именно сектор является плохим? Ответ состоит в том, что вы (или, точнее, BIOS) можете считать сектор, запрашивая контроллер по его номеру. Если контроллер обнаруживает ошибку, вы знаете, что сектор с указанным номером является плохим (или, возможно, неисправен позиционер головок).

Любой сектор (с плохим заголовком сектора) не способен воспринять соответствующую метку во флажке плохого сектора. Но все же программа FORMAT может выявить его, хотя и после многих попыток считывания, что является причиной раздражающих звуков, когда FORMAT тестирует диск после форматирования высокого уровня.

Никакая другая программа DOS даже не отмечает кластеры плохими. Поскольку вы обычно форматируете диск только один раз, а участок становится плохим после его использования, DOS не поможет найти его, а это приводит к тяжелым последствиям. Подробнее об этом в следующей главе.

Заключение

- В процессе форматирования высокого уровня (или логического форматирования) операционная система создает свою организацию в дисковом томе. В DOS это осуществляет программа FORMAT.
- Формат DOS можно образовать только в томе жесткого диска, ранее созданного программой FDISK (или ее эквивалентом в другой операционной системе ПК), и только после форматирования низкого уровня (физического форматирования). Однако для гибких дисков программа FORMAT выполняет все операции.
- Программа FORMAT разбивает логический том DOS на четыре области: загрузочная секция, таблицы размещения файлов, корневой каталог и область данных, в которой хранятся файлы.
- Подкаталоги — это просто файлы со специальным форматом, который DOS использует особым образом.
- В альтернативных операционных системах, например UNIX, OS/2, Windows NT и NetWare, применяются другие способы форматирования дисковых томов. Форматы UNIX и NetWare совершенно не совместимы с форматом DOS. Системы OS/2 и Windows NT могут использовать логические тома DOS, а также могут создавать несовместимые с DOS тома.
- Кроме программы COMPSURF системы NetWare, никакая другая операционная система для ПК не контролирует целостность поверхности при форматировании тома. Пользователи DOS могут реализовать этот уровень тестирования с помощью программы SpinRite.

Теперь вы знаете все о том, как работают "здоровые" жесткие диски. В следующей главе вы узнаете, как иногда жесткие диски "умирают", как предотвратить их преждевременную гибель и как попытаться оживить умершие накопители.

Глава 11. Как умирают жесткие диски

В этой главе вы узнаете

- Как "умирают" жесткие диски
 - Как различать аппаратные неисправности и выходы из строя жесткого диска из-за порчи информации
 - Некоторые версии DOS имеют опасную программу FORMAT
 - Кажущееся медленное умирание жесткого диска вызывается исправимым ухудшением его производительности
 - Когда реально можно надеяться на восстановление данных с отказавшего жесткого диска
 - Зачем, почему и как создавать эффективные резервные копии ценных данных.
-

11.1. Виды неисправностей

Имеются многочисленные причины выхода из строя жестких дисков. Некоторые из них связаны с ошибками человека, например падение жесткого диска, которое наверняка вызовет его серьезное повреждение. В накопителе возникают случайные отказы аппаратных элементов, могут повредить и внешние события, например выбросы питания. Некоторые накопители становятся жертвой "ползучей" болезни, которая подробно рассмотрена в этой главе.

В зависимости от степени зависимости пользователей от жестких дисков порча диска вызывает ужас и панику. Такое состояние нельзя признать подходящим для поиска неисправности в отказавшем накопителе! Знайте, что часто после отказа жесткого диска вы сможете восстановить свои данные. Одно это должно рассеять ваши страхи и снизить уровень паники.

Поскольку жесткий диск является частью большей подсистемы в ПК, неисправность любого элемента в этой подсистеме выглядит как неисправность жесткого диска. Однако приходится учитывать такие аппаратные части, как КМОП-ЗУПВ конфигурации, контроллер жесткого диска, кабели, соединяющие контроллер с накопителем и накопитель с блоком питания, электронику накопителя и механические детали (рис. 11.1).

Необходимо также рассмотреть различные части хранимой информации, которая требуется для доступа к любому накопителю на жестких дисках. К этой информации относятся тип накопителя и параметры, хранимые в КМОП-ЗУПВ конфигурации, таблицы разделов и таблица параметров BIOS, хранимые в главной загрузочной записи и загрузочной записи DOS, а также информация о размещении файлов и каталогов, хранимая в том же DOS, в FAT, корневом каталоге и различных подкаталогах (рис. 11.2). Загрузка компьютера требует, чтобы все эти аппаратные и информационные "части" функционировали правильно.

Есть несколько тонкостей вопросов о том, как эти информационные части в ПК применяются для доступа к жесткому диску. КМОП-ЗУПВ конфигурации хранит параметры накопителя. Параметры представлены числами в таблице разделов (хранимой в главной загрузочной записи). Эти же параметры хранятся в блоке параметров BIOS для каждого логического тома, который получает информацию из загруз-

зочной записи DOS для этого тома. Автоконфигурирующий контроллер жесткого диска поддерживает еще один набор параметров где-то на диске; обычно к этому набору параметров может обращаться только контроллер.

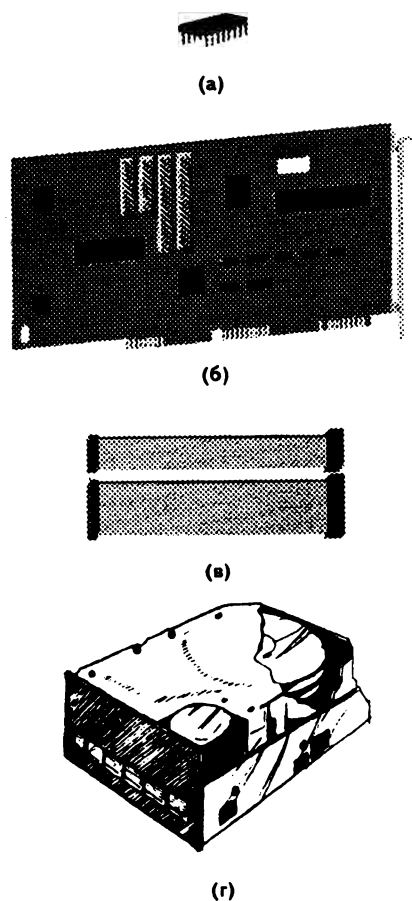


Рис. 11.1. Важнейшие аппаратные элементы подсистемы жесткого диска ПК:

- а* — микросхема КМОП-ЗУПВ конфигурации,
- б* — контроллер, часто на съемной плате,
- в* — кабели,
- г* — дисковый накопитель

А вдруг различные наборы параметров содержат неодинаковые значения? Станет ли это проблемой, зависит от многих обстоятельств. Большинство автоконфигурирующих контроллеров жесткого диска не обращает внимания, какой тип накопителя (и какой подразумеваемый набор параметров) хранится в КМОП-ЗУПВ. Действительно, эти контроллеры не заботятся об этом, если емкость в КМОП-ЗУПВ меньше фактической емкости накопителя. (Если накопитель и контроллер не проводят преобразования секторов, отдельные параметры для количества головок, количества цилиндров и количества секторов на дорожке должны быть меньше или равны соответ-

ствующим реальным количествам. Если накопитель или контроллер осуществляет преобразование секторов, он следит только за тем, чтобы общая емкость была не больше фактической емкости накопителя.)

Некоторые автоконфигурирующие контроллеры используют фактические параметры накопителя и предоставляют вам полную его емкость, даже если выбранный тип накопителя намного меньшей емкости. Другие автоконфигурирующие контроллеры, особенно в накопителях IDE/ATA, могут предоставить только такую емкость, которая соответствует выбранному типу накопителя, хотя реальный накопитель может иметь намного большую емкость. Для решения этой проблемы можно выбрать более подходящий тип накопителя, взять контроллер другого типа или BIOS с правильной емкостью среди емкостей, содержащихся в его таблице типов накопителей.

Иногда компьютер доверяет числам в КМОП-ЗУПВ конфигурации для загрузки системы, а после этого их не использует. Вместо них во всех последующих обращениях к жесткому диску применяются параметры из блока параметров BIOS. В таких системах для правильной работы оба набора чисел (в КМОП-ЗУПВ и блоке параметров BIOS) обязательно должны быть правильными. Если ошибочны только числа в КМОП-ЗУПВ, вы обнаружите, что можно загрузиться с гибкого диска, а затем обращаться к жесткому диску обычным образом, но вы не сможете провести загрузку с жесткого диска.

Иногда возникает еще более непонятная ситуация: вы обнаруживаете, что не можете загрузиться с жесткого диска, но после загрузки с системной дискеты возможно его использование. После проверки вы находите, что параметры накопителя в элементе типа накопителя КМОП-ЗУПВ и в загрузочной записи DOS правильные. По-видимому, вы забыли проверить, установлен ли правильно байт флажка загрузки (называемый также флажком активности) в главной загрузочной записи.

Это только несколько примеров того, как правильная работа ПК зависит от многих вещей. После рассмотрения всех тех мест, в которых должна храниться правильная информация, не удивительно, что большинство неисправностей жестких дисков вызываются не аппаратными неисправностями, а ошибками в данных.

Помогают найти ошибочную часть некоторые диагностические программы, одной из которых является SpinRite. Многие другие утилиты предлагают тестирование диска, но большинство из них проводит довольно беглое тестирование. Недорогой, удобной и простой является программа CheckIt фирмы Touchstone Software Corporation.

11.1.1. Как различать аппаратные и информационные неисправности

Если накопитель сообщает о таких ошибках, как Sector not found reading drive C (При считывании не найден сектор в накопителе C), Read [or write] error on drive C (Ошибка считывания [или записи] в накопителе C) или Cannot find system files (Нельзя найти системных файлов), то, по всей вероятности, контроллер, кабели и аппаратные элементы накопителя исправны. Проблема связана с информацией на диске, а не с диском. Проще всего в такой ситуации выполнить программу SpinRite.

Такие сообщения, как Invalid drive specification (Неправильная спецификация накопителя) или General failure reading drive C (Общая неисправность при считывании в накопителе C), показывают более серьезные аппаратные проблемы. Но даже здесь причиной неисправности может быть только неправильная информация, записанная на поверхности диска. Вспомните об этапах подготовки накопителя для DOS; если повреждается любой из наборов, записанных на этих этапах, диск может стать полностью недоступным до ремонта повреждения. Когда вы встречаетесь с такими проблемами, имеет смысл вначале попробовать самые простые и дешевые "лекарства".

Хуже всего для диагностики, когда ПК не делает ничего. Если вы пытаетесь загрузить жесткого диска и система зависает, не выдавая никаких сообщений об ошибках, это служит симптомом нескольких различных неисправностей. Вам нужно

пройти по списку упорядоченным и систематическим образом для выявления неисправности и ее исправления (возможно, если вы невезучий, вам придется искать и исправлять несколько поломок по очереди).

**Важнейшие информационные "части",
необходимые для доступа к файлу**

Тип накопителя (или параметры) в
КМОП - ЗУПВ

Элементы таблицы параметров жесткого диска

Элементы таблицы разделов
(находятся в главной загрузочной записи)

Элементы таблицы параметров BIOS
(находятся в загрузочной записи)

Таблицы размещения файлов

Каталоги и подкаталоги

Данные в файле

**Дополнительные информационные "части",
требуемые для загрузки**

Программа главной загрузочной записи

Скрытые системные файлы DOS

Файл CONFIG.SYS и все указанные в нем
драйверы устройств

Программа командного интерпретатора
(COMMAND.COM)

Файл AUTOEXEC.BAT и все указанные
в нем программы

Рис. 11.2. Важнейшие информационные "части" подсистемы жесткого диска ПК

11.1.2. Простые неисправности

Осмотрите жесткий диск. Прислушайтесь к нему. Не звучит ли он по-другому? Обычным ли образом работает индикатор? Нагрелся ли накопитель, как обычно? (Для проверки температуры накопителя нужно дать ПК поработать несколько часов, а затем открыть системный блок и коснуться корпуса накопителя.)

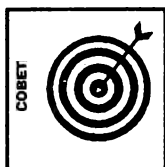
Чтобы узнать о необычной температуре накопителя, необходимо заранее, до возникновения неисправности, открыть системный блок и, потрогав накопитель, узнать его обычную температуру. (Эта процедура похожа на измерение вашей температуры, взвешивание и измерение пульса при посещении доктора, даже если вы вроде и не больны.)



При работе с ПК соблюдайте следующие меры предосторожности.

- ✓ Перед тем, как открыть ПК, отключите его от сети. Не просто поверните выключатель питания, а отсоедините шнур питания. Отсоедините также силовые шнуры для монитора и других подключенных к компьютеру периферийных устройств (или отсоедините кабели данных, которые связывают ПК с этими устройствами). Если проигнорировать эти меры, вы можете серьезно повредить ПК и себя.
- ✓ Коснитесь блока питания ПК перед касанием чего-то еще в системном блоке. Это гарантирует, что вы и ПК находитесь под одинаковым напряжением. Поступайте так всегда, открыв системный блок.
- ✓ Не двигайте компьютер, когда он включен. В частности, не наклоняйте корпус. Наклон вращающегося гироскопа создает очень большие усилия в его подшипниках.
- ✓ После проведения некоторых операций внутри ПК вы можете захотеть до закрытия крышки включить ПК и выполнить короткий тест. Это допустимо. Можно даже вставлять и вынимать кабели от ПК к периферийным устройствам, когда компьютер включен. (До подключения кабеля коснитесь внешней заземленной оболочкой разъема корпуса системного блока.) Но при включенном ПК никогда не вставляйте и не вынимайте кабели внутри ПК (например, кабели от блока питания к накопителю и от накопителя к контроллеру).
- ✓ Не закрывайте крышку включенного ПК. Очень легко уронить винт и закоротить что-нибудь.

Если отсоединен кабель от блока питания к накопителю, накопитель совсем не работает. Индикатор не светится и накопитель не нагревается. Такую ситуацию легко просмотреть, но просто исправить, вынимая кабель, а затем вставляя его с некоторым усилием. (Да, часто трудно вынимать эти кабели, так они плотно вставлены. Вам придется сделать некоторое усилие. При этом будьте осторожны и не согните печатную плату, на которой смонтирована неподвижная часть разъема. Не соблюдая мер предосторожности, можно нарушить проводники на печатной плате.)



До отсоединения или замены кабелей сделайте набросок, где и как они подключены. Если подключить кабели неправильно, вероятно, повреждения не будет, но и работать жесткий диск наверняка не будет. Нужно также соблюдать правильную ориентацию разъемов. Некоторые кабели имеют "ключ", предотвращающий неправильное подключение, но большинство такого ключа не имеет.

Отсоединение и подключение кабелей может помочь, так как при этом протираются их контакты. Трение удаляет оксиды и жир, препятствующие хорошему электрическому соединению. Во время такой чистки вы можете по одному отсоединять разъемы, а затем соединять. При этом маловероятно провести неправильное соединение.

Протирка также помогает обеспечить надежное соединение съемных плат, материнской платы или периферийного устройства. Частично вставленный разъем — еще одно распространенное затруднение.

Будьте внимательны при подключении кабелей. Нельзя прилагать большее усилие, чем усилие при отключении кабеля. Усилие зависит от типа и размера разъема, но в обеих операциях оно примерно одинаково.

Еще одной причиной неисправности может стать потеря соединений в кабелях данных (обычно это плоские ленточные кабели). По-прежнему нужно отсоединить кабель, а затем подключить его. Выполните эти действия с каждой стороны кабеля. Такой прием является единственным надежным способом диагностики этого сбоя. Если он не помогает, попробуйте заменить кабели новыми.

Очень редко могут порваться один или несколько проводников в кабеле. Даже изгиб контакта разъема на одном конце достаточно для нарушения контакта. Если это произошло с одним из кабелей жесткого диска, замените его. К счастью, при очень внимательном осмотре можно заметить повреждение кабеля, а стоимость его невелика.



Во многих ПК в ограниченном пространстве может произойти зажатие ленточного кабеля, что нарушает поток воздуха, вызывая чрезмерный нагрев ПК, или вызывает трение кабеля о заднюю сторону съемной платы. Это трение опасно, так как на задней стороне большинства плат есть острые концы проводников. Эти концы могут прорезать изоляцию кабеля, что приведет к закорачиванию проводников в схеме платы. Проще всего проложить между кабелем и острой стороной платы кусочек тонкого картона.

11.1.3. Медленная порча из-за нарушения выравнивания

Жесткие диски — это механическое чудо. Они перемещают головки на миллионные доли дюйма и обеспечивают их точное позиционирование на старое место, что по меньшей мере очень трудно осуществить. Со временем механизм позиционирования жесткого диска просто не может поместить головки точно туда, где они должны быть.

С точки зрения головок дорожки информации, записанной на поверхности диска, немного сдвигаются к одной стороне. Диск работает хорошо, пока этот сдвиг невелик, но при увеличении сдвига работа подсистемы диска нарушается.

Напомним, что компьютер помещает информацию в заголовки сектора только при проведении форматирования низкого уровня. По мере старения накопителя головки могут сдвинуться относительно записанных заголовков секторов. (Такого смещения не происходит в накопителях со встроенной или вмонтированной сервосистемой, но оно часто возникает в накопителях других типов.)

Сдвинутые относительно дорожек головки воспринимают все меньшую часть общей ширины дорожки, сигналы от головок уменьшаются и контроллеру все труднее

считывать заголовки. Такая проблема медленно нарастает до тех пор, пока однажды ПК не выдаст опасное сообщение Sector not found (Сектор не найден) -- (рис. 11.3).

Если при форматировании низкого уровня накопитель испытывает толчок, головка моментально немного отклоняется в сторону. При этом дорожка становится волнистой. В зависимости от способа дрейфа блока головок при старении, такой сектор рано или поздно вызовет затруднение (рис. 11.4).

Ошибка считывания может возникнуть, если считываемый сектор записан только один раз. Магнитный узор со временем ослабевает или искажается, поэтому регистрируемые головкой сигналы становятся слишком слабыми для правильной расшифровки или содержат неправильные биты. Вначале записанных вместе с информацией кодов исправления ошибок достаточно для получения данных, но, в конце концов, не помогают и они.

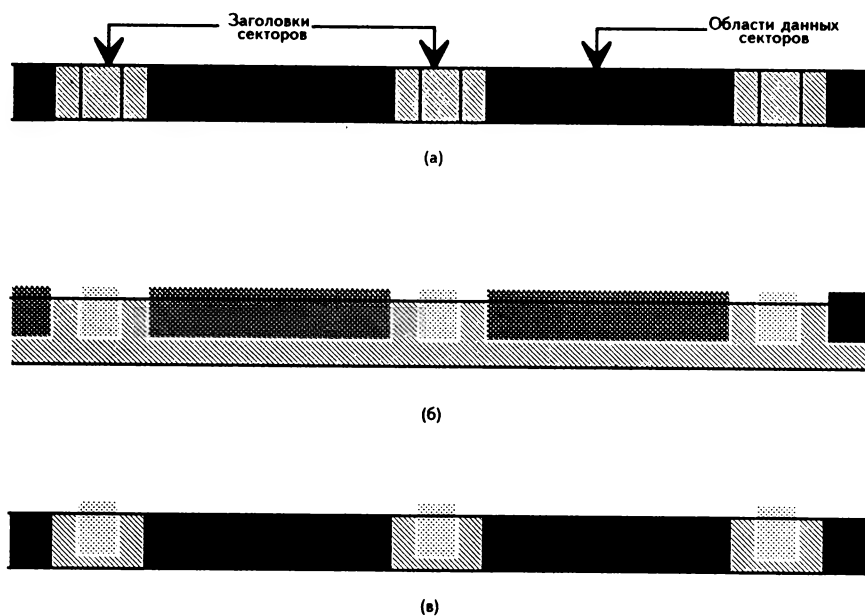
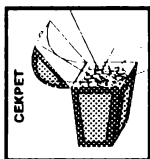


Рис. 11.3. Как дрейф головок вызывает неисправность:

- а* — при первой записи дорожка выравнена с записывающей головкой;
- б* — если позже головка дрейфует к одной стороне, она не может больше надежно считывать информацию с дорожки;
- в* — перезапись данных выравнивает данные (но не заголовки секторов) под новый путь головки. Это приводит к ошибкам Sector not found (Сектор не найден). Повторное форматирование низкого уровня восстанавливает первоначальное совершенное выравнивание



Сами головки постоянно намагничены. Если не парковать головки при каждом выключении компьютера, такие магниты могут опуститься на ценные данные. При запуске накопителя головки до подъема в воздух могут поцарапать дорожки. Из-за этого записанные сигналы ослабевают или искажаются.

Система время от времени записывает информацию в большинство секторов. Следовательно, при дрейфе выравнивания накопителя область данных постоянно перезаписывается под текущим положением головок. Постепенно сектор, который при форматировании низкого уровня был отмечен как плохой (возможно, из-за крошечного отверстия в магнитном покрытии поверхности), может стать хорошим, а хороший сектор превращается в плохой (рис. 11.5).

Повторное форматирование низкого уровня исключает некоторые затруднения, сдвигая заголовки секторов и области данных под новое положение головок. Если вы получили сообщение Sector not found (Сектор не найден), такой прием может исключить это. Но, как показывает рис. 11.5, форматирование низкого уровня не решает всех возникающих проблем. После повторения форматирования низкого уровня важно тщательно проверить секторы, чтобы не помещать данные в плохие секторы.

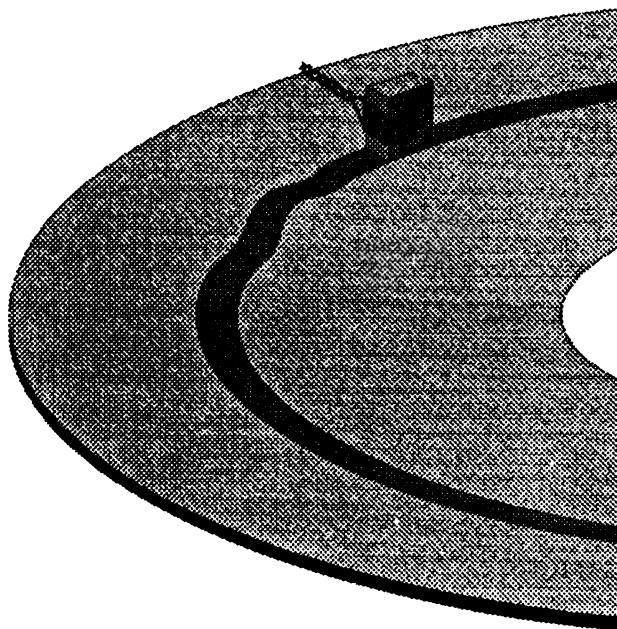


Рис. 11.4. Толчок жесткого диска в процессе форматирования низкого уровня вызывает появление волнистой дорожки

Для цилиндров накопителя, находящихся внутри раздела DOS или логического накопителя DOS, программы изменения коэффициента чередования типа SpinRite предоставляют наиболее удобный способ "освежения" формата, проверки поверхности диска и замены данных. Если выполнять одну из этих программ до того, как дрейф головок воспрепятствует считыванию данных, такие ошибки могут не встретиться никогда. Удивительно, но даже если вы дождались того, что DOS не может считать почти ничего, иногда такая программа восстанавливает все данные и устраняет ошибки накопителя.

11.1.4. Сбои данных

Некоторые неисправности накопителя вызываются случайными событиями, например выброс питания изменяет единицу на ноль или наоборот. Примерами распространенных ошибок пользователей служат стирание файла COMMAND.COM или копирование на него другой версии программы; стирание всех файлов в корневом каталоге или ошибочное выполнение команды FORMAT C:, несмотря на предупреждение DOS о потере всех данных на фиксированном диске.

Типичными индикаторами несчастий с данными служат такие сообщения:

Probably Non-DOS disk
Возможно, не-DOS диск

Bad or missing command interpreter
Плохой или отсутствующий интерпретатор команд

Bad or missing xxxx
Плохой или отсутствующий xxxx (где xxxx — имя драйвера устройства из файла CONFIG.SYS или программы из AUTOEXEC.BAT)

Bad Partition Table
Плохая таблица разделов

Disk boot failure
Ошибка загрузки с диска

Disk error reading FAT x
Ошибка считывания FAT x (где x равен 1 или 2)

Error loading operating system
Ошибка загрузки операционной системы

Invalid partition table
Неправильная таблица разделов

Missing operating system
Отсутствует операционная система

В зависимости от использованной версии DOS и ее адаптации производителем компьютера сообщения могут несколько отличаться от приведенных, но их общий смысл одинаков.

К счастью, некоторые сбои данных являются временными и исчезают при простой перезагрузке компьютера. Но устойчивые затруднения требуют поиска и исправления поврежденной информации.

Одна малоизвестная причина сбоев данных. Ошибка в данных может возникнуть, если неосторожно включить ПК сразу после его выключения. Для всех ПК рекомендуется никогда не включать компьютер до тех пор, пока он находится в середине процесса выключения.

При выключении питания в ПК происходит следующая последовательность событий. Напряжения от блока питания уменьшаются до нуля. Жесткий диск прекращает вращаться. Детали охлаждаются, и их физические размеры немного изменяются.

Продолжительность этих событий различна. Напряжения от блока питания уменьшаются до нуля за несколько секунд. Жесткий диск останавливается через несколько секунд. Для охлаждения деталей ПК до комнатной температуры требуется несколько минут.

Последствия слишком быстрого включения компьютера зависят от того, на какие процессы попадает включение. Инженеры фирмы IBM решили, что нельзя позволять пользователям включать ПК до истечения некоторого временного интервала. Поэтому во все ПК они встроили реле выдержки времени. Производители некоторых клонов поступают так же, но большинство такое реле не ставят. Если в блоке питания вашего ПК нет реле выдержки времени, настоятельно рекомендуется выработать привычку не включать ПК по истечении минимум 30 с после его выключения.

Если выключить ПК, имеющий в блоке питания реле выдержки времени, и сразу же включить его, ничего не произойдет. ПК остается выключенным. Вы должны выключить его и подождать срабатывания реле (если прислушаться, то его срабатывание можно услышать). Обычно выдержка составляет несколько секунд. После этого можно включить ПК обычным образом.

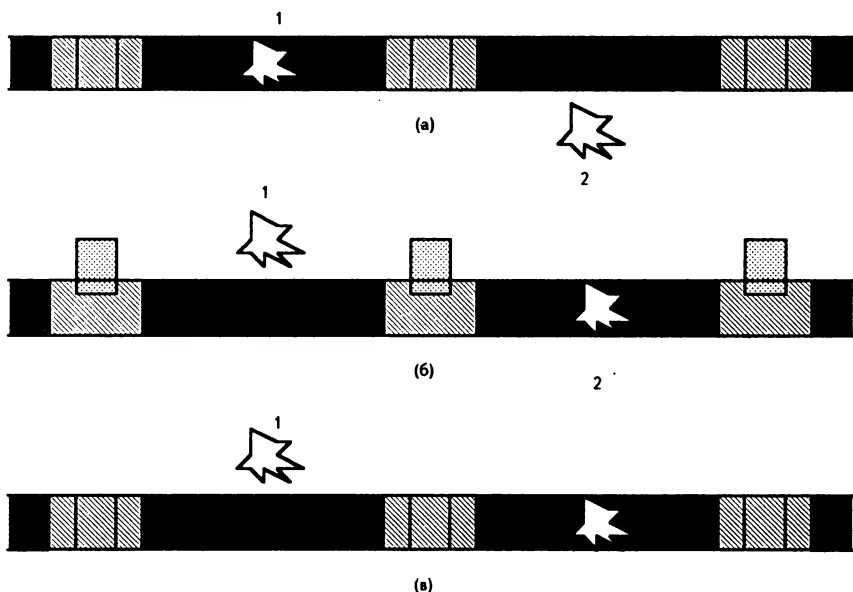


Рис. 11.5. Крошечные отверстия в поверхности могут появляться и исчезать, превращая секторы в плохие или хорошие:

- а — отверстие в поверхности превращает сектор #1 в плохой при форматировании низкого уровня;
- б — дрейф головки сдвигает данные в секторе к одной стороне, причем первое отверстие исчезает, но появляется другое. Теперь сектор #1 стал хорошим, но сектор #2 плохим;
- в — даже повторение форматирования низкого уровня (которое выравнивает заголовки секторов с данными) не помогает. Сектор #2 по-прежнему остается плохим

Что же происходит, если ПК без реле включить сразу после его выключения?

Наиболее часто возникают две трудности.

Во-первых, такое действие оказывается очень тяжелым для жесткого диска. Если он замедляется и головки начинают "приземляться" (или только что "приземлились") на поверхность диска, подача питания может привести к неприятным последствиям. Некоторые из них вызывают повреждение данных, а другие могут привести к отказу аппаратуры и сделать весь накопитель бесполезным.

Во-вторых возможна порча информации в конфигурации КМОП-ЗУПВ. В некоторых клонах ПК такое событие не просто возможно — в них содержимое КМОП-ЗУПВ искажается всегда при быстром включении после выключения.

Как исправить поврежденные данные. Искажение данных в ПК вызывается многими причинами. Далее рассмотрено несколько примеров и приведены рекомендации, как справиться с этими трудностями.

- ✓ Если вы реформатировали жесткий диск командой `FORMAT C:`, нужен способ отмены последствий форматирования. Первой программой для восстановления форматирования был пакет Mace Utilities (он и сейчас предлагается фирмой Fifth Generation Systems). В пакет утилит Нортон (фирма Symantec) также включена такая возможность. Имеется она и в пакете ПК Tools фирмы Central Point Software. Начиная с версии 5 в DOS имеется программа `UNDELETE`, которая также решает эту проблему. Во всех случаях такие программы действуют лучше, если вначале сохранить ключевые данные для восстановления.

При использовании программы `DOS 6 Doublespace` для уплотнения некоторых накопителей нужно знать об одной особенности. В строке команды `DBLSPACE.EXE` можно добавить ключ `/FORMAT`, который уничтожит все файлы в уплотненном накопителе. Это действие исправить с помощью `UNDELETE` невозможно.

- ✓ При удалении правильной версии файла `COMMAND.COM` или копировании на него другой информации решение проблемы заключается в следующем. Скопируйте файл на жесткий диск с оригинального диска DOS. При удалении всех файлов в корневом каталоге их можно восстановить с резервных файлов. Если резерва нет, можно воспользоваться одной из указанных ранее программ для восстановления после удаления ("`unerase`").
- ✓ Если загрузиться с диска нельзя или на нем испорчена таблица разделов, в такой ситуации могут помочь программы `Norton Disk Doctor`, `DiskTool` и `Disk Editor` из утилит Нортон. Для ремонта поврежденной таблицы разделов и обновления системных файлов DOS предназначена программа `KFDISK` из пакета `HTHF` фирмы Kolod Research. Применение коммерческих программ для решения рассматриваемых проблем имеет два преимущества: они работают автоматически и позволяют отменить выполненные действия, если они оказались бесполезными.
- ✓ Если вы решили применять для восстановления только команды DOS, нужно посоветоваться с опытным пользователем — есть много способов неправильного применения команд DOS и только один правильный.

Если, например, вы хотите исправить таблицу разделов с помощью `FDISK`, то можно уничтожить самую важную информацию на диске. Это связано с тем, что, кроме изменения элементов таблицы разделов, многие версии `FDISK` при создании раздела очищают первые несколько цилиндров.

При использовании `MS DOS 5+` можно привлечь недокументированную команду `FDISK/MBR` для восстановления главной загрузочной записи-таблицы разделов, не изменяя чисел в таблице разделов и не уничтожая ценных данных, находящихся в разделе DOS или расширенном разделе.

Разрушающие FORMAT-программы. Программа FORMAT может полностью инициализировать гибкий диск, проводя форматирование низкого и высокого уровня. Единственное исключение связано с применением команды FORMAT DOS 5 или 6 в безопасном или быстром режимах. На жестком диске в большинстве версий DOS команда FORMAT инициализирует только загрузочную запись, FAT и корневой каталог. Это допускает хотя бы частичное восстановление после форматирования диска.

Некоторые версии программы DOS FORMAT проводят больше действий на жестком диске. Например, система DOS фирмы AT&T до версии 3.1, система DOS фирмы Compaq до версии 3.2 и некоторые версии DOS фирмы Burroughs имеют такие программы FORMAT, в которых обычные функции дополнены инициализацией области данных раздела DOS или логического накопителя DOS. Это отличие не допускает восстановления данных в разделе или логическом накопителе DOS даже с помощью утилит Нортон. Чтобы избежать подобной катастрофы, следует перейти на новейшую версию DOS.

Если у вас нет DOS 5 или 6, целесообразно заменить программу FORMAT на программу Safe Format (Безопасное форматирование) из утилит Нортон или программу PCFORMAT из пакета PC Tools. Если форматирование низкого уровня уже проведено, эти программы выполняют только форматирование высокого уровня даже на гибких дисках (если специально не запросить разрушающее форматирование), допуская применение UnFormat для восстановления данных.

Начиная с версии 5, при использовании программы MS DOS FORMAT для реформатирования диска с данными она сохраняет информацию, допускающую восстановление файлов, если они не были перезаписаны. Программы-утилиты третьих фирм также обладают такой возможностью.

С другой стороны, при продаже своего компьютера вы захотите полностью стереть жесткий диск, чтобы защитить свои конфиденциальные данные. В этом случае вместо простого удаления файлов (командами DOS DEL или ERASE) или даже применения команды FORMAT, следует привлечь программу WIPEINFO из Утилит Нортон или команду Clear File (Очистка файла) программы PCSHELL из пакета PC Tools.

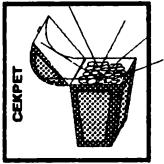
При работе с DOS 5 можно заставить FORMAT стереть весь диск, указав опцию /U. Эта опция действует достаточно хорошо, но программа WIPEINFO оказывается еще лучше.

11.1.5. Аппаратная неисправность в накопителе

Наиболее частая аппаратная неисправность связана с тем, что двигатель не вращается из-за того, что головки приклеились к поверхности диск. (На такие отказы приходится примерно 60% всех неисправностей жестких дисков.)

Статическое трение. Такой отказ называется статическим трением или Stiction — Static Friction. (Статическое трение — это сила, которая препятствует скольжению одного объекта по другому. Слово "статический" здесь означает отсутствие движения и совершенно не связано со статическим электричеством. В общем трение скольжения, т.е. сила, препятствующая движению, несколько меньше.) Оказывается, что после выключения ПК головки приклеиваются к дискам. При включении компьютера усилия двигателя недостаточно для сдвига головок. Если освободить диск, накопитель будет работать до следующего выключения.

Головки приклеиваются из-за многих причин. Одна из них связана с тем, что некоторые накопители покрываются шеллаком или лаком. В горячем состоянии покрытие разжижается. При выключении накопителя головки опускаются на поверхность до того, как покрытие затвердевает. После охлаждения накопителя покрытие действует как клей, удерживающий головки на диске.



При определенной смелости можно иногда на время устранить эту причину, применив простой прием. Выключите ПК и отсоедините все кабели. Откройте системный блок. Выньте жесткий диск. (Не пытайтесь открывать корпус накопителя, так как при этом он просто выйдет из строя.) Не забудьте набросать схему соединения кабелей.

Теперь для того чтобы заставить диски вращаться, можно выполнить несколько действий. Далее они описаны в порядке возрастания "жестокости" по отношению к накопителю.

- ✓ Держа накопитель в руках, резким движением поверните его. Нужно поворачивать корпус в том же направлении, в каком вращаются диски. Если провести эту операцию достаточно энергично, инерция дисков удерживает их от вращения, а поворот корпуса заставляет блок головок поворачиваться. При этом соединение между головками и дисками может разрушиться.

После того как, по вашему мнению, соединение разрушено, можно временно подключить накопитель к ПК с открытым системным блоком. (Можно поставить накопитель на раму ПК, только правильно подключите кабели. Обеспечьте, чтобы накопитель не соскользнул и не упал с рамы. Для защиты от коротких замыканий положите между накопителем и рамой ПК кусок картона.) Включите ПК и посмотрите, вращаются ли диски.

- ✓ Если этот прием не помогает, попробуйте следующий, более смелый (и более грубый) способ. Положите диск так, чтобы схемная плата была сверху, т.е. не так, как накопитель монтируется в ПК.

По всей вероятности вы заметите отверстие в плате, через которое виден конец шпинделя. (Отверстие может быть закрыто пластиковой крышкой и ее нужно аккуратно снять.) Если плавно провернуть ось длинной отверткой или пассатижами, можно заставить диск вращаться свободно. В случае успешного поворота временно подключите накопитель к ПК и проверьте его работу.

- ✓ Если затруднение осталось, можно сделать еще одну "последнюю и решительную" попытку. Сейчас вам терять уже нечего, поэтому можно попытаться "оживить" диск. Подключите накопитель к ПК и включите питание. Затем поднимите один угол накопителя примерно на дюйм или два, а затем уроните его на стол или раму ПК. Если вам повезет, диски начнут вращаться.

После того как диски начали вращаться, нужно сразу же зарезервировать все файлы. Возможно, следующего шанса у вас не будет. Разумеется, не нужно выключать ПК до окончания резервирования.

Неисправность подшипника шпинделя. Иногда полностью заедает подшипник шпинделя и диски не вращаются. Производители накопителей сообщают, что они смазывают двигатели на всю жизнь, и они не лгут: после заедания подшипника жизнь накопителя закончена. Примерно 15% накопителей погибают именно таким образом. Симптомы заедания подшипника напоминают симптомы статического трения, но теперь вы не сможете заставить диски вращаться, что намного усложняет получение данных.

Иногда накопитель сообщает о возможном заедании подшипника. У автора первый накопитель начал издавать свистящий звук через неделю после поставки. Автор вернул его дилеру и спросил о причине свиста. Техник заменил накопитель и сообщил о приближающемся заедании подшипника.

Если подшипники накопителя выходят из строя более скрытно, все же можно распознать надвигающуюся беду. Когда одна из программ, например SpinRite, показывает скорость вращения, внимательно посмотрите значение скорости. Большинство жестких дисков вращаются со скоростью 3600 обор./мин, но у некоторых она достигает 5400 обор./мин. Скорость вращения всегда должна быть примерно постоянной. Если вы при проверке заметите необычно большие отклонения скорости, вскоре

подшипники накопителя выйдут из строя. (Иногда отклонение скорости вызывается двигателем, но это случается намного реже.)

Единственный способ восстановить данные после заедания подшипника состоит в том, чтобы открыть накопитель и как-то заставить шпиндель вращаться; конечно, такие действия вам не под силу. Дорогое решение заключается в отправке накопителя в специализированную фирму. При наличии у вас хороших резервных копий, вы сэкономите деньги просто заменив накопитель новым и загрузив программы и данные с резервных копий.

Неисправности электроники накопителя. Еще примерно 20% аппаратных отказов накопителей приходится на неисправности электронных схем, смонтированных на накопителе. Накопитель и головки могут работать хорошо, но электронная схема не обеспечивает запись или считывание данных. При отказе электроники контроллера накопителя (двигателя) накопитель не работает или не вращается с требуемой скоростью. Для диагностики этой поломки можно заменить электронную плату работоспособной платой от аналогичного накопителя. (В ремонтной мастерской может быть запасная плата.)

Температурные трудности. Часто в накопителях возникают температурные проблемы. Некоторые накопители не работают до полного прогрева, а другие — при нагреве "гибнут". Иногда эта проблема решается повторным форматированием низкого уровня, которое удобно выполнять программой изменения коэффициента чередования. В другое время накопитель в самом деле "болеет" и требует ремонта или замены.

Накопитель может отказать из-за медленной деформации или износа механических деталей, которые изменили форму так, что могут работать правильно только после нагрева. В таких случаях необходимо зарезервировать данные, а затем заменить или отремонтировать накопитель.

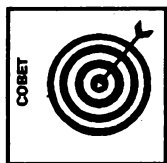
Одна из довольно распространенных температурных трудностей возникает, когда перестает работать деталь на схемной плате. Она может функционировать правильно, пока плата холодная, но при нагреве накопителя она перестает работать, возможно, из-за более быстрого включения или неправильной работы схемы обработки данных. В этом случае для получения данных с диска нужно отодвинуть схемную плату как можно дальше от накопителя. (Будьте аккуратны и не оторвите проводники, идущие в закрытый корпус накопителя.) После этого схема остается холодной долгое время, достаточное, чтобы зарезервировать файлы, а затем плату нужно отремонтировать или заменить.

Если диск начинает загрузку, а затем не может найти системные файлы, проблема скрыта в информации, хранимой в разделе DOS. Выполнение программы изменения коэффициента чередования диска типа SpinRite может сместить дорожки под головки и "усилить" записанную информацию достаточно для правильной работы накопителя.

Когда этот прием не помогает, необходимо загрузить компьютер с безопасной загрузочной дискеты (подробнее о создании такой дискеты см. в главе 1.) Если у вас пока нет безопасной загрузочной дискеты, можно воспользоваться любой загрузочной дискетой DOS с той же версией DOS, что и на жестком диске. Если после этого вы сможете обратиться к данным, повреждение данных невелико и программа Norton Disk Doctor обычно исправляет данные.

Когда накопитель даже не начинает загрузку, проблема связана с чтением информации вне раздела DOS, о чем свидетельствуют сообщения Invalid partition table (Неправильная таблица разделов), Error loading operating system (Ошибка загрузки операционной системы) или Missing operating system (Отсутствует операционная система). Если в вашей системе есть накопитель с аналогичными разделами тех же типа и емкости, можно скопировать таблицу разделов хорошего накопителя в отказавший. (Некоторые популярные утилиты позволяют сохранить таблицу разделов жесткого диска в виде файла на диске. Они также предоставляют простой способ исполь-

зовать этот диск для ремонта любого повреждения таблицы разделов жесткого диска.)



Замораживание данных

Вам придется ремонтировать (или заменять) накопитель, который перегревается. Один из способов извлечения данных с такого накопителя до отправки его в ремонт состоит в интенсивном охлаждении. Нужно выпнуть накопитель из ПК и поместить его на несколько часов в холодильник или морозильник, а затем подключить к ПК. После включения ПК и начала работы накопителя скопируйте с него все возможные данные. Когда накопитель перегреется, выключите ПК, вновь поместите накопитель в морозильник и повторите копирование.

Ставьте в морозильник только накопитель. Не охлаждайте сам ПК, так как конденсация влаги после его нагрева вызовет новые затруднения.

До охлаждения плотно заверните накопитель в пластик. Если оставить кабели подключенными к накопителю и торчащими из обертки, но все же не допустить поступления воздуха в накопитель, то при запуске накопителя обертку можно оставить. Такая мера предосторожности помогает предотвратить конденсацию влаги на электронной плате накопителя. Можно также работать в комнате с кондиционированием воздуха или в помещении с низкой влажностью.

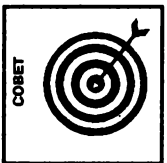
Если накопитель все же не работает, вам придется воспользоваться своими резервными копиями. Выполните форматирование низкого уровня как для нового накопителя. С помощью FDISK создайте информацию о разделах. Осуществите форматирование высокого уровня (программой FORMAT). Повторно инсталлируйте свои программы. (На это уходит много времени, поэтому нужно постараться обойтись более простыми "лекарствами"; но если они не помогают, воспользуйтесь приведенным способом.)

Если холодный накопитель работает, а при нагреве отказывает, причиной может быть дрейф механизма. В этом случае должны помочь описанные выше приемы. Но, с другой стороны, накопитель может сильно перегреваться из-за подшипников (при нагреве диски прекращают вращаться) или электроники.

Крах головки. Повреждения из-за краха головки возникают редко. При каждом выключении ПК головки плавно "приземляются", но благодаря их конструкции повреждение магнитного покрытия поверхности происходит редко.

Для дополнительной защиты от повреждения следует парковать головки, позиционируя их на такое место, где нет важных данных. Всегда перед выключением ПК выполняйте программу парковки (если, конечно, в накопителе не предусмотрена автоматическая парковка). Необходимо парковать головки, если ПК будет бездействовать длительное время.

Но даже при соблюдении правил парковки головок жесткие диски могут пострадать от краха головки, например при падении работающего ПК на пол при землетрясении. Приятно узнать, что при этом теряется примерно 20% данных, а остальные 80% можно зарезервировать.



Как избежать простых ошибок

При установке платы электроники на жесткий диск не нужно сильно крепить ее винтами, чтобы не вызвать деформации рамы. Деформация может привести к трению дисков о корпус и невозможности вращения.

Аналогично при монтаже накопителя в ПК постарайтесь не деформировать раму накопителя или закрытый корпус. В компьютерах PC или XT (и в некоторых AT)

накопитель крепится к корпусу винтами. Пользуйтесь очень короткими винтами или обеспечьте достаточное расстояние позади отверстия, чтобы винт не касался закрытого корпуса (рис. 11.6). Не применяйте больше двух винтов и никогда с противоположных сторон.

При монтаже накопителя винтами с двух сторон оставьте их свободными с одной стороны. Можно применить клей Loctite, чтобы предотвратить выпадение этих винтов со временем. Винты с другой стороны следует подтянуть как обычно, т.е. "сильно, но не очень сильно" (Клей Loctite имеет три показателя прочности и целесообразно выбрать наименьшую прочность. Когда-нибудь вам придется удалять эти винты; вам просто нужно, чтобы винты не выпадали.)

В большинстве компьютеров АТ и с процессором 386 жесткие диски имеют направляющие, привинченные с каждой стороны; направляющие скользят по каналам в раме ПК. Зажимы в передней части корпуса компьютера предотвращают соскальзывание накопителей из этих каналов.

Это очень хороший способ предотвратить напряжение в накопителе из-за сильного крепления. Нельзя только применять слишком длинные винты для крепления направляющих к накопителю; они не должны касаться закрытого корпуса жесткого диска.

Плохо то, что разрушенное магнитное покрытие попадает в другие места внутри накопителя. Оно может засорить вентиляционные отверстия, приклеить головки считывания-записи или еще как-то повредить спустя некоторое время. Можно, конечно, работать с накопителем, в котором возникла неисправность головки, но рассчитывать на его продолжительную эксплуатацию не приходится.

11.1.6. Отказ контроллера

Платы контроллеров жестких дисков отказывают редко, но последствия возникшего отказа весьма велики. Например, при выполнении команды с обращением к жесткому диску ПК может выдать такие сообщения: Invalid drive (Неправильный накопитель), General failure reading drive C (Общая ошибка чтения в накопителе C) или Drive not ready (Накопитель не готов).

Если отключение и подключение кабелей, ведущих от жесткого диска к контроллеру, не помогает, можно попробовать вынуть плату контроллера из ПК, а затем вставить ее. Когда плата вынута из ПК, целесообразно прочистить контакты, которыми она вставляется в материнскую плату. (Не забудьте до открывания ПК выключить питание и отсоединить все кабели. Помните также о необходимости коснуться блока питания или рамы до касания других частей в системном блоке.) Если и этот прием не помогает, причиной неисправности могут быть плата контроллера, диск или повреждение данных.

Диагностические программы помогают выяснить наличие неисправности в накопителе или контроллере. И иногда они даже исправляют отказ! Можно также попробовать заменить накопитель или контроллер и выявить неисправный. (Если вы не знаете, что заменять первым, имейте в виду следующее. Набор кабелей стоит примерно 10 долл. Новый контроллер стоит дороже, но не больше 100 долл. Новый накопитель потребует несколько сотен долларов и при его замене вы потеряете все данные, хранимые на старом накопителе.



Для считывания и записи данных на вашем диске должны быть наборы информации, записанные вашим контроллером. Помните, что каждый контроллер при форматировании низкого уровня помещает информацию несколько иначе, чем любой другой. При замене контроллера новым вы, по всей вероятности, потеряете доступ ко всем данным на диске, хотя диск не поврежден. Единственное исключение, когда новый контроллер той же модели и уровня ревизии, что и старый; в

этом случае обеспечивается доступ к данным. Замена контроллера — это крайний случай. Имейте также надежные резервные копии, так как они, по-видимому, вскоре понадобятся.

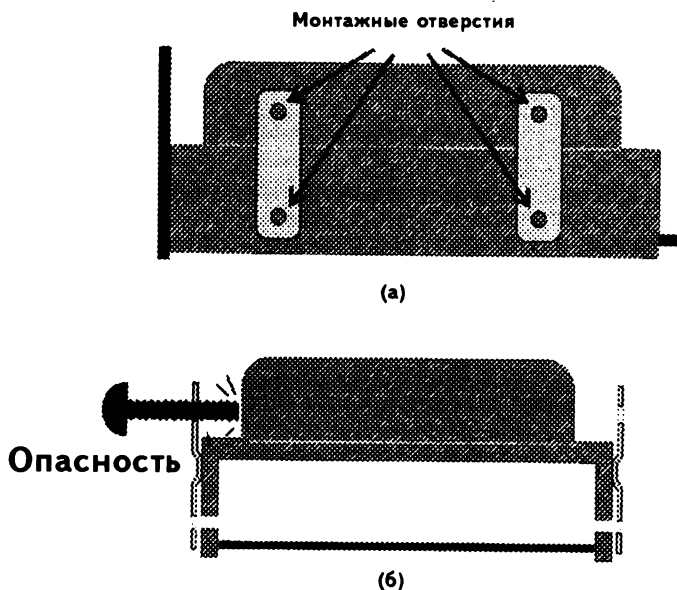


Рис. 11.6. Ошибка, которой нужно избегать при монтаже жесткого диска:

а — вид сбоку;

б — вид сзади

Теперь вы знаете множество способов гибели жестких дисков, но необходимо рассмотреть еще одну ситуацию — дезинформирующий симптом, вследствие которого можно сделать ошибочный вывод о порче жесткого диска.

11.2. Ухудшение, не связанное с отказом, — это фрагментация файлов

При работе жесткий диск кажется удивительно быстрым, особенно при сравнении с накопителем на гибком диске. Если, однако, через несколько месяцев вы замечаете, что скорость его заметно уменьшилась, "преступником", по всей вероятности, является фрагментация файлов.

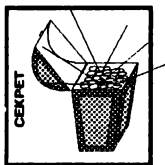
При записи файла на жесткий диск DOS вначале ищет в FAT свободные кластеры. Если файл занимает десять кластеров, DOS выделит ему первые десять кластеров на диске. Если следующему файлу нужно шесть кластеров, он получает следующие шесть и т.д. При выполнении только записи на диск новых файлов последующие файлы занимают соседние кластеры.

Конечно, большинство пользователей время от времени удаляют файлы или расширяют существующие файлы. Рассмотрим, как эти операции влияют на размещение файлов на диске.

Если после записи на диск нескольких файлов вы удаляете третий файл, на диске образуется "дыра". При записи на диск следующего файла, который больше третьего, DOS поместит часть нового файла в эту "дыру", а остаток — вдали от нее, фрагментируя файл на две части.

Существующий файл фрагментируется при добавлении в него данных, если нет достаточного места для добавляемых данных между первоначальным концом этого файла и началом следующего.

Когда DOS считывает файл с диска, она берет из каталога первый кластер файла. Затем она обращается к FAT для получения дополнительной информации о размещении файла. Если кластеры файлы находятся рядом, контроллер может считать их быстро. Чем больше разбросаны кластеры, тем дольше приходится считывать их.



В DOS 3.0 встроено специальное средство, которые должно помочь уменьшить фрагментацию файлов и перезаписать удаленных файлов. Этим средством является поддерживаемый DOS указатель, показывающий, где последний доступный кластер в FAT. Когда DOS требуется другой кластер, она начинает поиск в FAT сразу после этой точки. Если диск не заполнен, такой процесс позволяет DOS находить новый кластер быстрее. Кроме того, предотвращается помещение нового файла в "дыру", оставшуюся от удаленного файла.

Однако такой способ не предотвращает фрагментацию файла, возникающую при добавлении в существующий файл. Он также делает меньше ожидаемого по предотвращению фрагментации новых файлов. Поскольку DOS хранит указатель в памяти, он должен возвращаться к началу FAT при каждой перезагрузке компьютера. Несмотря на это средство DOS, через месяцы или годы работы появляется сильная фрагментация.

Через несколько месяцев работы большинство файлов данных на жестком диске и многие программные файлы оказываются фрагментированными. Этот процесс снижает эффективность диска, и со временем он начинает работать медленнее накопителей на гибком диске, хотя и не сигнализирует о начинающейся потере данных или неисправности. В отличие от проблемы нарушения выравнивания, независимо от степени фрагментации файлов к данным всегда можно обратиться, хотя и медленнее. Разумеется, интенсивное движение головок, требуемое для доступа к фрагментированным файлам, ускоряет гибель накопителя.

Кроме замедления вашей работы, фрагментация файлов затрудняет восстановление стертых файлов. После удаления файла теряется информация FAT о том, какие кластеры он занимал. Элемент каталога сообщает о том, сколько кластеров занимал файл и с какого кластера он начинался. Если файл не фрагментирован, можно восстановить его (при отсутствии перезаписи), просто отправляясь от указанного начального кластера и присоединяя нужное число кластеров. Но если файл фрагментирован, по такому способу найти нужные кластеры невозможно. Вы знаете число кластеров, но некоторые из них принадлежат другому файлу.

Серьезной проблемы такая ситуация не вызывает, особенно если вы пользуетесь системой слежения за удалением файлов или способом задержанного стирания. (Первый способ в DOS 5 или 6 называется Delete Tracker. Он следит за занятыми файлом кластерами при каждом удалении файла. Второй способ в DOS 6 называется Delete Sentry. В этом способе удаляемые файлы передаются в специальный каталог "мусорной корзины", откуда они не удаляются до истечения определенного числа дней или до занятия удаленными файлами определенной части пространства жесткого диска.)

Очевидное устранение снижающейся эффективности заключается в дефрагментации файлов. Сложный способ дефрагментации состоит в копировании файлов на

гибкие диски, стирании их с жесткого диска или форматировании раздела и последующего возвращения файлов на жесткий диск. Намного проще воспользоваться специальной программой дефрагментации файлов.

Такая программа имеется во всех указанных в разделе 11.1.4 дисковых утилитах. Очень хорошим дефрагментатором файлов является программа PC-Kwik Power Disk фирмы PC-Kwik Corporation; она особенно удобна для больших томов (более 200 Мбайт). Начиная с версии 6, в DOS включена программа DEFRAG — лицензированная версия программы Norton Speed Disk.

Программа DEFRAG не обладает всей гибкостью программы Norton Speed Disk. Она (рис. 11.7) предлагает только два способа оптимизации: Full optimization (Полная оптимизация) и Unfragment files only (Дефрагментация только файлов). В первом случае DEFRAG перемещает данные так, чтобы сделать все файлы смежными и сдвигает их как можно ближе к началу диска; о порядке файлов она не заботится. Во втором случае DEFRAG просто обеспечивает, чтобы каждый файл был смежным. Файлы могут остаться в любом месте, которое DEFRAG посчитала удобным, а между файлами может быть "разбросано" пустое пространство.

С другой стороны, программа Speed Disk предлагает множество опций. Помимо пяти различных способов оптимизации, она предлагает опцию определения "неподвижных" файлов и указания порядка размещения файлов. Более того, программа Speed Disk может показать карту файлов на диске для анализа степени фрагментации файлов.

Но программа DEFRAG может сделать то, чего не делают другие коммерческие программы дефрагментаторов. По крайней мере современные версии дефрагментаторов не могут работать с файлами уплотненного тома DoubleSpace (CVF — Compressed Volume Files), а программа DEFRAG работает. (Без сомнения, в новых версиях коммерческих программ такая возможность появится.)

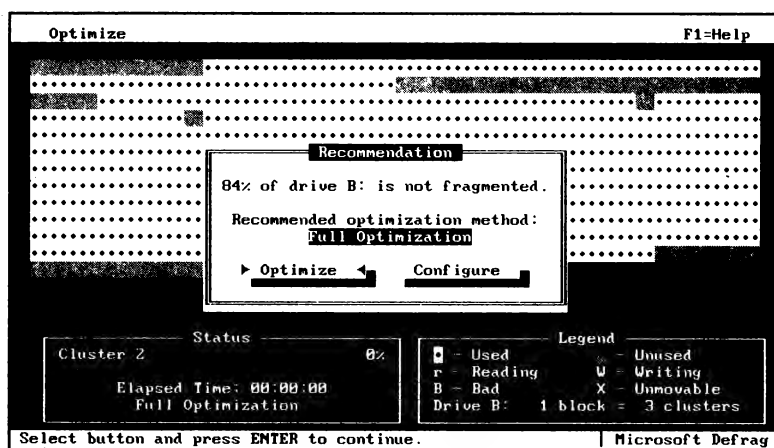


Рис. 11.7. Новая программа DEFRAG из MS DOS

При выполнении DEFRAG для неуплотненного тома выводится карта накопителя, показывающая, какие области заняты, а какие свободны. В ходе дефрагментации диска можно наблюдать, как программа считывает информацию из одной области, записывает в другую, а затем стирает ее с первоначальной области. Этот процесс повторяется до тех пор, пока не передвигаются все требуемые области.

Есть три способа выполнения DEFRAG на CVF. Во-первых, можно прямо вызвать DEFRAG и определить букву накопителя, которую DoubleSpace назначил этому

CVF. Во-вторых, можно выполнить DBLSPACE с опцией командной строки /DEFRAGMENT и определить букву накопителя. В-третьих, можно выполнить DBLSPACE.EXE в интерактивном режиме и выбрать опцию DEFRAG из меню после определения обрабатываемого диска.

В любом способе выполнения DEFRAG в том DoubleSpace карта диска не выводится. DEFRAG просто делает свою работу, а затем осуществляет выход.

DoubleSpace мало заботится о частой дефрагментации своих файлов CVF. Выполните CHKDSK на главном томе, затем выполните DBLSPACE/CHKDSK на уплотненном томе и, наконец, выполните DEFRAG на уплотненном томе. Произведение таких действий обеспечит максимальную производительность системы и сделает ее более стабильной (а ваши данные окажутся в большей безопасности).

11.3. Как исправить "больной" жесткий диск

Наверное, вы устали от изучения возможных вариантов гибели диска. Вам нужно просто знать, как же погиб ваш диск. Вы хотите соотнести вашу ситуацию с элементами списка симптомов и узнать, какова точная причина неисправности и как исправить ее.

Автору очень нравится такой подход. При подготовке этой главы он попытался создать универсальную диаграмму поиска неисправностей жесткого диска, но оказалось, что сделать это непросто. Автор полагает, что разработать диаграмму можно только для конкретных обстоятельств (например, конкретной марки BIOS, определенного типа жесткого диска и конкретной версии DOS).

11.3.1. В чем сложность задачи

Бескими причинами, объясняются трудности задачи, делающие ее практически неразрешимой. Для понимания этих причин рассмотрим возможные симптомы и их происхождение.

Прежде всего, есть чисто аппаратные симптомы — от очевидных до полностью неясных. Примером очевидного симптома служит пронзительный звук перед заеданием подшипника шпинделя. Пример второй крайности — невозможность выполнения любых операций (причиной может быть короткое замыкание шпиндельного двигателя и множество других). Где-то посередине находятся аппаратные неисправности, которые дают хотя бы туманные намеки на причину неисправности. К таким относятся многие неисправности контроллера.

Типы неисправностей жесткого диска во многом зависят от конкретной системы. Какой способ кодирования данных применяется в накопителе (MFM или RLL)? Какой стандартный интерфейс реализован в накопителе (ST412/ST506, IDE/ATA, ESDI, SCSI, PCMCIA или какой-то другой)? К какому контроллеру подключен накопитель (автоконфигурирующий или нет, с преобразованием секторов или нет и т.д.)? Естественно, выводимые сообщения об ошибках изменяются в зависимости от типа неисправности.

Но есть и чисто информационные неисправности. Они имеют другой набор симптомов, в основном в виде сообщений об ошибках, появляющихся на мониторе. Возможных источников таких сообщений очень много, и наиболее очевидные рассмотрены далее.

Некоторые сообщения об ошибках, являющиеся симптомами неисправности жесткого диска, выдает программа загрузки из ROM BIOS на материнской плате. Эти сообщения появляются только в процессе загрузки компьютера. Другая группа сообщений поступает от программ расширений ROM BIOS на съемной плате контроллера жесткого диска. Такие сообщения могут появиться при загрузке, а также в любой момент при обращении к данным.

Какая марка микросхемы BIOS имеется в компьютере? Каждый производитель BIOS использует свои сообщения об ошибках. На них влияет также версия BIOS.

Часто BIOS модернизируется производителем материнской платы и в ней появляются новые сообщения.

Программу главной загрузочной записи MBR помещает на жесткий диск программа инициализации диска определенной операционной системы. (В DOS такой программой является FDISK.) Какая марка и версия этой программы применялась для размещения MBR на ваш жесткий диск? Знаете ли вы об этом? Точная формулировка сообщений об ошибках зависит от версии программы.

Программу загрузочной записи DOS помещает в каждый том DOS программа FORMAT. В этой операции вы могли использовать другую версию DOS. Например, при переходе от версии 4.0 к 5.0, вы, по-видимому, не переделывали организацию разделов. В результате у вас есть главная загрузочная запись, созданная DOS 4.0, и загрузочная запись в каждом томе DOS, созданная DOS 5.0. И вновь формулировка и смысл сообщений об ошибках зависят от версии, используемой для инсталлирования этих, очень критичных при обращении к диску, программ.

Кроме системных программ, сообщающих о проблемах при обращении к диску, многие драйверы устройств, резидентные программы и прикладные программы имеют свои наборы сообщений об ошибках, относящихся к проблемам обращения к файлам или диску.

А знаете ли вы о разнообразии выпускаемых ПК и что это разнообразие приводит к множеству других наборов сообщений об ошибках? Таким образом, мы приходим к выводу о том, что невозможно создать диаграмму, позволяющую однозначно идентифицировать источник ваших трудностей просто по исходным симптомам. Конечно, какие-то хорошие намеки вы получите, но, по всей вероятности, потребуются дальнейшие эксперименты.

11.3.2. Когда можно создать диаграмму поиска неисправностей

В некоторых ситуациях все же имеет смысл подготовить подробную блок-схему, которая демонстрирует всевозможные ошибочные условия и предпринимаемые действия.

Когда производитель компонует всю систему, включая все аппаратные и программные части, он знает о всех возможных неисправностях и возникающих при этом симптомах. Для каждого случая производитель может рекомендовать, что необходимо делать.

Если вы работаете на компьютере, произведенном фирмой IBM, и применяете программы только этой фирмы (что в общем нереально), поиск неисправности упрощают фирменные диагностические средства. Фирма IBM публикует многотомные руководства по процедурам эксплуатации и ремонта. В них содержится много блок-схем, помогающих в процессе поиска неисправности и охватывающих почти все случаи.

При таком подходе (помимо невозможности работать только с программами фирмы IBM) есть один недостаток, — такие руководства довольно дорогие. А если ваш ПК не "чисто IBM-овский", то почти наверняка у вас столкнетесь с проблемами, при решении которых фирменные руководства не помогают. Приобретение руководств оправдано только при наличии большого числа компьютеров фирмы IBM.

11.3.3. Подход, который может помочь

В общем, при возникновении трудностей постарайтесь максимально упростить ситуацию, чтобы лучше разобраться в происходящем. При работе с жесткими дисками такое упрощение предполагает несколько действий. Попробуйте идентифицировать источник каждого сообщения об ошибке. Для этого придется посмотреть, какие сообщения содержатся в различных ПЗУ и ключевых программах на диске.

Зная диапазон адресов ПЗУ, посмотреть ее содержимое можно программой DEBUG. (После запуска DEBUG напечатайте команду `d segment:offset` для вывода

128 байт памяти, начиная с указанного адреса. Для вывода следующих 128 байт вводится команда `d` без аргументов. Продолжайте эти действия до просмотра всего содержимого ПЗУ.)

Для выяснения причины неисправности попробуйте загрузиться с безопасной загрузочной дискеты. Или переименуйте файлы `CONFIG.SYS` и `AUTOEXEC.BAT` на жестком диске, а затем проведите загрузку. При наличии DOS 6 имеется третий вариант: нажмите клавишу F5 при появлении сообщения `Starting MS DOS`. После этого DOS игнорирует файлы `CONFIG.SYS` и `AUTOEXEC.BAT`. (Для этих файлов автор применяет временные имена `CF` и `AE`.) При загрузке с дискеты или с жесткого диска без обычных файлов `CONFIG.SYS` и `AUTOEXEC.BAT` вы увидите, как ПК работает в простейшей конфигурации. Узнав, исчезает при этом неисправность или нет, вы определите, вызвана она программной средой, создаваемой в ПК, или аппаратными средствами и системными программами, которые служат основой этой среды.

При загрузке ПК наблюдайте за экраном. Разберитесь в смысле каждого увиденного сообщения и контексте его появления. Если, например, при обработке файла `CONFIG.SYS` все происходит нормально, то процесс загрузки, в основном, заканчивается успешно. Любая из оставшихся проблем, по-видимому, вызвана программной средой.

Сообщение типа `Missing operating system` (Отсутствует операционная система) выдает главная загрузочная запись. Оно вызывается несколькими причинами, включая неправильный элемент КМОП-ЗУПВ для типа первого жесткого диска, измененные данные таблицы разделов в MBR или отсутствующую загрузочную запись в первичном разделе DOS. Ясно, что это сообщение не могут выдать программы, загружаемые после самой DOS. Оно не показывает и аппаратной неисправности (которая препятствовала бы загрузке и выполнению программы главной загрузочной записи).

Если вы не видите ничего, показывающего, что ПК загружается хотя бы частично, проведите перезагрузку. Теперь внимательно наблюдайте, включается ли индикатор активности жесткого диска. Если он не включается, то (полагая, что соединение индикатора не нарушено) контроллер никогда не приказывает накопителю сделать что-либо. А когда вам не приказано чего-то делать, вы послушно ничего не делаете. Такая ситуация явно показывает аппаратную неисправность.



Светодиод, показывающий активность жесткого диска, находится либо на самом диске (обычно в компьютере XT) или на лицевой панели системного блока (в компьютерах модели AT и выше). Если ваш индикатор на лицевой панели, он соединен с материнской платой или съемной платой контроллера двухжильным ленточным кабелем. Разъем, в который он вставляется, имеет четыре контакта. Есть два правильных способа подключения (и четыре неправильных). Если вы замечаете, что индикатор не включается, а жесткий диск вроде бы работает, возможно, неправильное соединение. Откройте системный блок, соблюдая обычные меры предосторожности, и посмотрите состояние кабеля. Возможно, он не вставлен в разъем. Вставьте его сначала одним способом, а затем другим до правильной работы индикатора. Запишите его правильное соединение, чтобы в следующий раз действовать быстрее.

Разобравшись, где возникла неисправность, ее проще найти. В случае аппаратной неисправности заменяйте компоненты до загрузки с безопасной загрузочной дискеты и обращения к жесткому диску. Действуя таким образом, можно решить почти любую аппаратную проблему. При получении сообщений от главной загрузочной записи или программы загрузочного сектора DOS направьте усилия на исправление таблиц данных и программ, которые эти программы используют или загружают.

Успех в устранении проблем по способу "от симптомов к причинам" зависит от отчетливого понимания функционирования всех частей системы. Именно такому подходу посвящена данная книга. От вас требуются трудная работа и напряженные раздумья, что, конечно, сложнее просмотра определенной процедуры ремонта по блок-схеме. Но зато этот подход действует всегда, независимо от комбинации марок тех компонентов, из которых состоит ваш ПК.

11.4. Восстановление данных — надежда и реальность

Всякий, потерявший данные, хочет их восстановить. Иногда (но не всегда) можно "воскресить" потерянные на жестком диске данные.

Многое зависит от того, как пропали данные. Если с данными произошло небольшое несчастье, их восстановить довольно просто. Например, проверка соединений кабелей или вынимание и вставка платы контроллера в слот материнской платы ПК занимает всего несколько минут. Но если при крахе головки повреждено магнитное покрытие, данные с этого участка покрытия считать невозможно. Где-то посередине находится ситуация с заеданием подшипника шпинделя. Все файлы находятся на дисках, но вы не можете считать их. Человек, имеющий необходимое оборудование (и подходящую "чистую" комнату), может добиться работы накопителя и обеспечить доступ к данным. Такой вариант возможен, но он слишком дорог, часто дороже самого накопителя. Он показывает, что возможность не всегда означает осуществимость. Осуществимость зависит от ценности данных и "старости" резервных копий. Услуги по восстановлению данных оказывают фирма Workman and Associates и OnTrack (разработчик программы Disk Manager).

Гораздо проще потерять данные, ошибочно стирая файлы. Можно восстановить их после стирания, но эта задача оказывается слишком трудной и сложной. Некоторые утилиты, используемые заранее, делают восстановление файлов после стирания более осуществимым. К ним относятся программа IMAGE из утилит Нортон и программа MIRROR из пакета ПК Tools. Версия программы MIRROR включена в MS DOS 5 и ПК DOS 5. Такие же возможности обеспечивает программа UNDELETE в DOS 6.

Еще более сложная ситуация возникает при перезаписи удаленных файлов. Здесь простая программа восстановления после стирания не помогает, и вам придется обращаться к специализированным (и очень дорогим) услугам. Это будет еще одним сильным аргументом для регулярного резервирования.

11.5. Все о резервировании

Каждый, работающий на ПК, наверняка слышал о резервировании, но не всякий понимает, что оно в действительности означает. Еще меньше людей знают о способах создания и сопровождения резервов, а также средствах, помогающих в этом процессе. Материал этого раздела поможет вам присоединиться к хорошо информированным пользователям ПК.

11.5.1. Три вида резервирования

Есть три принципиально разных подхода к созданию резервных копий ценных данных — полное (или общее) резервирование, инкрементное резервирование и дифференциальное резервирование.

Полное резервирование. Полное резервирование означает именно это — копирование всей информации с жесткого диска на некоторый носитель для безопасного хранения. Для регулярного создания полных резервных копий многие применяют ленточный накопитель. Это хороший подход к резервированию, но его можно улучшить.

Например, довольно просто вставить кассету в накопитель и выполнить пакетный файл в конце каждой пятницы. Однако при таком подходе нет защиты от потери вашей работы в течение той недели, когда погибает жесткий диск. Кроме того, придется использовать ленты повторно (а это ведет к некоторой уязвимости), иначе вам вскоре не хватит места для их хранения и денег для их покупки.

Инкрементное резервирование. Большая часть информации на жестком диске не часто изменяется. Большинство программ не изменяются совсем. Многие файлы данных изменяются редко. На диске могут храниться документы, созданные много месяцев назад. Они нужны вам для справок, но вы не намерены изменять их. Так зачем же резервировать их каждую неделю или по другому расписанию полного резервирования? Конечно, этого делать не нужно.

Требуется образовать копии всех новых или измененных файлов. К счастью, DOS предоставляет простые средства поиска таких файлов.

Всякий раз, когда программа просит DOS открыть файл и дать разрешение на считывание и запись информации в этот файл, DOS устанавливает для файла флажок, показывающий, что это сделано. Флажком служит один бит в байте атрибутов, хранимом в элементе каталога для каждого файла. Этот бит называется архивным (см. рис. 10.4).

Примечание. DOS устанавливает флажок при открытии файла для программы, которой разрешено произведение изменений. DOS не знает, проводились ли фактические изменения.

Многие программы резервирования привлекают этот бит при выборе файлов для резервирования. Программам DOS BACKUP и XCOPY можно также сообщить об использовании этого бита. Более того, эти программы могут либо сбрасывать флажок после резервирования файла, либо оставлять его неизменным. Вариант выбирается пользователем при вызове программы.

До версии 6 команды DOS BACKUP и RESTORE критиковались за их простые функции, а одобрения заслужили программы третьих фирм, например FASTBACK и Norton Backup. В MS DOS 6 включена лицензионная версия программы Norton Backup под названием MSBACKUP.

Простой план инкрементного резервирования заключается в выполнении одной из таких программ и копировании на резервный носитель все файлов, у которых включен архивный бит; после этого биты сбрасываются.

Образовать два набора резервов почти также просто. Сначала образуйте набор, скомандовав программе использовать архивные биты для выбора файлов, но не изменять состояния этих битов. Затем для образования второго набора повторно выполните программу, но теперь сообщите ей о необходимости сброса архивного бита.

Это очень хороший способ образования резервов. Недостаток его в том, что резервируется много файлов, которые программы открывали только для просмотра содержимого файлов, но не изменяли его. В случае больших файлов базы данных у вас вскоре появятся многократные копии одних и тех же данных.

В простом инкрементном резервировании образуются копии временных и резервных (BAK) файлов, которые появились на жестком диске. Некоторые программы резервирования позволяют скрыть эти файлы, хотя они и ограничены тем, насколько точные инструкции им даются.

По-видимому, оптимальное инкрементное резервирование предполагает активное участие пользователя в процессе резервирования. Например, программа XtreeGold показывает список всех файлов с установленным архивным битом, а вы выбираете,

какие из них резервировать и какие игнорировать. Для последней группы программа Xtree может выключить архивные биты, а для резервирования остальных файлов можно воспользоваться этой или другой программой.

Главное достоинство инкрементного резервирования состоит в том, что в процессе резервирования участвует минимум байтов. Благодаря сокращению продолжительности резервирования можно проводить чаще. Кроме того, для хранения резервов требуется меньше дисков и лент.

Дифференциальное резервирование. Третий подход к созданию резервных копий называется дифференциальным резервированием. Его идея состоит в том, чтобы время от времени проводить полное резервирование. Затем при новом резервировании копируются только те файлы, которые изменились после последнего полного резервирования.

Дифференциальное резервирование находится где-то посередине между рассмотренными ранее подходами. Когда жесткий диск гибнет, при наличии достаточно свежего полного резерва вы просто возвращаете его в новый накопитель. Если вы проводите полное резервирование, дополненное частыми инкрементными резервами, при этом также обеспечивается полное восстановление, но с дополнительными действиями. Вначале нужно вернуть последний полный резерв, а затем по порядку возвращать промежуточные инкрементные резервы. При этом вам придется удалить все файлы, которые вы удалили, чтобы оставшиеся на жестком диске файлы были точной копией того, что вы имели в день создания каждого инкрементного резерва.

В дифференциальном подходе сочетаются простота полного резервирования и частичная экономия инкрементного резервирования. При каждом создании резерва сохраняются все файлы, которые изменились после последнего полного резервирования, а не просто последнего резервирования. При возвращении резервов на новый жесткий диск нужно вернуть только последний полный резерв и один (последний) инкрементный резерв.

11.5.2. Как сделать способ резервирования хорошим?

Чтобы способ резервирования стал хорошим, прежде всего требуется регулярность. Почти каждый подход хорош, если вы используете его, но все же некоторые лучше других.

11.5.3. Частота резервирования

Как часто проводить резервирование? Вот очень простое правило: если усилия на восстановление данных, потерянных при выходе компьютера из строя в следующую минуту, превышают усилия на образование резервных копий, осуществите резервирование.

Для некоторых людей это означает проведение инкрементного резервирования каждый день, а для других — каждый час. Полезно изредка делать полное резервирование, чтобы проще и быстрее возвращать файлы, но можно делать полное резервирование намного реже и все же оставаться полностью спокойным. Целесообразно хранить запасной набор инкрементного и полного резерва в другом здании, что защитит вас от потери резервных данных. Можно создавать второй набор резервных копий реже основного набора, так как опасность воровства, пожара или наводнения менее вероятна, чем искажение данных.

11.5.4. Хороший способ резервирования

Как же определить хороший способ резервирования? Для большинства учреждений подходит комбинация полного и инкрементного резервирования. Можно проводить полное резервирование раз в квартал или полгода, храня запасной набор в другом месте. Для дополнения полного резерва следует проводить инкрементное резервирование примерно раз в день.

После заполнения инкрементных диска или ленты, отметьте их и сделайте только считываемыми. Ради быстрого доступа можно хранить инкрементные резервы поближе к ПК, а для большей безопасности их следует запереть. После заполнения носителя используйте его только при необходимости возвращения файлов. Наконец, для резервирования удобно применять накопители WORM (Write Once, Read Mostly — Однократная запись и многократное считывание). В таких накопителях невозможно перезаписать диски, и они обеспечивают постоянные неизменяемые копии ваших файлов на каждой ступени их эволюции.

11.5.6. Ситуация, в которой требуется другой подход

Резервные копии превращаются в архив всей вашей прошлой работы, а это придает им большую ценность и служит еще одной причиной аккуратного создания и хранения их. Редкое полное резервирование и очень частое инкрементное резервирование обеспечивают безопасность за счет небольших усилий при самом резервировании. Возвращение резервной информации несколько сложнее, но это не так плохо, так как отказы жесткого диска возникают намного чаще, чем образование резервных копий.

Но есть одна ситуация, в которой этот подход не годится, — огромный файл базы данных, к которому каждый день добавляются несколько байтов. Если резервировать весь файл каждый день, вам придется расходовать почти столько же времени и носителя, как и на ежедневное полное резервирование. С другой стороны, если подожать и проводить резервирование только раз в неделю, появляется опасность потерять недельную работу.

В такой ситуации наиболее подходящим оказывается дифференциальное резервирование. Можно образовать несколько дифференциальных резервных копий и повторно использовать их; в этом случае при порче жесткого диска вы зависите от последней копии (и последнего полного резервирования).

Для такой ситуации имеется даже лучший подход. Он требует некоторой подготовки, но позволяет проводить резервирование намного чаще (уменьшая этим уязвимость при потере данных). Он позволяет также иметь точную картину того, как выглядят изменяющиеся файлы при создании резерва, и при этом не требует больше времени или носителя по сравнению с резервированием небольших файлов.

Ключ такого подхода — применение программы, которая анализирует копию огромного файла базы данных от последнего инкрементного резервирования и сравнивает ее с рабочей копией файла. Из этих двух файлов программа образует файл разностей. При необходимости восстановления последней базы данных вы просто "прилагаете" этот файл к копии базы данных, какой она была (применяя программу, которая создавала файл разностей), и в результате получается последняя копия файла базы данных.

Прилагая каждый такой файл по очереди к копии первоначальной базы данных (она хранится в последнем полном резерве), вы можете точно восстановить базу данных из полного резерва, образованного много дней назад. Для этого придется пройти много этапов — так, как если бы вы восстанавливали из полного резерва и множества инкрементных резервов.

Среди подходящих для этого подхода программ отметим DeltaFile фирмы Hyperkinetix и программу RT-Patch фирмы PocketSof. Программа DeltaFile образует разностный файл с помощью встроенной "восстановительной машины". Программа PC Patch сохраняет в каждом образуемом файле данных только информацию о различиях.

11.5.7. Еще одно применение хороших резервов

Большинство людей создают резервные копии для защиты данных при отказе диска, но резервные копии полезны и по другим причинам. Если каталогизировать резервные диски или ленты и хранить их всегда, вы сможете быстро найти копию файла спустя много времени после его удаления с жесткого диска. Иногда это будет важнее восстановления всего содержимого жесткого диска после его гибели. Хорошей программой каталогизации диска является бесплатная программа CATDISK, которую можно получить через доску объявлений.

11.6. Заменяет ли резервные копии зеркальный диск?

Уже много лет в больших вычислительных системах предусматриваются избыточные дисковые накопители. Ими управляет специальный контроллер, поддерживающий зеркальный накопитель. Все данные одновременно записываются в два накопителя. В случае гибели одного из накопителей контроллер считывает и записывает данные на другом накопителе. Лучшие системы с зеркальным диском предоставляют пользователю возможность удалить и заменить неисправный накопитель, не прерывая работы компьютера. После установки нового накопителя контроллер автоматически передает на него копии всех данных с другого накопителя.

Такой вариант позволяет реализовать в компьютере подсистему жесткого диска с бесконечным средним временем между отказами (Mean Time Between Failure — MTBF). Подсистема выглядит как совершенство. Конечно, в случае гибели накопителя для замены требуется вмешательство человека.

С недавних пор на рынке стали появляться "зеркальные" контроллеры для ПК. Значит ли это, что, приобретя один из таких контроллеров и дополнительный накопитель, вы сможете забыть о всех хлопотах резервирования? Нет, это не так! Вам по-прежнему придется регулярно создавать резервные копии и хранить их отдельно от компьютера для защиты от таких катастроф, как сильный пожар, которые могут полностью уничтожить вашу систему.

"Зеркальные" системы позволяют проводить резервирование, когда компьютер занят другими задачами: резервирование осуществляется с одного накопителя, который отключается от контроллера, а компьютер работает с другим накопителем. Закончив резервирование, можно подключить накопитель и заставить контроллер обновить его. Таким образом, резервирование оказывается удобнее и меньше мешает работе. Повышенная надежность и удобство резервирования делает такой способ подходящим для критических применений.

11.7. Предупреждения и рекомендации

- ✓ Не впадайте в заблуждение: все жесткие диски погибают и вопрос только в том, когда это произойдет. (Все накопители на гибких дисках тоже гибнут, поэтому не надейтесь на абсолютную надежность хранимых на них данных).
- ✓ Храните запасные резервные копии в другом помещении. Это защитит вас от потери данных основного набора резервов (для удобства их можно хранить вблизи компьютера) и потери компьютера, например в результате воровства или пожара.
- ✓ Периодически проверяйте свой способ резервирования. Способ приемлем только в том случае, если он позволяет возвратить на диск каждый зарезервированный файл без ошибки даже в одном байте.
- ✓ Избегайте паники. Не разрешайте себе в панике действовать по принципу "делать что-нибудь — все равно, что — только, чтобы сейчас что-то делать!" Критический случай должен настроить вас предпринять такие действия, которые защитят от потери данных в будущем.
- ✓ Помните совет из главы 1 о том, как не повредить ПК при работе внутри системного блока. Перед этим обязательно выключайте питание, отсоедините все периферийные устройства и выньте шнур из блока питания. Коснитесь рукой блока питания или рамы ПК до произведения любых действий внутри блока.
- ✓ Перечитайте материал главы 1. Обратите особое внимание на резервирование и создание безопасной загрузочной дискеты. Зная теперь многочисленные способы гибели жесткого диска, вы с большим усердием предпримете упреждающие меры.

Заключение

- В большинстве случаев жесткий диск не выходит из строя навсегда. Часто можно восстановить данные и даже привести накопитель в работоспособное состояние.
- При неисправности дисковой подсистемы прежде всего нужно определить, чем она вызвана: аппаратным отказом или искажением информации, используемой для обращения к накопителю.
- Некоторые дисковые ошибки вызываются нарушением выравнивания головок и дорожек с записанной информацией. Выполнение такой программы, как SpinRite, часто полностью устраняет такие проблемы.
- Несчастья с данными часто происходят из-за неправильного обращения с ПК (например, перезагрузка ПК в произвольный момент времени, а не на стимул DOS, или включение сразу после выключения). Конечно, такие проблемы возникают и при выбросах питания. Они ведут к различным последствиям, начиная от потери нескольких кластеров до полной невозможности доступа к накопителю.
- Для диагностики и ремонта большое значение имеет понимание работы жесткого диска. Почти все проблемы с данными можно исправить. Можно исправить и многие (но не все) аппаратные проблемы.
- Фрагментация файлов является естественным следствием продолжительной работы на ПК и приводит к существенному ухудшению производительности. Дефрагментация жесткого диска повышает видимое быстродействие ПК и затрудняет восстановление файлов.
- Исправление неисправностей диска для восстановления данных намного сложнее простого возвращения файлов с последней резервной копии.
- Есть несколько способов восстановления. Единственное важное решение — выбрать один из способов и практически применять его.

- Тщательно проверьте выбранный способ резервирования и почаще перепроверяйте его. Резервирование бесполезно, если вы не сможете безошибочно восстановить требуемые файлы.

Теперь вы знаете все о том, как действуют жесткие диски ПК, как они погибают и как работать с ними.

Содержание

Предисловие к русскому изданию	4
Введение	5
Кому предназначена книга	5
Те, кому особенно нужна книга	5
Те, кто хочет знать все	5
Любознательные читатели	5
Кто не имеет IBM PC или не работает с DOS	6
Что нужно знать предварительно	6
Предположения автора	6
Нужно знать, как открыть компьютер	7
Что делать, если вы чего-то не знаете	7
Как лучше всего пользоваться книгой	8
Три способа первого чтения книги	8
Принятые в книге соглашения	8
Приглашение к сотрудничеству	9
 Глава 1. Значение жестких дисков	10
1.1. Жесткий диск — это не просто одно из периферийных устройств	10
1.2. Жесткий диск — это самое дорогое устройство в ПК	10
1.2.1. Жесткие диски портятся чаще остальных компонентов	10
1.2.2. Самое ценное на дисках	11
1.3. Погибший жесткий диск можно исправить — иногда легко	11
1.4. И небольшие знания помогут в беде	12
1.4.1. Небрежность плоха, но паника еще хуже	12
1.4.2. Не выплескивайте с водой ребенка	12
1.4.3. Великое дело — помощь	13
1.5. Данные нельзя застраховать — их нужно защищать самому	14
1.5.1. Все диски портятся — никто не знает, когда это случится	14
1.5.2. Будьте готовыми — имейте надежный резерв	14
1.5.3. Способы защиты от ошибок	15
1.6. Как заставить жесткий диск работать лучше и безопаснее	16
1.6.1. Хорошая организация помогает вам и DOS	16
1.6.2. Помогает и ускорение диска	16
1.7. Некоторые предложения для восстановления	17
1.7.1. Готовьтесь — создайте безопасную загрузочную дискету	17
1.7.2. Держите другие полезные дискеты под рукой	21
1.7.3. Создание безопасной загрузочной дискеты после гибели жесткого диска	21

Глава 2. Магнитное хранение информации.....	24
2.1. Физические принципы	24
2.1.1. Три важных физических факта.....	25
2.1.2. Базовая конструкция жесткого диска	26
2.1.3. Важность чистоты.....	29
2.1.4. Запись и считывание информации.....	29
2.1.5. Размер бита	31
2.2. Способы кодирования данных	36
2.2.1. Простой импульсный подход.....	37
2.2.2. Частотная модуляция	39
2.2.3. Кодирование без возврата к нулю.....	40
2.2.4. Модифицированная частотная модуляция.....	41
2.2.5. RLL-кодирование.....	41
2.2.6. Улучшенное ARLL-кодирование.....	44
2.3. Проблема несовершенства и ее решение.....	44
2.3.1. Частичное решение — обнаружение ошибок.....	45
2.3.2. Лучшее решение — исправление ошибок	45
Глава 3. Некоторые важные технические вопросы	50
3.1. Повышение производительности жесткого диска с помощью чередования секторов и перекоса головок.....	50
3.1.1. Анатомия дискового сектора.....	50
3.1.2. Чередование секторов.....	51
3.1.3. Перекос головки	57
3.1.4. Где хранится информация о чередовании секторов и перекосе головки.....	59
3.1.5. Как изменить чередование и перекос.....	59
3.1.6. Когда и как можно установить неправильное чередование.....	59
3.1.7. Несовместимые значения чередования.....	60
3.2. Популярные изделия и стандарты.....	60
3.2.1. Интерфейс ST412/ST506	62
3.2.2. Интерфейс ESDI.....	63
3.2.3. Интерфейс SCSI	64
3.2.4. Интерфейс IDE	67
3.2.5. Жесткие диски на съемной плате	68
3.2.6. Интерфейс PCMCIA.....	68
3.3. Два вида архитектуры накопителя.....	70
3.3.1. Головки, цилиндры и секторы	71
3.3.2. Логические структуры информации	71
Глава 4. Как устроены накопители на жестких дисках	73
4.1. Физические элементы накопителя.....	73
4.1.1. Привод с шаговым двигателем	73
4.1.2. Преимущества двигателей со звуковой катушкой	80
4.1.3. Позиционирование головки	82
4.1.4. Заголовок сектора.....	93
4.1.5. Конструкции головок считывания-записи	93

4.2. Продольная запись	94
4.3. Вертикальная запись	100
4.4. Что происходит в электронике накопителя	104
4.4.1. Схема управления шпиндельным двигателем	104
4.4.2. Схема управления позиционированием головок	105
4.4.3. Коммутаторы и усилители	106
4.4.4. Интерфейсная электроника	107
4.5. Что происходит в электронике контроллера	107
4.6. Рекалибровка и странные ночные звуки	108
Глава 5. Кэширование диска	111
5.1. Назначение дискового кэша	111
5.2. Виды кэша	113
5.2.1. Буфер дорожки	113
5.2.2. Кэш считывания	113
5.2.3. Кэш со сквозной записью	114
5.2.4. Улучшенный кэш со сквозной записью	115
5.2.5. Кэш с отложенной записью	115
5.2.6. Элеваторный кэш	117
5.3. Выбор опций для программы кэширования	117
5.4. Различные организации кэша	117
5.4.1. Полностью ассоциативный кэш	117
5.4.2. Кэш с прямым отображением	118
5.4.3. Ассоциативный по множеству кэш	118
5.4.4. Как узнать тип кэша	118
5.5. Буферирующие и кэширующие контроллеры	119
5.5.1. Программа кэширования или аппаратный кэш	119
5.6. Дисковые буферы DOS	120
5.7. Выбор количества буферов в операторе BUFFERS	121
Глава 6. Жесткие диски в DOS	123
6.1. Введение в DOS	123
6.2. Взаимосвязь между жесткими дисками и DOS	124
6.3. Блочные и символьные устройства DOS	125
6.4. Действия DOS	128
6.4.1. Уровни программ в ПК	128
6.4.2. Прерывания	130
6.4.3. Резидентные программы	135
6.4.4. Драйверы устройств	137
6.4.5. Таблицы параметров жесткого диска	138
6.5. Определение типа накопителя	142
6.5.1. Первый ХТ и аналогичные платы старых контроллеров	142

6.5.2. Платы новых автоконфигурирующих контроллеров XT	142
6.5.3. Компьютеры AT и старшие модели	143
6.5.4. Таблица типа накопителя AT	144
6.5.5. Жесткие диски на съемных платах	145
6.5.6. ESDI-накопители	145
6.5.7. Накопители IDE/ATA	146
6.5.8. SCSI-накопители	147
Глава 7. Преодоление барьеров	153
7.1. Логические ограничения DOS на емкость диска	153
7.2. Физические ограничения BIOS на емкость диска	154
7.3. Преодоление логических ограничений	155
7.3.1. Большие секторы	155
7.3.2. Дисковые разделы	156
7.4. Преодоление физических ограничений	157
7.4.1. Чисто программный способ	157
7.4.2. Отображение секторов	158
7.4.3. Другие способы отображения в контроллере	160
7.5. RAID и другие способы	161
7.6. Сообщения об ошибках	162
Глава 8. Подготовка жесткого диска к хранению информации.	
Этап 1 — физическое форматирование	167
8.1. Что такое форматирование диска?	167
8.2. Трехэтапный процесс (и его аналогия)	167
8.3. Три способа физического форматирования	169
8.3.1. Только контроллер проводит форматирование	170
8.3.2. Некоторые нестандартные форматы	171
Глава 9. Подготовка жесткого диска к хранению информации.	
Этап 2 — организация разделов	175
9.1. Зачем на жестком диске нужны разделы	175
9.2. Разделение диска	176
9.2.1. Главная загрузочная запись и таблица разделов	176
9.2.2. Диагностический цилиндр	181
9.2.3. Секретный цилиндр	181
9.3. Где хранятся файлы	181
9.3.1. Активный или загрузочный раздел	181
9.3.2. Первичный раздел DOS	181
9.3.3. Расширенный раздел DOS	182
9.3.4. Другие виды разделов	184
9.3.5. Не-DOS операционные системы, которые могут использовать первичный и расширенный разделы DOS	184

9.4. Таблица разделов	184
9.4.1. Этапы организации разделов на новом диске	185
9.4.2. Программы организации разделов третьих фирм	187
9.4.3. Рекомендация	188
9.5. Предложение (только для смелых читателей!)	188
9.6. Еще одна рекомендация	189
Глава 10. Подготовка жесткого диска к хранению информации.	
Этап 3 — логическое форматирование	192
10.1. Что такое логическое форматирование?	192
10.2. Логическая организация раздела DOS	192
10.2.1. Загрузочная запись DOS	192
10.2.2. Область данных	194
10.2.3. Таблица размещения файлов	197
10.2.4. Корневой каталог	200
10.2.5. Подкаталоги	201
10.2.6. Метка тома	202
10.2.7. Вывод по структурным частям	202
10.2.8. Как кластеры "теряются"	204
10.2.9. Другие способы повреждения данных	204
10.3. Другие операции программы FORMAT	205
10.3.1. Инициализация диска	205
10.3.2. Простой тест диска	205
10.3.3. Интерпретация таблицы данных загрузочной записи DOS	205
10.3.4. Отличие форматирования гибких дисков	207
10.4. Альтернативы форматированию высокого уровня DOS	207
10.4.1. Система UNIX	209
10.4.2. OS/2 и HPFS	209
10.4.3. Windows NT и NTFS	210
10.4.4. Система NetWare	210
10.5. Как найти дефектные участки	211
Глава 11. Как умирают жесткие диски	213
11.1. Виды неисправностей	213
11.1.1. Как различать аппаратные и информационные неисправности	216
11.1.2. Простые неисправности	216
11.1.3. Медленная порча из-за нарушения выравнивания	218
11.1.4. Сбой данных	220
11.1.5. Аппаратная неисправность в накопителе	224
11.1.6. Отказ контроллера	227
11.2. Ухудшение, не связанное с отказом, — это фрагментация файлов	229
11.3. Как исправить "больной" жесткий диск	231
11.3.1. В чем сложность задачи	231
11.3.2. Когда можно создать диаграмму поиска неисправностей	232

11.3.3. Подход, который может помочь.....	233
11.4. Восстановление данных — надежда и реальность.....	234
11.5. Все о резервировании.....	235
11.5.1. Три вида резервирования.....	235
11.5.2. Как сделать способ резервирования хорошим	236
11.5.3. Частота резервирования.....	237
11.5.4. Хороший способ резервирования	237
11.5.6. Ситуация, в которой требуется другой подход	237
11.5.7. Еще одно применение хороших резервов.....	238
11.6. Заменяет ли резервные копии зеркальный диск.....	238
11.7. Предупреждения и рекомендации.....	239
Содержание	246

Новые книги

Издательство "Диалектика" совместно с "Информейшн Компьютер Энтерпрайз" (Москва) предлагает вниманию читателей новую



серию компьютерных книг
от "Диалектики" и ICE

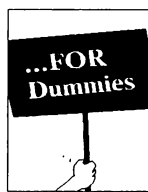
«... для "чайников"»



TM

СЕРИЯ КОМПЬЮТЕРНЫХ
КНИГ ОТ "ДИАЛЕКТИКИ"
И ICE

Вы побаиваетесь компьютера? Вы находите, что традиционные руководства перегружены техническими деталями, которые вам никогда не понадобятся? Обращаются ли к вам родные или приятели с просьбой разобраться с какими-то мелочами, связанными с работой на компьютере? Тогда серия компьютерных книг от "Диалектики" и ICE «... для "чайников"» специально для вас.



TM

COMPUTER
BOOK SERIES
FROM IDG

Книги серии «... для "чайников"» пригодятся даже тем раздраженным пользователям, которые на самом деле далеко не глупы, но запутались в компьютерной терминологии и чувствуют себя беспомощными, разбираясь в программном обеспечении для персональных компьютеров. Книги серии «... для "чайников"» написаны в легком шутиливом тоне, даже смешные рисунки и забавные пиктограммы специально предназначены для того, чтобы развеять страх компьютерных новичков и добавить им уверенности в собственных силах. Шутиливые, но не легкомысленные, эти книги служат идеальным руководством для любого, стремящегося освоить компьютер.

Если вы ищете самый легкий и веселый путь изучения компьютеров, загляните в предлагаемую серию книг «... для "чайников"», и вам обеспечена помощь и доступность!

Вашему вниманию предлагаются книги:

Энди Ратбон. Windows 3.1 для "чайников"

256 с., мягкий переплет.

Наилучшая книга о Windows 3.1 для начинающих пользователей. Ее второе издание знакомит читателя с основными концепциями и понятиями замечательной графической среды. Прочтя ее, вы не станете кудесником в Windows, но приобретете ценные знания. Они достанутся вам быстро, просто и с минимальными затратами, не отвлекая вас от гораздо более приятных вещей в нашей сложной жизни.

Ориентировочный выход в свет — сентябрь 1994 г.

Грег Харвей. Excel 5.0 для "чайников"

Мягкий переплет.

В этой книге вы найдете всю информацию, которую необходимо помнить, когда вы собираетесь выполнять ежедневные задачи, которые обычно решаются с помощью Excel. Наши намерения — преподнести понятия просто и не наскучить вам множеством технических деталей, в которых нет нужды. Освещены все основные технические приемы, которые вам понадобятся для проектирования, создания и ведения электронных таблиц.

Ориентировочный выход в свет — сентябрь 1994 г.

Дэн Гукин, Энди Ратбон. ПК для "чайников"

320 с., мягкий переплет.

Эта книга остается лучшим советчиком для начинающих пользователей компьютеров во всем мире. Во втором издании последние достижения в области программного, аппаратного обеспечения и периферии описаны тем же доступным языком серии «... для "чайников"», который завоевал признание у миллионов читателей. Описано как современное аппаратное обеспечение: *Pentium*, *laptop*, жесткие диски, видеоакселераторы, локальные шины, CD-ROM и др., так и программное обеспечение, включая графику, мультимедиа, а также последние версии операционных систем MS DOS 6.2, OS/2, DR DOS, UNIX. Кроме того, рассмотрены новые утилиты и программы сжатия информации. Все это преподносится просто, доходчиво, с юмором — для максимального понимания и удовольствия.

Ориентировочный выход в свет — сентябрь 1994 г.

Тина Ратбон. Модемы для "чайников"

Мягкий переплет.

Эта книга — справочное пособие. Вы можете листать ее, обращаться к ней за помощью. Вы найдете вполне доступную информацию по таким вопросам, как подключение модема, установка программного обеспечения, пересылка и получение электронной почты, подключение к сетям CompuServe, America Online или Internet, что могут модемы, а что не могут, что делать, когда модем не работает.

Ориентировочный выход в свет — сентябрь 1994 г.

Дэн Гукин. Word for Windows 6 для "чайников"

Мягкий переплет.

Описана новая версия популярного текстового процессора Word for Windows 6. Это доступное и удобное справочное руководство для начинающих пользователей. Большое внимание уделено новым свойствам, включая возможности настольных издательских систем. Рассмотрен новый подход, используемый при обработке графических изображений. Эта книга наполнена авторским юмором и легко читается.

Ориентировочный выход в свет — сентябрь 1994 г.

Продолжается публикация серии компьютерных книг от "Диалектики" и ICE



"Секреты..."



Б.Ливингстон. Секреты Windows 3.1

384 с., мягкий переплет.

Перевод на русский язык второго издания известного бестселлера Брайана Ливингстона "Секреты Windows 3.1" дает возможность вполне освоить и в дальнейшем использовать средства, предоставляемые графической средой Windows 3.1. Новичкам книга поможет подружиться с новой средой и стать грамотными пользователями, знающими Windows "снаружи" и "изнутри". Умудренные опытом пользователи Windows найдут много уникальной информации, которую невозможно получить даже в процессе длительного изучения документации.

Книга вышла в свет.

Джон Гудмэн. Секреты жесткого диска

256 с., мягкий переплет.

В книге приведены конкретные рекомендации для устранения проблем, возникающих при работе с жестким диском. Вице-президент Orange Coast IBM PC Users Group (OCIPUG) — одной из наибольших в США групп пользователей ПК, рассказывает о том, как наилучшим образом подготовить и использовать винчестер для хранения на нем информации. Приведено свыше ста практических советов и недокументированных возможностей. Рассмотрена DOS 6.

Книга вышла в свет.

Каролина Холидей. Секреты ПК

Мягкий переплет.

Автор более десяти книг по компьютерной тематике делится своими знаниями о том, как максимально эффективно использовать ПК. Рассмотрено множество утилит, используемых для анализа работы персонального компьютера. Освещены новые подходы к оптимизации и настройке как всего ПК, так и каждой его части в отдельности. Приведены конкретные рекомендации для пользователей Windows, систем CAD, настольных издательских систем, электронных таблиц и др.

Ориентировочный выход в свет — октябрь–ноябрь 1994 г.

Дэвид Стенг и Сильвия Мун. Секреты безопасности сети

Мягкий переплет.

Эксперты в области безопасности сетей делятся своим многолетним опытом, как защитить ваши данные от несанкционированного доступа и разрушения хакерами, вирусами, диверсантами и всеми остальными. Приведены обзор и оценка существующих сегодня систем защиты с помощью паролей и шифрования данных. Даны рекомендации по обеспечению безопасности NetWare и других операционных систем. Рассмотрены различные типы вирусов, способы их обнаружения и борьбы с ними.

Ориентировочный выход в свет — ноябрь–декабрь 1994 г.



Фирма "Диалектика" предоставляет возможность всем желающим получить издаваемую литературу по почте. Для этого вам необходимо выслать заявку с указанием названия книги и количества заказываемых экземпляров по адресу: *Украина, 252124 Киев-124, а/я 506, "Диалектика"* или Россия, 121467, Москва-467, а/я 9, "Инфосфера". Это наиболее удобный способ приобретения книг, особенно в удаленных регионах, где торговая наценка многократно превышает стоимость книги при отсылке почтой.

НАШ АДРЕС:

УКРАИНА, 252124 КИЕВ-124, а/я 506, "ДИАЛЕКТИКА"

РОССИЯ, 121467, МОСКВА-467, а/я 9, "ИНФОСФЕРА"

☎ / 📠 (044) 266-4074

E-MAIL: DIAL@DIAL.KIEV.UA

♣ *Обращаем внимание на то, что при получении книги по почте мы гарантируем возврат денег (за вычетом почтовых расходов), если книга не удовлетворяет вашим требованиям.*

Ответьте, пожалуйста, на следующие вопросы* :

1. Фамилия, имя, отчество: _____

2. Адрес: _____

3. На каком компьютере вы работаете:
☐ IBM-совместимом на базе процессора: ☐ 8086, ☐ 80286, ☐ 80386, ☐ 80486, ☐ Pentium
☐ Apple Macintosh (модель) _____
☐ другом (укажите, каком именно) _____
4. Какое аппаратное обеспечение вы используете:
☐ принтер ☐ стример ☐ плату для вывода звука ☐ локальную сеть
☐ сканер ☐ CD-ROM ☐ плату-видеоускоритель ☐ модем
☐ другое (укажите, какое именно) _____
5. В какой операционной системе вы работаете:
☐ DOS ☐ Windows ☐ Windows NT ☐ Windows for Workgroups
☐ OS/2 ☐ UNIX ☐ другой (укажите, какой именно) _____
6. Какое программное обеспечение вы используете (укажите, какое именно):
☐ текстовые процессоры _____ ☐ издательские системы _____
☐ CASE-системы _____ ☐ базы данных _____
☐ экспертные системы _____ ☐ графические редакторы _____
☐ электронные таблицы _____ ☐ интегрированные среды _____
☐ компиляторы _____ ☐ другое _____
7. Книги на какую тему вам хотелось бы увидеть в ближайшем будущем:

Вопросы для разработчиков программного обеспечения

8. Какое программное обеспечение вы разрабатываете (разрабатывали):
☐ драйверы ☐ языковые трансляторы ☐ системные утилиты
☐ программы для бухгалтерии и делопроизводства ☐ расчетные программы
☐ программы для работы с базами данных ☐ игровые и обучающие программы
☐ другое (укажите, какое именно) _____
9. Какие языки программирования вы используете:
☐ C/C++ ☐ Assembler ☐ FORTRAN ☐ PROLOG ☐ Clipper
☐ SQL ☐ Pascal ☐ Basic ☐ Smalltalk ☐ Lisp
☐ xBase ☐ другие (укажите, какие именно) _____
10. Какие подходы вы используете в процессе программирования:
☐ структурный ☐ модульный ☐ объектно-ориентированный ☐ декларативный
☐ визуальный ☐ другие (укажите, какие именно) _____
11. Какие компьютерные журналы вы читаете: _____

12. Каким образом вы чаще всего приобретаете журналы:
☐ покупаете ☐ временно берете почитать (в библиотеке или у знакомых)
☐ получаете по подписке
13. Статьи на какие темы вам хотелось бы увидеть в журнале "SoftReview/Компьютерное обозрение" в ближайшем будущем:

* Тому, кто заполнит эту анкету и отошлет ее в издательство "Диалектика", будет бесплатно выслан каталог издаваемых книг

Только книги серии "Секреты..."

от "Диалектики" и ICE помогут вам досконально разобраться в секретах компьютерных технологий



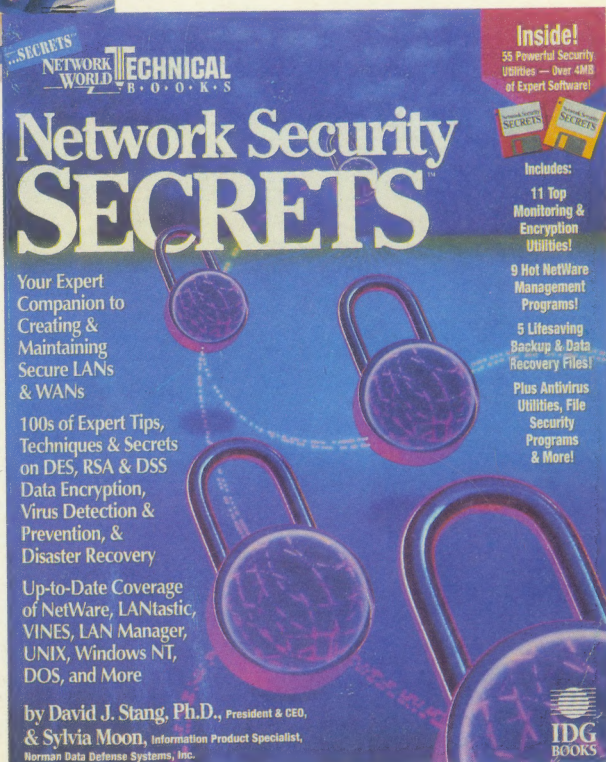
Дэвид Стенг и Сильвия Мун. **Секреты безопасности сети** 304с., мягкий переплет. Эксперты в области безопасности сетей делятся своим многолетним опытом, как защитить ваши данные от несанкционированного доступа и разрушения хакерами, вирусами, диверсантами и всеми остальными. Приведены обзор и оценка существующих систем защиты с помощью паролей и шифрования данных. Даны рекомендации по обеспечению безопасности NetWare и других операционных систем. Рассмотрены различные типы вирусов, способы их обнаружения и борьбы с ними. Перевод книги на русский язык выйдет в свет в ноябре 1994 г.



★ ICE

Каролина Холидей. Секреты ПК

Автор более десяти книг по компьютерной тематике делится своими знаниями о том, как максимально эффективно использовать ПК. Рассмотрено множество утилит, используемых для анализа работы персонального компьютера. Освещены новые подходы к оптимизации и настройке как всего ПК, так и каждой его отдельной части. Приведены конкретные рекомендации для пользователей Windows, систем CAD, настольных издательских систем, электронных таблиц и др. Перевод книги на русский язык выйдет в свет в ноябре 1994 г.



Секреты жесткого диска

"Читабельная и стоящая книга, действительно заполняющая брешь на книжном рынке, наполненном посредственными пересказами руководств и домыслами сомнительных авторов."

— Стив Гибсон, президент Gibson Research и обозреватель InfoWorld

Жесткий диск — самая сложная, дорогая и незаменимая часть вашего компьютера. Неудивительно, что люди паникуют, когда их жесткий диск приходит в негодность. После чтения "Секретов жесткого диска" вам станет легче дышать! Доктор Гудмэн не только сорвал пелену таинственности с жестких дисков, он предложил конкретные шаги для обнаружения и предотвращения часто встречающихся отказов жесткого диска.

Пусть секреты доктора Гудмэна помогут вам:

- ☼ Научиться предвосхищать проблемы с жестким диском и заставить ПК работать более эффективно
- ☼ Понять, что делать, если ваш жесткий диск приказал долго жить: как восстановить потерянные данные и избежать ущерба
- ☼ Понять работу утилиты DeFrag из поставки DOS 6.0
- ☼ Научиться с максимальной эффективностью использовать кэширование диска
- ☼ Без риска увеличить емкость вашего диска
- ☼ Изучить три основные возможности ускорения вашего жесткого диска и повышения эффективности всего ПК до 10000%
- ☼ "Технические секреты" по всему тексту преподносят наиболее существенную и современную техническую информацию для самых подготовленных пользователей

Будь вы хоть новичком, хоть самым "подкованным" пользователем, после чтения "Секретов жесткого диска" вы сумеете максимально предохранить свою информацию от всяческих катаклизмов и сэкономить многие часы своего драгоценного времени.

ОБ АВТОРЕ

Джон М. Гудмэн имеет степень доктора по физике и более 30 лет обучает техническим тонкостям и специалистов, и широкую публику. Он автор бестселлеров и бывший президент Orange Coast IBM PC User Group. Им написана книга "The Official SpinRite II and Hard Disk Companion", изданная IDG Books.

*Настольная книга для всех, работающих с жестким диском:
от начинающих до умудренных опытом пользователей*