

3.8. Файловые системы Windows.

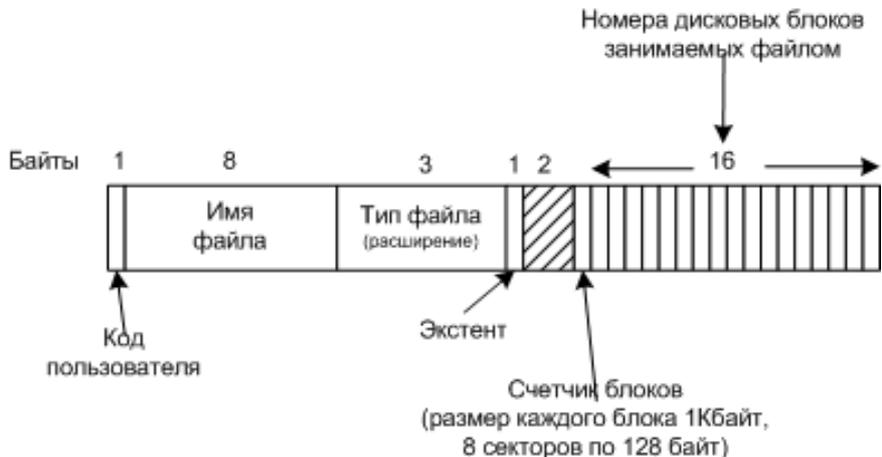
3.8.1. Файловая система CP/M

CP/M (Control Program for Microcomputers) - операционная система, предшественник MS-DOS.

В ее файловой системе только один каталог, с фиксированными записями по 32 байта.

Имена файлов - 8+3 символов верхнего регистра.

После каждой перезагрузки рассчитывается битовый массив занятых и свободных блоков. Массив находится постоянно в памяти (для 180Кбайтного диска 23 байта массива). После завершения работы, он не записывается на диск.



Каталоговая запись CP/M

Видно, что максимальный размер файла 16Кбайт (16*1Кбайт).

Для файлов размером от 16 до 32 Кбайт можно использовать две записи. Для до 48 Кбайт три записи и т.д.

Порядковый номер записи хранится в поле **экстент**.

Код пользователя - каждый пользователь мог работать только со своими файлами.

Порядок чтения файлов:

1. Файл открывается системным вызовом `open`
2. Читается каталоговая запись, из которой получает информацию о всех блоках.
3. Вызывается системный вызов `read`

3.8.2. Файловая система MS-DOS (FAT-12,16,32)

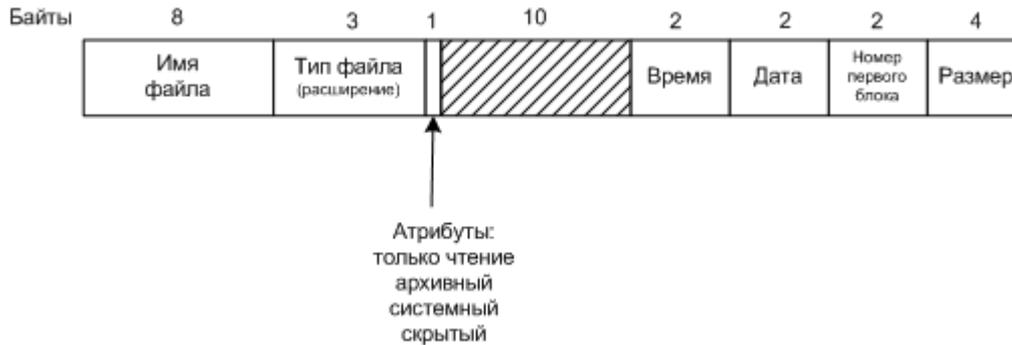
В первых версиях был только один каталог (MS-DOS 1.0). С версии MS-DOS 2.0 применили иерархическую структуру.

Каталоговые записи, фиксированны по 32 байта.

Имена файлов - 8+3 символов верхнего регистра.

Порядок чтения файлов:

1. Файл открывается системным вызовом `open`, которому указывается путь к файлу (может быть абсолютным и относительным).
2. Файловая система открывает каталоги (согласно пути), считывает последний каталог в память.
3. Ищет описатель файла.
4. Читается дескриптор файла, из которого получает информацию о всех блоках.
5. Вызывается системный вызов `read`



Каталоговая запись MS-DOS,
обратите внимание на пустые 10 байт, они будут задействованы в Windows 98

Атрибут **архивный** нужен для программ резервного копирования, по нему они определяют надо копировать файл или нет.

Поле время (16 разрядов) разбивается на три подполя:

1. секунды – 5 бит ($2^5=32$ поэтому хранятся с точностью до 2-х секунд)
2. минуты – 6 бит
3. часы – 5 бит

Поле даты (16 разрядов) разбивается на три подполя:

1. день – 5 бит
2. месяц – 4 бита
3. год – 7 бит (начинается с 1980г, т.е. максимальный 2107г.)

Теоретически размер файлов может быть до 4 Гбайт (32 разряда).

Все блоки файла в записи не хранятся, а только первый блок. Этот номер используется в качестве индекса для 64К (для FAT-16) элементов FAT-таблицы, хранящейся в оперативной памяти.

В зависимости от количества блоков на диске в системе MS-DOS применяется три версии файловой системы FAT:

1. FAT-12
2. FAT-16
3. FAT-32 - для адреса используются только 28 бит, поэтому правильнее назвать FAT-28

Размер блока (**кластера**) должен быть кратным 512 байт.

3.8.2.1 FAT-12

В первой версии MS-DOS использовалась **FAT-12** с 512 байтовыми блоками, поэтому максимальный размер раздела мог достигать 2Мбайта ($2^{12} \cdot 512$ байта).

С увеличением дисков, этого стало не хватать, стали увеличивать размер блоков 1,2 и 4 Кбайта (2^{12}) (при этом эффективность использования диска падает).

FAT-12 до сих пор применяется для гибких дисков.

3.8.2.2 FAT-16

Особенности:

- 16-разрядные дисковые указатели
- Размеры кластеров 512, 1, 2, 4, 8, 16 и 32Кбайт (2^{15})

Таблица постоянно занимала в памяти 128 Кбайт. Максимальный размер раздела диска мог достигать 2Гбайта ($2^{16} \cdot 32$ Кбайта). Причем кластер в 32 Кбайта для файлов со средним размером в 1Кбайт, не эффективен.

3.8.2.3 FAT-32

Особенности:

- 28-разрядные адреса
- Размеры кластеров 512, 1, 2, 4, 8, 16 и 32Кбайт

Максимальный размер раздела диска мог бы достигать $2^{28} \cdot 2^{15}$, но есть другое ограничение - 512 байтные сектора адресуются 32-разрядным числом, а это $2^{32} \cdot 2^9$, т.е. 2 Тбайта.

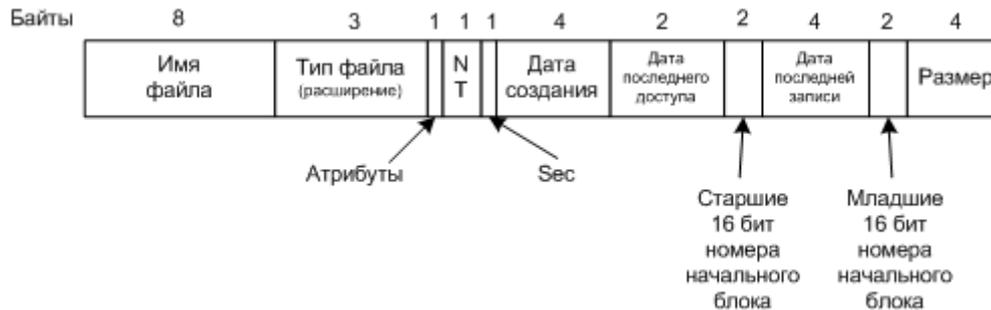
Максимальный размер раздела для различных размеров кластеров

Размер кластера, Кбайт	Fat-12, Мбайт	Fat-16, Мбайт	Fat-32, Тбайт
0.5	2	32	0.13
1	4	64	0.27
2	8	128	0.54
4	16	256	1
8		512	2
16		1024	2
32		2048	2

Из таблицы видно, что FAT-16 использовать не эффективно уже при разделах в 256 Мбайт, учитывая, что средний размер файла 1Кбайт.

3.8.2.4 Расширение Windows 98 для FAT-32

Для расширения были задействованы 10 свободных бит.



Формат каталоговой записи в системе FAT-32 с расширениями для Windows 98

Пять добавленных полей:

1. NT - предназначено для совместимости с Windows.
2. Sec - дополнение к старому полю время, позволяет хранить время с точностью до секунды (было 2 секунды)
3. Дата и время создания файла (Creation time)
4. Дата (но не время) последнего доступа (Last access)
5. Для хранения номера блока выделено еще 2 байта (16 бит), т.к. номера блоков стали 32-разрядные.

Основная надстройка над FAT-32, это длинные имена файлов.

Для каждого файла стали присваивать два имени:

1. Короткое 8+3 для совместимости с MS-DOS
2. Долгое имя файла, в формате Unicode

Доступ к файлу может быть получен по любому имени.

Если файлу дано длинное имя (или используются пробелы), то система делает следующие шаги:

- берет первые шесть символов
- преобразуются в верхний регистр ASCII, удаляются пробелы, лишние точки, некоторые символы преобразуются в "_"
- добавляется суффикс ~1
- если такое имя есть, то используется суффикс ~2 и т.д.

Короткие имена хранятся в в обычном дескрипторе файла.

Длинные имена хранятся в дополнительных каталоговых записях, идущих перед основным описателем файла. Каждая такая запись содержит 13 символов формата Unicode (для символа Unicode нужно два байта).



Формат каталогов записи с фрагментом длинного имени файла в Windows 98

Поле "Атрибуты" позволяет отличить фрагмент длинного имени (значение 0x0F) от дескриптора файла. Старые программы MS-DOS каталоговые записи со значением поля атрибутов 0x0F, просто игнорируют.

Последовательность - порядковый номер в последовательности фрагментов.

Длина имени файла ограничена 260 символами не из-за порядкового номера (1 байт), для номера используются только 6 бит $6 \times 13 = 819$ символов.

Контрольная сумма нужна для выявления ошибок, т.к. файл с длинным именем может удалить MS-DOS и создать новый, и тогда останутся не удаленные записи, которые "прилипнут" к новому файлу. Т.к. это поле один байт, есть вероятность $1/256$ что Windows 98 не заметит подмены.

3.8.3 **Файловая система NTFS**

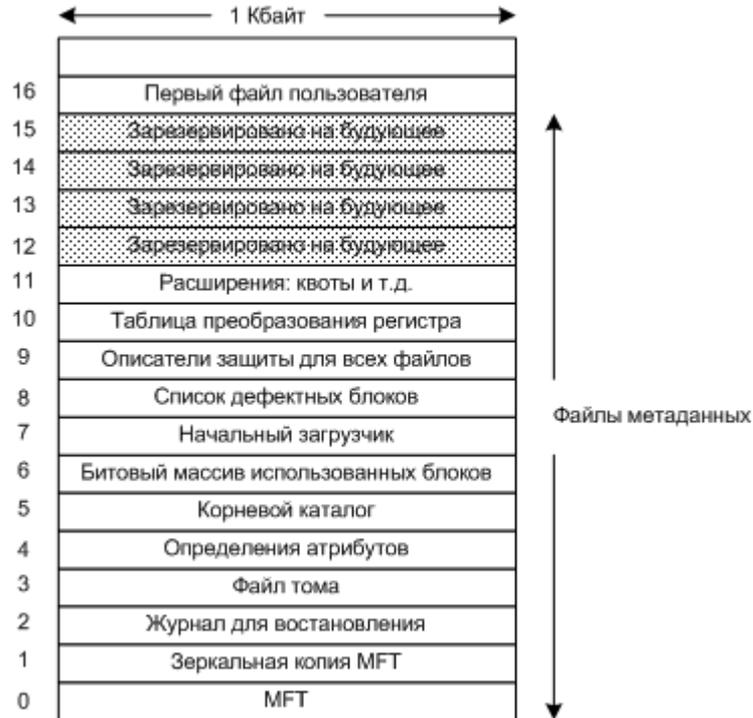
Файловая система NTFS была разработана для Windows NT.

Особенности:

- 64-разрядные адреса, т.е. теоретически может поддерживать $2^{64} \cdot 2^{16}$ байт (1 208 925 819 Пбайт~1Йбайт(2^{80})).
- Размеры блока (кластера) от 512байт до 64 Кбайт, для большинства используется 4Кбайта.
- Поддержка больших файлов.
- Имена файлов ограничены 255 символами Unicode.
- Длина пути ограничивается 32 767 (2^{15}) символами Unicode.
- Имена чувствительны к регистру, my.txt и MY.TXT это разные файлы (но из-за Win32 API использовать нельзя), это заложено на будущее.
- Журналируемая файловая система, т.е. не попадет в противоречивое состояние после сбоев.
- Контроль доступа к файлам и каталогам.
- Поддержка жестких и символических ссылок.
- Поддержка сжатия и шифрования файлов.
- Поддержка дисковых квот.

Главная файловая таблица MFT (Master File Table) - главная структура данных в каждом томе, записи фиксированные по 1Кбайту. Каждая запись описывает один каталог или файл. Для больших файлов могут использоваться несколько записей, первая запись называется - **базовой записью**.

MFT представляет собой обычный файл (размером до 2^{48} записей), который может располагаться в любом месте на диске.



Главная файловая таблица MFT, каждая запись ссылается на файл или каталог.

Первые 16 записей MFT зарезервированы для файлов метаданных. Каждая запись описывает нормальный файл, имена этих файлов начинаются с символа "\$".

Каждая запись представляет собой последовательность пар (заголовок атрибута, значение).

Некоторые записи метаданных в MFT:

0) Первая запись описывает сам файл MFT, и содержит все блоки файла MFT. Номер первого блока файла MFT содержится в загрузочном блоке.

1) Дубликат файла MFT, резервная копия.

2) Журнал для восстановления, например, перед созданием, удалением каталога делается запись в журнал. Система не попадет в противоречивое состояние после сбоев.

3) Информация о томе (размер, метка и версия)

4) Определяются атрибуты для MFT записей.

6) Битовый массив использованных блоков - для учета свободного места на диске

7) Указывает на файл начальной загрузки

Атрибуты, используемые в записях MFT:

- Стандартная информация - флаговые биты (только чтение, архивный), временные штампы и т.д.
- Имя файла - имя файла в кодировке Unicode, файлы могут повторяться в формате MS-DOS 8+3.
- Список атрибутов - расположение дополнительных записей MFT
- Идентификатор объекта - 64-разрядный идентификатор файла, уникальный для данного тома.
- Точка повторного анализа - используется для символьных ссылок и монтирования устройств.

- Название тома
- Версия тома
- Корневой индекс - используется для каталогов
- Размещение индекса - используется для очень больших каталогов
- Битовый массив - используется для очень больших каталогов
- Поток данных утилиты регистрации - используется для шифрования
- Данные - поточные данные, может повторяться, используется для хранения самого файла. За заголовком следует список дисковых адресов, определяющий положение файла на диске, если файл очень маленький (несколько сотен байт), то следует сам файл (такой файл называется - **непосредственный файл**).

Как правило, все данные файла не помещаются в запись MFT.

Дисковые блоки файлам назначаются по возможности в виде **серий последовательных блоков (сегментов файлов)**. В идеале файл должен быть записан в одну серию (не фрагментированный файл), файл, состоящий из n блоков, может быть записан от 1 до n серий.



Запись MFT для 9-блочного файла, состоящего из трех сегментов (серий).
 Вся запись помещается в одну запись MFT (файл не сильно фрагментирован).

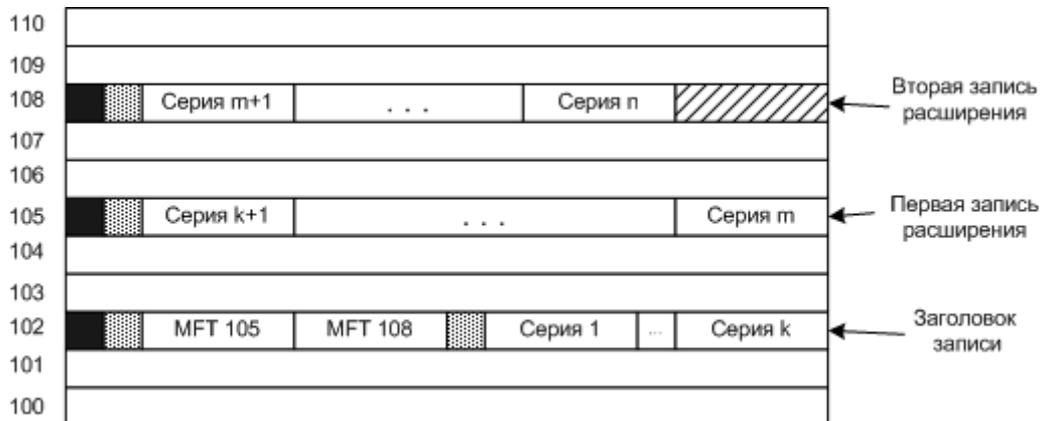
Заголовок содержит количество блоков (9 блоков).

Каждая серия записывается в виде пары, дисковый адрес - количество блоков (20-4, 64-2, 80-3).

Каждая пара, при отсутствии сжатия, это два 64-разрядные числа (16 байт на пару).

Многие адреса содержат большое количество нулей, сжатие делается за счет убирания нулей в старших байтах. В результате для пары требуется чаще всего 4байта.

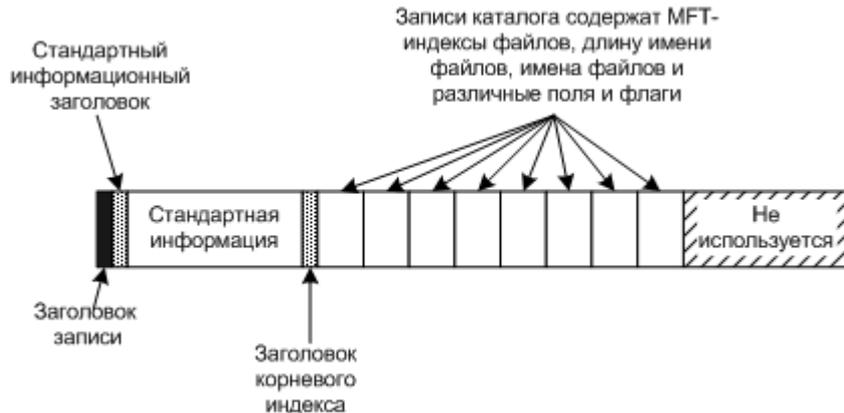
Если файл сильно фрагментирован, требуется несколько записей MFT.



Три записи MFT для сильно фрагментированного файла.

В первой записи указывается индексы на дополнительные записи.

Может потребоваться очень много индексов MFT, так что индексы не поместятся в запись. В этом случае список хранится не в MFT, а в файле.



Запись MFT для небольшого каталога

Поиск файла в каталоге по имени состоит в последовательном переборе имен файлов.

Для больших каталогов используется другой формат. Используется дерево B+, обеспечивающее поиск в алфавитном порядке.

3.8.3.1 Поиск файла по имени

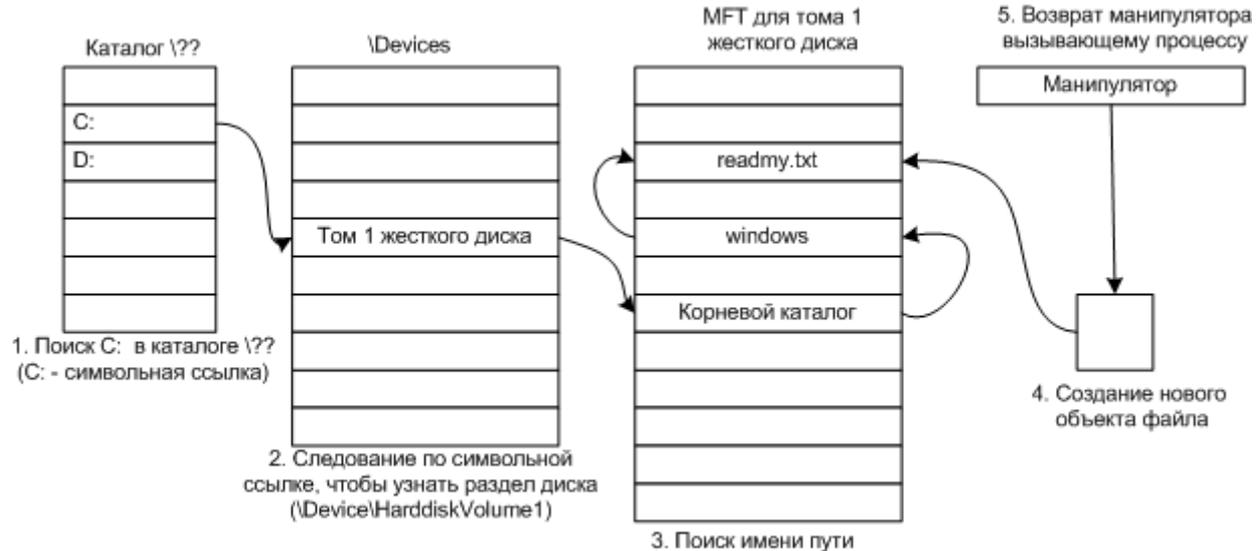
При создании файла, программа обращается к библиотечной процедуре

```
CreateFile("C:\windows\readmy.txt", ...)
```

Этот вызов попадает в совместно используемую библиотеку уровня пользователя kernel32.dll, где \??\ помещается перед именем файла, и получается строка:

```
\??\C:\windows\readmy.txt
```

Это имя пути передается системному вызову NtFileCreate в качестве параметра.



Этапы поиска файла `C:\windows\readmy.txt`

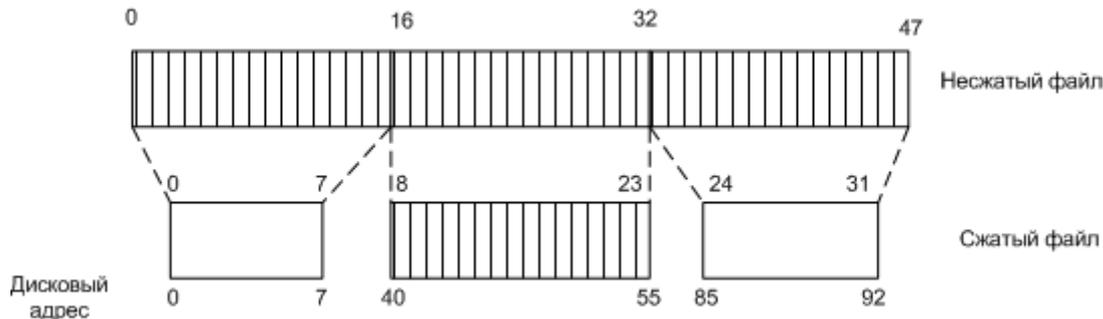
3.8.3.2 Сжатие файлов

Если файл помечен как сжатый, то система автоматически сжимает при записи, а при чтении происходит декомпрессия.

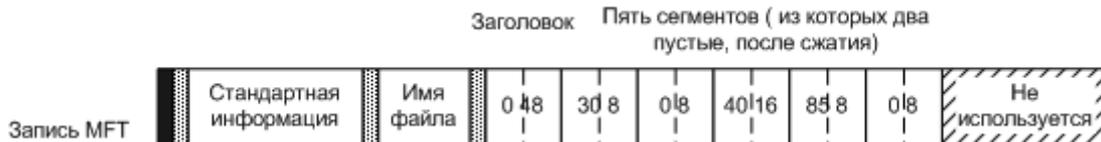
Алгоритм работы:

1. Берутся для изучения первые 16 блоков файла (не зависимо от сегментов файла).
2. При меняется к ним алгоритм сжатия.

3. Если полученные данные можно записать хотя бы в 15 блоках, они записываются в сжатом виде.
 Если их можно записать только в 16 блоках, то они записываются в несжатом виде.
4. Алгоритм повторяется для следующих 16 блоков.



Пример 48-блочного файла, сжатого до 32 блоков



Запись MFT для предыдущего файла.

Недостатки сжатия:

- Как видно из рисунка, сжатие приводит к сильной фрагментации.

- Чтобы прочитать сжатый блок системе придется распаковать весь сегмент. Поэтому сжатие применяют к 16 блокам, если увеличить количество блоков, уменьшится производительность (но возрастет эффективность сжатия).

3.8.3.3 Шифрование файлов

Любую информацию, если она не зашифрована, можно прочитать, получив доступ. Поэтому самая надежная защита информации от несанкционированного доступа - шифрование.

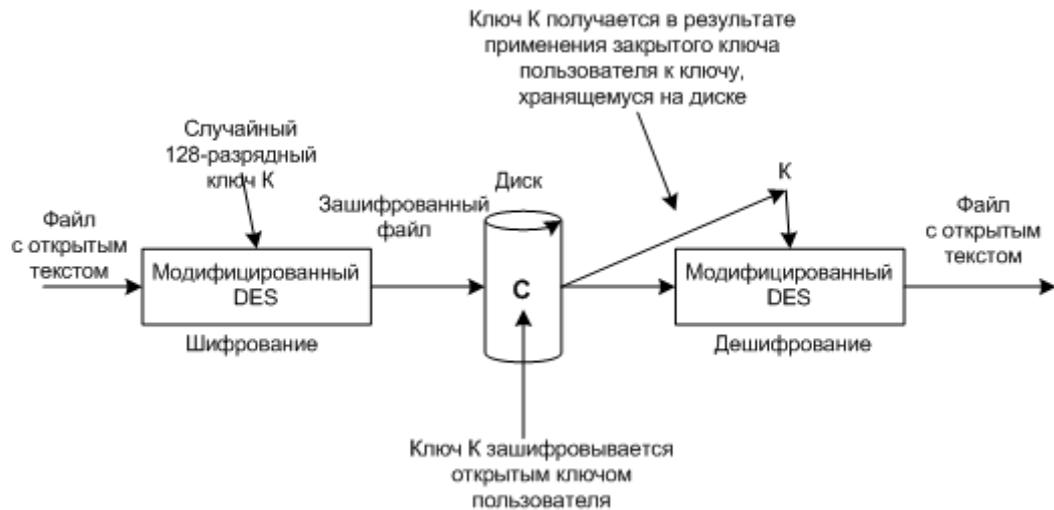
Даже если у вас украдут винчестер, прочесть данные не смогут (большинство не сможет).

Если файл помечен как зашифрованный, то система автоматически шифрует при записи, а при чтении происходит дешифрация.

Шифрование и дешифрование выполняет не сама NTFS, а специальный драйвер **EFS** (Encrypting File System).

Каждый блок шифруется отдельно.

В Windows 2000 используется случайно сгенерированный 128-разрядный ключ для каждого файла. Этот ключ шифруется открытым ключом пользователя и сохраняется на диске.



Шифрование файлов в NTFS