ФАЙЛОВАЯ СИСТЕМА NTFS

Операционные системы Microsoft семейства Windows NT нельзя представить без файловой системы NTFS - одной из самых сложных и удачных из существующих на данный момент файловых систем

ФАЙЛОВАЯ СИСТЕМА NTFS

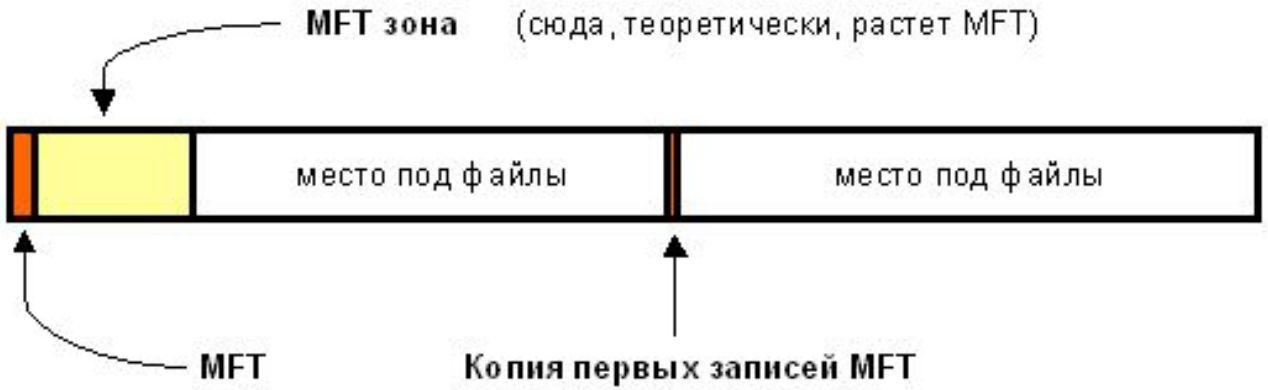
Раздел NTFS, теоретически, может быть почти какого угодно размера.

Максимальный размер раздела NTFS в данный момент ограничен лишь размерами жестких дисков.

Система NTFS делит все полезное место на кластеры - блоки данных, используемые единовременно. NTFS поддерживает почти любые размеры кластеров - от 512 байт до 64 Кбайт, неким стандартом же считается кластер размером 4 Кбайт. Никаких аномалий кластерной структуры NTFS не имеет.

ФАЙЛОВАЯ СИСТЕМА NTFS

СТРУКТУРА РАЗДЕЛА



Диск NTFS условно делится на две части. Первые 12% диска отводятся под так называемую MFT зону - пространство, в которое растет метафайл MFT (по другому – для всех файловых систем – таблица содержания). Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится пустой - это делается для того, чтобы самый главный, служебный файл (MFT) не фрагментировался при своем росте. Остальные 88% диска представляют собой обычное пространство для хранения файлов (для всех файловых систем – область данных – совокупность блоков (кластеров) на диске, индентифицируемых своими номерами/адресами).

ФАЙЛОВАЯ СИСТЕМА NTFS

СТРУКТУРА РАЗДЕЛА

ОДНАКО

свободное место на диски включают также - незаполненные куски MFT-зоны. Механизм использования MFT-зоны таков: когда файлы уже нельзя записывать в обычное пространство, MFT-зона просто сокращается (в текущих версиях операционных систем ровно в два раза), освобождая таким образом место для записи файлов. При освобождении места в обычной области MFT зона может снова расширится. При этом не исключена ситуация, когда в этой зоне остались и обычные файлы: никакой аномалии тут нет. Что ж, система старалась оставить её свободной, но ничего не получилось. Метафайл MFT все-таки может фрагментироваться, хоть это и было бы нежелательно.

ФАЙЛОВАЯ СИСТЕМА NTFS

MFT и его структура

В файловой структуре NTFS каждый элемент системы представляет собой файл - даже служебная информация. Самый главный файл на NTFS называется MFT, или Master File Table - общая таблица файлов.

Именно он размещается в MFT зоне и представляет собой централизованный каталог всех остальных файлов диска, и себя самого. MFT поделен на записи фиксированного размера (обычно 1 Кбайт), и каждая запись соответствует какому либо файлу (в общем смысле этого слова). Первые 16 файлов носят служебный характер и недоступны операционной системе - они называются метафайлами, причем самый первый метафайл - сам MFT. Эти первые 16 элементов MFT - единственная часть диска, имеющая фиксированное положение.

Вторая копия первых трех записей, для надежности хранится ровно посередине диска. Остальной MFT-файл может располагаться, как и любой другой файл, в произвольных местах диска - восстановить его положение можно с помощью его самого, "зацепившись" за самую основу - за первый элемент MFT.

ФАЙЛОВАЯ СИСТЕМА NTFS

Метафайлы

Первые 16 файлов NTFS (метафайлы) носят служебный характер. Каждый из них отвечает за какой-либо аспект работы системы. Преимущество настолько модульного подхода заключается в гибкости - например, на FAT-е физическое повреждение в самой области FAT фатально для функционирования всего диска, а NTFS может сместить, даже фрагментировать по диску, все свои служебные области, обойдя любые неисправности поверхности - кроме первых 16 элементов MFT.

Метафайлы находятся корневом каталоге NTFS диска - они начинаются с символа имени "$». Получить какую-либо информацию о них стандартными средствами сложно.

ФАЙЛОВАЯ СИСТЕМА NTFS

Метафайлы

В таблице приведены используемые в данный момент метафайлы и их назначение.

$MFT	**сам MFT**
$MFTmirr	**копия первых 16 записей MFT, размещенная посередине диска**
$LogFile	**файл поддержки журналирования (см. ниже)**
$Volume	**служебная информация - метка тома, версия файловой системы, т.д.**
$AttrDef	**список стандартных атрибутов файлов на томе**
$.	**корневой каталог**
$Bitmap	**карта свободного места тома**
$Boot	**загрузочный сектор (если раздел загрузочный)**
$Quota	**файл, в котором записаны права пользователей на использование дискового пространства (начал работать лишь в NT5)**
$Upcase	**файл - таблица соответствия заглавных и прописных букв в имен файлов на текущем томе. Нужен в основном потому, что в NTFS имена файлов записываются в Unicode, что составляет 65 тысяч различных символов, искать большие и малые эквиваленты которых очень нетривиально.**

ФАЙЛОВАЯ СИСТЕМА NTFS

Что такое файл NTFS?

- **Прежде всего, обязательный элемент - запись в MFT, все файлы диска упоминаются в MFT. В этом месте хранится вся информация о файле, за исключением собственно данных. Имя файла, размер, положение на диске отдельных фрагментов, и т.д. Если для информации не хватает одной записи MFT, то используются несколько, причем не обязательно подряд.**
- **Опциональный элемент - потоки данных файла:**

Во-первых, файл может не иметь данных - в таком случае на него не расходуется свободное место самого диска.
Во-вторых, файл может иметь не очень большой размер. Тогда идет в ход довольно удачное решение: данные файла хранятся прямо в MFT, в оставшемся от основных данных месте в пределах одной записи MFT.

Файлы, занимающие сотни байт, обычно не имеют своего "физического" воплощения в основной файловой области - все данные такого файла хранятся в одном месте - в MFT.

ФАЙЛОВАЯ СИСТЕМА NTFS

ПОТОКИ

Каждый файл на NTFS имеет несколько абстрактное строение - у него нет как таковых данных, а есть потоки (streams). Один из потоков и носит привычный нам смысл - данные файла. Большинство атрибутов файла - тоже потоки! Таким образом, получается, что базовая сущность у файла только одна - номер в MFT, а всё остальное опционально.

Данная абстракция может использоваться для создания довольно удобных вещей - например, файлу можно "прилепить" еще один поток, записав в него любые данные - например, информацию об авторе и содержании файла.

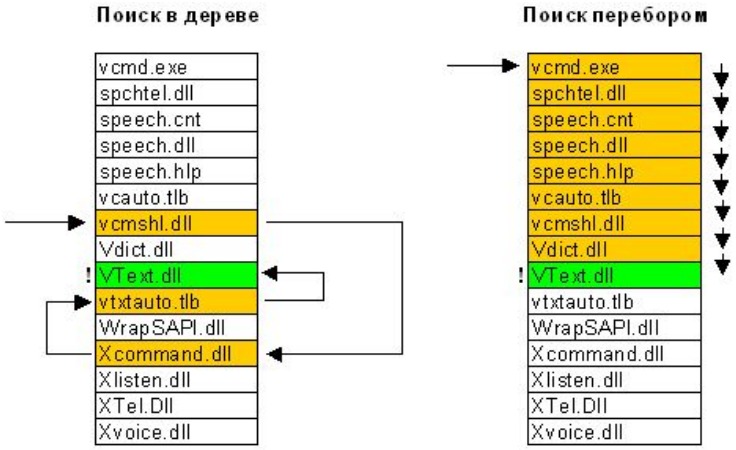
Дополнительные потоки не видны стандартными средствами: наблюдаемый размер файла - это лишь размер основного потока, который содержит традиционные данные.

Можно, к примеру, иметь файл нулевой длинны, при стирании которого освободится 1 Гбайт свободного места - просто потому, что какая-нибудь хитрая программа или технология прилепила в нему дополнительный поток (альтернативные данные) гигабайтового размера. Но на самом деле в текущий момент потоки практически не используются, так что опасаться подобных ситуаций не следует, хотя гипотетически они возможны. Просто надо иметь ввиду, что файл на NTFS - это более глубокое и глобальное понятие, чем можно себе вообразить просто просматривая каталоги диска.

Имя файла может содержать любые символы, включая полый набор национальных алфавитов, так как данные представлены в Unicode - 16-битном представлении, которое дает 65535 разных символов. Максимальная длина имени файла - 255 символов.

ФАЙЛОВАЯ СИСТЕМА NTFS

Каталоги



Каталог на NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске. Файл каталога поделен на блоки, каждый из которых содержит имя файла, базовые атрибуты и ссылку на элемент MFT, который уже предоставляет полную информацию об элементе каталога.

Внутренняя структура каталога представляет собой бинарное дерево. Бинарное дерево располагает имена файлов таким образом, чтобы поиск файла осуществлялся более быстрым способом - с помощью получения двухзначных ответов на вопросы о положении файла.

Вопрос, на который бинарное дерево способно дать ответ, таков: в какой группе, относительно данного элемента, находится искомое имя - выше или ниже? Мы начинаем с такого вопроса к среднему элементу, и каждый ответ сужает зону поиска в среднем в два раза. Файлы, скажем, просто отсортированы по алфавиту, и ответ на вопрос осуществляется очевидным способом - сравнением начальных букв.

Вывод - для поиска одного файла среди 1000, например, FAT придется осуществить в среднем 500 сравнений (наиболее вероятно, что файл будет найден на середине поиска), а системе на основе дерева - всего около 12-ти

ФАЙЛОВАЯ СИСТЕМА NTFS

Журналирование

NTFS - отказоустойчивая система, которая вполне может привести себя в корректное состояние при практически любых реальных сбоях. Любая современная файловая система основана на таком понятии, как транзакция - действие, совершаемое целиком и корректно или не совершаемое вообще. У NTFS просто не бывает промежуточных (ошибочных или некорректных) состояний - квант изменения данных не может быть поделен на до и после сбоя, принося разрушения и путаницу - он либо совершен, либо отменен.

Необходимо помнить, что журналирование - не абсолютная панацея, а лишь средство существенно сократить число ошибок и сбоев системы. Опыт показывает, что NTFS восстанавливается в полностью корректное состояние даже при сбоях в очень загруженные дисковой активностью моменты. Вы можете даже оптимизировать диск и в самый разгар этого процесса нажать reset - вероятность потерь данных даже в этом случае будет очень низка.

Однако, важно понимать, что система восстановления NTFS гарантирует корректность файловой системы, а не ваших данных. Если вы производили запись на диск и получили аварию - ваши данные могут и не записаться. Чудес не бывает.

ФАЙЛОВАЯ СИСТЕМА NTFS

Сжатие

Файлы NTFS имеют один довольно полезный атрибут - "сжатый". NTFS имеет встроенную поддержку сжатия дисков - то, для чего раньше приходилось использовать Stacker или DoubleSpace.

Любой файл или каталог в индивидуальном порядке может хранится на диске в сжатом виде - этот процесс совершенно прозрачен для приложений.

Сжатие файлов имеет очень высокую скорость и только одно большое отрицательное свойство - огромная виртуальная фрагментация сжатых файлов, которая, правда, никому особо не мешает.

Сжатие осуществляется блоками по 16 кластеров и использует так называемые "виртуальные кластеры" - опять же предельно гибкое решение, позволяющее добиться интересных эффектов - например, половина файла может быть сжата, а половина - нет.

ФАЙЛОВАЯ СИСТЕМА NTFS

Сжатие

Физической раскладки для реального, несжатого, файла:

кластеры файла с 1 по 43-й хранятся в кластерах диска начиная с 400-го
кластеры файла с 44 по 52-й хранятся в кластерах диска начиная с 8530-го
…

Физическая раскладка типичного сжатого файла:

кластеры файла с 1 по 9-й хранятся в кластерах диска начиная с 400-го
кластеры файла с 10 по 16-й нигде не хранятся
кластеры файла с 17 по 18-й хранятся в кластерах диска начиная с 409-го
кластеры файла с 19 по 36-й нигде не хранятся

Алгоритм: как только система видит такие виртуальные кластеры, она тут же понимает, что данные предыдущего блока, кратного 16-ти, должны быть разжаты, а получившиеся данные как раз заполнят виртуальные кластеры.

алгоритм сжатия: Лемпеля - Зива

ФАЙЛОВАЯ СИСТЕМА NTFS

Безопасность

NTFS содержит множество средств разграничения прав объектов - есть мнение, что это самая совершенная файловая система из всех ныне существующих. В теории это, без сомнения, так, но в текущих реализациях, к сожалению, система прав достаточно далека от идеала и представляет собой хоть и жесткий, но не всегда логичный набор характеристик.

Права файловой системы NTFS неразрывно связаны с самой системой - то есть они, вообще говоря, необязательны к соблюдению другой системой, если ей дать физический доступ к диску. Для предотвращения физического доступа (NT5) всё же ввели стандартную возможность.

Система прав в своем текущем состоянии достаточно сложна, и здесь мы не будем их рассматривать. Позже мы рассмотрим систему безопасности файловых систем ext, где организация безопасности более совершена и поняв систему безопасности там, станет понятна и система безопасности NTFS.

ФАЙЛОВАЯ СИСТЕМА NTFS

Особенности дефрагментации NTFS

Система сильнее всего фрагментирует файлы когда свободное место кончается, когда приходится использовать мелкие дырки, оставшиеся от других файлов.

Тут возникает первое свойство NTFS, которое прямо способствует серьезной фрагментации.
Диск NTFS поделен на две зоны. В начала диска идет MFT зона - зона, куда растет MFT, Master File Table. Зона занимает минимум 12% диска, и запись данных в эту зону невозможна. Это сделано для того, чтобы не фрагментировался хотя бы MFT. Но когда весь остальной диск заполняется - зона сокращается ровно в два раза. И так далее. Таким образом мы имеем не один заход окончания диска, а несколько.

В результате если NTFS работает при диске, заполненном на около 90% - фрагментация растет как бешенная. Попутное следствие - диск, заполненный более чем на 88%, дефрагментировать почти невозможно - даже API дефрагментации не может перемещать данные в MFT зону. Может оказаться так, что у нас не будет свободного места для маневра.

То, что NTFS препятствует фрагментации файлов - миф.

Фрагментация NTFS, из-за несовершенного алгоритма записи файла, через пол года работы доведет до искреннего удивления любого человека, знакомого с работой файловой системой. Поэтому приходится запускать дефрагментатор.

Но на этом все наши проблемы не заканчиваются, а, увы, только начинаются...

ФАЙЛОВАЯ СИСТЕМА NTFS

Особенности дефрагментации NTFS

В NT существует стандартное API дефрагментации. Обладающее интересным ограничением для перемещения блоков файлов: за один раз можно перемещать не менее 16 кластеров (!), причем начинаться эти кластеры должны с позиции, кратной 16 кластерам в файле. В общем, операция осуществляется исключительно по 16 кластеров.

Следствия:

- В дырку свободного места менее 16 кластеров нельзя ничего переместить (кроме сжатых файлов, но это неинтересные в данный момент тонкости).
- Файл, будучи перемещенный в другое место, оставляет после себя (на новом месте) "временно занятое место", дополняющее его по размеру до кратности 16 кластерам.
- При попытке как-то неправильно ("не кратно 16") переместить файл результат часто непредсказуем. Что-то округляется, что-то просто не перемещается… Тем не менее, всё место действия щедро рассыпается "временно занятым местом".

Складывание файлов ближе к началу - так называемая дефрагментация свободного места. Вот это - воистину страшный процесс...

Допустим, мы хотим положить файлы подряд в начало диска. Кладем один файл. Он оставляет хвост занятости дополнения до кратности 16. Кладем следующий - после хвоста, естественно. Через некоторое время, по освобождению хвоста, имеем дырку <16 кластеров размером. Которую потом невозможно заполнить через API дефрагментации! В результате, до оптимизации картина свободного места выглядела так: много дырок примерно одинакового размера.

После оптимизации - одна дыра в конце диска, и много маленьких <16 кластеров дырок в заполненном файлами участке.

Какие места в первую очередь заполняются? Правильно, находящиеся ближе к началу диска мелкие дырки <16 кластеров... Любой файл, плавно созданный на прооптимизированном диске, будет состоять из дикого числа фрагментов. Да, диск потом можно оптимизировать снова. А потом еще раз.. и еще.. и так - желательно каждую неделю. Бред? Реальность.

ФАЙЛОВАЯ СИСТЕМА NTFS

Особенности дефрагментации NTFS

Таким образом, имеется два примерно равнозначных варианта.

- **Первый - часто оптимизировать диск таким дефрагментатором, смиряясь при этом с дикой фрагментацией заново созданных файлов.**
- **Второй вариант - вообще ничего не трогать, и смириться с равномерной, но гораздо более слабой фрагментацией всех файлов на диске.**

Не достаток дефрагментатора тщательно скрывается, по крайней мере уж точно не афишируется на каждом шагу.

Speeddisk для W2k - единственная на сегодняшний день программа, которая может оптимизировать диск полностью, не создавая маленьких незаполненных фрагментов свободного места. Стоит добавить также, что при помощи стандартного API невозможно дефрагментировать тома NTFS с кластером более 4 Кбайт, а SpeedDisk и это может.

К сожалению, стандартно размещен дефрагментатор, который работает через API, и, соответственно, плодит дырки <16 кластеров.

Как некоторый вывод из всего этого: все остальные дефрагментаторы при одноразовом применении просто вредны. Если вы запускали его хоть раз - нужно запускать его потом хотя бы раз в месяц, чтобы избавится от фрагментации новоприбывающих файлов. В этом основная суть сложности дефрагментации NTFS теми средствами, которые сложились исторически.