# NTFS

New technology File System

На момент появления решала многие проблемы системы FAT.

Отсутствует ограничение на размер файлов. Имеются => возможности:

* Возможность возвращаться к работоспособному состоянию после сбоя. Такое возможно из-за атомарных транзакций, и наличия избыточности хранения информации. (*Пример: нажатие CTRL+Z)*
* Защита файлов от несанкционированного доступа (*В качестве объектов защиты выступают файлы)*
* Шифрование (*Можно сессионным ключом в FAT ?*)
* Дисковые квоты для пользователей. Для пользователя создается отдельное пространство для записи. Больше чем выделено пользователь записать не сможет. *Для сетевых пользователей.*

Система появилась на основе HPFS – HIGH PERFORMANCE FILE SYTEM.

**Структура**:

(рисунок 1)

Загрузочная запись – содержит информацию о томе и т.д. 16 секторов. 8 килобайт. В определенной области тома расположена основная системная структура MFT. Адрес начала этой области указывается в загрузочной записи. Она хранится вместе с остальными файлами на диске. Здесь вся информация о расположении файлов на томе. А небольшие файлы хранятся прямо внутри записей MFT.

Вся информация как пользовательская, так и системная хранится в виде файла. Имена системных файлов: начинаются с $ . *Загрузочная запись файлов тома в $Boot.*

Поскольку таблица MFT является важной структурой, то соответствующий файл хранится в непрерывной области диска, чтобы избежать его фрагментации.

С этой целью выделяется непрерывная область – зона MFT, и по мере увеличения – занимается зарезервированное место. Остальное – под файлы системные и пользовательские.

(рисунок 2)

Запись $LogFile – файл журнала, информация об операциях, меняющих структуру тома. Например это создание файлов и каталогов. Используется для восстановления.

$Volume – набор флагов состояний тома. «грязный бит» - том был поврежден и может (должен ) быть восстановлен

$AttrDef – таблица определения атрибутов.

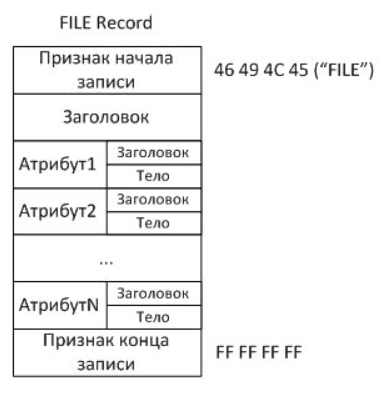
$RootDirectory – (обозначается также обратным слешем "\") – файл с информацией о корневом каталоге тома. В нем хранятся ссылки на файлы и каталоги, содержащиеся в корневом каталоге;

$BitMap – файл битовой карты. Если бит =1 кластер занят, = 0 – свободен.

$Boot – файл загрузочной записи тома;

$BadClus – файл плохих кластеров (bad clusters), содержащий информацию обо всех кластерах, имеющих сбойные секторы (bad sectors).

(рисунок 3)



Основная информация о файле содержится в файловой записи, а небольшие файлы целиком хранятся в файловой записи.

Файловая запись состоит из заголовка и атрибутов. В заголовке содержится служебная информация о файловой записи, например, её тип и размер. Все данные, относящиеся непосредственно к файлу, хранятся в виде атрибутов. Названия атрибутов, так же как и системных файлов, начинаются с "$". Например, отдельными атрибутами являются имя файла ($FILE\_NAME), информация о его свойствах ($STANDARD\_INFORMATION), данные файла ($DATA).

На диске файловая запись всегда расположена в начале сектора, первые байты файловой записи кодируют слово "FILE" (ASCII-коды: 46 49 4C 45). Конец записи определяется 4 байтовой последовательностью FF FF FF FF.

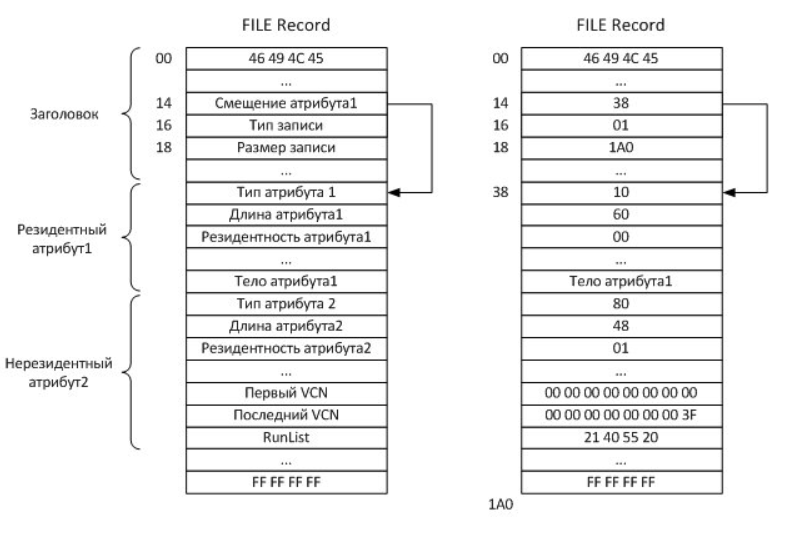
Физически атрибут файла хранится в виде потока байтов (stream) – простой последовательности байтов. Такое представление позволяет одинаковым образом работать с разнотипными атрибутами, а также добавлять нестандартные пользовательские атрибуты.

Каждый атрибут состоит из заголовка (attribute header), определяющего тип атрибута и его свойства, и тела (attribute body), содержащего основную информацию атрибута.

Атрибуты бывают:

* Резидентные полностью помещаются в файловую запись MFT
* Нерезидентные - хранятся вне MFT . Такая область называется группой.

Файловая запись при наличии нерезидентных атрибутов содержит ссылку на расположение группы на диске



В начале файловой записи находится признак её начала – слово "FILE" (46 49 4C 45). По смещению 0x14 расположено двухбайтовое поле, в котором записано смещение первого атрибута относительно начала файловой записи. В примере в этом поле записано 38, т. е. первый атрибут расположен по смещению 38.

В следующем поле хранится тип файловой записи: значение 1 обозначает файл, 2 – каталог (directory). В примере файловая запись соответствует файлу (значение 1 по смещению 16).

Ещё одно поле в заголовке содержит размер всей записи. В примере на в этом поле записано 1A0, т. е. размер записи составляет 416 байт.

Каждый атрибут имеет поля, указывающие тип, длину и резидентность атрибута. Все типы атрибутов имеют свои численные значения, например, атрибуту $FILE\_NAME соответствует значение 0x30, атрибуту $STANDARD\_INFORMATION – 0x10, атрибуту $DATA – 0x80.

Если атрибут резидентный, в поле резидентности записывается 0x00, иначе – 0x01. В случае нерезидентного атрибута предусмотрены поля для хранения номеров кластеров, в которых располагается группа или несколько групп, выделенных для размещения файла.

В примере на показаны два атрибута. Первый атрибут имеет тип $STANDARD\_INFORMATION (значение 10), длина атрибута 96 байт (6016 = 9610), атрибут является резидентным (00).

У второго атрибута тип $DATA (80), длина – 72 байта (4816 = 7210), атрибут является нерезидентным (01).

Для обозначения кластеров используются два типа номеров: LCN и VCN. При помощи первого типа, LCN (Logical Cluster Number – логический номер кластера), нумеруются все кластеры на диске, от первого до последнего. LCN применяются, чтобы найти начальный кластер группы. Номера VCN (Virtual Cluster Number – виртуальный номер кластера) обозначают порядковый номер кластера внутри группы. Схема нумерации кластеров LCN VCN проиллюстрирована на рис.17.6.

В случае нерезидентных атрибутов в заголовке атрибута содержатся следующие поля: номер VCN первого кластера группы (обычно равен 0х00), номер VCN последнего кластера группы и список групп (RunList), описывающий расположение групп на диске.

Рассмотрим пример описания расположения групп, приведенный на рис.17.5 (справа). В этом примере значения полей следующие:

* первый VCN = 0x00;
* последний VCN = 0x3F;
* список групп (RunList) = 0x21 40 55 20 00.

Групп может быть несколько.