**13. Расширенные возможности NTFS: сжатие и шифрация данных, точки повторной обработки, монтирование томов, жесткие и символические связи.**

*Сжатие*

Сжатие в ntfs похоже на фрагментацию файла

Сжатие происходит блоками по 16 кластеров.

Сначала анализируется попытка сжатия. Если получился эффект с сокращением хотя бы в 1 кластер, то происходит сжатие иначе блок хранится в исходном виде.

Сжатые файлы имеют “виртуальные” кластеры, которые не несут реальной информации. Если при прочтении информации система видит “виртуальный” кластер, то данные предыдущего блока кратного 16-ти должны быть разжаты, а полученные данные заполнят виртуальные кластеры.

Экстенты “виртуальных” кластеров так же имеют адреса (LFN, VCN, k) в MFT. В качестве признака виртуальности кластера используется LFN = 0, который соответствует блоку начальной загрузки тома и, следовательно, не может быть у файлов.

У сжатых 16 кластерных блоков имеет имеется 2 адреса в адресной части: первый отвечает за экстенты реальных физических блоков, пример: (60, 30, 4); и адрес для экстентов виртуальных блоков, где LFN = 0, пример: (0, 30, 3)

*Шифрация*

Encrypting file system. Шифрующая файловая система это тесно интегрированная с NTFS служба, располагающаяся в ядре Windows. Ее назначение: защита данных, хранящихся на диске, от несанкционированного доступа путем их шифрования.

EFS шифрует каждый файл симметричным алгоритмом блочного шифрования, используя при это м для каждого файла случайно сгенерированных ключ – File Encrypted Key (FEK).

Затем применяется инфраструктура открытого ключа Windows – Public Key Infrastructure с использованием цифровых сертификатов.

Для каждого пользователя генерируется пара открытый/закрытый ключ. Открытая часть используется для шифрации FEK и сохраняется в реестре. Закрытый ключ используется для дешифрации FEK драйвером EFS.

Закрытый ключ не хранится в открытом виде в системе, а шифруется паролем пользователя. При входе пользователя система получает пароль, дешифрует закрытую часть ключа и сохраняет её в виртуальном адресном пространстве. Очевидно, что стойкость такого подхода шифрации зависит от стойкости пароля пользователя.

Сопоставить зашифрованный файл с ключом, способным его дешифровать можно при помощи цифрового сертификата, который создается вместе с зашифрованным файлом. Сертификаты можно увидеть в консоли управления - certmgr.msc, а в атрибутах сжатия и шифрации файла можно увидеть отпечаток сертификата.

*Точки повторной обработки*

Точкой повторной обработки (Repairs point) называется атрибут файла, который назначает определённый драйвер фильтра, включающийся в обработку файла/каталога. Данный фильтр управляет обработкой файла.

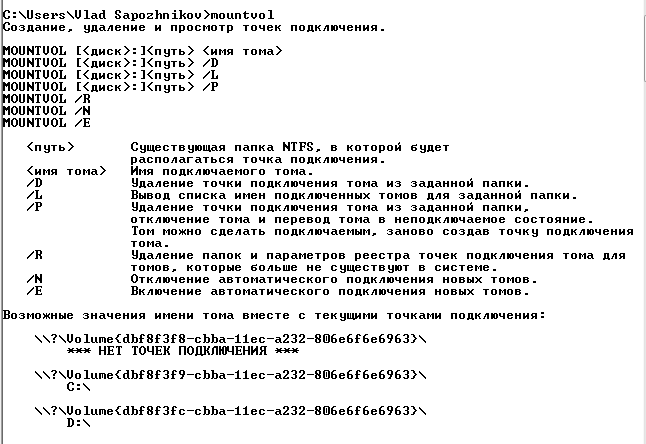
При создании точки повторной обработки у файла/каталога NTFS создает у объекта атрибут $Repairs. Операционная система позволяет приложениям перечислять все точки повторной обработки и/или точки монтирования, расположенные в пределах тома. Для этого NTFS хранит информацию о точках повторной обработки (включая точки монтирования) в файле $Extend\$Reparse. Все точки повторной обработки на томе NTFS индексируются в файле $Index, который находится в каталоге \$Extend. Таким образом, приложение может быстро индексировать все эти точки.

Файл/каталог с точкой обработки может быть обработан только при помощи соответствующего драйвера фильтра. Если драйвер фильтра не найден, то файл/каталог не откратоется и может быть только удален.

Точка монтирования (mount point) — это каталог или файл, с помощью которого обеспечивается доступ к новой файловой системе, каталогу или файлу. Точка монтирования используется для реализации возможности динамически присоединять/отсоединять разделы диска к файловой системе во время работы операционной системы.

Команда MOUNTVOL позволяет создавать, удалять и просматривать точки подключения томов (точки монтирования) в командной строке Windows. Точки монтирования доступны при использовании файловой системы NTFS. В среде операционных систем семейства Windows, существует два вида точек монтирования: точка монтирования каталога (junction point) и точка монтирования тома (volume mount point). Создание точек монтирования первого типа осуществляется через консольную команду mklink /J, создание точек монтирования второго типа — через команду mountvol.

При вводе команды mountvol без параметров отображается перечень томов с точками монтирования:

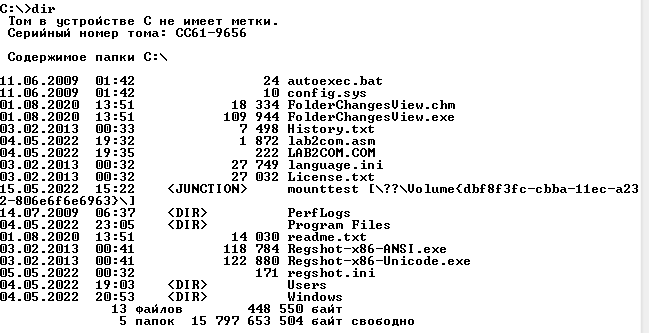


* mountvol [<диск>:]<путь> <имя тома> - подключить том с указным именем в качестве папки по указанному пути



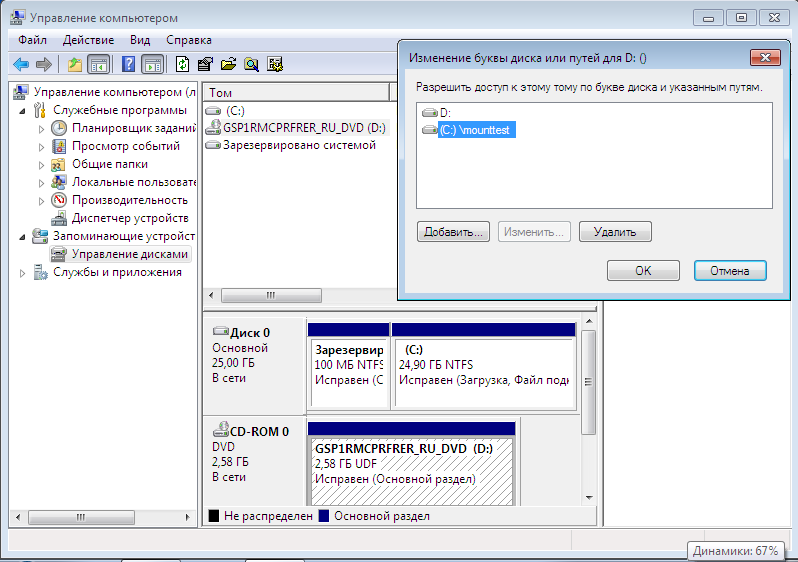
* mountvol [<диск>:]<путь> /l - отобразить список подключенных томов для папки по указанному пути





Как можно заметить была создана точка соединения (Junction Point) NTFS, находящаяся на диске C:\ и указывающая на диск D:\

Так же это можно увидеть и в «Управлении дисками»



*Жёсткие и символические связи*

**Hard Links** (жесткие ссылки) – могут указывать только на локальный файл, но не на папку. Такой файл – это ссылка на другой файла на этом же диске без фактического дублирования самого файла. У него отображается такой же размер и свойства, как у целевого файла (но реальное место на диске он не занимает)

Использование команды fsutil hardlink create позволяет создать несколько логических файлов в различных местах. При этом все они будут ссылаться на один и тот же физический файл. Использование постоянных ссылок позволяет прозрачно для пользователей перенаправить обращение к файлу в одном месте к данным, фактически хранящимся в другом месте.

При удалении файла он не удаляется физически до тех пор, пока все жёсткие ссылки, указывающие на него, не будут удалены. Порядок их удаления значения не имеет. При удалении жёсткой ссылки в корзину количество ссылок у файла сохраняется.

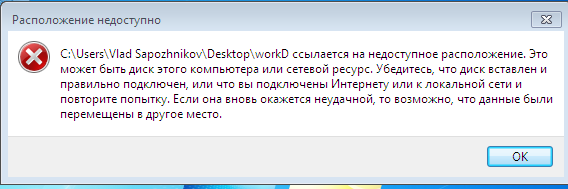
При редактировании по одной ссылок изменяется и сам файл, на который указывает ссылка. При перемещении файла ссылка на него не теряется, т.к. связь идет через индексный дескриптор.



**Symbolic Links** (мягкая ссылка, симлинк) – могут указывать на локальный файл, папку и сетевой каталог на удаленном компьютере (UNC), поддерживаются относительные пути.



При перемещении файла/каталога, на который указывает SymLink сама ссылка не меняется и указывает на недоступное расположение:



При удалении SymLink, файл на который указывает ссылка не удаляется.