Файловые системы Windows

Основные свойства файловой системы NTFS:

1. Поддержка больших файлов и больших дисков (объем до 2^{64} байт).
2. Восстанавливаемость после сбоев и отказов программ и аппаратуры управления дисками.
3. Высокая скорость операций, в том числе для больших дисков.
4. Низкий уровень фрагментации, в том числе для больших дисков.
5. Гибкая структура, допускающая развитие за счет добавления новых типов записей и атрибутов файлов с сохранением совместимости с предыдущими версиями ФС.
6. Устойчивость к отказам дисковых накопителей.
7. Поддержка длинных символьных имен.
8. Контроль доступа к каталогам и отдельным файлам.

Структура тома NTFS

Основой структуры тома является *главная таблица файлов* (***Master File Table, MFT***), которая содержит одну или несколько записей для каждого файла тома и одну запись для самой себя (размер записи – 1, 2 или 4 Кбайт).

Том состоит из последовательности кластеров, порядковый номер кластера в томе – ***логический номер кластера*** *(Logical Cluster Number, LCN).*

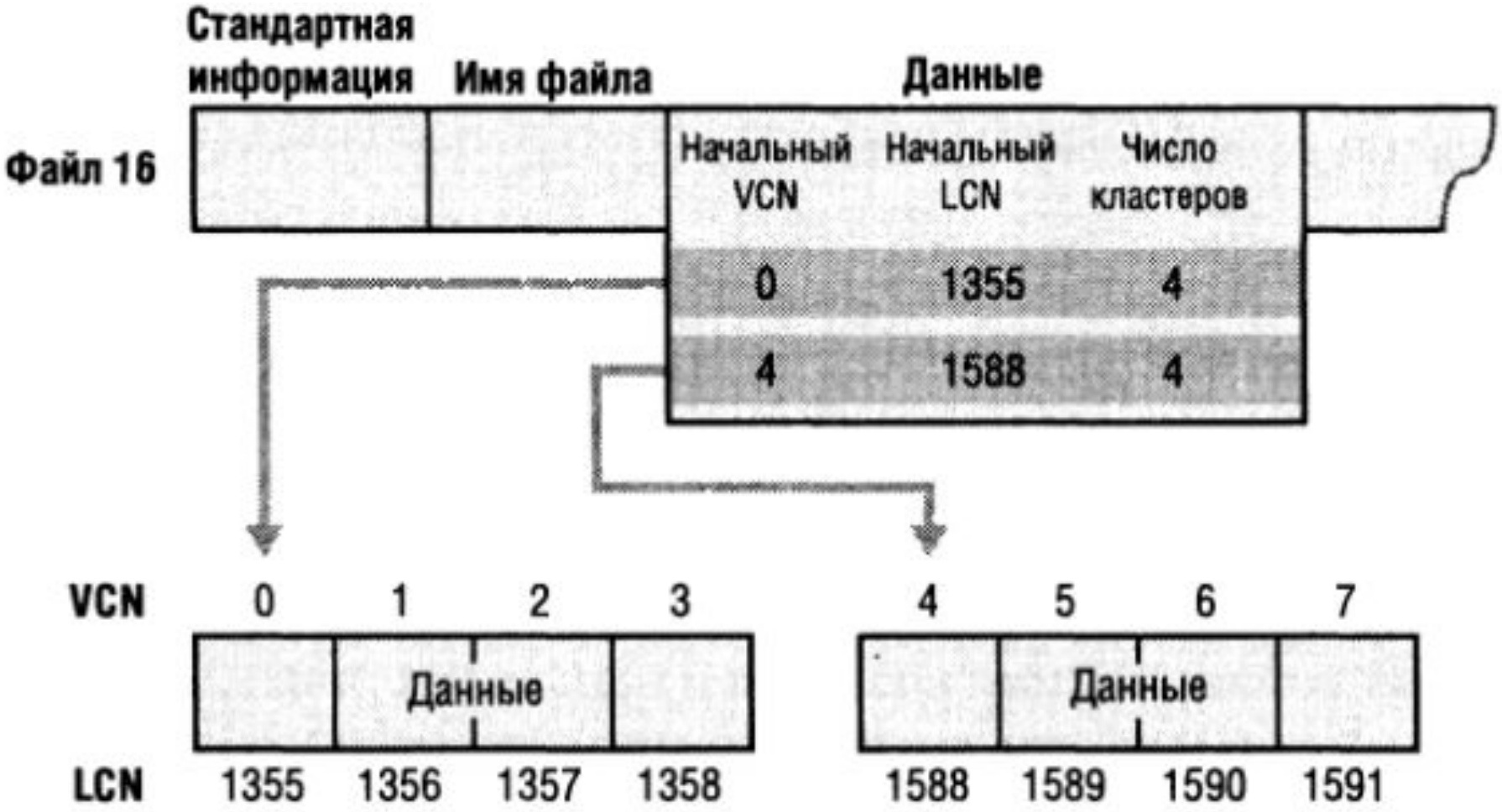
Файл состоит из последовательности кластеров, порядковый номер кластера внутри файла называется ***виртуальным номером кластера*** (*Virtual Cluster Number, VCN*). Размер кластера от 512 байт до 64 Кбайт.

Базовая единица распределения дискового пространства – ***отрезок*** – непрерывная область кластеров.

Адрес отрезка – (LCN, k), k –количество кластеров в отрезке.

Адрес файла (или его части) – (VCN, LCN, k).

Файл целиком размещается в записи таблицы MFT (если позволяет размер). В противном случае в записи MFT хранится резидентная часть файла (некоторые его атрибуты), а остальная часть файла хранится в отдельном отрезке тома или нескольких отрезках.


Стандартная информация
Имя файла
Данные
Файл 16
Начальный VCN
Начальный LCN
Число кластеров
0
1355
4
4
1588
4
VCN
0
1
2
3
4
5
6
7
Данные
Данные
LCN
1355
1356
1357
1358
1588
1589
1590
1591

Загрузочный блок содержит стандартный блок параметров BIOS, количество блоков в томе, начальный логический номер кластера основной и зеркальной копии MFT.

0. Описание MFT, в том числе адреса всех ее отрезков. 1. Зеркальная копия 3-х первых записей MFT.

2. Журнал для восстановления файловой системы.

3. Файл тома (имя, версия и др. информация).
4. Таблица определения атрибутов.
5. Индекс корневого каталога.
6. Битовая карта кластеров.
7. Загрузочный сектор раздела.
8. Список дефектных кластеров.
9. Описатели защиты файлов.
10. Таблица квот.
11. Таблица преобразования регистра символов (для Unicode).
12 – 15 – зарезервировано.

№ записи MFT	Имя файла	Описание
0	$MFT	Сама MFT
1	$MFTmirr	Зеркальная копия первых трех записей MFT
2	$LogFile	Список транзакций, который используется для восстановления файловой системы после сбоев
3	$Volume	Имя тома, версия NTFS и другая информация о томе
4	$AttrDef	Таблица имен, номеров и описаний атрибутов
5	$.	Индекс корневого каталога
6	$Bitmap	Битовая карта кластеров
7	$Boot	Загрузочный сектор раздела
8	$BadClus	Файл плохих кластеров
9	$Quota	Таблица квот
10	$Upcase	Таблица преобразования регистра символов для кодировки Unicode
11 − 15	Зарезервированы для будущего использо	Не используется

Структура файлов NTFS

Файлы и каталоги состоят из набора атрибутов. Атрибут содержит следующие поля: тип, длина, имя (образуют заголовок) и значение.

Системные атрибуты:

1. Стандартная информация (сведения о владельце, флаговые биты, время создания, время обновления и др.).

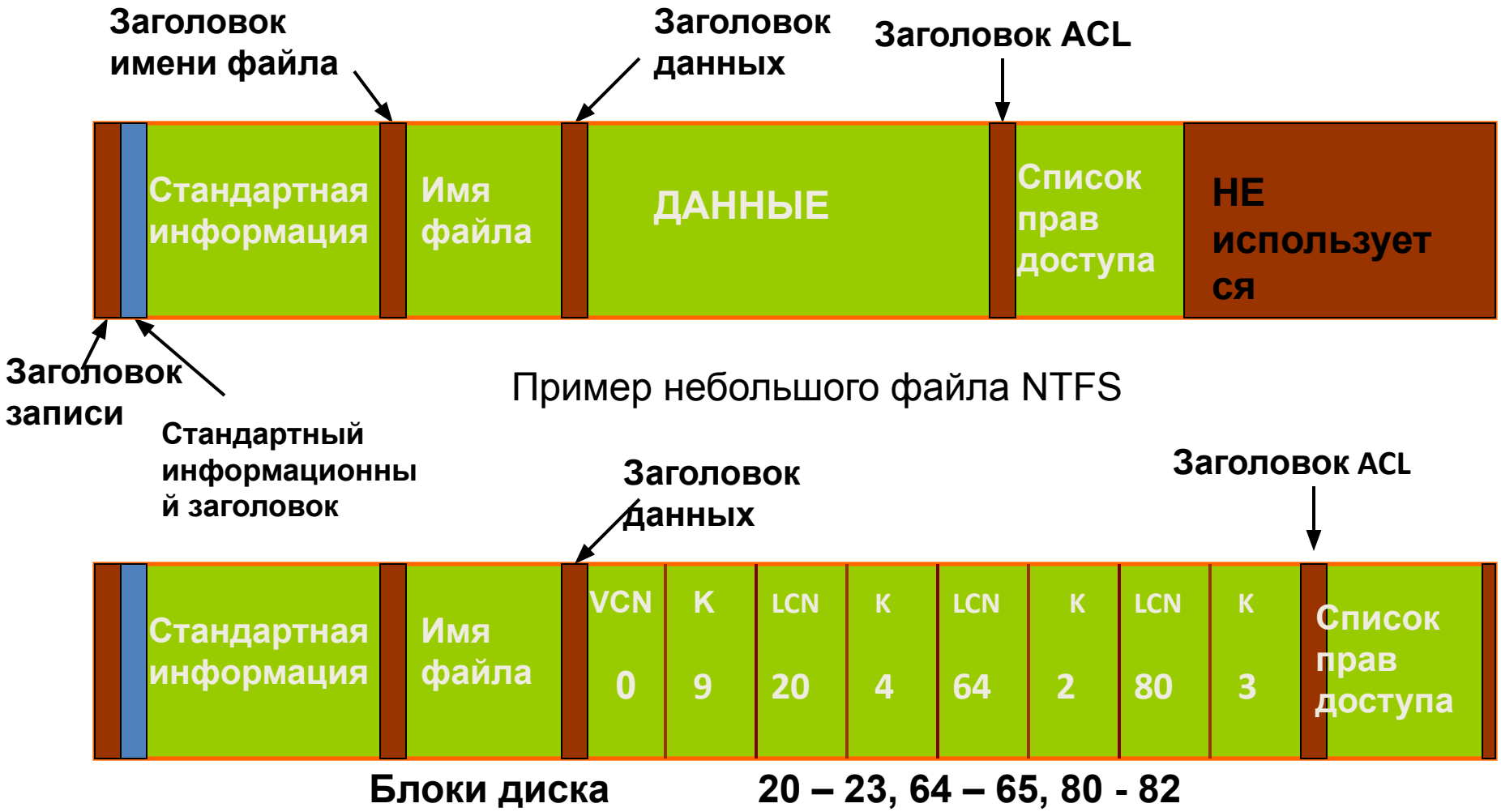
2. Имя файла в кодировке Unicode, м.б. повторено для имени MS DOS.
3. Список атрибутов (содержит ссылки на номера записей MFT, где расположены атрибуты), используется для больших файлов.

4. Версия – номер последней версии файла.
5. Дескриптор безопасности – список прав доступа ACL.
6. Версия тома –используется в системных файлах тома.
7. Имя тома.
8. Битовая карта MFT – карта использования блоков тома.
9. Корневой индекс – используется для поиска файлов в каталоге.
10. Размещение индекса – нерезидентная часть индексного списка (для больших файлов).

11. Идентификатор объекта – 64-разрядный идентификатор файла, уникальный для данного тома.
12. Данные файла.

13. Точка повторного анализа (монтирование и симв. ссылки)

- **Системный набор включает следующие атрибуты:**
- *Attribute List (список атрибутов)* – список атрибутов, из которых состоит файл; содержит ссылки на номер записи MFT, где расположен каждый атрибут; этот атрибут нужен только в том случае, если атрибуты файла не умещаются в основной записи и занимают дополнительные записи MFT;
- *File Name (имя файла)* – этот атрибут содержит длинное имя файла в формате Unicode, а также номер входа в таблице MFT для родительского каталога; если этот файл содержится в нескольких каталогах, то у него будет несколько атрибутов типа *File Name;* этот атрибут всегда должен быть резидентным;
- *Security Descriptor (дескриптор безопасности)* – этот атрибут содержит информацию о защите файла: список прав доступа (Access Control List – ACL) и поле аудита, которое определяет, какого рода операции над этим файлом нужно регистрировать;
- *Data (данные)* – содержит обычные данные файла;
- *Index Root (корень индекса)* – корень B-дерева, используемого для поиска файлов в каталоге;
- *Index Allocation (размещение индекса)* – нерезидентные части индексного списка B-дерева;
- *Standard Information (стандартная информация)* – этот атрибут хранит всю остальную стандартную информацию о файле, которую трудно связать с каким-либо из других атрибутов файла, например, время создания файла, время обновления и другие.



Пример небольшого файла NTFS

Пример большого файла NTFS

Файлы NTFS в зависимости от способа размещения делятся на небольшие, большие, очень большие и сверхбольшие.

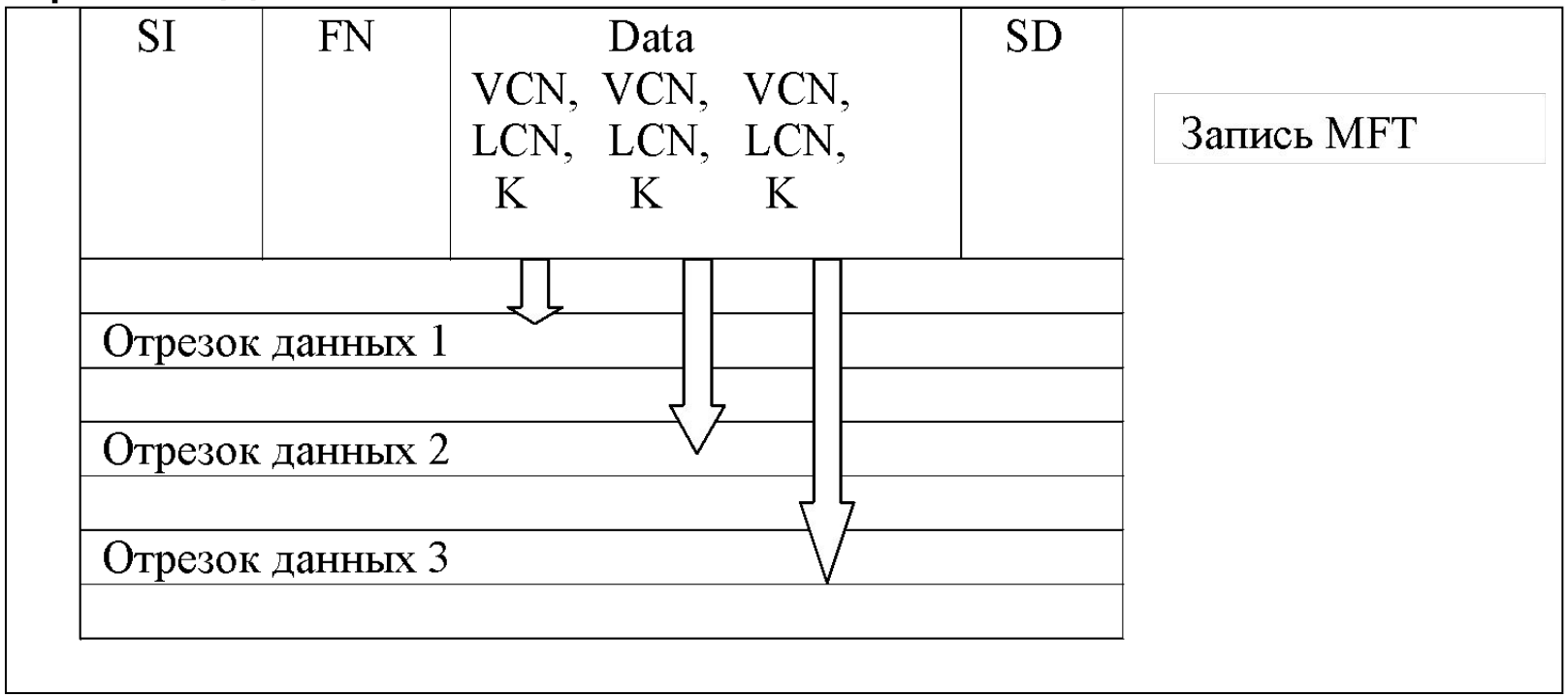
Небольшие файлы (small).

Если файл имеет небольшой размер, то он может целиком располагаться внутри одной записи MFT, имеющей, например, размер 4 Кбайт. Небольшие файлы NTFS состоят по крайней мере из следующих атрибутов:

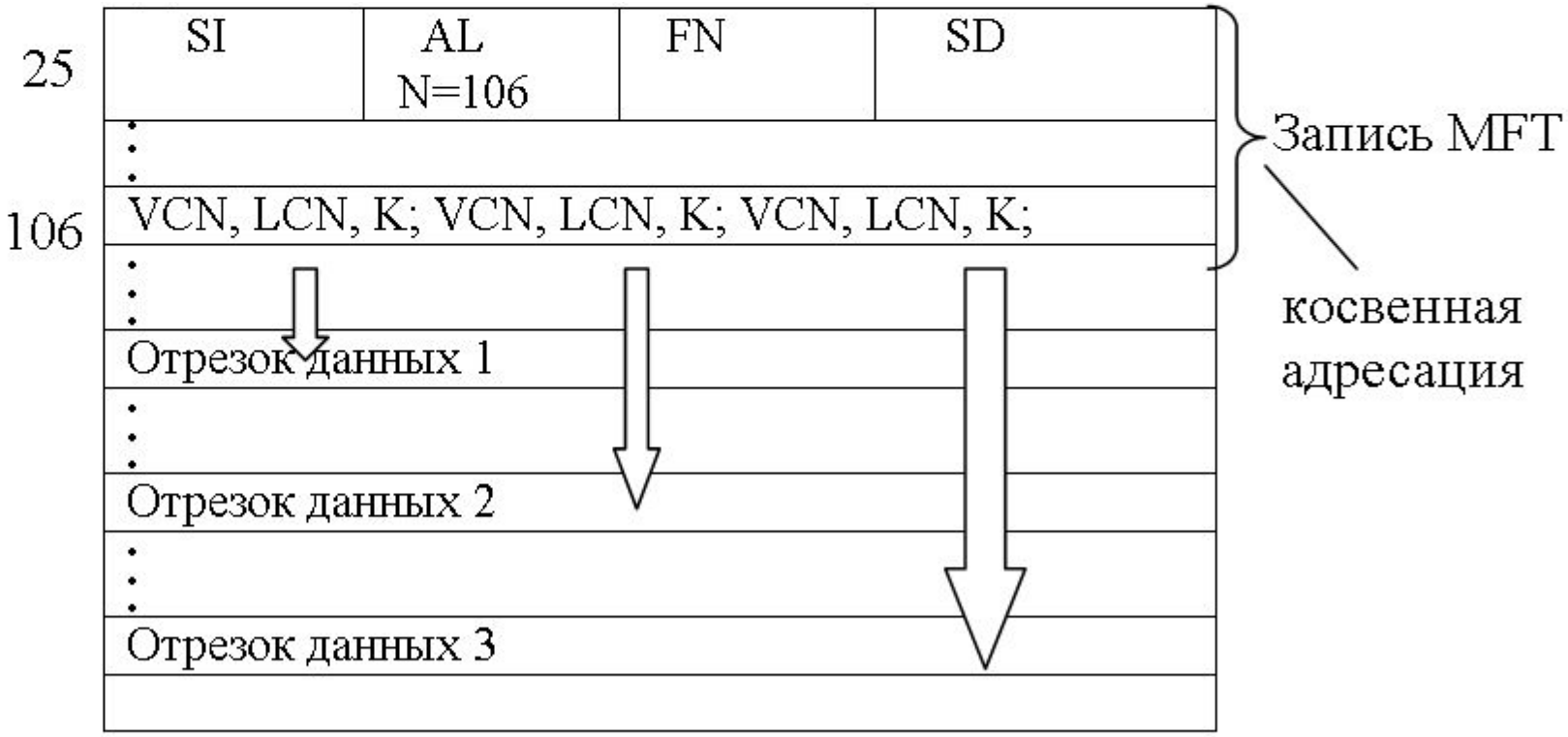
1. стандартная информация (SI – standard information);
2. имя файла (FN – file name);
3. данные (Data);
4. дескриптор безопасности (SD – security descriptor).

Большие файлы (large).

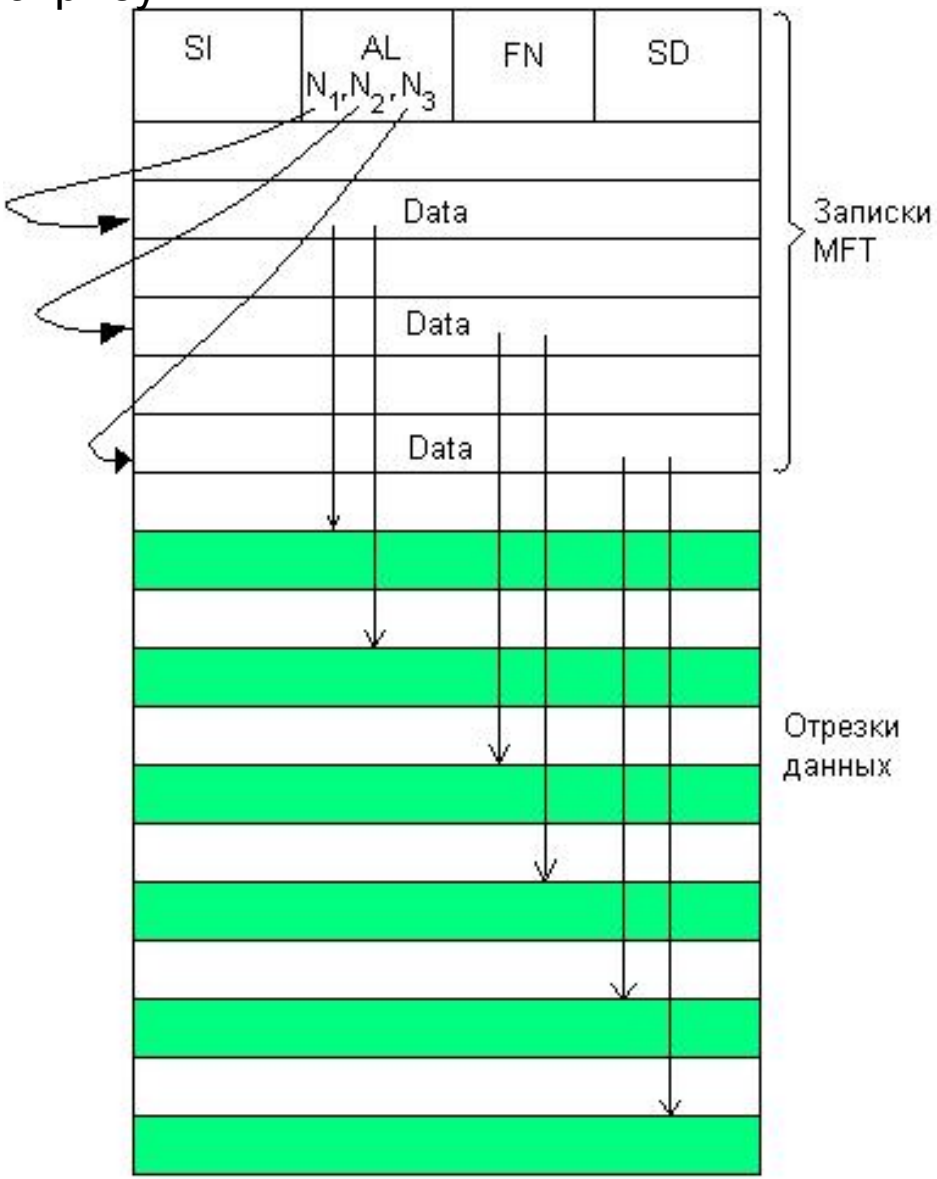
Если данные файла не помещаются в одну запись MFT, то этот факт отражается в заголовке атрибута *Data,* который содержит признак того, что этот атрибут является нерезидентным, то есть находится в отрезках вне таблицы MFT. В этом случае атрибут *Data* содержит адресную информацию (VCN, LCN, k) каждого отрезка данных.



Очень большие файлы. Если файл настолько велик, что его атрибут данных, хранящий адреса не резидентных отрезков данных, не помещается в 1-й записи, то этот атрибут помещается в 2-ю запись MFT, а ссылка на такой атрибут помещается в основную запись файла. Она (ссылка) содержится в атрибуте Attribute List. Сам атрибут данных по-прежнему содержит адреса нерезидентных отрезков данных.



Сверх большие файлы. Для этих файлов в атрибуте Attribute List можно указать несколько атрибутов, расположенных в дополнительных записях MFT. Кроме того, можно использовать двойную косвенную адресацию, когда нерезидентный атрибут будет ссылаться на другие не резидентные атрибуты.

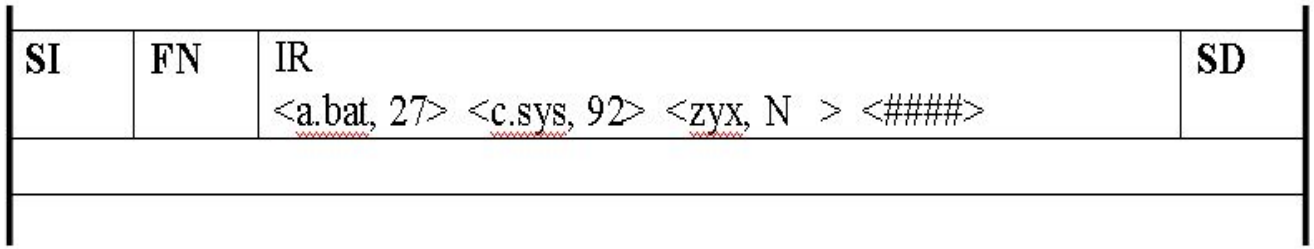


Каталоги NTFS

- Каждый каталог NTFS представляет собой один вход в таблицу MFT, который содержит атрибут *Index Root.* Индекс содержит список файлов, входящих в каталог. Индексы позволяют сортировать файлы для ускорения поиска, основанного на значении определенного атрибута. Обычно в файловых системах файлы сортируются по имени. NTFS позволяет использовать для сортировки любой атрибут, если он хранится в резидентной форме.
- Имеются две формы хранения списка файлов.

Небольшие каталоги (small indexes)

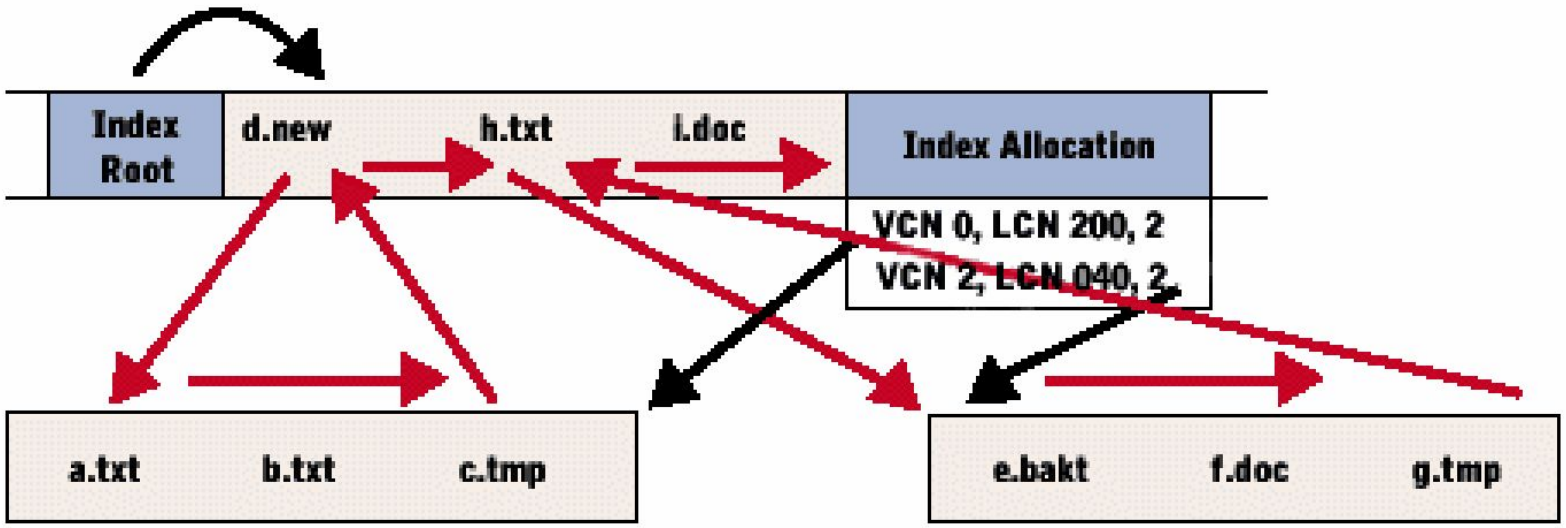
- Если количество файлов в каталоге невелико, то список файлов может быть резидентным в записи в MFT, являющейся каталогом
- Для резидентного хранения списка используется единственный атрибут – *Index Root.* Список файлов содержит значения атрибутов файла. По умолчанию – это имя файла, а также номер записи MFT, содержащей начальную запись файла.



– признак конца файлов

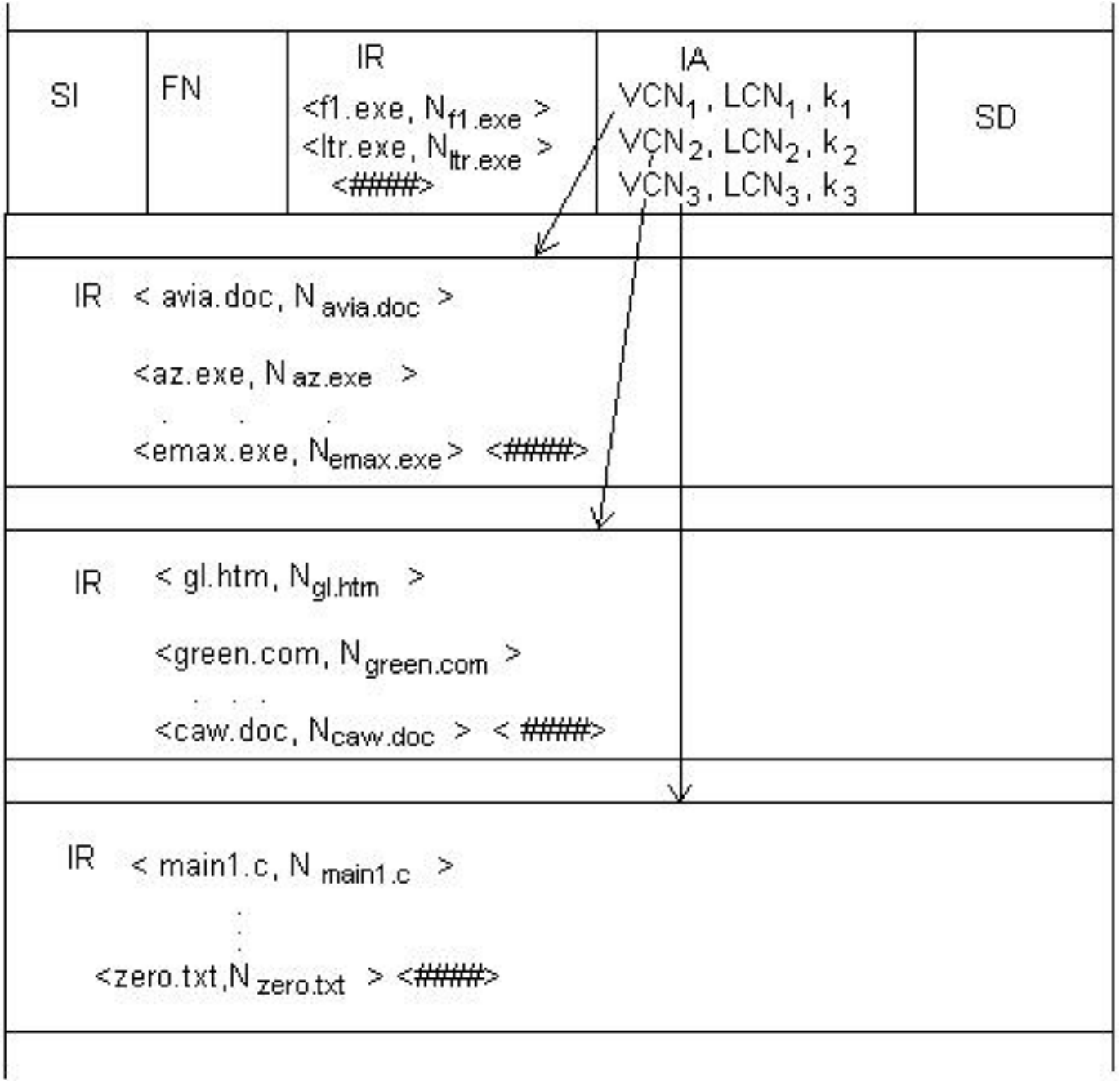
Большие каталоги (large indexes)

- По мере того как каталог растет, список файлов может потребовать нерезидентной формы хранения. Однако часть списка всегда остается резидентной в корневой записи каталога в таблице MFT. Имена файлов резидентной части списка файлов являются узлами так называемого B-дерева (сбалансированного дерева). Остальные части списка файлов размещаются вне MFT. Для их поиска используется специальный атрибут *Index Allocation,* представляющий собой адреса отрезков, хранящих остальные части списка файлов каталога. Узлы B-дерева делят весь список файлов на несколько групп. Имя каждого файла-узла является именем последнего файла в соответствующей группе.



Индексный буфер

Индексный буфер

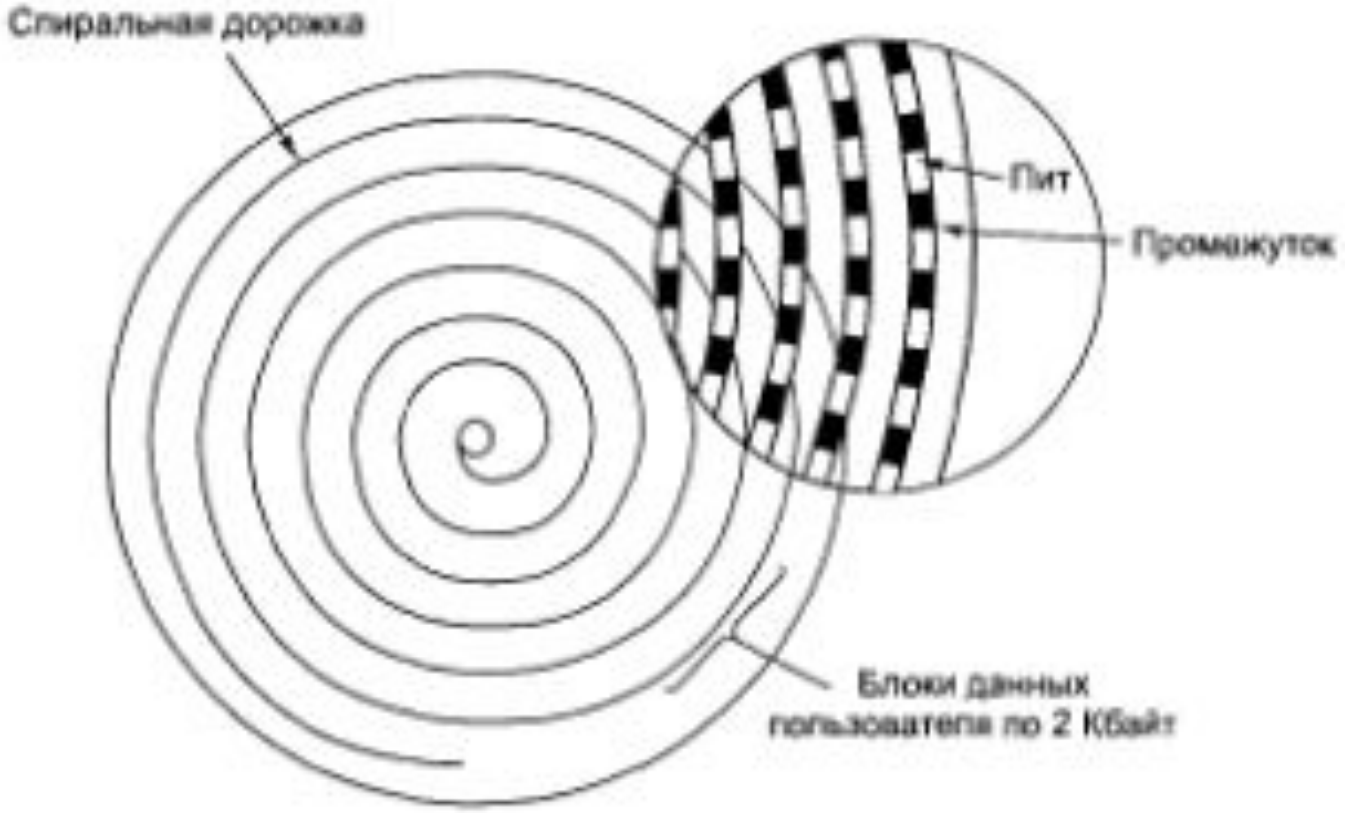

SI
FN
IR
<f1.exe, $N_{f1.exe}$ >
<ltr.exe, $N_{ltr.exe}$ >
<#####>
IA
VCN_1, LCN_1, k_1
VCN_2, LCN_2, k_2
VCN_3, LCN_3, k_3
SD
IR < avia.doc, $N_{avia.doc}$ >
<az.exe, $N_{az.exe}$ >
<emax.exe, $N_{emax.exe}$> <#####>
IR < gl.htm, $N_{gl.htm}$ >
<green.com, $N_{green.com}$ >
<caw.doc, $N_{caw.doc}$ > < #####>
IR < main1.c, $N_{main1.c}$ >
<zero.txt,$N_{zero.txt}$ > <#####>

Файловая система CDFS

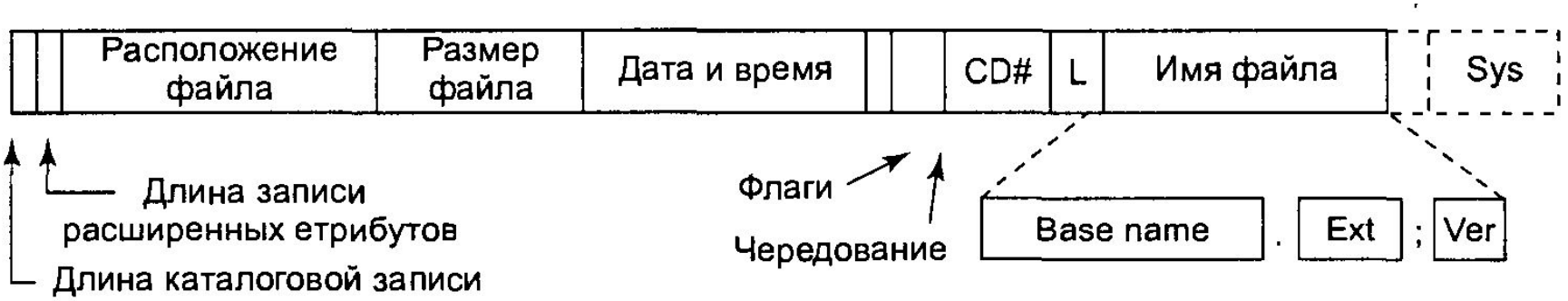
В Windows обеспечивается поддержка файловой системы CDFS, отвечающей стандарту ISO’9660, описывающему расположение информации на CD-ROM.

ISO 9660 — стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Цель стандарта - обеспечить совместимость носителей под разными ОС (Unix, Mac OS, Windows).

•



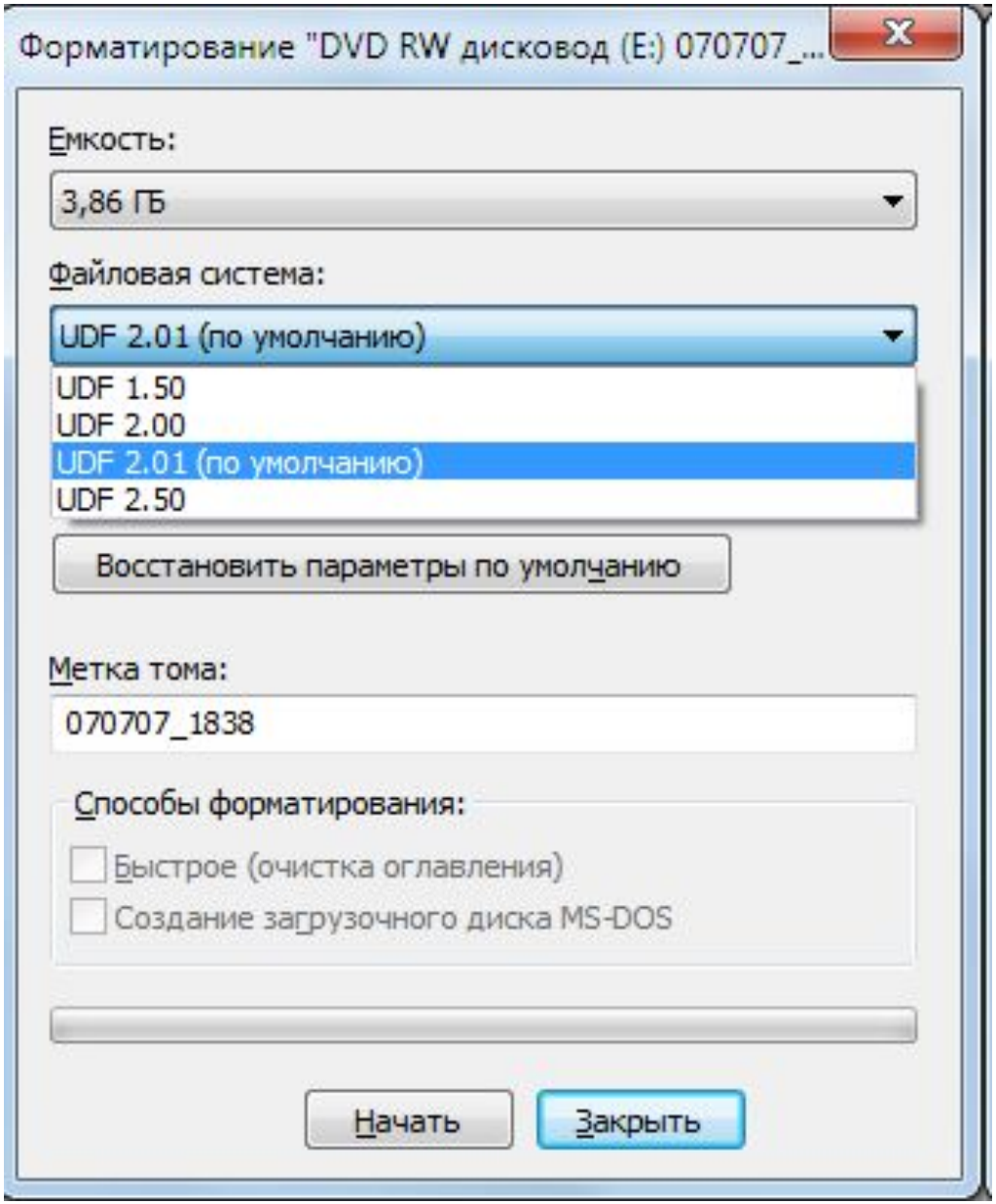
Структура записи компакт-диска или CD-ROM



Каталоговая запись стандарта ISO 9660

Universal Disk Format (UDF)

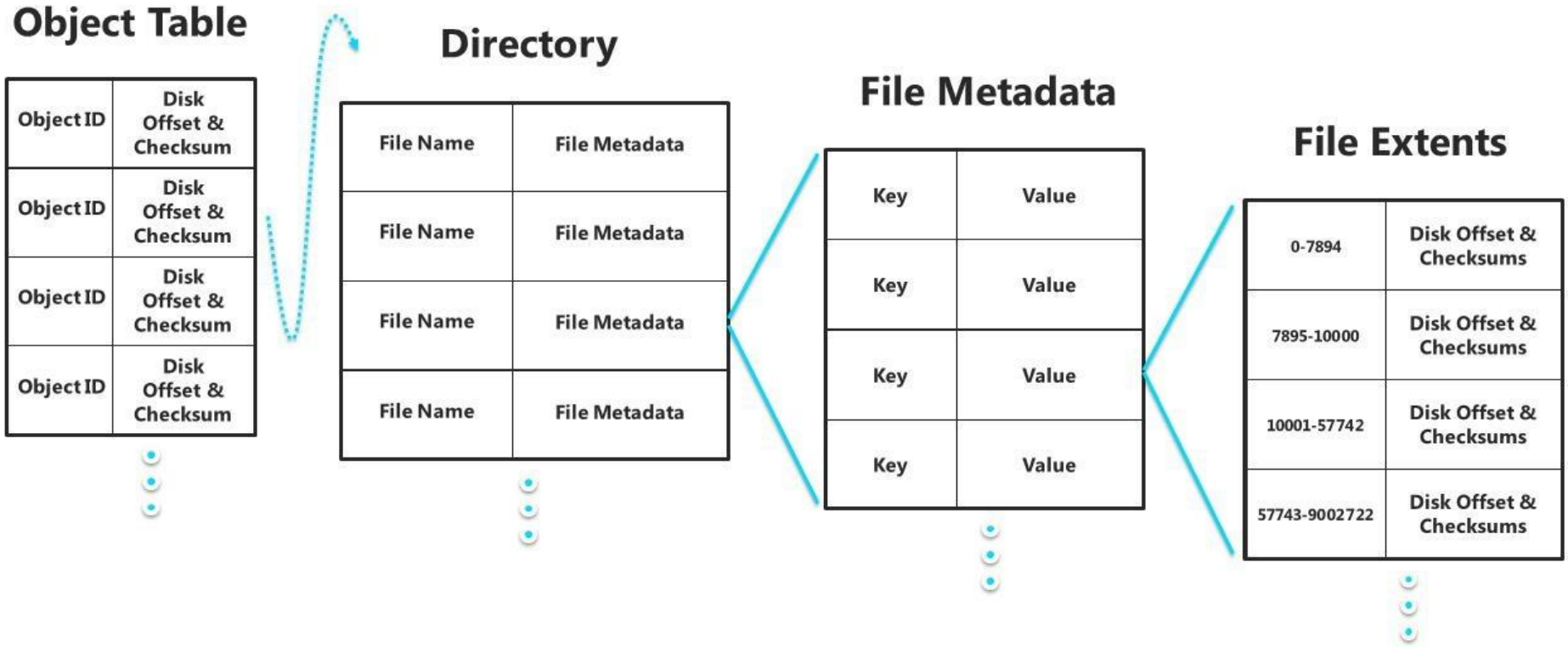
- спецификация формата файловой системы, независимой от ОС для хранения файлов на оптических носителях (CD-ROM и DVD). UDF является реализацией стандарта ISO/IEC 13346. Формат UDF призван заменить ISO 9660.
- UDF учитывает возможность выборочного стирания некоторых файлов на перезаписываемых носителях CD-RW, освобождая место на диске.
- Метаданные файловой системы, такие, как корневая директория, могут находиться где угодно на диске, «корень» метаданных должен находиться в двух из трех следующих мест: сектор 256, сектор (N-257) и (N-1), где N — размер дорожки.


Форматирование "DVD RW дисковод (E:) 070707_...
Емкость:
3,86 ГБ
Файловая система:
UDF 2.01 (по умолчанию)
UDF 1.50
UDF 2.00
UDF 2.01 (по умолчанию)
UDF 2.50
Восстановить параметры по умолчанию
Метка тома:
070707_1838
Способы форматирования:
Быстрое (очистка оглавления)
Создание загрузочного диска MS-DOS
Начать
Закрыть

ReFS (Resilient file system)

- предварительное название Protogon — файловая система, используемая в Windows Server 2012, Windows Server 2012 R2, бета-версиях Microsoft Windows 8, Windows 8.1. Является дальнейшим развитием NTFS. Protogon поддерживает точки повторной обработки (reparse points) — технологию, которая ранее содержалась только в файловой системе NTFS. Через точки повторной обработки реализована поддержка символьных ссылок и точек монтирования в Windows, так что Protogon также поддерживает их.
- Protogon не поддерживается Windows 7 и более ранними системами.

1. Для открытия, закрытия, а также чтения файлов, файловая система ReFS использует те же самые интерфейсы доступа к данным API, что и файловая система NTFS.
2. Изменения в области создания структур папок и файлов и управления ими обеспечивают автоматическое исправление ошибок объектов файловой системы и самой системы, максимальное масштабирование, работу в режиме постоянного подключения (Always Online).
3. Для всех этих нововведений, используется концепция B+ -деревьев. Данная концепция заключается в том, что папке в данной файловой системе структурированы в виде обычных таблиц, а файлы выступают в роли записей данной таблицы.
4. Ядром файловой системы ReFS является таблица объектов, которая называется центральным каталогом. В ней перечислены все таблицы в системе.
5. Свободное место на диске описывается 3 отдельными иерархическими таблицами для малых, средних и больших фрагментов свободного пространства.
6. Имена файлов и длина пути ограничена 32 килобайтами, для их хранения используется Unicode.


Object Table
Object ID
Disk Offset & Checksum
Object ID
Disk Offset & Checksum
Object ID
Disk Offset & Checksum
Object ID
Disk Offset & Checksum
Directory
File Name
File Metadata
File Name
File Metadata
File Name
File Metadata
File Name
File Metadata
File Metadata
Key
Value
Key
Value
Key
Value
Key
Value
File Extents
0-7894
Disk Offset & Checksums
7895-10000
Disk Offset & Checksums
10001-57742
Disk Offset & Checksums
57743-9002722
Disk Offset & Checksums

Сравнение файловых систем NTFS и ReFS

	NTFS	REFS
МАКСИМАЛЬНЫЙ РАЗМЕР ФАЙЛА	16 Тбайт	18,3 Эбайт
МАКСИМАЛЬНЫЙ РАЗМЕР ТОМА	18,4 Эбайт	402 Эбайт
МАКСИМАЛЬНОЕ ЧИСЛО ФАЙЛОВ В ПАПКЕ (ПРИБЛИЗИТЕЛЬНО)	4,3 млрд	18 трлн
МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СИМВОЛОВ В ИМЕНИ ФАЙЛА	255	32 767
МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СИМВОЛОВ В ИМЕНИ ПУТИ	255	32 767

- Как и файловая система NTFS, система ReFS по своему же принципу различает информацию о файле (это метаданные), а также содержимое файла (это пользовательские данные), однако ReFS предоставляет защиту данных, как и тем, так и другим. Например, метаданные, используют защиту контрольных сумм. Такую защиту можно предоставить и пользовательским данным. Эти контрольные суммы, размещаются на жестком диске, которые находятся на безопасном доступе друг от друга, это делается для того, чтобы при возникновении какой-либо ошибки, данные можно было восстановить.

- **Передача данных из файловой системы NTFS в ReFS**
- Можно ли будет без всяких проблем передавать данные из файловой системы, например Windows XP в файловую систему Windows 8 (то есть из NTFS в ReFS) и соответственно наоборот. В самой компании Microsoft по этому вопросу отвечают следующим образом: что никакой встроенной функции преобразования формат не будет, но производить простое копирование можно будет.
- Использовать файловую систему ReFS на сегодняшний день, можно как крупный диспетчер данных для сервера. Исходя из этого, невозможно запустить Windows 8 с диска под управлением новой файловой системы ReFS.
- Внешних накопителей с файловой системой ReFS пока что не предвидится, будут только внутренние накопители. Со временем файловая система ReFS будет дополняться огромным количеством различных функций и сможет заменить старую файловую систему.

Восстанавливаемость файловых систем

- Восстанавливаемость файловой системы — это свойство, которое гарантирует, что в случае отказа питания или краха системы, когда все данные в оперативной памяти безвозвратно теряются, все начатые файловые операции будут либо успешно завершены, либо отменены без отрицательных последствий для работоспособности файловой системы.
- Для восстановления некорректных файловых систем, использующих кэширование диска, в операционных системах предусматриваются специальные утилиты, такие как, ScanDisk для FAT или Chkdisk для NTFS.
- Проблемы, связанные с восстановлением файловой системы, могут быть решены при помощи техники протоколирования транзакций. В системе должны быть определены транзакции (transactions) — неделимые работы, которые не могут быть выполнены частично. Они либо выполняются полностью, либо вообще не выполняются. Любая операция над файлом (создание, удаление, запись, чтение и т. д.) может быть представлена в виде некоторой последовательности подопераций.

Упреждающее протоколирование транзакций.

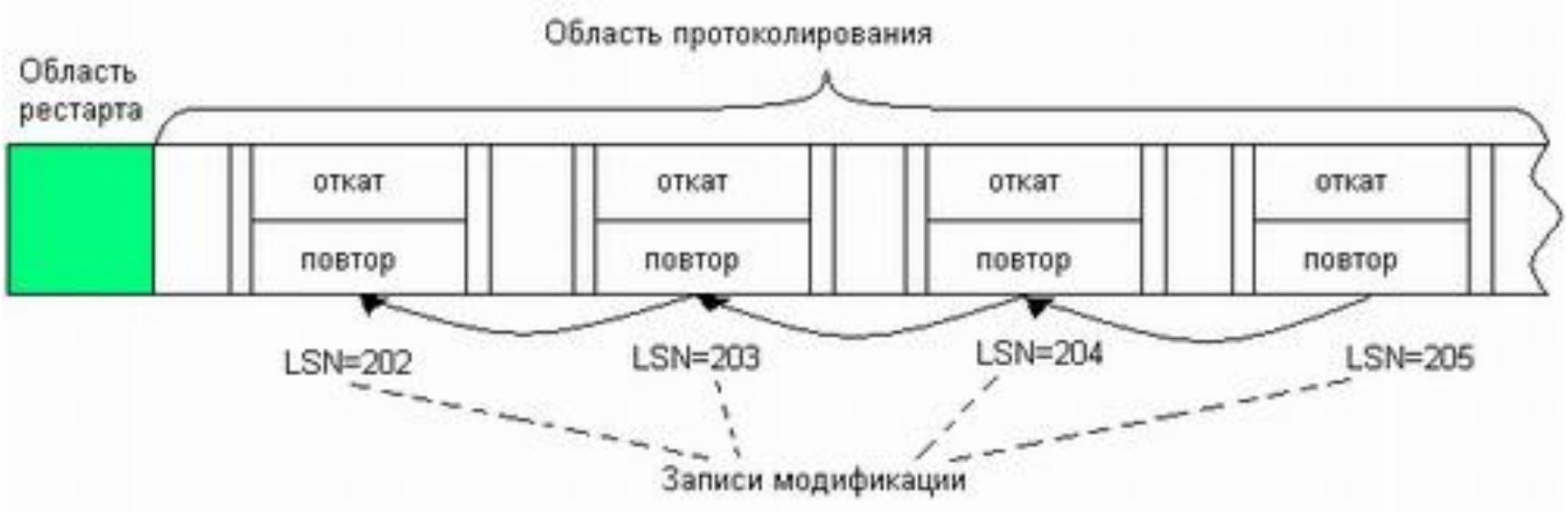
1. Перед изменением какого-либо блока данных на диске или в дисковом кэше производится запись в специальный системный файл — журнал транзакций (log file), где отмечается, какая транзакция делает изменения, какой файл и блок изменяются и каковы старое и новое значения изменяемого блока. Только после успешной регистрации всех подопераций в журнале делаются изменения в исходных блоках.
2. Если транзакция прерывается, то информация журнала регистрации используется для приведения файлов, каталогов и служебных данных файловой системы в исходное состояние, то есть производится *откат*.
3. Если транзакция фиксируется, то и об этом делается запись в журнал регистрации, но новые значения измененных данных сохраняются в журнале еще некоторое время, чтобы сделать возможным повторение транзакции, если это потребуется.

Восстанавливаемость файловой системы NTFS

Файловая система NTFS является восстанавливаемой файловой системой, однако восстанавливаемость обеспечивается *только для системной информации файловой системы*, то есть каталогов, атрибутов безопасности, битовой карты занятости кластеров и других системных файлов. Сохранность данных пользовательских файлов, работа с которыми выполнялась в момент сбоя, в общем случае не гарантируется.

Журнал регистрации транзакций в NTFS делится на две части:

- область рестарта
- область протоколирования



- Область рестарта содержит информацию о том, с какого места необходимо будет начать читать журнал транзакций для проведения процедуры восстановления системы после сбоя или краха ОС - указатель на определенную запись в области протоколирования. Для надежности в файле журнала регистрации хранятся две копии области рестарта.
- Область протоколирования содержит записи обо всех изменениях в системных данных файловой системы, произошедших в результате выполнения транзакций в течение некоторого периода. Все записи идентифицируются логическим последовательным номером LSN (Logical Sequence Number). Записи о подоперациях, принадлежащих одной транзакции, образуют связанный список: каждая последующая запись содержит номер предыдущей записи.
- Заполнение области протоколирования идет циклически: после исчерпания всей памяти, отведенной под область протоколирования, новые записи помещаются на место самых старых.

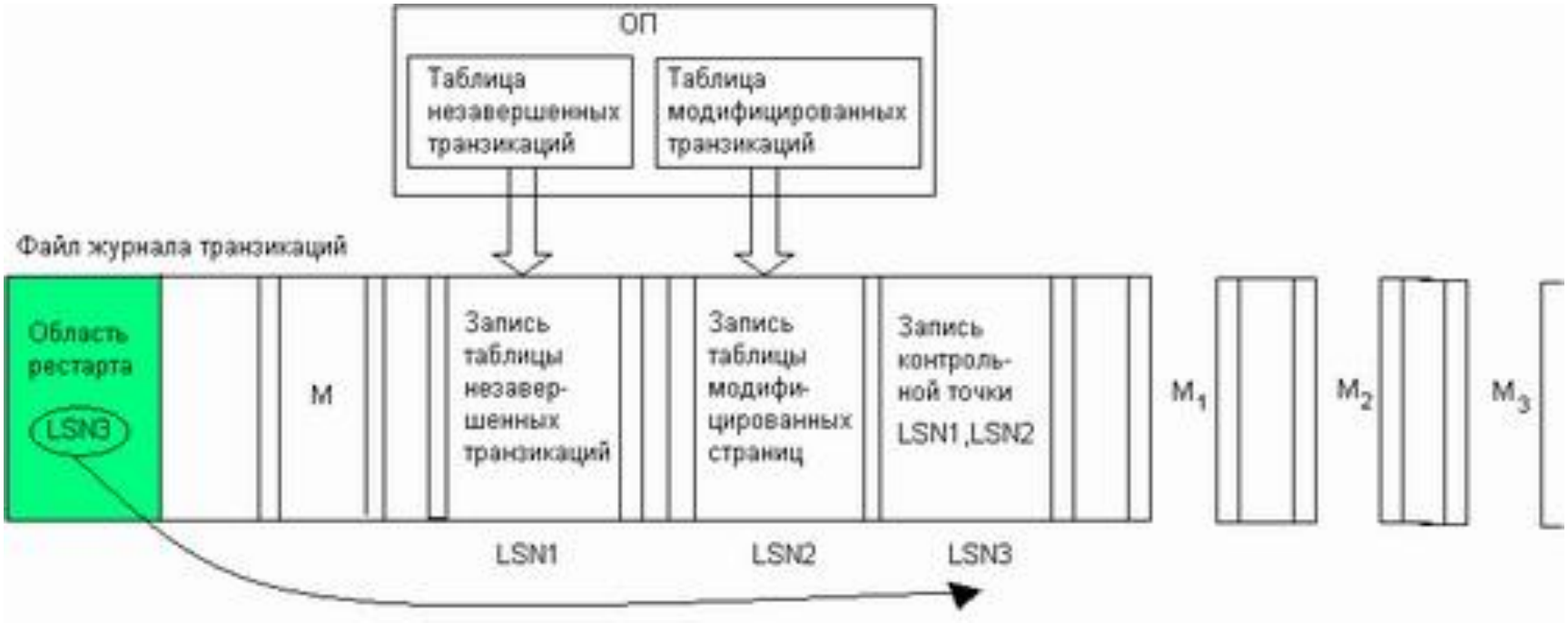
- Существует несколько типов записей в журнале транзакций: запись модификации, запись контрольной точки, запись фиксации транзакции, запись таблицы модификации, запись таблицы модифицированных страниц.

Структура записи модификации

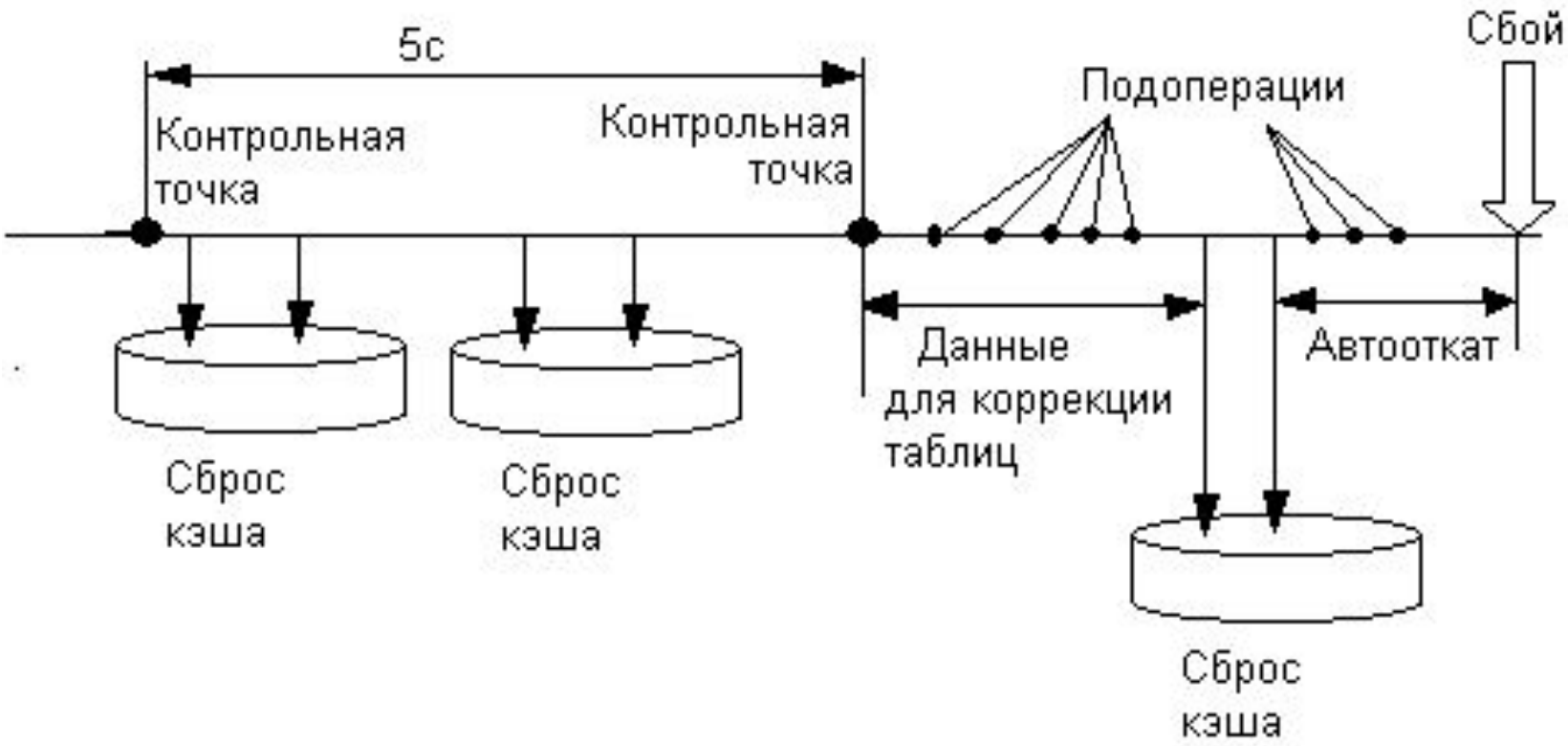
Запись модификации	**Информация для повторения транзакции**	**Информация для отката транзакции**
LSN=202	Выделить и инициировать запись для нового файла lotus.doc из таблицы MFT	Удалить запись о файле lotus.doc в таблице MFT
LSN=203	Добавить имя файла в индекс	Исключить имя файла из индекса
LSN=204	Установить биты 3-9 в битовой карте	Обнулить биты 3-9 в битовой карте

- Журнал транзакций, как и все остальные файлы, кэшируется в буферах оперативной памяти и периодически сбрасывается на диск.
- Файловая система NTFS все действия с журналом транзакций выполняет только путем запросов к специальной службе LFS (Log File Service). Эта служба размещает в журнале новые записи, сбрасывает на диск все записи до некоторого заданного номера, считывает записи в прямом и обратном порядке.

Операция контрольная точка выполняется каждые 5 секунд



Асинхронность процессов сброса кэша и создания контрольных точек



Сравнение надежности файловых систем NTFS и ReFS

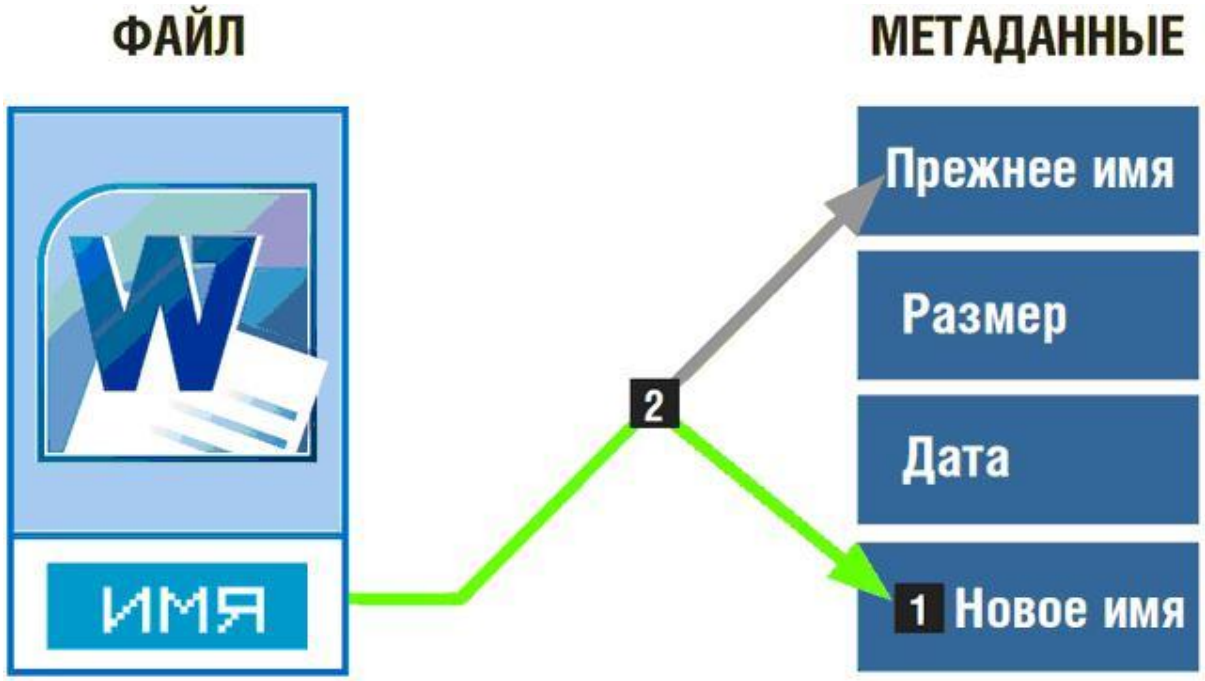
- С точки зрения архитектуры файловой системы ReFS имеет все требуемые инструменты для безопасного восстановления файлов даже после серьезного сбоя оборудования. Главный минус системы журналов в файловой системе NTFS и ей подобных — то, что обновление диска может повредить записанные ранее метаданные при сбое питания во время записи — этот эффект получил уже устойчивое название: т.н. «*оборванная запись*».
- Для предотвращения *оборванных записей*, разработчики из Microsoft избрали новый подход, при котором части структур метаданных содержат собственные идентификаторы, что позволяет проверить принадлежность структур; ссылки на метаданные содержат 64-битные контрольные суммы блоков, на которые производится ссылка.
- Всякое изменение структуры метаданных происходит в два этапа. Сперва создается новая (измененная) копия метаданных в свободном дисковом пространстве, и только после этого, в случае успеха, атомарной операцией обновления ссылка переводится со старой (неизмененной) на новую (измененную) область метаданных. Это позволяет обойтись без журналирования, автоматически сохраняя целостность данных.

- **Отказоустойчивость ReFS при повреждении диска. Система способна выявить все формы повреждений диска, включая потерянные или сохраненные не в том месте записи, а так же т. н. *битовый распад* (ухудшение состояния данных на носителе)**
- Когда включена опция «целостные потоки», ReFS проверяет по контрольным суммам также и содержимое файлов и всегда записывает изменения файлов в стороннем месте. Это дает уверенность в том, что существовавшие ранее данные не будут потеряны при перезаписи. Обновление контрольных сумм происходит автоматически при записи данных, так что если в ходе записи произойдет сбой, у пользователя останется доступная для проверки версия файла.

Сравнение файловых систем NTFS и ReFS на примере переименования файлов

1. Файловая система NTFS записывает в журнал, что файл должен быть переименован, там же она регистрирует и все остальные действия.
2. Только после того, как она запишет в журнал, что должно быть переименовано, выполняется переименование.
3. В конце выполнения операции, в журнале появляется сообщение о том, что было произведено успешное или неуспешное переименование файла.

1. В файловой системе ReFS, новое имя для файла или папки записывается в свободное место, при этом старое имя сразу не удаляется.
2. Как только новое имя будет записано, в файловой системе ReFS происходит создание ссылки на новое имя.



Как происходит переименование файла или папки в файловых системах NTFS и ReFS, при отказе системы

В файловой системе NTFS

- **1.** NTFS, как обычно, записывает запрос на изменение в Журнал.
- **2.** После этого из-за отказа питания процесс переименования прерывается, и не остается записи ни о прежнем, ни о новом именах.
- **3.** Происходит перезагрузка Windows.
- **4.** Вслед за этим запускается программа для исправления ошибок — Chkdisk.
- **5.** Только теперь с помощью Журнала при применении отката восстанавливается изначальное имя файла.

ReFS

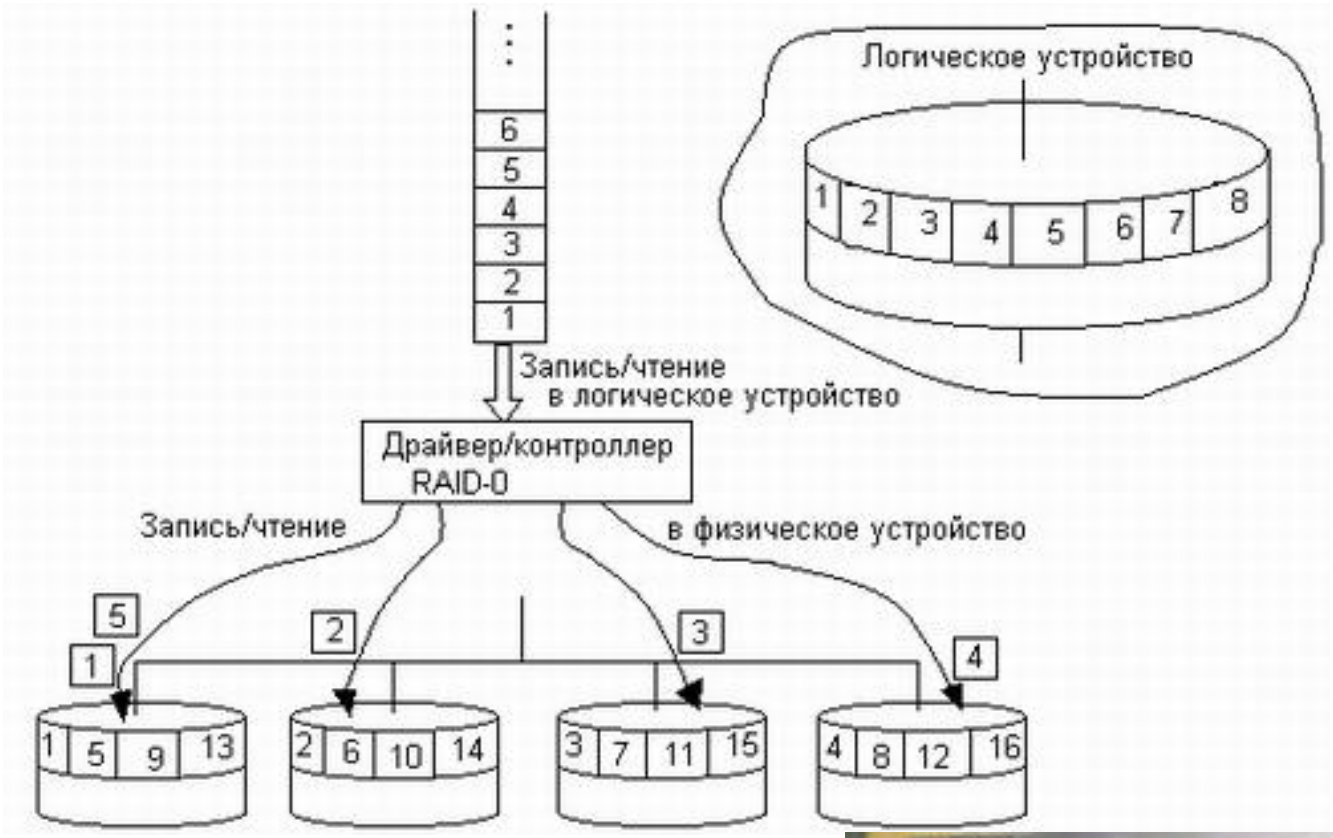
1. На первом этапе ReFS записывает новое имя в другом месте файловой системы, однако в этот момент
2. Электропитание прекращается.
3. Отказ приводит к автоматической перезагрузке Windows.
4. Стартует программа Chkdisk. Она анализирует файловую систему на наличие ошибок и при необходимости исправляет их. Между тем набор данных ReFS находится в стабильном состоянии. Прежнее имя файла снова становится действующим сразу после отказа питания.

RAID (Redundant Array of Inexpensive Disks, «избыточный массив недорогих дисков»)
RAID (англ. redundant array of independent disks — избыточный массив независимых дисков) — массив из нескольких дисков (запоминающих устройств), управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое.

- Калифорнийский университет в Беркли представил следующие уровни спецификации RAID, которые были приняты как стандарт де-факто:
- RAID 0 представлен как дисковый массив повышенной производительности, без отказоустойчивости.
- RAID 1 определён как зеркальный дисковый массив.
- RAID 2 зарезервирован для массивов, которые применяют код Хемминга.
- RAID 3 и 4 используют массив дисков с чередованием и выделенным диском чётности.
- RAID 5 используют массив дисков с чередованием и "невыделенным диском чётности".
- RAID 6 используют массив дисков с чередованием и двумя независимыми "чётностями" блоков.
- RAID 10 — RAID 0, построенный из RAID 1 массивов
- RAID 50 — RAID 0, построенный из RAID 5
- RAID 60 - RAID 0, построенный из RAID 6

RAID 0

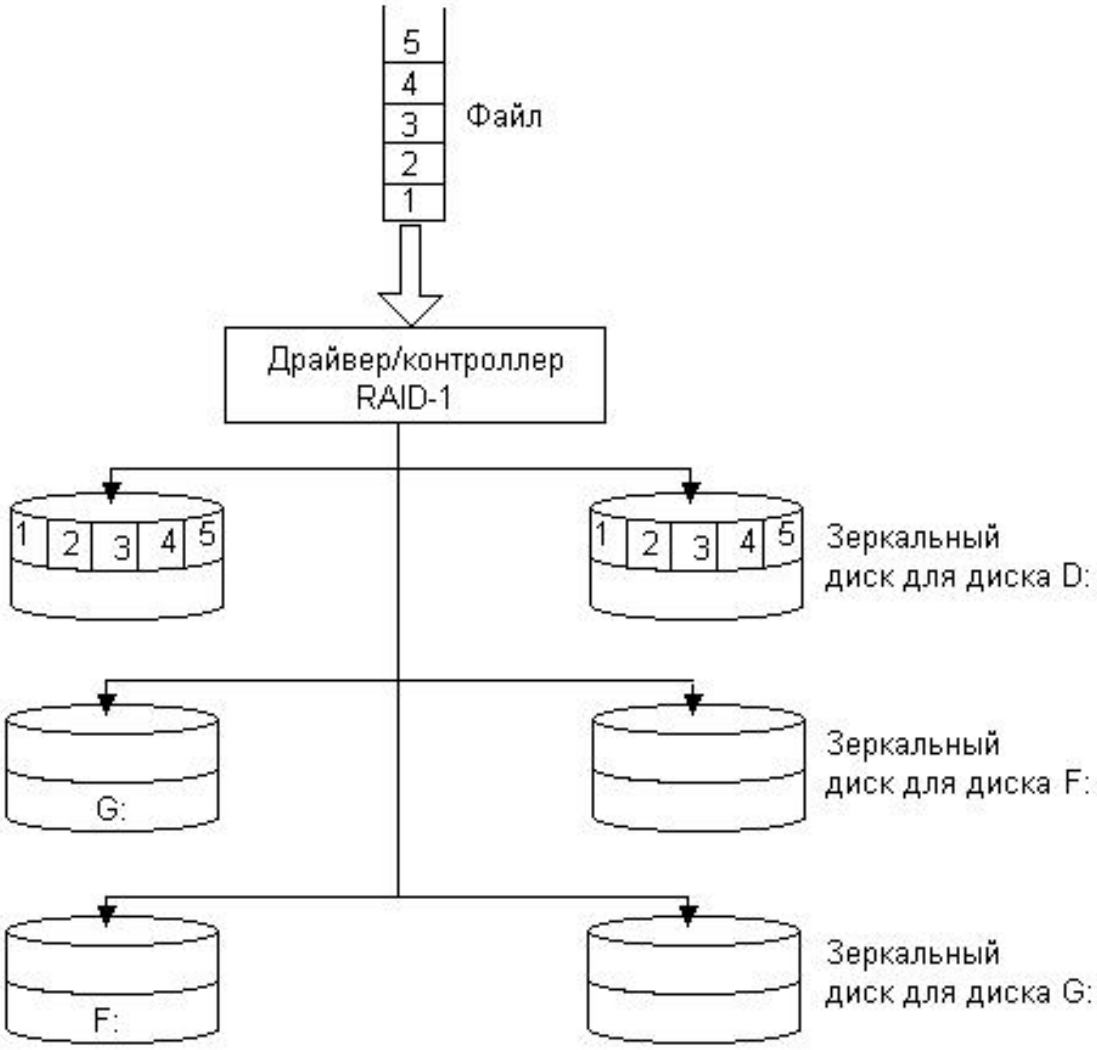
- В логическом устройстве RAID-0 общий для дискового массива контроллер при выполнении операции записи расщепляет данные на блоки и передает их параллельно на все диски, при этом первый блок данных записывается на первый диск, второй — на второй и т. д.
- Например в наборах с чередованием, представляющих собой программную реализацию RAID-0 в Windows NT, на диски поочередно записываются полосы данных (strips) по 64 Кбайт. При чтении контроллер мультиплексирует блоки данных, поступающие со всех дисков, и передает их источнику запроса.
- (+) Производительность дисковой конфигурации RAID-0 значительно выше за счет одновременности операций записи/чтения по всем дискам массива.
- (-) Уровень RAID-0 не обладает избыточностью данных, а значит, не имеет возможности повысить отказоустойчивость.



Два диска — минимальное количество для построения «зеркального» массива

RAID 1

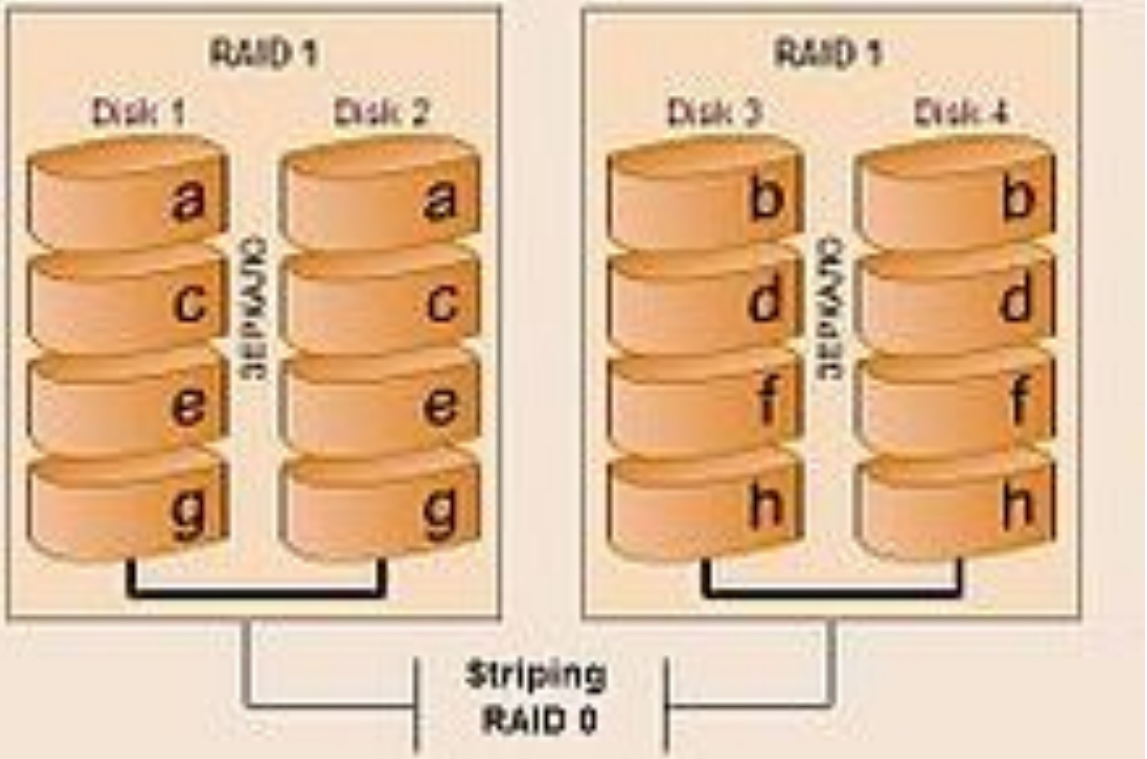
- Реализует подход, называемый зеркальным копированием (mirroring). Логическое устройство в этом случае образуется на основе одной или нескольких пар дисков, в которых один диск является основным, а другой диск (зеркальный) дублирует информацию, находящуюся на основном диске
- (+): Обеспечивает приемлемую скорость записи и выигрыш по скорости чтения при распараллеливании запросов.
- (+): Имеет высокую надёжность — работает до тех пор, пока функционирует хотя бы один диск в массиве. Вероятность выхода из строя сразу двух дисков равна произведению вероятностей отказа каждого диска. На практике при выходе из строя одного из дисков следует срочно принимать меры — вновь восстанавливать избыточность. Для этого с любым уровнем RAID (кроме нулевого) рекомендуют использовать диски горячего резерва. Достоинство такого подхода — поддержание постоянной доступности.
- (-): Недостаток заключается в том, что приходится выплачивать стоимость двух жёстких дисков, получая полезный объём одного жёсткого диска (классический случай, когда массив состоит из двух дисков).


5
4
3
2
1
Файл
Драйвер/контроллер
RAID-1
1
2
3
4
5
1
2
3
4
5
Зеркальный
диск для диска D:
G:
Зеркальный
диск для диска F:
F:
Зеркальный
диск для диска G:

- Зеркало на многих дисках — RAID 1+0 или RAID 0+1. Под RAID 1+0 имеют в виду вариант RAID 10, когда два RAID 1 объединяются в RAID 0. Вариант, когда два RAID 0 объединяются в RAID 1 называется RAID 0+1, и "снаружи" представляет собой тот же RAID 10. Достоинства и недостатки такие же, как и у уровня RAID 0. Как и в других случаях, рекомендуется включать в массив диски горячего резерва из расчёта один резервный на пять рабочих

RAID 10

Отказоустойчивый массив с дублированием и параллельной обработкой



RAID 2

- В массивах такого типа диски делятся на две группы — для данных и для кодов коррекции ошибок, причём если данные хранятся на 2^n-n-1 дисках , то для хранения кодов коррекции необходимо n дисков. Данные записываются на диски так же, как и в RAID 0, они разбиваются на небольшие блоки по числу дисков, предназначенных для хранения информации. Оставшиеся диски хранят коды коррекции ошибок, по которым в случае выхода какого-либо жёсткого диска из строя возможно восстановление информации. Метод Хемминга давно применяется в памяти типа ECC и позволяет на лету исправлять однократные и обнаруживать двукратные ошибки.
- Недостаток массива RAID 2 в том, что для его функционирования нужна структура из почти двойного количества дисков, поэтому такой вид массива не получил распространения.
- Минимальное количество дисков, при котором имеет смысл его использовать — 7.

Расчетное количество дисков для организации RAID 2

Количество дисков с данными	Количество дисков коррекции	Перерасход дисков %	Всего дисков
0	1	100,0000	1
1	2	66,6667	3
4	3	42,8571	7
11	4	26,6667	15
26	5	16,1290	31
57	6	9,5238	63
120	7	5,5118	127
247	8	3,1373	255
502	9	1,7613	511
1013	10	0,9775	1023
2036	11	0,5374	2047
4083	12	0,2930	4095
8178	13	0,1587	8191
16369	14	0,0855	16383
32752	15	0,0458	32767
65519	16	0,0244	65535
131054	17	0,0130	131071
262125	18	0,0069	262143
524268	19	0,0036	524287
1048555	20	0,0019	1048575

RAID 3

В массиве RAID 3 из n дисков данные разбиваются на блоки размером 1 байт и распределяются по $n - 1$ дискам. Ещё один диск используется для хранения блоков чётности.

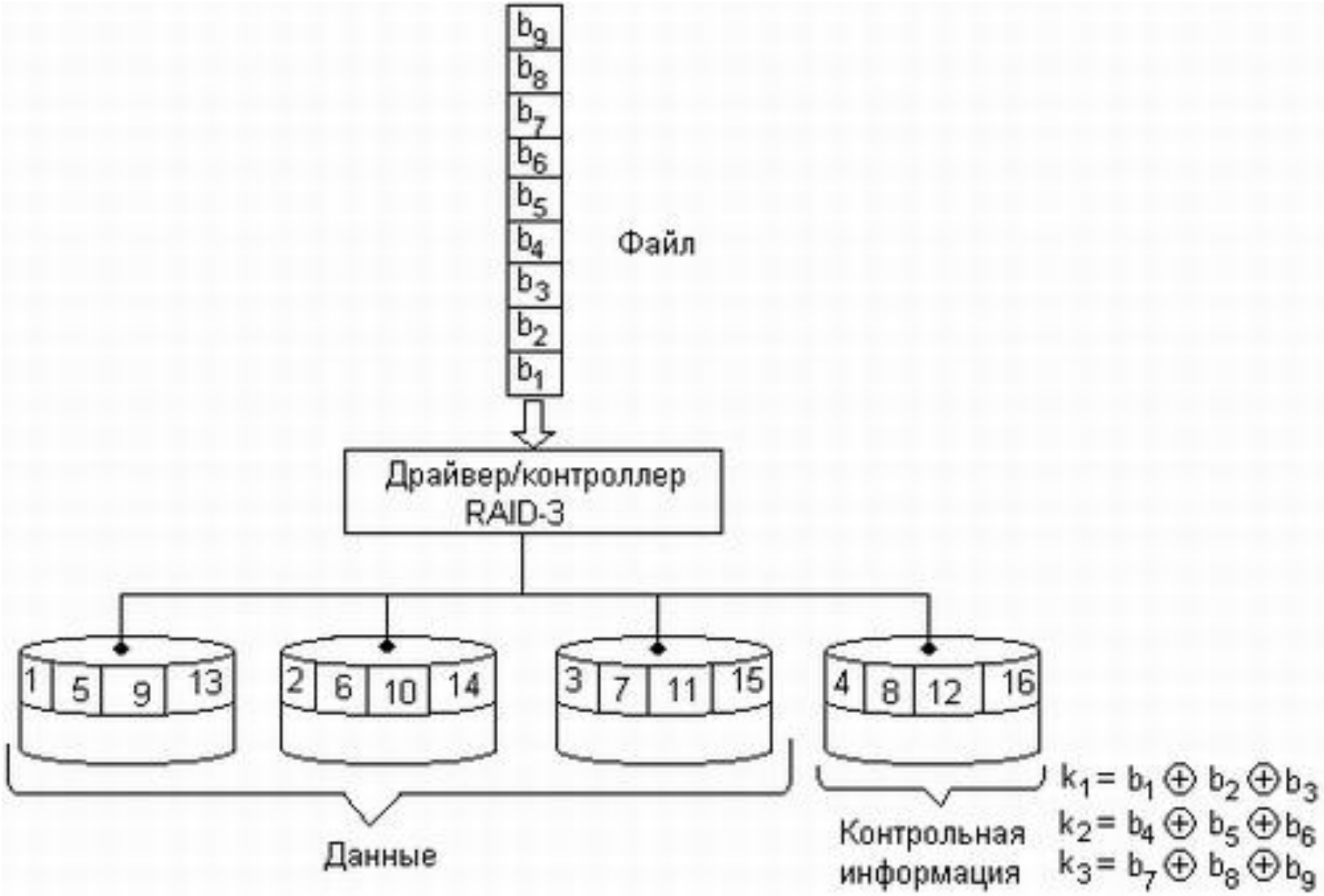
Отличия RAID 3 от RAID 2: невозможность коррекции ошибок на лету и меньшая избыточность.

Достоинства:

1. высокая скорость чтения и записи данных;
2. минимальное количество дисков для создания массива равно трём.

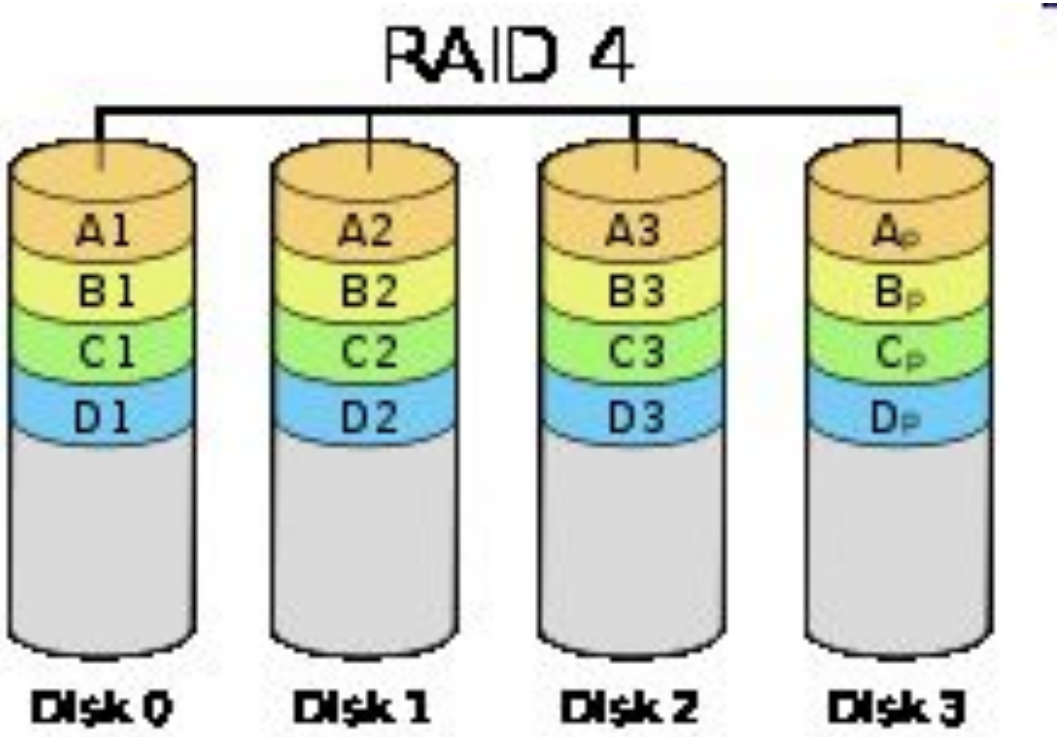
Недостатки:

1. массив этого типа хорош только для однозадачной работы с большими файлами, так как время доступа к отдельному сектору, разбитому по дискам, равно максимальному из интервалов доступа к секторам каждого из дисков. Для блоков малого размера время доступа намного больше времени чтения.
2. большая нагрузка на контрольный диск, и, как следствие, его надёжность сильно падает по сравнению с дисками, хранящими данные.


b9
b8
b7
b6
b5
b4
b3
b2
b1
Файл
Драйвер/контроллер
RAID-3
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
Данные
Контрольная
информация
k1 = b1 ⊕ b2 ⊕ b3
k2 = b4 ⊕ b5 ⊕ b6
k3 = b7 ⊕ b8 ⊕ b9

RAID 4

- RAID 4 похож на RAID 3, но отличается от него тем, что данные разбиваются на блоки, а не на байты. Таким образом, удалось отчасти «победить» проблему низкой скорости передачи данных небольшого объёма. Запись же производится медленно из-за того, что чётность для блока генерируется при записи и записывается на единственный диск. Из систем хранения широкого распространения RAID-4 применяется на устройствах хранения компании NetApp (NetApp FAS), где его недостатки успешно устранены за счет работы дисков в специальном режиме групповой записи, определяемом используемой на устройствах внутренней файловой системой WAF


RAID 4
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
Ap
Bp
Cp
Dp
Disk 0
Disk 1
Disk 2
Disk 3

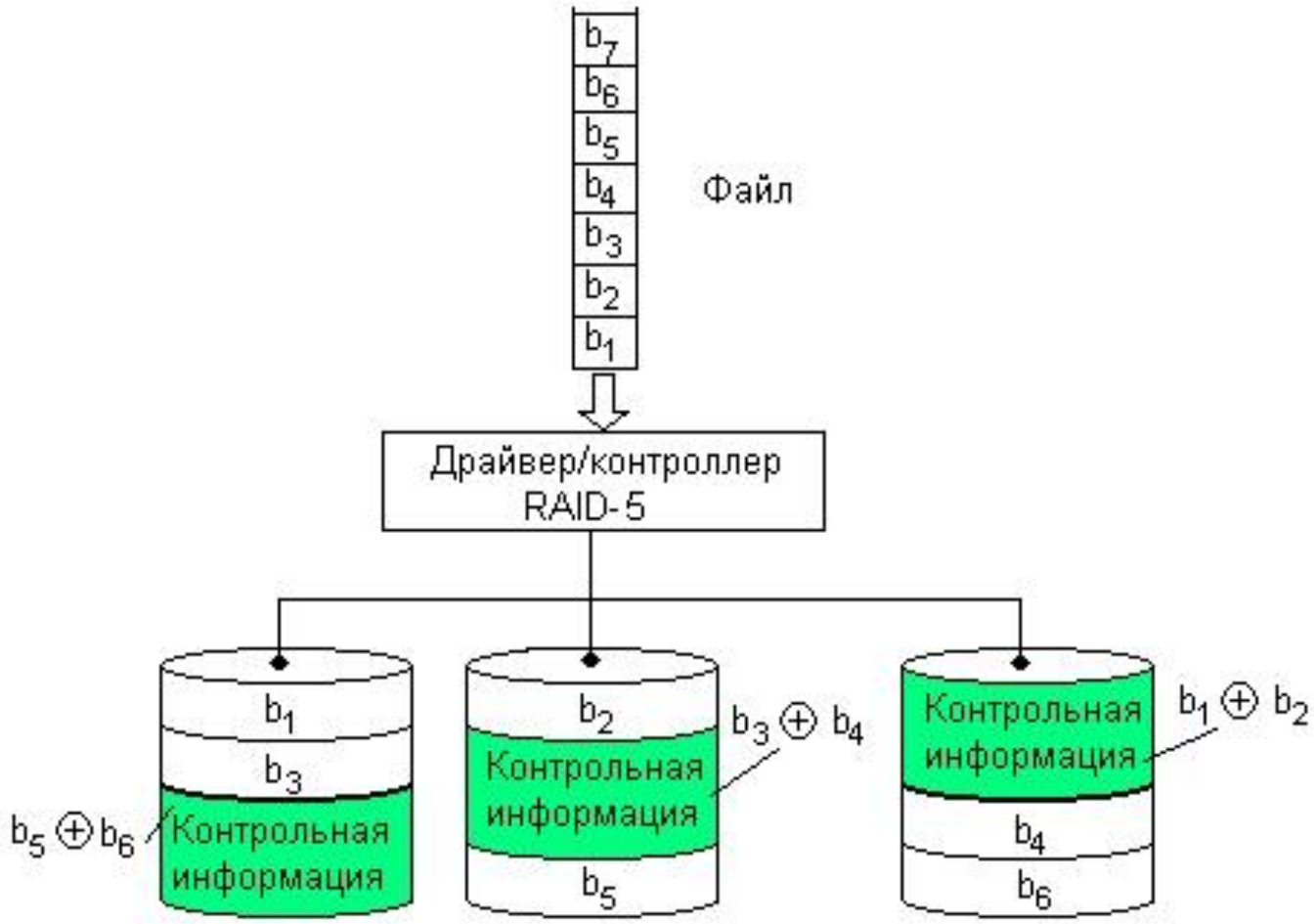
RAID 5

- Блоки данных и контрольные суммы циклически записываются на все диски массива, нет асимметричности конфигурации дисков. Под контрольными суммами подразумевается результат операции XOR(исключающее или). Это даёт возможность заменить любой операнд результатом, и применив алгоритм xor, получить недостающий операнд. Для хранения результата xor требуется всего 1 диск, размер которого равен размеру любого другого диска в raid.
- (+): RAID5 получил широкое распространение, в первую очередь, благодаря своей экономичности. Объём дискового массива RAID5 рассчитывается по формуле

(n-1)*hddsize,

где n — число дисков в массиве, а hddsize — размер наименьшего диска. Например, для массива из 4-х дисков по 80 гигабайт общий объём будет (4 — 1) * 80 = 240 гигабайт.

На запись информации на том RAID 5 тратятся дополнительные ресурсы и падает производительность, так как требуются дополнительные вычисления и операции записи, зато при чтении (по сравнению с отдельным винчестером) имеется выигрыш, потому что потоки данных с нескольких дисков массива могут обрабатываться параллельно.


b7
b6
b5
b4
b3
b2
b1
Файл
Драйвер/контроллер RAID-5
b1
b3
Контрольная информация
b5 ⊕ b6
b2
Контрольная информация
b3 ⊕ b4
b5
Контрольная информация
b1 ⊕ b2
b4
b6

Характеристики уровней RAID

Конфигурация	**Избыточность**	**Отказо-устойчивость**	**Скорость чтения**	**Скорость записи**
RAID-0	Нет	Нет	Повышенная	Повышенная
RAID RAID-1	50%	Есть	Повышенная	Пониженная (в варианте без дуплексирования)
RAID-3, RAID-4, RAID-5	До 33%	Есть	Повышенная	Пониженная (в разной степени)
RAID-10	50%	Есть	Повышенная	Повышенная

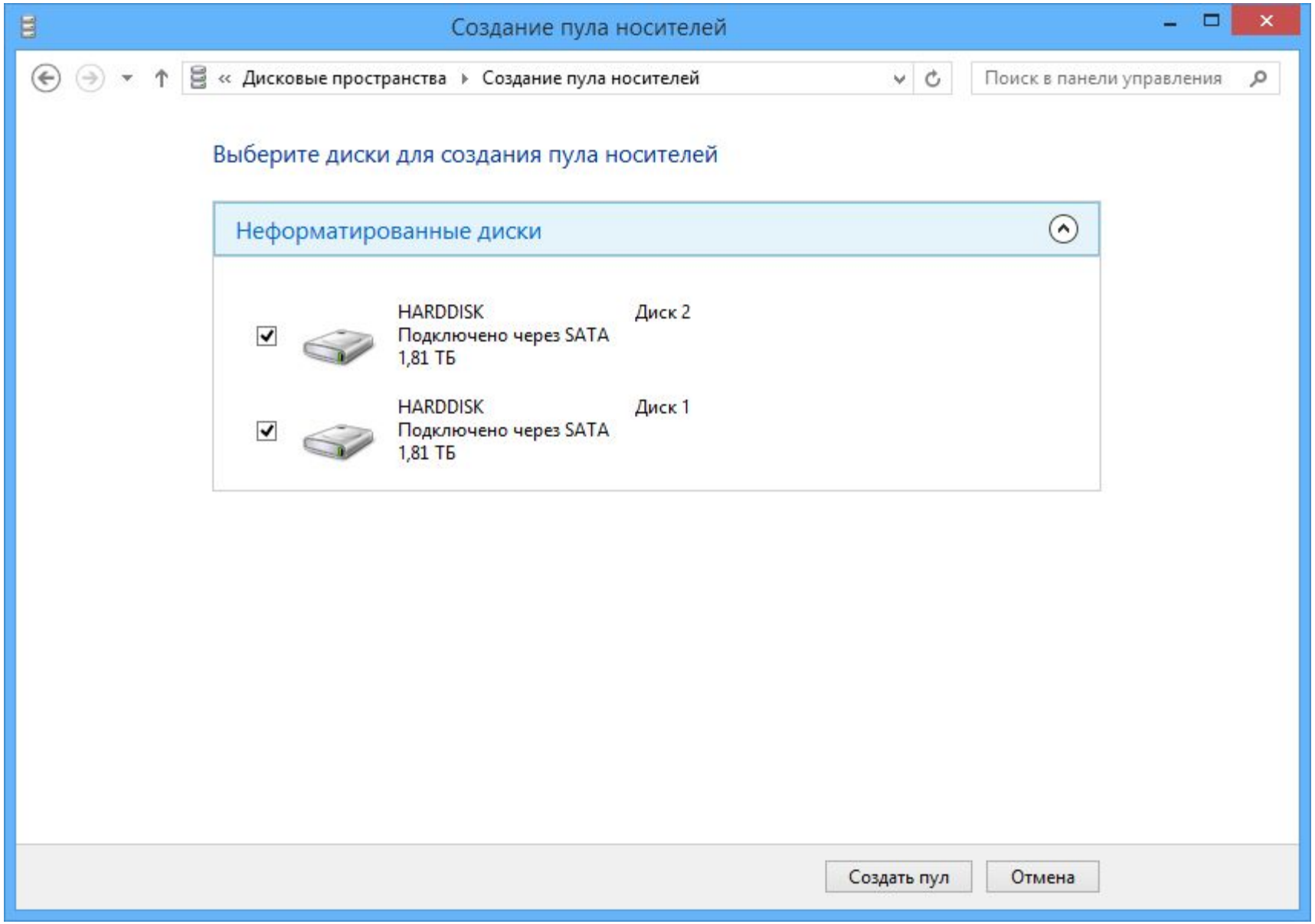
- Matrix RAID — это технология, реализованная фирмой Intel в своих чипсетах, начиная с ICH6R.
- Эта технология позволяет, используя небольшое количество дисков, организовать на разных разделах этих дисков одновременно несколько массивов уровня RAID 1, RAID 0 и RAID 5. Что позволяет за сравнительно небольшие деньги обеспечить для одних данных повышенную надёжность, а для других — высокую производительность.

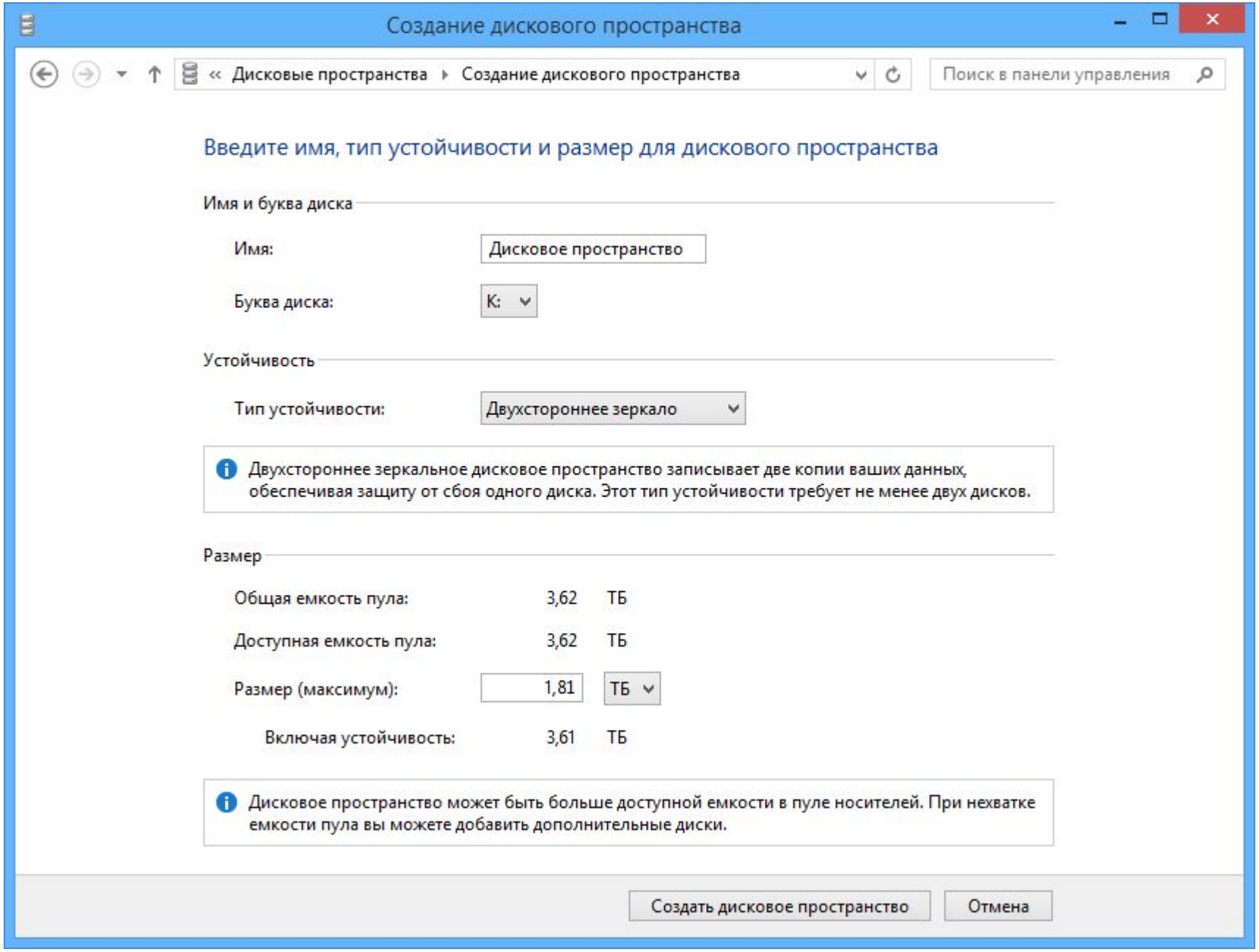
Программный (software) RAID

- Для реализации RAID можно применять не только аппаратные средства, но и полностью программные компоненты (драйверы).
- Например, в системах на ядре Linux существуют специальные модули ядра, а управлять RAID-устройствами можно с помощью утилиты mdadm. Программный RAID имеет свои достоинства и недостатки. С одной стороны, он ничего не стоит (в отличие от аппаратных RAID-контроллеров, цена которых от $250). С другой стороны, программный RAID использует ресурсы центрального процессора, и в моменты пиковой нагрузки на дисковую систему процессор может значительную часть мощности тратить на обслуживание RAID-устройств.

- Ядро Linux поддерживает программные RAID следующих уровней: 0, 1, 4, 5, 6, 10. Реализация позволяет создавать RAID на отдельных разделах дисков, что аналогично описанному выше Matrix RAID. Поддерживается загрузка с RAID.
- ОС семейства Windows NT, такие как Windows NT4/2000/XP/2003 изначально, с момента проектирования данного семейства, поддерживают программный RAID 0, RAID 1 и RAID 5.
- Windows 7 поддерживает программный RAID 0 и RAID 1, Windows Server 2003 — 0, 1 и 5.

- В Windows 8 появилась функция Storage Spaces (в русской версии Дисковые пространства), которая, *в какой-то степени*, является программным аналогом RAID-массива.
- Storage Spaces предлагает объединить физические накопители с интерфейсами USB, SATA, SAS, SCSI (причем в любых комбинациях) в пулы, а сами пулы в дисковые пространства, которые выглядят для пользователя обычными логическими дисками.
- Также с помощью Storage Spaces можно обеспечить сохранность данных с помощью

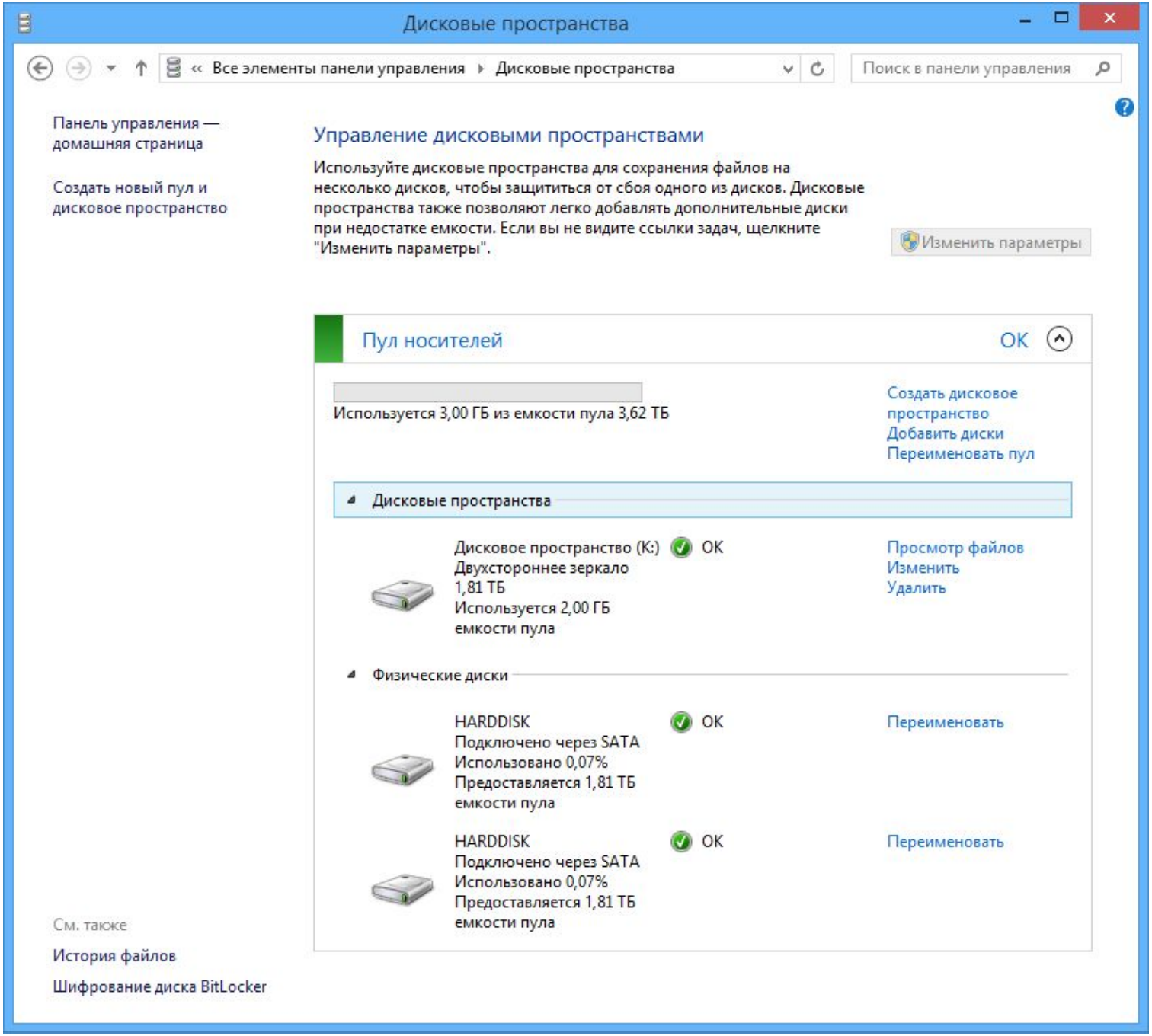

Создание пула носителей
« Дисковые пространства ▸ Создание пула носителей
Поиск в панели управления
Выберите диски для создания пула носителей
Неформатированные диски
HARDDISK
Подключено через SATA
1,81 ТБ
Диск 2
HARDDISK
Подключено через SATA
1,81 ТБ
Диск 1
Создать пул
Отмена


Создание дискового пространства
« Дисковые пространства ▸ Создание дискового пространства
Поиск в панели управления
Введите имя, тип устойчивости и размер для дискового пространства
Имя и буква диска
Имя:
Дисковое пространство
Буква диска:
K:
Устойчивость
Тип устойчивости:
Двухстороннее зеркало
Двухстороннее зеркальное дисковое пространство записывает две копии ваших данных, обеспечивая защиту от сбоя одного диска. Этот тип устойчивости требует не менее двух дисков.
Размер
Общая емкость пула: 3,62 ТБ
Доступная емкость пула: 3,62 ТБ
Размер (максимум): 1,81 ТБ
Включая устойчивость: 3,61 ТБ
Дисковое пространство может быть больше доступной емкости в пуле носителей. При нехватке емкости пула вы можете добавить дополнительные диски.
Создать дисковое пространство
Отмена

Windows предлагает выбрать четыре типа устойчивости:

- Простой. Хранимая информация распределяется по всем дискам в пуле последовательно. Общая емкость пространства в этом случае равна общему объему всех накопителей, включенных в него. Данный тип является аналогом RAID 0.
- Двухстороннее зеркало. Запись осуществляется одновременно на каждый накопитель. Для данного типа устойчивости требуется минимум два диска. Общая емкость пространства равна половине от общего объема накопителей. Данный тип устойчивости является аналогом RAID 1.
- Трехстороннее зеркало. Запись осуществляется одновременно на три накопителя. Данные сохраняются при отказе сразу двух дисков. Требуется минимум 5 дисков. Данная схема имеет максимальную надежность, но снижает скорость чтения и записи. Общая емкость пространства равна ¼ от общей емкости всех дисков.

- Четность. Может задействоваться от 3 до 8 накопителей. На все накопители, кроме одного записываются данные, а на последний записывается информация о четности (блок контрольных сумм, вычисляемы по алгоритму XOR). В случае отказа одного из накопителей, недостающие данные могут быть вычислены. По надежности аналогично двухстороннему зеркалу, но количество избыточных данных равно 1/N (количество накопителей), а у двухстороннего зеркала всегда ½. То есть количество пространства, доступного для записи полезных данных в случае использования четности, больше. Но так как требуется рассчитывать контрольные суммы, скорость записи существенно меньше.


Дисковые пространства
« Все элементы панели управления ▸ Дисковые пространства
Поиск в панели управления
Панель управления — домашняя страница
Создать новый пул и дисковое пространство
Управление дисковыми пространствами
Используйте дисковые пространства для сохранения файлов на несколько дисков, чтобы защититься от сбоя одного из дисков. Дисковые пространства также позволяют легко добавлять дополнительные диски при недостатке емкости. Если вы не видите ссылки задач, щелкните "Изменить параметры".
Изменить параметры
Пул носителей
OK
Используется 3,00 ГБ из емкости пула 3,62 ТБ
Создать дисковое пространство
Добавить диски
Переименовать пул
Дисковые пространства
Дисковое пространство (K:)
OK
Двухстороннее зеркало
1,81 ТБ
Используется 2,00 ГБ емкости пула
Просмотр файлов
Изменить
Удалить
Физические диски
HARDDISK
OK
Подключено через SATA
Использовано 0,07%
Предоставляется 1,81 ТБ емкости пула
Переименовать
HARDDISK
OK
Подключено через SATA
Использовано 0,07%
Предоставляется 1,81 ТБ емкости пула
Переименовать
См. также
История файлов
Шифрование диска BitLocker

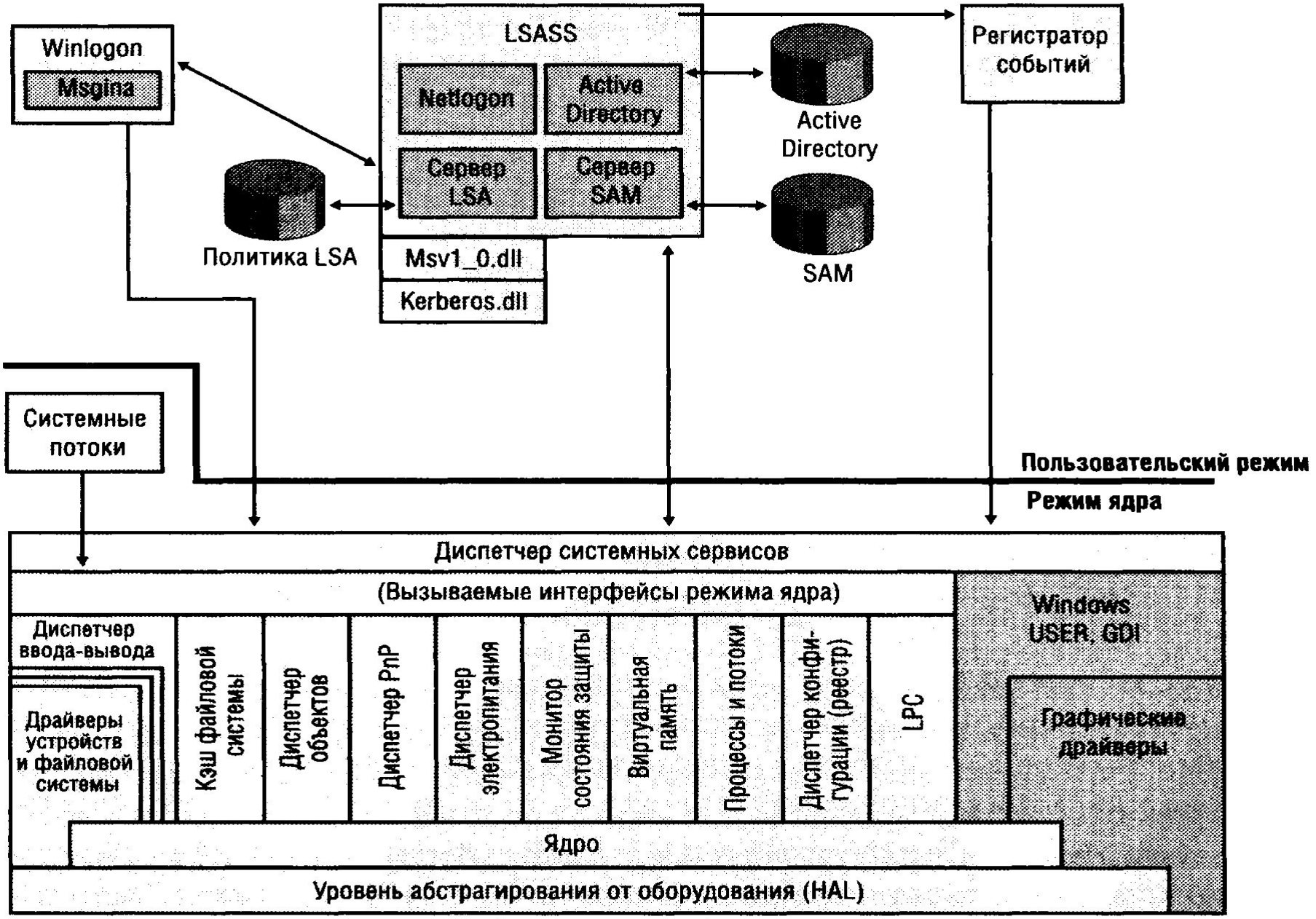
Компоненты системы защиты Windows

- **Монитор состояния защиты (Security Reference Monitor, SRM) -** Компонент исполнительной системы, отвечающий за определение структуры данных маркера доступа для представления контекста защиты, за проверку прав доступа к объектам, манипулирование правами пользователей и генерацию сообщений аудита безопасности.
- **Подсистема локальной аутентификации (local security authentication subsystem, LSASS)** Процесс пользовательского режима, который отвечает за политику безопасности в локальной системе (например, круг пользователей, имеющих право на вход в систему, правила, связанные с паролями, привилегии, выдаваемые пользователям и их группам, параметры аудита безопасности системы), а также за аутентификацию пользователей и передачу сообщений аудита безопасности в Event Log. Основную часть этой функциональности реализует сервис локальной аутентификации Lsasrv (\\Windows\Windows\\Windows\System\Windows\System32\\Windows\System32\Lsasrv\Windows\System32\Lsasrv.\Windows\Sy

- **База данных политики LSASS.** База данных, содержащая параметры политики безопасности локальной системы. Она хранится в разделе реестра HKLM\SECURITY и включает следующую информацию: каким доменам доверена аутентификация попыток входа в систему, кто имеет права на доступ к системе и каким образом, кому предоставлены те или иные привилегии и какие виды аудита следует выполнять.
- **Диспетчер учетных записей безопасности (Security Accounts Manager, SAM).** Набор подпрограмм, отвечающих за поддержку базы данных, которая содержит имена пользователей и группы, определенные на локальной машине. Служба SAM, реализованная как \\Windows\Windows\\Windows\System\Windows\System32\ Samsrv.dll, выполняется в процессе Lsass.

- **База данных SAM.** База данных, которая в системах, отличных от контроллеров домена, содержит информацию о локальных пользователях и группах вместе с их паролями и другими атрибутами. На контроллерах домена SAM содержит определение и пароль учетной записи администратора, имеющего права на восстановление системы. Эта база данных хранится в разделе реестра HKLM\SAM.
- **Active Directory.** Служба каталогов, содержащая базу данных со сведениями об объектах в домене. ***Домен*** — это совокупность компьютеров и сопоставленных с ними групп безопасности, которые управляются как единое целое. Active Directory хранит информацию об объектах домена, в том числе о пользователях, группах и компьютерах.

- **Пакеты аутентификации.** DLL-модули, выполняемые в контексте процесса Lsass и клиентских процессов и реализующие политику аутентификации в Windows. DLL аутентификации отвечает за проверку пароля и имени пользователя, а также за возврат LSASS (в случае успешной проверки) детальной информации о правах пользователя, на основе которой LSASS генерирует маркер (token).
- **Процесс входа (Winlogon)** Процесс пользовательского режима (\Windows\System32\Winlogon.exe), отвечающий за поддержку SAS и управление сеансами интерактивного входа в систему. Например, при регистрации пользователя Winlogon создает оболочку — пользовательский интерфейс.
- **GINA (Graphical Identification and Authentication)** DLL пользовательского режима, выполняемая в процессе Winlogon и применяемая для получения пароля и имени пользователя или PIN-кода смарт-карты.



Компоненты защиты Windows

Организация контроля доступа в Windows

- Для разделяемых ресурсов в Windows применяется общая модель объекта, который содержит такие характеристики безопасности, как набор допустимых операций, идентификатор владельца, список управления доступом. Объекты создаются для **любых ресурсов** в том случае, когда они **являются или становятся разделяемыми** — файлов, каталогов, устройств, секций памяти, процессов. Характеристики объектов делятся на две части — общую часть, состав которой не зависит от типа объекта, и индивидуальную, определяемую типом объекта.
- Все объекты хранятся в древовидных иерархических структурах, элементами которых являются объекты-ветви (каталоги) и объекты-листья (файлы). Для объектов файловой системы такая схема отношений является прямым отражением иерархии каталогов и файлов. Для объектов других типов иерархическая схема отношений имеет свое содержание, например, для процессов она отражает связи «родитель-потомок», а для устройств отражает принадлежность к определенному типу устройств и связи устройства с другими устройствами, например SCSI

- Проверка прав доступа для объектов любого типа выполняется централизованно с помощью **монитора безопасности** (Security Reference Monitor), работающего в привилегированном режиме.
- Для системы безопасности Windows характерно большое количество различных предопределенных (встроенных) субъектов доступа — как отдельных пользователей, так и групп. В системе всегда имеются пользователи Administrator, System и Guest, а также группы Users, Administrators, Server Operators, Everyone и др.
- Смысл этих встроенных пользователей и групп состоит в том, что они наделены некоторыми правами. При добавлении нового пользователя администратору остается только решить, к какой группе или группам отнести этого пользователя. Администратор может создавать новые группы и добавлять права встроенным группам для реализации собственной политики безопасности.

Windows поддерживает **три класса** операций доступа, которые отличаются типом субъектов и объектов, участвующих в этих операциях.

- **Разрешения** (permissions) — это множество операций, которые могут быть определены для субъектов всех типов по отношению к объектам любого типа: файлам, каталогам, принтерам, секциям памяти и т. д.
- **Права** (user rights) — определяются для субъектов типа группа на выполнение некоторых системных операций: установку системного времени, архивирование файлов, выключение компьютера и т. п. В этих операциях участвует особый объект доступа — операционная система в целом. Некоторые права у встроенной группы являются также встроенными — их у данной группы нельзя удалить.
- **Возможности пользователей** (user abilities) определяются для отдельных пользователей на выполнение действий, связанных с формированием их операционной среды, например изменение состава главного меню программ.

При входе пользователя в систему для него создается так называемый токен доступа (access token), включающий:

- идентификатор пользователя,
- идентификаторы всех групп, в которые входит пользователь,
- список управления доступом (ACL) по умолчанию, который состоит из разрешений и применяется к создаваемым процессом объектам,
- список прав пользователя на выполнение системных действий.

Все объекты, включая файлы, потоки, события, даже токены доступа, когда они создаются, снабжаются дескриптором безопасности. Дескриптор безопасности содержит список управления доступом — ACL (Access Control List). Владелец объекта, обычно пользователь, который его создал, обладает правом избирательного управления доступом к объекту и может изменять ACL объекта, чтобы позволить или не позволить другим осуществлять доступ к объекту.

ACL состоит из элементов управления доступом (Access Control Element, ACE), при этом каждый элемент соответствует одному идентификатору. Список ACL с добавленным к нему идентификатором владельца называют характеристиками безопасности.

Разрешения на доступ к каталогам и файлам

- Доступ к каталогам и файлам контролируется за счет установки соответствующих разрешений.
- Разрешения в Windows бывают индивидуальные и стандартные. Индивидуальные разрешения относятся к элементарным операциям над каталогами и файлами, а стандартные разрешения являются объединением нескольких индивидуальных разрешений.

Разрешение	Для каталога	Для файла
Read (R)	Чтение имен файлов и каталогов, входящих в данный каталог, а также атрибутов и владельца каталога	Чтение данных, атрибутов, имени владельца и разрешений файла
Write (W)	Добавление файлов и каталогов, изменение атрибутов каталога, чтение владельца и разрешений каталога	Чтение владельца и разрешений файла, изменение атрибутов файла, изменение и добавление данных файла
Execute (X)	Чтение атрибутов каталога, выполнение изменений в каталогах, входящих в данный каталог, чтение имени владельца и разрешений каталога	Чтение атрибутов файла, имени владельца и разрешений. Выполнение файла, если он хранит код программы
Delete (D)	Удаление каталога	Удаление файла
Change Permission (P)	Изменение разрешений каталога	Изменение разрешений файла
Take Ownership (O)	Стать владельцем каталога	Стать владельцем файла

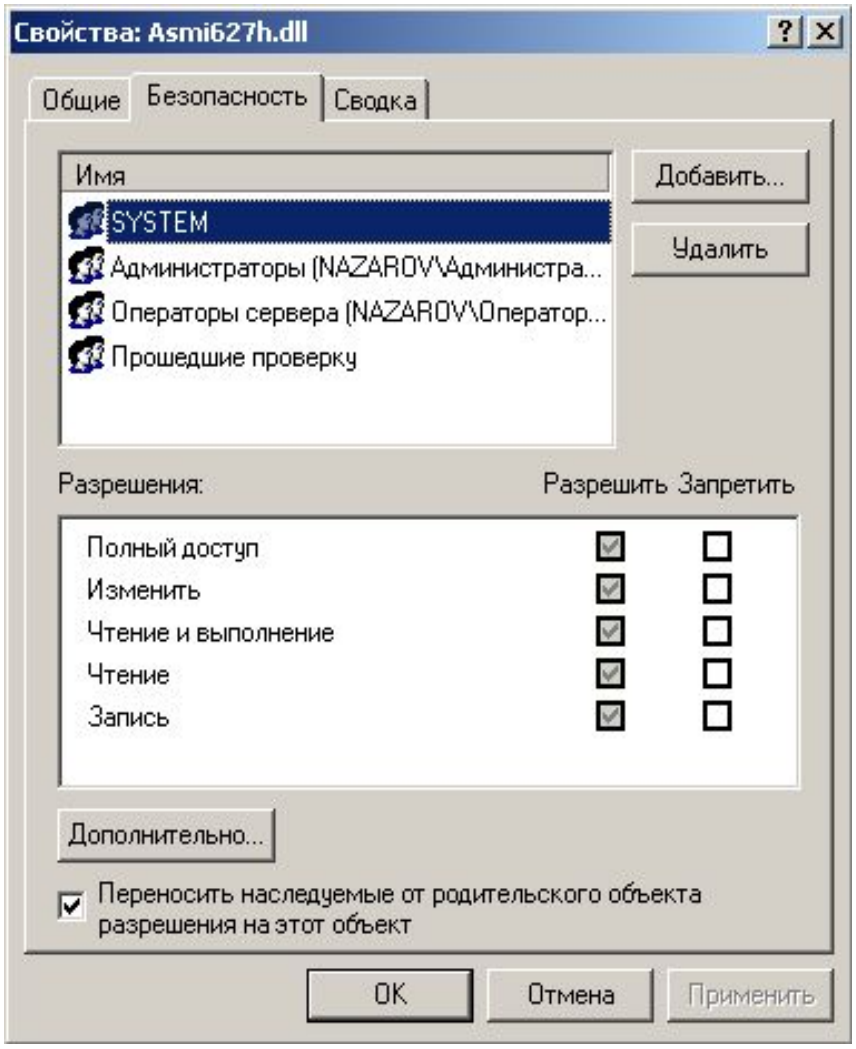
Для файлов в Windows определено четыре стандартных разрешения: No Access, Read, Change и Full Control, которые объединяют индивидуальные разрешения

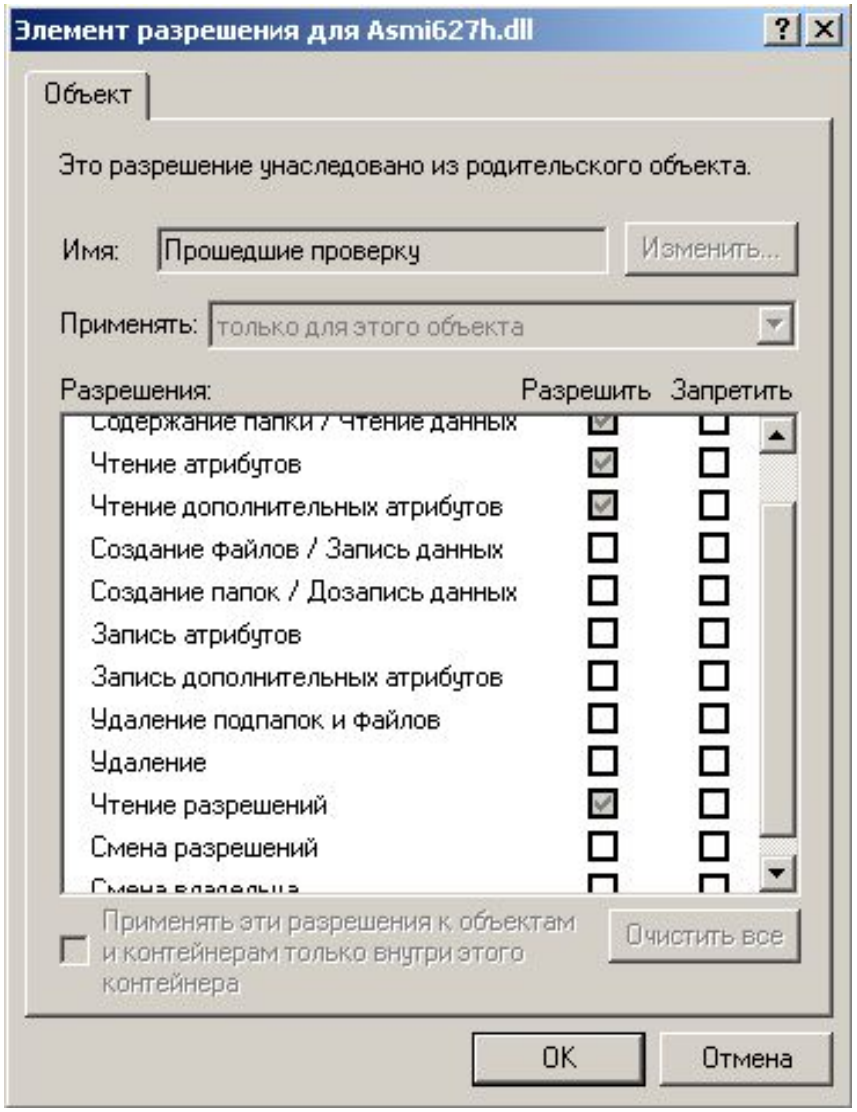
Стандартное разрешение	Индивидуальные разрешения
No Access	Ни одного
Read	RX
Change	RWXD
Full Control	Все

Стандартные разрешения	**Индивидуальные разрешения для каталога**	**Индивидуальные разрешения для файлов каталога при наследовании**
No Access	Ни одного	Ни одного
List	RX	Не определены
Read	RX	RX
Add	WX	Не определены
Add & Read	RWX	RX
Change	RWXD	RWXD
Full Control	Все	Все

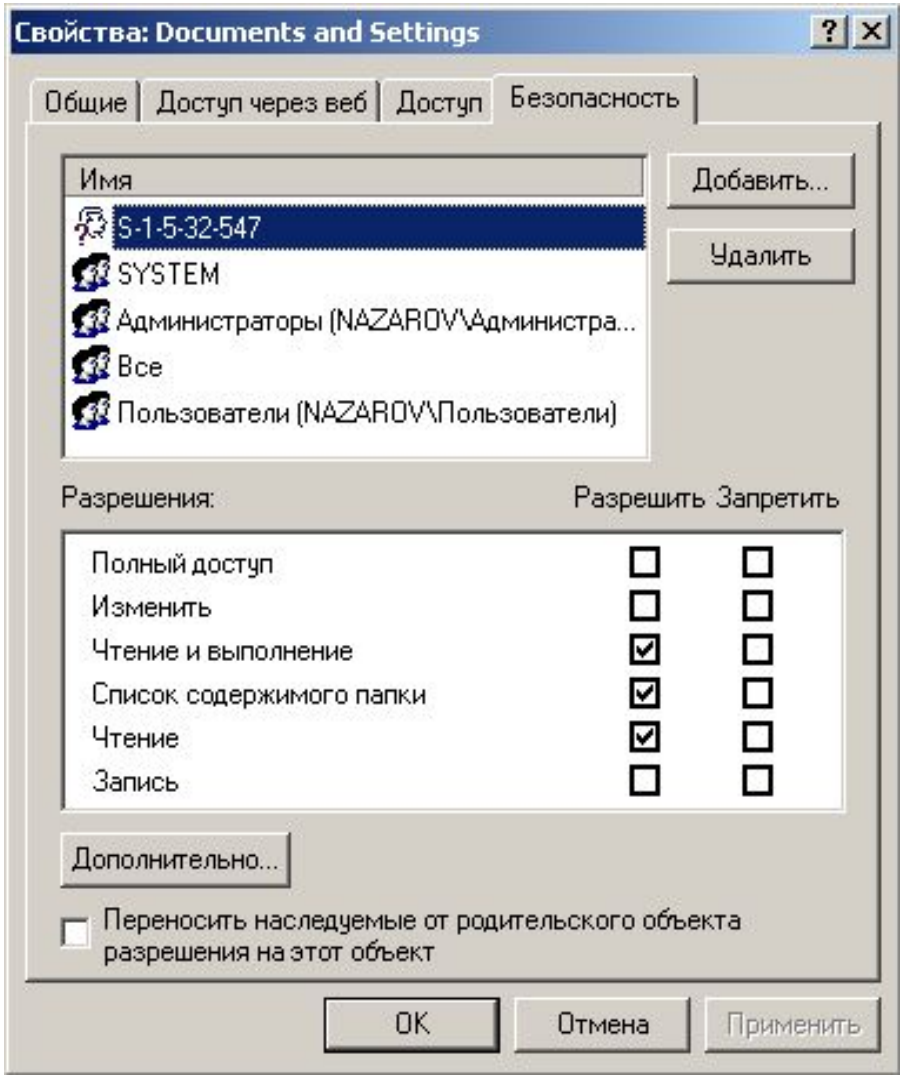
- Тип разрешения *Смена владельца* по умолчанию присвоен группе *Администраторы*.
- Создатель файла всегда считается его владельцем, который имеет права *Полный доступ*, даже в том случае, если учетная запись владельца не указана на вкладке *Безопасность файла*

Разрешения на доступ к файлам

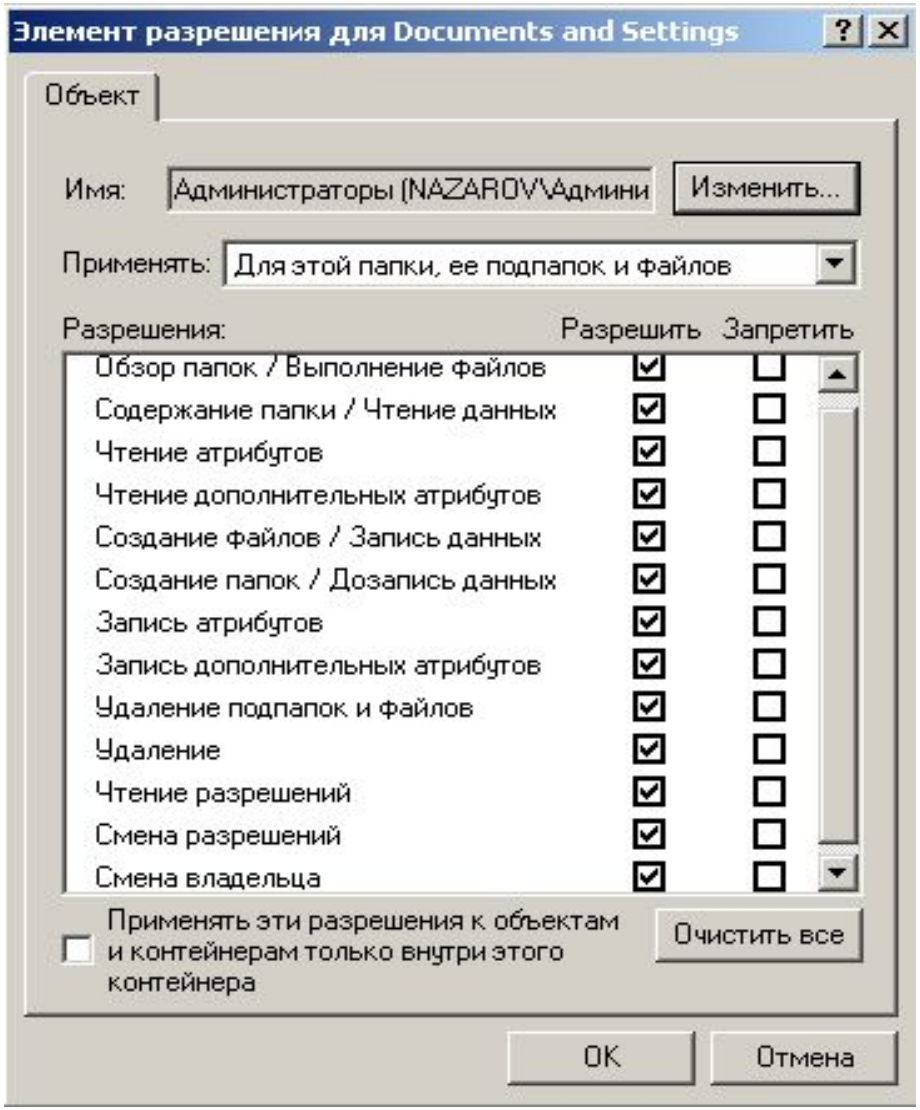




Разрешения на доступ к каталогам



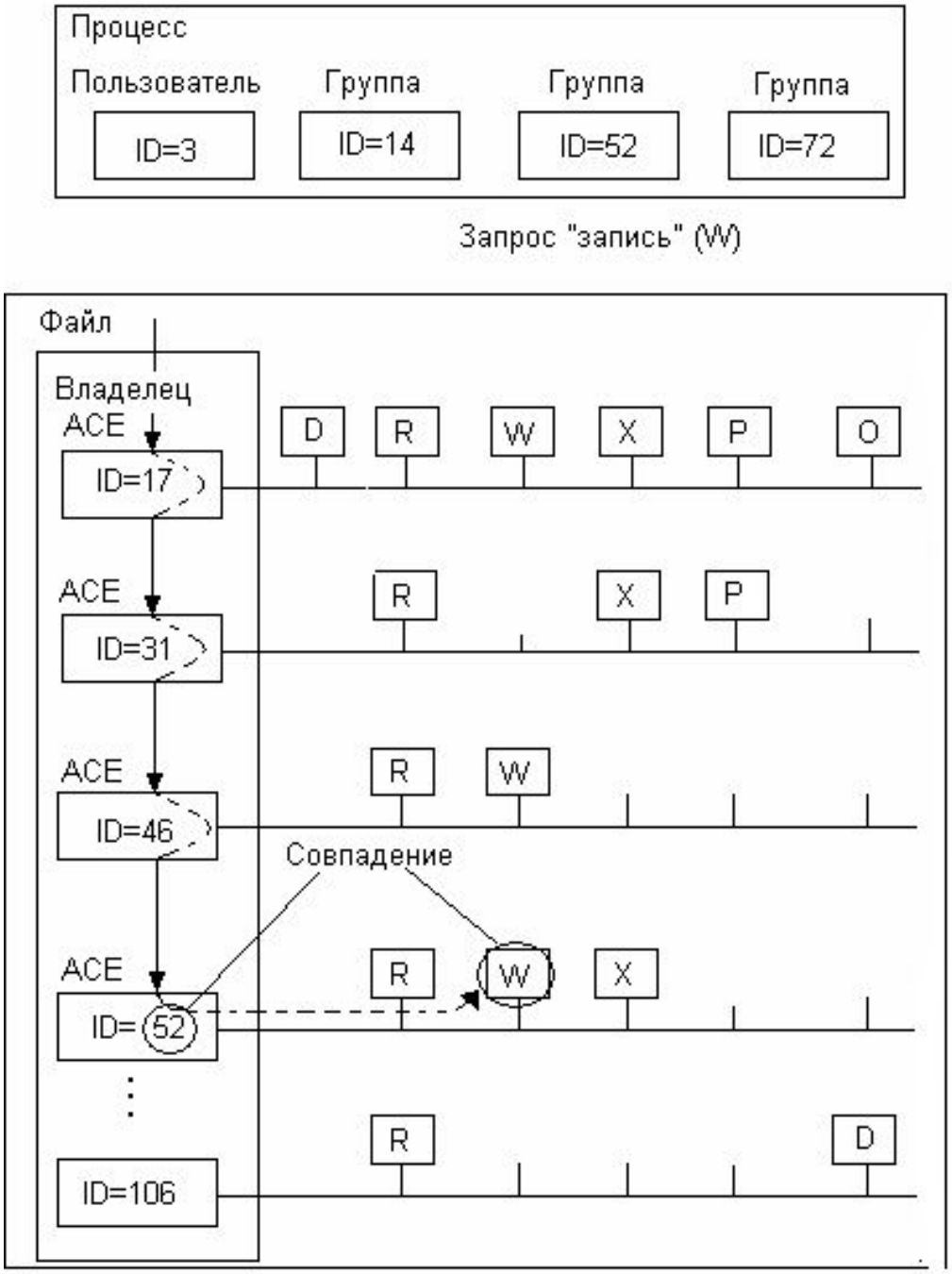
Стандартные разрешения



Специальные разрешения

Существует ряд правил, которые определяют действие разрешений.

- Пользователи не могут работать с каталогом или файлом, если они не имеют явного разрешения на это или же они не относятся к группе, которая имеет соответствующее разрешение.
- Разрешения имеют накопительный эффект, за исключением разрешения No Access, которое отменяет все остальные имеющиеся разрешения.


Процесс
Пользователь
Группа
Группа
Группа
ID=3
ID=14
ID=52
ID=72
Запрос "запись" (W)
Файл
Владелец
ACE
ID=17
D
R
W
X
P
O
ACE
ID=31
R
X
P
ACE
ID=46
R
W
Совпадение
ACE
ID=
52
R
W
X
R
D
ID=106

- В приведенном на рисунке примере процесс, который выступает от имени пользователя с идентификатором 3 и групп с идентификаторами 14, 52 и 72, пытается выполнить операцию записи (W) в файл. Файлом владеет пользователь с идентификатором 17. Операционная система, получив запрос на запись, находит характеристики безопасности файла (на диске или в буферной системной области) и последовательно сравнивает все идентификаторы процесса с идентификатором владельца файла и идентификаторами пользователей и групп в элементах ACE. В данном примере один из идентификаторов группы, от имени которой выступает процесс, а именно 52, совпадает с идентификатором одного из элементов ACE. Так как пользователю с идентификатором 52 разрешена операция чтения (признак W имеется в наборе операций этого элемента), то ОС разрешает процессу выполнение операции.

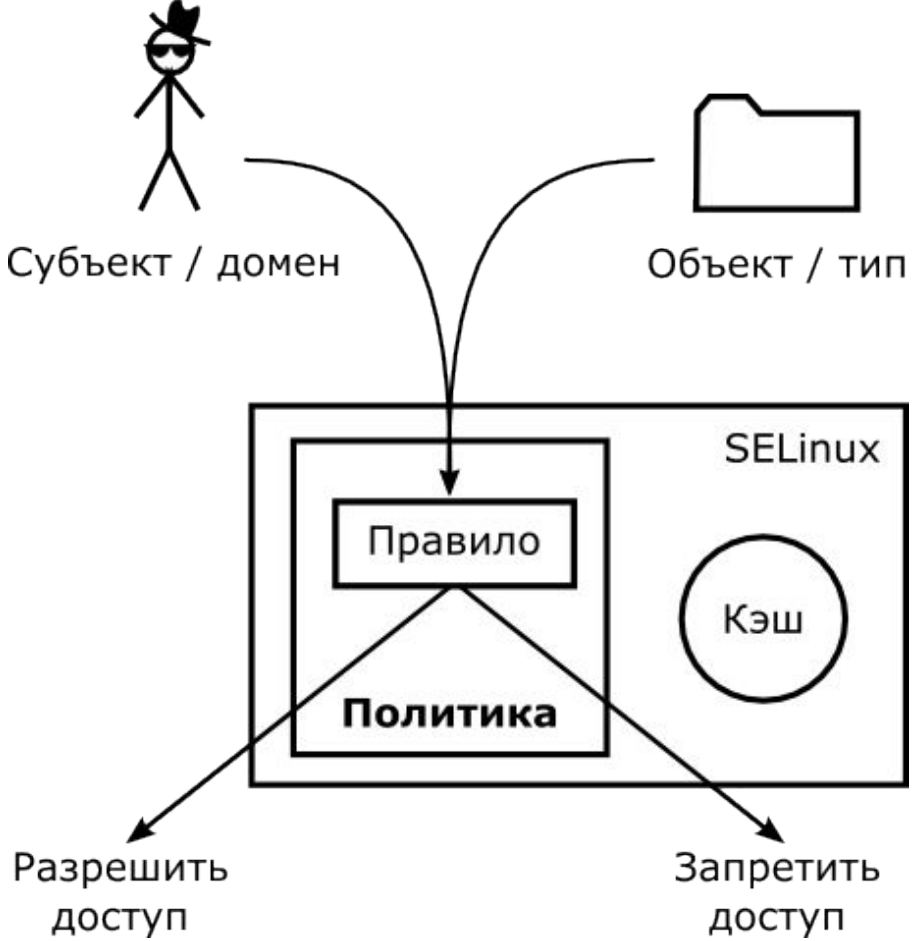
- Использование утилит командной строки takeown и icacls (применимо только к файлам, папкам и дискам)
- Чтобы принять на себя владения всеми файлами в каталоге D:\Game введите такую команду:
- takeown /f "D:\Game "
- Чтобы принять на себя владения файлом pesenka.mp3 на удаленном компьютере nya.xxxxx.com:
- takeown /f "D:\pesenka.mp3" /s nya.xxxxx.com
- Чтобы изменять разрешения для файла или папки:
- icacls папка_или_файл /grant:r пользователь: разрешение
- icacls "C:\Program Files (x86)\UltraISO\" /grant:r Putin:RX /T

Два основных подхода к определению прав доступа

- **Избирательный доступ** имеет место, когда для каждого объекта сам владелец может определить допустимые операции с объектами. Этот подход называется также произвольным (от discretionary — предоставленный на собственное усмотрение) доступом, так как позволяет администратору и владельцам объектов определить права доступа произвольным образом, по их желанию. Между пользователями и группами пользователей в системах с избирательным доступом нет жестких иерархических взаимоотношений, то есть взаимоотношений, которые определены по умолчанию и которые нельзя изменить. Исключение делается только для администратора, по умолчанию наделяемого всеми правами.

- **Мандатный доступ** (от mandatory — обязательный, принудительный) — это такой подход к определению прав доступа, при котором система наделяет пользователя определенными правами по отношению к каждому разделяемому ресурсу (в данном случае файлу) в зависимости от того, к какой группе пользователь отнесен. От имени системы выступает администратор, а владельцы объектов лишены возможности управлять доступом к ним по своему усмотрению. Все группы пользователей в такой системе образуют строгую иерархию, причем каждая группа пользуется всеми правами группы более низкого уровня иерархии, к которым добавляются права данного уровня.

Мандатная модель разграничения доступа предполагает назначение объекту метки (грифа) секретности, а субъекту – уровня допуска. Доступ субъектов к объектам в мандатной модели определяется на основании правил «*не читать выше*» и «*не записывать ниже*». Использование мандатной модели предотвращает утечку конфиденциальной информации, но снижает производительность



- Пример: субъект «Пользователь № 2», имеющий допуск уровня «не секретно», не может получить доступ к объекту, имеющего метку «для служебного пользования». В то же время, субъект «Пользователь № 1» с допуском уровня «секретно» право доступа к объекту с меткой «для служебного пользования» имеет.

Уровни секретности документов в СССР

- СС/ОП (Совершенно Секретно. Особая Папка)
- ОП (Особая Папка)
- ОВ (Особой Важности)
- СС (Совершенно Секретно)
- С (Секретно)
- ДСП (Для служебного пользования)

- Согласно требованиям ФСТЭК (Федеральная служба по техническому и экспортному контролю) мандатное управление доступом или "метки доступа" являются ключевым отличием систем защиты Государственной Тайны РФ старших классов 1В и 1Б от младших классов защитных систем на классическом разделении прав по матрице доступа.
- В сертифицированной в системах сертификации Минобороны России и ФСТЭК России **операционной системе специального назначения** "*Astra Linux Special Edition*", механизм мандатного разграничения доступа реализован, как и механизм дискреционного разграничения доступа в ядре ОС и СУБД.
- Решение о запрете или разрешении доступа субъекта к объекту принимается на основе типа операции (чтение\запись\исполнение), мандатного контекста безопасности, связанного с каждым субъектом, и мандатной метки, связанной с объектом. В сетевые пакеты протокола внедряются мандатные метки. В защищенных комплексах гипертекстовой обработки данных, электронной почты и в других сервисах, мандатное разграничение реализовано на основе программного интерфейса библиотек подсистемы безопасности PARSEC.

- Изначально такой принцип был воплощён в операционных системах Flask, и других ориентированных на безопасность операционных системах.
- Исследовательский проект АНБ SELinux добавил архитектуру мандатного контроля доступа к ядру Linux.
- Мандатная система разграничения доступа реализована в ОС FreeBSD Unix.
- В *SUSE Linux* и Ubuntu и Ubuntu есть архитектура мандатного контроля доступа под названием AppArmor.

Пример утилиты, позволяющей реализовать мандатный доступ в Windows

SecrecyKeeper позволяет решать следующие задачи:

- На любой файл можно установить метку конфиденциальности - гриф (общедоступная, служебная, секретная)
- Замкнутая программная среда - позволяет ограничить список программ, разрешенных для использования сотрудниками
- Защита конфигурации рабочей станции от изменения пользователем с правами администратора
- Динамическая и статическая блокировка доступа к USB, COM, LPT портам
- Журнал отслеживания загружаемых модулей и запускаемых программ
- Гриф может быть установлен на информацию доступ к которой предоставляется по сети (удаленно), такую, как, например, базы данных, корпоративные интернет-порталы и т.п.

Для сотрудников вводятся следующие уровни доступа (каждый из которых также может принимать значения - общедоступная, служебная, секретная):

- Уровень доступа к информации - определяет к информации с каким максимальным грифом может получить доступ сотрудник
- Уровень доступа к сети - определяет информацию с каким максимальным грифом, сотрудник может передать в сеть
- Уровень доступа к сменным носителям - определяет информацию с каким максимальным грифом, сотрудник может скопировать на сменный носитель

Квоты дискового пространства