

Управление устройствами
ВВОДА-ВЫВОДА И
ФАЙЛОВЫЕ СИСТЕМЫ

Управление вводом-выводом



Основные функции подсистемы ввода-вывода

1. Организация параллельной работы устройств ввода-вывода и процессора.
2. Согласование скоростей обмена и кэширование данных.
3. Разделение устройств и данных между процессами.
4. Обеспечение удобного логического интерфейса между устройствами и остальной частью системы.
5. Поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера.
6. Динамическая загрузка и выгрузка драйверов.
7. Поддержка нескольких файловых систем.
8. Поддержка синхронных и асинхронных операций ввода-вывода.

Операционная система



**Интерфейс драйвер – ядро
(Driver Kernel Interface, DKI)**

**Интерфейс драйвер – устройство
(Driver Device Interface, DDI)**

**Аппаратный низкоуровневый
интерфейс контроллер - устройство**

Функции драйвера

1. Обработка запросов записи-чтения от программного обеспечения управления устройствами. Постановка запросов в очередь
2. Проверка входных параметров запросов и обработка ошибок
3. Инициализация устройства и проверка статуса устройства
4. Управление энергопотреблением устройства.
5. Регистрация событий в устройстве
6. Выдача команд устройству и ожидание их выполнения возможно в заблокированном состоянии до поступления прерывания от устройства
7. Проверка правильности завершения операции
8. Передача запрошенных данных и статуса завершенной операции
9. Обработка нового запроса при незавершенном предыдущем запросе (для реентерабельных драйверов)

Классификация устройств ввода-вывода

Типы устройств по функциональному назначению:

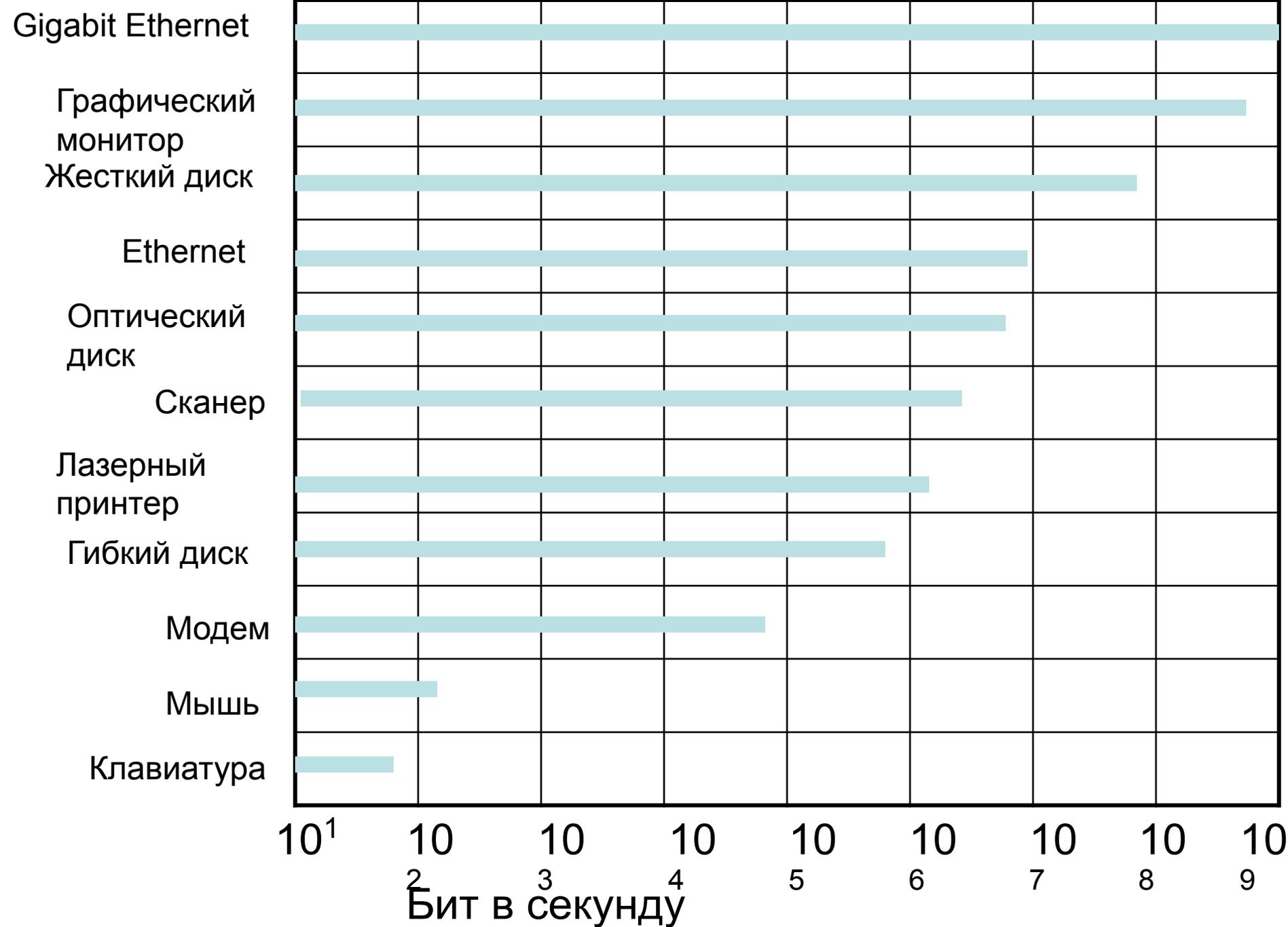
1. Работающие с пользователем. Используются для связи с пользователем компьютера (принтеры, дисплеи, клавиатура, манипуляторы (мышь, джойстик и т. п.).
2. Работающие с компьютером. Используются для связи с электронным оборудованием (диски, магнитные ленты, датчики, контроллеры, преобразователи и т. п.).
3. Коммуникации. Используются для связи с удаленными устройствами (модемы, адаптеры цифровых линий и др.).

Типы устройств по принципам функционирования:

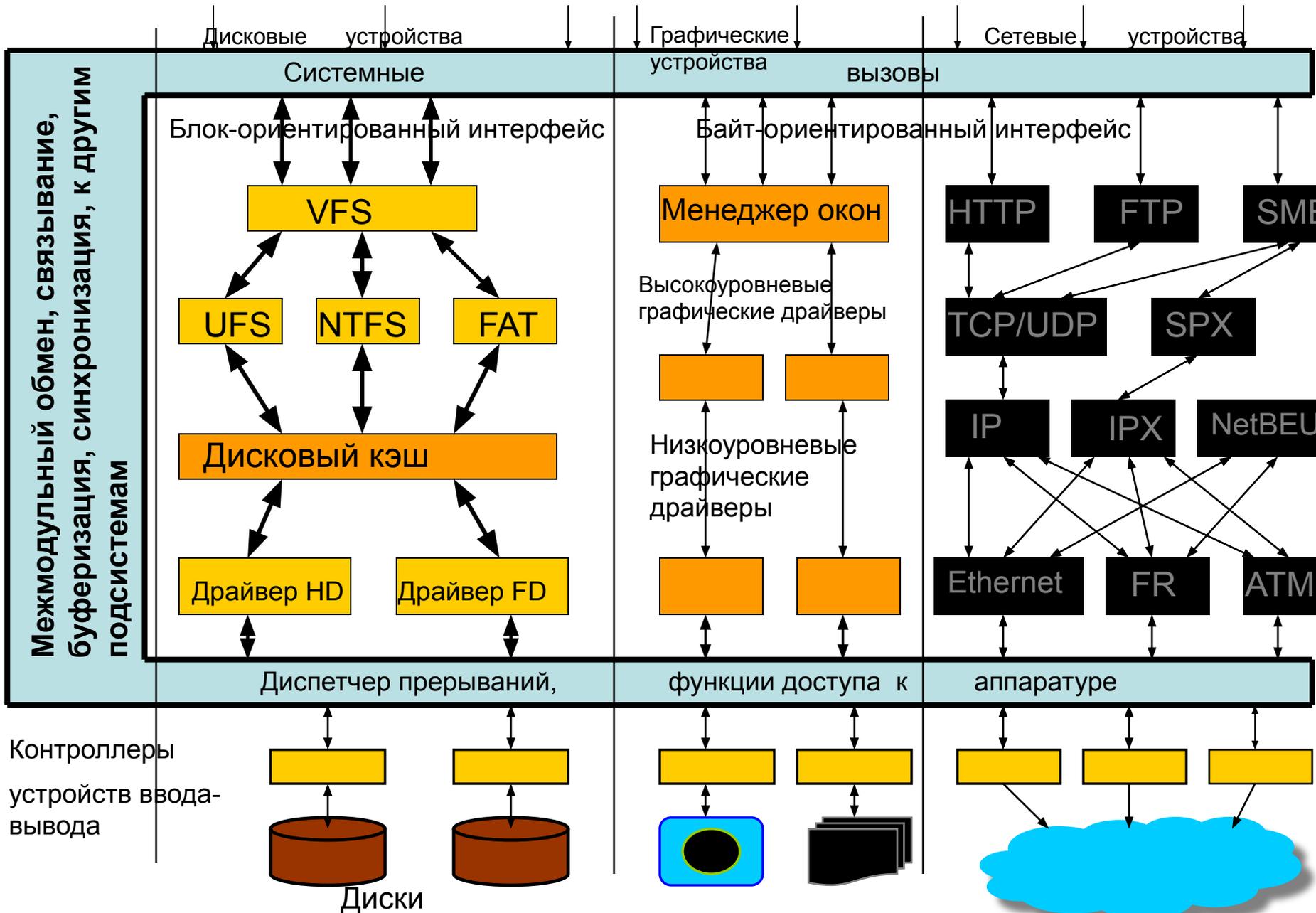
1. Блочные, хранящие информацию в виде адресуемых блоков фиксированного размера и позволяющие работать с каждым блоком независимо от других блоков (дисковые устройства).
2. Символьные, принимающие или предоставляющие поток символов без какой-либо структуры (принтеры, модемы, сетевые карты).

Различия в характеристиках устройств ввода-вывода

- ❖ Скорость передачи данных (на несколько порядков).
- ❖ Применение. Один и тот же тип устройства может требовать различного ПО и стратегии операционной системы (диск для хранения файлов приложений и файла подкачки, терминал пользователя и администратора).
- ❖ Сложность управления (для принтера относительно простой интерфейс управления, для диска – намного сложнее).
- ❖ Единицы передачи данных. Данные могут передаваться блоками или потоком байтов или символов.
- ❖ Представление данных. Различные устройства используют разные схемы кодирования данных, включая различную кодировку символов и контроль четности.
- ❖ Условия ошибок. Природа ошибок, способ сообщения о них, возможные ответы резко отличаются от одного устройства к другому.

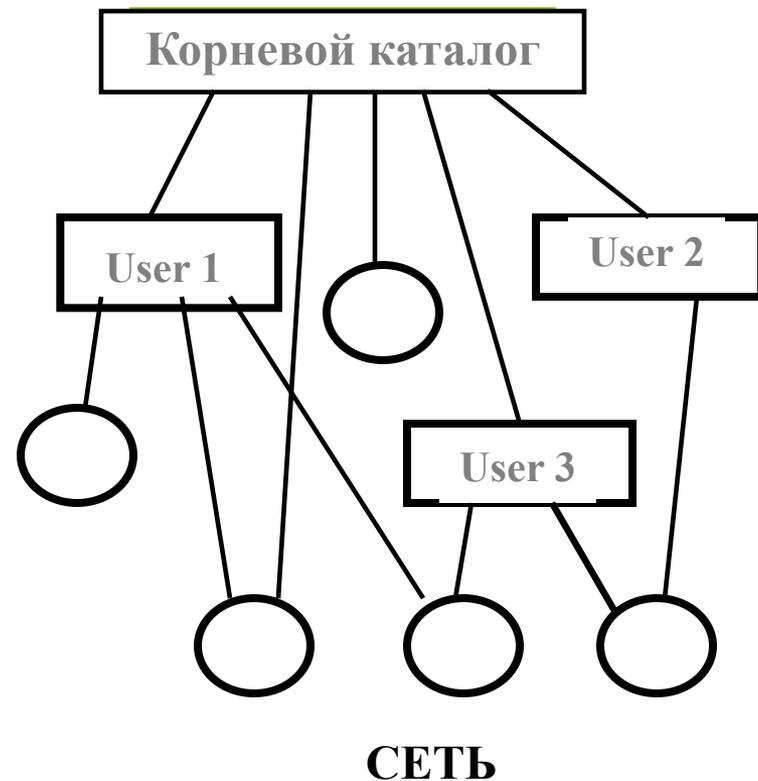
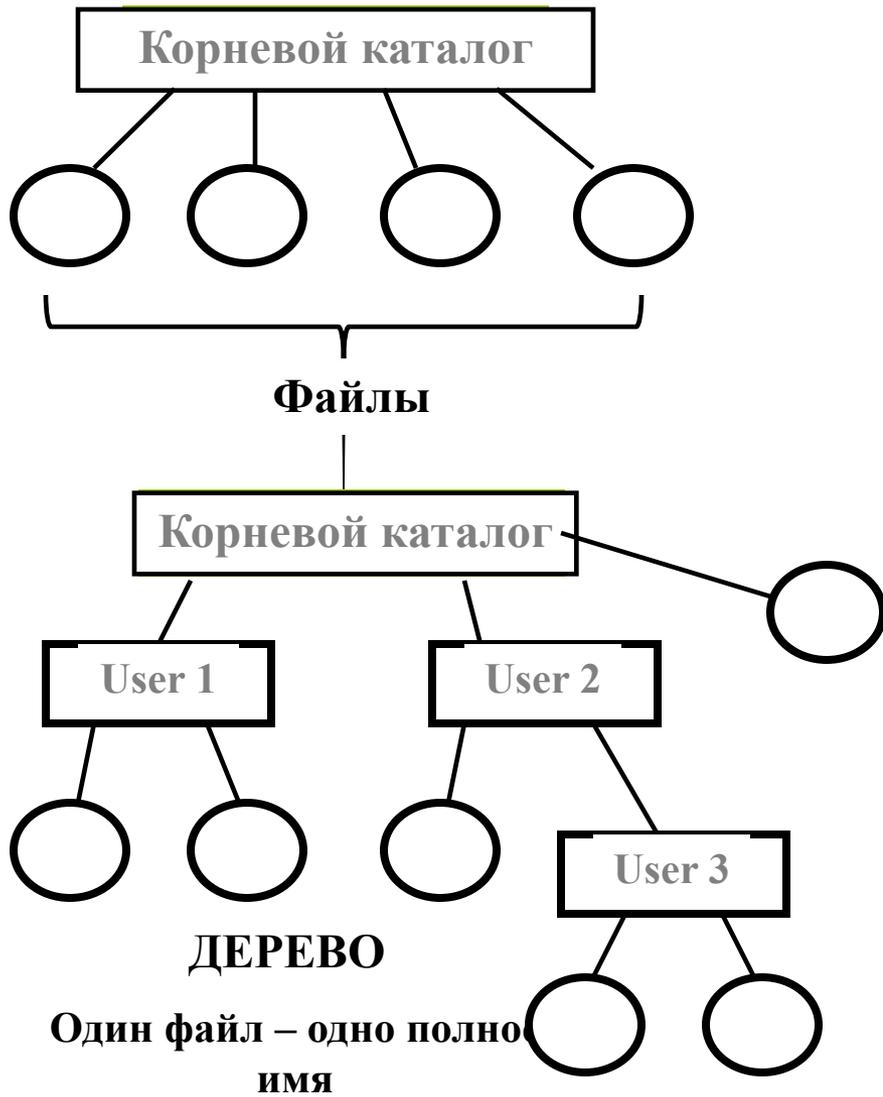


Многослойная модель подсистемы ввода-вывода



Файловая система – это часть операционной системы, включающая:

- совокупность всех файлов на различных носителях информации (магнитные диски, магнитные ленты, CD-ROM и т. п.);
- наборы структур данных, используемых для управления файлами (каталоги и дескрипторы файлов, таблицы распределения свободного и занятого пространства носителей информации);
- комплекс системных программных средств, реализующих различные операции над файлами (создание, чтение, запись, уничтожение, изменение свойств и др.).

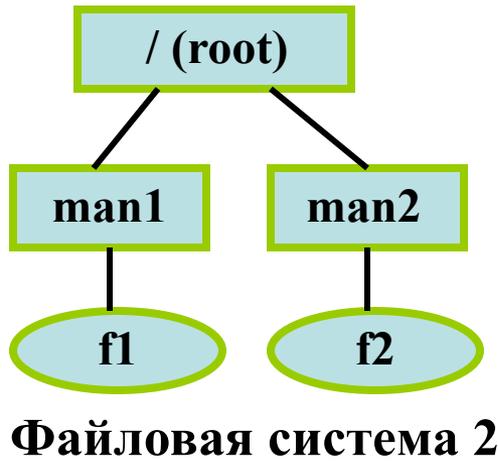
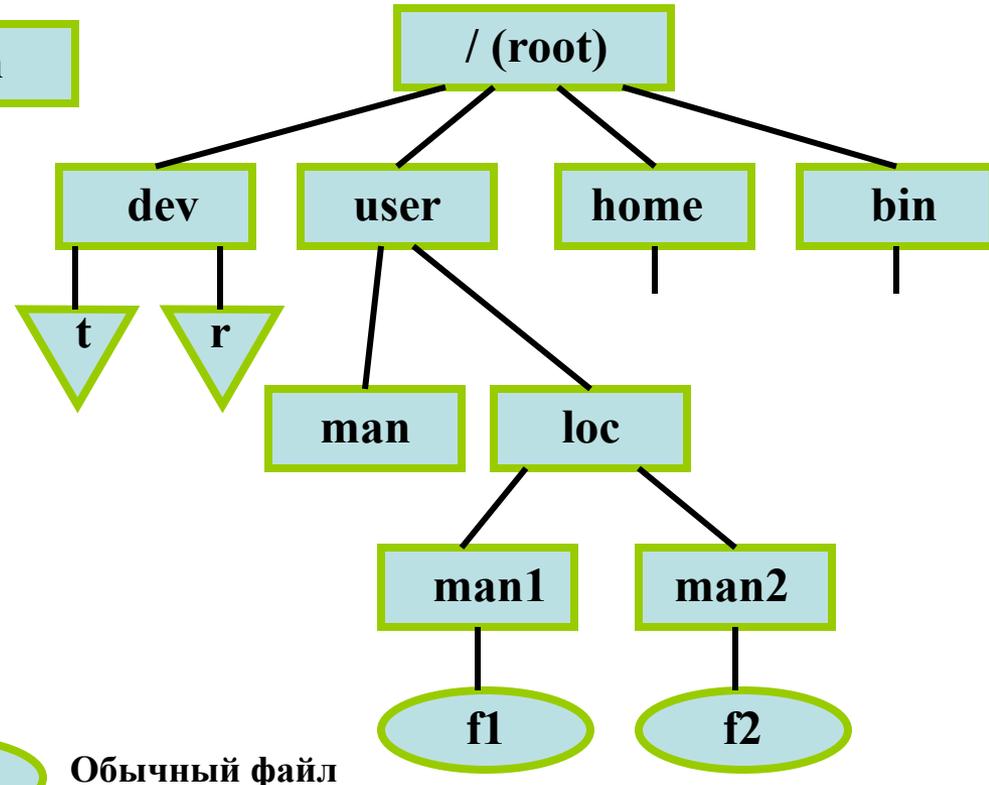
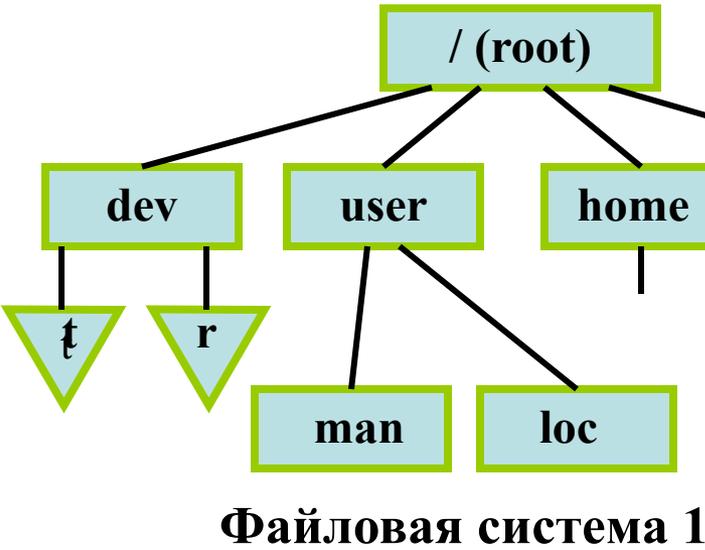


Один файл – много полных имен

Файловый каталог является связующим звеном между системой управления файлами и набором файлов

Атрибут	Значение
Тип файла	Обычный, каталог, специальный и т. д.
Владелец файла	Текущий владелец
Создатель файла	Идентификатор пользователя, создавшего файл
Пароль	Пароль для получения доступа к файлу
Время	Создания, последнего доступа, последнего изменения
Текущий размер файла	Количество байтов в записи
Максимальный размер	Количество байтов, до которого можно увеличивать размер
Флаг «только чтение»	0 – чтение-запись, 1 – только чтение
Флаг «скрытый»	0 – нормальный, 1 – не показывать в перечне файлов каталога
Флаг «системный»	0 – нормальный, 1 – системный
Флаг «архивный»	0 – заархивирован, 1- требуется архивация
Флаг ASCII/двоичный	0 – ASCII, 1 – двоичный
Флаг произвольного доступа	0 – только последовательный доступ, 1 – произвольный доступ
Флаг «временный» процесса	0 – нормальный, 1 – удаление после окончания работы

Монтирование



Общая файловая система после монтирования

1956 год — жёсткий диск IBM 350 в составе первого серийного компьютера IBM 305 RAMAC. Накопитель занимал ящик размером с большой холодильник и имел вес 971 кг, а общий объём памяти 50 вращавшихся в нём покрытых чистым железом тонких дисков диаметром 610 мм составлял около 5 миллионов 6-битных слов (3,5 Мб в пересчёте на 8-битные слова — байты).

1980 год — первый 5,25-дюймовый Winchester, Shugart ST-506, 5 Мб.

2013 год — Western Digital выпускает диск на 6 ТБ, используя 7 пластин.



Шесть типоразмеров жёстких дисков. Для масштаба рядом лежит дюймовая линейка

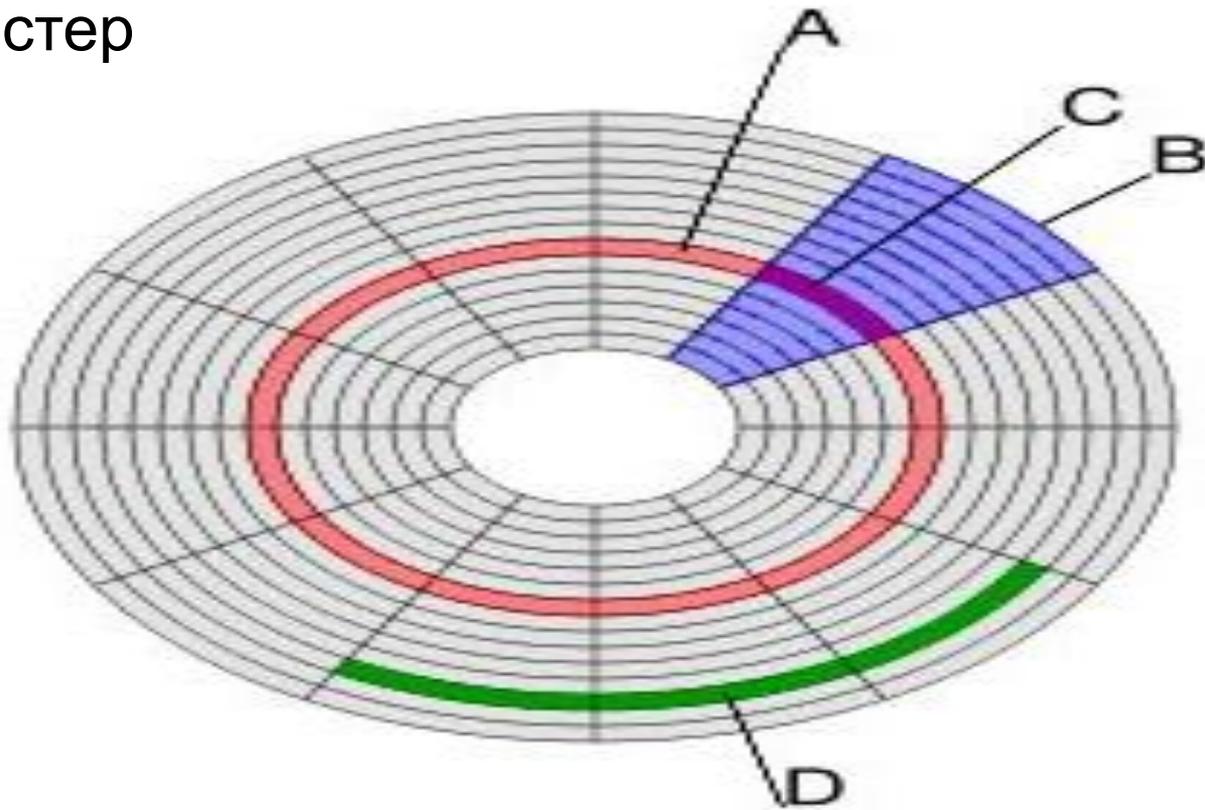
Структура диска:

(A) дорожка

(B) геометрический сектор

(C) сектор дорожки

(D) кластер



Адресация блоков данных диска

1 способ: c – h - s

c – номер цилиндра,

h – номер головки,

s – номер сектора

2 способ: LBA

= (c * H + h) * S + s - 1 H –
число рабочих поверхностей в
цилиндре, S – количество
секторов на дорожке

Системные идентификаторы: 06h
– FAT16, 07h – NTFS, 0Bh –
FAT32

Структура элемента таблицы разделов

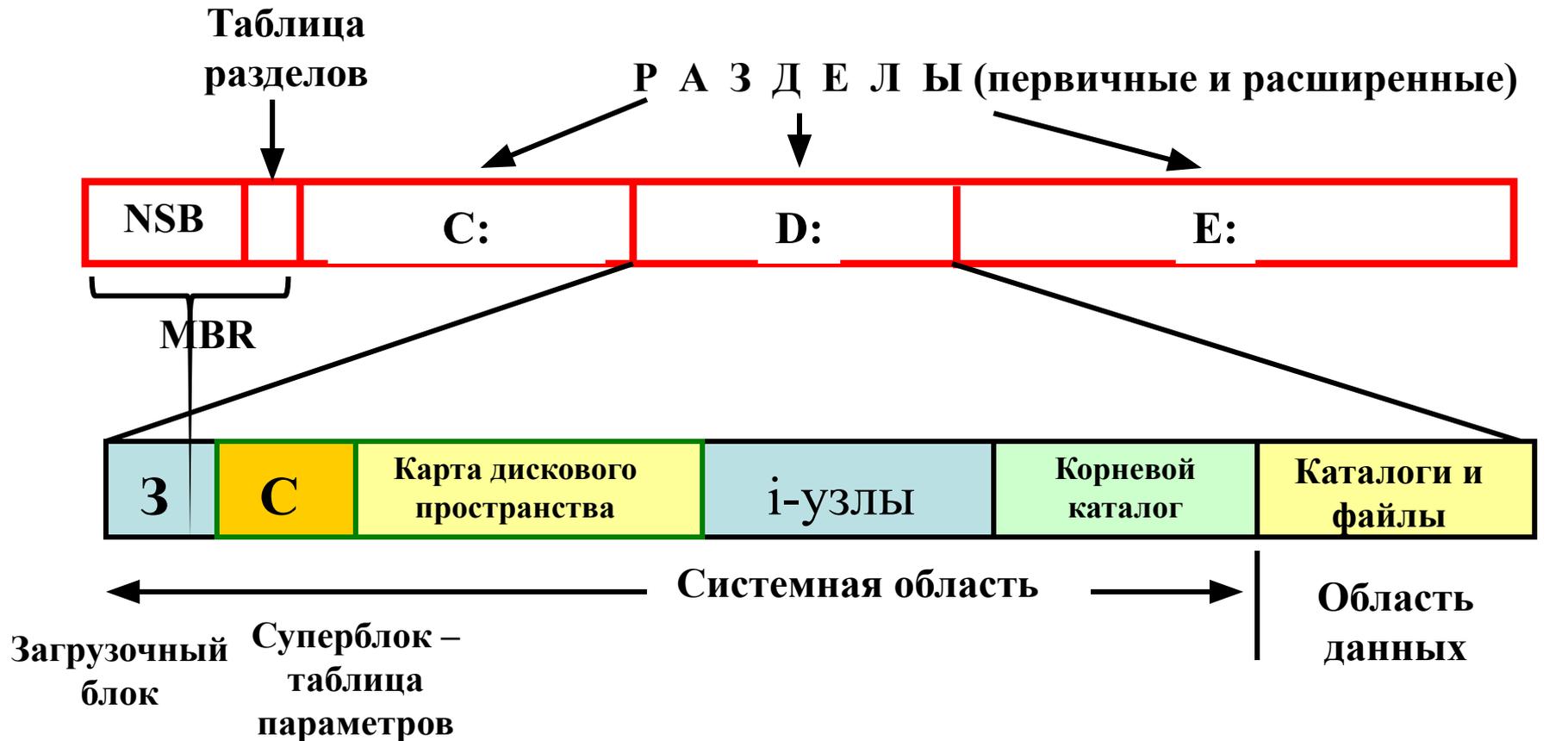
N п/п	Назначение	Размер в байтах
1.	Флаг активности раздела (Boot Indicator)	1
2.	Номер головки начала раздела	1
3.	Номер сектора и цилиндра загрузочного сектора раздела	2
4.	Системный идентификатор, показывающий на принадлежность к ОС и ФС	1
5.	Номер головки конца раздела	1
6.	Номер сектора и цилиндра последнего сектора раздела	2
7.	Младшее и старшее двухбайтовые слова относительного номера начального сектора	4
8.	Младшее и старшее двухбайтовые слова размера раздела в секторах	4
9.	Сигнатура-признак MBR и загрузочных секторов – 55AA h (только в конце MBR)	2

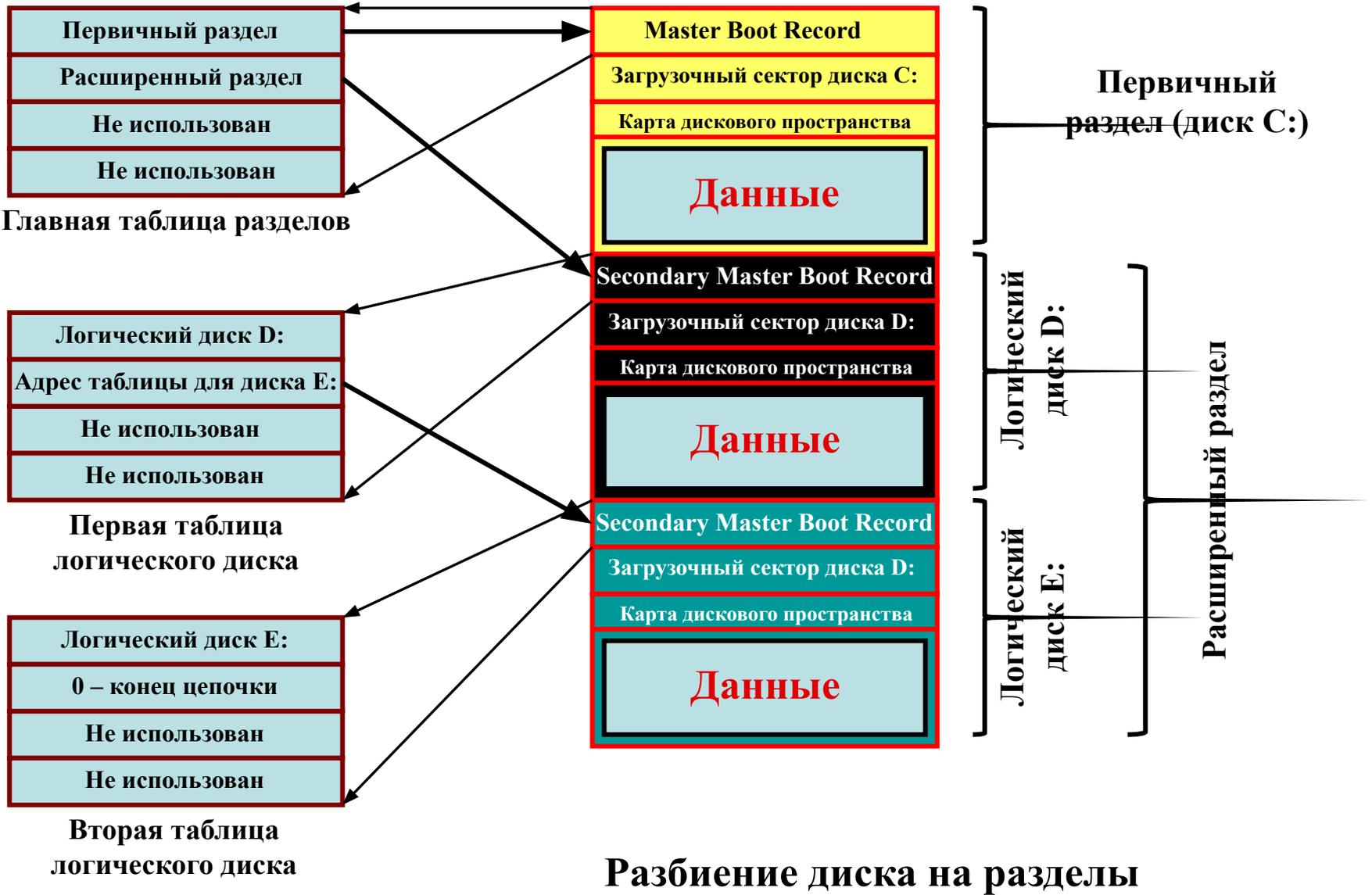
Физическая организация файловой системы

Структура диска: пластины, дорожки, цилиндры, секторы, кластеры.

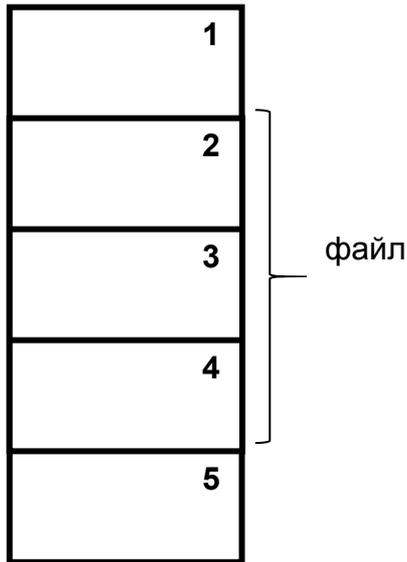
Низкоуровневое форматирование – создание дорожек и секторов.

Высокоуровневое форматирование – создание разделов и кластеров для определенной файловой системы или нескольких файловых систем.





непрерывное размещение

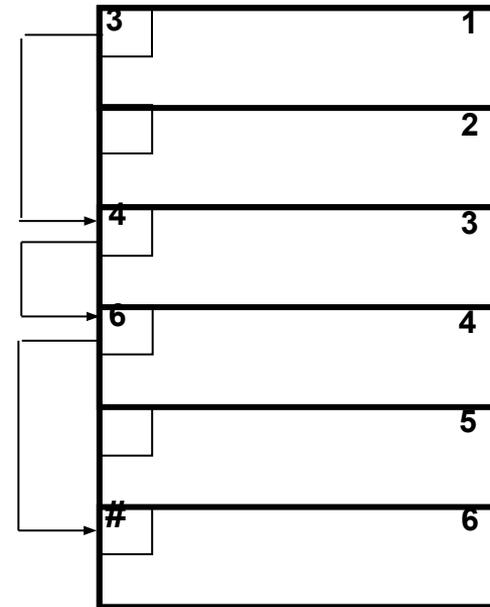


Достоинства: высокая скорость доступа, минимальный объем адресной информации, нет ограничений на размер файла.

Недостатки: нет возможностей для изменения размера файла, высокая степень возможной внешней фрагментации

Область применения –
компакт-диски

размещение в виде связанного списка кластеров



Достоинства: минимальная адресная информация, отсутствие внешней фрагментации, возможность изменения размеров файла.

Недостатки: медленный доступ, сложность доступа к произвольному блоку файла, некратность блока файла степени двойки.

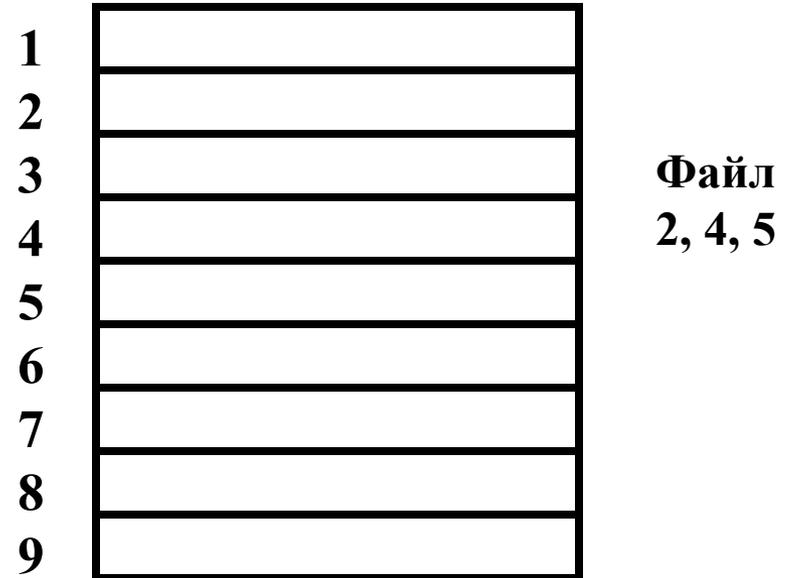
Связный список индексов



Все достоинства варианта А), быстрый доступ к произвольному кластеру файла, полное заполнение кластера, кратное степени двойки

Недостаток: рост адресной информации с увеличением емкости диска

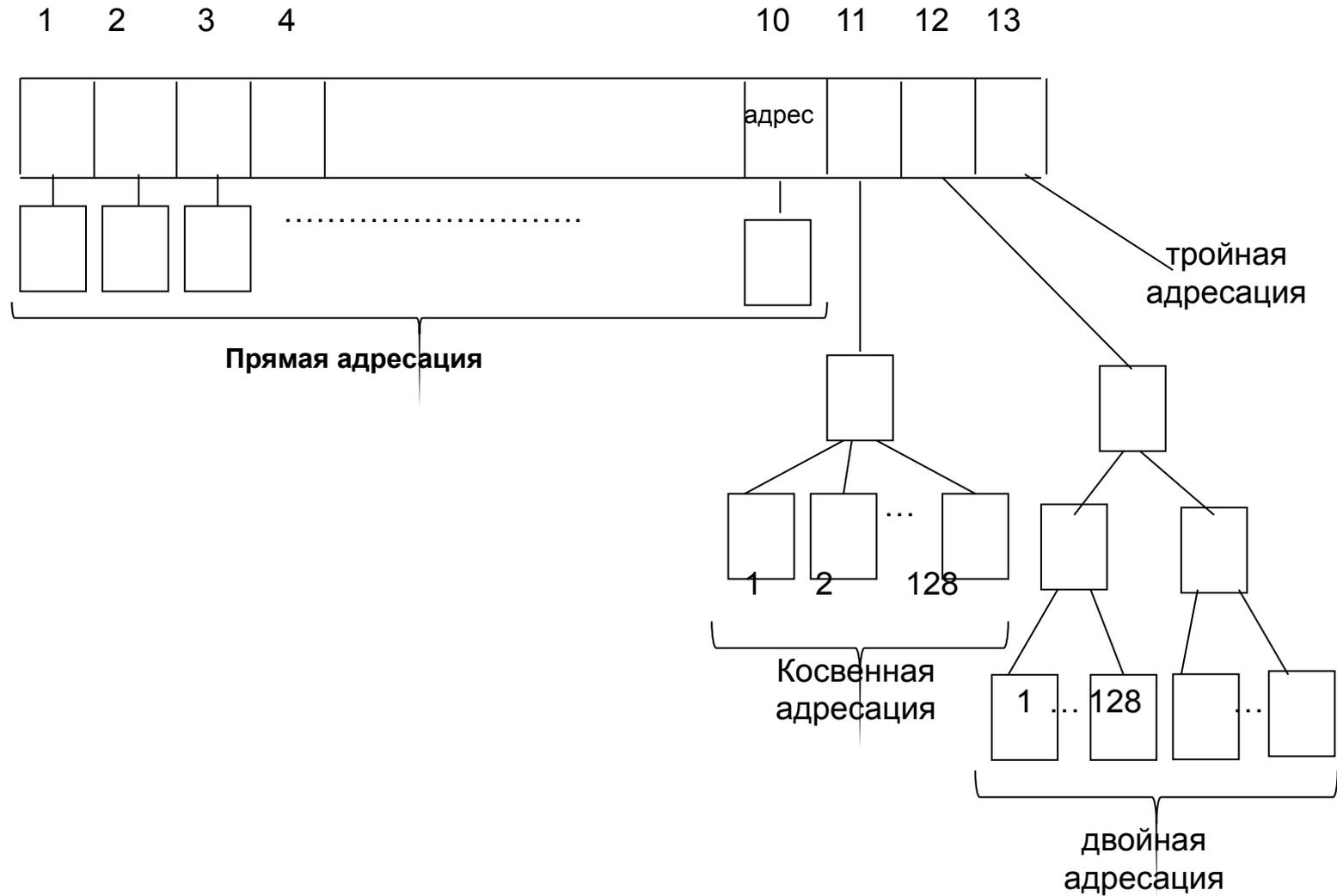
Перечень номеров кластеров

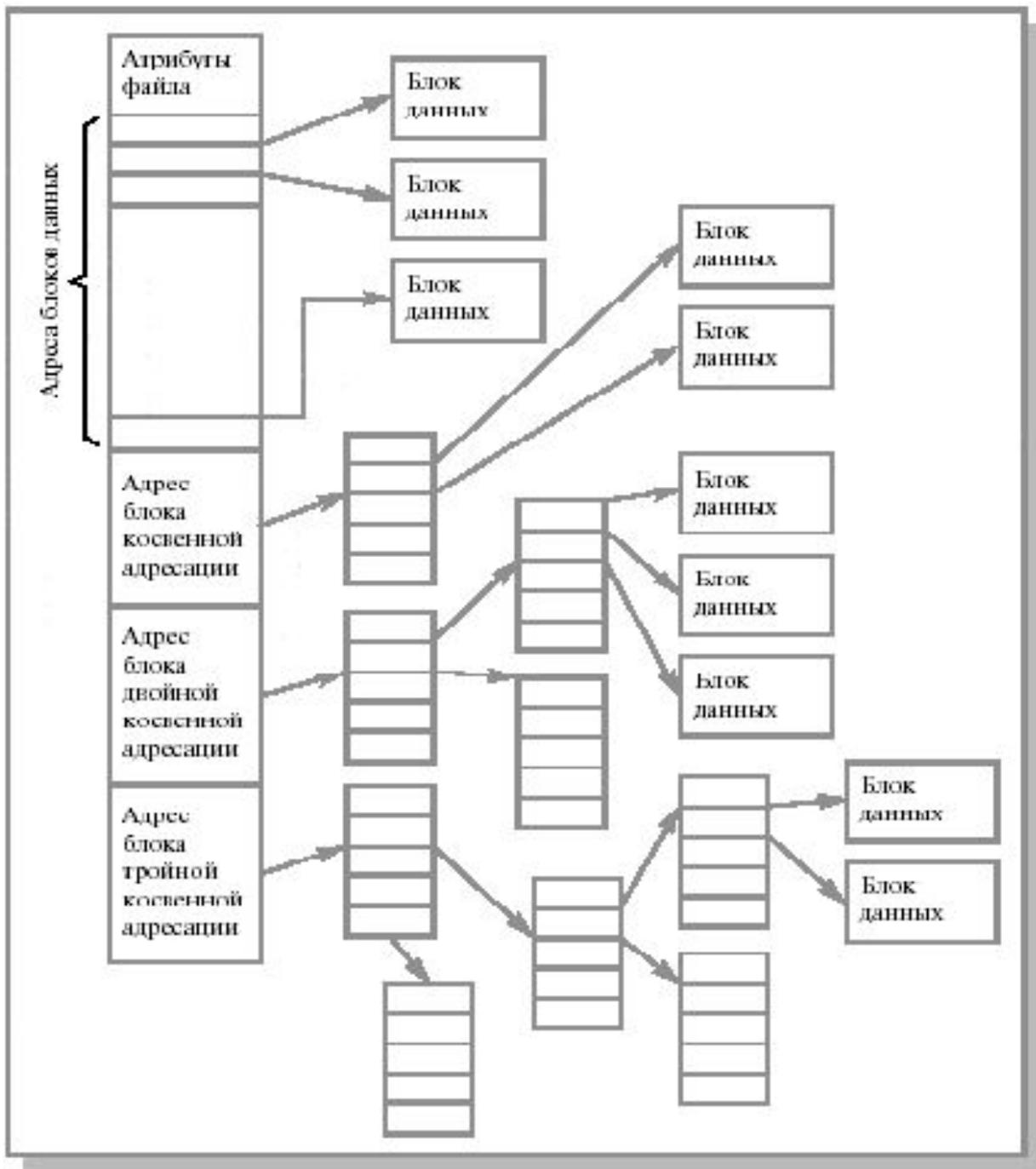


Достоинства: высокая скорость доступа к произвольному кластеру благодаря прямой адресации, отсутствие внешней фрагментации.

Недостаток: длина адреса зависит от размера файла и может быть значительной.

перечисление номеров блоков, занимаемых этим файлом





FAT



Структура записи каталога FAT

Длина поля	Описание
8 байт	Имя файла
3 байт	Расширение файла
1 байт	Атрибуты файла
1 байт	Зарезервировано
3 байт	Время создания
2 байт	Дата создания
2 байт	Дата последнего доступа
2 байт	Зарезервировано
2 байт	Время последней модификации
2 байт	Дата последней модификации
2 байт	Начальный кластер
4 байт	Размер файла

Разряды

8

3

1

10

2

2

2

4

Имя
файла

Расшире-
ние

Атрибуты
(обычный
файл или
директория)

Резервное
поле

Время

Дата

Номер
первого
блока

Размер

Disk Editor

Объекты Редактор Связь Просмотр **Информация** Средства Справка

Имя	Тип	Размер	Дата	Время	Кластер	Арх	T/C	Сис	Скр	Кат	Том
xПЕРСИ~1	RTF	591825	22.09.08	13:05	2	Арх					
x...d.o.c	...	4294967295	31.15.07	31:63	0		T/C	Сис	Скр		Том
x+.0.1.	...										Том
xAB_OC~1	DO										Том
UNERASE	EX										
Сектор 20											
SYMCFG	BI										
NLIB100	RT										
IF1	BA										
HELP	TX										
DISKEDIT	EX										
DE	TX										
xC.<.5.=	.B										Том
x".5.:.A	.B										Том
xEKSTO~1	TX										Том
xC.r.a.s	.B										Том
CRASH1K	TX										Том
xC.<.5.=	.B										Том
x".5.:.A	.B										Том
xEKSTO~1	TX										Том
		25	31.10.08	10:20	986	Арх					

Информация о диске

Диск _____

Диск A: 1.44M Тип: 3 1/2" дискета

Логические характеристики

Байтов в секторе: 512 Тип FAT: 12-бит

Секторов в кластере: 1

Количество кластеров: 2 47

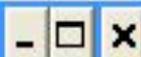
Физические характеристики

Сторон: 2 Дорожек: 80 Секторов на дорожке: 18

Номер диска: 0 Hex

OK

C:\ A:\DISKEDIT.EXE



Disk Editor

Объекты	Редактор	Связь	Просмотр	Информация	Средства	Справка						
Имя	Тип	Размер	Дата	Время	Кластер	Арх	T/Ч	Сис	Скр	Кат	Том	
xПЕРСИ~1	RTF	591825	22.09.08	13:05	2	Арх						
x..d.o.c	...	4294967295	31.15.07	31:63	0		T/Ч	Сис	Скр		Том	
x*.0.1.	...	2687060	1.02.80	0:02	0		T/Ч	Сис	Скр		Том	
xАБ_0С~1	DOC	49152	28.10.08	9:48	2	Арх						
UNERASE	EXE	114489	20.03.93	8:00	98	Арх						
Сектор 20												
SYMCFG	BIN	677	31.10.08	9:15	322	Арх						
NLIB100	RTL	200438	20.03.93	8:00	324	Арх						
IF1	BAT	251	2.10.08	6:53	716	Арх						
HELP	TXT	4515	2.10.08	6:50	717	Арх						
DISKEDIT	EXE	132489	20.03.93	8:10	726	Арх						
DE	TXT	233	16.02.08	11:13	985	Арх		Сис	Скр			
xC.<.5.=	.B.	4294967295	31.15.07	0:00	0		T/Ч	Сис	Скр		Том	
x".5.:.A	.B.	70911038	20.01.82	0:01	0		T/Ч	Сис	Скр		Том	
xЕКСТО~1	TXT	0	31.10.08	8:37	0	Арх						
xC.r.a.s	.h.	4294901760	20.03.80	0:03	0		T/Ч	Сис	Скр		Том	
CRASH1K	TXT	11937	31.10.08	8:39	986	Арх						
xC.<.5.=	.B.	4294967295	31.15.07	0:00	0		T/Ч	Сис	Скр		Том	
x".5.:.A	.B.	70911038	20.01.82	0:01	0		T/Ч	Сис	Скр		Том	
xЕКСТО~1	TXT	25	31.10.08	10:20	986	Арх						

Корневой каталог
A:\

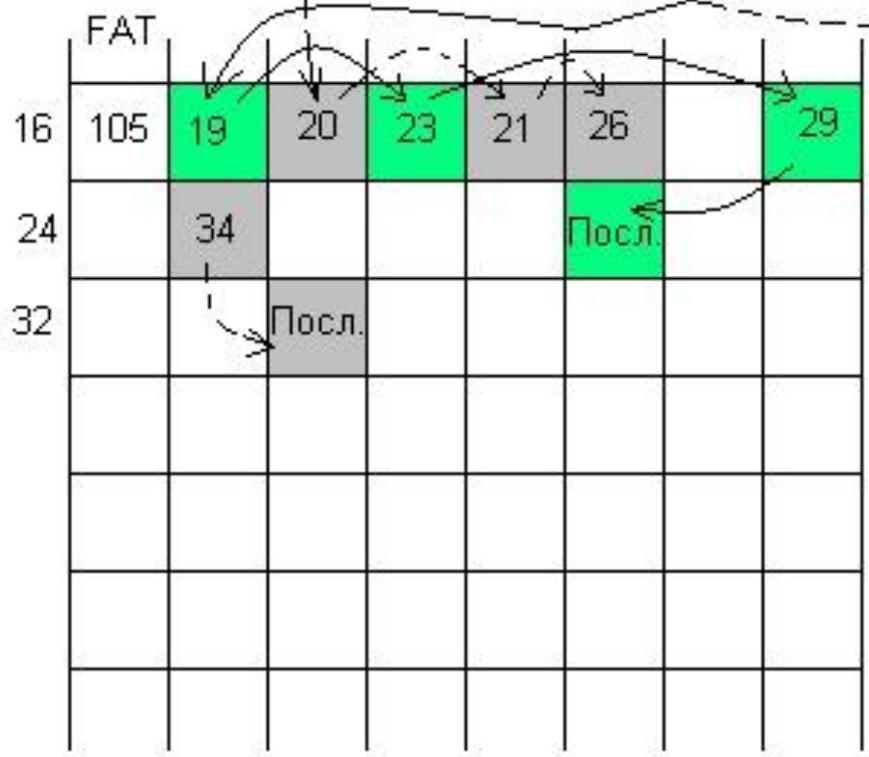
Сектор 20
Смещение 64, hex 40

Таблица FAT (как основная копия, так и резервная) состоит из массива индексных указателей, количество которых равно количеству кластеров области данных. Между кластерами и индексными указателями имеется взаимно однозначное соответствие — нулевой указатель соответствует нулевому кластеру и т. д.

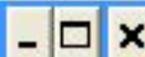
Индексный указатель может принимать следующие значения, характеризующие состояние связанного с ним кластера:

- кластер свободен (не используется);
- кластер используется файлом и не является последним кластером файла; в этом случае индексный указатель содержит номер следующего кластера файла;
- последний кластер файла;
- дефектный кластер;
- резервный кластер.

Каталог	
file1	17
file 2	18



- элементы, указывающие на кластере файла file1
- элементы, указывающие на кластере файла file 2



Disk Editor

Объекты	Редактор	Связь	Просмотр	Информация	Средства	Справка	
654	655	656	657	658	659	660	661
662	663	664	665	666	667	668	669
670	671	672	673	674	675	676	677
678	679	680	681	682	683		
Сектор 3							
684	685	686	687	688	689	690	691
692	693	694	695	696	697	698	699
700	701	702	703	704	705	706	707
708	709	710	711	712	713	714	715
<EOF>	<EOF>	718	719	720	721	722	723
724	725	<EOF>	727	728	729	730	731
732	733	734	735	736	737	738	739
740	741	742	743	744	745	746	747
748	749	750	751	752	753	754	755
756	757	758	759	760	761	762	763
764	765	766	767	768	769	770	771
772	773	774	775	776	777	778	779
780	781	782	783	784	785	786	787
788	789	790	791	792	793	794	795
796	797	798	799	800	801	802	803
804	805	806	807	808	809	810	811

FAT (1-я копия)

Диск A:

Сектор 3

Кластер 716, hex 2CC

Disk Editor

Объекты	Редактор	Связь	Просмотр	Информация	Средства	Справка					
Имя	Тип	Размер	Дата	Время	Кластер	Арх	Т/Ч	Сис	Скр	Кат	Том
PERСИ~1	RTF	591825	22.09.08	13:05	2	Арх					
..d.o.c	...	4294967295	31.15.07	31:63	0		Т/Ч	Сис	Скр		Том
+.0.1.	▲.	2687060	1.02.80	0:02	0		Т/Ч	Сис	Скр		Том
AB_OC~1	DOC										
NERASE	EXE										
ектор 20											
YMCFG	BIN										
LIB100	RTL										
F1	BAT										
ELP	TXT										
ISKEDIT	EXE										
E	TXT										
C.<.5.=	.B.	429									
"5.:.A	.B.	?									
EKCTO~1	TXT										
C.r.a.s	.h.	429									
RASH1K	TXT										
C.<.5.=	.B.	429									
"5.:.A	.B.	70911035	20.01.82	0:01	0		Т/Ч	Сис	Скр		Том
EKCTO~1	TXT	25	31.10.08	10:20	986	Арх					

Выбор диапазона кластеров

Диск A:

Верные номера кластеров
с 2 по 2 848.

Начальный кластер: [216 -]

Конечный кластер: [2848 -]

OK

Отменить

Disk Editor

Объекты Редактор Связь Просмотр Информация Средства Справка

Кластер 716, Сектор 747

00000000:	0D	0A	40	65	63	68	6F	20	-	6F	6E	0D	0A	72	65	6D	20	·.Echo on..rem
00000010:	4D	59	43	4F	50	59	2E	42	-	41	54	20	EA	EE	EF	E8	F0	MУСОPY.BAT ьюяшЕ
00000020:	F3	E5	F2	20	EF	F0	EE	E8	-	E7	E2	EE	EB	FC	ED	EE	E5	exС яЕюшчтюттЭюх
00000030:	20	F7	E8	F1	EB	EE	20	F4	-	E0	E9	EB	EE	E2	0D	0A	72	ушёню Ырщют..r
00000040:	65	6D	20	E2	20	EA	E0	F2	-	E0	EB	EE	E3	2E	0D	0A	72	em т ьрЕрыюу...r
00000050:	65	6D	20	C8	F1	EF	EE	EB	-	FC	E7	F3	E5	F2	F1	FF	20	em .ёяюыЧехСё
00000060:	F1	EB	E5	E4	F3	FE	F9	E8	-	E9	20	F1	E8	ED	F2	E0	EA	ёыхфе шщ ёшэЕрь
00000070:	F1	E8	F1	3A	0D	0A	72	65	-	6D	20	6D	79	63	6F	70	79	ёшё:..rem mусору
00000080:	20	64	69	72	20	F4	E0	E9	-	EB	31	20	F4	E0	E9	EB	32	dir Ырщ1 Ырщ2
00000090:	20	2E	2E	2E	20	0D	0A	3A	-	67	55	74	66	69	6C	65	0D:getfile.
000000A0:	0A	69	66	20	22	25	31	22	-	3D	7D	22	22	20	67	6F	74	.if "%1"==" " got
000000B0:	6F	20	65	6E	64	0D	0A	63	-	6F	70	79	20	25	31	20	25	o end..copy %1 %
000000C0:	32	0D	0A	73	68	69	66	74	-	20	2F	32	0D	0A	70	61	75	2..shift /2..pau
000000D0:	73	65	0D	0A	67	6F	74	6F	-	20	67	65	74	66	69	6C	65	se..goto getfile
000000E0:	0D	0A	3A	65	6E	64	0D	0A	-	0D	0A	65	63	68	6F	20	45	..:end....echo E
000000F0:	78	69	74	21	0D	0A	70	61	-	75	73	65	00	00	00	00	00	xit!..pause.....
00000100:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000110:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000120:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000130:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

Кластеры 716 - 2 848

Кластер 716
Смещение 0, hex 0

C:\ A:\DISKEDIT.EXE

Disk Editor

Объекты Редактор Связь Просмотр Информация Средства Справка

Имя	Тип	Размер	Дата	Время	Кластер	Арх	Т/Ч	Сис	Скр	Кат	Том
x..d.o.c ...		4294967295	31.15.07	31:63	0		Т/Ч	Сис	Скр		Том
x+.0.1.▲.		2687060	1.02.80	0:02	0		Т/Ч	Сис	Скр		Том
xAB_OC~I	DOC	49152	28.10.08	9:48	2	Арх					
UNERASE	EXE	114489	20.03.93	8:00	98	Арх					
Сектор 20											
SYMCFG	BIN	677	31.10.08	9:15	322	Арх					
NLIB100	RTL	200438	20.03.93	8:00	324	Арх					
xF1	BAT	251	2.10.08	6:53	716	Арх					
HELP	TXT	4515	2.10.08	6:30	717	Арх					
DISKEDIT	EXE	132489	20.03.93	8:00	726	Арх					
DE	TXT	233	16.02.08	11:13	985	Арх		Сис	Скр		
xC.<.5.=	.B.	4294967295	31.15.07	0:00	0		Т/Ч	Сис	Скр		Том
x".5.:.A	.B.	70911038	20.01.82	0:01	0		Т/Ч	Сис	Скр		Том
xEKCTO~1	TXT	0	31.10.08	8:37	0	Арх					
xC.r.a.s	.h.	4294901760	20.03.80	0:03	0		Т/Ч	Сис	Скр		Том
CRASH1K	TXT	11937	31.10.08	8:39	986	Арх					
xC.<.5.=	.B.	4294967295	31.15.07	0:00	0		Т/Ч	Сис	Скр		Том
x".5.:.A	.B.	70911038	20.01.82	0:01	0		Т/Ч	Сис	Скр		Том
xEKCTO~1	TXT	25	31.10.08	10:20	986	Арх					
x5♦3♦>♦@	...	4294967295	31.15.07	31:63	0		Т/Ч	Сис	Скр		Том

Корневой каталог

A:\

Сектор 20

Смещение 448, hex 1C0

C:\ A:\DISKEDIT.EXE

Disk Editor

Объекты	Редактор	Связь	Просмотр	Информация	Средства	Справка
654	655	656	657	658	659	660
662	663	664	665	666	667	668
670	671	672	673	674	675	676
678	679	680	681	682	683	684
Сектор 3						
684	685	686	687	688	689	690
692	693	694	695	696	697	698
700	701	702	703	704	705	706
708	709	710	711	712	713	714
<EOF>	0	718	719	720	721	722
724	725	<EOF>	727	728	729	730
732	733	734	735	736	737	738
740	741	742	743	744	745	746
748	749	750	751	752	753	754
756	757	758	759	760	761	762
764	765	766	767	768	769	770
772	773	774	775	776	777	778
780	781	782	783	784	785	786
788	789	790	791	792	793	794
796	797	798	799	800	801	802
804	805	806	807	808	809	810

FAT (1-я копия)
Диск A:Сектор 3
Кластер 716, hex 2CC



Disk Editor

Объекты Редактор Связь Просмотр Информация Средства Справка

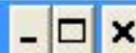
Кластер 716, Сектор 747

00000000:	0D	0A	40	65	63	68	6F	20	-	6F	6E	0D	0A	72	65	6D	20	.Echo on..rem
00000010:	4D	59	43	4F	50	59	2E	42	-	41	54	20	EA	EE	EF	E8	F0	MYSORU.BAT ъмяшЕ
00000020:	F3	E5	F2	20	EF	F0	EE	E8	-	E7	E2	EE	EB	FC	ED	EE	E5	exС яЕюшчтюдтѣэюх
00000030:	20	F7	E8	F1	EB	EE	20	F4	-	E0	E9	EB	EE	E2	0D	0A	72	ушёю Ырщют..r
00000040:	65	6D	20	E2	20	EA	E0	F2	-	E0	EB	EE	E3	2E	0D	0A	72	em т ърЕрыюу...r
00000050:	65	6D	20	C8	F1	EF	EE	EB	-	FC	E7	F3	E5	F2	F1	FF	20	em .ёяюыѣчехСё
00000060:	F1	EB	E5	E4	F3	FE	F9	E8	-	E9	20	F1	E8	ED	F2	E0	EA	ёыхфе щ ёшэСрь
00000070:	F1	E8	F1	3A	0D	0A	72	65	-	6D	20	6D	79	63	6F	70	79	ёше:..rem мусор
00000080:	20	64	69	72	20	F4	E0	E9	-	EB	31	20	F4	E0	E9	EB	32	dir Ырщы1 Ырщы2
00000090:	20	2E	2E	2E	20	0D	0A	3A	-	67	75	74	66	69	6C	65	0D:getfile.
000000A0:	0A	69	66	20	22	25	31	22	-	3D	7D	22	22	20	67	6F	74	.if "%1"==" " got
000000B0:	6F	20	65	6E	64	0D	0A	63	-	6F	70	79	20	25	31	20	25	o end..copy %1 %
000000C0:	32	0D	0A	73	68	69	66	74	-	20	2F	32	0D	0A	70	61	75	2..shift /2..pau
000000D0:	73	65	0D	0A	67	6F	74	6F	-	20	67	65	74	66	69	6C	65	se..goto getfile
000000E0:	0D	0A	3A	65	6E	64	0D	0A	-	0D	0A	65	63	68	6F	20	45	..:end....echo E
000000F0:	78	69	74	21	0D	0A	70	61	-	75	73	65	00	00	00	00	00	xit!..pause.....
00000100:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000110:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000120:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00
00000130:	00	00	00	00	00	00	00	00	-	00	00	00	00	00	00	00	00

Кластеры 716 - 2 848

Кластер 716
Смещение 0, hex 0

C:\ A:\UNERASE.EXE



UnErase

Файл Поиск Параметры Справка

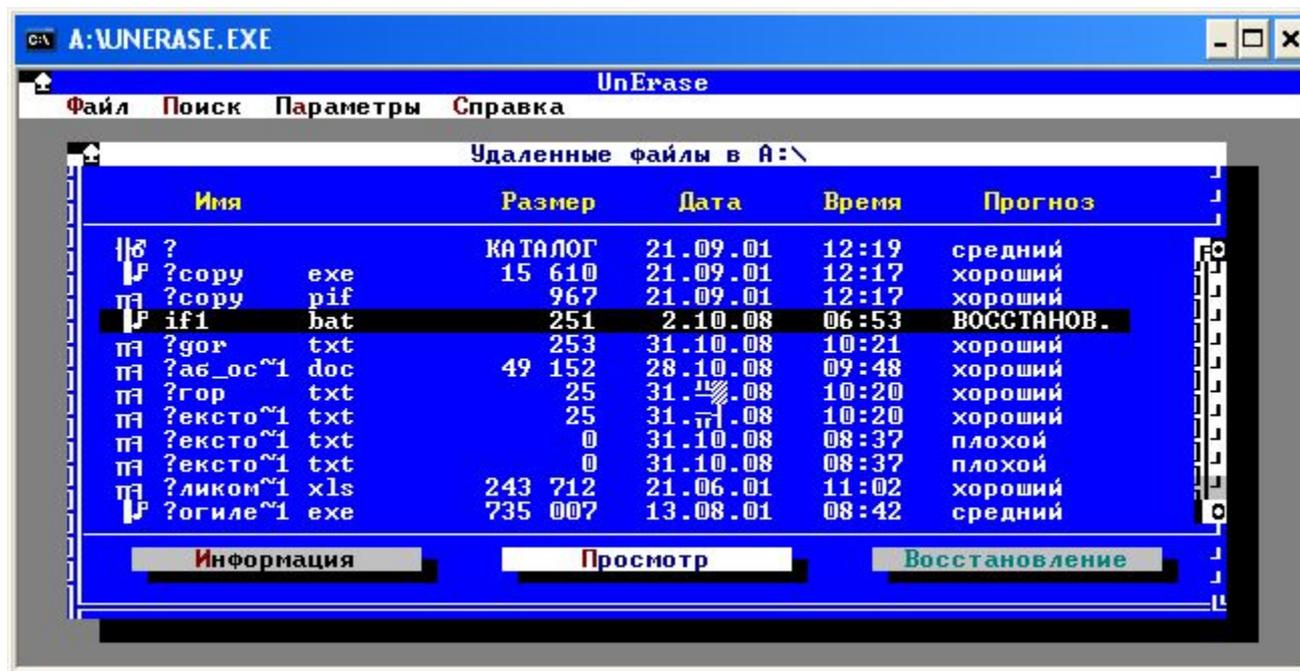
Удаленные файлы в A:\

Имя	Размер	Дата	Время	Прогноз
з ?	КАТАЛОГ	21.09.01	12:19	средний
Р ?сору ехе	15 610	21.09.01	12:17	хороший
п ?сору pif	967	21.09.01	12:17	хороший
Р ?f1 bat	251	2.10.08	06:53	хороший
п ?гор txt	253	31.10.08	10:21	хороший
п ?аб_ос~1 doc	49 152	11.10.08	09:48	хороший
п ?гор txt	25	31.10.08	10:20	хороший
п ?ексто~1 txt	25	31.10.08	10:20	хороший
п ?ексто~1 txt	0	31.10.08	08:37	плохой
п ?ексто~1 txt	0	31.10.08	08:37	плохой
п ?ликом~1 xls	243 712	21.06.01	11:02	хороший
Р ?огиле~1 ехе	735 007	13.08.01	08:42	средний

Информация

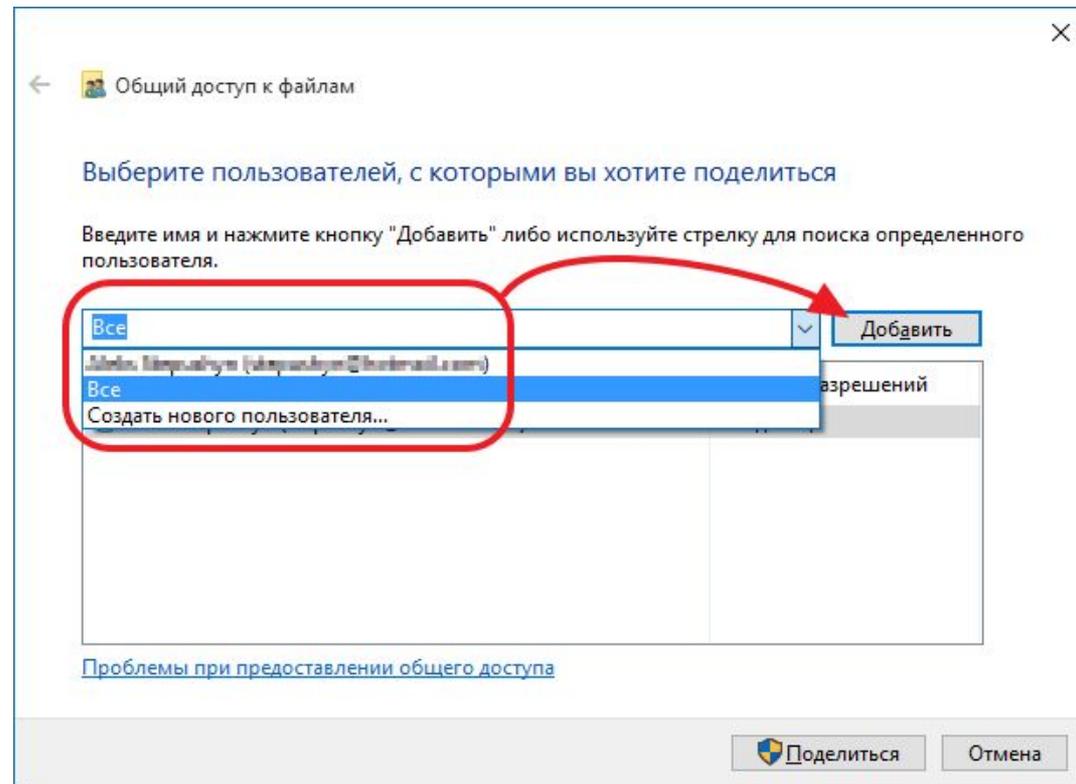
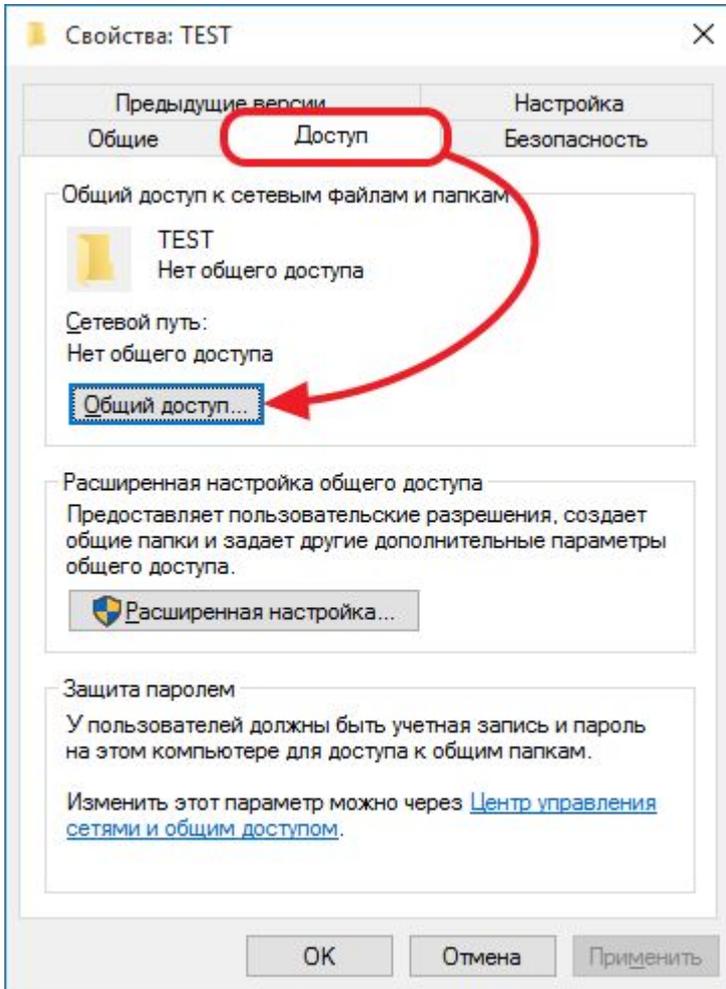
Просмотр

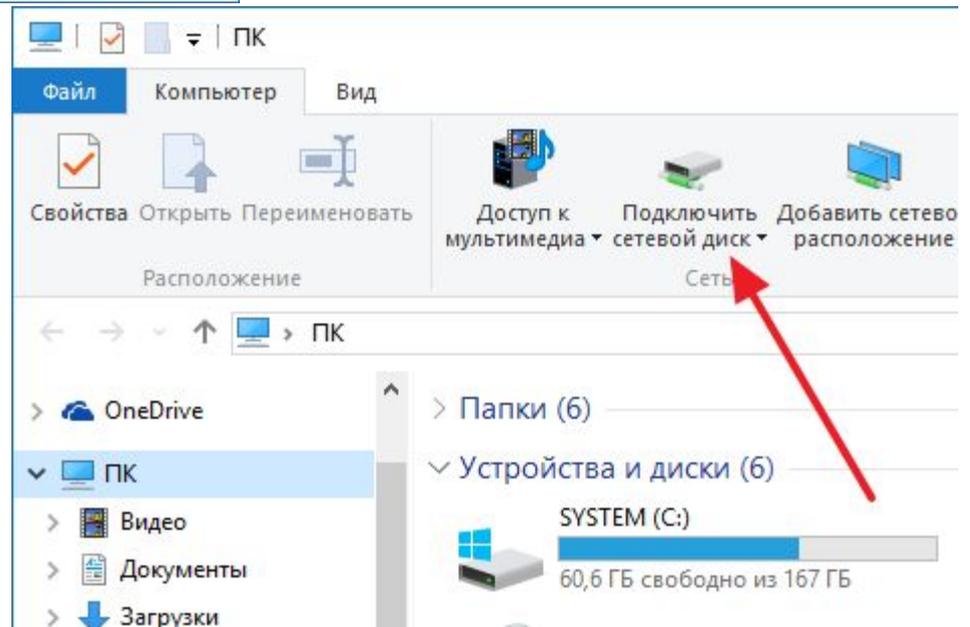
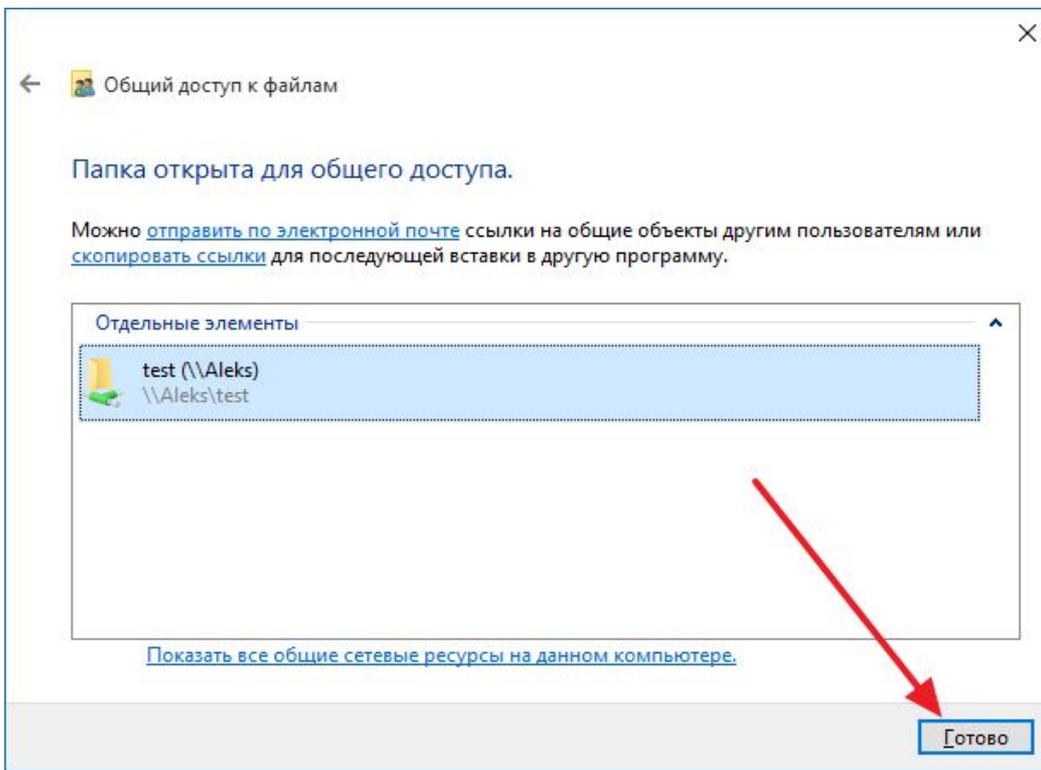
Восстановление



Подключение сетевого диска

Сетевой диск— назначенный логический диск (папка), который служит для хранения «общих» файлов, доступных для всех пользователей, на других персональных компьютерах, включенных в общую локальную сеть.





← Подключение сетевого диска

Какую сетевую папку вы хотите подключить?

Укажите букву диска для подключения и папку, к которой вы хотите подключиться:

Диск: Z: ▾

Папка: \\notebook\tmp ▾ Обзор...

Пример: \\сервер\общий_ресурс

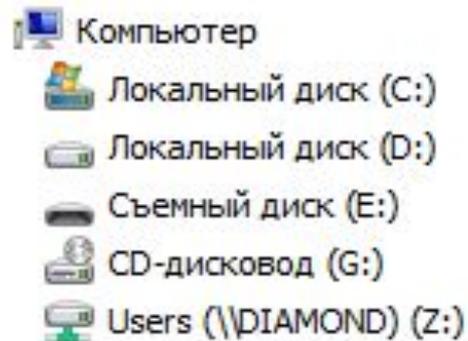
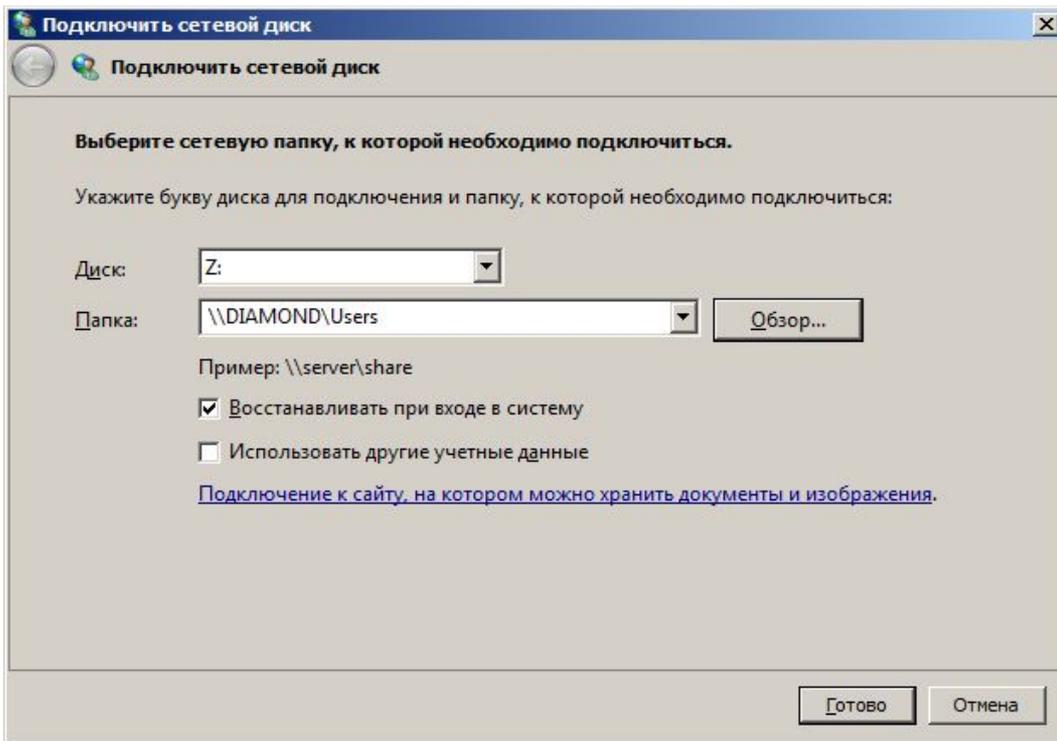
Восстанавливать подключение при входе в систему

Использовать другие учетные данные

[Подключение к веб-сайту, на котором вы можете хранить документы и изображения.](#)

Готово

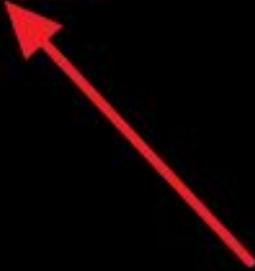
Отмена



```
Командная строка
Microsoft Windows [Version 10.0.10240]
(c) Корпорация Майкрософт (Microsoft Corporation), 2015 г. Все права защищены.

C:\Users\Aleks>net use z: \\notebook\tmp
Команда выполнена успешно.

C:\Users\Aleks>
```



Основные характеристики файловых систем

FAT Имя файла	Разрядность указателя	Число кластеров	Максимальный объем кластера	Максимальный размер раздела
FAT12 8.3	12	4096	4 Кбайт	16 Мбайт
FAT16 8.3 255.3	16	65536	64 Кбайт	4 Гбайт
FAT 32 255.3	32	4 Г	32 Кбайт	2³² по 32 Кбайт
NTFS 255.3	64	2⁶⁴	4 Кбайт	2⁶⁴ по 4 Кбайт

Программа Fdisk автоматически определяет размер кластера на основе выбранной файловой системы и размера раздела. Существует недокументированный параметр команды Format, позволяющий явно указать размер кластера:

Format /z:n, где n – размер кластера в байтах, кратный 512.

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.**

Файл NTFS – не просто линейная последовательность байтов, характерная для FAT-систем и Unix, а множество атрибутов, представляемых в виде потока байтов. Файл имеет несколько коротких потоков (имя, идентификатор и др.) и один или несколько длинных потоков с данными (ff:stream1, ff:stream2 и др.).

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***).

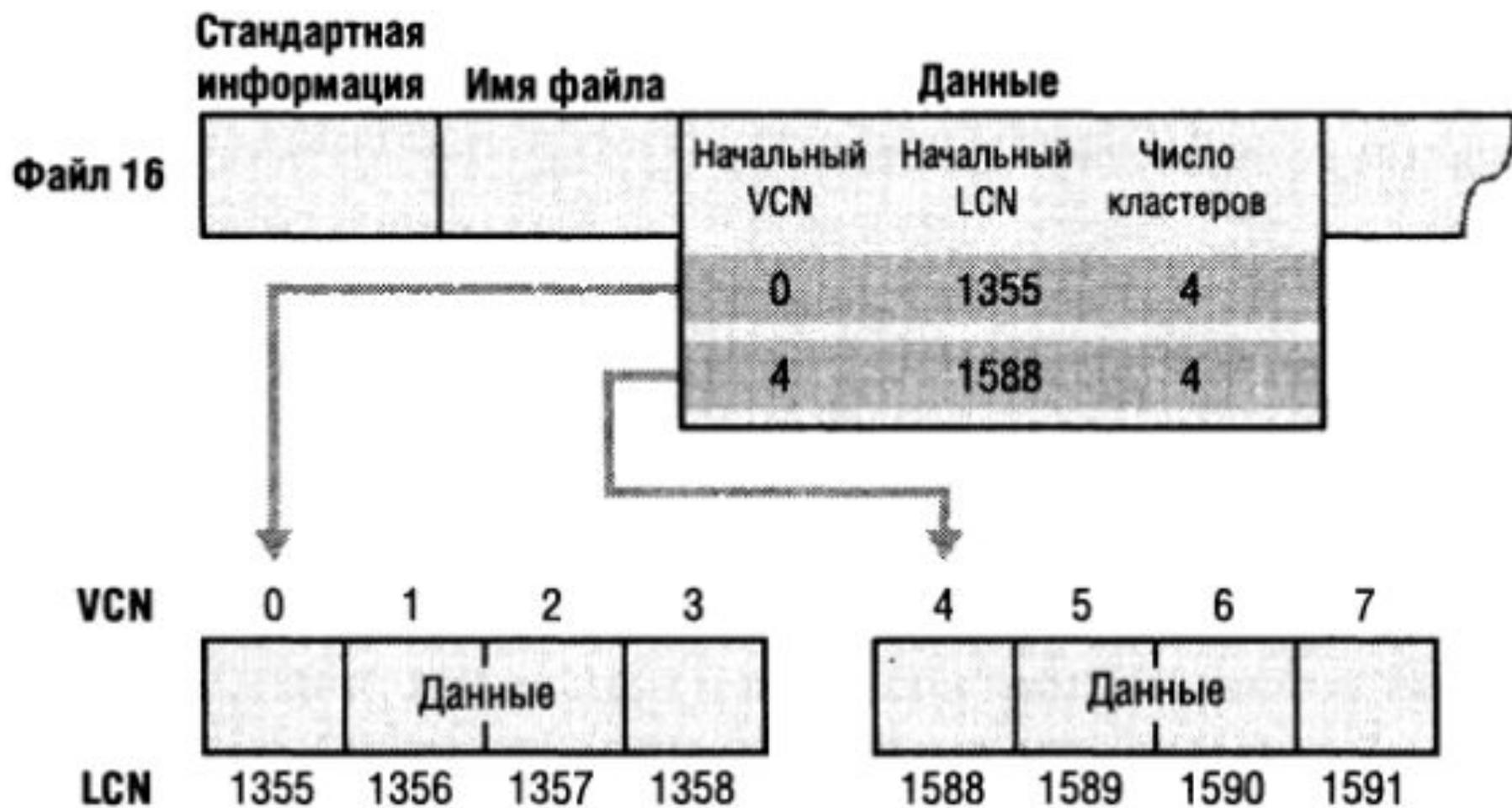
Файл состоит из последовательности кластеров, порядковый номер кластера внутри файла называется виртуальным номером кластера (***Virtual Cluster Number, VCN***). Размер кластера от 512 байт до 64 Кбайт.

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

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

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





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

1-й отрезок MFT

Файлы метаданных

0. Описание MFT, в том числе адреса всех ее отрезков.
1. Зеркальная копия 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.

2-й отрезок MFT

3-й отрезок MFT

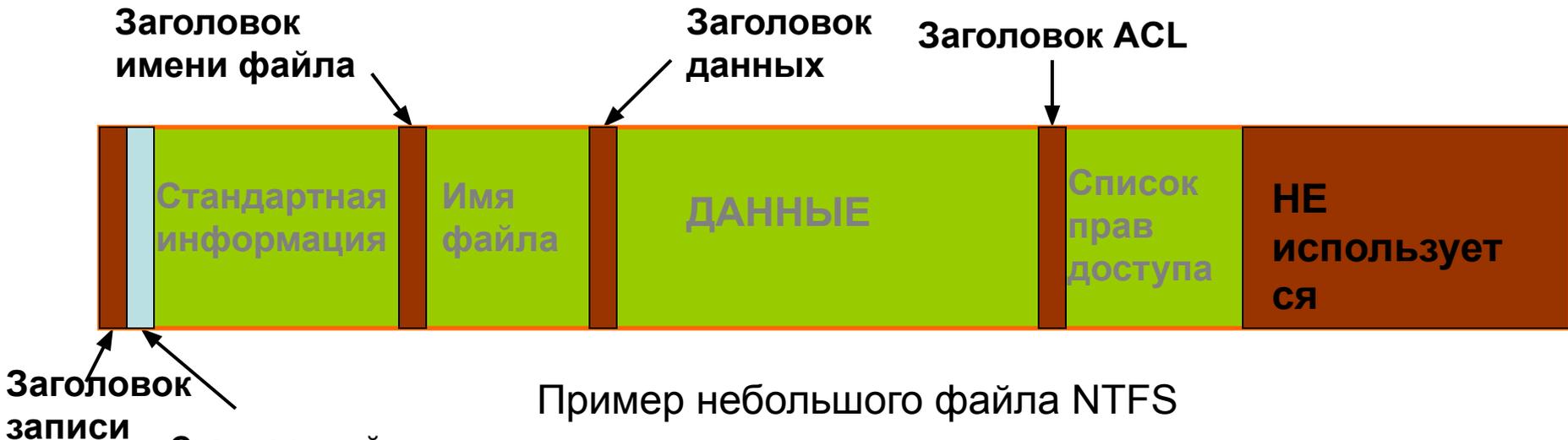
Структура файлов NTFS

Файлы и каталоги состоят из набора атрибутов. Каждая запись MFT состоит из заголовка, за которым следуют атрибуты. Атрибуты содержат следующие поля: тип, длина, имя (образуют заголовок) и значение.

Атрибуты, используемые в записях MFT:

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 (мониторинг и симв. ссылки)

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



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

Небольшие файлы (small).

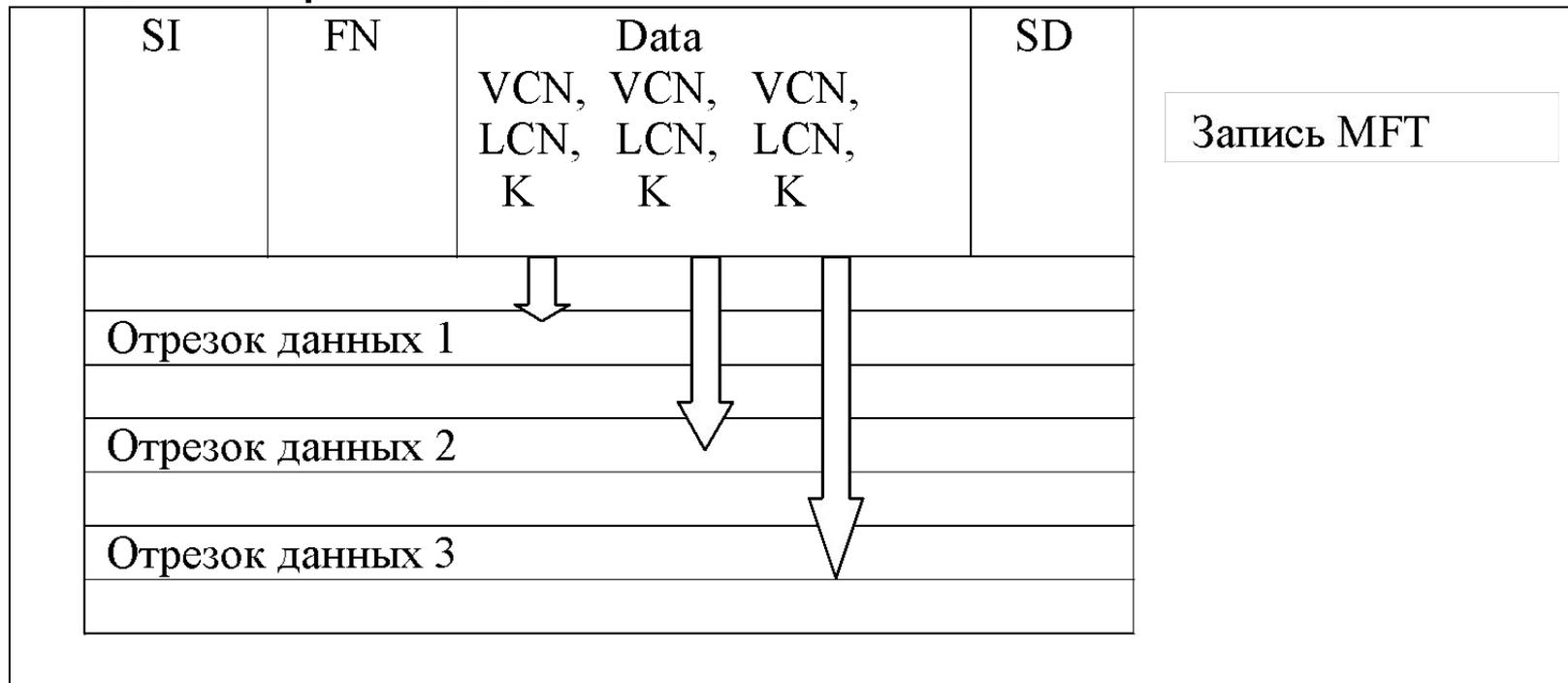
Если файл имеет небольшой размер, то он может целиком располагаться внутри одной записи MFT, имеющей, например, размер 4 Кбайт. Небольшие файлы NTFS состоят по крайней мере из следующих атрибутов:

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

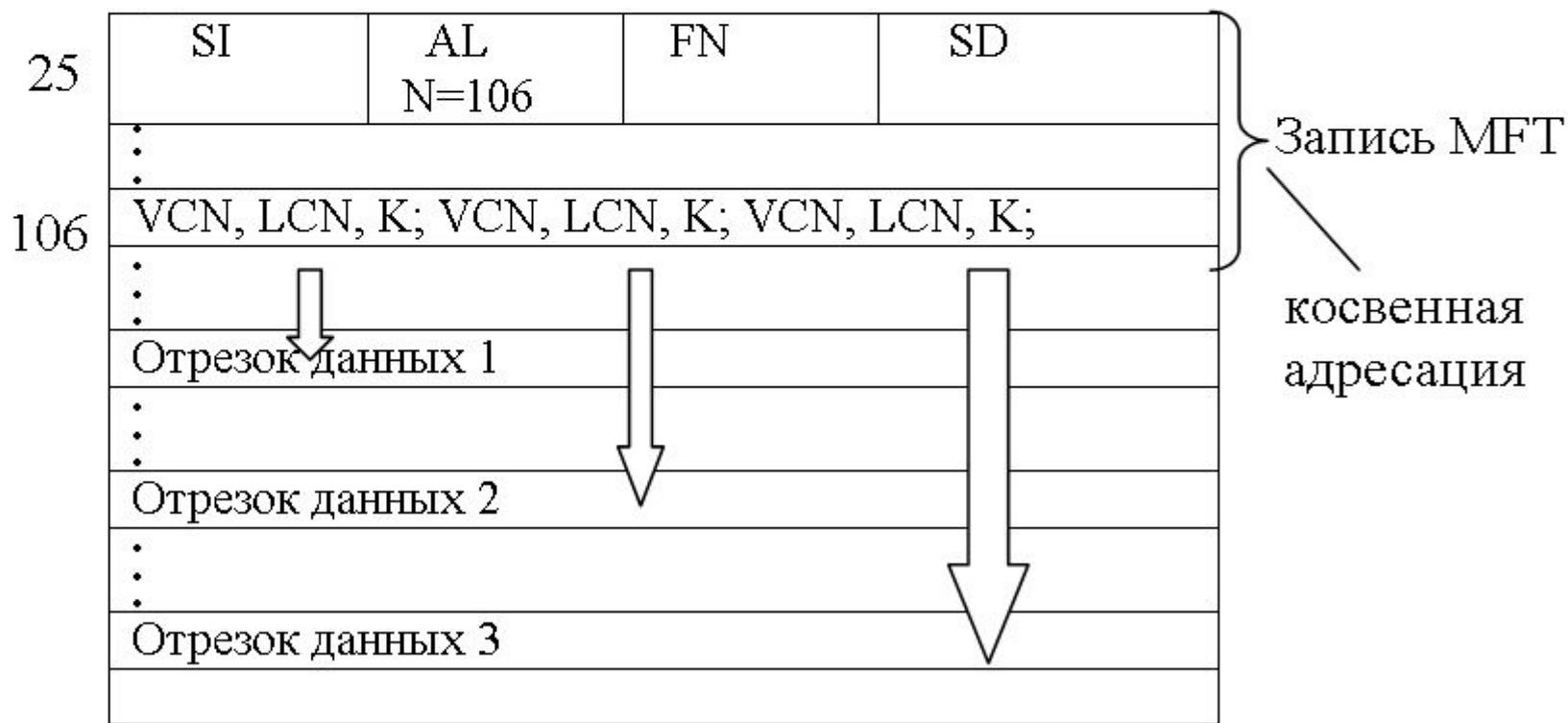


Большие файлы (large).

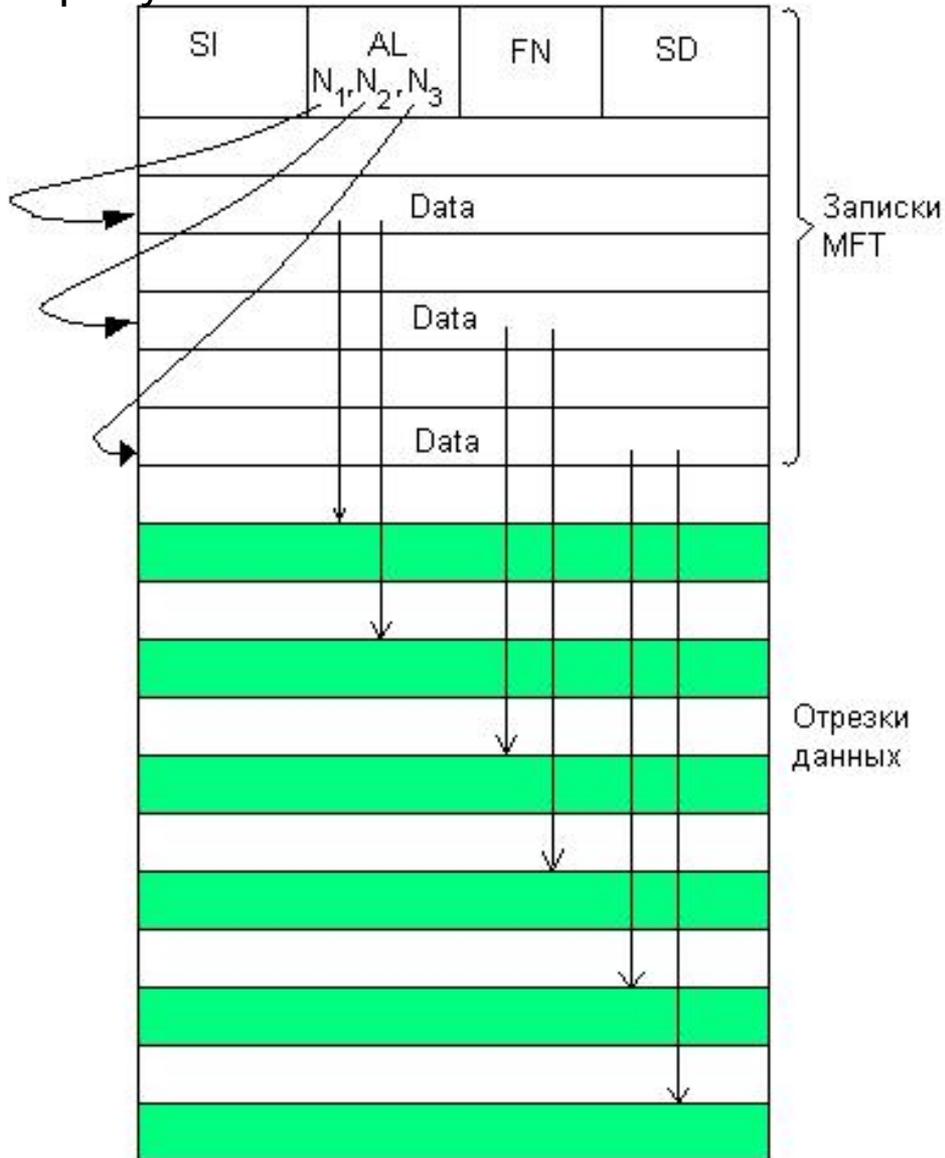
Если данные файла не помещаются в одну запись MFT, то этот факт отражается в заголовке атрибута *Data*, который содержит признак того, что этот атрибут является нерезидентным, то есть находится в отрезках вне таблицы MFT. В этом случае атрибут *Data* содержит адресную информацию (VCN, LCN, k) каждого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

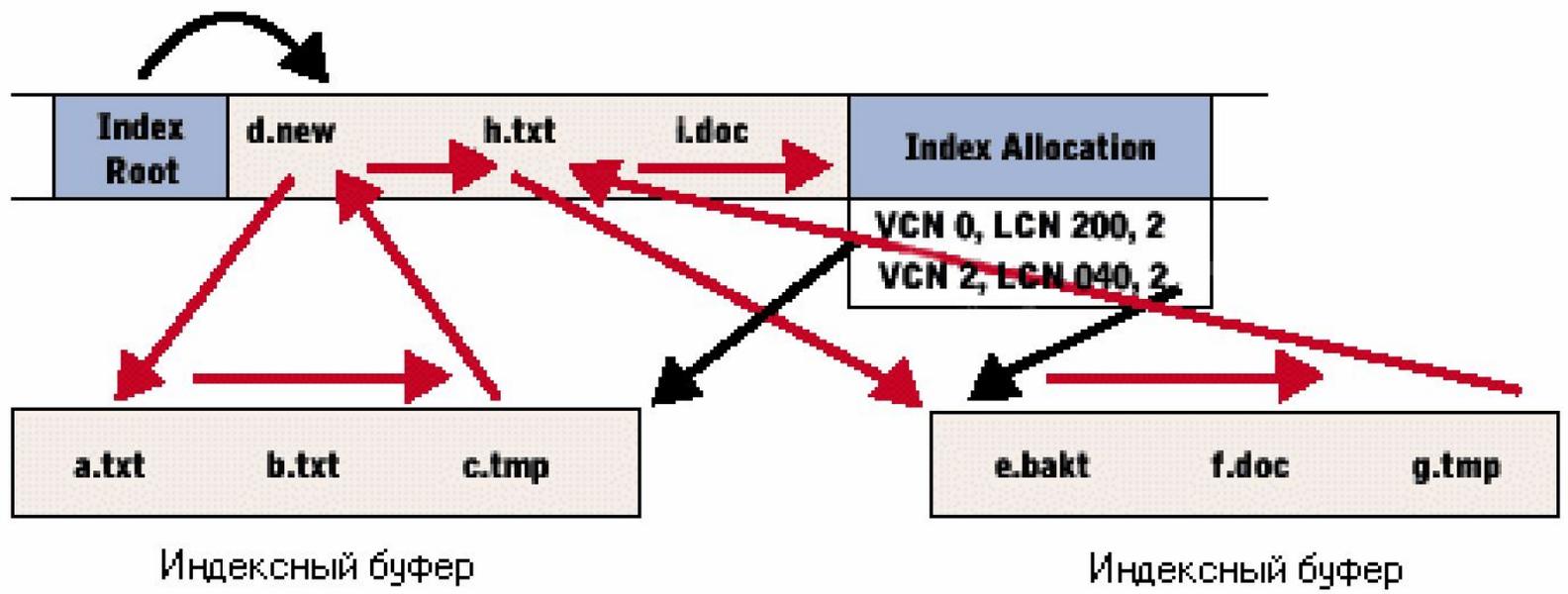
- Если количество файлов в каталоге невелико, то список файлов может быть резидентным в записи в MFT, являющейся каталогом
- Для резидентного хранения списка используется единственный атрибут – *Index Root*. Список файлов содержит значения атрибутов файла. По умолчанию – это имя файла, а также номер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

SI	FN	IR	SD
		<a.bat, 27> <c.sys, 92> <zyx, N > <####>	

– признак конца файлов

Большие каталоги (large indexes)

- По мере того как каталог растёт, 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 $\langle \text{fl.exe}, N_{\text{fl.exe}} \rangle$ $\langle \text{ltr.exe}, N_{\text{ltr.exe}} \rangle$ $\langle \text{#####} \rangle$	IA $\text{VCN}_1, \text{LCN}_1, k_1$ $\text{VCN}_2, \text{LCN}_2, k_2$ $\text{VCN}_3, \text{LCN}_3, k_3$	SD
IR $\langle \text{avia.doc}, N_{\text{avia.doc}} \rangle$ $\langle \text{az.exe}, N_{\text{az.exe}} \rangle$ $\langle \text{emax.exe}, N_{\text{emax.exe}} \rangle \langle \text{#####} \rangle$				
IR $\langle \text{gl.htm}, N_{\text{gl.htm}} \rangle$ $\langle \text{green.com}, N_{\text{green.com}} \rangle$ $\langle \text{caw.doc}, N_{\text{caw.doc}} \rangle \langle \text{#####} \rangle$				
IR $\langle \text{main1.c}, N_{\text{main1.c}} \rangle$ \vdots $\langle \text{zero.txt}, N_{\text{zero.txt}} \rangle \langle \text{#####} \rangle$				

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.

Сравнение файловых систем NTFS и ReFS

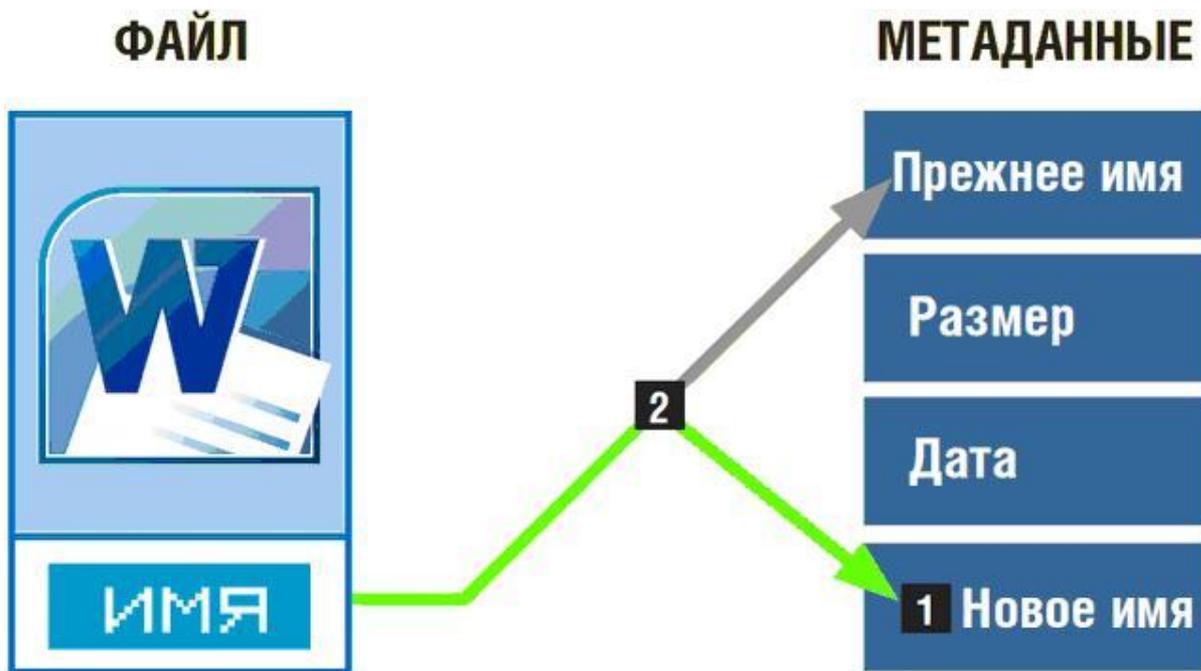
	NTFS	ReFS
МАКСИМАЛЬНЫЙ РАЗМЕР ФАЙЛА	16 Тбайт	18,3 Эбайт
МАКСИМАЛЬНЫЙ РАЗМЕР ТОМА	18,4 Эбайт	402 Эбайт
МАКСИМАЛЬНОЕ ЧИСЛО ФАЙЛОВ В ПАПКЕ (ПРИБЛИЗИТЕЛЬНО)	4,3 млрд	18 трлн
МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СИМВОЛОВ В ИМЕНИ ФАЙЛА	255	32 767
МАКСИМАЛЬНОЕ КОЛИЧЕСТВО СИМВОЛОВ В ИМЕНИ ПУТИ	255	32 767

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.



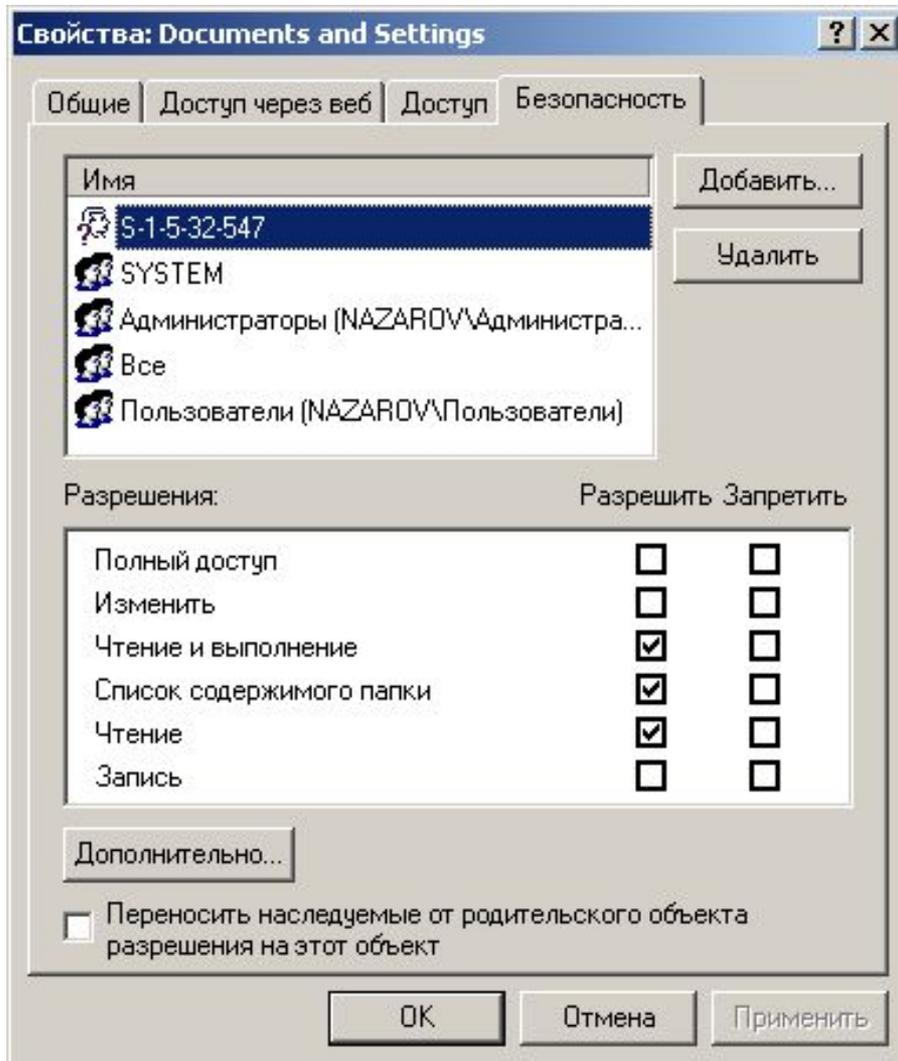
Матрица прав доступа

Имена файлов

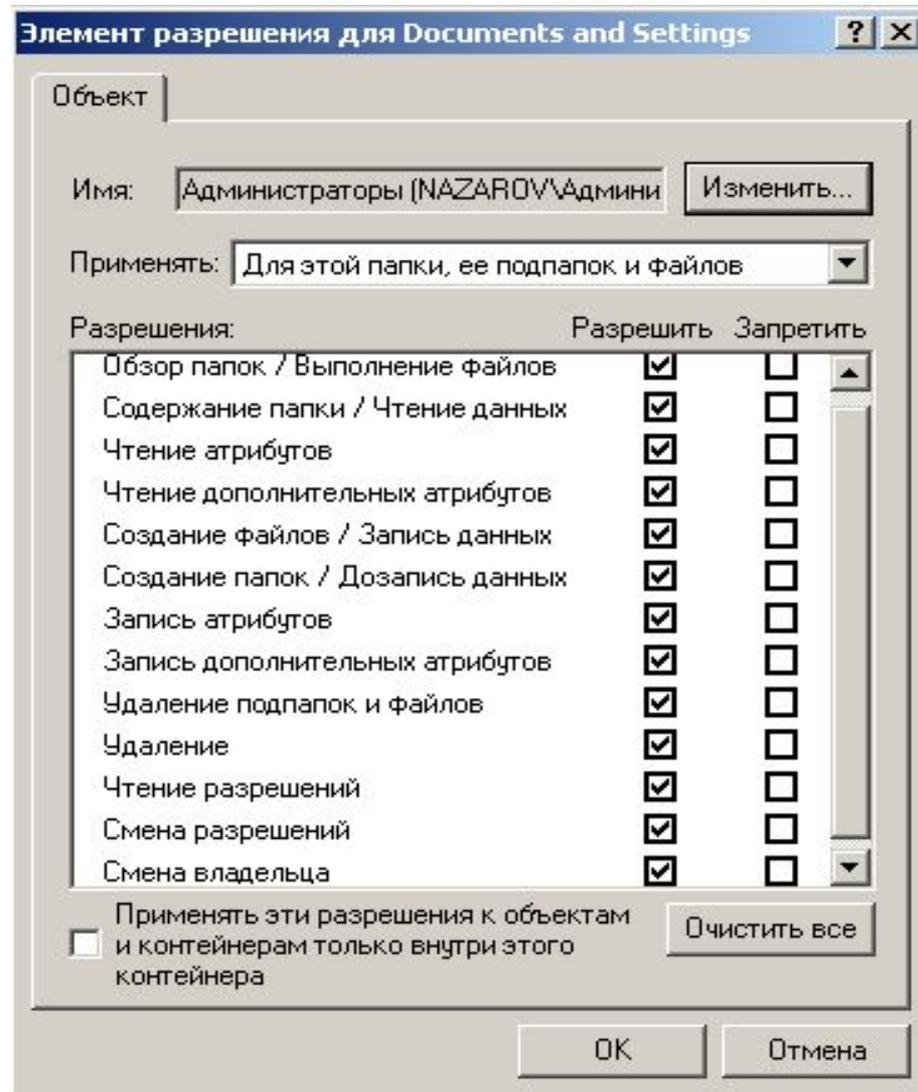
Имена
пользователей

	modern.txt	win.exe	class.dbf	unix.ppt	
kira	читать	выполнять	—	выполнять	
genua	читать	выполнять	—	выполнять читать	
nataly	читать	—	—	выполнять читать	
victor	читать писать	—	создать	—	

Разрешения на доступ к каталогам

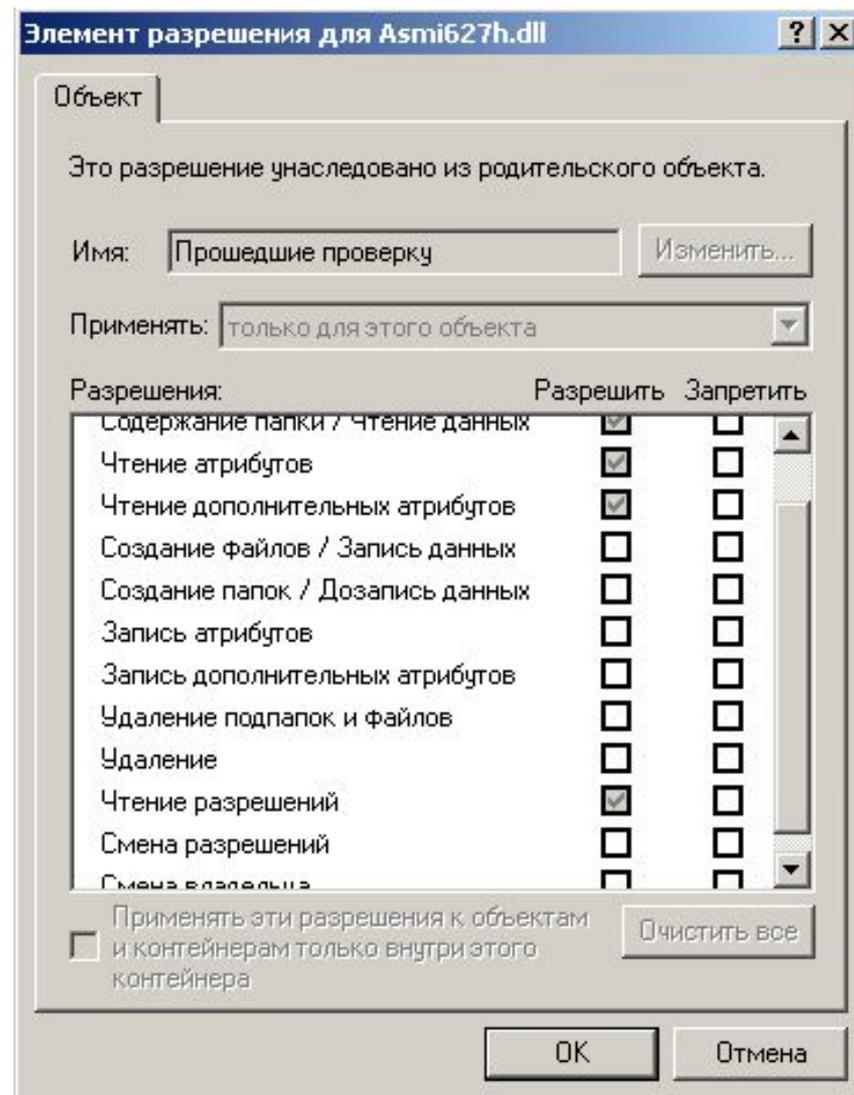
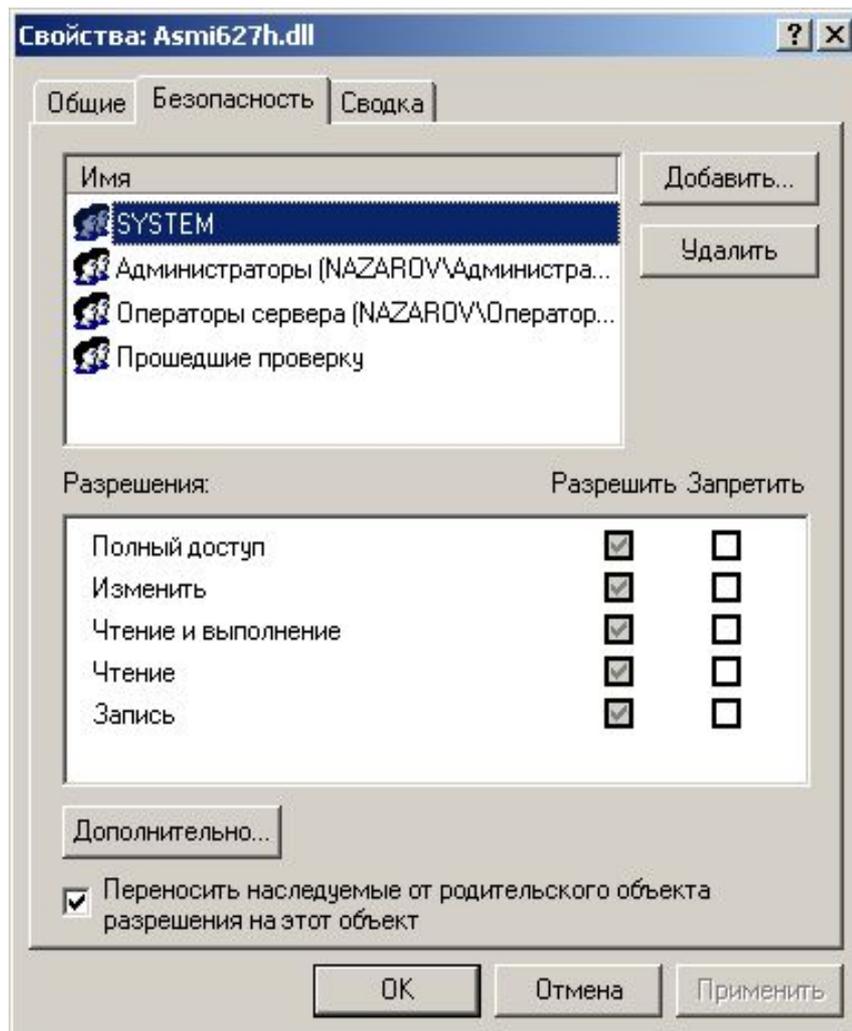


Стандартные разрешения

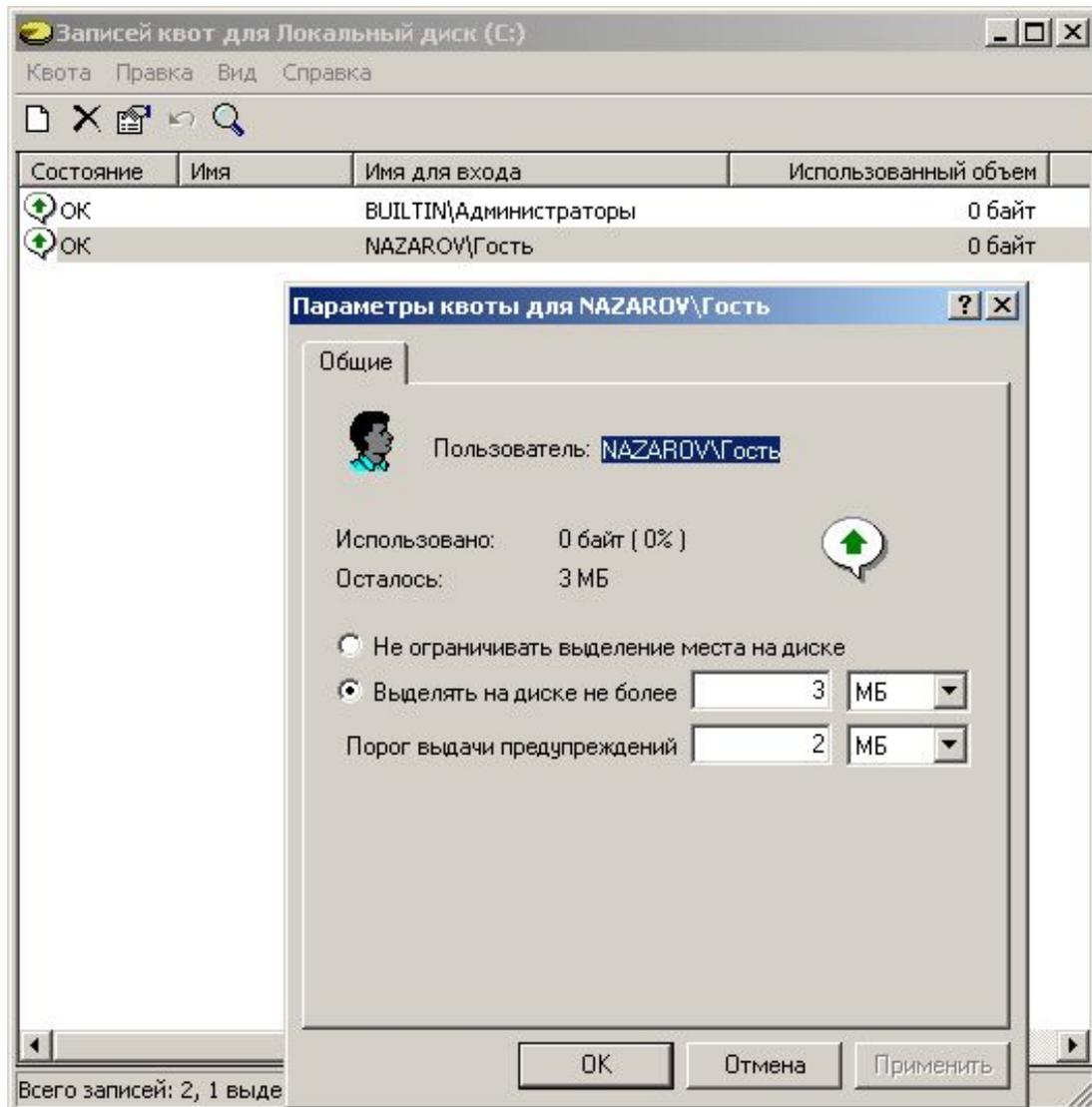
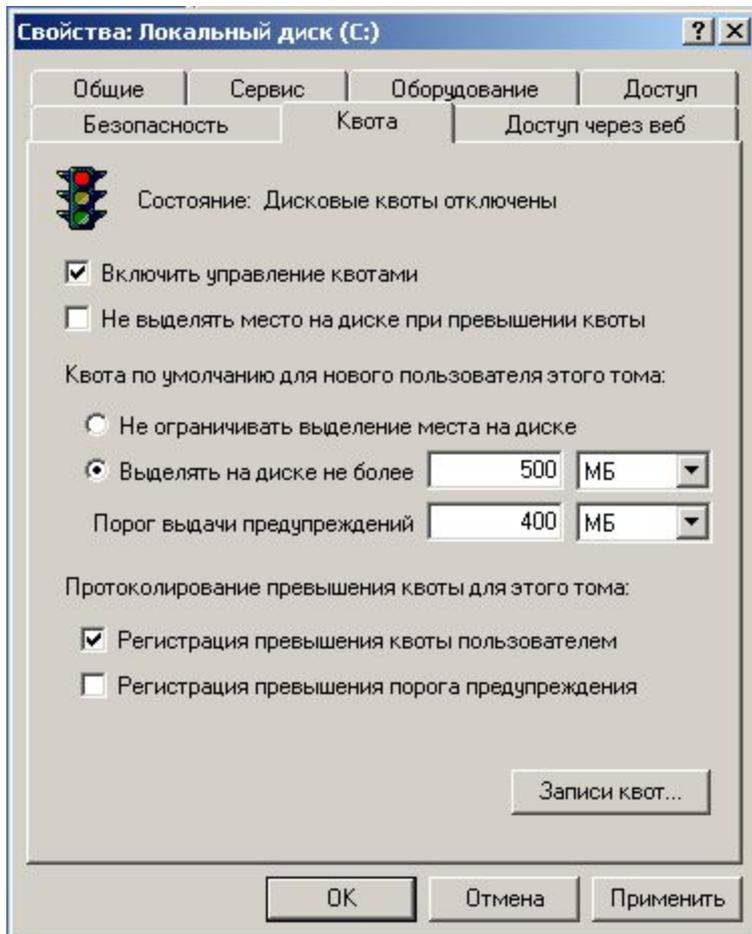


Специальные разрешения

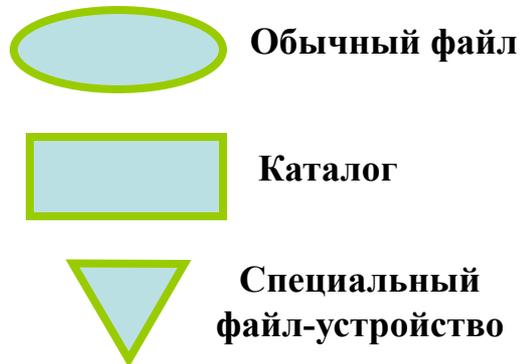
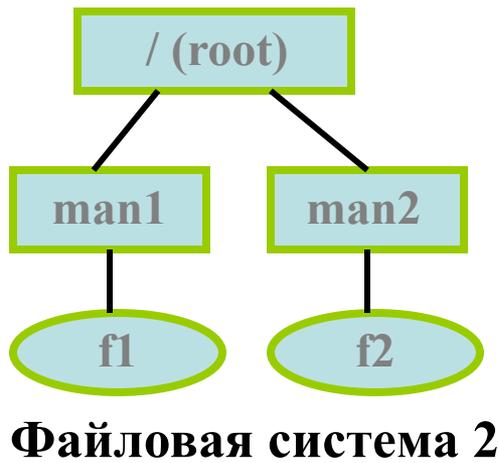
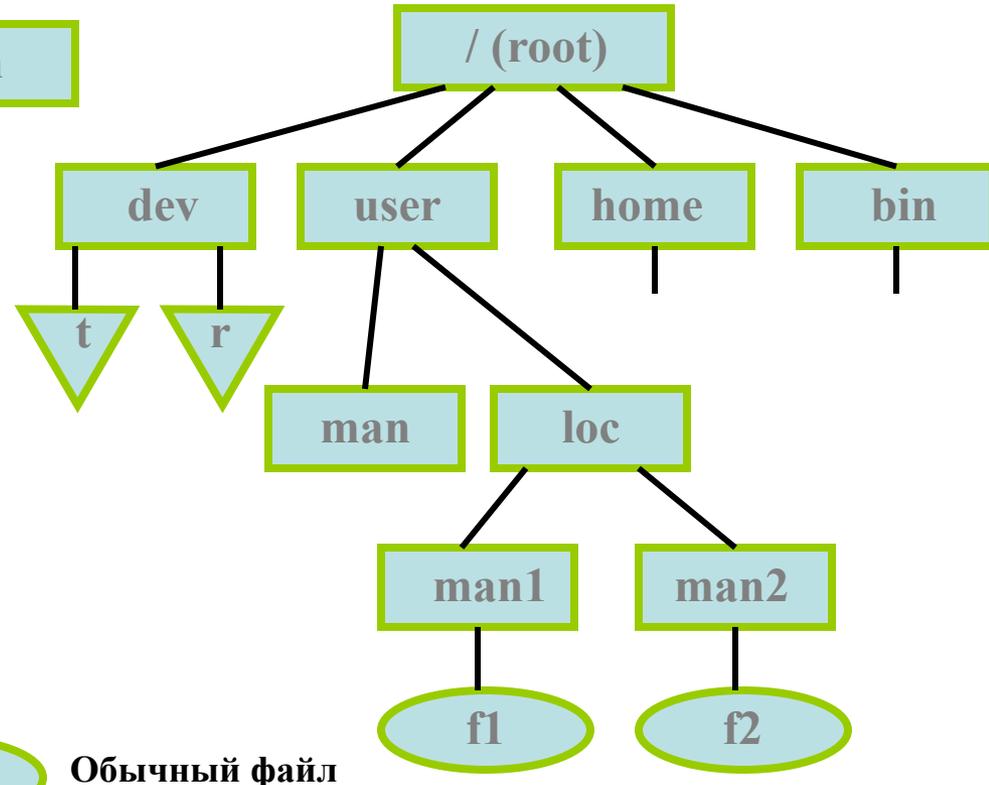
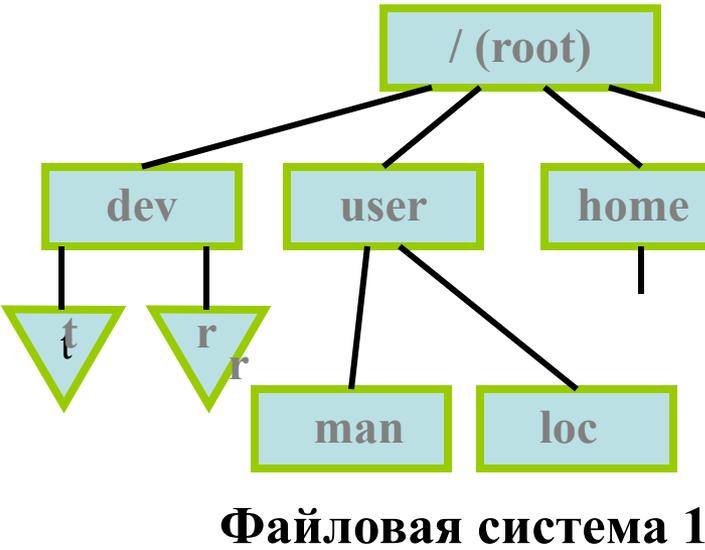
Разрешения на доступ к файлам



Квоты дискового пространства



Монтирование



Общая файловая система после монтирования

Типы файлов

- (дефис) — обычный файл;
- d — каталог;
- c — символьное устройство;
- b — блочное устройство;
- p — именованный канал (named pipe);
- s — сокет (socket);
- | — символическая ссылка.

```
dsl@box:~$ cd .
dsl@box:~$ cd ..
dsl@box:~/home$ cd ..
dsl@box:/$ ls -l
```

drwxrwxr-x	19	root	root	4096	Jul	1	2007	KNOPPIX
lrwxrwxrwx	1	root	root	12	May	17	2004	bin -> /KNOPPIX/bin
lrwxrwxrwx	1	root	root	13	May	17	2004	boot -> /KNOPPIX/boot
dr-xr-xr-x	5	root	root	2048	Jul	1	2007	cdrom
drwxr-xr-x	3	root	root	120	Jan	5	00:04	dev
drwxr-xr-x	52	root	root	280	Jan	5	00:02	etc
lrwxrwxrwx	1	root	root	13	Jan	4	19:01	home -> /rawdisk/home
lrwxrwxrwx	1	root	root	12	May	17	2004	lib -> /KNOPPIX/lib
drwx-----	2	root	root	1024	Mar	24	2006	lost+found
drwxr-xr-x	7	root	root	1024	Jan	5	00:02	wnt
lrwxrwxrwx	1	root	root	12	Jan	5	00:02	opt -> /rawdisk/opt
dr-xr-xr-x	44	root	root	0	Jan	4	19:01	proc
drwxrwxrwt	12	root	root	240	Jan	5	00:02	rawdisk
drwxr-xr-x	2	root	root	1024	Jan	5	00:02	root
lrwxrwxrwx	1	root	root	13	Jan	4	19:01	sbin -> /KNOPPIX/sbin
drwxr-xr-x	2	root	root	1024	Jan	12	2004	sys
lrwxrwxrwx	1	root	root	12	Jan	5	00:02	tmp -> /rawdisk/tmp
lrwxrwxrwx	1	root	root	12	May	17	2004	usr -> /KNOPPIX/usr
lrwxrwxrwx	1	root	root	12	Jan	4	19:01	var -> /rawdisk/var

```
dsl@box:/$
```

Bash

Физическая организация S5 и ifc



Расположение файловой системы s5 на диске

Структура *индексного дескриптора* (i-node)

- идентификатор владельцев файла;
- тип файла, файл может быть файлом обычного типа, каталогом, специальным файлом, конвейером и символьной связью;
- права доступа к файлу;
- временные характеристики: время последней модификации файла, время последнего обращения к файлу, время последней модификации индексного дескриптора;
- число ссылок на данный индексный дескриптор равно количеству псевдонимов файла;
- адресная информация ;
- размер файла в байтах.

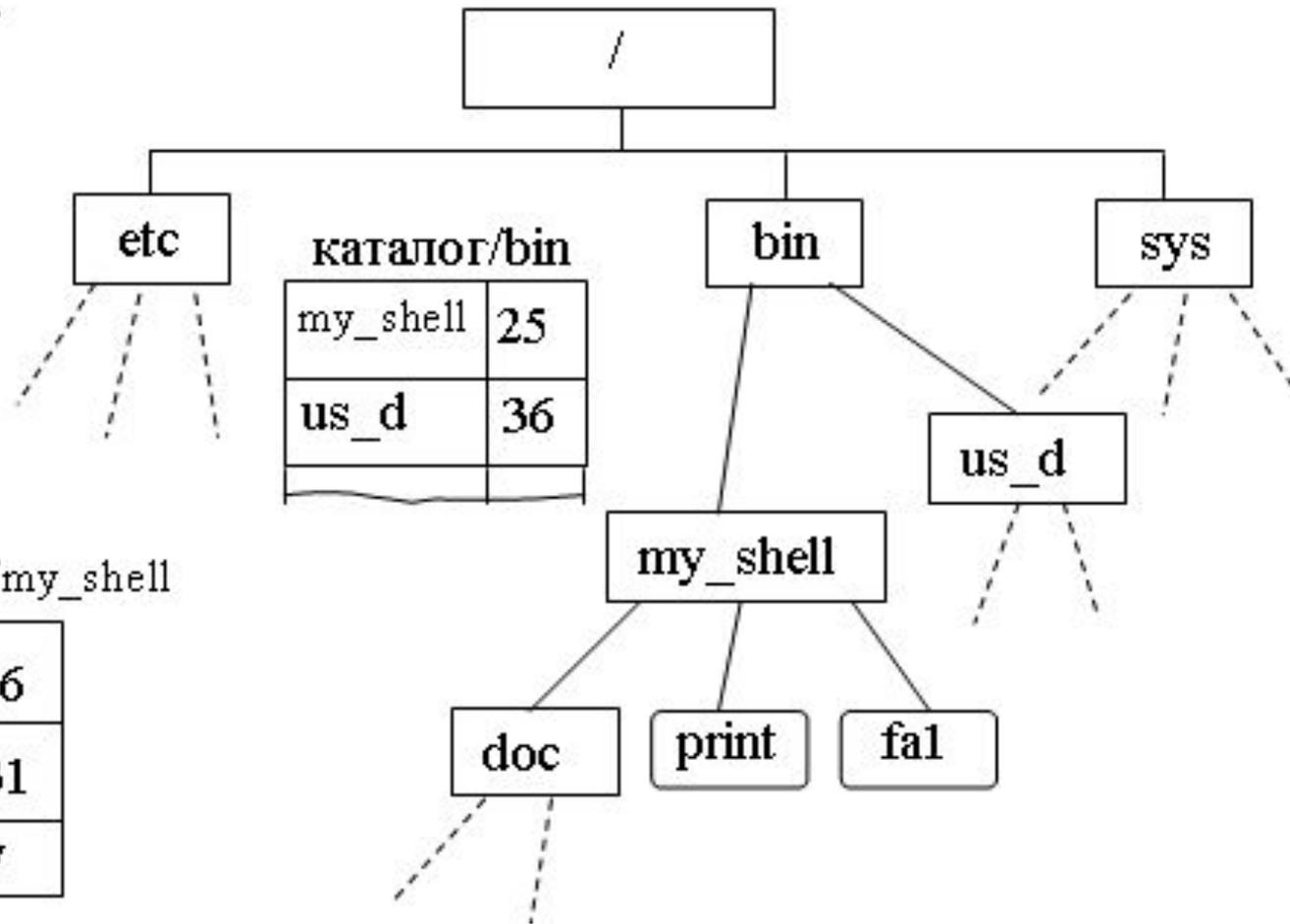
копирование индексного дескриптора входит в
процедуру открытия файла. При открытии файла
ядро выполняет следующие действия:

- Проверяет, существует ли файл; если не существует, то можно ли его создать. Если существует, то разрешен ли к нему доступ требуемого вида.
- Копирует индексный дескриптор с диска в оперативную память; если с указанным файлом уже ведется работа, то новая копия индексного дескриптора не создается.
- Создает в области ядра структуру, предназначенную для отображения текущего состояния операции обмена данными с указанным файлом. Эта структура, называемая *file*, содержит данные о типе операции (чтение, запись или чтение и запись), о числе считанных или записанных байтов, указатель на байт файла, с которым проводится операция.
- Делает отметку в контексте процесса, выдавшего системный вызов на операцию с данным файлом.

Поиск файла */bin/my_shell/print*

Корневой каталог/

bin	6
sys	3
etc	11



Каталог/bin/my_shell

doc	126
print	131
fal	17

1. просматривается корневой каталог с целью поиска первого составляющего символического имени – это *bin*. Определяется номер индексного дескриптора каталога – это 6, адрес корневого каталога системе известен;
2. из области индексных дескрипторов считывается дескриптор №6, начальный адрес дескриптора определяется на основании известных системе номера начального сектора номера индексного дескриптора и размера индексного дескриптора. Из индексного дескриптора 6 определяется физический адрес каталога *bin*.
3. просматривается каталог *bin*, целью поиска имени *my_shell* и определяется его номер – это 25;
4. считывается индексный дескриптор 25, определяется физический адрес каталога */bin/my_shell/print*;
5. просматривая каталог */bin/my_shell/print*, определяется номер индексного дескриптора файла *print* – это 131;
6. из индексного дескриптора 131 определяются номера блоков данных и другие характеристики искомого файла.

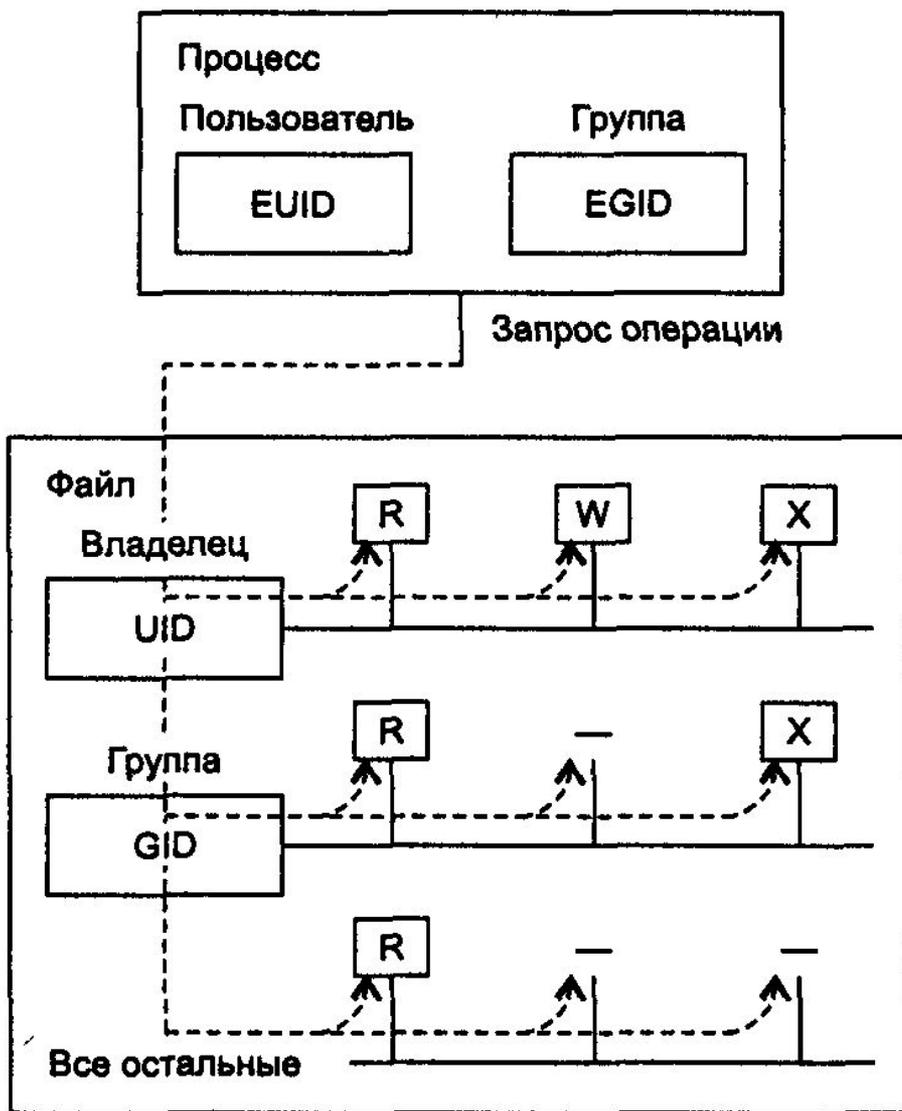
Загрузочный блок
Суперблок
Блок группы цилиндров
Список i-node
Блоки данных
Суперблок
Блок группы цилиндров
Список i-node
·
·
·

Физическая
организация
файловой системы
ufs

Реализация прав доступа в UNIX

- С каждым процессом UNIX связаны два идентификатора: пользователя, от имени которого был создан этот процесс, и группы, к которой принадлежит данный пользователь. Эти идентификаторы носят название *реальных идентификаторов пользователя*: Real User ID, RUID и *реальных идентификаторов группы*: Real Group ID, RGID.
- При проверке прав доступа к файлу используются так называемые *эффективные идентификаторы пользователя*: Effective User ID, EUID и *эффективные идентификаторы группы*: Effective Group ID, EGID.
- Файл имеет два признака разрешения смены идентификатора — Set User ID on execution (SUID) и Set Group ID on execution (SGID), которые разрешают смену идентификаторов пользователя и группы при выполнении данного файла.

Проверка прав доступа в UNIX



Смена эффективных идентификаторов

ПРОЦЕСС

