Учебный курс

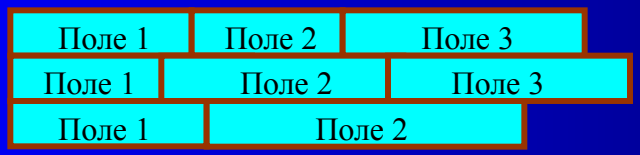
Операционные среды, системы и оболочки

4.4.3.2. Логическая организация файлов

Модель 1. Неструктурированная последовательность байт (ОС UNIX).

Модель 2. Структурированный файл : смешанный, последовательный, индексно-последовательный, индексированный, прямого доступа.

Смешанный файл



Каждое поле описывает само себя (имя, длина, значение). Доступ – полный перебор.

Достоинства: рациональное использование дискового пространства, хорошо подходят для полного перебора

Недостатки: сложность вставки и обновления записей

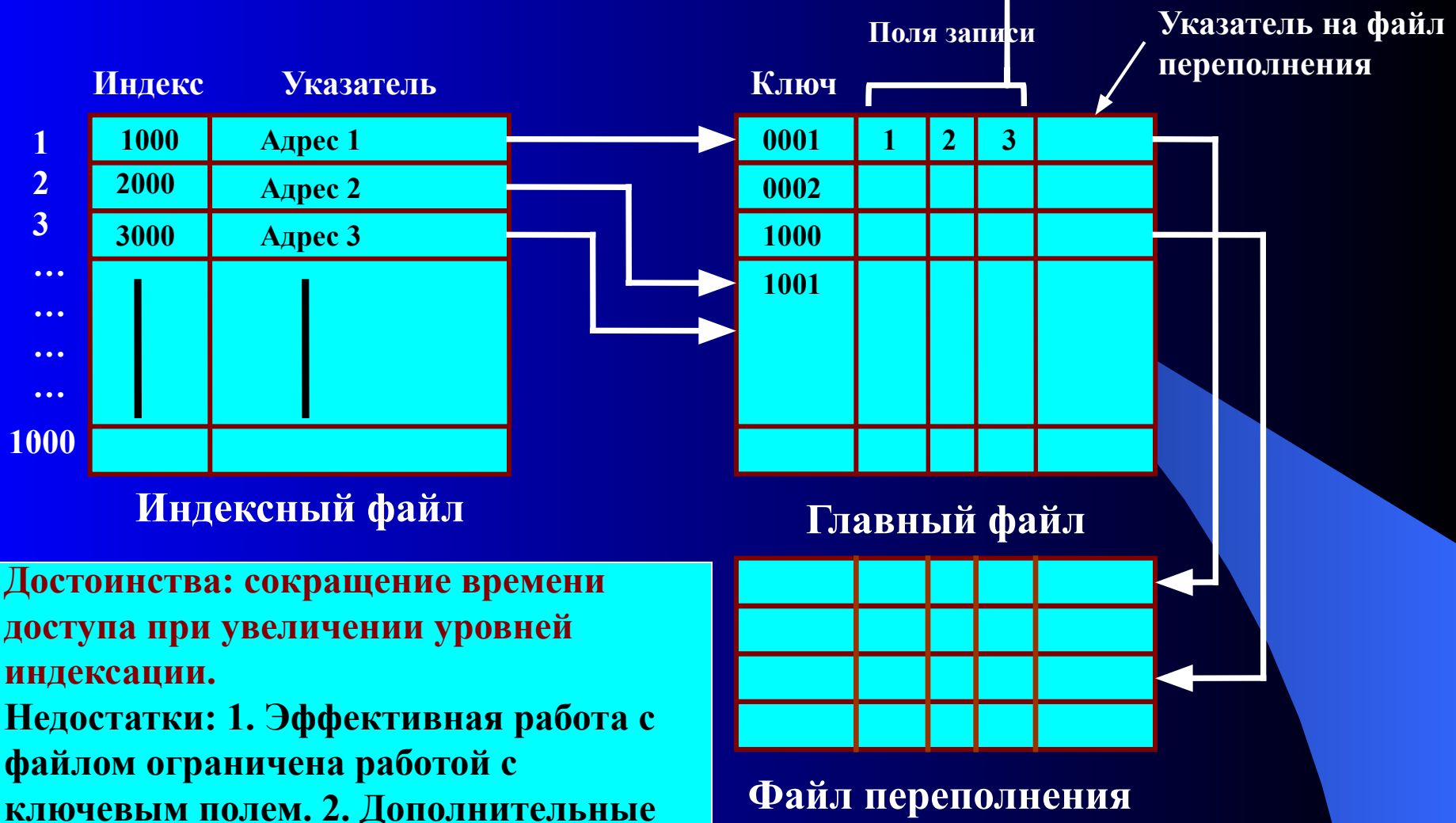
Последовательный файл

Записи имеют одну длину, одни и те же поля и хранят только значения полей (одно поле – ключевое). Атрибуты файловой структуры: имя и длина каждого поля.

Достоинства: оптимальный вариант для пакетных приложений, записи хранятся в ключевой последовательности, возможно хранение на диске и МЛ. Возможна организация в виде списка, что упрощает вставку новых записей.

Недостатки: малоэффективен для диалоговых приложений

Индексно-последовательный файл



Достоинства: сокращение времени доступа при увеличении уровней индексации.

Недостатки: 1. Эффективная работа с файлом ограничена работой с ключевым полем. 2. Дополнительные затраты времени на периодическое слияние с файлом переполнения.

Индексированный файл

Типы индексов:

1. Полный индекс – содержит по одному элементу для каждой записи главного файла.

2. Частный индекс содержит элементы для записей, в которых имеется интересующее пользователя поле.

3. При добавлении новой записи в главный файл необходимо обновлять все индексные файлы.

4. Индексы организуются в виде последовательных файлов.

Достоинство: быстрый доступ. **Недостатки**: большая избыточность данных, неэффективность обработки всех записей файла.

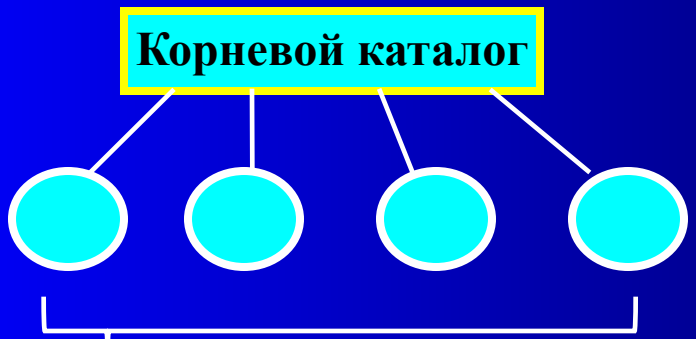
Файл прямого доступа

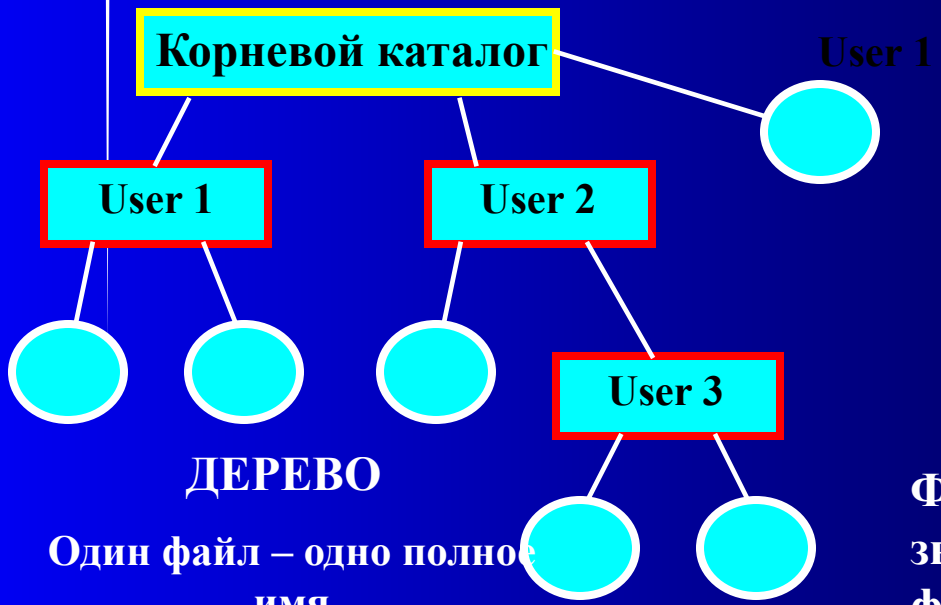
1. Обеспечивает прямой доступ к любой записи фиксированной длины по известному адресу (ключу) при хранении файлов на диске.

2. Достоинства: быстрый доступ к любой записи, простота вставки, удаления и модификации записей.

3. Недостатки: записи фиксированной структуры и длины.

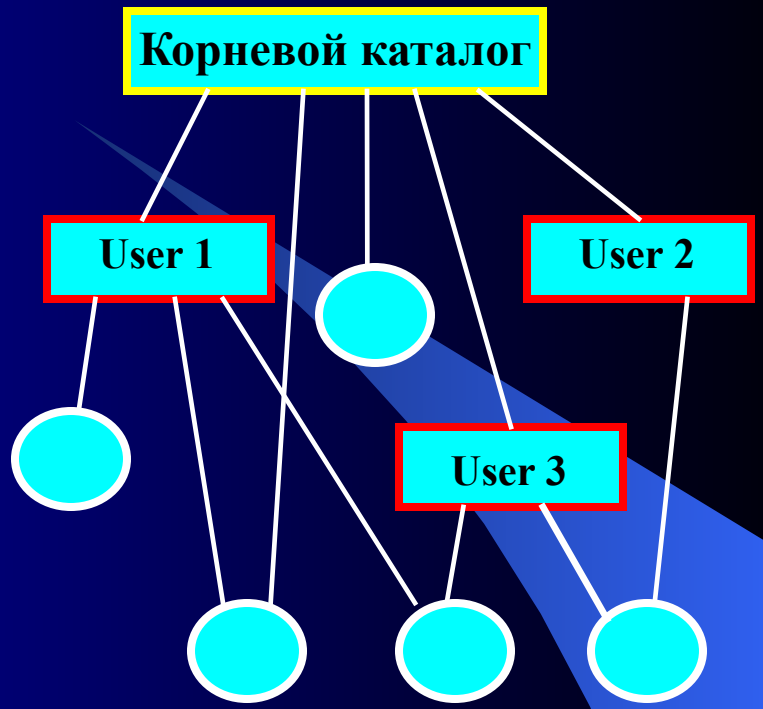
4.4.4. Каталоговые системы





ДЕРЕВО

Один файл – одно полное имя

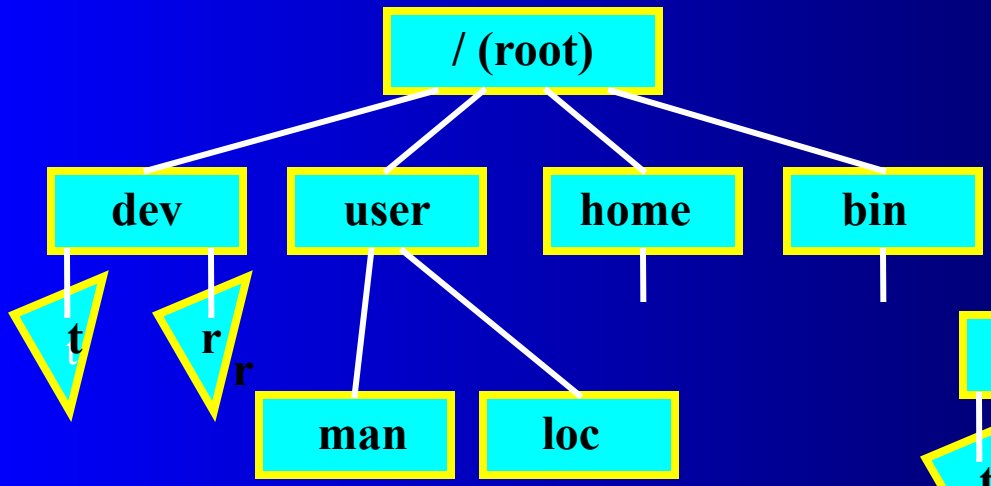


СЕТЬ

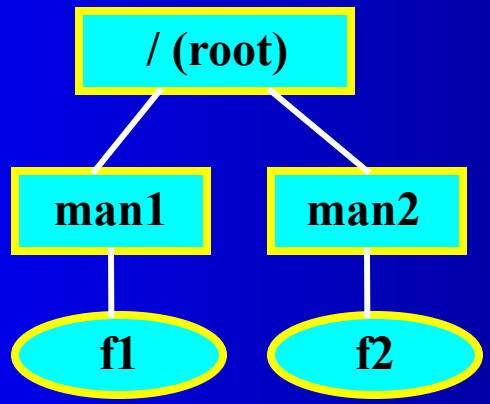
Один файл – много полных имен

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

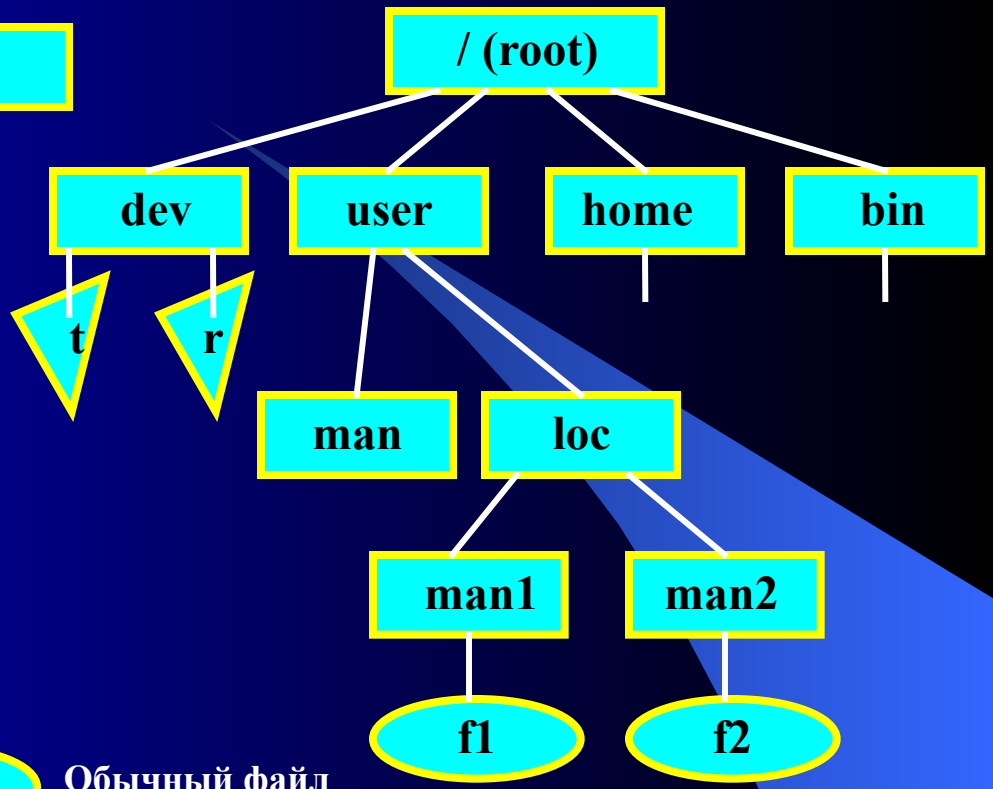
Монтирование



Файловая система 1



Файловая система 2

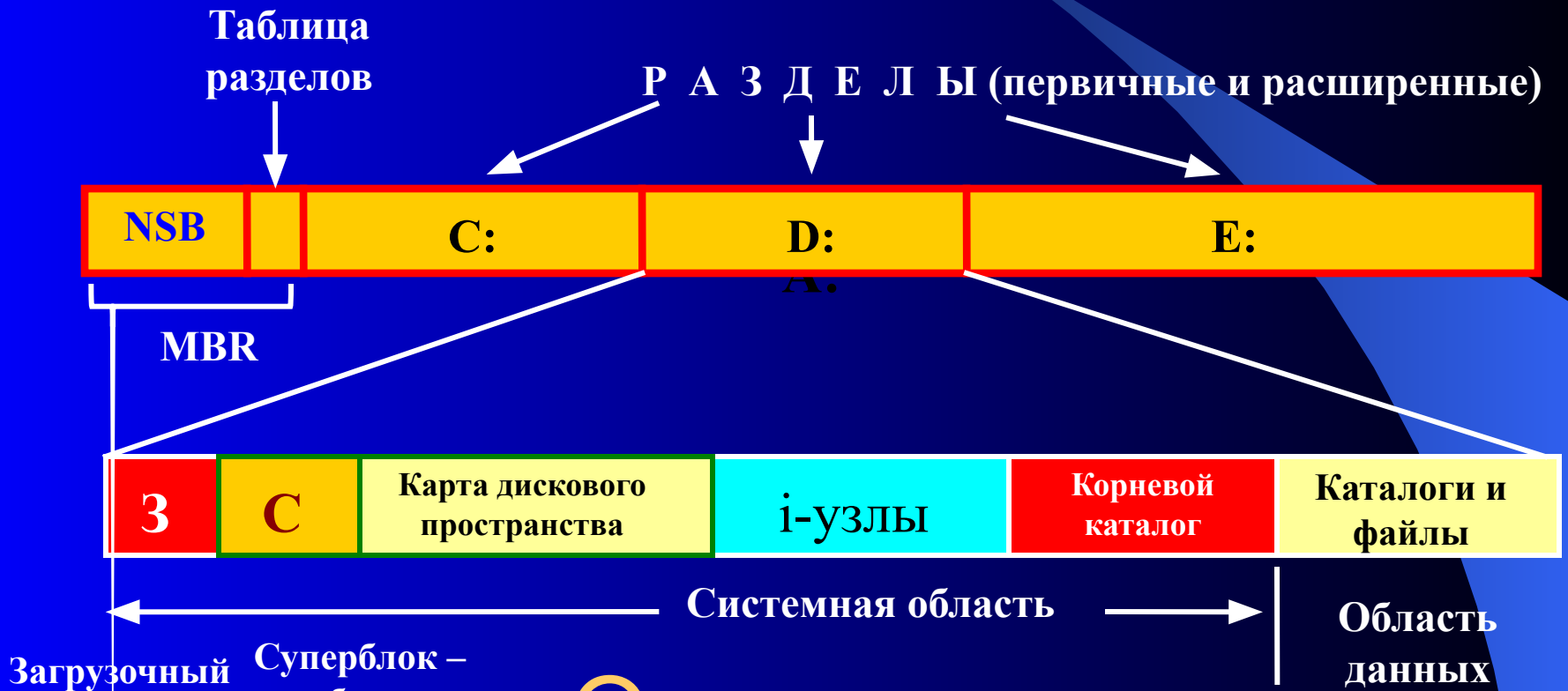


Общая файловая система
после монтирования

4.4.3. 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.



NSB (Non –System Bootstrap) – внесистемный загрузчик

Адресация блоков данных диска

1 способ: c – h - s

c – номер цилиндра,

h – номер головки,

s – номер сектора

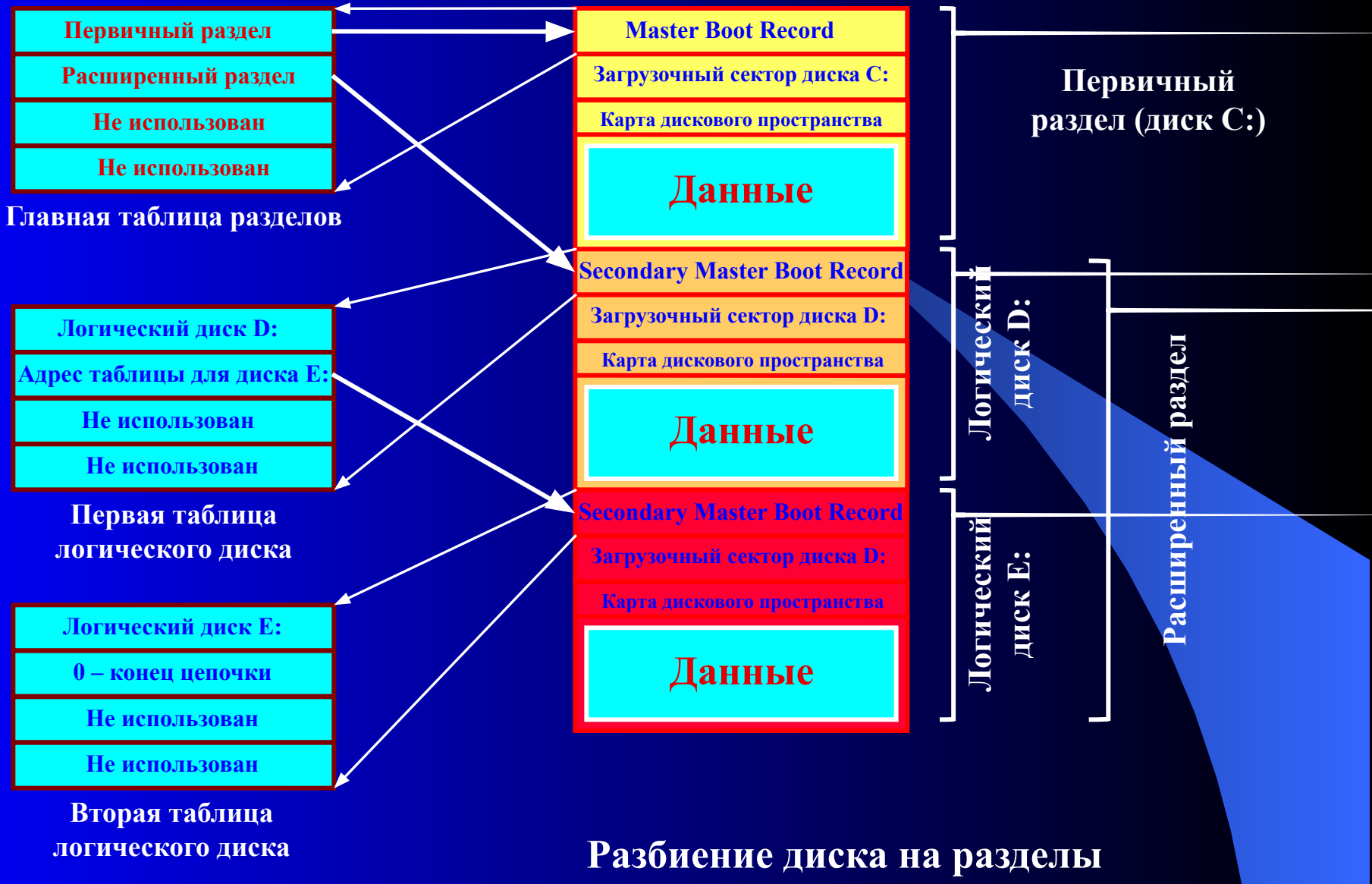
2 способ: LBA

= (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



Разбиение диска на разделы

Физическая организация и адресация файла

Критерии эффективности физической организации файла:

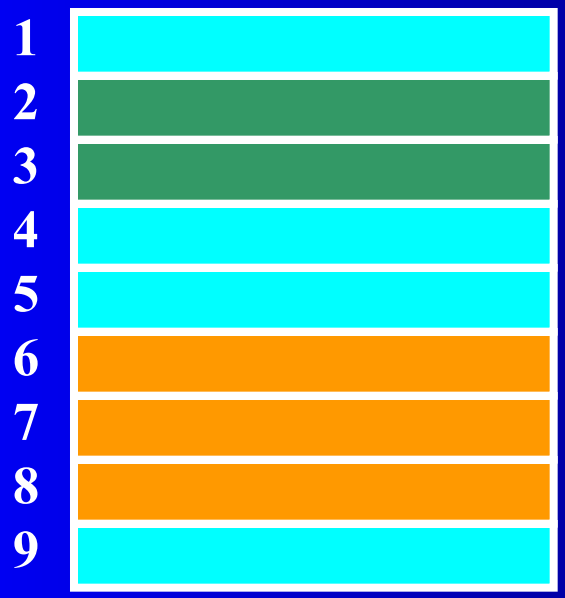
- ✔ скорость доступа к данным;
- ✔ объем адресной информации файла;
- ✔ степень фрагментированности дискового пространства;
- ✔ максимально возможный размер файла.

Возможные схемы размещения файлов:

- ❑ - непрерывное размещение (непрерывные файлы);
- ❑ - связный список блоков (кластеров) файла;
- ❑ - связный список индексов блоков (кластеров) файла;
- ❑ - перечень номеров блоков (кластеров) файлов;
- ❑ - структуры, называемые I-узлами (index-node – индекс-узел).

Операционные

Непрерывное размещение

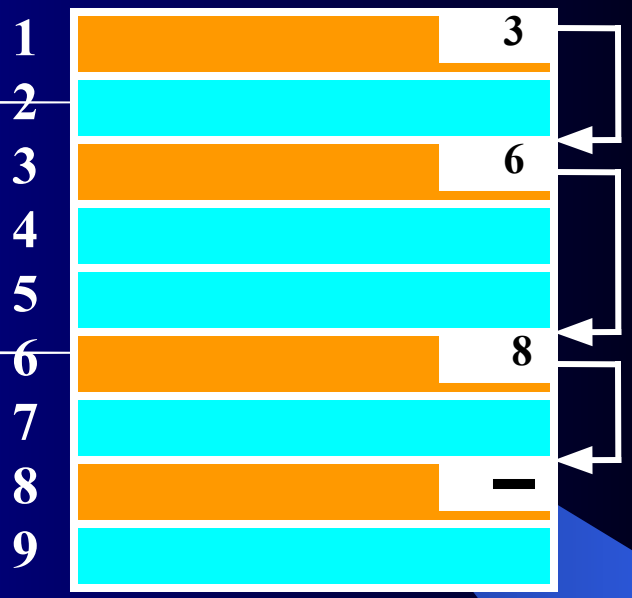


Достоинства: высокая скорость доступа, минимальный объем адресной информации, нет ограничений на размер файла.

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

Область применения – компакт-дики

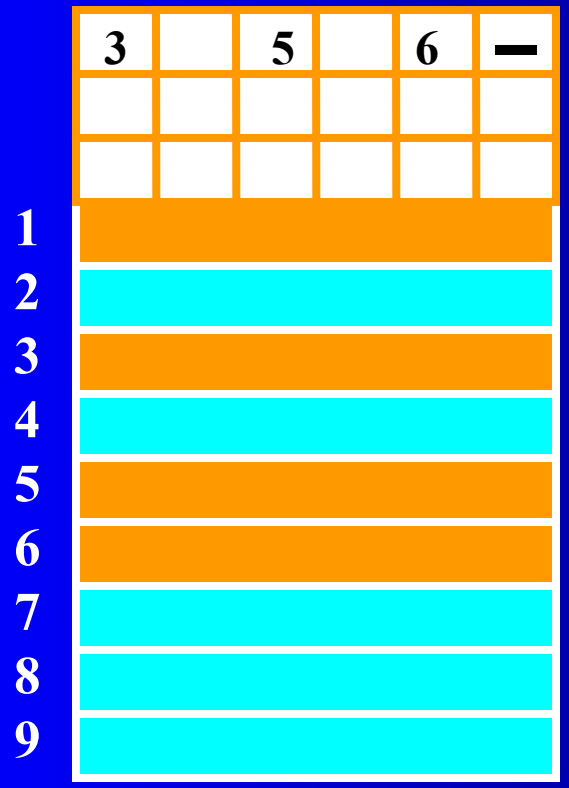
Связный список кластеров



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

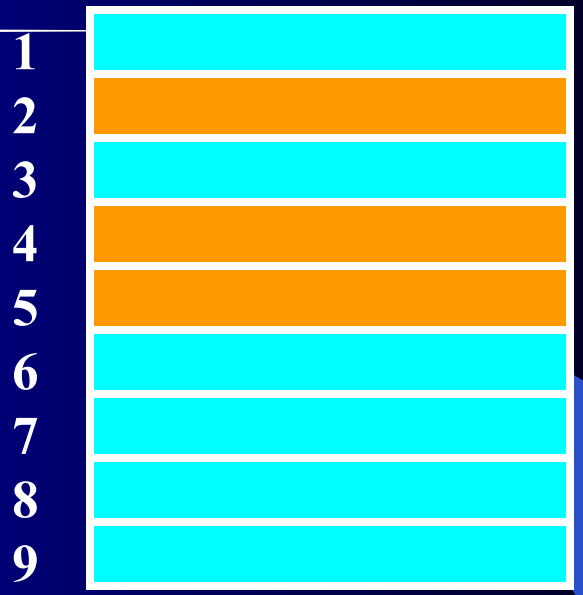


Область индексов

В)

Файл 1, 3, 5, 6

Перечень номеров кластеров



Файл 2, 4, 5

Г)

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

Недостаток: рост адресной информации с увеличением емкости диска

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

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

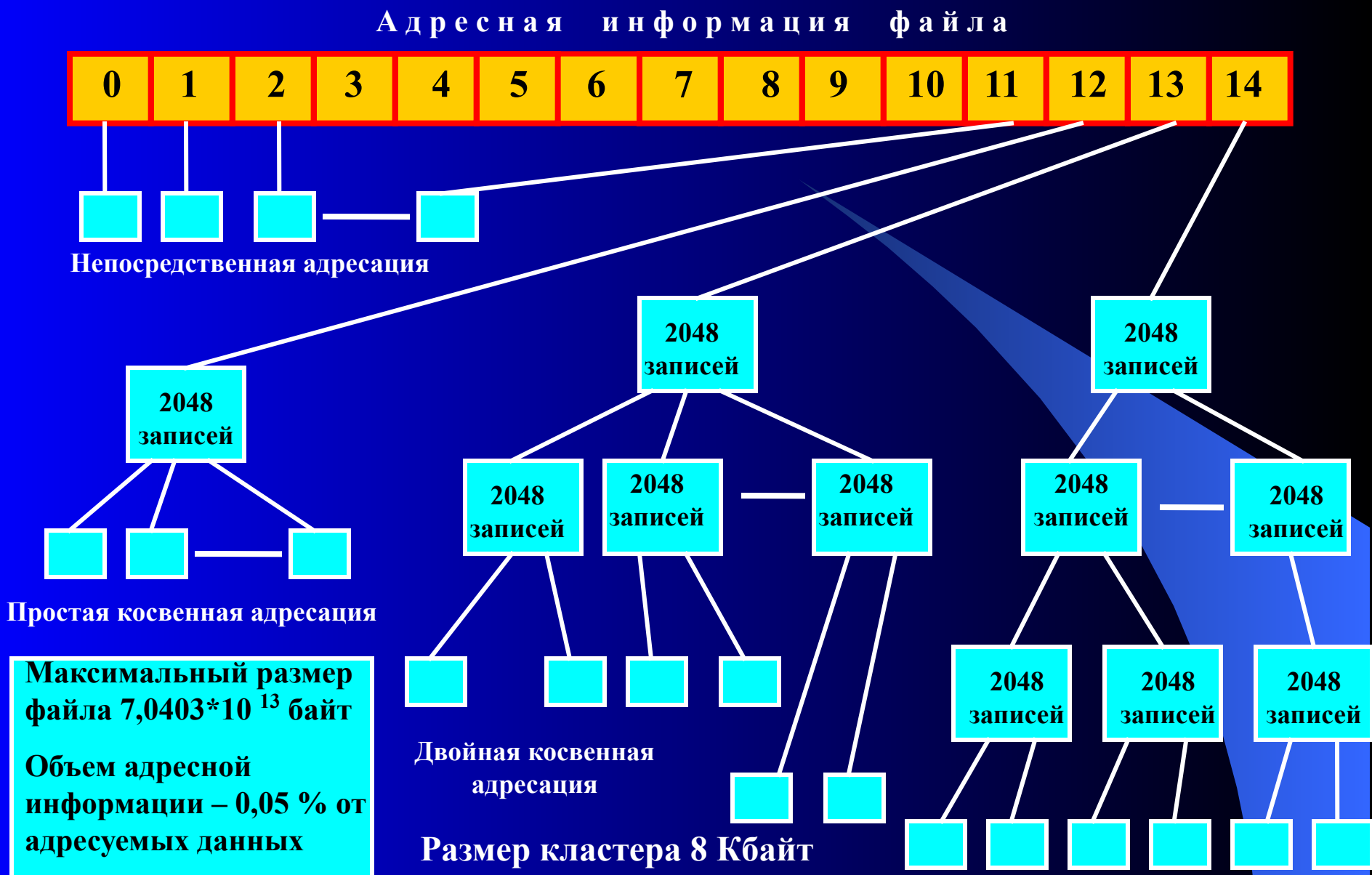
I- узел (index node)

Достоинства: I-узел находится в памяти только при открытии файла, что сокращает объем адресной информации; объем адресной информации не зависит от емкости диска, а лишь от числа открытых файлов; высокая скорость доступа к произвольному кластеру файла благодаря прямой адресации.

Недостатки: фиксированного количества адресов может оказаться недостаточным для адресации файла, отсюда необходимость сочетания прямой и косвенной адресации

Кластер, содержащий дополнительные дисковые адреса

Файловая система ОС UNIX ufs

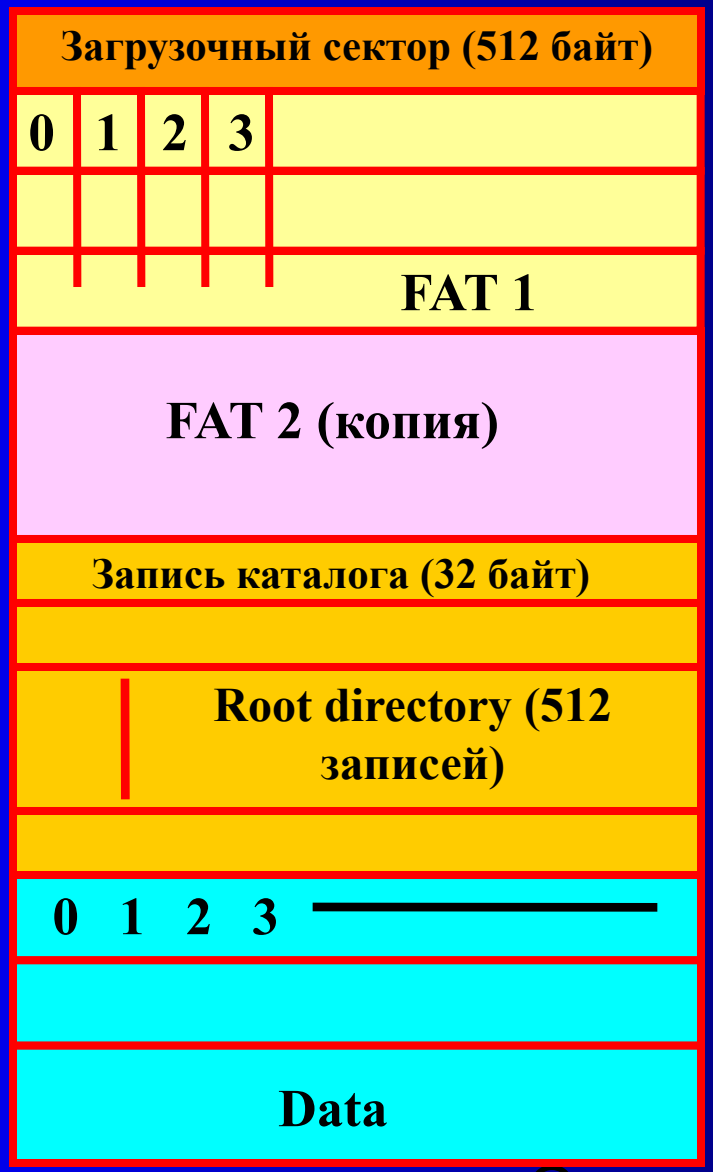


Максимальный размер файла 7,0403*10 ^{13} байт

Объем адресной информации – 0,05 % от адресуемых данных

Размер кластера 8 Кбайт

Физическая организация FAT



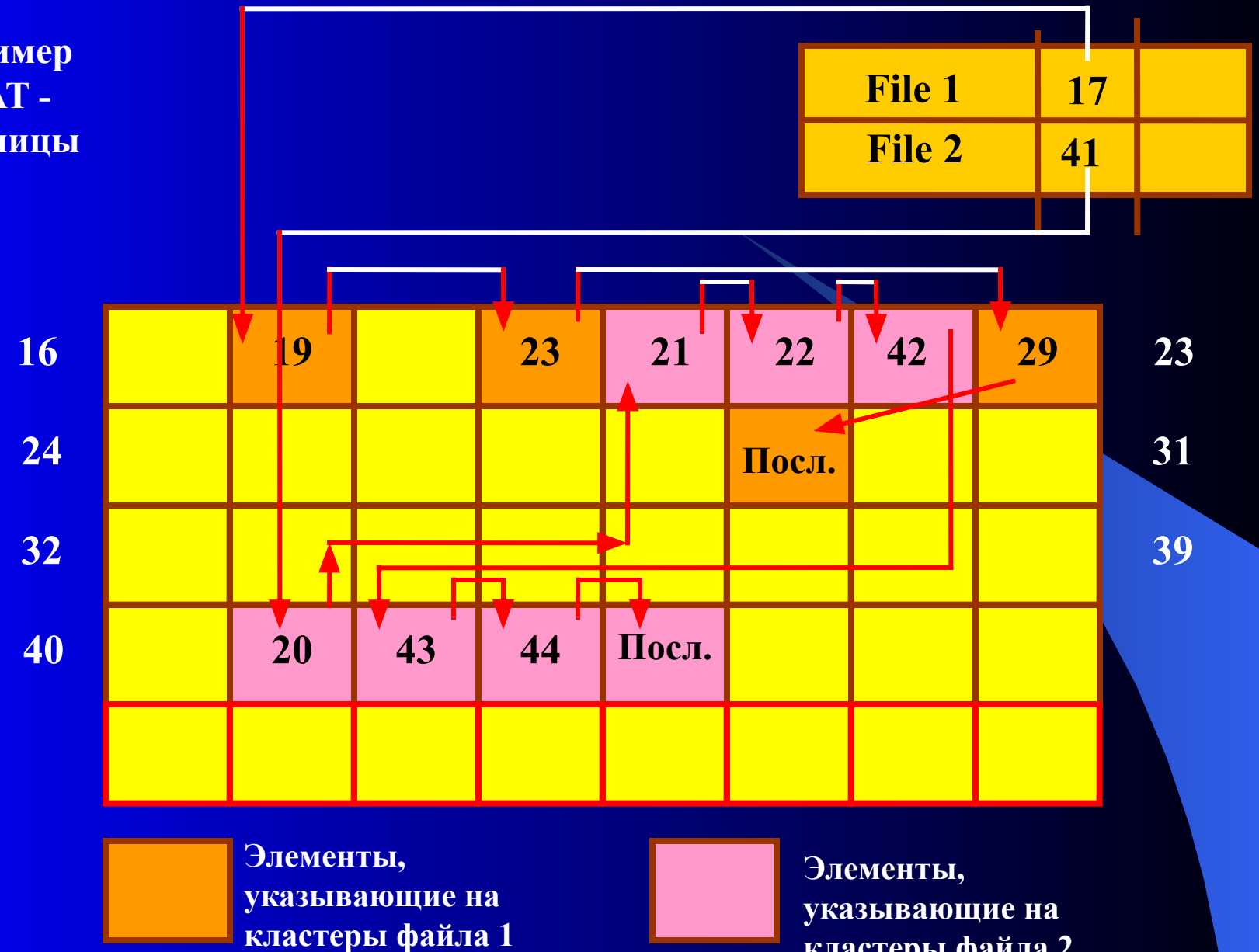
Индексные указатели, связанные с кластерами принимают значения:

кластер свободен (0000h); последний кластер файла (fff8h – ffffh); кластер поврежден (fff7h); резервный кластер (fff0h - fff6h)

Формат каталога

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

Пример FAT - таблицы



Основные характеристики файловых систем

FAT	Разрядность указателя	Число кластеров	Максимальный объем кластера	Максимальный размер раздела	Имя файла
FAT12	12	4096	4 Кбайт	16 Мбайт	8.3
FAT16 255.3	16	65536	64 Кбайт	4 Гбайт	8.3
FAT 32	32	4 Г	32 Кбайт	2^{32} по 32 Кбайт	255.3
NTFS	64	2^{64}	4 Кбайт	2^{64} по 4 Кбайт	255.3

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.

4.4.6. Операции управления каталогами и файловые операции

Win32 API	UNIX	Описание
CreateDirectory	mkdir	Создать новый каталог
RemoveDirectory	rmdir	Удалить пустой каталог
FindFirstFile	opendir	Инициализация для начала чтение записей каталога
FindNextFile	readdir	Прочитать следующую запись каталога
MoveFile	rename	Переместить файл из одного каталога в другой
SetCurrentDirectory	chdir	Изменить текущий рабочий каталог
CreateFile	open	Создать (открыть) файл, вернуть дескриптор файла
DeleteFile	unlink	Удалить существующий файл
CloseHandle	close	Закрыть файл
ReadFile	read	Прочитать данные из файла
WriteFile	write	Записать данные в файл
SetFilePointer	lseek	Уст-вить указатель в файле в определенную позицию
GetFileAttributes	stat	Вернуть атрибуты файла
LockFile	fcntl	Заблокировать файл для взаимного исключения
Unlock File	fcntl	Отменить блокировку области файла

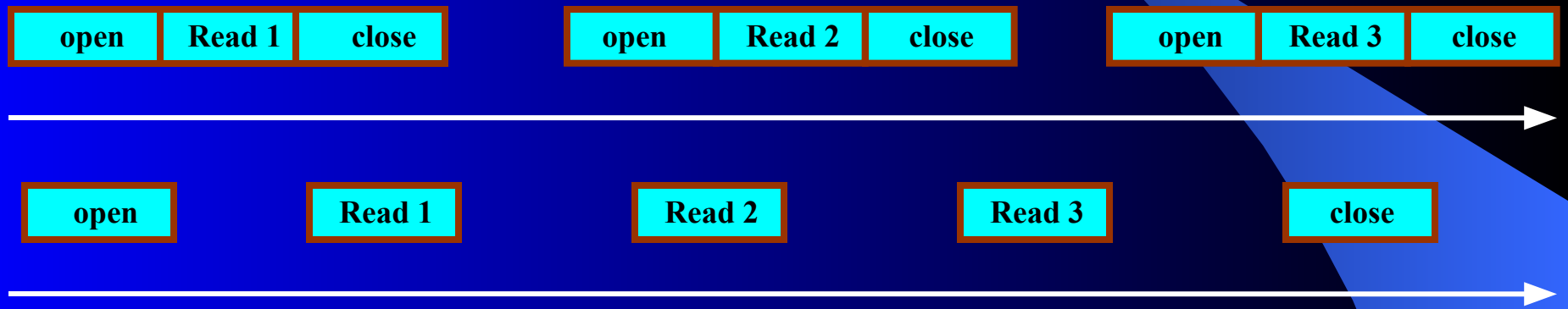
Операцио

нные

Способы выполнения файловых операций

Последовательность универсальных действий:

1. По символьному имени файла найти его характеристики, которые хранятся в файловой системе на диске.
2. Скопировать характеристики файла в оперативную память, поскольку только в этом случае программный код может их использовать.
3. На основании характеристик файла проверить права пользователя на выполнение запрошенной операции (чтение, запись, удаление и т. п.).
4. Очистить область памяти, отведенную под временное хранение характеристик файла.



Примеры системных вызовов для работы с файлами:

fd = create (“abc”, mode); fd = open (“file”, how);
read (fd, buffer, nbytes); write(fd, buffer, nbytes);

Стандартные файлы ввода – вывода, перенаправление вывода

read (stdin, buffer, nbytes); write(stdout, buffer, nbytes);
< file - перенаправление ввода,
> file – перенаправление вывода на файл

Примеры системных вызовов для работы с файлами

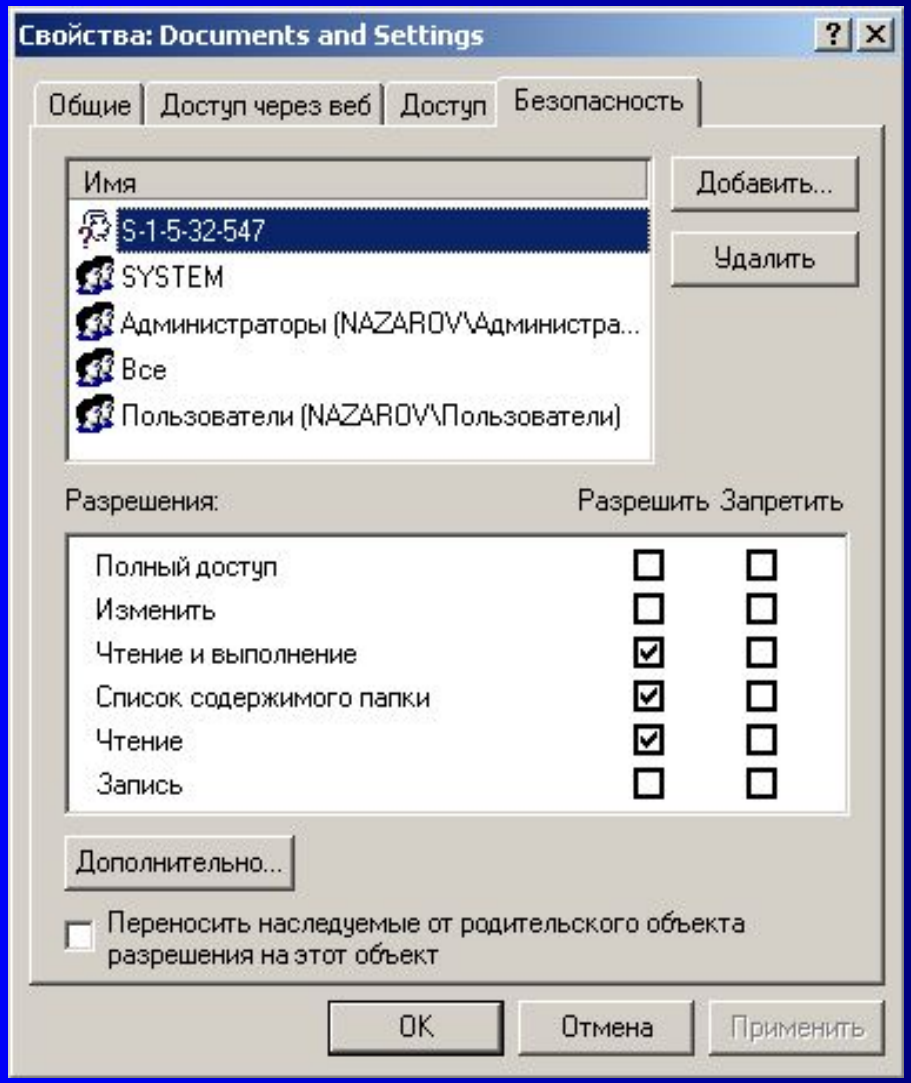
fd = creat (“name”, mode) – файла с заданным режимом защиты;
fd = open (“name”, how) – открыть файл для чтения, записи или и того и другого;
s = close (fd) – закрыть открытый файл;
n = read (fd, buffer, nbytes) – прочитать данные из файла в буфер;
n = write (fd, buffer, nbytes) – записать данные из буфера в файл;
position = lseek (fd, offset, whence) – переместить указатель в файле;
s = fstat | stat (fd | “name”, &buf) - получить информацию о состоянии файла.

При выполнении программы стандартным образом файлы с дескрипторами 0, 1 и 2 уже открыты для стандартного ввода, стандартного вывода и стандартного потока сообщений об ошибках.

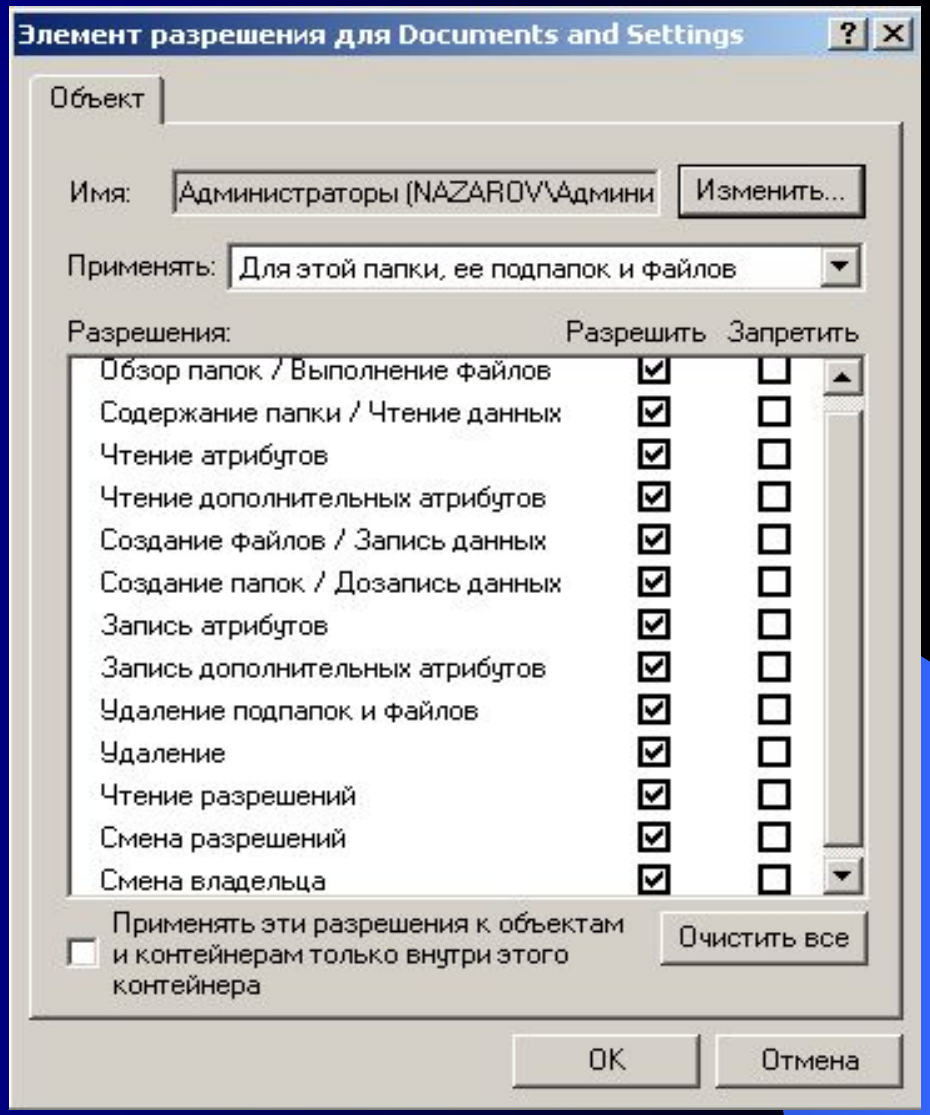
n = read (stdin, buffer, nbytes); n = write (stdout, buffer, nbytes)

stdin = 0; stdout =1; stderr = 2.

Разрешения на доступ к каталогам

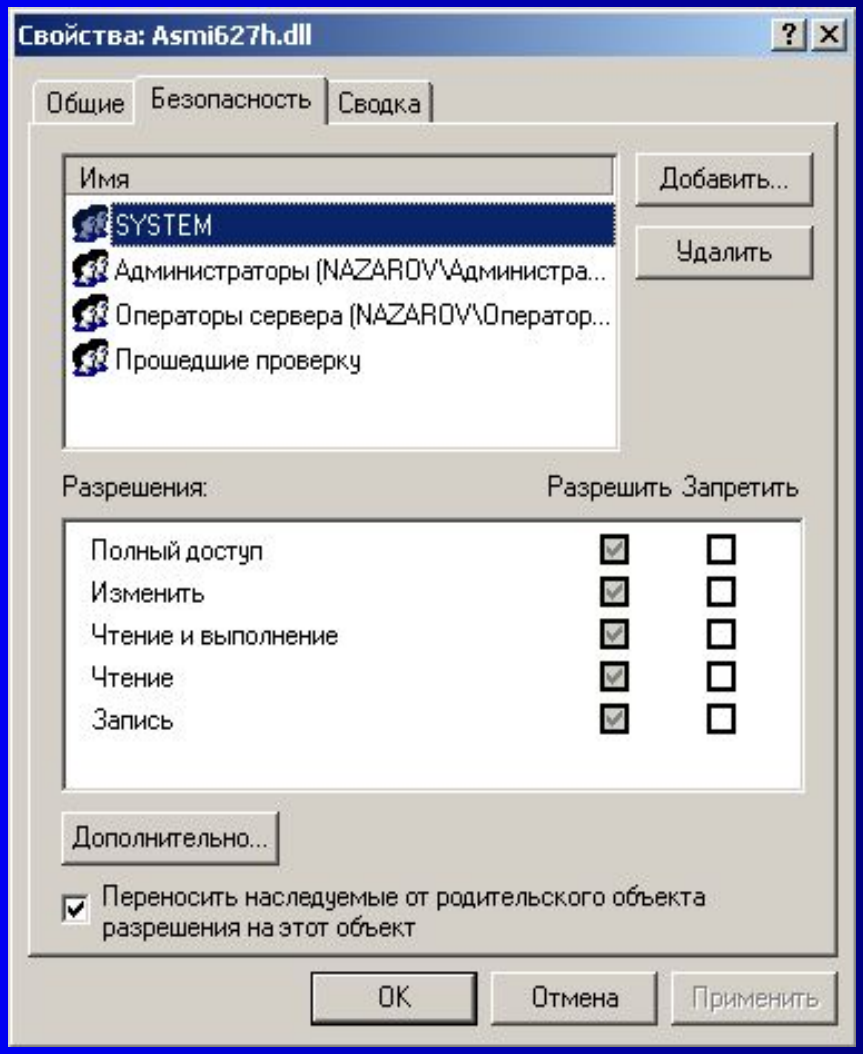


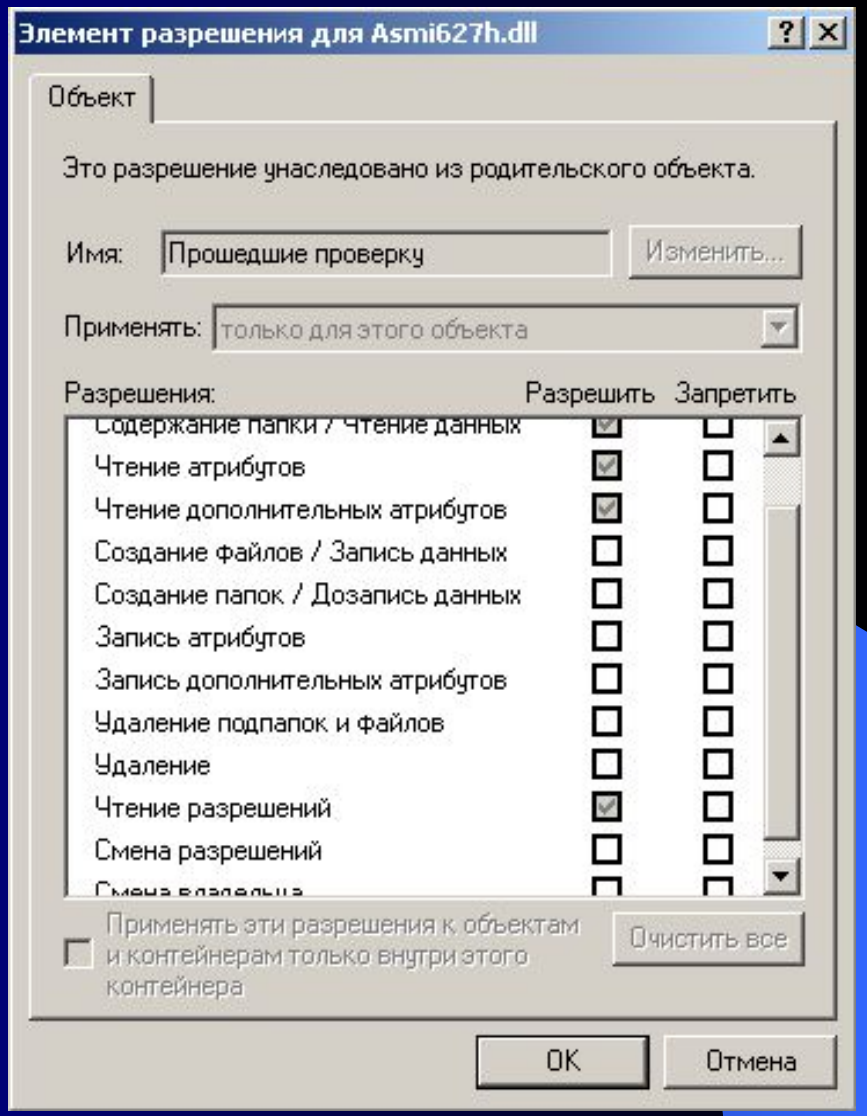
Стандартные разрешения



Специальные разрешения

Разрешения на доступ к файлам





Квоты дискового пространства