

# ОПЕРАЦИОННЫЕ СИСТЕМЫ

<https://vikchas.ru>

## Лекция 5 «Файловая система операционной системы»

Часовских Виктор Петрович  
доктор технических наук, профессор кафедры  
ШИиКМ, ФГБОУ ВО «Уральский государственный  
экономический университет

Екатеринбург 2026

## Файловая система операционной системы

**Файловая система** – это часть ОС, организующая работу с данными, хранящимися во внешней памяти, и обеспечивающей пользователю удобный интерфейс при работе с такими данными. Термин файловая система определяет, прежде всего, принципы доступа к данным, организованным в файлы. Говоря о файловых системах иногда употребляют термин **система управления файлами** – под которой следует понимать некую конкретную реализацию файловой системы, то есть комплекс программных модулей, обеспечивающих работу с файлами в конкретной ОС.

Непосредственное взаимодействие с диском при организации хранения информации на магнитном диске требует, например, знания устройства контроллера диска, особенностей работы с его регистрами. Очевидно, что такое взаимодействие – прерогатива системы ввода-вывода ОС (драйвера диска).

Для того чтобы избавить пользователя компьютера от сложностей взаимодействия с аппаратурой, была применена абстрактная (логическая) модель файловой системы, в которой операции записи или чтения файла концептуально проще, чем низкоуровневые операции работы с устройствами.

Логическая модель файловой системы «материализуется» в виде дерева каталогов, выводимого на экран такими утилитами, как Norton Commander или Windows Explorer, в символьных составных именах файлов, в командах работы с файлами.

Базовым элементом этой модели является файл, который так же, как и файловая система в целом, может характеризоваться как логической, так и физической структурой.

# Логическая организация и физическая реализация

## 1. Логическая организация файлов

- 1.1 Определение файла. Атрибуты и данные файла
- 1.2 Файловые системы и системы управления файлами
- 1.3 Логическая организация данных файла
- 1.4 Интерфейс файловой системы. Операции с каталогами и файлами

## 2. Физическая реализация хранения файловых систем

- 2.1 Низкоуровневое форматирование и создание разделов жёстких дисков
  - 2.1 Модели размещения логических томов файловых систем на жёстких дисках
- 2.2 Высокоуровневое форматирование логических томов (форматирование файловых систем)
- 2.3 Физическая реализация хранения атрибутов файла
- 2.4 Физическая реализация хранения данных файла

# Понятие файла

Под **файлом** понимают

1. некоторый **набор данных**,
2. связанные с этим набором **атрибуты** (имя, размер и т.д.)
3. множество допустимых **операций над атрибутами и данными**

# Логическая организация файла

Файл

=

Атрибуты файла

+

Данные файла

+

Операции с атрибутами и данными

# Файловая система и система управления файлами

<b>Файловая система</b>	<b>способ организации данных на диске</b> или на каком-нибудь ином носителе данных <b>принципы доступа к данным,</b> организованных в файлы
<b>Система управления файлами (файловая подсистема ОС)</b>	<b>конкретная реализация файловой системы</b> - комплекс программных модулей, обеспечивающих работу с файловой системой в конкретной операционной системе.

## Файловая система

Файловая система предоставляет возможность пользователям ФС работать

- с **логическим уровнем** структуры данных
- и **операциями**, выполняемых над структурами данных в процессе их обработки.

# Система управления файлами

## Функции системы управления файлами

1. Распределение дискового пространства файловой системы
2. Операции с файлами с помощью обращений к программным модулям системы управления файлами (часть API ОС ориентирована на работу с файлами)
3. Защита файлов от несанкционированного доступа

В ОС может быть несколько систем управления файлами, что обеспечивает им возможность работать с несколькими файловыми системами

Системы управления файлами используют соответствующие вызовы API. С другой стороны, системы управления файлами сами дополняют API ОС новыми вызовами.

# Атрибуты файлов

Список атрибутов файлов зависит от конкретной файловой системы

При создании файла, файлу присваивается **основной атрибут** – его **имя**. По имени осуществляют **доступ к файлу**.

Точные правила именования файлов варьируются от системы к системе. В именах файлов разрешается использование букв, цифр и специальных символов.

Файловые системы могут поддерживать **короткие** или **длинные** имена файлов.

В некоторых файловых системах в именах файлов различаются **прописные и строчные символы**.

# Возможные атрибуты файлов

<b>Атрибут</b>	<b>Значение</b>
<b>Имя</b>	Имя файла
<b>Защита</b>	Кто и каким образом может получить доступ к файлу
<b>Пароль</b>	Пароль для получения доступа к файлу
<b>Создатель</b>	Идентификатор пользователя, создавшего файл
<b>Владелец</b>	Текущий владелец
<b>Флаг «только чтение»</b>	0 - для чтения/записи; 1 - только для чтения
<b>Флаг «скрытый»</b>	0 - нормальный; 1 - не показывать в перечне файлов каталога
<b>Флаг «системный»</b>	0 - нормальный; 1 - системный
<b>Флаг «архивный»</b>	0 — заархивирован; 1 - требуется архивация
<b>Флаг ASCII/двоичный</b>	0- ASCII; 1 - двоичный
<b>Флаг произвольного доступа</b>	0 - только последовательный доступ; 1 - произвольный доступ
<b>Флаг «временный»</b>	0 — нормальный; 1 — для удаления файла по окончании работы процесса

<b>Атрибут</b>	<b>Значение</b>
<b>Длина записи</b>	Количество байтов в записи
<b>Время создания</b>	Дата и время создания файла
<b>Время последнего доступа</b>	Дата и время последнего доступа файла
<b>Время последнего изменения</b>	Дата и время последнего изменения файла
<b>Текущий размер</b>	Количество байтов в файле
<b>Максимальный размер</b>	Количество байтов, до которого можно увеличивать размер файла

# Данные файла

<p><b>Поле</b> (field)</p>	<p>основной элемент данных. Содержит единственное значение. Характеризуется <b>длиной</b> и <b>типом</b> данных(например , строка ASCII, десятичное число и т.п.).</p> <p>В зависимости от структуры файла поля м.б. либо <b>фиксированной, либо переменной длины.</b></p>
<p><b>Запись</b> (record)</p>	<p>набор связанных между собой <b>полей</b>, которые могут быть обработаны как <b>единое целое</b> некоторой прикладной программой. В зависимости от структуры записи могут быть <b>фиксированной или переменной</b> длины. Запись имеет переменную длину, если некоторые из ее полей – переменной длины или если переменное количество полей в записи.</p>

# Данные файла

## Файл (file)

- набор данных, организованных в виде
- **совокупности записей** одинаковой структуры(однородных записей) и
- **совокупности атрибутов**, определяющих характеристики файла

<b>Атрибут</b>	<b>Значение</b>
<b>Имя</b>	Имя файла
<b>Защита</b>	Кто и каким образом может получить доступ к файлу
<b>Пароль</b>	Пароль для получения доступа к файлу
<b>Создатель</b>	Идентификатор пользователя, создавшего файл
<b>Владелец</b>	Текущий владелец
<b>Флаг «только чтение»</b>	0 - для чтения/записи; 1 - только для чтения
<b>Флаг «скрытый»</b>	0 - нормальный; 1 - не показывать в перечне файлов каталога
<b>Флаг «системный»</b>	0 - нормальный; 1 - системный
<b>Флаг «архивный»</b>	0 — заархивирован; 1 - требуется архивация
<b>Флаг ASCII/двоичный</b>	0- ASCII; 1 - двоичный
<b>Флаг произвольного доступа</b>	0 - только последовательный доступ; 1 - произвольный доступ
<b>Флаг «временный»</b>	0 — нормальный; 1 — для удаления файла по окончании работы процесса

<b>Атрибут</b>	<b>Значение</b>
<b>Длина записи</b>	Количество байтов в записи
<b>Время создания</b>	Дата и время создания файла
<b>Время последнего доступа</b>	Дата и время последнего доступа файла
<b>Время последнего изменения</b>	Дата и время последнего изменения файла
<b>Текущий размер</b>	Количество байтов в файле
<b>Максимальный размер</b>	Количество байтов, до которого можно увеличивать размер файла

# Логическая организация данных файла

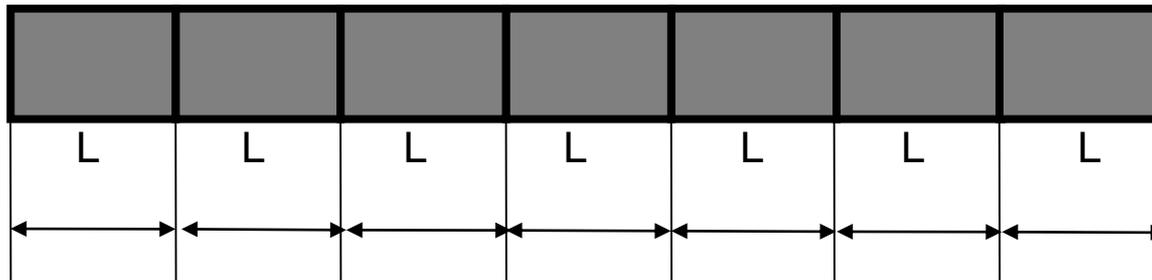
- Неструктурированная последовательность **байтов** (поток байтов)
- Последовательная организация с логическими **записями фиксированной длины**
- Последовательная организация с логическими **записями переменной длины**
- Организация размещения записей файлов с использованием **таблицы индексов**

# Логическая организация данных файла

Файл - неструктурированная последовательность байтов

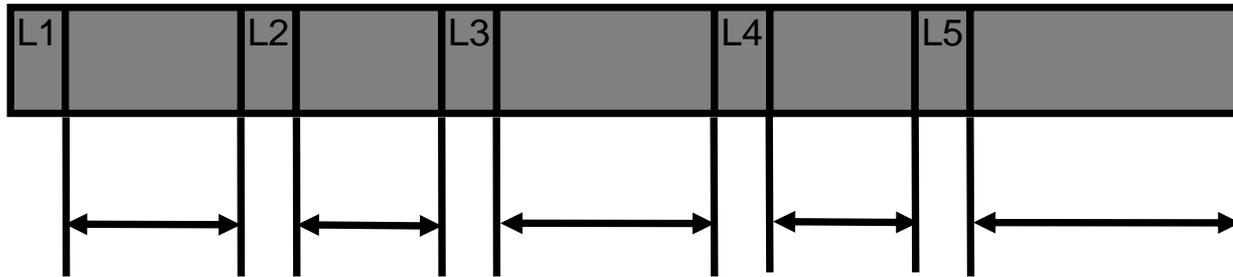


Последовательный файл с записями фиксированной длины

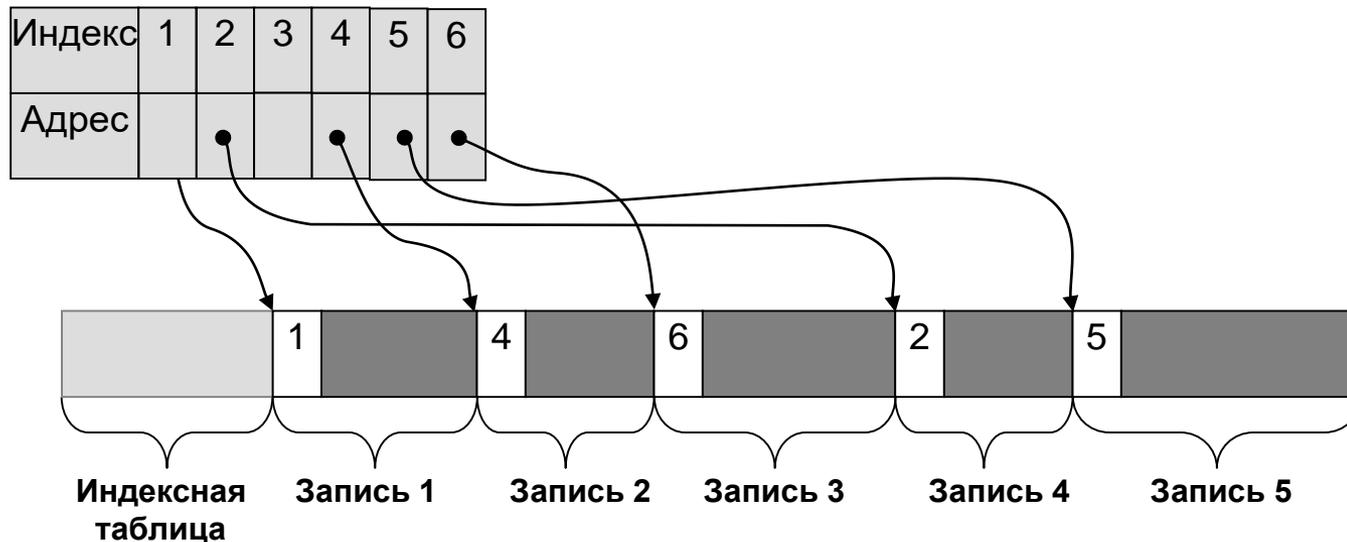


# Логическая организация данных файлов

Последовательный файл с записями переменной длины



Индексированный файл



# Структура файловой системы на логическом уровне

- **Каталог** - структурированный объект, состоящий из списка элементов
- Каждый элемент каталога содержит информацию:
  - либо о **файле**, включённом в данный каталог,
  - либо о **каталоге**, являющимся каталогом **нижнего уровня** по отношению к данному
- Каталогии образуют **иерархическую структуру файловой системы**
- Каталогии и файлы в такой структуре именуются относительно **каталога верхнего уровня файловой системы (корневого каталога)**

# Имя пути к файлу при иерархической структуре каталога

При организации файловой системы в виде дерева каталогов требуется способ указания **пути** к файлу.

**Обычно используются два метода.**

## Абсолютное имя пути

Каждому файлу дается имя, состоящее из имен всех каталогов от корневого до того, в котором содержится файл, и имени самого файла.

# Имя пути к файлу при иерархической структуре каталога

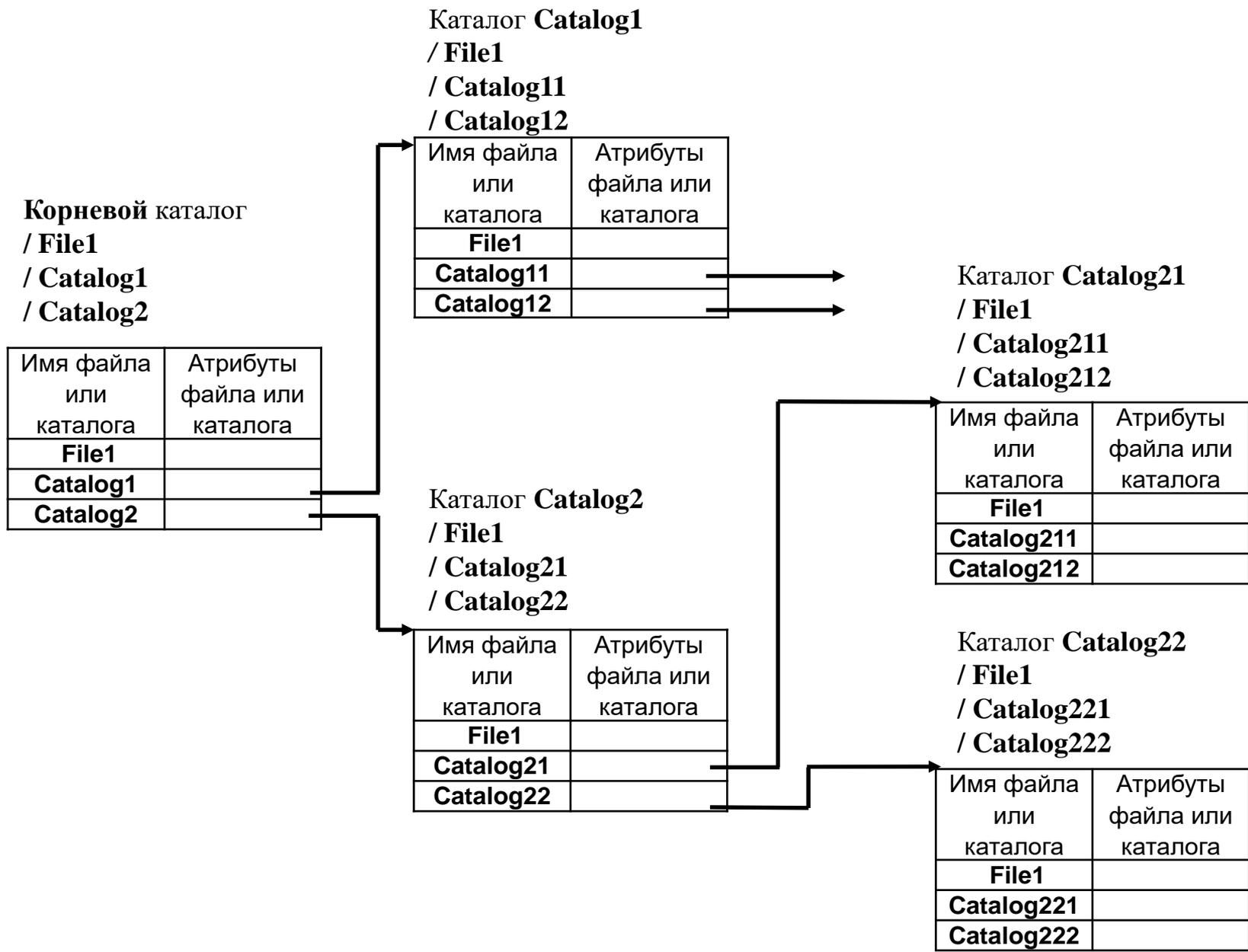
## Относительное имя пути

Используется концепция **рабочего(текущего)** каталога

Пользователь может назначить один из каталогов текущим рабочим каталогом.

В этом случае все имена путей, не начинающиеся с символа разделителя, считаются относительными и отсчитываются относительно текущего каталога.

# Иерархическая структура файловой системы



# Некоторые операции с каталогами, доступные клиентам файловой системы

**Создание** каталога

**Удаление** каталога

**Установка** прав доступа к каталогу

**Создание** файла

**Удаление** файла

**Создание символической ссылки** на файл  
другого каталога

**Удаление символической ссылки** на файл  
другого каталога

# Некоторые операции с файлами, доступные клиентам файловой системы

**Открытие** файла

**Запись** в файл

**Чтение** из файла

**Установка** прав доступа к файлу

**Закрытие** файла

# Файловая система операционной системы проекта ИИ

Файловая система в проекте, связанном с искусственным интеллектом, обычно включает в себя несколько ключевых компонентов:

## Основные компоненты

Каталог данных- для хранения обучающих, тестовых и валидационных наборов данных

Каталог моделей- для хранения обученных моделей и контрольных точек

Каталог исходного кода- для кодовой базы проекта

Каталог конфигураций- для хранения настроек и гиперпараметров

Каталог логов- для записи результатов экспериментов и отладочной информации

# Типичная структура

Проект\_ии/

├── Данные/

| ├── необработанные исходные данные

| | ├── датасет\_изображений.zip

| | └── текстовый\_набор.txt

| ├── подготовленные для обучения данные /

| | ├── обучающий\_набор.npz

| | └── тестовый\_набор.npz

| └── данные из внешних источников

| └── Предобученные эмбединги

(распределённые представления слов) в формате .bin

## └─ **Модели/**

| └─ финальные обученные модели /

| | └─ нейросеть\_v1.h5

| | └─ нейросеть\_v2.pt

| └─ промежуточные состояния обучения /

| | └─ единый проход по всему обучающему набору данных, в котором все примеры используются для корректировки весов и смещений модели. \_10.ckpt (с неполной безопасностью)

| | └─ единый проход по всему обучающему набору данных, в котором все примеры используются для корректировки весов и смещений модели \_20.ckpt

## |— Исходники/

| |— предобработка/

| | |— \_\_init\_\_.py

| | |— очистка\_данных.py

| |— обучение/

| | |— \_\_init\_\_.py

| | |— тренер.py

| |— оценка/

| | |— \_\_init\_\_.py

| | |— метрики.py

| |— утилиты/

| | |— \_\_init\_\_.py

| | |— визуализация.py

└─ **Конфигурация/**

| └─ параметры архитектуры  
модели.yaml

| └─ параметры обучения и  
эксперимента.yaml

└─ **Наборы/**

└─ исследование\_данных.ipynb

└─ эксперименты.ipynb

## Особые требования

Файловая система для проектов ИИ должна обеспечивать:

- Эффективный доступ к большим объемам данных
- Версионирование (способ группировки и маркировки изменений в ходе разработки) моделей и данных
- Кэширование для оптимизации доступа
- Распределенный доступ при необходимости параллельной обработки

# Физическая реализация файловой системы

Физическая реализация файловых систем решает вопросы

хранения файловых систем на внешних устройствах,

физической организации файлов и каталогов файловой системы,

управления дисковым пространством,

обеспечения надежной и эффективной работы файловой системы.

# Объекты физической реализации файловой системы

**Физический жёсткий диск** (или совместимое внешнее устройство)

**Раздел(partition)** – непрерывная часть адресного пространства физического диска

**Логический том** – область, занимаемая файловой системой (адресное пространство файловой системы)

**Файловая система** – физическая организация логического тома для хранения файлов и каталогов

# Физическая адресация данных на жёстком диске

**Сектор**(sector) – минимальный блок хранения и передачи информации

**Дорожки**(track) – concentric magnetic tracks on the surface of the disk

**Цилиндр**(cylinder) – group of tracks of the same radius

**Физический адрес сектора на диске:**

[ **c-h-s** ]

**c** – номер цилиндра(cylinder)

**h** – номер рабочей поверхности(head)

**s** – номер сектора на дорожке(sector)

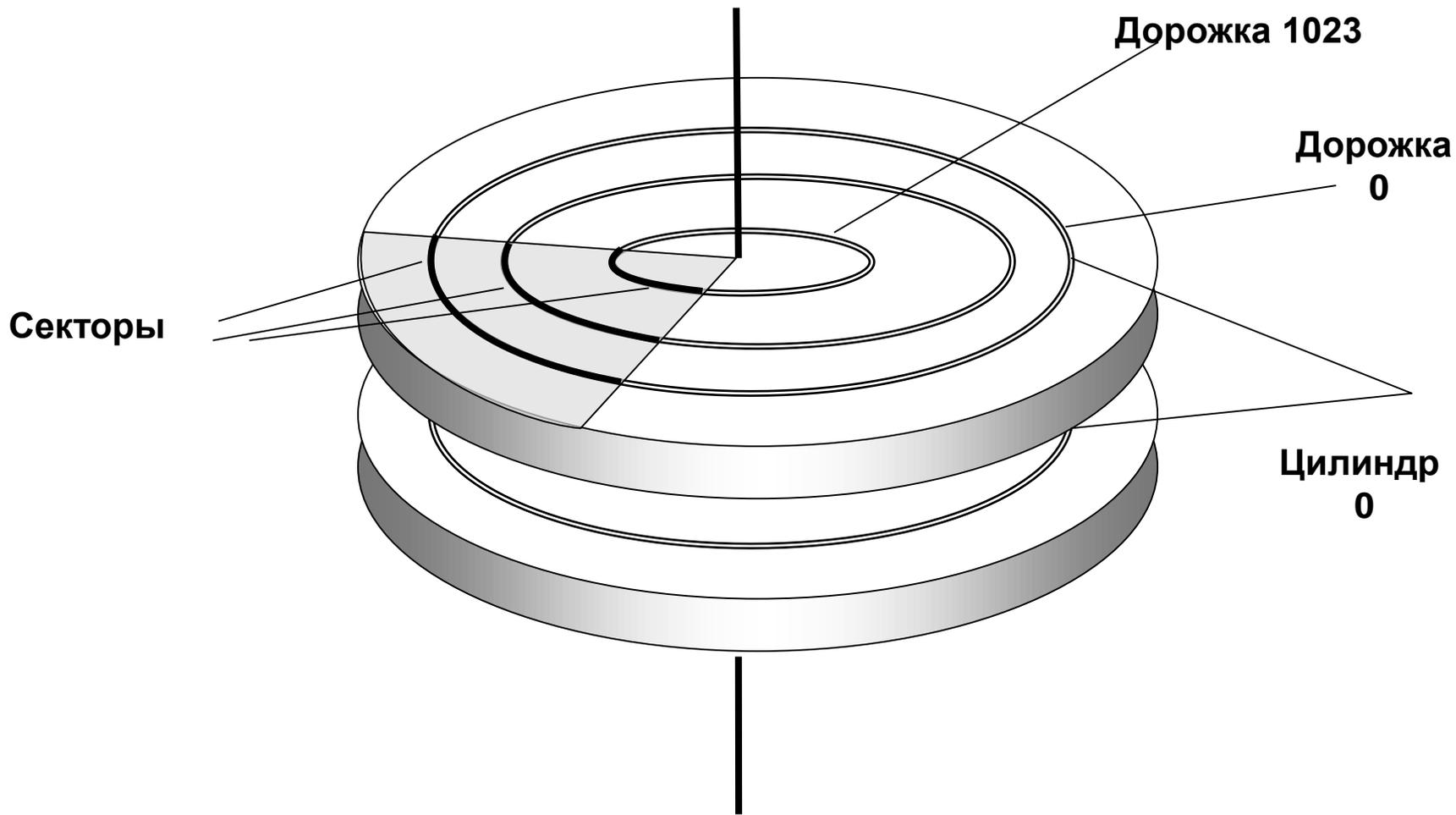
# Этапы форматирования жесткого диска

**Низкоуровневое форматирование, Low-Level formatting** (физическая разметка диска на цилиндры, дорожки, секторы)

Разбиение диска на **разделы (partition)**

Создание **ЛОГИЧЕСКИХ ТОМОВ** файловых систем

**Высокоуровневое форматирование**  
логических томов (форматирование файловых систем)



## Низкоуровневое форматирование магнитного диска

## Разбиение жёсткого диска на разделы

С использованием таблицы разделов **PT** (partition table) и **MBR**(Master Boot Record)

Разделы:

**первичные** (primary)

**расширенные** (extended)

**логические диски**(logical)

С использованием таблицы разделов **GPT (GUID Partition Table)**

Допускает неограниченное количество разделов. Лимит устанавливает операционная система(Windows допускает не более 128 разделов).

# Модели размещения логических томов файловых систем на жёстких дисках

## 1. Модель на основе создания логического тома в разделе физического устройства

Каждый жёсткий диск делится на **один** или **несколько** разделов, каждый из которых может содержать **один логический том** и одну файловую систему

# **Модели размещения логических томов файловых систем на жёстких дисках**

## **2. Модель на основе создания логического тома в нескольких разделах одного или нескольких физических устройств**

Концепция диспетчера томов , обеспечивающая представление нескольких устройств в виде одного устройства (динамические диски **Windows** и логические тома **Linux**)

Диспетчер томов поддерживает **изменение конфигурации логического тома**

# Модели размещения логических томов файловых систем на жёстких дисках

## 3. Модель на основе создания логического тома в пуле устройств хранения данных

Концепция объединения устройств в пул устройств хранения данных. Пул устройств хранения данных описывает физические характеристики хранения (размещение устройств, избыточность данных и т.д.) и выступает в качестве хранилища данных для создания файловых систем. Файловые системы автоматически расширяются в рамках пространства, выделенного для пула

# **Высокоуровневое форматирование логических томов (форматирование файловых систем)**

## **Стратегия форматирования логических томов**

Адресное пространство логического тома  
разбивается на **дисковые блоки данных**

Файловая система хранит адреса дисковых  
блоков и информацию об их состоянии

# Высокоуровневое форматирование логических томов (форматирование файловых систем)

## Выбор размера дискового блока

Дисковый блок данных состоит из **одного или нескольких смежных секторов**

Минимальный размер блока – **один сектор**

**Маленький размер блока** – хорошее использование дискового пространства, низкая производительность

**Большой размер блока** – высокая производительность, неэффективное использование дискового пространства

## Учет свободных и занятых блоков

Связные списки свободных и занятых дисковых блоков

Хранение информации о свободных и занятых блоках в виде **битового массива**

# Физическая реализация хранения файла

**Хранение файла**

**=**

**Хранение  
атрибутов файла**

**+**

**Хранение данных  
файла**

# Способы физической реализации хранения атрибутов файла. Каталоги

## 1. Хранение атрибутов файла только в каталоге

Все атрибуты файла хранятся **только в элементе каталога**, относящегося к данному файлу.

Каталог состоит из списка элементов **фиксированной длины**

Каждый элемент списка описывает **файл** (атрибуты файла и **дисковые адреса блоков данных файла**) или **каталог** нижнего уровня

**Корневой каталог** файловой системы создаётся при форматировании логического тома

# Хранение атрибутов файла в каталоге

Корневой каталог

/ File1

/ Catalog1

Имя файла или каталога	Атрибуты файла или каталога	Адреса данных файла или каталога
File1		
....		
Catalog1		

Данные файла File1

Данные файла

Данные каталога Catalog1

/ Catalog1 / File1

/ Catalog1 / Catalog2

Имя файла или каталога	Атрибуты файла или каталога	Адреса данных файла или каталога
File1		
Catalog2		

# Способы физической реализации хранения атрибутов файла. Индексные узлы

## 2. Хранение атрибутов файла в каталоге и индексном узле

Атрибуты файла хранятся как в элементе каталога, так и в специальной структуре данных (индексном узле)

Каталог содержит имя файла и адрес индексного узла, относящегося к данному файлу

Индексный узел содержит атрибуты файла и дисковые адреса блоков данных файла

# Хранение атрибутов файла в каталоге и индексном узле

Корневой каталог  
/ **File1**  
/ **Catalog1**

Имя файла или каталога	Адрес индексного узла файла или каталога
<b>File1</b>	
....	
<b>Catalog1</b>	

узле

индексный узел файла **File1**

атрибуты файла и адреса данных

Блоки данных файла

индексный узел каталога **Catalog1**

атрибуты и адрес каталога **Catalog1**

Каталог **Catalog1**  
/ **Catalog1 / File2**  
/ **Catalog1 / Catalog2**

Имя файла или каталога	Номер i-node файла или каталога
<b>File2</b>	
....	
<b>Catalog2</b>	

# **Физическая реализация хранения данных файла**

**Непрерывные файлы**

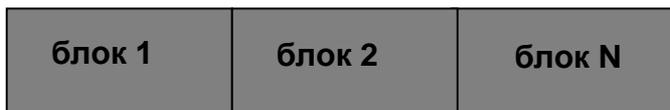
**Связные списки блоков данных**

**Связный список с таблицей адресов  
блоков данных**

**Адресация блоков данных с помощью  
индексных узлов**

# Физическая реализация хранения данных файла

## 1. Непрерывные файлы

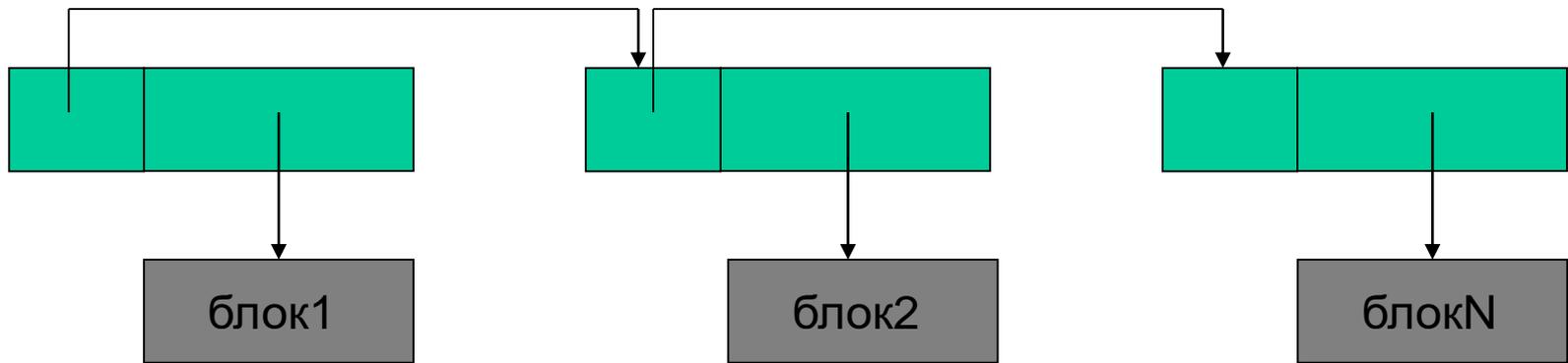


## 2. Связные списки

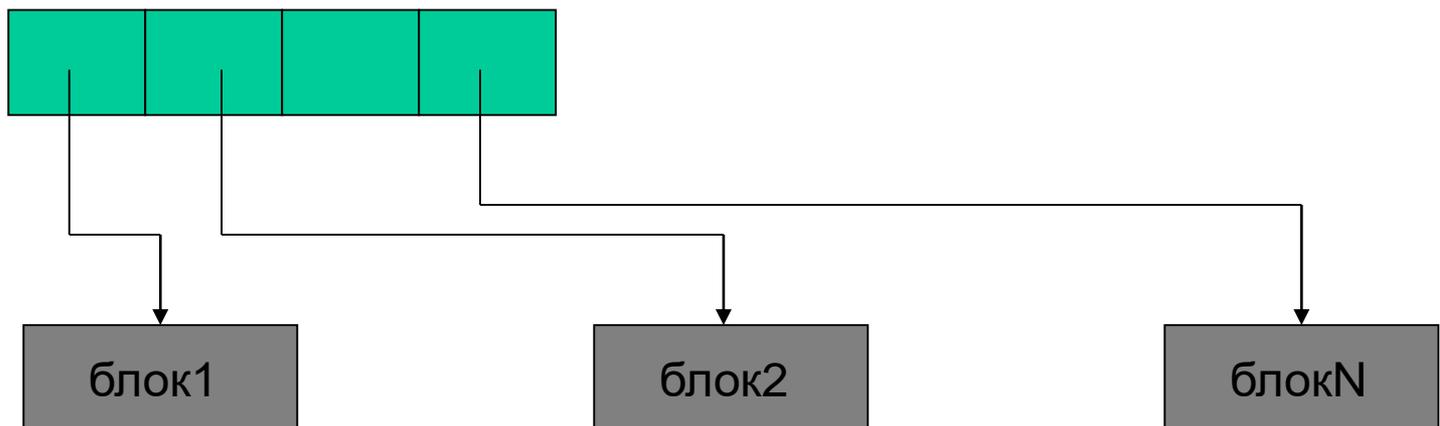


# Физическая реализация хранения данных файла

## 3. Связный список с таблицей адресов блоков данных



## 4. Адресация блоков данных с помощью индексных узлов



## Совместное использование файлов

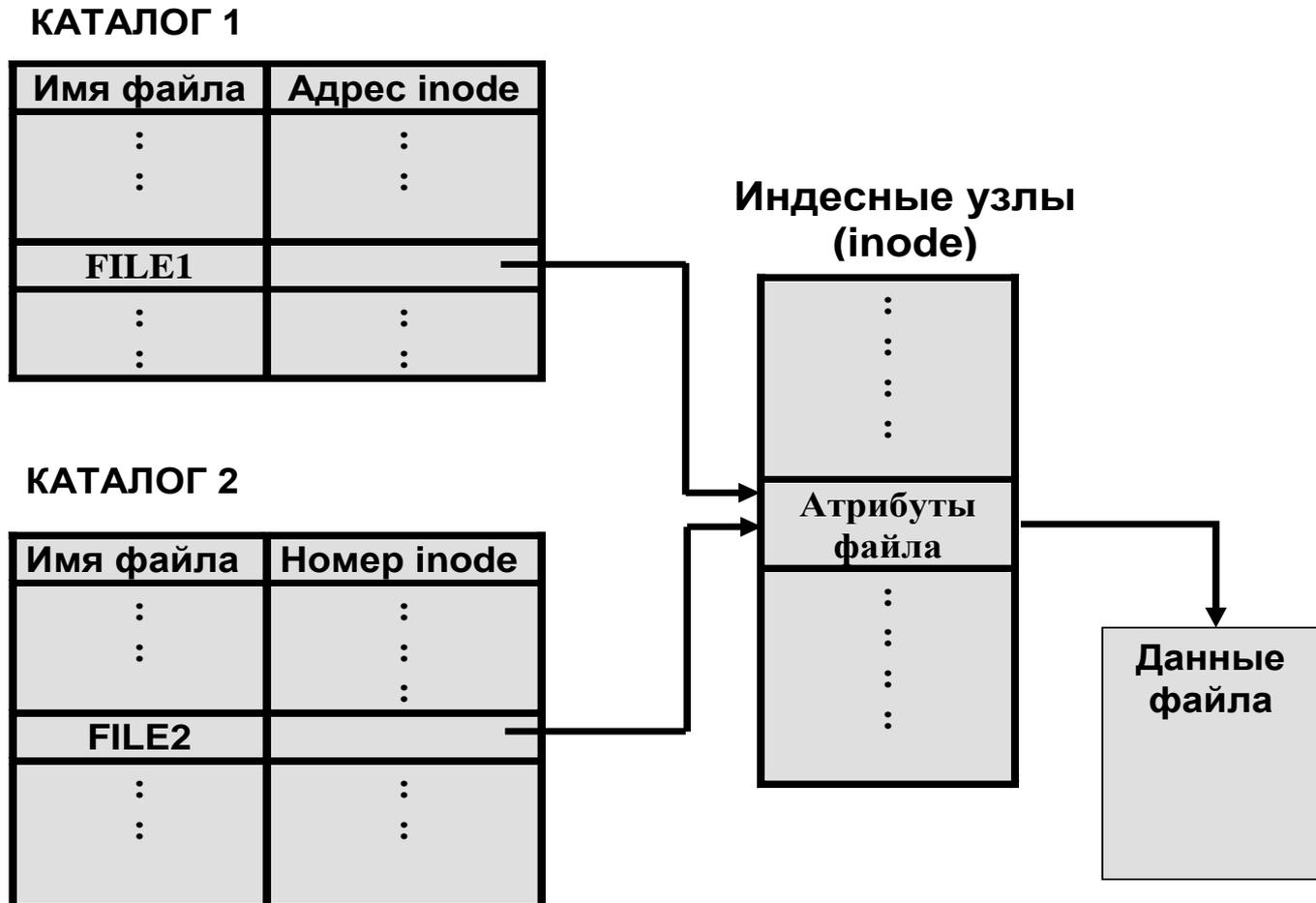
При совместном использовании файла несколькими процессами, изменения, вносимые в **атрибуты и данные файла** каким-либо процессом, должны быть видны **другим процессам**, использующим данный файл.

### Два способа решения проблемы:

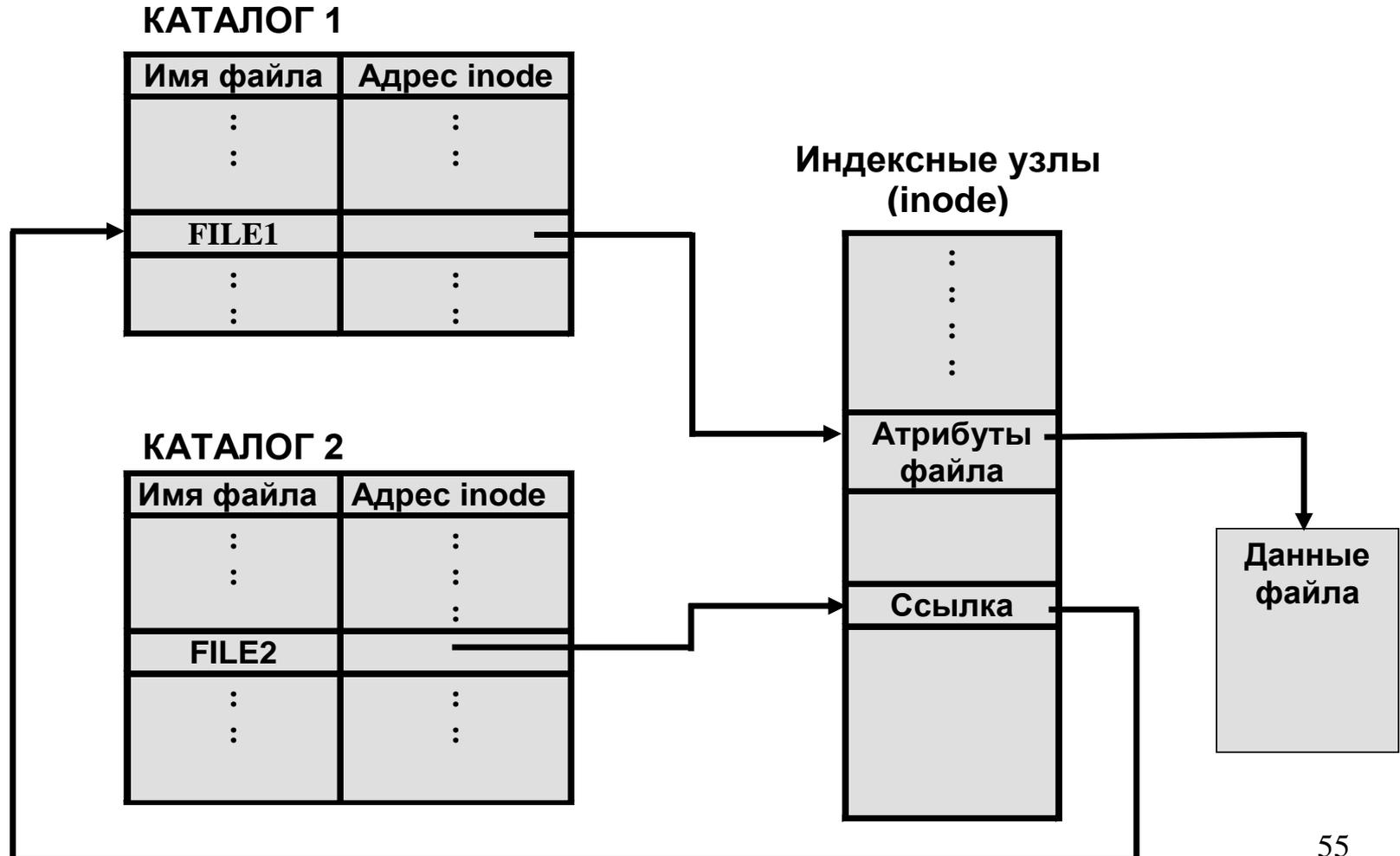
информация об атрибутах и блоках данных, занимаемых файлом, содержится не в каталоге, а в связанном с данным файлом **индексном узле**. В этом случае записи в каталогах будут просто указывать на эту структуру данных (**жёсткая ссылка**).

элемент каталога содержит символическую ссылку на элемент другого каталога, имеющего жёсткую ссылку на данные (**мягкая ссылка**)

# Жесткая связь имен файлов с атрибутами и данными



# Мягкая (символическая) связь имен файлов с атрибутами и данными



# Непротиворечивость файловой системы

<b>Традиционная файловая система</b>	<p>В промежутке времени между выполнением операций файловая система остается в противоречивом состоянии (команды <code>fsck</code> используется для просмотра и проверки состояния файловой системы с попыткой устранения противоречивости).</p>
<b>Файловая система с журналированием</b>	<p>Файловые операции регистрируются в отдельном журнале, и при необходимости могут быть безопасно воспроизведены.</p>
<b>Транзакционная файловая система</b>	<p>Любая последовательность операций либо полностью выполняется, либо полностью игнорируется.</p> <p>Состояние файловой системы всегда является непротиворечивым.</p>

# Благодарю за внимание!

