

# Файловые системы как предшественники баз данных

Файловые системы были первой попыткой компьютеризации известных всем ручных картотек.

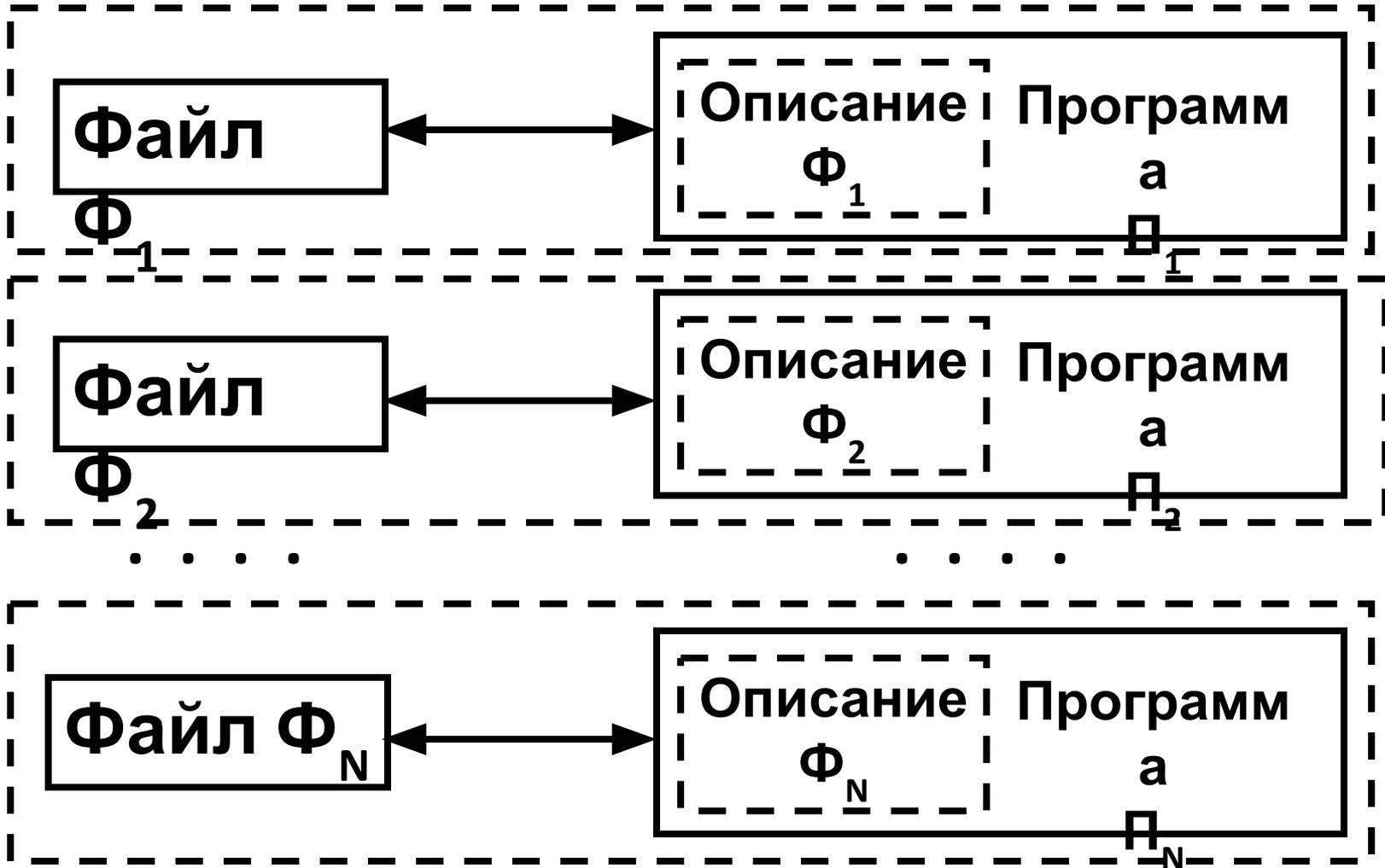
Хотя такие системы давно устарели, важно понимать присущие им недостатки, чтобы избежать их повторения в базах данных.

Кроме того, файловые системы до сих пор находят применение и принципы их работы нужно знать при переходе к новым технологиям.

Файловые системы реализуют децентрализованный подход к хранению и обработке информации, необходимой для функционирования некоторой организации.

Эта децентрализация обусловлена тем, что разные структурные подразделения (бухгалтерия, деканат, библиотека, здравпункт и т.п.) выполняют различные функции и, соответственно, пользуются разными видами данных.

# Файловая организация информационного обеспечения



# Недостатки файловых систем

1. Наличие **избыточности**, т.к. некоторые однотипные данные повторяются в разных файлах
2. Из-за этого неэкономно расходуются ресурсы памяти, а также возникают **противоречия (несоответствия)** для отдельных элементов данных (т.е. может **нарушаться целостность данных**)
3. Данные разделены и изолированы друг от друга, поэтому **невозможен прямой доступ к ним для получения обобщенной информации.**
4. В большинстве случаев приходится создавать временный файл, объединяющий исходные

5. Жесткая зависимость программ от данных: при **реорганизации файла** данных **приходится изменять соответствующую программу**, т.к. она написана под конкретную структуру файла.
6. **Несовместимость форматов** отдельных файлов и использование **несогласованных средств программирования** при создании приложений для работы с этими файлами.
7. **Фиксированный набор запросов**, которые изначально заложены в приложения на этапе их создания, и отсутствие универсальных средств для расширения этого набора.

# Парадоксы действительности

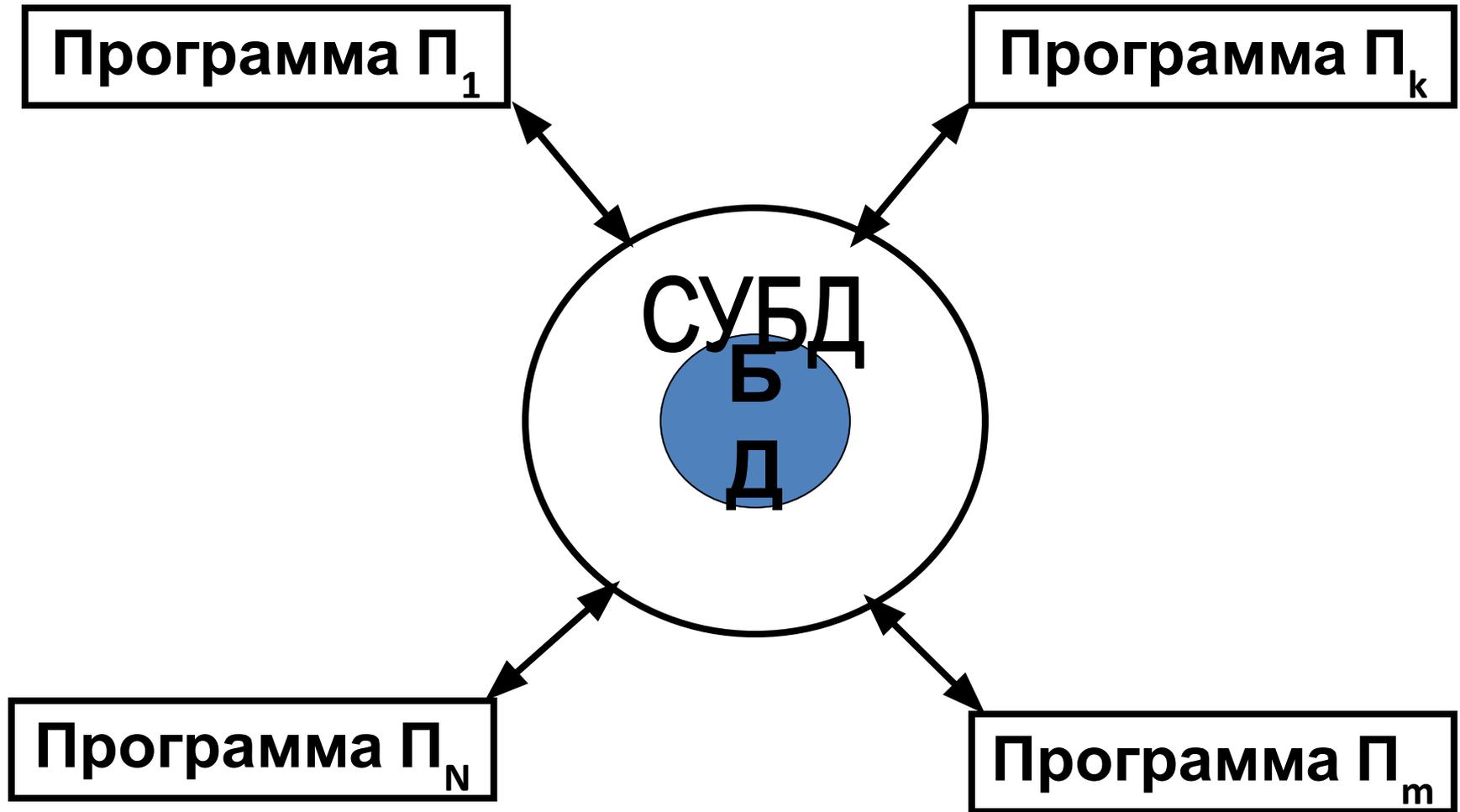
Несмотря на перечисленные недостатки, файловые системы до сих пор находят применение в некоторых организациях, обеспечивая локальную («кусочную», «островную», «лоскутную») автоматизацию их деятельности.

В этом случае можно легко расширять сферу автоматизации путем приобретения недорогих обновленных или дополнительных подсистем (либо их самостоятельной разработки).

Часто создается иллюзия постоянного прогресса (особенно с учетом необходимости разрешения нескончаемых проблем по налаживанию «мостов между островами»).

Однако, если честно посчитать все затраты по сопровождению такого «зоопарка», а также учесть многочисленные неудобства для пользователей, то быстро выясняется, что экономия средств практически отсутствует и *давно пора переходить на систему с централизованным управлением данными.*

# Централизованное управление данными



Основой централизованного управления данными является **база данных (БД)** как набор **структурированных и логически взаимосвязанных** данных, которые удовлетворяют информационные потребности большого количества пользователей.

Как единое хранилище сведений, которые одновременно используются представителями разных подразделений в составе организации (фирмы), **БД является общим корпоративным ресурсом.**

Вместо разрозненных файлов с избыточными данными, в которых могут присутствовать противоречия, здесь все данные собраны вместе, причем **каждый элемент данных имеется в единственном экземпляре.**

Принципиальным моментом является то, что в базе данных хранятся **не только рабочие данные, но и их описание.**

Элементы описания данных принято называть **метаданными** (meta-data), т.е. данными о данных.

Метаданные выступают в виде **системного каталога (словаря)** данных.

Именно эти элементы БД обеспечивают **независимость приложений от организации (структуры и формы хранения) данных.**

Для повышения эффективности работы с хранилищем данных предусматривается **система управления базой данных (СУБД).**

**СУБД — это программные средства,** которые позволяют:

- **создать БД в соответствии с имеющимся описанием ее структуры, а затем**
- **поддерживать БД в рабочем состоянии, осуществляя контролируемый доступ к данным со стороны пользователей.**

Предоставление **контролируемого доступа** к БД означает следующее:

- защита от несанкционированного доступа (разграничение полномочий и прав доступа к данным, защита паролем, шифрование данных);
- автоматическая поддержка целостности данных, что обеспечивает постоянное отсутствие противоречий между отдельными элементами данных;
- управление совместным доступом к данным при параллельной работе приложений;
- восстановление БД при нарушениях ее целостности в результате аппаратных сбоев или программных ошибок.

# Языковые средства СУБД

1. При **создании БД** используются директивы **языка определения данных** (Data Definition Language — DDL). Язык DDL позволяет:
  - **указать структурную организацию данных** (например, структуру таблиц для хранения отдельных наборов данных);
  - **задать типы данных и ограничения** на допустимые значения.

2. Инструменты **для обращения к данным** представлены директивами **языка манипулирования данными** (Data Manipulation Language → ДМЛ). Для любой СУБД работа с данными сводится к выполнению следующих операций:
- **INSERT** — добавление новых элементов данных;
  - **DELETE** — удаление существующих элементов данных, которые стали ненужными пользователям;
  - **UPDATE** — обновление значений для существующих элементов данных;
  - **SELECT (SEARCH)** — выбор (поиск) данных по заданному признаку (критерию).

Язык DML часто называют **языком запросов**, поскольку все перечисленные операции манипулирования данными выполняются с помощью запросов соответствующих типов:

- запросы на добавление;
- запросы на удаление;
- запросы на обновление;
- запросы на поиск.

Наличие такого языка снимает ограничение файловых систем, когда пользователь имеет дело только с фиксированным набором запросов, заложенных в приложение при его разработке.

Конкретной реализацией языков DDL и DML является **стандартизованный язык SQL (Structured Query Language)**, который поддерживается всеми современными СУБД. Директивы языка SQL встраиваются в современные языки программирования, что дает возможность **стандартным способом** получать доступ к БД из прикладных программ. Некоторые промышленные СУБД содержат **языки программирования на основе SQL** (например, язык PL/SQL для СУБД Oracle), что позволяет создавать полноценные процедуры обработки данных на сервере БД с использованием операторов цикла, ветвления и