

ОСНОВЫ ПОСТРОЕНИЯ БАЗ ДАННЫХ

Введение в базы.

В разделе рассматриваются :

- базы данных и информационные системы, понятия баз данных и систем управления базами данных
- даётся характеристика вариантов организации информационной системы по архитектуре клиент-сервер
- приводится классификация СУБД, и описываются основные их функции
- рассматриваются варианты создания приложений и организации взаимодействия пользователей с информационными системами.

1.1. Базы данных и информационные системы

В основе решения многих задач лежит обработка информации. Для облегчения обработки информации создаются информационные системы (ИС). Автоматизированными называют ИС, в которых применяют технические средства, в частности ЭВМ. Большинство существующих ИС являются автоматизированными, поэтому для краткости просто будем называть их ИС.

В *широком понимании* под определение ИС подпадает любая система обработки информации.

По *области применения* ИС можно разделить на системы, используемые в производстве, образовании, здравоохранении, науке, военном деле, социальной сфере, торговле и других отраслях.

По *целевой функции* ИС можно условно разделить на следующие основные категории: управляющие, информационно-справочные, поддержки принятия решений.

Банк данных является разновидностью ИС, в которой реализованы функции центрального хранения и накопления обрабатываемой информации, организованной в одну или несколько баз данных.

Банк данных (БНД) в общем случае состоит из следующих компонентов: базы (нескольких баз) данных, системы управления базами данных, словаря данных, администратора, вычислительной системы и обслуживающего персонала.

1. База данных (БД) представляет собой совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.

Логическую структуру хранимых в базе данных называют *моделью представления данных*. К основным моделям представления данных (моделям данных) относятся следующие:

- иерархическая,
- сетевая,
- реляционная,
- постреляционная,
- многомерная,
- объектно-ориентированная.

2. Система управления базами данных (СУБД) — это комплекс языковых программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями.

Обычно СУБД различают по используемой модели данных. Так, СУБД, основанные на использовании реляционной модели данных, называют реляционными СУБД.

3. Приложение представляет собой программу или комплекс программ, обеспечивающих автоматизацию обработки информации для прикладной задачи.

4. Словарь данных (СД) представляет собой подсистему БНД, предназначенную для централизованного хранения информации о структурах данных, взаимосвязи файлов БД друг с другом, типах данных и форматах их представления, принадлежности данных пользователям, кодах защиты и разграничения доступа и т. п.
5. Администратор базы данных (АБД) есть лицо или группа лиц, отвечающих за выработку требований к БД, ее проектирование, создание, эффективное использование и сопровождение. В вычислительной сети АБД, как правило, взаимодействует с *администратором сети*.

В обязанности последнего входят контроль за функционированием аппаратно-программных средств сети, реконфигурация сети, восстановление программного обеспечения после сбоев и отказов оборудования, профилактические мероприятия и обеспечение разграничения доступа

6. Вычислительная система (ВС) представляет собой совокупность взаимосвязанных и согласованно действующих ЭВМ или процессоров и других устройств, обеспечивающих автоматизацию процессов приема, обработки и выдачи информации потребителям. Поскольку основными функциями БНД являются хранение и обработка данных, то используемая ВС, наряду с приемлемой мощностью центральных процессоров(ЦП) должна иметь достаточный объем оперативной и внешней памяти прямого доступа.
7. Обслуживающий персонал выполняет функции поддержания технических и программных средств в работоспособном состоянии. Он проводит профилактические, регламентные, восстановительные и другие работы по планам, а также по мере необходимости.

1.2. Архитектура информационной системы

Эффективность функционирования информационной системы (ИС) во многом зависит от ее архитектуры.

В настоящее время перспективной является архитектура клиент-сервер. В достаточно распространенном варианте она предполагает наличие компьютерной сети и распределенной базы данных, включающей корпоративную базу данных (КБД) и персональные базы данных (ПБД).

КБД размещается на компьютере-сервере, ПБД размещаются на компьютерах сотрудников подразделений, являющихся клиентами корпоративной БД.

Сервером определенного ресурса в компьютерной сети называется компьютер (программа), управляющий этим ресурсом, клиентом компьютер (программа), использующий этот ресурс.

В качестве ресурса компьютерной сети могут выступать, к примеру, базы данных, службы печати, почтовые службы.

Например, если управляемым ресурсом является база данных, то соответствующий сервер называется *сервером* *базы данных*.

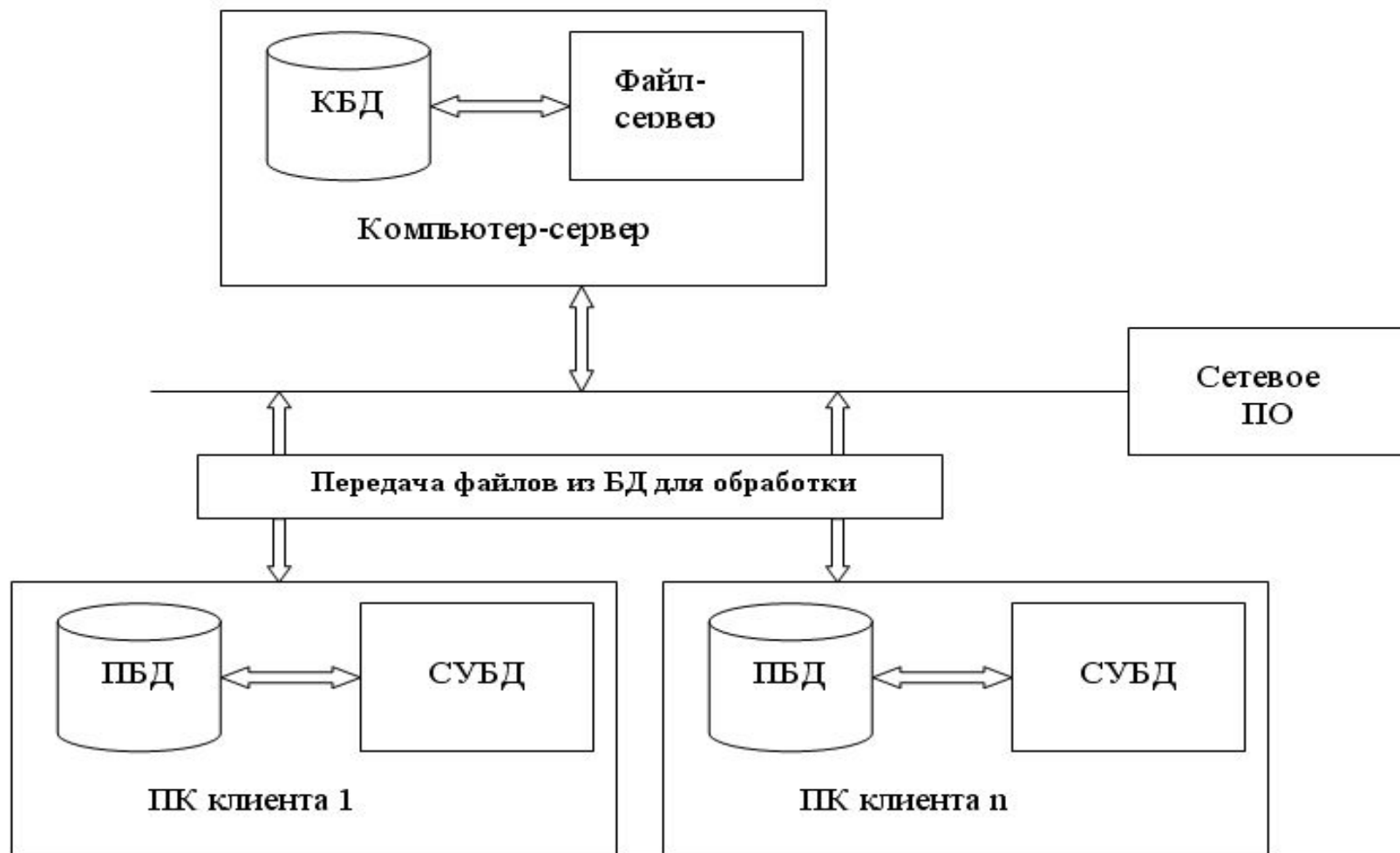


Рис.1.1. Структура ИС с файл-сервером.

Структура распределенной ИС, построенной по архитектуре *клиент-сервер* с использованием сервера баз данных, показана на рис. 1.2.

При такой архитектуре сервер базы данных обеспечивает выполнение основного объема обработки данных. Формируемые пользователем или приложением запросы поступают к серверу БД в виде инструкций языка SQL. Сервер базы данных выполняет поиск и извлечение нужных данных, которые затем передаются на компьютер пользователя.

Достоинством такого подхода в сравнении предыдущим является заметно меньший объем передаваемых данных.

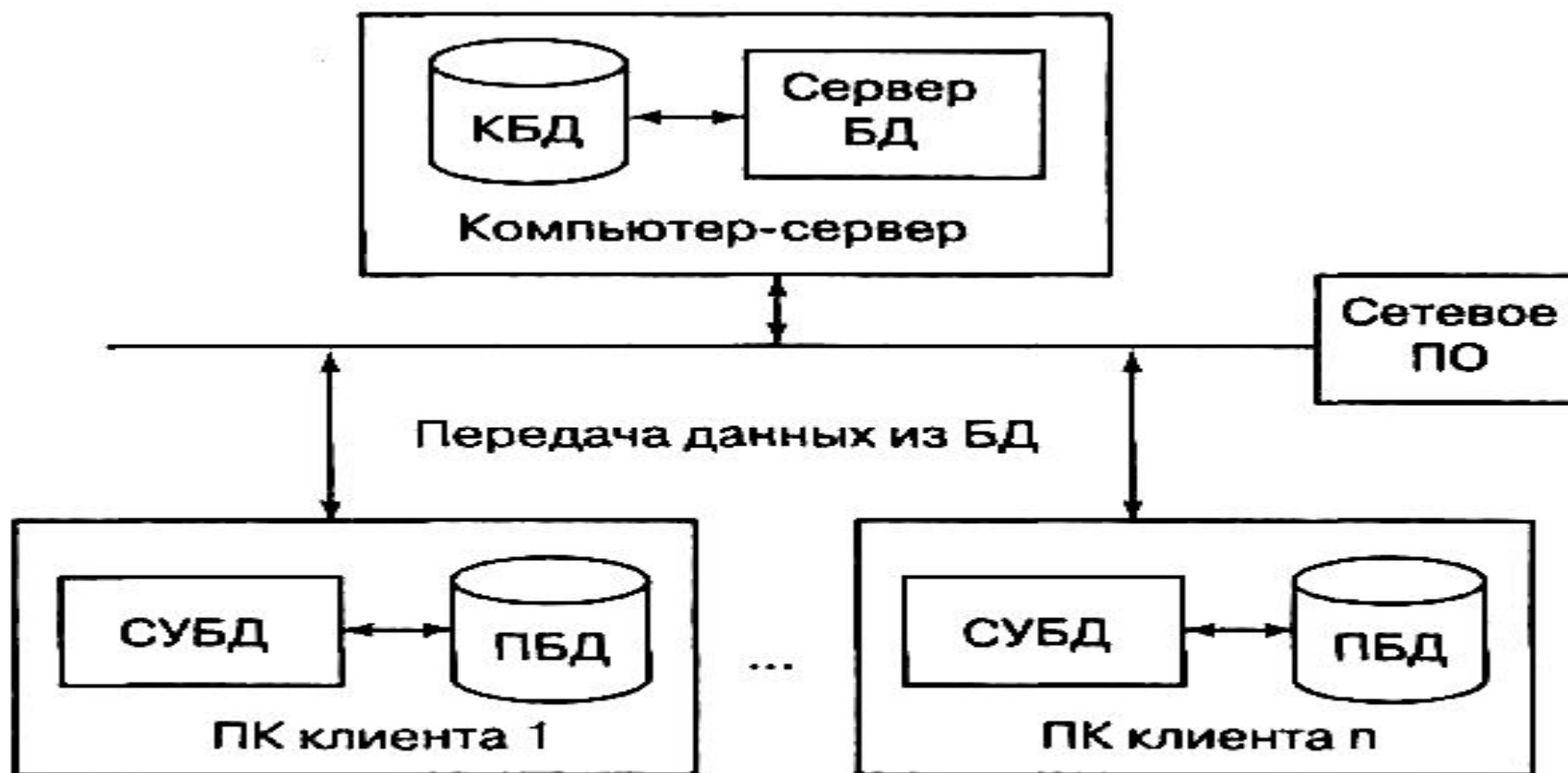


Рис.1.2. Структура ИС с сервером баз данных.

В зависимости от размеров организации и особенностей решаемых задач ИС может иметь одну из следующих конфигураций :

- компьютер-сервер, содержащий корпоративную и персональные базы ;
- компьютер-сервер и персональные компьютеры с ПБД;
- несколько компьютеров-серверов и персональных компьютеров с ПБД.

Использование архитектуры клиент-сервер дает возможность постепенного наращивания информационной системы предприятия, во-первых, по мере развития предприятия; во-вторых, по мере развития самой информационной системы.

Разделение общей БД на корпоративную БД и персональные БД позволяет уменьшить сложность проектирования БД по сравнению с нейтрализованным вариантом, а значит, снизить вероятность ошибок при проектировании и стоимость проектирования.

Важнейшим *достоинством* применения БД в информационных системах является обеспечение независимости данных от прикладных программ. Это дает возможность пользователям не заниматься проблемами представления данных на физическом уровне: размещения данных в памяти, методов доступа к ним и т. д.

Такая независимость достигается поддерживаемым СУБД многоуровневым представлением данных в БД на логическом (пользовательском) и физическом уровнях. Благодаря СУБД и наличию логического уровня представления данных обеспечивается отделение концептуальной (понятийной) модели БД от ее физического представления в памяти ЭВМ.

1.3. Системы управления базами данных

В этом подразделе приводится классификация СУБД и рассматриваются основные их функции. В качестве основных классификационных признаков можно использовать следующие:

- вид программы;
- характер использования;
- модель данных.

Названные признаки существенно влияют на целевой выбор СУБД и эффективность использования разрабатываемой информационной системы.

Классификация СУБД.

В общем случае под СУБД можно понимать любой программный продукт, поддерживающий процессы создания, ведения и использования БД. Рассмотрим, какие из имеющихся на рынке программ имеют отношение к БД и в какой мере они связаны с базами данных.

К СУБД относятся следующие основные виды программ:

- полнофункциональные СУБД;
- серверы БД;
- клиенты БД;
- средства разработки программ работы с БД.

Полнофункциональные СУБД (ПФСУБД)
представляют собой традиционные СУБД, которые сначала появились для больших машин, затем для минимашин и для ПЭВМ.

Из числа всех СУБД современные ПФСУБД являются наиболее многочисленными и мощными по своим возможностям, например: *Clarion Database Developer, Data Ease, Data Flex, dBase IV*

Обычно ПФСУБД имеют развитый интерфейс, позволяющий с помощью команд меню выполнять основные действия с БД: создавать и модифицировать структуры таблиц, вводить данные, формировать запросы, разрабатывать отчеты, выводить их на печать и т. п.

Для создания запросов и отчетов не обязательно программирование, а удобно пользоваться языком *QBE* (*Query By Example* - формулировки запросов по образцу).

Многие ПФСУБД включают средства

Серверы БД предназначены для организации центров обработки данных в сетях ЭВМ.

Эта группа БД в настоящее время менее многочисленна, но их количество постепенно растет. Серверы БД реализуют функции управления базами данных, запрашиваемые другими (клиентскими) программами обычно с помощью операторов SQL.

Примерами серверов БД являются следующие программы: *NetWare SQL (Novell)*, *MS SQL Server (Microsoft)*, *InterBase (Borland)*, *SQLBase Server (Gupta)*, *Intelligent Database (Ingress)*.

В роли клиентских программ для серверов БД в общем случае могут использоваться различные программы: ПФСУБД, электронные таблицы, текстовые процессоры, программы электронной почты и т. д. Элементы пары «клиент - сервер» могут принадлежать одному или разным производителям программного обеспечения.

Если клиентская и серверная части выполнены одной фирмой, то распределение функций между ними выполнено рационально. В остальных случаях обычно преследуется цель обеспечения доступа к данным «любой ценой».

Примером такого соединения является случай, когда одна из полнофункциональных СУБД играет роль сервера, а вторая СУБД (другого производителя) - роль клиента.

Так, для сервера БД *SQL Server (Microsoft)* в роли клиентских (фронтальных) программ могут выступать многие СУБД, такие как *dBASE IV, Blyth Software, Paradox, DataEase, Focus, 1-2-3, MDBS III, Revelation* и другие.

Средства разработки программ работы с БД могут использоваться для создания разновидностей следующих программ:

- клиентских программ;
- серверов БД и их отдельных компонентов;
- пользовательских .

Программы *первого* и *второго* вида довольно малочисленны и предназначены для системных программистов. Пакетов *третьего* вида гораздо больше, но меньше, чем полнофункциональных СУБД.

К средствам разработки пользовательских приложений относятся системы программирования, например *Clipper*, разнообразные библиотеки программ для различных языков программирования, а также пакеты автоматизации разработок (в том числе систем типа клиент-сервер).

В числе наиболее распространенных можно назвать следующие инструментальные системы: Delphi и Power Builder (Borland), Visual Basic (Microsoft), SILVERRUN (Computer Advisers Inc.),

По характеру использования СУБД делят на персональные и многопользовательские.

Персональные СУБД обычно обеспечивают возможность создания персональных БД и недорогих приложений, работающих с ними.

Персональные СУБД или разработанные с их помощью приложения зачастую могут выступать в роли клиентской части многопользовательской СУБД.

К персональным СУБД, например, относятся *Visual FoxPro, Paradox, Clipper, dBase, Access* и др.

Многопользовательские СУБД включают в себя сервер БД и клиентскую часть и, как правило, могут работать в неоднородной вычислительной среде (с разными типами ЭВМ и операционными системами).

К многопользовательским СУБД относятся, например, СУБД *Oracle* и *Informix*.

*По используемой модели данных СУБД (как и БД),
разделяют на иерархические, сетевые, реляционные,
объектно-ориентированные и другие типы.*

Некоторые СУБД могут одновременно
поддерживать несколько моделей данных.

Функции СУБД

С точки зрения пользователя, СУБД реализует функции хранения, изменения (пополнения, редактирования и удаления) и обработки информации, а также разработки и получения различных выходных документов.

Перечисленные выше функции СУБД, в свою очередь, используют следующие основные функции более низкого уровня, которые назовем *низкоуровневыми*:

- управление данными во внешней памяти;
- управление транзакциями;
- ведение журнала изменений в БД;
- обеспечение целостности и безопасности БД.

- Реализация функции управления данными во внешней памяти в разных системах может различаться и на уровне управления ресурсами (используя файловые системы ОС или непосредственное управление устройствами ПЭВМ), и по логике самих алгоритмов управления данными.

В основном методы и алгоритмы управления данными являются «внутренним делом» СУБД и прямого отношения к пользователю не имеют.

Качество реализации этой функции наиболее сильно влияет на эффективность работы специфических ИС, например, с огромными БД, со сложными запросами, большим объемом обработки данных.

Необходимость *буферизации* данных и как следствие реализации функции *управления буферами* оперативной памяти обусловлено тем, что объем оперативной памяти меньше объема внешней памяти.

Буферы представляют собой области оперативной памяти, предназначенные для ускорения обмена между внешней и оперативной памятью.

В буферах временно хранятся фрагменты БД, данные из которых предполагается использовать при обращении к СУБД или планируется записать в базу после обработки.

- Механизм транзакций используется в СУБД для поддержания целостности данных в базе.

Транзакцией называется некоторая неделимая последовательность операций над данными БД, которая отслеживается СУБД от начала и до завершения.

Если по каким-либо причинам (сбои и отказы оборудования, ошибки в программном обеспечении, включая приложение) транзакция остается незавершенной, то она отменяется.

Говорят, что транзакции присущи три основных свойства:

- *атомарность* (выполняются все входящие в транзакцию операции или ни одна);
- *сериализуемость* (отсутствует взаимное влияние выполняемых в одно и то же время транзакций);
- *долговечность* (даже крах системы не приводит к утрате результатов зафиксированной транзакции).

Примером транзакции является операция перевода денег с одного счета на другой в банковской системе. Здесь необходим, по крайней мере, двухшаговый процесс. Сначала снимают деньги с одного счета, затем добавляют их к другому счету. Если хотя бы одно из действий не выполнится успешно, результат операции окажется неверным и будет нарушен баланс между счетами.

- Ведение журнала изменений в БД (журнализация изменений) выполняется СУБД для обеспечения надежности хранения данных в базе при наличии аппаратных сбоев и отказов, а также ошибок в программном обеспечении.

Журнал СУБД — это особая БД или часть основной БД, непосредственно недоступная пользователю и используемая для записи информации обо всех изменениях базы данных. В различных СУБД в журнал могут записываться записи, соответствующие изменениям в СУБД на разных уровнях: от минимальной внутренней операции модификации страницы внешней памяти до логической операции модификации БД (например, вставки записи, удаления столбца, изменения значения в поле) и даже транзакции.

Для эффективной реализации функции ведения журнала изменений в БД необходимо обеспечить повышенную надежность хранения и поддержания в рабочем состоянии самого журнала. Иногда для этого в системе хранят несколько копий журнала.

- Целостность БД есть свойство базы данных, означающее, что в ней содержится полная, непротиворечивая и адекватно отражающая предметную область информация.

Поддержание целостности БД включает проверку целостности и ее восстановление в случае обнаружения противоречий в базе данных.

Целостное состояние БД описывается с помощью *ограничений целостности* в виде условий, которым должны удовлетворять хранимые в базе данные.

Примером таких условий может служить *ограничение диапазонов возможных значений атрибутов объектов, сведения о которых хранятся в БД, или отсутствие повторяющихся записей в таблицах реляционных БД.*

Обеспечение безопасности достигается в СУБД шифрованием прикладных программ, данных, защиты паролем, поддержкой уровней доступа к базе данных и к отдельным ее элементам (таблицам, формам, отчетам и т. д.).

1.4. Локальные информационные системы

Функциональные части информационной системы могут размещаться на одном или на нескольких компьютерах. Рассмотрим варианты организации ИС на одном ПК. Соответствующую ИС обычно называют *локальной* или *однопользовательской* (хотя последнее не совсем строго, поскольку на одном компьютере поочередно могут работать несколько пользователей).

Организация функционирования локальной ИС на одном компьютере в среде некоторой операционной системы (ОС) возможна с помощью следующих вариантов использования программных средств:

- «полной» СУБД;
- приложения и «усеченной» (ядра) СУБД;
- независимого приложения.

Первый способ обычно применяется в случаях, когда в дисковой памяти компьютера помещается вся СУБД и она часто используется для доработки приложения (рис. 1.3).

Взаимодействие пользователя с СУБД происходит напрямую через пользовательский (терминальный) интерфейс СУБД, либо с помощью приложения. Приложение выполняется в режиме интерпретации (пооператорная трансляция и последующее выполнение оттранслированного оператора).

Основное достоинство этой схемы: простота разработки и сопровождения БД и приложений при наличии развитых соответствующих средств разработки и сервисных средств.

Недостатком этой схемы являются затраты дисковой памяти на хранение программы СУБД.



Рис. 1.3. Использование приложения и СУБД

Приложение с ядром СУБД (рис. 1.4) используют для достижения следующих целей:

- повышения скорости работы приложения;
- уменьшения объема занимаемого СУБД пространства жесткого диска и оперативной памяти;
- защиты приложения от модификации со стороны пользователя (обычно ядро не содержит средств разработки приложений).

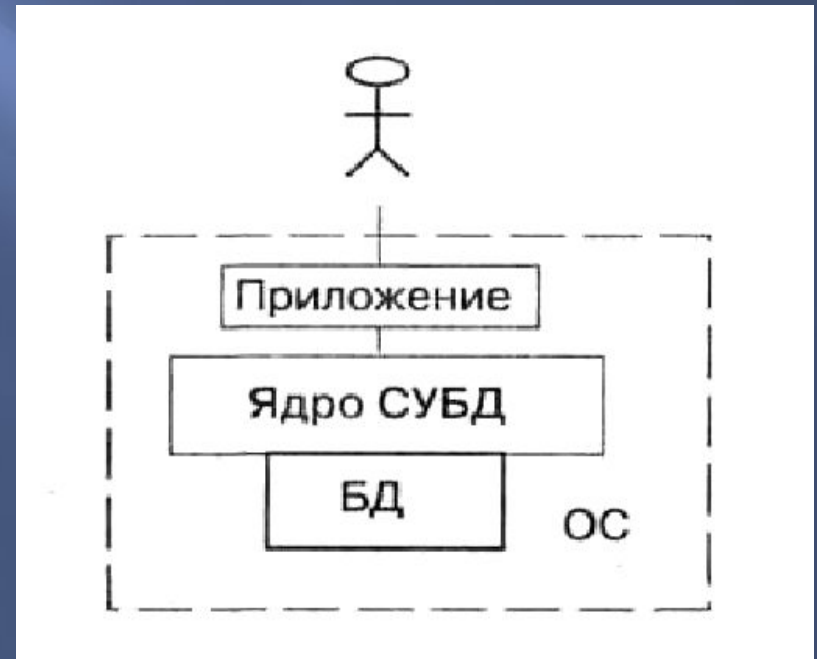


Рис. 1.3. Использование приложения и ядра СУБД

Достоинствами использования ядра СУБД по сравнению с использованием полной версии СУБД являются:

- меньшее потребление ресурсов памяти компьютера
- ускорение работы приложения и возможность защиты приложения от модификации.

К основным *недостаткам* можно отнести:

- значительный объем дисковой памяти, необходимой для хранения ядра СУБД
- недостаточно высокое быстродействие работы приложений (выполнение приложения по-прежнему происходит путем *интерпретации*).

При третьем способе организации ИС исходная программа предварительно компилируется — преобразуется в последовательность исполняемых машинных команд. В результате получается готовая к выполнению независимая программа, не требующая для своей работы ни всей СУБД ни ее ядра (рис. 1.5).

С точки зрения выполнения основных функций хранения и обработки данных такая программа мало отличается от приложения, работающего под управлением СУБД или ее ядра.



Рис.1.5. Использование независимого приложения

Основными *достоинствами* этого варианта по сравнению с двумя предыдущими является :

- экономия внешней и оперативной памяти компьютера
- ускорение выполнения приложения и полная защита приложения от модификации.

К *недостаткам* можно отнести:

- трудоемкость доработки приложения
- отсутствие возможности использовать стандартные средства СУБД по обслуживанию БД.

При выборе средств для разработки приложения следует учитывать три основных фактора:

1. ресурсы компьютера
2. особенности приложения (потребность в модификации функций программы, время на разработку, необходимость контроля доступа и поддержание целостности информации)
3. цель разработки (отчуждаемый программный продукт или система автоматизации своей повседневной деятельности)

1.5. Способы разработки и выполнения приложений

Современные СУБД позволяют решать широкий круг задач по работе с базами данных без разработки приложения. Тем не менее, есть случаи, когда целесообразно разработать приложение.

Например, если требуется автоматизация манипуляций с данными, терминальный интерфейс СУБД недостаточно развит, либо имеющиеся в СУБД стандартные функции по обработке информации не устраивают пользователя.

Для разработки приложений СУБД должна иметь программный интерфейс, основу которого составляют функции или процедуры соответствующего языка программирования.

Существующие СУБД поддерживают следующие технологии (и их комбинации) разработки приложений:

- ручное кодирование программ (*Clipper, FoxPro, Paradox*);
- создание текстов приложений с помощью генераторов (*FoxApp в FoxPro, Personal Programmer в Paradox*);
- автоматическая генерация готового приложения методами визуального программирования (*Delphi, Access, Paradox for Windows*).

1.6. Схема обмена данными при работе с БД.

Пользователю любой категории (администратору БД, разработчику приложения, обычному пользователю) для грамотного решения задач полезно представлять вычислительный процесс, происходящий в ОС при работе с БД.

Раскроем внутренние механизмы этого процесса на примере наиболее общего случая организации ИС, функционирующей на одном ПК, - когда пользователь работает с «полной» версией программы СУБД (рис. 1.3).

Варианты, представленные на рис. 1.4 и рис. 1.5, можно считать частными случаями.

При работе пользователя с базой данных над ее содержимым выполняются следующие основные операции: выбор, добавление, модификация (замена) и удаление данных.

Рассмотрим, как происходит обмен данными между отдельным пользователем и персональной СУБД при выполнении наиболее часто используемой операции *выбора данных* (обмен данными между пользователем и БД для других операций отличается несущественно).

Схематично обмен данными при работе пользователя с БД можно представить так, как показано на рис. 1.6, где обычными стрелками обозначены связи по управлению, утолщенными - связи по информации.

Цикл взаимодействия пользователя с БД с помощью приложения можно разделить на следующие основные этапы:

1. Пользователь терминала (1) в процессе диалога с приложением формулирует запрос (2) на некоторые данные из БД.

2. Приложение (3) на программном уровне средствами языка манипулирования данными формулирует запрос (4), с которым обращается к СУБД.

3. Используя свои системные управляющие блоки и таблицы, СУБД с помощью *словаря данных* определяет местоположение требуемых данных и обращается (5) за ними к ОС.

4. Программы методов доступа файловой системы ОС считывают (6) из внешней памяти искомые данные и помещают их в системные буферы СУБД.

5. Преобразуя полученные данные к требуемому формату, СУБД пересылает их (7) в соответствующую область программы и сигнализирует (8) о завершении операции каким-либо образом (например, кодом возврата).

6. Результаты выбора данных из базы приложение (3) отображает (9) на терминале пользователя (1).

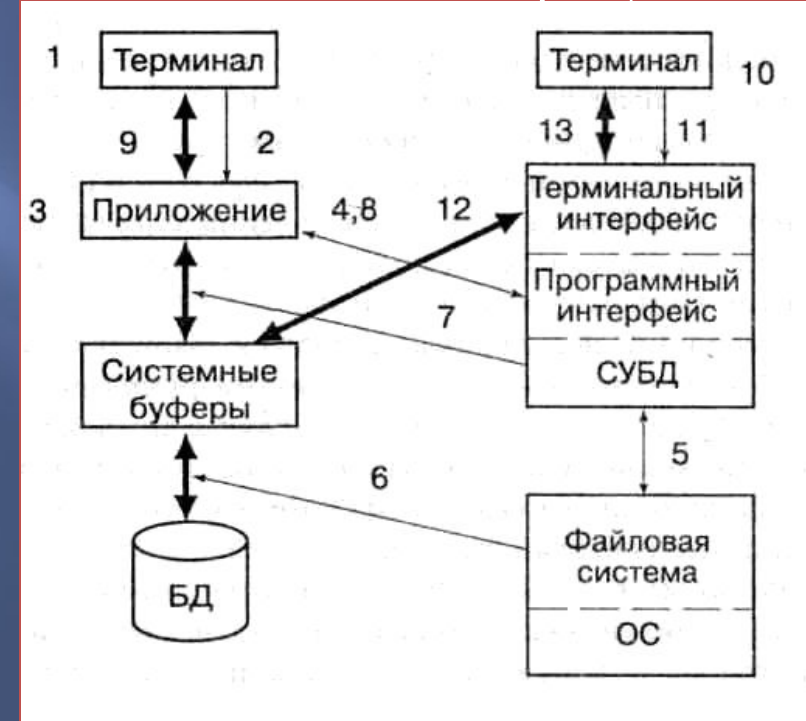


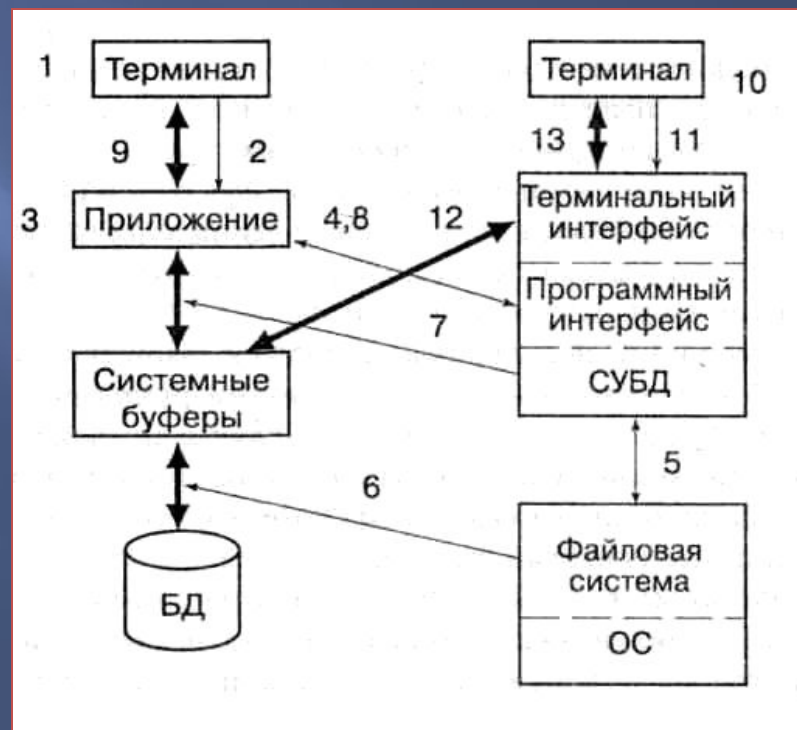
Рис.1.6.Схема обмена данными при работе с БД.

В случае работы пользователя в диалоговом режиме с СУБД (без приложения) цикл взаимодействия пользователя с БД упрощается. Его можно представить следующими этапами:

1. Пользователь терминала (10) формулирует на языке запросов СУБД, например QBE, по связи (11) требование на выборку некоторых данных из базы.

2. СУБД определяет местоположение требуемых данных и обращается (5) за ними к ОС, которая считывает (6) из внешней памяти искомые данные и помещает их в системные буферы СУБД.

3. Информация из системных буферов преобразуется (12) к требуемому формату, после чего отображается (13) на терминале пользователя (10).



На персональных ЭВМ пользователь обычно работает один, но с различными программами, в том числе и одновременно (точнее, попеременно). Иногда такими программами оказываются СУБД: различные программы или разные копии одной и той же СУБД. Последняя ситуация возникает, например, при работе с различными базами данных с помощью СУБД Access.

Технология одновременной работы пользователя с несколькими программами неплохо реализована в Windows. Здесь каждая выполняемая программа имеет свое окно взаимодействия с пользователем и имеются удобные средства переключения между программами. При работе в Windows СУБД избавлена от необходимости поддержания нескольких сеансов работы с пользователями.