

Архитектура информационной системы

1. Виды архитектуры информационной системы
2. Локальные информационные системы
3. Файл-серверная архитектура
4. Клиент-серверная архитектура
5. Трехуровневая архитектура

Виды архитектуры информационной системы

Любая информационная система (ИС) включает в себя три компонента:

- Управление данными;
- Бизнес-логику;
- Пользовательский интерфейс.

Данные хранятся в базах данных, а управление ими осуществляется с помощью системы управления базами данных (СУБД). Бизнес-логика определяет правила, по которым обрабатываются данные. Она реализуется набором процедур, написанных на различных языках программирования. Пользователь работает с интерфейсом, где логика работы ИС представлена в виде элементов управления – полей, кнопок, списков, таблиц и т.д.

Однако, эти три компонента в разных ИС взаимодействуют друг с другом различными способами.

Определение 1: *Архитектурой информационной системы называется концепция, согласно которой взаимодействуют компоненты информационной системы.*

Существуют следующие виды архитектур ИС:

- Локальная;
- Файл-серверная;
- Клиент-серверная;
- Трехслойная.

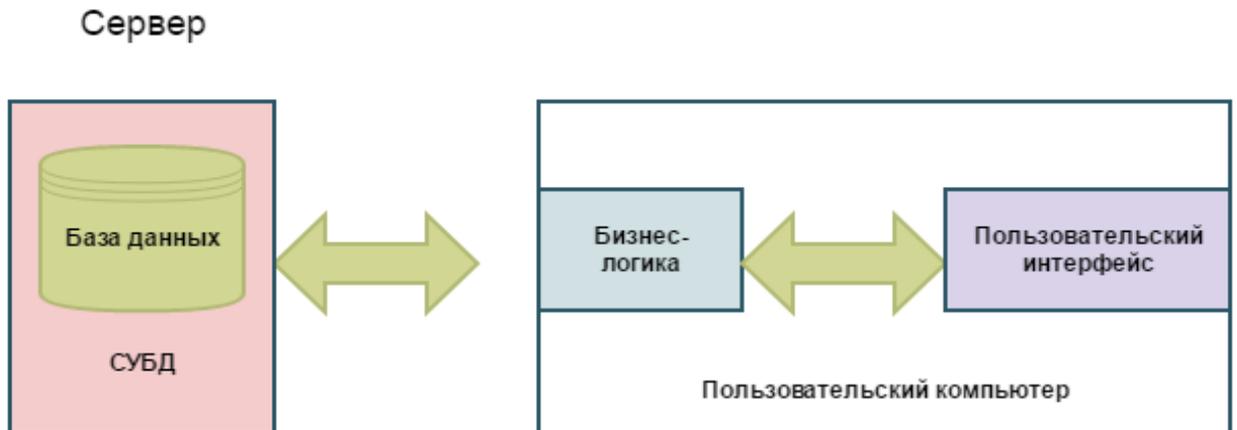
Локальные информационные системы

Локальные информационные системы широко использовались до появления компьютерных сетей. В этом случае все компоненты ИС располагаются на одном компьютере. Очевидным недостатком этой архитектуры является возможность работать в ИС только одному пользователю. Другие пользователи не имеют возможности получить доступ к данным даже для чтения.

Файл-серверная архитектура

С появлением компьютерных сетей возникла возможность хранить данные в файлах на выделенном специально для этой цели компьютере. Такой компьютер называется файловым сервером или просто сервером. Компьютеры пользователей соединены с сервером сетью, поэтому доступ к данным, могут получить несколько пользователей одновременно. Однако, кроме функции хранения данных и обеспечения доступа к ним, сервер никаких функций не выполняет. Приложения, обрабатывающие данные,

находятся на пользовательских компьютерах.



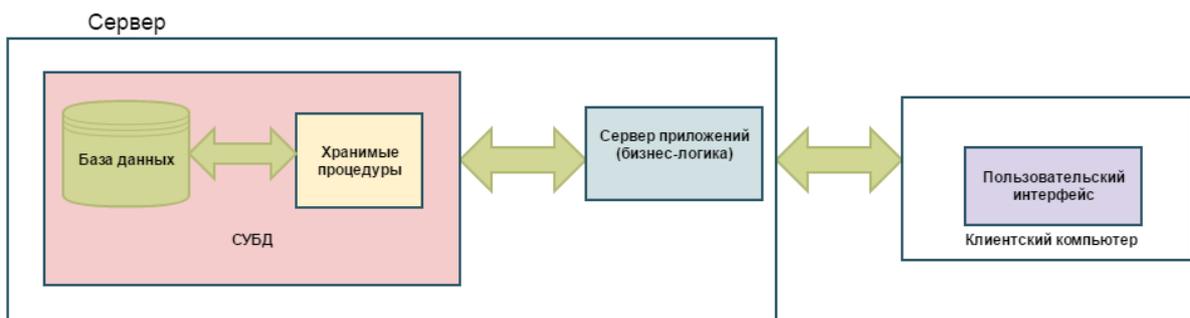
Пример 1: предположим, что в базе данных на сервере хранится список сотрудников крупного предприятия. На предприятии 1500 сотрудников и 10 подразделений. Пользователю нужно получить число сотрудников, работающих в каждом подразделении. Для решения этой задачи пользователь должен запросить данные всех 1500 сотрудников с сервера по сети, после чего на пользовательском компьютере выполнится процедура, которая осуществит подсчет сотрудников в каждом подразделении. Результатом процедуры будет 10 строк. Таким образом, чтобы получить 10 строк придется передать по сети 1500 строк.

Обработка данных на пользовательском компьютере всегда сопровождается передачей по сети большого количества «лишней» информации. Основными недостатками файл-серверной архитектуры являются:

- высокая загруженность сети и, как следствие, низкая скорость работы;
- сложность поддержания непротиворечивости данных, из-за их несогласованной обработки разными пользователями.

Клиент-серверная архитектура

До определенного момента на СУБД возлагались лишь задачи хранения данных и организации доступа к ним. С развитием технологий в состав СУБД разработчики стали включать новый компонент – процедурный язык программирования. С его помощью в СУБД стало возможным создавать процедуры для обработки данных, которые можно вызывать повторно. Такие процедуры называются **хранимыми процедурами**. Наличие хранимых процедур дало возможность осуществлять некоторую часть обработки данных на сервере.



Пример 2: рассмотрим задачу из примера 1 в условиях клиент-серверной архитектуры. Пользователь отправит на сервер запрос, который запустит процедуру. Процедура

выполнится непосредственно на сервере. Она подсчитает количество сотрудников в каждом подразделении и отправит полученные 10 строк по сети на клиентский компьютер. Таким образом, произойдет существенная экономия трафика: вместо 1500 строк будет передано по сети всего 10.

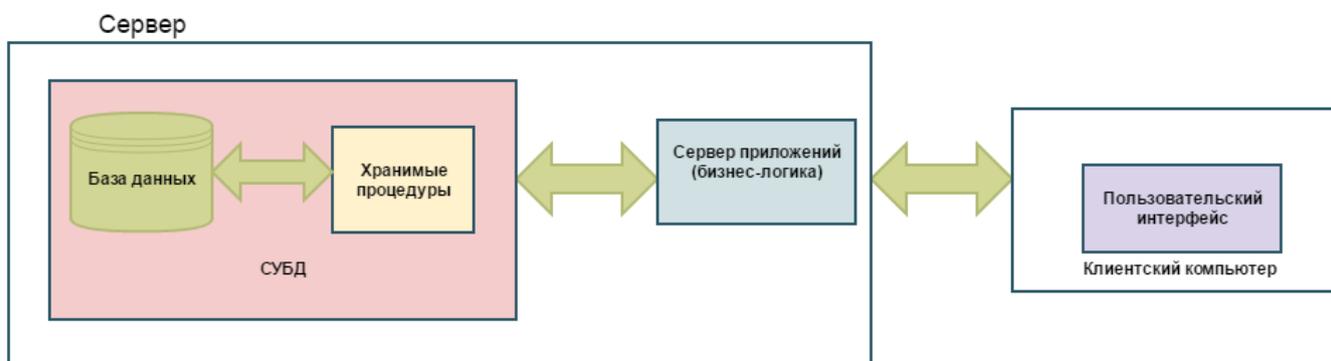
Клиент-серверная архитектура позволяет разгрузить сеть и поддерживать непротиворечивость данных за счет их централизованной обработки. Однако, языки хранимых процедур не приспособлены для полноценной реализации бизнес-логики. Поэтому бизнес-логика в клиент-серверных ИС по-прежнему реализуется на клиентских компьютерах. Такой подход имеет следующие недостатки:

- любые изменения в бизнес-логике требуют обновления на клиентском компьютере;
- клиентские компьютеры должны быть достаточно производительными;
- слабая защита данных от взломов.

Трехуровневая архитектура

Все недостатки клиент-серверной архитектуры связаны с тем, что на клиентском компьютере лежит слишком большая нагрузка, которую можно было бы перенести на сервер. Поэтому дальнейшее развитие технологий двигалось в направлении переноса нагрузки с клиентских компьютеров на сервер. В дополнение к хранимым процедурам разработчики стали использовать серверные языки программирования. Это дало возможность создавать в ИС промежуточный уровень - сервер приложений.

Определение 2: *Сервер приложений – это комплекс программ, выполняемых на сервере и реализующих бизнес-логику ИС.*



Использование сервера приложений позволяет максимально разгрузить клиентские компьютеры и сделать обработку данных еще более централизованной, что повышает скорость и надежность ИС.