

## ТЕХНОЛОГИИ ФИЗИЧЕСКОГО УРОВНЯ ПЕРЕДАЧИ ДАННЫХ

### Занятие №22

#### Организация доступа к данным по сети

1. Классификация сетей по принципу передачи данных и по типу коммуникационной среды, по способу доступа к данным
2. Файл-сервер
3. Клиент-сервер
4. DBS-модель
5. Сервер приложений
6. Вопросы

#### **Классификация сетей по принципу передачи данных, по типу коммуникационной среды, по способу доступа к данным**

*По принципу организации передачи данных* сети можно разделить на:

1. **Последовательные** — передача данных выполняется последовательно от одного узла к другому, и каждый узел транслирует принятые данные дальше. К этому типу относятся все глобальные, региональные и многие локальные сети;
2. **Широковещательные** — в каждый момент времени передачу ведет только один узел, остальные узлы только принимают информацию. К этому типу сетей относится значительная часть ЛВС, использующая один общий канал связи (моноканал) или одно общее пассивное коммутирующее устройство.

*По типу коммуникационной среды* сети можно разделить на:

1. **Сети с моноканалом** — данные могут следовать только по одному пути. Все пакеты доступны всем абонентам сети, но использовать пакет может только абонент, чей адрес указан в пакете. Такие сети называют также сетями с селекцией информации.
2. **Сети с маршрутизацией информации** — в процессе передачи данных в каждом узле происходит выбор пути дальнейшего движения.

*По способу доступа к данным* различают:

1. **Данные с локальным доступом.**
2. **Данные с удаленным (сетевым) доступом.**

Для систем с удаленным доступом используют различные модели построения сетей:

1. *файл-сервер*;
2. *клиент-сервер* (двухуровневая модель);
3. *сервер приложений* (трехуровневая модель).

#### **Файл-сервер**

При реализации модели *файл-сервер* выделяют центральную вычислительную машину (*сервер файлов*), на которой хранятся совместно используемые данные (база данных, БД). Все другие машины сети являются рабочими станциями (РС), с помощью которых осуществляют доступ пользователей ИД. Файлы БД в соответствии с пользовательскими запросами на РС, где и производится обработка данных (см. рис.1).



Рис.1. Обработка информации в модели файл-сервер

На РС находятся прикладные программы, программы-презентации и программы управления информационными ресурсами. Файл-сервер содержит файлы, необходимые для работы приложений и системы управления БД (СУБД) и поддерживает доступ к файлам. СУБД рабочей станции посылает запросы файл-серверу. Запрос осуществляется в командах языка манипулирования данными (ЯМД). СУБД переводит этот запрос в последовательность файловых команд. Каждая файловая команда вызывает перекачку блока данных. РС анализирует перекаченную часть, и если в полученном блоке не содержится ответ на запрос, то СУБД запрашивает следующий блок и т.д.

При большой интенсивности доступа к одним и тем же данным производительность файл-сервера падает. Скорость обработки данных файл-серверными системами ограничена транспортными возможностями сети.

Модель файл-сервер имеет и другие **недостатки**:

1. На каждой рабочей станции должна находиться полная копия СУБД;
2. Управление данными усложняется, поскольку доступ к одним и тем же файлам могут осуществлять сразу несколько СУБД клиентов;
3. Манипулирование данными осуществляется только файловыми командами;
4. Защита данных осуществляется только на уровне файловой системы.

Основное **достоинство** этой модели заключается в разделении системы на два взаимодействующих процесса. При этом сервер может обслуживать множество клиентов, обращающихся к нему с запросами.

### Клиент-сервер

При реализации модели *клиент—сервер* центральный компьютер сети (сервер БД) не только хранит данные, но и осуществляет выполнение основного объема их обработки (см. рис.2).



Рис.2. Обработка данных в двухуровневой системе «несколько клиентов — один сервер»

За запросом на данные от клиента следует поиск и извлечение данных на сервере, после чего эти данные (*не файлы ИД!*) передаются по сети на РС.

Спецификой модели клиент-сервер является использование языка запросов *SQL* (англ. *Structured Query Language* — язык структурированных запросов). При построении возможен ряд модификаций, в зависимости от того, какие уровни информационной системы (ИС) входят в состав клиента, а какие - сервера. Возможные варианты изображены на рис.3.

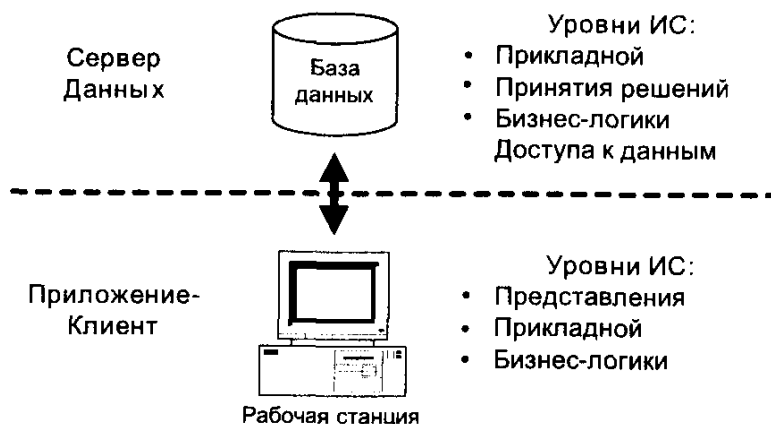


Рис.3. Уровни ИС для систем «клиент-сервер»

Самая первая клиент-серверная система получила название *модель доступа к удаленным данным* (англ. *Remote Data Access, RDA*). Ее также называют клиент-серверной архитектурой с «толстым» клиентом. В такой системе на сервере хранятся БД и ядро СУБД. На клиенте располагаются приложения, обеспечивающие ввод и отображение данных и прикладные программы. Клиент обращается к серверу с запросами на языке SQL.

Эта система по сравнению с моделью файл-сервер имеет **преимущества**:

1. Взаимодействие клиента и сервера осуществляется на языке SQL;
2. Сервер принимает и обрабатывает запросы, проверяет полномочия пользователей, гарантирует соблюдение целостности БД, выполняет обновление БД;
3. SQL — запросы имеют меньший размер, по сравнению с файловыми командами, поэтому загрузка сети уменьшается.

**Недостатки** такой системы:

- Презентационные и прикладные приложения должны дублироваться на каждом клиенте;
- Управление выполняется клиентом, а сервер играет пассивную роль.

### **DBS-модель**

Существует клиент-серверная архитектура с «тонким» клиентом. Она получила название *модель сервера базы данных* (англ. *DataBase Server, DBS*). В этой модели клиент осуществляет только функцию отображения информации. В основе системы лежат механизмы хранимых процедур и триггеров.

**Механизм хранимых процедур** позволяет создавать подпрограммы, работающие на сервере и управляющие его процессами. Клиент обращается к серверу с командой запуска хранимой процедуры, а сервер выполняет эту процедуру и регистрирует все изменения в БД. Сервер возвращает клиенту затребованные данные. Очевидно, что трафик обмена между клиентом и сервером в такой модели уменьшается.

Централизованный контроль целостности БД выполняется с помощью механизма триггеров, которые являются частью БД.

*Триггер* — это особый тип хранимой процедуры, реагирующий на возникновение определенного события в БД. Он активируется после выполнения операции добавления, редактирования или удаления. Если триггер вызывает ошибку в запросе, обновление данных не производится, а в приложение, выполняющее это действие, возвращается сообщение об ошибке.

В DBS-модели сервер является активным, т.к. не только клиент, но и сам сервер, используя механизм триггеров, может быть инициатором обработки данных в БД. Функции клиента облегчены переносом части прикладных задач на сервер, поэтому клиент и называется «тонким».

*Достоинством* модели является то, что процедуры триггеры, хранящиеся в БД, могут быть использованы несколькими клиентами, что уменьшает дублирование алгоритмом работы данных.

Главным *недостатком* данной модели является большая нагрузка сервера.

По числу используемых ВМ на серверном и клиентском уровнях двухуровневые системы можно разделить на:

1. «Один клиент — один сервер» (см. рис.4);
2. «Несколько клиентов — один сервер» (см. рис.2);
3. «Несколько клиентов — несколько серверов» (см.рис.5).



Рис. 4. Двухуровневая система «один клиент — один сервер»

В модели «один клиент — один сервер» сервер обслуживает запросы только одного клиента и для обслуживания нескольких клиентов требуется запустить соответствующее число серверов.

Модель «несколько клиентов — один сервер» позволяет обрабатывать запросы от многих клиентов.

В модели «несколько клиентов — несколько серверов» клиенты подключаются не к реальному серверу, а к промежуточному звену (диспетчеру), которое выполняет обработку запросов к реальным серверам.



Рис.5. Двухуровневая система «несколько клиентов несколько серверов»

### Сервер приложений

Система «сервер приложений» является трехуровневой моделью (рис.6). В ней вводится дополнительный промежуточный уровень между клиентом и сервером. На клиенте располагаются приложения ввода и отображения данных, локальные редакторы, коммуникационные средства, обеспечивающие доступ в вычислительную сеть.

Серверы БД в этой модели выполняют функции управления БД: создание и ведение БД, поддержка целостности БД, резервное копирование БД, восстановление БД после сбоев.

Промежуточный уровень может содержать несколько серверов приложений. Он отвечает за обработку извлеченных из БД данных, поддержку сетевой доменной среды, каталогов с данными, обеспечивает обмен сообщениями и т.д.



Рис.6. Трехуровневая система «сервер приложений»

Трехуровневая модель обладает большей гибкостью, лучшей масштабируемостью системы.

### Вопросы

1. Перечислите возможные модели доступа к данным.
2. Дайте характеристику файл-серверной модели.
3. В чем отличие клиент-серверных моделей с «толстым» и «тонким» клиентом?
4. Опишите трехуровневую модель «сервер приложений».