

ИНФОРМАЦИОННЫЕ СИСТЕМЫ В СЕТЯХ INTERNET И INTRANET

1. Организация обмена информацией в сети **Internet**.
2. Базы данных в сети **Internet** и корпоративной сети **intranet**.

ВВЕДЕНИЕ

Обработка информации в сети **Internet** существенно отличается от обработки информации в локальной сети и, тем более, на отдельном компьютере. Основные отличительные особенности состоят в следующем.

- Большая протяженность коммуникационных линий не позволяет выполнять пересылку больших объемов данных в реальном масштабе времени.

□ Взаимодействие **распределенных** элементов информационной системы (ИС) происходит с помощью обмена **пакетами** или **сообщениями**. Отдельные программные компоненты ИС могут быть одного или различных производителей. В последнем случае особую роль приобретает решение проблемы **поддержки стандартов на сетевые протоколы и на язык SQL**.

□ Сеть **Internet** по масштабам больше всех других сетей, и принципы ее организации оказывают существенное влияние и на использование БД.

На данной лекции мы рассмотрим модели доступа к БД на стороне сервера и на стороне клиента с помощью специально разработанного для сети **Internet** инструментария. В частности, ознакомимся с **общим шлюзовым интерфейсом (CGI)** и языком создания апплетов **Java**.

1. Организация обмена информацией в сети Internet

Основными видами услуг (сервиса), предоставляемых пользователям при подключении к сети Internet, являются:

- электронная почта (E-mail);
- телеконференции (UseNet);
- система эмуляции удаленных терминалов (TelNet);
- поиск и передача двоичных файлов (FTP);

- поиск и передача текстовых файлов с помощью системы меню (Gopher);
- поиск и передача документов с помощью гипертекстовых ссылок (W W W).

Все эти услуги характеризуются своими возможностями и различием в организации протоколов обмена информацией. Под **протоколом**, в общем случае, понимается **набор инструкций**, регламентирующих работу взаимосвязанных систем или объектов в сети.

Электронная почта

Электронная почта – наиболее простой и доступный способ доступа в сети Internet. Она позволяет выполнять пересылку любых типов файлов (включая тексты, изображения, звуковые файлы) по адресам электронной почты в любую точку планеты за короткий промежуток времени в любое время суток. Для передачи сообщения необходимо знать электронный адрес получателя.

Работа электронной почты основана на последовательной передаче информации по сети от одного почтового сервера к другому, пока сообщение не достигнет адресата. К *достоинствам* электронной почты относятся *высокая оперативность и низкая стоимость*. *Недостаток* электронной почты состоит в *ограниченности объема пересылаемых файлов*.

Система телеконференций

Система **UserNet** разработана как система обмена текстовой информацией. Она позволяет всем пользователям сети **Internet** участвовать в групповых дискуссиях, называемых телеконференциями, в которых обсуждаются всевозможные проблемы. В настоящее время телеконференции позволяют передавать файлы любых типов.

Протоколы *TelNet* и *FTP*

TelNet – это протокол, позволяющий одному компьютеру использовать ресурсы другого (удаленного) компьютера, то есть это протокол удаленного терминального доступа в сети.

FTP (*File Transfer Protocol*) – это протокол, позволяющий передавать файлы произвольного формата между двумя компьютерами сети.

Программное обеспечение FTP разработано по архитектуре «клиент-сервер» и разделено на две части: серверную (FTP-сервер) и клиентскую (FTP-клиент). FTP-клиент в общем случае позволяет пользователям просматривать файловую систему FTP-сервера и производить с ней обмен файлами любого типа, в том числе, исполняемыми программами. **Недостатком** этого протокола является необходимость априорного знания местоположения разыскиваемой информации (FTP-адреса).

Протокол *Gopher*

Протокол *Gopher* и реализующее его ПО предоставляют пользователям возможность работы с информационными ресурсами, не зная заранее их местонахождения. Для начала работы по этому протоколу достаточно знать адрес одного *Gopher-сервера*. В дальнейшем работа заключается в выборе команд, представленных в виде простых и понятных меню, при этом пункты меню одного сервера могут содержать ссылки на меню других серверов.

Гипертекст

W W W (*World Wide Web* – всемирная паутина) представляет собой самое популярное и современное средство организации сетевых ресурсов и строится на основе гипертекстового представления информации.

Гипертекстовый документ - это текст, содержащий ссылки на другие фрагменты текстов произвольных документов, в том числе и этого документа.

Гипертекстовый документ подготавливается на стандартизованном языке **HTML** (*HiperText Markup Language* – *разметки гипертекста*). Он состоит из страниц (web-страниц), доступ к которым основан на протоколе передачи гипертекста (*HiperText Transfer Protocol, HTTP*). **HTML-документ** представляет собой **ASCII-файл**, доступный для просмотра и редактирования в любом редакторе текстов.

В отличие от обычного текстового файла в нем присутствуют специальные команды – *тэги*, которые указывают правила форматирования документа. С помощью тэгов описываются различные элементы документа: *заголовки*, *абзацы* (параграфы), *списки*, *ссылки*, *формы* и т.д. Простейшим примером гипертекста является книга, оглавление которой содержит *внутренние* ссылки в виде номеров страниц на разделы и т.п.

Кроме того, в книге могут содержаться *внешние* ссылки на другие используемые источники информации. Фрагмент документа может включать в себя информацию в виде обычного текста, графического изображения, звука и движущегося изображения (анимации). Гипертекст с нетекстовыми документами часто называют *гипермедиа*.

Протокол TCP / IP

Работа с сети **Internet** основана на использовании протокола **TCP / IP** (*Transmission Control Protocol / Internet Protocol – Протокол управления передачей данных/Протокол Internet*), который используется для передачи данных в глобальной сети и во многих локальных сетях. Протокол **TCP / IP** в основном реализует функции транспортного и сетевого уровней эталонной модели архитектуры открытых систем (**OSI**).

Этот протокол представляет собой семейство *коммуникационных* протоколов, которые по назначению можно разделить на следующие группы:

- ▣ ***транспортные протоколы***, служащие для управления передачей данных между двумя компьютерами;
- ▣ ***протоколы маршрутизации***, обрабатывающие адресацию данных и определяющие кратчайшие доступные пути к адресату;

- ▣ **протоколы поддержки сетевого адреса**, предназначенные для идентификации компьютера по его уникальному номеру или имени;
- ▣ **прикладные протоколы**, обеспечивающие получение доступа к всевозможным сетевым услугам;
- ▣ **шлюзовые протоколы**, помогающие передавать по сети сообщения о маршрутизации и информацию о состоянии сети, а также обрабатывать данные для ЛС.

Существуют и другие протоколы, которые обеспечивают клиенту удобство работы в сети. Доступ пользователей к ресурсам Internet обычно производится с помощью программ-навигаторов (браузеров): **Netscape**, **Internet Explorer**, **Mozilla**, **Opera** и др. Хотя эти программы основаны на использовании протокола **HTTP**, они предоставляют простой доступ к другим **сервисам Internet**: электронной почте, новостям и т.д.

2. Базы данных в сети **Internet** и корпоративной сети **intranet**

Самым распространенным и доступным ресурсом для извлечения информации в сети **Internet** являются так называемые *машины поиска* (сетевые роботы – *спайдеры* и *индексы*). Главной задачей машин поиска, по сути, является индексация ресурсов сети, а также поддержка и расширение соответствующих баз данных.

Фактически в **базах данных** машин поиска хранится информация о том, где и что лежит в сети. Поэтому можно считать, что существующие машины поиска обеспечивают низкоуровневый сервис для клиентских поисковых программ более высокого уровня.

Технология **intranet** по существу представляет собой технологию **Internet**, перенесенную в среду **корпоративных информационных систем**.

Архитектура ИС в **Internet** и **intranet** является результатом эволюционного перехода от первых многопользовательских вычислительных систем (*мэйнфреймов*) через системы *клиент-сервер* к *распределенным системам с централизованной обработкой* и подготовкой информации к непосредственному потреблению. Для понимания дальнейшего рассмотрим кратко три основных этапа указанной эволюции.

Мэйнфреймы

В мэйнфреймах вычислительные ресурсы, хранимые данные и программы обработки информации сконцентрированы в одной ЭВМ. Основным средством доступа был алфавитно-цифровой терминал (клавиатура и дисплей), управляемый ЭВМ. Вся обработка информации и подготовка ее к выдаче выполнялась на центральной ЭВМ.

С терминалов, как правило, в машину передавались коды нажатия клавиш или содержимое буфера экрана, а обратно на терминал пересылались отображаемые экраны с соответствующими кодами управления отображением.

Обработка, подготовка к выдаче и управление терминалом

МЭЙНФРЕЙМ

Соединительные кабели

Отображение информации

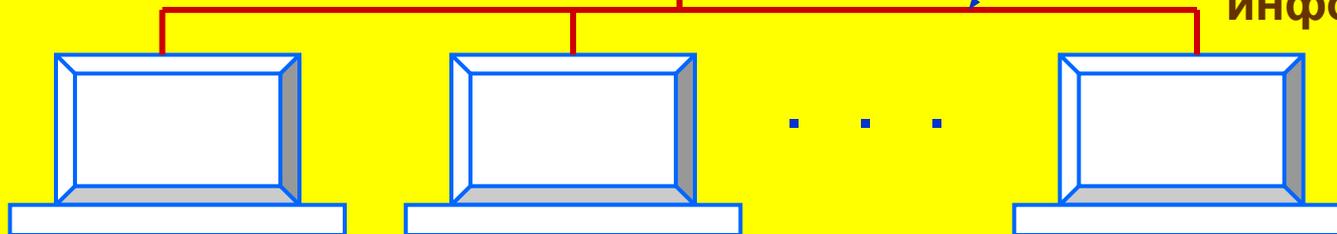


Рисунок 1 – Централизованная многопользовательская система

Достоинством системы является простота администрирования, защиты информации и модификации системы. К **недостаткам** системы можно отнести высокую загрузку процессоров и линий связи (как следствие – невысокую реакцию при большом количестве пользователей), низкую надежность (выход из строя ЭВМ приводит к полному отказу всей системы), сложность масштабирования системы и др.

Архитектура клиент-сервер

Мы эту архитектуру уже подробно рассмотрели на предыдущей лекции. Добавим лишь то, что в этих системах место терминала заняла ПЭВМ, а роль мейнфрейма — компьютер-сервер. **Достоинства этой системы:** высокая живучесть и надежность, легкость масштабирования, качественный пользовательский интерфейс и др.

Недостатки: ориентация на данные, а не на информацию; сложность переноса таких систем на другие компьютерные платформы и интеграцию с другими пакетами из-за «**закрытости**» используемых протоколов взаимодействия компонентов систем. Еще один недостаток заключается в сложности администрирования системы и ее уязвимости при непредсказуемых или злонамеренных действиях пользователя или компьютерных вирусов.

Корпоративные системы *intranet*

Корпоративные системы *intranet* в отличие от систем клиент-сервер ориентированы не на данные, а на информацию в ее окончательном и пригодном для использования неквалифицированными пользователями виде. Эти системы объединяют в себе преимущества централизованных многопользовательских систем и систем типа клиент-сервер.



Рисунок 2 – Системы, поставляющие информацию

Им присущи следующие черты:

- на сервере порождается информация, а не данные (напр., в случае СУБД – кортежи БД);
- при обмене между клиентской и серверной частями используется протокол открытого стандарта, а не какой-то конкретной фирмы;
- прикладная система находится на сервере, и поэтому для работы пользователя на компьютере-клиенте достаточно иметь программу-навигатор.

Взаимодействие компонентов WWW и традиционных СУБД

Когда источником информации в сетях Internet и intranet являются БД, возможны два варианта функционирования ПО WWW по доступу к БД:

- на стороне Web-сервера;
- на стороне Web-клиента.

Схематически это взаимодействие можно изобразить следующим образом.



а) доступ к БД на стороне Web-сервера



б) доступ к БД на стороне Web-клиента

Рисунок 3 – Модели доступа к БД в сетях Internet и intranet

Интерфейс CGI

В модели доступа к БД на стороне сервера обращение к серверу БД обычно производится путем вызова программами Web-сервера внешних по отношению к ним программ в соответствии с соглашениями одного из интерфейсов: CGI (Common Gateway Interface – общий шлюзовый интерфейс), FastCGI или API (Application Program Interface – интерфейс прикладного программирования).

Внешние программы взаимодействуют с сервером БД на языке **SQL**, непосредственно обращаясь к конкретному серверу или используя драйвер **ODBC**. Внешние программы пишутся на обычных языках программирования типа **Си**, **Си++**, **Паскаль**, или на специализированных языках типа **Perl** или **PHP**. Программы, разработанные в соответствии с интерфейсом **CGI**, называются **CGI-сценариями** или **CGI-скриптами**.

Для поддержки этого механизма на стороне клиента в языке HTML имеется средство включения в документ форм представления запросов к БД.

Процедура доступа к БД с использованием интерфейса CGI включает в себя следующие этапы:

1. Запрос Web-клиентом у Web-сервера страницы, содержащей форму обращения к БД, если при просмотре документа пользователем Web-клиент встречает ссылку на такую страницу.

2. Заполнение **Web-клиентом** содержащейся на полученной странице формы запроса к БД и отправка ее **Web-серверу**.

Правильность заполнения формы можно контролировать с помощью несложной программы, непосредственно находящейся в области **HTML-страницы**, в которой описана форма (обычно для этого используют языки **VBScript** или **JavaScript**).

3. **Web-сервер**, получив эту форму, запускает соответствующую внешнюю **CGI-программу**, передавая ей параметры.
4. **Внешняя программа** преобразует описанный в форме запрос к БД в соответствующий текст запроса на **языке SQL**, с которым обращается к серверу БД.
5. После получения результатов запроса внешняя программа формирует требуемую **HTML-страницу**, передает ее **Web-серверу** и завершает свое выполнение.
6. **Web-сервер** передает сформированную **HTML-страницу Web-клиенту**.

Достоинства интерфейса CGI

- *независимость от языка программирования;*
- *процессная независимость*, то есть сценарий выполняется на сервере как отдельный процесс, не имеющий доступа к защищенной системной информации сервера;
- *широкая распространенность*, так как CGI-стандарт применим на каждом Web-сервере;
- *независимость от архитектуры сервера.*

Недостатки интерфейса CGI

- необходимость всякий раз устанавливать и разрывать соединение БД, поскольку отсутствуют средства поддержки постоянного соединения Web-сервера с СУБД;
- ограничения на обработку исходной информации для запросов и результатов их выполнения;
- трудоемкость выполнения программ, связанная с запуском программы как отдельного процесса.

Для устранения недостатков CGI-спецификации разработана спецификация API. Программы, разработанные по этой спецификации, быстрее и эффективнее выполняются, поскольку организованы в виде динамических библиотек. Наиболее известными являются два интерфейса этого вида: NSAPI (компания Netscape) и ISAPI (компания Microsoft).

Основное достоинство технологии **API**

Основным достоинством этой технологии является **ускорение выполнения программ**, так как программа выполняется в рамках основного серверного процесса. Сами программы имеют **большую функциональность, чем CGI-сценарии**, например, появилась возможность контролировать доступ к файлам сервера.

Недостатки технологии API

- **языковая зависимость** – программа может быть написана только на языке, поддерживаемом API;
- **слабая защита сервера от ошибок** прикладных программ и **от несанкционированного доступа** к системным ресурсам;
- поскольку программы привязаны к интерфейсу и архитектуре сервера, они **не переносимы на другие платформы**.

Язык создания апплетов *Java*

Java-апплет — прикладная программа на языке **Java** в форме **байт-кода**. **Java-апплеты** выполняются в **веб-браузере** с использованием виртуальной **Java-машины (JVM)**, или в **Sun's AppletViewer** - автономном инструменте для тестирования апплетов. **Java-апплеты** были внедрены в первой версии языка **Java** в 1995 г.

Апплеты используются для предоставления интерактивных возможностей веб-приложений, которые не могут быть предоставлены **HTML**. Так как байт-код **Java** платформенно-независим, то **Ja-va-апплеты** могут выполняться с помощью **плагинов** браузерами многих платформ, включая **Microsoft Windows**, **Unix** и др. Плагин - от англ. *plug-in* - независимо компилируемый программный модуль, динамически подключаемый к основной программе, предназначенный для расширения и/или использования ее возможностей.

Язык **Java** является основным средством реализации механизмов взаимодействия **Web-клиента** и сервера **БД** при доступе к **БД** на стороне клиента. Может также использоваться язык **JavaScript**, разработанный для расширения возможностей декларативного языка **HTML** на основе добавления процедурных средств. Программы на языке **JavaScript** выполняются на компьютере **Web-браузером** в режиме интерпретации.

Если в HTML-документе требуется получить данные из БД, то поступают следующим образом.

1. Пишут Java-апплеты, которые затем компилируются. В результате получаются машинно-независимые программы, которые могут выполняться в режиме интерпретации или служить исходной информацией для генерации программ, готовых к выполнению на разных аппаратно-программных платформах.

2. В тексте **HTML-документа** в нужных местах ставятся **ссылки на соответствующие апплеты**. Сами программы хранятся на сервере.
3. В процессе работы с гипертекстом при обнаружении в тексте ссылки на **апплет** происходит автоматическая пересылка **Java-программы** с сервера в среду выполнения **браузера** и загрузка на выполнение. Эта программа в диалоге с пользователем уточняет параметры запроса к БД.

4. Получив управление, **Java-апплет** осуществляет взаимодействие с **сервером БД**, в результате чего полученная из БД информация предоставляется пользователю.

Для обращений к серверам БД разработан **стандарт JDBC** (**Java DataBase Connectivity** – совместимость БД для Java), основанный на концепции **ODBC**. Стандарт **JDBC** фирм **Sun/Ja-vaSoft** обеспечивает универсальный доступ к БД на языке Java.

ЗАКЛЮЧЕНИЕ

Из двух рассмотренных схем доступа к БД (на стороне сервера и на стороне клиента) однозначного предпочтения тому или иному варианту отдать нельзя. Все зависит от целей и условий разработки клиент-серверных программ (наиболее существенной оказывается зависимость от ОС и от вида Web-сервера).

Достоинством модели доступа к БД на стороне сервера является сравнительная простота программ-навигаторов (Web-клиентов) и удобство администрирования системы, так как основная часть ПО находится на машине Web-сервера. Очевидным недостатком является возможное ухудшение характеристик оперативного получения информации при большой нагрузке на Web-сервер и нехватке его мощности.

Во второй модели клиентская часть системы оказывается сложнее, чем в первой. Это усложняет навигатор, но в то же время **разгружает Web-сервер**.

В настоящее время в программных продуктах фирмы Microsoft **поддерживаются обе схемы**. На стороне клиента (в среде **Internet Explorer**) существует возможность использовать **динамический HTML**, который реализуется на языке **VBScript**.

Литература

1. **Хомоненко А.Д., Цыганков В.М., Мальцев М.Г.** Базы данных: Учебник для высших учебных заведений /под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт. – 2000. – 416 с.
2. **Гаврилова Т.А.** Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский – СПб.: Питер. – 2000. – 384 с.