

О ПОСЛЕДОВАТЕЛЬНОСТИ ИЗУЧЕНИЯ СТУДЕНТАМИ ПРИНЦИПОВ ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И СЕРВЕРА ПРИ ДОСТУПЕ К УДАЛЕННОЙ БАЗЕ ДАННЫХ (РАЗДЕЛ «РАСПРЕДЕЛЕННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ», ДИСЦИПЛИНА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»)

В. Ю. Пятницкий

*Белорусский государственный педагогический
университет имени Максима Танка
Минск, Беларусь
E-mail: vladimir-piatnitski@yandex.ru*

В статье предлагается один из возможных способов изучения студентами распределенных информационных систем. Описывается роль CGI при организации доступа к серверу базы данных.

Ключевые слова: распределенные информационные системы, веб-программирование, CGI, практико-ориентированное обучение, квалификация преподавателя информатики.

В процессе реформирования системы высшего образования и введения образовательного стандарта Республики Беларусь для высшего образования (первая ступень) предполагается при подготовке студентов по специальности «Математика. Информатика» изучение дисциплины «Компьютерные системы и сети». «Распределенные информационные системы» – один из разделов данной дисциплины.

Приоритетность вопросов данной области не вызывает сомнений. Квалификация преподавателя информатики должна позволять при работе с учащимися не только владеть определенными навыками и умениями использовать интернет в целях поиска и отбора информации, но и применять средства управления данными на удаленном сервере для конструирования сложных и профессиональных веб-проектов. К этим средствам относятся технологии обработки данных, находящихся на серверах, которые могут быть расположены на значительных расстояниях от пользователя, и связь с которыми обеспечивается только посредством глобальной сети. Актуальность этих вопросов подтверждает ряд публикаций [1, 3, 5].

Одной из целей данной дисциплины является научить будущего специалиста использовать технологии веб-программирования для разработки распределенных информационных систем, а именно: создавать новые базы данных, создавать запросы к базам данных, обрабатывать запросы пользователя, генерировать выходные данные в заданном формате.

Существует большое количество инструментов разработки и манипулирования информацией баз данных, в том числе и на удаленных серверах. Каждое из них обладает различными характеристиками и особенностями, в зависимости от назначения, спектра выполняемых задач и платформ, на которых они работают. В данном курсе основное внима-

ние уделяется изучению способов обработки данных в системах удаленного пользования. Изложены основы веб-программирования обработки данных на стороне удаленного сервера. Акцент сделан на технологию PHP во взаимодействии с сервером данных MySQL.

Практика изучения данных технологий в рамках спецкурса подтвердила эффективность практико-ориентированного подхода. Последовательность изучения студентами принципов взаимодействия клиента и сервера предполагает овладение вначале практическими навыками. Затем, на лекционных занятиях, осуществляется систематизация и обобщение полученных знаний. Ниже приводится пример изложения лекционного материала по данной проблеме.

ОБОСНОВАНИЕ ЦЕЛЕСООБРАЗНОСТИ ИСПОЛЬЗОВАНИЯ ИНТЕРНЕТ ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ

К основным достижениям WWW, благодаря которым она быстро завоевала широкую популярность среди пользователей интернет, можно отнести простоту опубликования информации в сети, удобство и сравнительную унифицированность доступа к документам, наличие на сегодняшний день достаточно развитых средств поиска. Однако в целом способы представления, хранения и поиска информации в WWW относятся к категории информационно-поисковых систем (ИПС). Хотя хранилища данных в узлах веб иногда называют базами данных, этот термин в данном случае можно использовать только в самом широком смысле. Исторически ИПС применялись для хранения слабоструктурированной и редко изменяемой информации.

Базы данных в узком смысле – это хранилища структурированной, изменяемой информации, причем информация в базе данных должна всегда находиться в согласованном состоянии. Именно базы данных содержат основные знания человечества. В конце двадцатого века с появлением технологии баз данных было накоплено больше информации, чем за всю предыдущую историю. Однако доступ к базам данных (даже к тем, которые содержат полностью открытую информацию) ограничен. Чтобы получить интересующую его информацию, пользователь должен иметь физический доступ к соответствующей СУБД, быть в курсе модели данных, знать схему базы данных и, наконец, уметь пользоваться соответствующим языком запросов. Проблему языка запросов частично решает протокол ODBC, позволяющий направлять ограниченный набор операторов SQL (с промежуточной обработкой соответствующим драйвером ODBC) к произвольному серверу баз данных. Но это только частичное решение, поскольку оно никак не помогает пользователю понять схему базы данных (даже в терминах SQL) и, конечно, не способствует созданию унифицированного интерфейса конечного пользователя (нельзя же заставить всех работать в строчном режиме на языке SQL) [2, 4].

Таким образом, имеются удобные средства разработки распределенных в интернет гипермедийных документов, простые, удобные, развитые и унифицированные интерфейсы для доступа к информации WWW. Кроме того, имеется большое количество ценных баз данных, управляемых разнородными СУБД, а также желание сделать эти базы доступными всем людям (в случае публичных баз данных) или членам территориально-распределенной корпорации (в случае корпоративных баз данных). Возникает естественное желание скрестить эти две технологии и обеспечить доступ к базам данных в интерфейсе веб. Еще в конце XX в. существовали только идеи такого скрещивания и не очень тщательно разработанные подходы к реализации. На сегодняшний день имеется несколько работающих механизмов, и далее мы их обсудим. В целом механизмы делятся на два класса: обеспечивающие доступ к базе данных (по запросу клиента) на стороне веб-сервера и работающие непосредственно на стороне клиента [7].

О РОЛИ CGI-СКРИПТА ПРИ ОРГАНИЗАЦИИ ДОСТУПА К СЕРВЕРУ БАЗЫ ДАННЫХ ЧЕРЕЗ ВЕБ-СЕРВЕР

Механизм реализуется за счет наличия двух более или менее стандартизованных средств: возможности включения форм в документ, составленный с использованием языка гипермедийной разметки HTML, и возможности использования внешних по отношению к серверу веб программ, взаимодействие которых происходит через специфицированный протокол CGI (Common Gateway Interface) или внедренный позже API (Application Program Interface). (Хотя CGI называется «общим интерфейсом шлюзования», по сути дела, это одновременно некоторое подмножество протокола HTTP, и способ его соблюдения при взаимодействии сервера с внешней программой.)

При реализации на основе CGI общая схема реализации доступа к базе данных на стороне веб-сервера выглядит следующим образом: при просмотре документа клиент встречает ссылку на страницу, содержащую одну или несколько форм, предназначенных для запроса данных из базы данных; клиент запрашивает эту страницу; помимо незаполненных форм страница может содержать общую информацию о базе данных и о назначении предлагаемых форм; если клиента действительно интересует информация из базы данных, которую можно получить на основе предложенных форм, то он заполняет одну из форм и отправляет заполненную форму на сервер; получив заполненную форму, сервер запускает соответствующую внешнюю программу, передавая ей параметры и получая результаты на основе протокола CGI; внешняя программа преобразует запрос, выраженный с помощью заполненной формы, в запрос на языке, понятном серверу баз данных (обычно это язык SQL) [2, 6].

Стоит заметить, что, говоря формально, внешняя программа осуществляет компиляцию языка форм в базовый язык сервера баз данных; это может быть очень простая компиляция с использованием заготовок SQL-операторов, но тем не менее компиляция. Обычно сложность этой проблемы недооценивается. Перевести запрос, заданный на языке форм, в представление на языке SQL ничуть не проще, чем откомпилировать программу на языке Фортран в машинные коды. Внешняя программа взаимодействует с сервером баз данных; взаимодействие может быть прямым, если внешняя программа жестко привязана к конкретному SQL-серверу, или с использованием, например, протокола и соответствующего драйвера ODBC, если жесткая привязка отсутствует; после получения результатов запроса внешняя программа формирует соответствующую виртуальную или реальную HTML-страницу, передает ее серверу и завершает свое выполнение; сервер передает сформированную HTML-страницу клиенту, и на этом процедура доступа к базе данных завершается (как обычно, сервер разрывает транспортное соединение с клиентом).

На сленге веб-мастеров любая внешняя программа, запускаемая веб-сервером в соответствии со спецификациями CGI, называется CGI-скриптом. CGI-скрипт может быть написан на языке программирования (Си, Си++, Pascal и т. д.) или на командном языке (языки семейства shell, perl и т. д.). CGI-скрипт, выполняющий роль посредника между веб-сервером и другими видами серверов (например, сервером баз данных), называется шлюзом (или CGI-шлюзом). Наличие CGI-скриптов на стороне веб-сервера позволяет, в частности, перенести часть логики приложения из клиента в сервер. CGI-шлюзы представляют собой средство для организации трехзвенной (в общем случае многозвенной) архитектуры клиент-сервер.

Как видно, при использовании CGI вся интерпретация пользовательского запроса производится CGI-скриптом. Скрипт может быть предельно жестким, ориентированным на выполнение запроса к фиксированной таблице фиксированной базы данных, или относительно гибким, способным выполнить произвольный запрос к одной или нескольким таблицам базы данных, идентифицируемой в параметрах клиента [4].

Использование CGI-скриптов на стороне веб-сервера позволяет иметь на стороне клиента только сравнительно простые программы просмотра. Вся хитроумная логика работы с базами данных (возможно, с обработкой полученных данных) переходит на сторону веб-сервера. Это легкий способ построения трехзвенной архитектуры приложения. Отрицательным моментом является то, что при необходимости подключения нового CGI-скрипта, вообще говоря, требуется (относительная) модификация кода сервера.

О ВОСТРЕБОВАННОСТИ КВАЛИФИЦИРОВАННЫХ ВЕБ-ПРОГРАММИСТОВ

При применении описанного подхода возникает следующая организационно-производственная проблема: кто должен проектировать, писать, отлаживать и сопровождать процедурный код? Веб-мастера, производящие HTML-документы, обычно считают себя, скорее, дизайнерами нежели программистами. А здесь требуется исключительно программистская работа. Стала актуальной проблема подготовки специалистов, обладающих достаточными эстетическими способностями и являющихся в то же время квалифицированными и ответственными программистами.

ВЫВОДЫ

Таким образом, исходя из количества лекционных часов, отведенных на раздел «Распределенные информационные системы», а также уровня подготовленности студентов к изучению данного раздела нам видится более целесообразным практико-ориентированное изучение взаимосвязи удаленной базы данных с пользователем.

ЛИТЕРАТУРА

1. *Тохтуева, Н. А.* Профильное обучение веб-программированию на основе PHP И MySQL / Н. А. Тохтуева, А. П. Шестаков // Конгресс конференций. Информационные технологии в образовании [Электронный ресурс]. 2003. Сек I. Подсек. I. Режим доступа: <http://www.ito.su/2003/I/1/I-1-2303.html>.
2. *Гутманс, Э.* PHP 5. Профессиональное программирование / Э. Гутманс, С. Баккен. СПб. : Символ-Плюс, 2006. 438 с.
3. *Харрис, Э.* PHP / MySQL для начинающих / Э. Харрис. М. : Кудиц-образ, 2005. 512 с.
4. *Томсон, Л.* Разработка веб-приложений на PHP и MySQL / Л. Томсон, Л. Веллинг. М. : DiaSoft. 2003. 672 с.
5. *Фленов, М.* PHP глазами хакера / М. Фленов. СПб. : БХВ, 2005. 305 с.
6. *Колисниченко, Д. Н.* Самоучитель PHP 5 / Д. Н. Колисниченк. СПб. : Наука и техника, 2007. 578 с.
7. *Таненбаум, Э. С.* Распределенные системы. Принципы и парадигмы / Э. С. Таненбаум, В. С. Маартен. СПб. : Питер, 2003. 345 с.