**УДК 681.3**

***Н.Г. Пунин***

# ОТДЕЛЬНЫЕ АСПЕКТЫ ОРГАНИЗАЦИИ РАЗГРАНИЧЕНИЯ ДОСТУПА К WWW-РЕСУРСАМ[1]

В настоящее время Интернет-технологии продолжают проникать в различные сферы нашей жизни, i-бизнес непрерывно расширяет спектр предлагаемых товаров и услуг, а некоторые ресурсы должны быть доступны лишь определенному кругу пользователей.

В связи с этим перед разработчиками Web-приложений часто возникает задача ограничить доступ к некоторым WWW-ресурсам. Под ресурсами будем понимать данные сайта, статьи, форумы, формы управления и т.д. Для реализации решения данной задачи WWW-сервера предлагается решение посредством ограничения доступа к директориям. Для того чтобы определенные посетители имели доступ к файлам из некоторой директории, можно использовать встроенные в Web-сервер средства ограничения доступа, разместив ресурсы в отдельных директориях, решив таким образом поставленную задачу.

Но подобное решение не всегда удобно, а в случае разграничения доступа к базе данных – вовсе невозможно. Стандартных методов решения этой проблемы не существует, так как нет единого способа организации данных Web-приложения. Предлагается следующий метод организации данных с разграничением доступа.

Пусть необходимые данные хранятся в реляционных таблицах. Одна строка каждой таблицы – часть ресурса, к которому ограничивается доступ. Каждый ресурс должен иметь уникальный для его таблицы идентификатор (ID). В качестве уникальных идентификаторов ресурсов (ID), имеющихся в каждой таблице, используются случайные числа. Получение этих чисел происходит с помощью функции генерации случайных целых четырехбайтовых чисел стандартной библиотеки, при этом, если полученное число уже используется в этой таблице, то процесс генерации необходимо продолжать до тех пор, пока не будет найден не используемый ранее идентификатор.

Ко всем таблицам базы данных добавим еще три: таблицу пользователей, таблицу групп пользователей, таблицу разграничения доступа к ресурсам. Содержимое первых двух таблиц очевидно из их названия. В таблице разграничения доступа к ресурсам будем хранить следующие данные: идентификатор (ID) пользователя-владельца, идентификатор (ID) группы пользователей, дату создания и последней модификации, права доступа к ресурсу, номер таблицы с ресурсами и идентификатор ресурса. Таблица разграничения доступа должна содержать данные о всех ресурсах, к которым необходимо разграничить доступ.

Каждый ресурс содержит набор прав доступа, по которому определяется, как пользователь взаимодействует с данным ресурсом. Этот набор хранится в соответствующей ячейке таблицы в виде целого числа, из которого обычно используется 9 бит. Причем каждый бит используется как переключатель, разрешая (значение 1) или запрещая (значение 0) тот или иной доступ.

Девять бит делятся на три группы по три, определяя права доступа для владельца, группы и остальных пользователей. Каждая группа задает права на чтение, запись и выполнение.

Базовые биты прав доступа представлены в табл. 1. Там дано восьмеричное значение, задающее соответствующий бит, вид этого бита в первом столбце таблицы и право, задаваемое этим битом.

Рассмотрим пример организации базы данных ресурсов web-приложения (рис. 1.). Web-разработчику для использования данной структуры необходимо реализовать несколько функций: доступ к ресурсу, создание ресурса, изменение данных ресурса, удаление ресурса, а также поиск по ресурсам. Web-приложение получает идентификатор ресурса, например из GET параметров, после обработки возвращает соответствующие этому идентификатору данные. На момент работы алгоритма приложение обладает информацией о текущем пользователе, т.е. хранит идентификатор из таблицы пользователей.

Алгоритм доступа к ресурсу:

1. Запрашиваем строку таблицы разграничения доступа к ресурсам по полученному идентификатору ресурса.
2. Если текущему пользователю разрешено использовать этот ресурс, то с помощью идентификатора данных (DataID) производится поиск в таблице с номером, указанным в TypeID, иначе сообщаем об ошибке.
3. Если поиск оказался успешным, выводим найденные данные, иначе сообщаем об ошибке.

Алгоритм изменения данных ресурса.

1. Используя алгоритм доступа к ресурсу, проверяем, существует ли данный ресурс, а также есть ли разрешение на его изменение.
2. Производим изменение данных в таблице с номером, указанным в TypeID.
3. Изменяем дату и время последней модификации данных в таблице разграничения доступа к ресурсам в поле, соответствующему измененному ресурсу.

## 1. Права доступа к ресурсам

[1] Работа выполнена под руководством канд. техн. наук, проф. Ю.Ф. Мартемьянова.

| Восьмеричное значение | Право или назначение бита |
|---|---|
| 400 | Право владельца на чтение |
| 200 | Право владельца на запись |
| 100 | Право владельца на выполнение |
| 040 | Право группы на чтение |
| 020 | Право группы на запись |
| 010 | Право группы на выполнение |
| 004 | Право всех прочих на чтение |
| 002 | Право всех прочих на запись |
| 001 | Право всех прочих на выполнение |

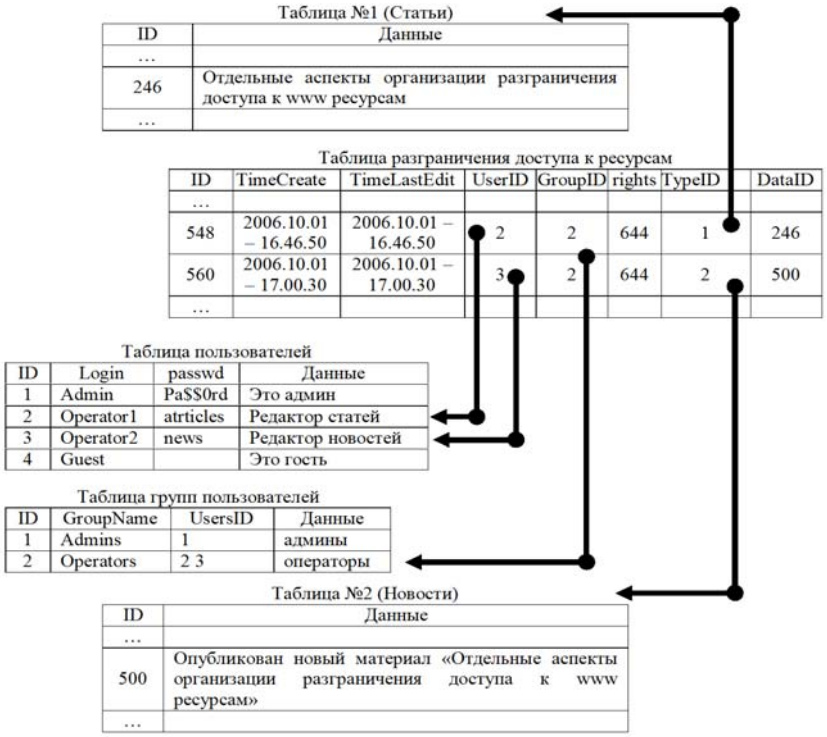


**Рис. 1. Пример организации базы данных ресурсов**

Алгоритм удаления ресурса.

1. Используя алгоритм доступа к ресурсу, проверяем, существует ли данный ресурс, а также есть ли разрешение на его изменение.
2. Удаляем строку таблицы. Номер таблицы соответствует TypeID из таблицы разграничения доступа, строка, ID которой соответствует идентификатору DataID из таблицы разграничения доступа.
3. Удаляем соответствующую строку таблицы разграничения доступа.

Алгоритм создания ресурса в таблице с номером N, группой G и правами доступа R.

1. Получение нового уникального четырехбайтового идентификатора для таблицы с номером N.
2. Добавление в таблицу с номером N новую строку, ID которой будет равен полученному на шаге 1 идентификатору.
3. Получение нового уникального четырехбайтового идентификатора для таблицы разграничения доступа.
4. Добавление в таблицу разграничения доступа новую строку, ID которой будет равен полученному на шаге 3 идентификатору. В TimeCreate TimeLastEdit записывается текущее время. В UserID – идентификатор текущего пользователя, который получим из таблицы пользователей, осуществляя поиск по ячейке Login. В GroupID и rights заносим идентификатор G и права доступа R. В TypeID и DataID заносим N и полученный на шаге 1 идентификатор.

Подобная организация доступа к данным позволит Web-разработчику удобным и надежным способом разграничивать доступ к ресурсам Web-приложения. Главное преимущество подобной организации в том, что все ресурсы собраны в одну таблицу, что позволяет увеличить производительность приложения по поиску ресурсов и минимизировать ошибки разработчика.

*Кафедра «Информационные технологии и защита информации»*