

# Web 系统移植的 PHP + MySQL 设计研究

Design and Research of Web Systems Migration by Means of PHP and MySQL

刘耀钦 ( 郧阳师范高等专科学校 计算机科学系 湖北 丹江口 442700 )

袁承芬 ( 郧阳师范高等专科学校 图书馆 湖北 丹江口 442700 )

**摘 要 :** Web 系统移植的设计实现需要解决诸如数据库创建、数据表初始化、配置文件参数等一系列问题。以 PHP + MySQL 网站为例,就设计其移植程序过程中出现的问题做了深入的分析,并对用到的 PHP 内置函数的具体使用方法也做了详细的说明。

**关键词 :** 数据库 移植 安装 创建 服务器

## 1 引言

PHP 作为一种开源式的跨平台服务器端的嵌入式脚本语言,具有执行速度快、可伸缩性、可扩展性、面向对象、支持多种数据库等特点,受到众多 Web 开发人员的青睐,最为显著的特性是其能支持 Informix、MySQL、Microsoft SQL Server、Oracle 等多种主流和非主流数据库。在其所支持的众多数据库中,以可跨平台运行的 PHP + MySQL 组合为最佳。为了顺利实现基于这些组合的网站系统的跨平台移植(安装),就需要解决与其紧密相关的诸如系统数据库创建、数据表初始化、系统文件配置等多方面的问题。

## 2 配置系统数据库

创建 MySQL 数据库,首先要建立与 MySQL 服务器的连接,PHP 提供了 `mysql_connect()` 函数用以打开或重复使用一个到 MySQL 服务器的非持久连接(随脚本运行结束而结束或使用 `mysql_close()` 关闭即关闭的连接)。该函数的具体使用说明如下:

```
resource mysql_connect ( [ string $server [ ,  
string $username [ , string $password [ , bool  
$new_link [ , int $client_flags ] ] ] ] )
```

`mysql_connect()` 以 `$username` 和 `$password` 所指定的用户及其密码打开由 `$server` 标识的数据库服务器或到本地套接字的路径,连接成功后会返回一个 MySQL 连接标识符,失败时返回 `FALSE`。若用相同的参数第二次调用 `mysql_connect()` 时,返回的是已经打开

的连接标识,指定 `$new_link` 则会改变这种情况,也即 `mysql_connect()` 返回的是新打开的连接标识,另外 `$client_flags` 表示客户端的诸如压缩通讯协议、空闲等候时间及 SSL 加密等常量的组合。

虽然使用 PHP 中的 `mysql_create_db()` 可以创建 MySQL 数据库,但推荐使用 `mysql_query()` 函数来提交一条“CREATE DATABASE”的 SQL 语句来代替。因为如果 MySQL 是基于 MySQL 4.x 客户端库编译扩展的话 `mysql_create_db()` 会不可用。其函数具体使用说明如下:

```
resource mysql_query ( string $query [ , re-  
source $link_identifier ] )
```

`mysql_query()` 将 `$query` 表示的 SQL 语句发向由 `$link_identifier` 关联的服务器中的当前活动数据库。如果没有指定 `link_identifier`,则使用上一个打开的连接。该函数对诸如 `SELECT`、`SHOW`、`EXPLAIN`、`DESCRIBE` 的 SQL 语句执行成功则返回一个资源标识符,对其它类型的 SQL 语句执行成功时返回 `TRUE`,两者出错时均返回 `FALSE`。

根据以上两个函数的描述,可以得到如下实现创建数据库的算法:

```
< ?php  
function createDb( $hostlink, $dbname, $chare-  
set ) {  
    if ( ! $hostlink ) { echo " < script language =  
JavaScript > alert( 提示 : \n\n 数据库连接失败或没有  
安装 MySql ! \n\n 点击确定重新设置 ! ^ ) ; window.
```

```

history.back( -1 );</script > " ; }
//若数据库连接失败则返回至上一页
else{ $ dblink = mysql_select_db( " $ dbname" ,
$ hostlink ) ;//打开需创建的数据库
if( ! $ dblink ){echo " < script language = JavaS-
cript > alert( 提示 : \n \n 你所创建的数据库已经存在 !
\n \n 点击确定重新设置 ! ^ ) ; window.history.back( -
1 ) ; " ; } //若需创建的数据库存在则返回上一页
else{ $ sql = `CREATE DATABASE ` $ dbname . `
CHARACTER SET ` $ charset ;//若不存在则创建该数
据库
if( ! mysql_query( $ sql , $ hostlink ) ) {
echo " < script language = JavaScript > alert( 提
示 : \n \n 无法创建 - $_POST[ dbname ] - 数据库 ! 请
检查用户权限 ! \n \n 点击确定重新设置 ! ^ ) ; win-
dow.history.back( -1 );</script > " ; } } }
//若无创建权限则提示
? >

```

算法中 \$ hostlink 表示打开的 MySQL 连接标识 , \$ dbname、\$ charset 表示由客户端提交的数据库的名称和字符集。

### 3 创建系统数据表

PHP 中没有专门用来创建 MySQL 数据表的函数 , 但是我们可以使用前文讲过的 mysql\_query() 函数 , 它可以将一条合法的 SQL 语句提交给当前关联服务器中的当前活动数据库。通常情况下 , 获得创建数据表的 SQL 语句有两种途径 , 一是通过客户端用户提交 , 二是通过将 PHP 提供的 fopen()、fread() 函数置入程序中打开含有数据表 SQL 的文件。第一种方式较为简单 , 不再赘述 , 下面着重讲一下第二种方式下两个函数的具体用法 :

```

resource fopen( string $ filename , string $ mode
[ , bool $ use _ include _ path [ , resource
$ zcontext ] )

```

fopen() 以 \$ mode 指定的访问类型打开由 \$ filename 所表示的本地路径或 URL 下的文件 , 并在打开成功时返回文件句柄 , 失败时返回 FALSE。如果要在 php.ini 文件中“ include\_path ”指定的路径下搜索文件则要将 \$ use\_include\_path 指定为 TRUE 或 ‘ 1 ’。

```

string fread( int $ handle , int $ length )

```

fread() 从文件句柄 \$ handle 开始读取最多 \$ length 个字节 , 读取完 \$ length 字节后或读到文件末尾时停止进而返回所读取的字符串 , 否则返回 FALSE。具体算法如下 :

```

< ? php
function openFile( $ filePath ){
$ handle = @ fopen( $ filePath , " rb" ) ; // " rb "
表示访问文件的类型 ; " b " 表示强制使用二进制模式 ,
建议使用
$ contents = @ fread( $ handle , filesize( $ file-
Path ) ) ; //从 $ handle 开始读取 filesize( $ filePath ) 个
字节的数据
if( $ handle ) return $ contents ; //若正确读取则
返回读取到的内容
else print " < script language = JavaScript > alert( ^
提示 : \n \n SQL 文件不存在 ! \n \n 点击确定返回 ! ^ ) ;
window.history.back( -1 );</script > " ;
//若读取文件不存在则返回上一页
@ fclose( $ handle ) ; //关闭文件句柄 }
? >

```

通过前面的介绍 , 我们知道 mysql\_query() 函数每次只能提交一条合法的 SQL 语句 , 通常情况下 , 所获取文件的内容或表单提交来的数据中会包含多条 SQL 语句 , 需要用 PHP 提供的 explode() 函数将其分隔。该函数具体说明如下 :

```

array explode( string $ separator , string $ string
[ , int $ limit ] )

```

explode() 将 \$ string 表示的字符串以 \$ separator 为分隔符进行分隔 , 并返回由这些分隔后的子串组成的数组。由此得到以下实现创建数据表的算法 :

```

< ? php
function createTable( $ hostlink , $ dbname ){
mysql_select_db( $ dbname , $ hostlink ) ; //设定
由 $ hostlink 关联服务器上的活动数据库 $ dbname
$ tablesql = openFile( ' ./myDataSql.sql ' ) ; //调
用 openFile() 打开 SQL 文件
$ newTSql = explode( ` ; ` , $ tablesql ) ; //以 “ ; ” 为
分隔符分隔获取的完整 SQL 语句
$ count = count( $ newTSql ) - 1 ; //将读取到的完
整 SQL 语句数量减 1
for( $ i = 0 ; $ i < = $ count ; $ i + + ){ //遍历读

```

取到的完整 SQL 语句

```
$ result = mysql_query( stripslashes( $ newTSql
[ $ i ] ), $ hostlink ) ;//对去掉反斜线的 SQL 语句执行
查询
```

```
if( ! $ result ){
    echo " < script language = JavaScript > alert( '提示 :\n\n 第 - { $ i } - 段 SQL 语句含有不合法字符 !
( SQL 语句段要以 - " ; " - 为分隔符 ) , window. history.
back( - 1 ) ; < /script > " ; } //判断读取到的完整 SQL 语句是否合法 } }
```

? >

需要说明的是,如果 PHP 的配置选项 magic\_quotes\_gpc 设置为 ON 时,由 GET、POST、COOKIES 传递来的字符串中的“'”、“”、“\”和“\’”、“\”、“\”,解决这个问题需设置 magic\_quotes\_gpc 为 OFF 或使用 stripslashes() 函数,magic\_quotes\_gpc 的设置事关系统安全,要慎重设置。

#### 4 生成系统配置文件

系统配置文件通常以包含数据库连接、系统参数等信息的外在形式存在。一般情况下,配置文件以 PHP 的形式存储于系统当前目录下,如果配置文件随同网站整体移植会使网站不能正常运行,这是因为移植后的平台上还没有安装系统所需要的数据库,为了在不移植配置文件的情况下保证系统正常运行,就需要在网站的主页中添加如下程序段来判断配置文件是否存,进而让浏览器决定显示安装界面或是网站主界面。

```
< ? php
function filefExsits( $ filePath ){
    define( " currentPathFile" ,dirname( __FILE__ ). "
\ . $ filePath ) ;//获取系统所需配置文件的路径
    if( is_file( currentPathFile ) )return true ;// 如果文件存在且为正常的文件则返回 true
    elsereturn false ;//若文件不存在或不正常则返回 false }
    if( ! filefExsits( includefile \web_config. php` ) ){
        echo " < table class = border_all align = center cellpadding = 15 > < tr > < td align = center > 系统所需配置文件不存 , < b > < a href = `./install/install. php` > 点击此处 < /a > < /b > 安装该文件 ! < /td > < /tr > < /table > " ;//若配置文件不存在则输出提示
```

```
return false ;//终止程序继续运行 }
```

? >

安装完系统所需的数据后,要将 MySQL 服务器连接标识、设定的活动数据库、使用的字符编码等基本信息写入配置文件。其实现算法如下:

```
< ? php
function createConfigFile ( $ filePath , $ contentStr ){
    define( `parentDir` ,substr( dirname( __FILE__ ) , 0 , - 8 ) ) ;//获取当前目录的父目录
    $ fp = fopen ( parentDir . " $ filePath " , " w + " ) ;//“ w ”表示以读写方式打开,指向文件头并清空文件内容,如果文件不存在则尝试创建之
    fwrite( $ fp , $ contentStr ) ;//将配置内容写入文件
    fclose( $ fp ) ;//关闭文件 }
? >
```

#### 5 小结

对 PHP + MySQL 组合网站系统来说,网站移植程序不仅是一种补充,更是其功能上的扩展,为系统跨平台移植后各项参数的配置提供了方便。PHP 以其功能强大的内置函数和数据库处理能力为我们解决了设计过程中诸如数据库创建、数据表初始化、系统文件配置等一系列问题,但是其所支持的每种数据库均有其特定的处理函数,也就是说其缺少通用的数据库处理函数,这给数据库的跨平台移植和升级更新均带来诸多不便。本文所有算法均在 PHP5.2.5 + MySQL4.1 + Apache2.0.55 环境下调试通过。

#### 参考文献

- 1 推进互联网技术学习应用的文化. <http://www.w3pop.com/learn/index/t/3/c/16/>. 2007 - 4 - 28/2008 - 1 - 12.
- 2 Gabor Hojtsy 编著,王远之等译. php\_manual\_zh.chm. <http://www.php.net/docs.php>. 2007 - 12 - 2/2008 - 1 - 12.
- 3 杜江编著. PHP5 与 MySQL5 Web 开发技术详解. 北京:电子工业出版社,2007. 11. 268 - 338.
- 4 黄炳强编著. PHP5 + MYSQL 网站开发实例精讲. 北京:人民邮电出版社,2007. 7. 195 - 224.