**“笔者用XAMPP的，默认不需要开php权限**

**-- Sidney”**

**exec 或者 system 都可以调用cmd 的命令**

直接上代码：

复制代码代码如下:

<?php  
/\*\* 打开windows的计算器 \*/  
exec('start C:WindowsSystem32calc.exe');

/\*\* php生成windows的批处理文件后，再执行这个批处理文件\*/  
$filename = 't.bat';  
$somecontent = 'C:  
';  
$somecontent .= 'cd "C:/Program Files/MySQL-Front"';  
$somecontent .= '  
start MySQL-Front.exe';  
if (!$handle = fopen($filename, 'w')) {  
 echo "不能打开文件 $filename";  
 exit;  
}

/\*\* 首先我们要确定文件存在并且可写\*/  
if (is\_writable($filename)) {

 /\*\* 那就是当我们使用fwrite()的时候，$somecontent将要写入的地方  
  将$somecontent写入到我们打开的文件中 。\*/  
 if (fwrite($handle, $somecontent) === FALSE) {  
  echo "不能写入到文件 $filename";  
  exit;  
 }  
 echo "成功地将 $somecontent 写入到文件 $filename";  
 fclose($handle);  
} else {  
 echo "文件 $filename 不可写";  
}  
exec($filename);  
?>

有一个遗留问题，就是exec（）调用，php会一直执行，直到你关闭启动的应用程序，这样会造成php执行超时，不知道怎么解决这个问题，希望高手路过此地，留下答案！我日后解决了，也会更新到这里的！

**以下来自资料**

=================================================

php的内置函数exec，system都可以调用系统命令(shell命令)，当然还有passthru，escapeshellcmd等函数。

在很多时候利用php的exec，system等函数调用系统命令可以帮助我们更好更快的完成工作。

**注意:要想使用这二个函数 php.ini 中的安全模式必须关闭，要不然为了安全起见php是不让调用系统命令的。**

先看一下php手册对这二个函数的解释:

**exec --- 执行外部程式**

语法 ： string exec ( string command [, array &output [, int &return\_var]] )

说明 ：  
exec()执行给予的命令command，不过它并不会输出任何东西，它简单的从命令的结果中传回最后一行，如果你需要去执行一个命令，并且从命令去取得所有资料时，可以使用passthru()这个函数。  
如果有给予参数array，则指定的数组将会被命令所输出的每一行填满，注意 ： 如果数组先前已经包含了一些元素的话，exec()将会把它附加在数组的后面，如果你不想要此函数附加元素的话，你可以在传递此数组给exec()之前呼叫unset()。  
如果有给予参数array和return\_var，则传回执行的状态命令将会写到这个变量。

注意： 如果你允许来自使用者输入的资料，可以传递到此函数，那么你应该使用escapeshellcmd( )来确定此使用者无法哄骗(trick)系统来执行武断的(arbitrary)命令。

注意： 如果你使用此函数来启动一个程式，而且希望在背景里(background)执行的时候离开它，你必须确定此程式的输出是转向(redirected)到一个文件或是一些输出的资料流，否则PHP将会悬挂(hang)直到程式执行结束。

**system --- 执行外部程式并且显示输出**

语法：string system ( string command [, int &return\_var] )

说明：

system( )执行给予的命令command，并且输出结果。如果有给予参数return\_var，则执行命令的状态码将会写到这个变量。

注意： 如果你允许来自使用者输入的资料，可以传递到此函数，那么你应该使用escapeshellcmd( )来确定此使用者无法哄骗(trick)系统来执行武断的(arbitrary)命令。

注意： 如果你使用此函数来启动一个程式，而且希望在背景里(background)执行的时候离开它，你必须确定此程式的输出是转向(redirected)到一个文件或是一些输出的资料流，否则PHP将会悬挂(hang)直到程式执行结束。  
如果PHP是运作成伺服器模组，在输出每一行后，system( )会试着自动地清除web伺服器的输出缓冲。

成功则传回命令的最后一行，失败则传回false。

如果你需要去执行一个命令，并且从命令去取得所有资料时，可以使用passthru( )这个函数。

这二个都是用来调用系统shell命令，

**不同点:**

exec可以把执行的结果全部返回到$output函数里(数组)，$status是执行的状态 0为成功 1为失败

systerm不需要提供$output函数，他是直接把结果返回出来，同样$return\_var是执行的状态码 0为成功 1为失败

exec示例：

复制代码代码如下:

<?php  
 $a = exec("dir", $out, $status);  
 print\_r($a);  
 print\_r($out);  
 print\_r($status);  
?>

system示例：

复制代码代码如下:

<?php  
 $a = system("dir", $status);  
 print\_r($a);  
 print\_r($status);  
?>

上面说明看起来有点乱，运行两个例子后，你就会明白的！

**【system】**

复制代码代码如下:

<?php  
set\_time\_limit(0);  
define('ROOT\_PATH', dirname(\_\_FILE\_\_));

include ROOT\_PATH . '/include/global.func.php';

$cmdTest = 'ps -ef | grep magent';

$lastLine = system($cmdTest, $retVal);

write\_log('$lastLine');  
write\_log($lastLine);

write\_log('$retVal');  
write\_log($retVal);  
?>

输出：

复制代码代码如下:

++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:28:52  
$lastLine  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:28:52  
root      5375  5373  0 16:28 pts/1    00:00:00 grep magent  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:28:52  
$retVal  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:28:52  
0

**【exec】**

复制代码代码如下:

<?php  
set\_time\_limit(0);  
define('ROOT\_PATH', dirname(\_\_FILE\_\_));

include ROOT\_PATH . '/include/global.func.php';

$cmdTest = 'ps -ef | grep magent';

$lastLine = exec($cmdTest, $output, $retVal);

write\_log('$lastLine');  
write\_log($lastLine);

write\_log('$output');  
write\_log($output);

write\_log('$retVal');  
write\_log($retVal);  
?>

输出：

复制代码代码如下:

++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
$lastLine  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
root      5360  5358  0 16:25 pts/1    00:00:00 grep magent  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
$output  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
Array  
(  
    [0] => root      2838     1  0 15:39 ?        00:00:00 magent -u root -n 51200 -l 192.168.137.100 -p 12001 -s 192.168.137.100:11211 -b 192.168.137.100:11212  
    [1] => root      5358  5356  0 16:25 pts/1    00:00:00 sh -c ps -ef | grep magent  
    [2] => root      5360  5358  0 16:25 pts/1    00:00:00 grep magent  
)

++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
$retVal  
++++++++++++++++++++++++++++++++++++++++++  
2014-10-15 16:25:00  
0

**结论：**

如果需要详细的输出结果，那就用exec()吧！我一般用exec()执行外部命令！