

## CGI编程教程(二)

### CGI教程第七章

上传文件是建立交互式网站的一个重要手段。一般来说，考虑到CGI程序的安全和一些敏感问题，服务器是不会轻易开放文件上传权利的。

今天的程序是由WEBSITE1.1附带的一个upload.exe程序，由VB编写而成。所以我要讲的也就是VB写WIN-CGI时候的一些问题。

#### VB写Win-CGI:

在Windows环境下，WEB服务器和Windows CGI之间是通过临时文件来传送数据的。这个过程如下：

1. Web服务器接受一个客户机对Windows CGI程序的请求，这个请求也称为Windows CGI请求。
  2. Web服务器把要传送给Windows CGI程序的数据放在一个叫CGI profile的主输入文件。在某些情况下，还会增加一个文件。
  3. Web服务器执行CGI程序，把CGI profile作为它的第一个参数。
  4. Windows CGI开始执行，服务器等待Windows CGI进程结束。在等待过程中，服务器还可以处理其他请求，比如另一个新的CGI进程或HTTP请求。
  5. Windows CGI程序读取作为命令行参数的profile文件并完成必要的处理，CGI程序也许要使用profile文件中的数据和其他相关的辅助文件中的数据。
  6. 在Windows CGI程序结束前，把它的应答写在profile文件指定的输出文件中。
  7. 得到CGI程序处理结果后，WEB服务器读取输出文件，看看输出数据是否要打包，并把最后数据发给客户机。
  8. WEB服务器随后就删除处理这个请求时产生的临时文件。如果在服务器管理器注册时，不删除临时文件，则保留这些临时文件。
- 总的说，基于文件的传输过程是比较古老的。但它是最常用的通讯方式。

#### CGI32. BAS

CGI32. BAS是给windows系列的使用者在VB4.0或更高的环境下开发Windows-cgi而写的一个程序接口。本文假设你已经对VB编程非常熟悉，所以对于使用VB的过程我就省略不说了。在写Win-CGI程序的时候，请将VB环境中的有效对象全部关掉（Custom controls），并将CGI32.bas加入project中。随后创建一个子程序

```
CGI-Main()
```

```
Sub CGI-Main()
```

```
'Win-CGI程序内容
```

```
end sub
```

```
Sub Main()
```

```
On Error GoTo ErrorHandler
```

```

If Trim$(Command$) = "" Then ' Interactive start
Inter_Main ' Call interactive main
Exit Sub ' Exit the program
End If

InitializeCGI ' Create the CGI environment

'=====
CGI_Main ' Execute the actual "script"
'=====

Cleanup:
Close #CGI_OutputFN
Exit Sub ' End the program
'-----

ErrorHandler:
Select Case Err ' Decode our "user defined" errors
Case ERR_NO_FIELD:
ErrorString = "Unknown form field"
Case Else:
ErrorString = Error$ ' Must be VB error
End Select

ErrorString = ErrorString & " (error #" & Err & ")"
On Error GoTo 0 ' Prevent recursion
ErrorHandler (Err) ' Generate HTTP error result
Resume Cleanup
'-----

End Sub

```

上面是一个Win-cgi的例子。

### upload.exe

限于篇幅，这个程序不打算写出所有代码，只是把编译好的upload.exe放在这里。有兴趣的朋友可以在[这里下载](#)。现在后，请将upload.exe放在你的Win-CGI目录下，如果你使用

其他的Server（不是Website的话），可能在执行时候会有些麻烦，那就没有办法了。

## CGI简明教程 第八章

所谓Form to Mail的功能，就是在页面上建立一个表单(form)，该表单内记录着用户的信息，当用户提交了此表单，程序自动将表单内容发送的管理员所设定的信箱里。听起来这非常简单，设置根本用CGI就可以写这样的功能。当然没有这么简单，实际上许多公司网页还要求此表单在提交时存入另一个数据库，这样就需要写CGI程序了。为了便于解说form to mail的功能，我们这里不考虑复杂情况，只考虑最简单的情况。就是唯一的form to mail功能，其他的功能只要在程序内加入相应的程序块就可以了。在Unix中，我们最多使用的就是Email网关，是CGI开发环境中少数几个标准unix工具之一。

在CGI程序中发送e-mail的命令是mail和Sendmail。我比较喜欢sendmail，因为它功能更强，Sendmail从STDIN接受数据，遇到Eof或只包含.的一行信息就将信息发送出去。Perl中使用Sendmail的一个方式是将mail内容写入一个临时文件，然后使用Cat，同时把输出管道与Sendmail命令相联系起来。

如：

```
open(EMAIL, ">tempfile $ $,txt" );

print(EMAIL, " stuff to be emailed .. la la la ..\n" );

close(EMAIL);

system( "cat tempfile $ $,txt | /usr/sbin/sendmail $tokens( 'email' )" );

system( "rm tempfile $ $,txt" );
```

名字tempfile \$\$,txt中的\$ \$是Perl中的一个特殊变量，即进程号(PID)。之所以使用是为了

避免两个用户同时启动这个程序会覆盖tempfile文件，使用这种方法每个tempfile都是唯一的。（由pid表示）。这种好的思想可以用在许多程序中。Sendmail命令从关联数组元素\$token( 'email' )中找到e-mail目标地址，你可以假设这个变量的值在用户提交的表单由用户输入。如果没有安全措施，这样做是很危险的，设想有恶意的拥护在Web表单中给你如下地址：

```
Noone@nowhere.net;cd/;rm-R*
```

分号是表示unix命令结束的符号并将命令提交unix命令解释器。精通web的人谁也不会会在根目

录运行httpd，这个例子将删除整个文件系统。所以在设计整个系统的时候，必须考虑到这种用户直接到达Shell的情况。

## Windows 下的Form to Mail

很不幸，Windows没有类似于Unix的mail网关，不管是Nt和Windows都无法做到类似的功能。当然也有些非常热心的程序员写过Dos接口的mail程序，但是相当不幸，此类程序直到现在也没有很成功的例子。它们的问题是过于占用系统资源，并非常容易被黑客攻击。所以这里我就不提供程序了。

## CGI简明教程 第九章

Guest-book在网页设计中占有相当重要的地位，以至于可以和计数器相提并论。留言本可以用许多编程语言来设计，当然使用最多的还是perl和vb两种。本次cgi教程我们将谈谈在编写Guest-book中最应该注意的问题，并将给出一个不算很完整的程序代码。同时提供一个C语言的win-cgi程序供大家下载。就现在的情况来看，一个标准的guest-book程序一般包括三个文件。提交表单文件（form.html）、cgi处理文件(guest-book.cgi)和最后的输出文件（book.html）。有些vb程序以access数据库为后台存储格式，以方便用户查找留言。下图给出了guest-book的工作流程。

form.html

首先我们来看看form.html的文件内容：

```
<html>
<title>guest-book.html</title>
<h1>留言本</h1>
<form method="post" action="/cgi-shl/guest-book.cgi">
<p>名字: <input name="name" size="20">
<p>email:<input name="email" size="30">
<p>意见: <textarea name="suggest" cols=30 rows=4></textarea>
</textarea><p>
<input type="submit" value="提交">
<input type="reset" value="取消">
</form></html>
```

```
<form method="post" action="/cgi-shl/guest-book.cgi">
```

将此表单的提交后的动作交给guest-book.cgi程序。

guest-book.cgi

```
#!/ perl
```

```
print "Content-type:text/html\n\n";
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs=split(/&/, $buffer);
foreach $pair(@pairs) {
```

```

($name,$value)=split(/=/,$pair);
$value=~tr/+///;
$value=~s/%([a-f A-F 0-9][a-f A-F 0-9])/pack("C",hex($1))/eg;
$FORM{$name}=$value;
}
print"<title>留言本</title>";
print"<h1>留言本</h1>";

foreach $check(values%FORM) {
if($check eq "") {
print"<hr size=3>";
print"<h2>请不要留空";
exit(0);
}
}

$FORM{' suggest' }=~s/cM\n/<br>\n/g;

$filename="c:/website/htdocs/book.html";

open(FILE,">> $filename");
print FILE "名字: $FORM{' name' }<br>\n";
print FILE "Email:<a href='mailto:$FORM{' email' }>$FORM{' email' }</a><br>\n";
print FILE "留言: $FORM{' suggest' }";
close{FILE};

print "<h2>谢谢留言</h2>";

print"<a href='\"book.html\"'>请观看留言本</a>";

exit(0);

```

这个程序相当简陋，但它具有了所有的CGI-Guest程序的特点，我们一一分析如下：  
首先，

```

read(STDIN,$buffer,$ENV{'CONTENT_LENGTH'});
@pairs=split(/&/, $buffer);
foreach $pair(@pairs){
($name,$value)=split(/=/,$pair);

```

```
$value=~tr/+/;/
$value=~s/%([a-f A-F 0-9][a-f A-F 0-9])/pack("C",hex($1))/eg;
$FORM{$name}=$value;
}
```

这是一段表单分析的语句，在前面的程序里，估计大家也都见过了。

其次：

```
foreach $check(values%FORM){
if($check eq ""){
print"<hr size=3>";
print"<h2>请不要留空";
exit(0);
}
}
```

做输入字段的检查，看看是否有留空的现象。从前这种任务都是要CGI程序来验证的，比如email是否输入的是mail地址等检查。不过现在我们可以通过javascript来做这样的检查，这样服务器的负担将大大减轻。

最后：

```
$filename="c:/website/htdocs/book.html";

open(FILE,">> $filename");
print FILE "名字: $FORM{'name'}<br>\n";
print FILE "Email:<a href='mailto:$FORM{'email'}>$FORM{'email'}</a><br>\n";
print FILE "留言: $FORM{'suggest'}";
close{FILE};

print "<h2>谢谢留言</h2>";

print"<a href=\"book.html\">请观看留言本</a>";
```

```
exit(0);
```

输出内容到文件和屏幕上。完成留言的程序。

book.html:

这个程序中book.html没有什么特别的要求，可以不用写，就让程序自动生成。现在有的cgi程序，对book.html也有设定，比如在<head></head>之间就写上注释

```
<!--这是流言开始--!>
<!--这是流言结束--!>
```

这样，在程序对留言数据操作的时候，就可以忽略掉文件内的其他内容。

## Cgi简明教程 第十章

毫无疑问，当今Web程序设计中最吸引人而且最复杂的当然就是web数据库程序了。平心而论，其复杂高深的程序决不是你们可以想象的到的。我们先来看看其中涉及到的几个方面的

问题：

1. 最基本的html设计
2. CGI程序编写调试
3. 网络管理和客户协调
4. 数据库编写
5. 客户/服务体系程序编写

另外你还得不时地同你的客户进行斗争，以取得他们对你工作的认同。因此编写一个大型WEB数据库，是综合素质的考验。笔者自问尚不完全具备以上的能力，但有幸曾参加过几个web数据库的开发，算是具备了初步的经验吧。这里给大家分享，也算是感谢各位对这个断断续续的教程的总结吧。

### 一 数据库选择

从一般情况来看，使用web数据库往往是要解决数据的归纳、索引和维护的问题。我们一般选择最流行的关系型数据库，比如NT下的sql，win95和nt下的Access，NT下的sybase，unix下的mysql等等。当然还有Oracle、FileMaker PRO、Paradox等等。这些都是很流行的Sql数据库。Sql给数据管理提供了一个标准而坚实的接口，它对数据库操作所有函数必需在数据库语言中实现。这种函数包括：数据对象的创建、插入数据和数据修改等。对于数据量不大的小型数据库，一般使用access。

### 二 接口程序

虽然我们可以直接编写WEB的CGI接口，但是那是工作量巨大且麻烦异常的工作。为了减轻开发者的负担，数据库程序员为我们编写了许多可以直接使用的接口程序，配合不同的数据

库。下面就是一些十分常用的接口程序：

#### 1. DB2WWW

<http://www.software.ibm.com/data/db2/db2wfac2.html>

这是一个用于和IBM DB2数据库进行连接的工具。在HTTP服务器中，这种工具将HTML文件和SQL命令作为宏文件存储，然后，DB2WWW在接受到浏览器请求后就处理这些宏文件。

#### 2. dbCGI

<http://www.progress.com/webtools.html>

这个工具通过在html文件中内置SQL语句来实现和数据库的连接。适用于

Progress, Sybase, Oracle, Informix, Ingres和ODBC。下面是一个典型的dbCGI程序代码：

```
<title>dbCGI test</title>
<sql init>
```

```
INFORMIXDIR=/usr/informixdir
SQLEXEC=/usr/informix/sqlturbo
TBCONFIG=tbconfig
</sql>
<sql connect conn1>
<DATABASE=stores>
<sql>
<sql format>
% [! 8:<h1>%8d</h1>%]
....
</sql>
<sql query conn1>
selet * from customer order by date,city,zipcode
</sql>
<sql disconnect conn1>
</sql>
<sql uninit>
</sql>
```

这个工具通过<sql>标签和%转意字符来把sql语句放置在html语句内。

### 3 Genera

<http://gdbdoc.gdb.org/letovsky/genera/general.html>

这个是sybase数据库系统的网关。它通过在文本文件中加入特定的语句来和数据库进行连接。

### 4 WebDBC

<http://www.ndev.com>

以windows NT为平台，可以访问所有对odbc兼容的数据库。

### 5 A-Xorion

<http://www.clark.net/infouser/endidc.html>

他是一个CGI网关，它可以访问大多数的PC机数据库。它以Windows3.1/95/NT为平台。数据库为Ms Access2.0。

### 三编程语言

上面介绍的是一些常用的接口程序，它们需要Server的配合才能正常工作。当然如果是很小的数据库程序，不需要用如此复杂的数据库接口程序。用C,Perl,VB等等，写成CGI程序，

在CGI程序内调用数据库。虽然速度方面不是很快，但绝对是非常简单的。