

CGI 程序与 FORM 表单交互的实现

吕 峥 朱逸芬

摘 要 如何运用 CGI 程序扩展 HTTP 服务器的功能,实现 WWW 上前台与后台的交互及页面的动态显示。本文就其原理、方法和性能进行了论述,并给出了一个简化的实例。

关键词 公共网关接口, HTML, 表单, HTTP

CGI PROGRAM INTERACTING WITH FORM

Lu Zheng Zhu Yifen

Institute of Computer Technology, Hefei University of Technology,
Anhui·Hefei 230009

Abstract How to use CGI programs to extend the functions of HTTP server, to realize the interaction of browser with HTTP server, and to display dynamic homepages. This paper discuss the theories, methods and performances about these questions, and a simple sample is given.

Keywords Common gateway interface, HTML, Form, HTTP

1 CGI 基本原理和概念

CGI (Common Gateway Interface) 是服务器和前台 HTML 文件之间的接口程序,它负责处理 HTML 文件与运行在服务器中的程序 (HTML 以外的程序) 之间的数据交换。从本质上讲, CGI 是一种服务机制。当用户输入了数据后,通过浏览器向 HTTP 服务器发出一条具有 GET 或 POST 命令格式的请求,服务器收到后即作出响应,并且激活一个 CGI 程序。该 CGI 程序可以调用其它外部程序,以完成前台的指令。当系统完成任务之后,再把执行结果传给 HTTP 服务器。可见, CGI 程序在前台用户与服务器之间进行数据交互以及动态生成页面时起着至关重要的作用。没有 CGI 程序,就很难实现前台采用 HTML 的 FORM 表单与后台的交互。

图1 浏览器与服务器间的交互

具体步骤如下:

- ① 前台客户机使用 TCP/IP 协议,与服务器建立联接,发送 URL 请求;
- ② 服务器到相应的目录中调用 CGI 程序,使用客户机传递的参数作为 CGI 程序的参数,而 CGI 程序调用相应的外部程序完成操作;
- ③ CGI 程序以能被识别的格式返回处理结果给 HTTP 服务器;
- ④ HTTP 服务器将数据返回客户机处理,显示 CGI 执行结果。

至于使用哪一种语言编写 CGI 程序,一般取决于设计人员的喜爱,同时也要考虑工程维护和性能的要求。常用的 CGI 程序语言有 Shell Script, Perl, VB, C/C++, 等等。

具体交互的实现牵涉到前后台编程,要考虑前后台如何交换信息,如何处理用户数据,如何存储用户数据等问题。本文给出的实例中, HTTP 服务器采用 Websitel.1, CGI 接口采用 VB 5.0, 前台 FORM 的编制采用 HTML3.0。

2 前台 FORM 表单的编制

在前台，从客户机传送数据到服务器的方法有三种：通过 URL 的查询行；通过外加路径信息；通过消息体。而用户提交的查询或维护请求数据主要是采用 FORM 表单形式提交给 CGI 程序的。FORM 是 HTML（超文本标记语言）的一种标识，可以用任何一种文本编辑器编写，但最后必须以纯文本格式保存，并且以 .htm 为扩展名才能被 WWW 浏览器识别。已编好的 HTML 文件必须通过 WWW 浏览器来解释并呈现在操作平台上。

- ① 前台浏览器向 HTTP 服务器请求某个 HTML 文档；
- ② 服务器向前台发送一个含有 FORM 表单的页面；
- ③ 前台用户在 FORM 表单中填入用户数据提交，并发送要运行的 CGI 程序的 URL；
- ④ HTTP 服务器启动 CGI 程序并将 FORM 表单数据传给 CGI 程序；
- ⑤ CGI 程序处理 FORM 表单数据并将结果通过服务器返回给前台。

图2 FORM 表单查询的基本流程

如下给出了一个简化了的前台 FORM 例程，并作了简单的说明。

```
<html>
<head>
<title>FORM 表单交互例程</title>
</head>
<body bgcolor="#FFFFFF" background="backgrnd.gif">
<form method="post" action="/cgi-win/jiaoyan.exe">
<pre>
姓名:<input name="xm" type="text" value="" size=30
maxlength=30>
密码:<input name="mm" type="text" value="" size=30
maxlength=30>
</pre>
<input type="submit" value="提交验证"><input type="reset" value="重置表单"
">
</form>
</body>
</html>
```

其中，ACTION 属性说明当这个 FORM 提交后需要执行的 CGI 程序名和路径。

Method 属性说明由服务器发送信息时的方法，“GET”和“POST”是两种有效的适合于 CGI 的方法。两者的区别是使用 GET 方法时，输入的信息作为字符串

附加在 ACTION 所设定的 URL 后面,中间用“?”隔开, CGI 程序用 QUERY_STRING 环境变量去取得 FORM 所能传送的信息。字符串长度受到环境变量长度的限制。GET 方法还可以提交 HTML 的数据采集表 FORM, 实现搜集用户数据并传给服务器, 由于 GET 请求只用于 URL 的内容, 不能包括任何对象, 因此 FORM 的内容只能在 URL 请求行中。由于每个服务器对于 URL 请求行的长度有规定 (如 NCSA 规定请求行长度不超过8192), 因此提交 FORM 的内容大多用 POST 方法。POST 方法主要用于从前台向后台 CGI 程序传送 FORM 内容, 然后由 CGI 程序处理这些 FORM 的域。在使用 POST 方法时, CGI 程序用 CONTENT_LENGTH 去决定输入信息的长度, 再从标准输入<stdin>读出信息的内容。

3 后台 CGI 程序的编制

作为实用程序, 对 CGI 接口的基本功能设计 requirements 是: 读取浏览器提交的 FORM 数据; 检查数据的完整性和安全性; 完成需要的操作, 并产生 HTML 响应前台; 友好地处理出错信息。

3.1 环境变量赋值并把收到的信息传给 CGI 程序

从服务器传送数据到 CGI 程序的方法有三种: 通过命令行参数; 通过标准方法输入; 通过特定的环境变量。

在调用 CGI 程序前, 服务器在特定的环境变量中放入信息, CGI 程序从这些环境变量中读取信息。不同的环境变量与 URL 请求的不同部分对应。这种信息传递的方式满足相应的 CGI 标准。如下表所示常用的环境变量:

表1 常用的环境变量及其描述

SERVER_NAME	信息服务器的主机名 DNS 的别名或 IP 地址
GATEWAY_INTERFACE	服务器的 CGI 版本号
SERVER_PROTOCOL	信息协议及其版本
SERVER_PORT	发送信息的端口
REQUEST_METHOD	请求方式
QUERY_STRING	在 URL 中 “?” 后用于指明该运行工具的字符串
CONTENT_TYPE	查询的数据类型
CONTENT_LENGTH	客户机指定内容的长度

要注意的是标准 CGI 工具的输入和输出。一般 HTTP 服务器的信息均在一个标识头之后通过标准输入送给 CGI 工具。服务器所送的信息类型由 CONTENT_TYPE 定义, 而信息的长度为 CONTENT_LENGTH 所定义的字节数, 因此文件不会发送“文件结束”的标志。

3.2 设计应用程序的后台

为了正确处理数据, CGI 程序应该实现以下功能: 根据环境变量 REQUEST_METHOD 的值定位数据或信息; 能对客户发送给服务器的 URL 进行解码; 能解释 URL 转义如产生查询关键字序列及要求的文件名; 为实现交互功能而能够为用户提供空白 FORM 以及处理来自客户的 FORM 的信息。

CGI 程序根据环境变量 REQUEST_METHOD 的值定位数据或信息: 如果请求方法 REQUEST_METHOD 的值为 GET, CGI 程序读入环境变量 QUERY_STRING; 如果请求方法为 POST, CGI 程序就可确定数据来自标准输入<stdin> (<stdin>是标识数据流的文件句柄), 根据 CONTENT_LENGTH 的值确定从<stdin>读入的数据流大小。

对客户发送给服务器的 URL 进行解码，解释 URL 转义如产生查询关键字序列及要求的文件名。

如下是一个简化了的 WinCGI 例程 jiaoyan.exe 的源代码。

首先，在主程序的说明部分为处理 FORM 表单的数据定义全局变量：

```
Global xm as string
```

```
Global mm as string
```

然后，为读取表单数据设计一个 ReadFormData 过程，内容如下：

```
Public Sub ReadFormData ()
```

```
Xm = GetSmallField ("xm")
```

```
Mm = GetSmallField ("mm")
```

```
End Sub
```

其中，GetSmallField() 函数只是简单地扫描，直到发现与输入参数匹配的值，然后返回该相关的值，当函数找不到匹配值时，就产生一个用户定义错误，GetSmallField() 过程代码如下：

```
Function GetSmallField (Key as string) as string
```

```
Dim I as integer
```

```
For I = 0 To (CGI_NumFormTuples - 1)
```

```
If CGI_FormTuples (I).key = key then
```

```
GetSmallField = Trim$(CGI_FormTuples (I).value)
```

```
Exit Function '**DONE**
```

```
End if
```

```
Next I
```

```
Error ERR_NO_FIELD
```

```
End Function
```

在设计好以上代码以后，就可以编写作为整个程序入口的 main 模块了。代码如下：

```
Sub main ()
```

```
On Error Goto ErrorHandler
```

```
InitilizeCGI ' 获取环境变量过程
```

```
ReadFormData ' 读入参数
```

```
,
```

```
CGI_MAIN
```

```
' 调用外部过程处理数据，验证身份，完成操作，并产生响应传递给 HTTP 服务器
```

```
Clear up
```

```
Close #CGI_OutputFN
```

```
Exit Sub ' 程序结束错误处理过程
```

```
ErrorHandler ' 根据捕获的错误类型进行相应处理
```

```
End Sub
```

由简化的代码可知，CGI 程序启动后，先执行 InitializeCGI 过程以获取环境变量，执行 ReadFormData 读入参数 xm 和 mm，然后转入外部数据处理过程 CGI_main，完成数据库访问、查询、统计等工作。本实现将诸多功能的实现集中于一个 CGI 程序。

3.3 CGI 程序把结果返回 HTTP 服务器

CGI 的输出可以有很多类型，如 HTML 文档、图像、纯文本或声音等，还可以是指向其它文档的链接。通常，CGI 程序应该返回标准 HTTP 响应前台，该 HTTP 服务器响应包括首部和实体。

CGI 定义了三个首部: Location, Content-type, Status。在服务器读入它们之后先进行解释, 然后形成适当的 HTTP 头作为服务器应答的组成部分。

其中 Status 的格式为: Status: Status-code Reason。Status-code 表示服务是否成功地满足了前台的请求。服务器直接把它放入应答首部中。

Content-type: 指定服务器返回给前台数据的内容/类型, 用于标识数据; 例如: Content-type: text/html 表示返回给前台的数据是 HTML 文档。

Location 的格式为: Location: URL。Location 的作用是 CGI 告诉另一个实体取回一个对象。当 URL 以 “/” 开头时, CGI 告诉服务器自己取回该对象送给前台; 当 URL 不以 “/” 开头时, CGI 告诉服务器是送重新定向给前台, 即 CGI 给前台的是地址, 由前台自己取回该对象。

从 CGI 程序向 HTTP 服务器返回数据有两种方式: (1) 写入标准输出: 一般服务器借助 CGI 程序输出的头生成服务器的应答头, 连同 CGI 输出的数据返回给前台; (2) 名字以 “npg (nonparsed header)” 开头的 CGI 程序, 它把输出直接送给前台, 不需要服务器作任何处理。

以下给出了一个简单的反馈程序段:

```
Public Sub GenerateResponse ()
Send ("Content-type: text/html")
' 设定输出形式为 HTML 格式
Send ("")
Send ("<html>")
Send ("<head>")
Send ("<title>反馈例程</title>")
Send ("</head>")
Send ("<body>")
Send ("<h1>您的密码正确! </h1>")
' 返回程序处理结果
Send ("</body>")
Send ("</html>")
End Sub
```

以上, CGI 程序实现了向前台反馈处理结果。为实现交互, 可以根据处理结果形成 FORM 表单并发给前台, 再由用户输入信息, 启动 CGI 程序, 如此反复即可。当然, 这大大增加了编程的复杂程度。

4 小结

WWW 与外部连接的传统方法是 CGI 接口, 利用 CGI 程序的功能可以实现前台与后台的交互, 并形成动态显示。同时, 可以通过 IP 地址确认, 用户/口令验证机制实现安全应用。

但是, CGI 模式也有它不足的地方:

- 缺乏通用性, 对每一个具体的应用必须有专门的 CGI 程序;
- 返回结果 HTML 由 CGI 程序硬编码形成, 结果的改变必然要求修改 CGI 程序;
- 这种方法的效率不高, 对于每个服务请求要产生一个进程响应, 数据在进程间经常交换, 对硬件要求较高;
- 对于应用设计的 FORM 编程人员, 必须自己编写相应的程序; 而且 CGI 程序

的结构相当繁琐，为实现交互要反复嵌套调用，因此程序员的负担很重。因此，需要更好的方法以解决 CGI 接口存在的上述问题，这里就不加以讨论了。

吕 峥 硕士研究生。主要研究方向：计算机网络技术和 WWW 信息系统。 **朱逸芬** 教授。主要研究方向：多媒体技术和网络通讯技术。

作者单位： 合肥工业大学微机所 安徽·合肥（230009）

参考文献

- (1) 陈银山，等译. Web 数据库的建立与管理. 北京：机械工业出版社，1997
- (2) 钱 新，等译. 看实例学 HTML. 北京：清华大学出版社，1997
- (3) 张录娥，等译. Web 数据库开发人员指南. 北京：机械工业出版社，1997