**WEB之CGI----CGI详解（原理，配置及访问）**



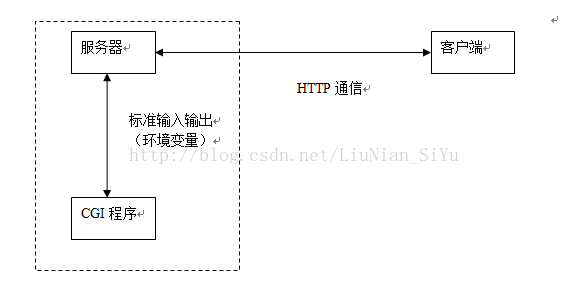
[锥子A](https://blog.csdn.net/weixin_39609623) 2019-01-11 18:49:45 https://csdnimg.cn/release/blogv2/dist/pc/img/articleReadEyes.png 22839 https://csdnimg.cn/release/blogv2/dist/pc/img/tobarCollect.png 收藏 79

分类专栏： [WEB之CGI服务](https://blog.csdn.net/weixin_39609623/category_8604875.html)

**一．基本原理**

**CGI**：通用网关接口（Common Gateway Interface）是一个Web服务器主机提供信息服务的标准接口。通过CGI接口，Web服务器就能够获取客户端提交的信息，转交给服务器端的CGI程序进行处理，最后返回结果给客户端。

**组成CGI通信系统的是两部分**：一部分是html页面，就是在用户端浏览器上显示的页面。另一部分则是运行在服务器上的Cgi程序。它们之间的通讯方式如下图：



服务器和客户端之间的通信，是客户端的浏览器和服务器端的http服务器之间的HTTP通信，我们只需要知道浏览器请求执行服务器上哪个CGI程序就可以了，其他不必深究细节，因为这些过程不需要程序员去操作。

服务器和CGI程序之间的通讯才是我们关注的。一般情况下，服务器和CGI程序之间是通过标准输入输出来进行数据传递的，而这个过程需要环境变量的协作方可实现。

**1.   服务器将URL指向一个应用程序**

**2.   服务器为应用程序执行做准备**

**3.   应用程序执行，读取标准输入和有关环境变量**

**4.   应用程序进行标准输出**

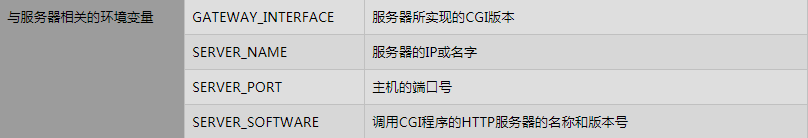
对于Windows系统而言，还可以通过profile文件进行数据传输（如ini文件），但在这里不做研究。环境变量在CGI中有着重要的地位！每个CGI程序只能处理一个用户请求，所以在激活一个CGI程序进程时也创建了属于该进程的环境变量。

**二．环境变量**

       对于CGI程序来说，它继承了系统的环境变量。CGI环境变量在CGI程序启动时初始化，在结束时销毁。

       当一个CGI程序不是被HTTP服务器调用时，它的环境变量几乎是系统环境变量的复制。当这个CGI程序被HTTP服务器调用时，它的环境变量就会多了以下关于HTTP服务器、客户端、CGI传输过程等项目。







       CONTENT\_TYPE:如application/x-www-form-urlencoded，表示数据来自HTML表单，并且经过了URL编码。

ACCEPT:客户机所支持的MIME类型清单，内容如：”image/gif,image/jpeg”

**REQUEST\_METHOD**：它的值一般包括两种:POST和GET，但我们写CGI程序时，最后还要考虑其他的情况。

**1．POST方法**  
       如果采用POST方法，那么客户端来的用户数据将存放在CGI进程的标准输入中，同时将用户数据的长度赋予环境变量中的CONTENT\_LENGTH。客户端用POST方式发送数据有一个相应的MIME类型（通用Internet邮件扩充服务：Multi-purpose Internet Mail Extensions）。目前，MIME类型一般是：application/x-wwww-form-urlencoded，该类型表示数据来自HTML表单。该类型记录在环境变量CONTENT\_TYPE中，CGI程序应该检查该变量的值。

**2．GET方法**  
       在该方法下，CGI程序无法直接从服务器的标准输入中获取数据，因为服务器把它从标准输入接收到得数据编码到环境变量QUERY\_STRING（或PATH\_INFO）。

**GET与POST的区别**：采用GET方法提交HTML表单数据的时候，客户机将把这些数据附加到由ACTION标记命名的URL的末尾，用一个包括把经过URL编码后的信息与CGI程序的名字分开：http://www.mycorp.com/hello.html？name=hgq$id=1，QUERY\_STRING的值为name=hgq&id=1

有些程序员不愿意采用GET方法，因为在他们看来，把动态信息附加在URL的末尾有违URL的出发点：URL作为一种标准用语，一般是用作网络资源的唯一定位标示。

**环境变量是一个保存用户信息的内存区。当客户端的用户通过浏览器发出CGI请求时，服务器就寻找本地的相应CGI程序并执行它**。在执行CGI程序的同时，服务器把该用户的信息保存到环境变量里。接下来，CGI程序的执行流程是这样的：查询与该CGI程序进程相应的环境变量：第一步是request\_method，如果是POST，就从环境变量的len，然后到该进程相应的标准输入取出len长的数据。如果是GET，则用户数据就在环境变量的QUERY\_STRING里。

**3．POST与GET的区别**  
       以 GET 方式接收的数据是有长度限制，而用 POST 方式接收的数据是没有长度限制的。并且，以 GET 方式发送数据，可以通过URL 的形式来发送，但 POST方式发送的数据必须要通过 Form 才到发送。

**三．CGI程序实现步骤**  
**1．从服务器获取数据**  
     C语言实现代码：

1. #include <stdio.h>
3. #include <stdlib.h>
5. #include <string.h>

8. int get\_inputs()
10. {
12. int length;
14. char \*method;
16. char \*inputstring;
18. method = getenv(“REQUEST\_METHOD”); *//将返回结果赋予指针*
20. if(method == NULL)
22. return 1;       *//找不到环境变量REQUEST\_METHOD*
24. if(!strcmp(method, ”POST”))  *// POST方法*
26. {
28. length = atoi(getenv(“CONTENT\_LENGTH”)); *//结果是字符，需要转换*
30. if(length != 0)
32. {
34. inputstring = malloc(sizeof(char)\*length + 1) *//必须申请缓存，因为stdin是不带缓存的。*
36. fread(inputstring, sizeof(char), length, stdin); *//从标准输入读取一定数据*
38. }
40. }
42. else if(!strcmp(method, “GET”))
44. {
46. Inputstring = getenv(“QUERY\_STRING”);
48. length = strlen(inputstring);
50. }
52. if(length == 0)
54. return 0;
56. }

**Perl实现代码：**

1. $method = $ENV{‘REQUEST\_METHOD’};
3. if($method eq ‘POST’)
5. {
7. Read(STDIN, $input, $ENV{‘CONTENT\_LENGTH’});
9. }
11. if($method eq ‘GET’ || $method eq ‘HEAD’)
13. {
15. $input = $ENV{‘QUERY\_STRING’};
17. }
19. if($input eq “”)
21. {
23. &print\_form;
25. exit;
27. }

**PYTHON代码实现**

1. *#!/usr/local/bin/python*
3. import cgi
5. def main():
7. form = cgi.FieldStorage()

       Python代码实现更简单，cgi.FieldStorage()返回一个字典，字典的每一个key就是变量名，key对应的值就是变量名的值，更本无需用户再去进行数据解码！

       获取环境变量的时候，如果先判断“REQUEST\_METHOD”是否存在，程序会更健壮，否则在某些情况下可能会造成程序崩溃。因为假若CGI程序不是由服务器调用的，那么环境变量集里就没有与CGI相关的环境变量（如REQUEST\_METHOD，REMOTE\_ADDR等）添加进来，也就是说“getenv(“REQUEST\_METHOD”)”将返回NULL！

**2．URL编码**

      不管是POST还是GET方式，客户端浏览器发送给服务器的数据都不是原始的用户数据，而是经过URL编码的。此时，CGI的环境变量Content\_type将被设置，如Content\_type = application/x-www-form-urlencode就表示服务器收到的是经过URL编码的包含有HTML表单变量数据。

**编码的基本规则是**：

    变量之间用“&”分开；

    变量与其对应值用“=”连接；

    空格用“+”代替；

    保留的控制字符则用“%”连接对应的16禁止ASCII码代替；

    某些具有特殊意义的字符也用“%”接对应的16进制ASCII码代替；

    空格是非法字符；

    任意不可打印的ASCII控制字符均为非法字符。

例如，假设3个HTML表单变量filename、e-mail和comments，它们的值对应分别为hello、mike@hotmail.com和I’ll bethere for you，则经过URL编码后应为：

filename=hello&e-mail=hello@hotmail.com&comments=I%27ll+be+there+for+you

所以，CGI程序从标准输入或环境变量中获取客户端数据后，还需要进行解码。解码的过程就是URL编码的逆变：根据“&”和“=”分离HTML表单变量，以及特殊字符的替换。

**在解码方面：**

PYTHON代码实现是最理想的，cgi.FieldStorage()函数在获取数据的同时就已自动进行代码转换了，无需程序员再进行额外的代码编写。Perl其次，因为在一个现成的Perl库：cgi-lib.pl中提供了ReadParse函数，用它来进行URL解码很简单：

1. require ‘cgi-lib.pl’;
3. &ReadParse(\*input);

**CGI数据输出方面：**

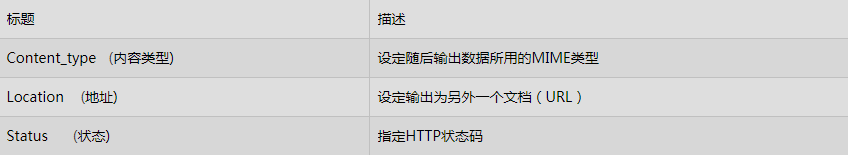
      CGI程序如何将信息处理结果返回给客户端？这实际上是CGI格式化输出。

      在CGI程序中的标准输出stdout是经过重定义了的，它并没有在服务器上产生任何的输出内容，而是被重定向到客户浏览器，这与它是由C，还是Perl或Python实现无关。

所以，我们可以用打印来实现客户端新的HTML页面的生成。比如，C的printf是向该进程的标准输出发送数据，Perl和Python用print向该进程的标准输出发送数据。

（1）   CGI标题

CGI的格式输出内容必须组织成标题/内容的形式。CGI标准规定了CGI程序可以使用的三个HTTP标题。标题必须占据第一行输出！而且必须随后带有一个空行



( 2 )  MIME：

向标准输出发送网页内容时要遵守MIME格式规则：

任意输出前面必须有一个用于定义MIME类型的输出内容（Content-type）行，而且随后还必须跟一个空行。如果遗漏了这一条，服务将会返回一个错误信息。（同样使用于其他标题）

例如Perl和Python：

1. print “Content-type:text/html\n\n”;   *//输出HTML格式的数据*
3. print “<body>welcome<br>”
5. print “</body>”

C语言：

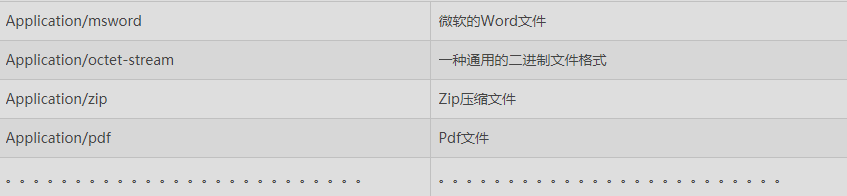
1. printf( “Content-type:text/html\n\n”);
3. printf(“Welcome\n”);

MIME类型以类型/子类型（type/subtype）的形式表示。

其中type表示一下几种典型文件格式的一种：

Text、Audio、Video、Image、Application、Mutipart、Message

Subtype则用来描述具体所用的数据格式。



**Location**：

使用Location标题，一个CGI可以使当前用户转而访问同一服务器上的另外一个程序，甚至可以访问另外一个URL，但服务器对他们的处理方式不一样。

使用Location的格式为：Location：Filename/URL，例如：

1. print “Location:/test.html\n\n”;
3. 这与直接链接到test.html的效果是一样的。
5. print “Location:http:*//www.chinaunix.com/\n\n”*
7. 由于该URL并不指向当前服务器，用户浏览器并不会直接链接到指定的URL，而是给用户输出提示信息。

**HTTP状态码**：

       表示了请求的结果状态，是CGI程序通过服务器用来通知用户其请求是否成功执行的信息码，本文不做研究。

**就是获得getResponseCode(),一般值在0~200为正常，超过视为不正常或特殊意思**

**四．CGI中的信号量和文件锁**

       因为CGI程序时公用的，而WEB服务器都支持多进程运行，因此可能会发生同时有多个用户访问同一个CGI程序的情况。比如，有2个用户几乎同时访问同一个CGI程序，服务器为他们创建了2个CGI程序进程，设为进程A和进程B。假如进程A首先打开了某个文件，然后由于某种原因被挂起（一般是由于操作系统的进程调度）；而就在进程A被挂起的这段时间内，进程B完成了对文件的整个操作流程：打开，写入，关闭；进程A再继续往下执行，但进程A所操作的文件依旧是原来文件的就版本，此时进程A的操作结果将覆盖进程B的操作结果。

为了防止这种情况发生，需要用到文件锁或者信号量。

钥匙文件？

假如有多个不同的HTML可以调用同一个CGI程序，那么CGI程序如何区分它们呢？一个是通过隐含的INPUT标签。不过觉得这个比较麻烦，因为CGI必须经过一系列解码后才能找到这个隐含INPUT的变量和其值。

**五．设置HTTP服务器以兼容CGI**

       用Perl编写的CGI程序后缀为：.pl；Python编写的CGI程序后缀为：.py；而C编写的CGI程序后缀为：.cgi，如果在win下编译出来的是.exe，最好将它重命名为.cgi。这些都是为了HTTP服务能够识别并调用它们。

       当使用appche httpd服务器时，请编辑它的配置文件httpd.conf如下：

       修改AddHandler cgi-script一句为AddHandler cgi-script .cgi .py.pl

**六．关于CGI的C语言库——cgihtml**

       Cgihtml是一个应用非常广泛的C语言编写的CGI库。它提供的功能函数如下：

       Read\_cgi\_input()：获取并解析HTML表单输入，返回一个指向某结构体的指针

       Cgi\_val()：获取每个表单变量的值

       Html\_header():输出HTML标题栏

       Html\_begin():输出HTML文档的开始部分

       H1():输出一行字符，字体为H1

Html\_end():输出HTML文档的结尾部分。

#include “cgi-lib.h”

#include “html-lib.h”

#include “string-lib.h”

**六．后话**

      有的人认为可以用JavaScript来代替CGI程序，这其实是一个概念上的错误。JavaScript只能够在客户浏览器中运行，而CGI却是工作在服务器上的。他们所做的工作有一些交集，比如表单数据验证一类的，但是JavaScript是绝对无法取代CGI的。但可以这样说，如果一项工作即能够用JavaScript来做，又可以用CGI来做，那么绝对要使用JavaScript，在执行的速度上，JavaScript比CGI有着先天的优势。只有那些在客户端解决不了的问题，比如和某个远程数据库交互，这时就应该使用CGI了。

SSI:一种用来动态输出HTML文本的特殊程序。

网页里包含有某个变量，提交给服务器后，只有该变量改变。此时我们希望服务器不要把整个页面内容都发送过来，而只需要告诉客户端的浏览器，哪个变量的值便成什么样了，浏览器会自动更新。

SSI在服务器端运行。

SSI不需要外部接口，它不像CGI从标准输入接收信息。

你浏览你的HTML文档时看不到SSI标记，因为它已经被相应的程序输出所替代。

所有的SSI命令都是嵌入在普通的HTML注释行中的。当服务器无法解释SSI时，它将不解释并直接把文档传给浏览器，由于命令在注释中，故浏览器将忽略它们。而当服务器识别SSI时，它并不将该命令传给浏览器，相反，服务器将从上到下扫描HTML文档，执行每一个嵌入注释的命令，并将命令的执行结果代替原注释。

<! –注释文本-->。服务器将根本不查看注释，除非已启动SSI。

与纯注释不同的是，所有的SSI命令都是以#打头。

<!--#command tagname = “parameter”-- >,command指出服务器做什么，tagname指出参数类型，parameter是该命令的用户定义值。

The currentdate is<! --#echo var = “DATE.LOCAL”-- >，服务器将向浏览器输出时间。

