

# WWW Server 中 CGI 的工作原理及编程方法

罗东川 (中科院网络信息中心数据库工程室)

**摘要:**本文论述了 HTTPD 中 CGI(公共网关接口)的一般工作原理,分析了从 client 端发出请求,到 WWW server 启动 CGI 程序,再到 CGI 程序将运行结果返回给 client,并加以正确显示的各个环节。系统地描述了编制一个 CGI 程序所需的步骤及方法。

**关键词:**CGI WWW Internet

## 1. 前言

一个联网用户不仅可以通过这个网络查询网上其他节点的信息资源,同时也可以方便地将自己所拥有的信息资源提供给网上其他用户共享。目前在 Internet 上存在着多种应用服务系统,如 News、gopher、www 等,针对不同的系统,存在着不同的提供信息资源的方法,而本文所要论述的是如何向 WWW 系统提供信息服务,具体地讲,就是分析了 WWW Server HTTPD 中 CGI 接口的一般工作原理。通过这个接口,用户可以向 WWW 系统提供动态的信息资源。

## 2. WWW 概述

在目前 Internet 上提供的各种服务中,尤以基于 WWW 的应用为最受用户欢迎,这种以 Hypertext 和 Hypermedia 为基础的应用以其能提供图、文、声并茂的多媒体信息和方便、快捷地获取不同节点信息的机制而得到了广泛的应用。这种应用的一个最成功的范例就是 Netscape,目前国内很多用户都是通过 Netscape 来认识 Internet 的,它几乎与 Internet 成了同义词了,可见其影响之大。

类似于 Internet 上的其他应用一样,基于 WWW 的应用也采用 client / server 结构。这种应用通常是由位于 Internet 上的两台不同机器共同完成的,其中一台机器作为 client 端,拥有浏览信息的检索软件如 Netscape;而另一端作为信息资源的提供者,而拥有 server 端的软件。client 端与 server 端通过 HTTP 协议进行通讯,并依靠一种称之为 URL 的统一标识符来寻址不同 server 上的信息。目前 server 端的软件主要有两种: Netscape server 和 NCSA 的 HTTPD Server,由于 NCSA 的 HTTPD 是免费提供的,所以目前国内用户大都是采用此种 Server。

一个联网用户如果想向 Internet 提供自己所拥有的信息资源,他就应首先获得一个 HTTPD 的拷贝,这个拷贝应与自己的主机平台相一致,在正确地安装与配置了这个软件以后,你就可以通过它向 Internet 提供信息服务了。具体提供信息的方式可以大致分为两类;一类是基本的,也就是目前国内大多数用户所使用的方法,即通过 HTML 语言编制一个超文本(Hyper-text)或超媒体(Hypermedia)的文档。在这个文档中包括用户要提供的信息资源和一些链接到其他 Server 上相应文档的链接信息。通过这种方式用户可以向网上提供各类信息资源;如声音、图像、文本等。并且通过系统内部的 URL 寻址机制,可以方便地将位于其他 Server 上的 HTML 文档链接到你的 HTML 文档之中,这种机制确实简单、方便、快捷。可以利用 Internet 网上的已有资源来补充你自己的信息系统。采用这种方式提供信息,它的关键是 HTML 语言,只要掌握了它,用户就可以随心所欲的编制各种文档,然后将其置于 server 端的特殊目录下,这样位于 Internet 网上的任一个用户,只要它具有一个浏览器,就可以访问到你所提供的信息。

但这种方式有一个根本的弱点,即用 HTML 语言编制的文档从本质上来说是一种文本文件,也就是说它是一种固定格式的文件,它提供的信息内容是死的、静止的。这种方式适合于那种信息内容变化不是很快的地方。比如用来对某个单位或个人进行简单介绍。而从提供信息服务的角度,即要求要有动态的查询、统计等功能来说,这种方式是不能满足要求的。为了解决这一问题,在 HTTPD 中提供了一个 CGI(common gateway interface)接口,这个接口的作用就是在 HTTPD 中运行一个外部程序,这一点的意义很大,它极大地丰富了 WWW Server 的能力,建立了 WWW 与外部世界之间的联系,通过它可以编制一些外部程序来完成 HTML 语言不能完

成的特定功能,并使其集成在 WWW 环境之下,形成一个统一的应用。这种外部应用的一个较典型的例子就是数据库查询,可以通过 CGI 编写一个查询数据库的应用程序,用户就可以在 WWW 的浏览器(比如 netscape)中直接查询数据库。这样,通过外部程序就可以向 Internet 提供各种动态的信息资源。

### 3. CGI 的工作原理

CGI 接口是集成于 HTTPD 之中的,当成功地安装了一个 HTTPD 之后,这个 CGI 运行环境就存在于 Server 之中了。构造一个 CGI 应用大致包括下面几步:

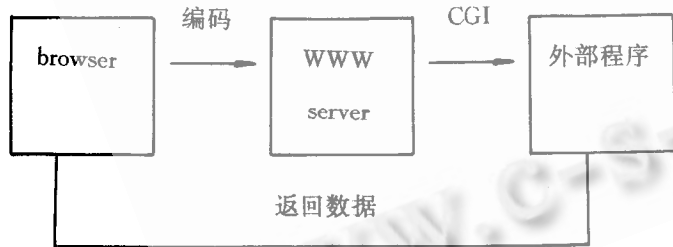
(1) 编制一个称作 fill-out form 的 HTML 文档,这个文档实际是一个与外部程序交换数据的用户界面,通过这个界面,用户可以将参数传递给外部程序,从而控制外部程序的运作,并获得不同的显示结果。

(2) 确定传递参数的不同方式,目前 CGI 程序提供了两种方式: Get 和 Post,在这两种方式中传递参数的机制是不同的。

(3) 由于通过 fill-out form 传递的参数在传递过程中进行了特殊的编码。所以外部程序在获得这些参数之前,需要对其进行解码。

(4) 编制外部应用程序以完成某一个特定功能。

(5) 将 CGI 程序运行的结果正确地在 client 端加以显示。整个过程的逻辑原理如下图所示:



下面就较详细地对各个步骤进行介绍。

① 所谓 fill-out form,实际上是对 HTML 语言的扩充,这种 Form 可以嵌在一般 HTML 文档之中,并且在一般 HTML 文档之中,可以有多个 Fill-out form。只是这种 fill-out form 不能嵌套使用。Fill-out form 的主要作用在于提供一个接口给用户,使得用户可以通过这个接口和 CGI 程序之间进行参数和数据的交换,从而使得用户能够控制 CGI 程序的运行,达到程序与用户之间的交互对话,这一点对于一个信息检索系统来说是必不可少

的。一般 HTML 语言只是提供了对要显示的文档进行正确标识的规则。以使得浏览器能够正确地显示它们,这些方式包括:文本的字体大小、位置、图像的格式以及一些对其他节点信息的链接信息,它并没有提供系统接受用户输入信息的机制。这一点是与一般 HTML 文档只能提供静态信息内容相一致的。而当一个用户通过 CGI 运行一个外部程序时,比如一个数据库查询程序,进行一些动态信息的查询时,那么用户与查询程序之间需要相互传递一些数据,就是必不可少的了。比如用户总要输入一些查询条件来完成自己的查询。Fill-out Form 提供了许多可供用户输入数据和选择数据的域,以及像 check box、radio 和 submit 等类型的按钮,这些都是构造一个交互式界面所不可缺少的元素。在这些输入域中用户可以通过选择的方式或是直接键入的方式来输入自己的数据,并通过按 submit 钮向系统提交输入。由于 Fill-out form 可以嵌在 HTML 文档之中,所以一般都是将两者结合在一起使用,这样就可以构造一个较为满意的界面。

② 第二步是要理解数据的传递方式,即 CGI 是采用什么方式将用户输入的数据传递到系统内部,这一点将直接影响到 CGI 程序的开发,具有非常重要的意义。

在 HTTPD Server 中,系统提供了许多环境变量,当然这些环境变量在不同的 Server 中可能有所不同,下面是一些标准的环境变量:

AUTH-TYPE  
CONTENT-LENGTH  
CONTENT-TYPE

SERVER-PORT  
SERVER-PROTOCOL  
SERVER-SOFTWARE

正是通过这些环境变量,系统达到与外部程序进行通讯的目的,这些变量有些是在安装 Server 时设定的,当然可以更改这些变量的设置,以达到不同的目的。控制一个 CGI 程序运行的参数也是通过这些环境变量来传递的。目前系统提供了两种方式用来传递参数:一是“GET”方式;另一个是“POST”方式,它们在传递数据的机制上有所不同,GET 方式是系统缺省的方式,在这种方式中,当你在 fill-out form 中的 INPUT 字段中输入

了数据,并按 submit 钮之后,各 INPUT 字段的值将被编码形成一个如下的数据串:

```
action? name = value&name = value&name = value.
```

其中 action 是你在 fill-out form 中标识的 URL 地址,每一个 name = value 代表一个值对, name 是 INPUT 字段的名字, value 是你在其中输入的值。& 符号将每一个值对分开, 这样的数据串被称作一个 URL 编码的字符串,因为你输入的各个值是附着在一个 URL 地址后面,并且它们也是按照 URL 格式编码的。? 以后的部分将送入环境变量 QUERY-STRING 中。这种方式的一个缺点是:输入的数据串不能过长,因为它受 shell 命令行参数长度的限制,如果超过这个长度,超过部分将被系统截断。

在 POST 方式中,用户输入的数据采用与 GET 方式相同的编码方式进行编码,只不过它不是作为 URL 地址的一部分来传递,而是作为一个独立的数据体来传递的,在这种方式中, CGI 程序不是通过环境变量 QUERY-STRING 来接受编码输入,而是通过 stdin。并且 server 也不在数据尾部发送 EOF 符号,需要通过环境变量 CONTENT-LENGTH 中的值来确定你需要读的数据长度。这种方式没有输入数据串长度的限制。

③在明白了输入数据的传输原理以后,就可以在你的 CGI 程序中对其进行引用了,但是在你对这些数据引用之前,还有一步必须要作,那就是要对这些数据串进行解码,因为这些数据串都是采用 URL 格式编码的,并且其中采用了 =、& 符号作为分隔符,在 URL 格式中,空格是采用 + 号代替,还有一些字母是采用十六进制表示,这些符号在使用之前都必须去掉或转换成常规表示,将各个 INPUT 字段中输入的值分离出来,然后在 CGI 程序中加以引用。由于编码是固定格式的,所以对其进行解码的程序,也可以是标准的,目前已经有针对性不同种类 CGI 程序的解码程序包,在 Internet 网上可以很容易得到,只需将其集成到你的 CGI 程序中,并对其加以引用即可。如果采用 C 语言编制你的 CGI 程序的话,那么在 HTTPD server 的 cgi-bin 目录下就有现成的 C 函数来完成解码工作。如果你采用 shell 程序或 PERL 语言来开发外部程序,那么你也可以很容易地在 Internet 网上找到相应的解码程序,并不用你自己编制。

④这一步就是采用一种具体的编程语言来编制你的

外部程序。几乎可以采用任一种语言来写你的 CGI 程序,下面是一些例子。

C/C++

Fortran

PERL

TCL

Any Unix shell

Visual Basic

Apple Script

⑤接下来的一个问题是 CGI 程序如何向 Client 返回信息,也就是说当 CGI 程序产生了运行结果,比如查询一个数据库之后所得到的输出,如何将它 在 client 端上正确地显示出来。这个问题也是刚开始使用 CGI 编程的人员容易忽视的一个问题。

一般来说, CGI 程序可以返回多种文档类型,包括图像、声音文件和 HTML 文档。当然也可以返回对其他文档调用的链接信息。在 client 端的 browser 软件中,一般都具有理解上述各类文件格式的能力,但这种理解不是自动的,它需要开发者在返回给 client 端的文档中显示地给予标识。client 端必须知道 CGI 程序返回文档的类型。如此它才能正确地加以显示出来。这就需要一种方法来告诉 client 端 CGI 返回文档的类型,目前是采用在 CGI 程序中,相应于输出语句的前面加一行标识语句。这是一行 ASCII 文本,且与输出体之间要有一行空格。如,当程序返回的是一般 ASCII 文档时,只需在输出语句前加上:

```
content-type :text/plain.
```

如果返回 HTML 文档,那么要加如下一行:

```
content-type :text/html
```

下面是一个具体例子:

```
Content-type :text/html
```

```
<HTML><HEAD>
```

```
<TITLE>out put of HTML form CGI script </TITLE>
```

```
</HEAD><BODY>
```

```
<H1>sample output </H1>
```

```
what do you think of <STRONG>this? </STRONG>
```

```
</BODY></HTML>
```