

CGI程序 设计指南

张移山 编著

万水Web技术精品丛书



中国水利水电出版社
www.waterpub.com.cn

万水 Web 技术精品丛书

主编：裴有福

CGI 程序设计指南

张移山 编著
裴有福 审校



中国水利水电出版社
www.waterpub.com.cn

JS117/07

内 容 提 要

通用网关接口 (CGI) 程序主要用来实现 Web 服务器、浏览器和外部服务程序间的交互。对用户而言, 它提供了双向交互能力, 使得 Internet 漫游更加有趣和人性化。

本书共分十二章, 从比较直观的角度出发, 系统介绍了 CGI 规范的基本原理。同时结合一些具体的实用 CGI 程序, 以 C 语言和 Visual Basic 为编程语言, 介绍了编写 CGI 程序的基本方法和应注意的问题。

阅读本书的读者应该对万维网 WWW 有一定的了解, 具备基本的 Web 网页制作经验, 同时也应具有使用 C 语言或 Visual Basic 编写程序的经验。

图书在版编目(CIP)数据

CGI 程序设计指南/张移山 编著—北京: 中国水利水电出版社, 1998.6
(万水 Web 技术精品丛书)
ISBN 7-80124-740-X

I. C… II. 张… III. 程序设计, CGI-指南, IV. TP312

中国版本图书馆 CIP 数据核字 (98) 第 16055 号

书 名	CGI 程序设计指南
作 者	张移山 编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010)63202266(总机)、68331835(发行部)
发 行	新华书店北京发行所
经 售	全国各地新华书店
排 版	北京门槛创作室 WORD 照排部
印 刷	北京牛山世兴印刷厂
规 格	787×1092 毫米 16 开本 14.25 印张 337 千字
版 次	1998 年 7 月第一版 1998 年 7 月北京第一次印刷
印 数	0001—5000 册
定 价	24.00 元

凡购买我社图书, 如有缺页、倒页、脱页者, 本社发行部负责调换

版权所有·翻版必究

前 言

不管您愿意不愿意、承认不承认, Internet 正在迅速改变着我们的生活习惯和生活质量, 并且必将越来越深刻地影响人类社会的方方面面。作为 Internet 的重要内容, World Wide Web (简称 Web, 缩写为 WWW, 中译为万维网) 已被公认为是 20 世纪的重大科技突破。它以一种简单方便、易于操作的方式链接着世界范围内各种类型的信息和各种层次的人们, 并将为人们构筑更加美好的生活和工作。在这种形势下, 学习和掌握 Web 技术, 就成为越来越多的人的迫切要求和强烈愿望。

《万水 Web 技术精品丛书》正是在这种形势下应运而生的。本套丛书的作者长期关注和研究 Web 技术, 目睹着它的迅猛发展, 也体会到它的意义重大。但同时也注意到许多 Web 使用者, 特别是一些初学者, 在广袤的 Web 空间里不知所措, 找不到重点, 分不清主次, 常常事倍功半。有很多读者, 没有条件或不能经常连入 Internet, 不能及时学习 Web 技术, 反被市面上的一些图书弄得懵懵懂懂。因此, 这套丛书力图系统地全面地介绍 Web 技术的各个方面, 努力使读者能够在短期内迅速掌握 Web 技术。

通用网关接口 CGI (Common Gateway Interface) 程序主要用来实现 Web 服务器、浏览器和外部服务程序间的交互。虽然目前已出现一些可以完全或部分替代 CGI 的技术, 但由于 CGI 程序的跨系统平台和服务器平台的特性, 至少目前 CGI 还是有生命力的, 仍然是 Web 交互技术的支柱。对于大多数 Web 站点, CGI 程序作用巨大, 其固有缺陷并不十分明显。相对于其他技术, CGI 也更容易掌握, 可以使用的编程工具也非常多。因此, 《CGI 程序设计指南》在本套丛书中占有重要地位。

本书共分十二章, 它从比较直观的角度出发, 系统介绍了 CGI 规范的基本原理。同时结合一些具体的实用 CGI 程序, 以 C 和 Visual Basic 为编程语言, 介绍了编写 CGI 程序的基本方法和应注意的问题。

衷心希望读者可以从本书中获得收益, 并在学习和掌握 Web 技术的过程中享受到无穷的乐趣! 本书作者努力作得更好, 但由于水平有限、时间紧迫, 不足与错误在所难免。希望读者和同行专家不吝指教, 以便在未来的版本中予以修改, 使本书真正成为 Web 技术的众望所归之作。



1998.5

目 录

前 言

第一章 通用网关接口 (CGI) 简介	1
1.1 WWW 是如何工作的	1
1.2 CGI 是如何工作的	3
1.3 标准输入(STDIN)和标准输出(STDOUT)	4
1.4 和 CGI 有关的 Web 服务器上的目录	6
1.5 CGI 编程的特点	6
1.6 CGI 的未来	7
第二章 SSI 的使用	10
2.1 SSI 简介	10
2.2 SSI 命令简介	11
2.3 SSI 简单示例	12
2.4 有关 SSI 使用的 Web 服务器的文件设置	13
2.5 config 命令的使用	14
2.6 include 命令的使用	18
2.7 fsiz 命令的使用	18
2.8 echo 命令的使用	19
2.9 exec 命令的使用	21
2.10 if 命令的使用	22
2.11 email 命令的使用	23
2.12 odbc 命令的使用	24
2.13 SSI 命令的安全问题	25
2.14 SSI 命令使用实例	26
2.14.1 给主页添加动态欢迎信息	26
2.14.2 对将来某一日期进行倒计时	28
2.14.3 给 Web 网页添加有关最后访问者的信息	30
第三章 CGI 编程的基本原理	33
3.1 HTML 表单 (Form)	33
3.2 第一个 CGI 程序	35
3.3 MIME 类型	38
3.4 表单 (Form) 的制作	38
3.4.1 TEXTAREA	39
3.4.2 SELECT	39
3.4.3 INPUT	39
3.5 传送表单数据的两种方法: GET 和 POST	42

3.6	URI 译码 (URI Encoding)	45
3.7	URI 解码	47
3.8	利用 CGI 程序生成 HTTP 头信息	54
第四章	环境变量的使用	57
4.1	和 Web 服务器状态有关的环境变量	57
4.2	和浏览器的请求头有关的环境变量	58
4.3	和浏览器状态有关的环境变量	61
4.4	SSI 环境变量和 CGI 环境变量之间的区别	61
4.5	环境变量 PATH	64
4.6	在 CGI 程序中使用环境变量	64
第五章	深入 CGI 编程	70
5.1	CGI 编程语言的选择	70
5.1.1	UNIX SHELL	70
5.1.2	Perl	71
5.1.3	TCL	72
5.1.4	C 语言	72
5.1.5	C++	73
5.1.6	Visual Basic	73
5.2	CGI 程序的基本框架	74
5.2.1	程序的初始化准备工作	74
5.2.2	数据处理	75
5.2.3	程序处理结果的输出	75
5.2.4	程序的结束	76
5.3	CGI 编程的注意事项	76
5.3.1	程序的可维护性	76
5.3.2	程序的安全性	76
5.3.3	CGI 程序和系统平台的无关性	77
5.3.4	CGI 程序和 Web 服务器的无关性	77
5.3.5	CGI 程序输出的一致性	78
5.4	CGI 程序的测试	78
5.5	CGI 程序的出错处理	80
第六章	利用现有 CGI 程序库资源	81
6.1	CGI-UTIL 库	81
6.2	CGIC	92
6.3	GD 库	93
6.4	CGI4VB	95
6.5	其他一些有关 CGI 的网络资源	103
第七章	CGI 编程实例	105
7.1	C 语言编程实例	105

7.1.1	下拉式菜单链接选择器.....	105
7.1.2	主页意见箱.....	110
7.1.3	贵宾留言簿.....	119
7.1.4	给需要身份认证的 Web 资源增加用户	128
7.1.5	主页计数器.....	137
7.2	Visual Basic 编程实例.....	140
7.2.1	用 VB 编写的 CGI 程序处理中文信息.....	140
7.2.2	主页意见箱.....	141
7.2.3	贵宾留言簿.....	149
第八章	客户端用户状态信息的保存.....	156
8.1	利用环境变量 PATH_INFO 和 QUERY_STRING	156
8.2	利用表单的隐含文本框.....	157
8.3	使用隐含文本框的 CGI 程序实例	157
8.4	HTTP Cookie.....	164
8.4.1	什么是 HTTP Cookie	164
8.4.2	HTTP Cookie 工作原理	165
8.4.3	HTTP Cookie 使用时的一些限制	166
8.4.4	用 CGI 程序设置和读取 HTTP Cookie.....	166
8.5	使用 HTTP Cookie 的计数器	168
第九章	动态网页的制作技术.....	172
9.1	普通 CGI 程序.....	172
9.2	Client Pull	174
9.3	Server Push	175
9.4	外部程序	177
第十章	Web 网页和数据库的连接.....	178
10.1	VB 数据库 CGI 程序	179
10.1.1	VB 的数据库编程能力	179
10.1.2	用 VB 编写处理数据库的 CGI 程序.....	179
10.2	IDC	187
10.2.1	IDC 简介.....	187
10.2.2	IDC 文件和 HTX 文件.....	187
10.2.3	用 IDC 实现贵宾留言簿.....	190
10.3	ASP.....	192
10.4	Livewire.....	195
10.4.1	Livewire 简介	195
10.4.2	用 Livewire 实现贵宾留言簿.....	196
第十一章	Image Map 的使用.....	199
11.1	什么是 Image Map	199
11.2	Web 服务器内置的 Image Map	199

11.3	客户端浏览器内置的 Image Map.....	200
11.4	用 CGI 程序实现 Image Map.....	202
第十二章	Windows CGI 规范.....	211
12.1	Windows CGI 的工作原理.....	211
12.2	CGI 数据文件.....	212
12.3	Win-CGI 程序的输出.....	215
12.4	Win-CGI 编程	215
附录 A	MIME 类型	216
附录 B	服务器的状态码	219

第一章 通用网关接口（CGI）简介

利用通用网关接口（CGI），Web 服务器可以执行一些外部的小程序，可以将这些外部小程序所产生的输出结果和 Web 服务器所管理的静态文本、图像和声音融合在一起传给相应的 Web 浏览器。Web 服务器和 CGI 程序的结合大大增强了整个万维网（World Wide Web, WWW）的信息提供和交互能力。

通用网关接口（CGI）只是规定了一个标准的接口规范，只要遵守这个标准规范，程序开发人员就可以利用各种编程工具，例如 Perl、C、Fortran、Visual Basic 等进行 CGI 编程。利用 CGI 程序，Web 服务器可以收集 Web 页面中表单内的信息，可以对特定的 Web 网页进行访问计数，可以根据浏览器的请求对数据库进行检索和更新。

本章将简要介绍 CGI 的一些基本概念。读完本章后，读者将对 CGI 如何工作的全貌有一个基本的了解。

本章主要讨论以下问题：

- WWW 是如何工作的；
- CGI 是如何工作的；
- 标准输入和标准输出；
- Web 服务器上和 CGI 程序有关的目录；
- CGI 编程的特点；
- CGI 的未来。

1.1 WWW 是如何工作的

目前流行全球的因特网（Internet）热有很大一部分是由 WWW 的盛行而引起的，WWW 的实现使得以前基于字符方式的 Internet 信息交流方式进入了图形和多媒体时代。

WWW 其实是一个基于超文本的信息查询和交互系统，最初是由欧洲粒子物理实验室为满足世界各地的研究人员交流信息而开发的。它采用超文本和多媒体技术，将不同文件通过关键字建立链接，为用户提供一种非线性的查询方式。在一个超文本中，一个关键字链接着另一个与该关键字有关的文件，该文件既可以在同一台主机上，也可以在 Internet 的另一台主机上。同样，被链接的文件也可以是一个超文本文件。超文本文件可以把不同类型的文件，如文本、声音、图像和视频等连接起来。即使是一个超文本文件，也可以包含各种声音、图像和视频等。这样，借助于这类超文本文件的链接，人们可以把各类信息组织起来，形成一个遍及全球的万维网 WWW，其工作方式如图 1-1 所示。

如果您已经使用过 Windows 应用程序的帮助文件，那么实际上您已经和超文本打过交道并享受到超文本所带来的优越性了。

发生在计算机网络上的一切活动都必须遵守一定的协议，WWW 也不例外。Web 服务器和浏览器间所要遵守的协议是超文本传输协议（HTTP）。网络协议通常是分层次的，一层网络协议建立在下一层网络协议之上并利用下一层网络协议所提供的功能来实现。事实上关于 WWW 的超文本传输协议是位于整个国际互联网协议栈的最上层即应用层，如图

1-2 所示。

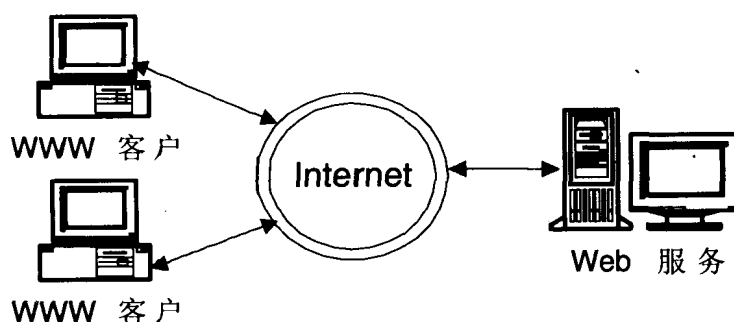


图 1-1 WWW 的工作方式

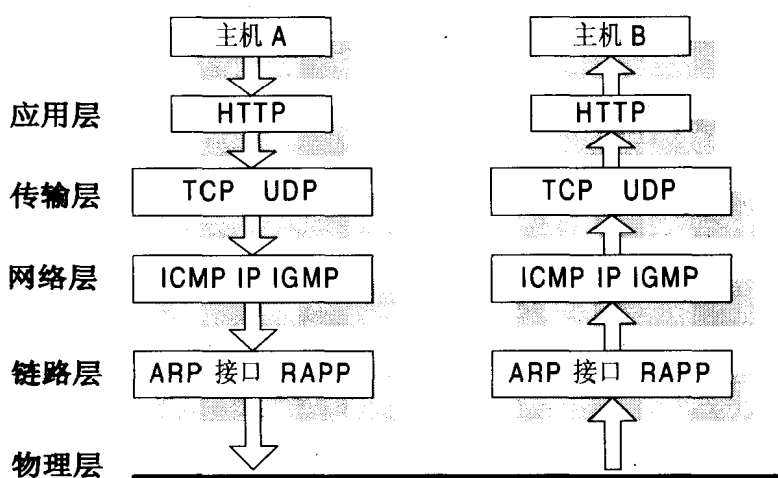


图 1-2 WWW 的协议栈

WWW 采用客户/服务器的工作方式。客户/服务器的计算模型在现今的计算机技术领域已经被应用得非常广泛，WWW 更是一个成功的例子。一般来说，Web 浏览器为客户端，负责提供用户界面，向用户显示服务器所提供的信息。而服务器负责生成查询结果，并把查询结果传送给客户端。

当您激活了 Web 浏览器并准备访问 WWW 上的一个站点时，在客户端的 Web 浏览器和 Web 服务器间将会发生下列事件。

(1) Web 浏览器根据您提供的 URL 和相关的 Web 服务器接触。

URL (Universal Resource Locator) 是一个非常重要的有关 WWW 的概念。有了 URL，Web 浏览器才知道要到哪里去、如何到达目的地以及到达目的地以后又干些什么。从技术的角度来看，URL 是用已有的协议来表示 Internet 上存在对象的 URI (Universal Resource Identifier) 的一种形式，它通常具有以下结构：

`http://<host>:<port>/<path>?<search-part>`

其中 http 表示超文本传送协议，<host>为服务器的 Internet 域名，<port>为服务器的 TCP 端口，如果在 URL 中没有指明端口，浏览器将默认为 80。

(2) 如果能和 Web 服务器联系上, 浏览器将把整个 URL 传送给 Web 服务器。

(3) Web 服务器将根据 URL 所提供的路径查询 Web 浏览器所需要的资源, 并将该资源传送给 Web 浏览器。

(4) Web 服务器中断和浏览器的链接。

(5) Web 浏览器显示 Web 服务器传回的 Web 页面。

Web 浏览器将根据收到文件中的 HTML (超文本描述语言) 代码显示 Web 页面。HTML 代码描述了 Web 页面在浏览器窗体中的显示方式。它是一种描述性语言。例如为了使问候语 “Good Morning” 能以斜体方式出现在浏览器窗体中, 就应该用 HTML 代码将它标记为 “<I> Good Morning </I>”。<I>表示接下来的文本为斜体, </I>表示斜体部分的结束。

通过上面的描述我们现在已经了解到一个 Web 页面是如何被服务器提交的。此时理解超文本链接的工作方式就变得简单了。当用鼠标点取了一个超文本链接时, 浏览器就根据有关超文本链接的 HTML 语句中的 URL 和另一个服务器或超文本文件发生联系。

1.2 CGI 是如何工作的

通用网关接口 (CGI) 是 Web 服务器和您编写的外部程序之间的一个接口。CGI 可以使您编写的程序处理 WWW 上客户端送来的表单和数据并对此作出某种反应。这种反应可以是 HTML 文件、图片、声音、视频等可以在浏览器窗体上出现的任何数据。通用网关接口有两个含义。首先它为客户端用户通过 WWW 获取 Internet 上的其他服务, 例如 Archie、WAIS、Database 等提供了一个接口, 虽然不能直接在 WWW 上访问这些服务, 但是您可以通过一个 CGI 程序和 Web 服务器访问它们。其次它就是您编写的程序和 Web 服务器间的接口标准。您所编写的处理客户端有关信息的程序必须遵循这个标准。

本节将讨论 CGI 是如何在客户端的 Web 浏览器 (如 Netscape Navigator) 和 Web 服务器 (如 NSCA HTTPd) 间工作的。从上一节对 WWW 的介绍可知, 当客户端的浏览器向 Web 服务器请求一个 HTML 文件时, 服务器在收到请求后就去寻找这个文件并将找到的文件传送给客户端。而当客户端的请求是一个 CGI 程序时, Web 服务器将激活客户端所请求的 CGI 程序。此时 Web 服务器所起的作用就好像是客户端和 CGI 程序间的中间人, 如图 1-3 所示。CGI 程序的工作一般可以分为以下几个步骤。

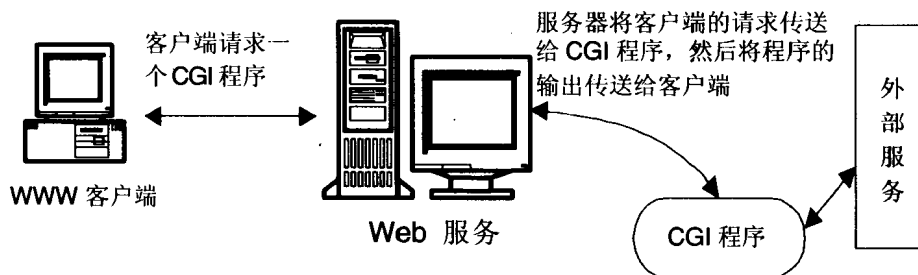


图 1-3 CGI 的工作原理

1. 客户端发出请求

首先是客户端的浏览器发出一个请求给 Web 服务器。如果这个请求是一个普通的文件 (例如 HTML 文件、GIF 或 JPEG 文件), Web 服务器就将文件直接传送给客户端的浏览

器。如果这个请求是一个 CGI 程序或外部应用，Web 服务器将激活相应的 CGI 程序。例如客户端传送来的请求可能是对一个数据库的检索，此时一个相应的 CGI 程序将被 Web 服务器激活，这个 CGI 程序将根据客户端要求的准则去检索相应的数据库，并通过 Web 服务器将结果传送给客户端。

2. Web 服务器激活 CGI 程序

当 Web 服务器发现客户端浏览器传送来的请求不是一个普通文件而是一个 CGI 程序时，Web 服务器就要激活这个 CGI 程序并执行它。

在这个 CGI 程序被执行前，Web 服务器要为该 CGI 程序设置一些环境变量。这些环境变量被服务器用来向 CGI 程序传递一些非常重要的信息，例如当前 Web 服务器的状态、谁发出的调用等等。Web 服务器为 CGI 程序所设置的环境变量的使用和正常的环境变量的使用没有任何区别。当 CGI 程序运行结束时，Web 服务器为它设置的环境变量也随着消失。每个被激活的 CGI 程序都有自己唯一的一组环境变量。事实上对于一个非常繁忙的 Web 服务器，可能同时会有同一个 CGI 程序的很多个进程在运行，这时每个 CGI 程序的进程都有自己的运行环境，互不影响。

在以后的章节本书将继续讨论通用网关接口的环境变量，在此处只要了解到这些环境变量是存在的并包含有可以被 CGI 程序读取的重要信息就行了。

3. CGI 程序对客户端的请求做出反应

CGI 程序通过环境变量或其他途径读取 Web 服务器所提供的客户端的输入数据并对这些数据进行处理，如果有必要 CGI 程序还要和一些外部的服务例如 Archie、WAIS 和 Database 等发生联系。最后 CGI 程序将通过 Web 服务器将处理结果传送给客户端。CGI 程序可以根据需要产生各种类型的数据。

4. Web 服务器将 CGI 程序的处理结果传送给客户端

Web 服务器收到了 CGI 程序的处理结果后将对 CGI 程序的输出进行检查，如有必要，将为 CGI 程序的输出结果加上一些 HTTP 协议所必须的 HTTP 头信息。检查完成后将把 CGI 程序的输出传送给客户端。

5. Web 服务器中断和客户端浏览器的连接

6. Web 浏览器将 CGI 程序的输出显示在浏览器的窗体上

从以上的介绍可以看出，CGI 程序的工作要比普通的 WWW 工作要复杂一些。

1.3 标准输入(STDIN)和标准输出(STDOUT)

一般来说大部分情况下 CGI 程序都是一些小应用程序，编写 CGI 程序并不比编写普通的程序难。初次接触 CGI 程序的人都觉得它很神秘，它怎么就可以处理客户端的表单并进行数据库的检索的呢？在这里最重要的是理解，是对通用网关接口规范 CGI 的理解。Web 服务器和 CGI 程序之间不仅仅通过环境变量和 Web 服务器交换信息，大多数的数据交流都是通过标准输入和标准输出进行的。

STDIN 和 STDOUT 是表示标准输入和标准输出的两个文件句柄。一般主机系统所产生的进程都拥有这两个文件句柄，并且都已经被打开。一个程序可以通过向标准输出(STDOUT)写实现向屏幕输出信息，通过标准输入(STDIN)读从键盘获取信息。一个

程序如果不从键盘而从其他地方获取输入信息，它的标准输入（STDIN）就被重新定向；一个程序如果不向屏幕而是向其他地方输出信息，那么它的标准输出也被重新定向。

例如在 UNIX 系统下敲入命令 `ls`，进程 `ls` 将把它的输出写到标准输出即终端屏幕上。如果敲入命令 `ls > file1`，屏幕上将看不到任何信息，因为通过特殊的 `shell` 字符“>”，进程 `ls` 的标准输出被定向到名称为 `file1` 的文件上，以前在屏幕上输出的信息这次将被写到文件 `file1` 中去。

一个程序的标准输入/输出 (STDIN/STDOUT) 也可以和另一个程序的标准输出/输入 (STDOUT/STDIN) 相连, 这通常被称为管道连接。例如在 UNIX 系统下敲入命令 `ls | sort`, 就可以将命令 `ls` 的标准输出和命令 `sort` 的标准输入相连了, 命令 `sort` 执行时将把命令 `ls` 的输出作为它的输入。

对于一个 CGI 程序来说，当 CGI 程序被服务器激活以后，它的标准输入（STDIN）被连接到 Web 服务器的标准输出（STDOUT）上，而 CGI 程序的标准输出（STDOUT）则被连到服务器的标准输入（STDIN）上，这就形成了完整的管道循环。从 CGI 程序这一方来看，它从标准输入读取信息也就是从 Web 服务器的标准输出读信息，而它向标准输出写信息也就是向 Web 服务器的标准输入写信息。

Web 服务器一般将客户端传送来的信息放在它的标准输出和相关环境变量中，而 CGI 程序则从环境变量和它的标准输入（也就是 Web 服务器的标准输出）获取所需的信息，程序的最终输出结果则被写向它的标准输出（也就是 Web 服务器的标准输入）。Web 服务器将从它的标准输入（也就是 CGI 程序的标准输出）获取 CGI 程序的输出结果并将它传送给客户端。客户端、Web 服务器和 CGI 程序之间的信息交流如图 1-4 所示，显然，Web 服务器就像是客户端和 CGI 程序间的中介人。

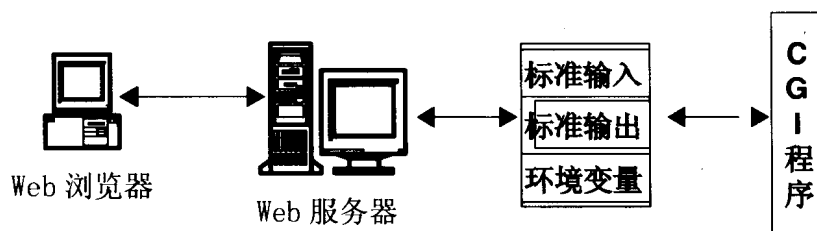


图 1-4 CGI 程序和 Web 浏览器间的信息交流

Web 服务器、CGI 程序间的这种标准框架在 UNIX 系统下和微软视窗环境中的字符方式下可以工作得非常好，因为此时系统所产生的所有进程都可以存取标准输入和标准输出。但对于微软视窗环境图形方式下的程序就不行了，这是由于它们无法存取标准输入和标准输出。为了解决这一问题，微软在 WIN32 系统中创建了另一类型的标准输入和标准输出，程序可以通过调用 WIN32 API 函数在图形方式下存取标准输入和标准输出，不过这就意味着使用这类标准输入和标准输出的 CGI 程序都必须是 32 位的。

微软视窗环境下的其他一些 Web 服务器（例如 Website）则使用另外一种特殊的技术，即利用 INI 文件来实现 Web 服务器和 CGI 程序间的数据交流。采用这种被称为 WIN-CGI 规范编写的 CGI 程序通常只能在部分 Web 服务器上运行。本书将在以后的章节对其进行介绍。一般支持 WIN-CGI 的 Web 服务器将客户端的输入和有关的状态信息写入一个 INI

文件中，而 CGI 程序则从该 INI 文件获取相关信息。

在进行 CGI 编程时，您只要使您的 CGI 程序从标准输入和环境变量中获取客户端提供的信息，把要传送给客户端的输出结果写入标准输出，剩下的信息传递工作 Web 服务器将自动为您完成。

1.4 和 CGI 有关的 Web 服务器上的目录

通常，一个 Web 服务器拥有两个目录树（一个目录树由一个根目录和根目录下的所有子目录组成）。为了将 WWW 资源和 Web 服务器的管理资源分开，对于 UNIX 系统来说，存放 WWW 资源的目录树通常叫 document root，存放服务器系统管理资源的目录树通常叫 server root。

Web 服务器识别 CGI 程序有两种方式。当它发现客户端送来的请求是一个特殊目录下的文件时，就认为该文件是一个 CGI 程序，或者当它发现客户端传送来的请求是带有特殊后缀名（例如 .CGI）时，就认为它是一个 CGI 程序。两种方式的使用取决于 Web 服务器本身和服务器管理员（Webmaster）的个人习惯。相对应于第一种方式，CGI 程序通常被放在 document root 目录树的一个子目录中。比如 document root 树的根目录为 /var/www/docs/，所有的 CGI 程序位于目录 /var/www/docs/cgi-bin/ 中，这个目录通常被称为 CGI 目录。相对应于第二种方式，所有位于 /var/www/docs/ 及其子目录下带后缀名 .CGI 的文件都是 CGI 程序。由于 CGI 程序可以存取系统资源，为了安全和管理上的方便，大多数服务器管理员都使用第一种方式，第二种方式即使能用也不采纳。

现今的很多 Web 服务器都支持多 CGI 目录，即可以在目录树 document root 下定义多个 CGI 目录。如果 Web 服务器还支持虚拟服务器，您还可以定义多个 document root 目录树，每个 document root 目录树将对应一个虚拟服务器。

在目录树 sever root 下包含了所有 Web 服务器管理员（Webmaster）和系统管理员（System Administrator）需要负责的有关 Web 服务器的管理资源。虽然只有服务器管理员和系统管理员才能对它们进行设置和改动，但了解它们对 CGI 编程还是有很大帮助的。

在 server root 目录树下通常有 log 和 conf 两个子目录（各种 Web 服务器的名称都大同小异），在 log 目录下存放的都是有关服务器运行的各种日志文件，和 CGI 编程有关的文件一个是 error.log 文件，该文件记录了服务器运行时所发生的错误以及客户端对服务器所采取的一些错误行为，这个文件对于 CGI 程序的调试有一定的帮助作用。另一个文件是 access.log 文件，它详细记录了客户端对服务器资源的访问和使用，可以利用该文件编制一些对 Web 服务器受访问的情况进行分析和统计的 CGI 程序。在 conf 目录下则存放了一些服务器的配置文件，CGI 程序的正确运行还有赖于这些文件的正确设置。目前大多数商业 Web 服务器都提供了方便的图形菜单来管理这些文件。

1.5 CGI 编程的特点

由于 CGI 程序的运行环境和普通程序的运行环境有很大的不同，和普通编程相比，CGI 编程有着自身的特点。

首先，CGI 程序必须考虑程序运行时的安全性。您编写的 CGI 程序是准备放在 Web

服务器上供大家使用的。也许会有成千上万的人使用这个程序，在这当中也许会有很多好奇心特别强的人，他们十分关心您编写的 CGI 程序的安全漏洞并千方百计地想寻找这样的漏洞，在找出这些漏洞以后他们就有可能攻击您的系统。您不能阻止他们这么做，您唯一能做的是在 CGI 程序中消除这些漏洞。对于那些用解释性语言如 Perl 编写的 CGI 程序，其源代码比用编译性语言编写的 CGI 程序的源代码更容易在网上获得，对于这类程序更应该注意安全性的问题。

其次，CGI 程序获取信息和输出信息的方式和普通程序不一样。

在编写 CGI 程序之前，您必须了解 Web 页面中的超文本链接或表单中的数据是如何被传给 CGI 程序的。在设计 CGI 程序时首先要设计好 Web 页面上的客户端用户界面。Web 页面上的数据传送给 CGI 程序的方法有两种，您必须对这两种方法的优缺点有所了解。

而且，WWW 上存在着各种不同类型的 Web 浏览器，例如 Netscape、MSIE 和文本方式的 Lynx 等等。有的浏览器除了支持标准的 HTML 规范外还对其做了自己的扩充，您编写的 CGI 程序必须注意各类浏览器之间甚至同类浏览器的不同版本之间的区别。

最后，不像普通的程序，CGI 程序是工作在无状态的运行环境中。这种无状态的运行环境有两个特征：(1) 不提供永久性的上下文信息，即在会话连接期间不保存上下文信息；(2) 系统可能会拥有同一 CGI 程序的多个进程，这样，一旦 Web 服务器响应了客户端一个运行 CGI 应用程序的请求，Web 服务器在 CGI 程序运行结束后将立刻切断与发出请求的客户端的连接。如果这个连接重新建立，上一次连接的状态信息将彻底丢失。同时同一个 CGI 程序可能会有多个拷贝在系统中运行，这在某些情况下也会带来问题。

例如您正在建立一个拥有多个网页的 Web 站点，其中的一页可能会在其他网页上产生变化，除非采取一些特殊的步骤，否则在某一网页上发生的变化在下次存取它时将看不到这些变化。所以您的 CGI 程序必须采取一些特殊的方法来跟踪上一次发生的变化，一个最普通的方法是让 CGI 程序将最后一次的变化信息存入一个文件中，不过这个方法有个缺陷，可能会有很多客户同时调用 CGI 程序，CGI 程序将无法确定是谁调用了它。

解决问题的另一个方法是使用 HTML 表单的隐藏文本框。CGI 程序将这一次的状态信息传送给客户端 Web 页面的隐藏文本框。当客户端再次调用其他网页时，CGI 程序就可以从隐藏文本框中获取上一次的状态信息。当然这一方法也有缺陷，虽然 Web 网页将不向客户端用户展示隐藏文本框的内容，但用户却可以用浏览器的 View Source 的菜单命令看到它的内容，这就使得客户端用户可以改变传送给 CGI 程序的数据。使用 HTTP Cookie 则是一个更好的办法，本书的第八章还将详细讨论这个问题。

运行在无状态环境中的 CGI 程序可能同时会被多个浏览器唤醒。一个 CGI 程序可能会修改某个文件，而当该 CGI 程序同时有多个拷贝在运行时，就可能出现有多个进程试图对同一个文件进行读写的情况。此时如果不采取一些特殊的保护措施，某些数据就有可能丢失。当两个进程同时打开一个文件时，最先关闭文件的进程所保存的信息总是被最后关闭文件的进程所保存的信息所覆盖。为了解决这个问题，可以设计一个特殊的句柄数据库用来锁住正在进行读写操作的文件，而此时其他 CGI 程序的进程只能等待文件的有关读写操作完成以后才能打开文件。

1.6 CGI 的未来

WWW 的出现为许多公司提供了巨大的商业机会，为了占领市场的最高点以使自己处

于有利的位置,很多大公司都纷纷推出自己的新技术,例如 Java、Javascript 和 VBscript 等等。这就使得 WWW 的发展日新月异,从而不可避免要对 WWW 上最早出现的 CGI 产生一定的冲击。

Java 语言是太阳微系统公司 (Sun Microsystem) 推出的一种和平台无关的编程语言。Java 源程序被编译以后产生的并不是执行代码而是一种中间代码,该中间代码在执行时需要 Java 虚拟机的解释才能运行。正是由于 Java 虚拟机的存在才使得 Java 语言具有和系统平台的无关性。WWW 只是 Java 语言应用的一个方面。通常以 Java applet 的形式出现。

Javascript 是 Netscape 公司和太阳微系统公司联合推出的一种脚本编程语言,可以将它看成 Java 语言的一个子集。用 Javascript 编写的小程序可以被包含在 Web 网页内并在浏览器上执行,这就使得 Web 网页不用和服务器打交道就可以具有很多动态效果。

VBscript 是微软推出的一种和 Javascript 类似的脚本编程语言,不过比 Javascript 更简单、更容易使用。它是 Visual Basic 的一个子集,熟悉 Visual Basic 的编程人员可以很快就掌握它。

Java applet、Javascript 和 VBscript 可以实现以前很多用 CGI 才能实现的功能。但它们和 CGI 程序仍然有着本质的不同。Java applet、Javascript 和 VBscript 所编写的程序是先被传送到客户端然后在客户机上运行,而 CGI 程序是在服务器上运行以后再将结果传送给客户端。这就决定了 Java applet、Javascript 和 VBscript 还不能完全代替 CGI 程序。

由于 Java applet、Javascript 和 VBscript 充分发挥了客户机的潜力,降低了服务器的负担,在实际 CGI 编程中能用 Java applet、Javascript 和 VBscript 的地方要尽量使用它们而不用 CGI,最好能将二者结合起来使用。例如您的某一 Web 网页是利用 HTML 表单向 Web 服务器上的数据库插入或更新记录。若这一功能都用 CGI 程序来实现,CGI 程序首先要检查输入数据的正确性。若不正确,则向客户端用户返回一个错误信息,若正确则插入数据。其实完全可以用 Javascript 或 VBscript 编写的小程序在客户端就完成用户输入数据的正确性检查,如果用户输入的数据不正确,就不能调用 CGI 程序更新数据库。显然,这样做较前者可以缩短网络的延迟,降低对网络带宽的使用要求和服务器的负担。第七章的主页意见箱就是一个 CGI 程序和 Javascript 程序联合使用的例子。

目前还有一种完全可以取代 CGI 的技术,即 ISAPI。一个 ISAPI 应用程序被某一 Web 网页激活后将完全驻留在内存中以等待其他客户端对它的调用,这就避免了在 CGI 的运行环境中为产生并发进程所付出的额外开销,缩短了服务器的响应时间。和 CGI 程序不同,ISAPI 应用程序和 Web 服务器装载在同一地址空间,这就使得 ISAPI 程序不用借助于环境变量和标准输入/输出就可以和 Web 服务器通信。当然,这样的高性能是有条件的,所有的 ISAPI 程序都必须是安全的多线程程序,而一个 ISAPI 程序发生错误时可能会引起整个 Web 服务器的崩溃。ISAPI 应用程序只能运行在微软的 Web 服务器上或其他厂商支持 ISAPI 的服务器上。

NSAPI 则是 Netscape 公司推出的和 ISAPI 类似的技术。NSAPI 应用程序也只能运行在 Netscape 公司的系列 Web 服务器或其他支持 NSAPI 的 Web 服务器上。

虽然 ISAPI 或 NSAPI 程序和 CGI 相比具有很高的性能,但开发它们需要用到专门的知识 and 工具,掌握它们对一个普通 Web 开发人员并不是一件容易的事情。而且 ISAPI 或 NSAPI 只能运行于特定的 Web 服务器平台。而 CGI 就不同了,一般开发人员,不管他是

精通 C、Perl、Fortran、Visual Basic 等语言还是 Delphi、PowerBuilder 等开发工具，只要熟悉了 CGI 体系的规范，很快就可以开始 CGI 编程并享受到制作动态网页的乐趣。开发人员甚至可以不用编写一些具体的 CGI 程序细节，因为在 WWW 上可以很容易找到满足他要求的 CGI 程序或 CGI 程序库。

由于 CGI 的易学性、易用性和跨服务器平台性，它还将在 WWW 上占有一席之地。