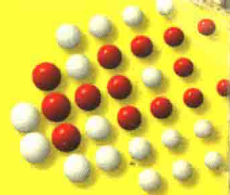


*Teach Yourself
CGI Programming
in a Week*



全面介绍了如何用
CGI进行编程

每位读者均可从书中找到
适合自己口味的内容

(美) Kate Gregory 著
徐丹 唐程杰 等译

CGI 程序设计

自学通



机械工业出版社

西蒙与舒斯特
国际出版公司



sams
net

CMP

CGI 程序设计自学通

(美) Kate Gregory 著

徐 丹 唐程杰 等译

毅 弘 审校

机 械 工 业 出 版 社

西蒙与舒斯特国际出版公司

译者序

网络已成为当今人们生活中不可缺少一个组成部分,因为通过它们不但可以足不出户地获取遍布全球的各种信息,而且可以得到各种日常服务。网络信息传递的快速性、安全性以及价格低廉性使其应用得到迅速发展。然而,要将自己的信息发布到网络供其他人使用,往往要涉及到 Web 页的编制工作。随着网络技术的不断发展,只含有静态页的 Web 站点已不能满足要求。Web 页所要提供的交互式操作已经使原来用于编写 Web 页的 HTML 落伍。

目前为了在 Web 站点上创建动态应用程序,以适应信息的动态变化,已经推出了数量众多的 Web 编程方法。本书集中阐述了 CGI(Common Gateway Interface)编程。因为 CGI 可以安装在大多数 Web 服务器上,它几乎可以用任何一种语言来进行工作,同时 CGI 本身也是一个独立的平台,所以它应该成为创建 Web 应用程序的出发点。

本书作者介绍了 CGI 的技术概貌和 HTTP 协议,包括的许多例程全面阐述了 Web 应用程序的编程方法。全书安排读者一周学完,每天两章。内容由浅入深,不但介绍了 CGI 的工作原理及其技术基础,而且还用众多的例子阐明了从 Web 浏览器接收数据及发送数据的方法;使用 gd 库动态地创建 GIF 图像、使用 CGI 来创建定制错误响应页面以及为邮件网关创建 CGI 的方法;同时,还讨论了创建复杂应用程序的重要工具,讨论了数据库与 CGI 之间的关系,并介绍了结构化查询语言的使用方法;CGI 安全性的阐述为读者创建安全可靠的应用程序提供了基本保证。本书的最后简要地介绍了一些 CGI 的替代产品,使读者可以有更多的选择。提供的附录为读者正确地使用 CGI 编程提供了丰富的材料。读完本书后,读者将会对 CGI 编程有一个全面的了解,并能着手为自己的 Web 站点创建具有相当复杂程度的应用程序。

本书是集体劳动的结晶,参加本书翻译的除封面署名外,还有陈辉、郑明峰、姚正思、沈毅明、汪宝川、高朴真、王立、顾仁、刘航、方东、许胜利、冯卫国、黄丽云、张梦天、程浩、高雪松、余一鸣、丁易伟、汪平、柳飞驰、林旭、刘文燕、扬仲贤、陆春生、肖君、韩志学、张露、徐子望、李铁山、刘健、赵志刚、张忠荣和蒋才国等。

因译者水平有限,错误之处在所难免,敬请读者和同行批评指正。

译者

1998 年 4 月

前 言

现在,只含有静态页面的 Web 站点已不能满足需要了。也许用户正在修建一座联机购物中心,或者正在创建 Web 上的调查表单,或是只是建立一个站点,用来接受用户的输入并且根据用户的偏爱来创建页面。无论用户的目的是什么,如果他想在 Web 上创建真正的应用程序,仅仅拥有 HTML 是远远不够的。

当今,当 Web 的开发者在 Web 站点上创建动态应用程序时,将面对数量众多的、使人眩目的选择,本书将集中论述 CGI(Common Gateway Interface)这一可供 Web 使用的第一个应用程序接口。

CGI 有众多的理由成为读者学习创建 Web 应用程序的出发点。CGI 安装在大多数 Web 服务器上,它几乎可以用任何一种程序语言来进行工作。CGI 也是一个独立的平台。

本书提供了 CGI 的技术概貌和 HTTP 协议。书中包括了许多例程,读者可在自己的站点上使用或修改它们。另外,本书也讨论其他几种可用来建立 Web 应用程序的技术,并且说明为什么在一些情况下读者更乐意使用这些技术。

本书的第 1 章描述了 CGI 的工作原理,并为读者介绍了 CGI 的历史。第 2 章中,将讨论几种能用来创建 CGI 程序的编程语言。

本书的第 3 章讨论了 HTML 表单和应用 JavaScript 使表单目录合法化的方法。第 4 章讨论如何从 Web 浏览器接收数据并把它们发送给了一个 CGI 程序。

第 5 章论述 CGI 的基本技术基础,并对 HTTP 协议进行概述。第 6 章讲述 Server Side Includes,它将为开发者提供一种使用动态目录的方法,这种方法不需要编写完整的应用程序。

第 7 章的内容将使读者逐步深入了解三个例程,读者可以在自己的站点上使用它们。第 8 章将说明读者如何使用 gd 库在自己的网上动态地产生 GIF 图像。

第 9 章介绍如何在用户的 Web 站点上使用 CGI 来创建定制错误响应页面,以及如何为邮件网关创建 CGI。第 10 章将讨论 Session 管理,它是创建复杂应用程序的重要工具。

第 11、12 章讨论带有数据库的 CGI 的使用方法。第 11 章着重论述简单的平面文件(flat-file)数据库,而第 12 章则讨论关系数据库及 Structured Querly Language(SQL)。

关于 CGI 的安全,将在第 13 章论述,这部分内容对任何一个编写将在 Internet 中使用的基于 CGI 应用程序的人而言,是十分重要的。最后,在第 14 章中,介绍一些 CGI 的替代技术。

本书的约定

- 真正的代码用特定的半等宽字体来排版。
- “⇒”提示代码是前行代码的继续,它们实际上是一行里的内容。

注意: 注意提供必要的信息,使读者不仅学习了用 CGI 编程方法处理事务,而且对正在完成的工作以及原因有很好的理解。

技巧: 技巧提供关于如何使用 CGI 编程的洞察和捷径,从而发挥其最大的潜能。

警告: 警告框告诫读者,避免完成那些可能引起严重后果的操作。

新术语: 新术语图标出现在为新的、必要的术语提供清晰定义的文本旁边。

目 录

译者序

前言

第一部分 CGI 的基本知识

第 1 章 CGI 介绍	1
1.1 什么是共同网关接口	1
1.2 Web 上的 CGI 程序举例	1
1.3 CGI:更趋完善的后端技术	3
1.4 CGI 的优点	4
1.5 CGI 工作原理举例	4
1.6 用 UNIX 的方式思维	5
1.6.1 标准输入	6
1.6.2 标准输出	6
1.6.3 文件使用权	6
1.7 关于术语的一些说明	8
1.7.1 URL 和 URI	8
1.7.2 实体、文档和资源	8
1.8 Web 服务器	9
1.8.1 Web 服务器的本质	9
1.8.2 站点结构	9
1.8.3 CGI 平台概述	10
1.8.4 UNIX Web 服务器	11
1.8.5 Windows NT Web 服务器	11
1.8.6 Macintosh Web 服务器	11
1.9 小结	12
1.10 专题讨论	12
1.10.1 问与答	12
1.10.2 小测验	12
1.10.3 练习	12
第 2 章 创建 CGI 程序	13
2.1 选择编程语言	13
2.1.1 编译语言与解释性语言	13
2.1.2 CGI 库	14
2.2 Perl 语言	14
2.2.1 用 Perl 创建 CGI 程序	16
2.2.2 不应使用 cgi-lib.pl 的理由	17

2.3 Visual Basic	17
2.4 C/C++	17
2.5 供 CGI 使用的 C 库	18
2.6 Bourne 外壳程序	18
2.7 AppleScript	19
2.8 调试 CGI 脚本	20
2.8.1 在安全环境下测试程序	20
2.8.2 首先从命令行测试程序	20
2.8.3 通过 Web 服务器测试脚本	21
2.8.4 现在是较困难的部分	22
2.9 小结	22
2.10 专题讨论	22
2.10.1 问与答	22
2.10.2 小测验	23
2.10.3 练习	23

第二部分 CGI 基础

第 3 章 HTML 表单概述	25
3.1 最基本的表单:ISINDEX	25
3.2 创建表单	27
3.2.1 INPUT 标记	28
3.2.2 文本录入域	28
3.2.3 Submit 按钮	28
3.2.4 Reset 按钮	29
3.2.5 复选框	30
3.2.6 单选按钮	31
3.2.7 口令	31
3.2.8 图像按钮	32
3.2.9 文件选择域	32
3.2.10 隐含域	33
3.2.11 TEXTAREA 标记	33
3.2.12 SELECT 标记	33
3.3 确证表单内容	35
3.3.1 JavaScript 程序	36
3.3.2 总汇	36
3.3.3 将程序再推进一步	37
3.4 通过电子邮件发送表单	39

3.5 小结	39
3.6 专题讨论	39
3.6.1 问与答	39
3.6.2 小测验	40
3.6.3 练习	40
第4章 使用表单中的数据	41
4.1 再访 FORM 标记	41
4.1.1 ENCTYPE 属性	41
4.1.2 multipart/form-data 编码类型	42
4.2 请求方法	43
4.2.1 GET 方法	43
4.2.2 POST 方法	44
4.2.3 HEAD 方法	44
4.3 脚本举例	44
4.3.1 简要的题外话	45
4.3.2 回到正题	46
4.4 GET 与 POST	49
4.5 额外路径信息	49
4.6 其他头部信息	50
4.7 使用头部信息	51
4.8 小结	52
4.9 专题讨论	52
4.9.1 问与答	53
4.9.2 小测验	53
4.9.3 练习	53

第三部分 CGI 程序构建块

第5章 HTTP 协议	55
5.1 HTTP 基础知识	55
5.2 HTTP 通话的例子	56
5.2.1 第一步:建立 TCP 连接	56
5.2.2 第二步:Web 浏览器向 Web 服务器发送命令	57
5.2.3 第三步:Web 浏览器发送请求头 信息	57
5.2.4 第四步:Web 服务器应答	57
5.2.5 第五步:Web 服务器发送应答 头信息	58
5.2.6 第六步:Web 服务器向浏览器 发送数据	58
5.2.7 第七步:Web 服务器关闭 TCP 连接	58
5.3 服务器应答码	58

5.4 应答头信息	60
5.4.1 Cache-control	60
5.4.2 Content-length	60
5.4.3 Content-type	60
5.4.4 Expires	60
5.4.5 Pragma	60
5.4.6 Server	60
5.4.7 Set-Cookie	60
5.5 内容类型	61
5.5.1 服务器使用内容类型的方法	61
5.5.2 内容类型的类别	62
5.5.3 非标准类型	63
5.6 安全连接	63
5.7 小结	64
5.8 专题讨论	64
5.8.1 问与答	64
5.8.2 小测验	64
5.8.3 练习	65

第6章 Server Side Includes(SSI)

6.1 Server Side Includes 的工作原理	66
6.2 为 SSI 配置 Web 服务器	66
6.2.1 Apache 和 NCSA	66
6.2.2 CERN 服务器	67
6.2.3 Netscape Web 服务器	67
6.2.4 Microsoft Internet 信息服务器	67
6.2.5 O'Reilly 的 WebSite	67
6.2.6 WebStar	67
6.3 使用 SSI	68
6.4 使用 SSI 设计页面	73
6.4.1 使用 # include 指令	73
6.4.2 最后修改的日期	75
6.4.3 使用 # exec 指令	75
6.5 XSSI	76
6.5.1 printenv	76
6.5.2 set	76
6.5.3 if 语句	77
6.6 小结	78
6.7 专题讨论	78
6.7.1 问与答	78
6.7.2 小测验	78
6.7.3 练习	78

第四部分 创建 CGI 应用程序

第7章 基本的 CGI 应用程序	79
------------------------	----

7.1 在线调查程序	79
7.1.1 第一步:创建表单	79
7.1.2 第二步:编写 CGI 程序	81
7.1.3 创建链接的下拉式菜单	89
7.1.4 第一步:创建表单	89
7.1.5 第二步:增加 JavaScript 功能	89
7.1.6 第三步:编写 CGI 脚本	91
7.2 使用 CGI 根据模板来创建页面	92
7.2.1 第一步:弄明白必须使用的内容	92
7.2.2 第二步:发送头部信息	93
7.2.3 第三步:读取稿件并将其转换为 HTML	93
7.2.4 第四步:显示页面脚注	94
7.2.5 运行脚本	96
7.3 小结	96
7.4 专题讨论	96
7.4.1 问与答	97
7.4.2 小测验	97
7.4.3 练习	97
第 8 章 创建动态图像	98
8.1 gd 库	98
8.1.1 gd 库的来源	98
8.1.2 在程序中使用 gd	98
8.2 gd 库的实际应用	101
8.2.1 程序的运作原理	101
8.2.2 预处理器指令	101
8.2.3 打开 main() 函数和说明变量	102
8.2.4 创建图像并绘制标准部分	102
8.2.5 从命令行获取数据	103
8.2.6 绘制和连接图形上的点	103
8.2.7 标注图形上的点	104
8.2.8 保存图像、清理和退出	104
8.2.9 完整的源代码	105
8.3 供 Perl 使用的 gd	107
8.3.1 重访股票图表程序	107
8.3.2 完整的源代码	110
8.4 使用动态图像	111
8.5 小结	111
8.6 专题讨论	112
8.6.1 问与答	112
8.6.2 小测验	112
8.6.3 练习	112

第五部分 利用 CGI 进行 更好的 Web 发布操作

第 9 章 创建定制的错误响应页	113
9.1 什么叫错误信息页	113
9.2 改变 Web 服务器的设置	113
9.2.1 Apache	115
9.2.2 Netscape Web 服务器	115
9.3 基本的错误响应页	116
9.4 稍高级的错误页	117
9.4.1 返回到引用页的链接	117
9.4.2 创建其他定制链接	117
9.5 401 错误	120
9.6 邮件网关的 CGI	121
9.6.1 创建电子邮件表单	122
9.6.2 检查表单和发送邮件	125
9.7 小结	133
9.8 专题讨论	133
9.8.1 问与答	133
9.8.2 小测验	134
9.8.3 练习	134
第 10 章 保存用户信息和通话管理	135
10.1 什么是通话管理	135
10.2 基本的身份鉴定	136
10.3 表单中的隐含字段	136
10.4 Magic 存储块	141
10.4.1 存储块的工作原理	142
10.4.2 使用存储块保存用户信息	143
10.4.3 用 JavaScript 设置和检索 存储块	145
10.4.4 与存储块的通话管理	146
10.4.5 为什么不能混用 CGI 和 存储块	150
10.5 小结	150
10.6 专题讨论	150
10.6.1 问与答	150
10.6.2 小测验	151
10.6.3 练习	151

第六部分 数据库集成

第 11 章 使用 CGI 与数据库	153
11.1 什么是数据库	153

11.1.1 平面文件数据库	153	13.3.2 缓冲区溢出问题	187
11.1.2 用字符分隔数据	154	13.3.3 不要给外壳命令发送原始 输入	188
11.1.3 按列分隔数据	154	13.3.4 使用文件路径是危险的	189
11.2 数据库与 Web	155	13.3.5 不要将 Perl 解释器置于 cgi-bin 中	189
11.3 创建数据库应用程序	155	13.3.6 DOS 批文件的安全漏洞	190
11.3.1 创建 HTML 表单	155	13.3.7 保持服务器信息隐密	190
11.3.2 编写确认脚本	157	13.4 安全编程	190
11.3.3 编写交付脚本	164	13.5 小结	191
11.3.4 读取数据库的脚本	166	13.6 专题讨论	191
11.4 小结	169	13.6.1 问与答	191
11.5 专题讨论	169	13.6.2 小测验	192
11.5.1 问与答	169	13.6.3 练习	192
11.5.2 小测验	169	第 14 章 CGI 的替换产品	193
11.5.3 练习	170	14.1 回顾	193
第 12 章 结构化查询语言	171	14.1.1 为什么会出现 CGI 的替代品	193
12.1 什么是关系数据库	171	14.1.2 消除 CGI 的局限性	194
12.2 结构化查询语言	172	14.2 CGI 替代产品的类型	194
12.2.1 SELECT 语句	172	14.3 Active Server Pages	195
12.2.2 连接(Joins)	175	14.3.1 组件	195
12.2.3 增加、删除和更新记录	176	14.3.2 脚本	195
12.3 开放式数据库连接	177	14.3.3 HTML 页	196
12.4 DBD/DBI	178	14.3.4 其他 ASP 平台	196
12.4.1 使用 DBI 的一个样本程序	178	14.4 Netscape LiveWire	196
12.4.2 更深入的讨论	180	14.5 Allaire Cold Fusion	197
12.4.3 建议	180	14.6 Apache mod_perl 模块	198
12.5 小结	181	14.7 PHP/FI	200
12.6 专题讨论	181	14.8 ePerl	202
12.6.1 问与答	181	14.9 Server Side Java	203
12.6.2 小测验	181	14.10 Netscape Server API	204
12.6.3 练习	181	14.11 Microsoft 的 Internet Server API	204
第 7 部分 安全性与其他内容		14.12 小结	205
第 13 章 CGI 安全性	183	14.13 专题讨论	205
13.1 安全的必要性	183	14.13.1 问与答	205
13.1.1 入侵 Mac 竞赛	183	14.13.2 小测验	205
13.1.2 冒险评估	184	14.13.3 练习	205
13.2 保护 Web 服务器	185	第 8 部分 附录	
13.2.1 保持软件最新	185	附录 A MIME 类型	207
13.2.2 统一存放 CGI 程序	186	附录 B 应答码和相应说明	210
13.2.3 文件许可权	186	附录 C 环境变量和请求头	214
13.2.4 不利于安全的服务器选项	186		
13.3 常见 CGI 安全漏洞	187		
13.3.1 关于 CGI 工作原理的说明	187		

附录 D CGL/Perl 快速参考	219	附录 F ASCII 字符表	261
附录 E 常规表达式总结	259	附录 G 小测验答案	271

第一部分 CGI 的基本知识

第 1 章 CGI 介绍

本章是使用 CGI 创建功能强大、交互作用的 Web 应用程序的第一步。读者将逐步了解使用 CGI 往 Web 站点上添加诸如购物手推车、动态生成的图片以及数据库连接这样一些特性。然而在创建这些应用程序之前,读者应该理解一些重要的概念,这些概念将为学习后继内容提供坚实的技术基础。

本章包括以下内容:

- CGI 的作用
- Web 上 CGI 程序举例
- CGI 的优点
- UNIX 操作系统的基本概念
- Web 服务器平台的概述

1.1 什么是共同网关接口

共同网关接口(Common Gateway Interface)是人们非常熟悉的 CGI,它将 Web 服务器连接到外部应用程序。CGI 可以做两件事情:第一,它可以收集从 Web 浏览器发送给 Web 服务器的信息,并使这些信息供外部程序利用;第二,对提出请求的 Web 浏览器发送程序的输出。大多数 CGI 程序都可以同时完成这两件事情。

注意:本书中 CGI 程序和 CGI 脚本在功能上是一致的。CGI 程序经常被称为 CGI 脚本,是因为程序经常用一些脚本语言(比如 Perl)来编写。

用 CGI 可以完成的一些事情包括:收集来自数据库的信息并在 Web 上发送记录;提供调查或问题让用户回答;为用户动态创建图像,如图表和动画。

1.2 Web 上的 CGI 程序举例

Web 上的一些给人印象最深的站点就是使用 CGI 创建的。例如,Onsale (<http://www.onsale.com>)就创建了一个基于 Web 的拍卖行,该行出售计算机配件和家用电器。整个拍卖过程都是通过使用 CGI 程序来完成的。

要购买产品,Web 站点的访问者必须以投标者的身份进行登记,然后进入站点并开始对产品进行投标。这个站点使用 Windows NT Web 服务器和 Oracle 数据库。图 1-1 显示了 Onsale 的部分投标屏幕。

CGI 另一个常见的用途是通过用户的 Web 页提供实时信息检索。这种服务最明显的例子之一就是投资和财经消息站点的优势,这些站点提供股票报价的最新检索。例如,以 PC Quote

图 1-1 Onsale Web 站点上的投标者表单

(<http://www.pcquote.com>)为例,该站点不仅通过它自己的 Web 页向它的用户提供最新的财经信息,而且也向很多认购者,比如 Motley Fool (<http://www.fool.com>)提供信息。

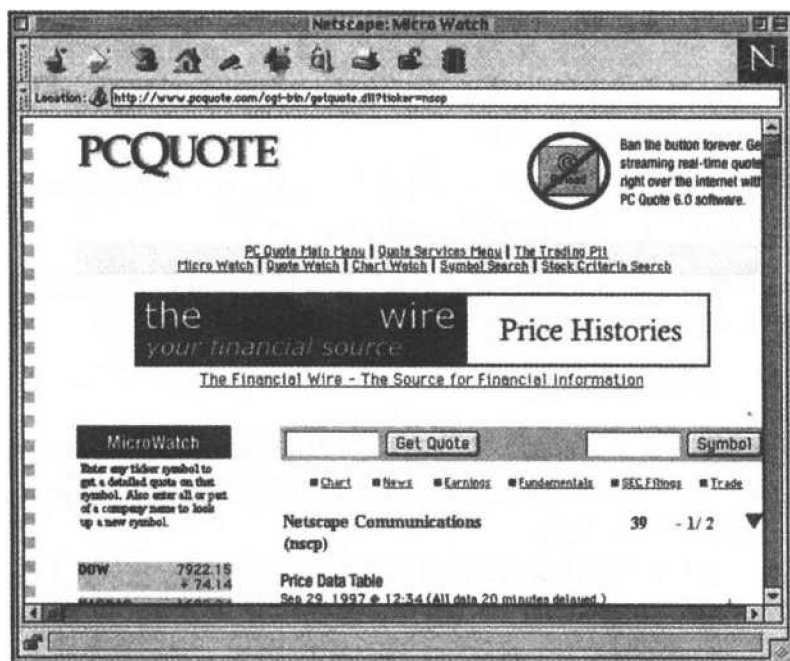


图 1-2 利用 Netscape 有关股票价格的最新信息动态产生的页

CGI 应用程序接受股票符号并且利用有关股票价格和当天市场活动的最新信息动态产生一个页。图 1-2 就是 Netscape 股票的 PC Quote 页。

CGI 编程还有另一个常见的使用,这就是构造联机布告板,用户们可以登录和参与讨论。很多软件包都已经为 Windows NT 和 UNIX 服务器上的站点提供了顶刮刮的会议规则。Motet 是一个基于 Web 的会议软件包,该软件包可以供 UNIX 服务器使用。Motet 是一个 CGI 程序套件,用户可以使用这个套件来创建带有多多个会议及每个会议内的多个议题的联机社团。图 1-3 是从 Motet 捕获到的一个议题页屏幕。Motet 主页为 <http://www.motet.com>。

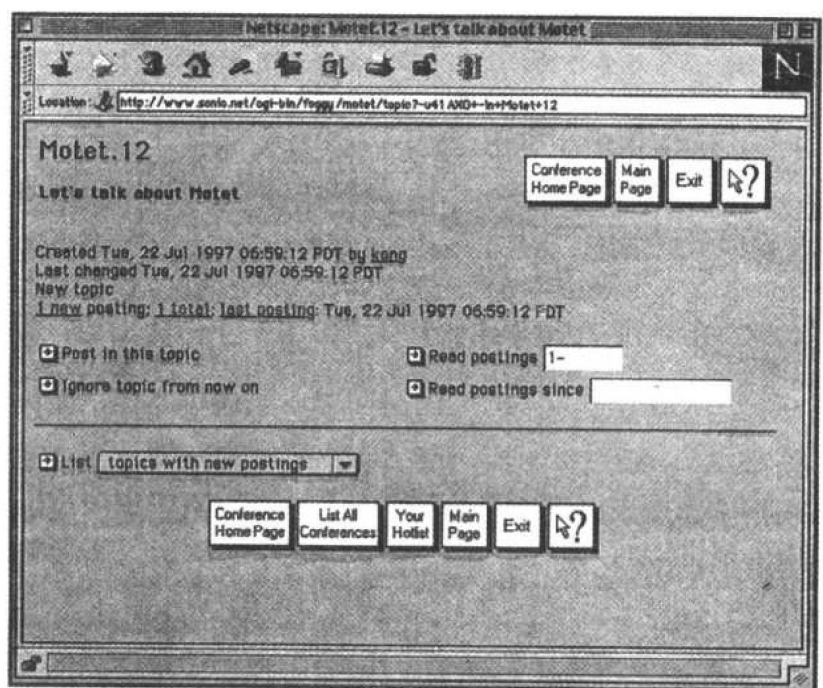


图 1-3 基于 Web 的 Motet 会议软件包的一个议题页面

1.3 CGI:更趋完善的后端技术

用户很可能已使用 HTML 创建过 Web 页面和 Web 站点。HTML 拥有许多强大的功能,使这种协议受到了广泛的采纳,包括在页上同时提供图像和文本的能力以及将页链接到一起能力。然而,HTML 没有提供创建交互式 Web 应用程序的任何手段。如果只用 HTML, Web 页面就不能向服务器上的应用程序发送数据,也不能编写可以产生动态页面的程序。

幸运地是,像 CGI 这样的后端(back-end)技术可以填补这种空白。CGI 是使用户能够通过创建将多个页连接到一起的应用程序来延伸 HTML 的引擎,同时,该引擎也使用户能与服务器打交道以及使用 HTML 表单来处理检索到的数据。

每当用户到一个包括有联机购物、简单的顾客簿或者联机消息库的 Web 站点上时,作者就使用 CGI(或某些其他后端技术)将其全部连接在一起。本书将向读者提供创建这种交互式应用程序所需的信息。学过 CGI 之后,你将不会仅仅满足于静态的 HTML 页。

1.4 CGI 的优点

下面简要地列出 CGI 所提供的几个优点:

- 平台独立性: CGI 被内置于大多数 Web 服务器软件包中, 这与它们运行在哪个平台上无关。大多数流行的 Web 服务器都支持 CGI, 包括 UNIX 环境下的 Apache、Netscape 和 NCSA; Windows NT 环境下的 Netscape、Microsoft IIS 和 O'Reilly WebSite; 以及 Macintosh 环境下的 WebStar。大量的其他 Web 服务器也都支持 CGI 应用程序。

警告: 正是因为多种平台支持 CGI 并不意味着为一个平台编写的 CGI 程序可以在另一个平台上运行。尽管服务器侧的 Java 小程序和一些 Perl 脚本稍加或不加修改就可以在不同的平台上运行, 但 C 程序必须被重新编译之后才可以在别的平台上运行。

第 2 章将讨论可以用来编写 CGI 程序的几种语言。

- 语言独立性: CGI 程序独立于编写它们所使用的语言。这个特征意味着用户可以使用最为熟练的语言来编写 CGI 程序。往往用 Perl、TCL、C、C++、Visual Basic 以及 AppleScript 来编写 CGI 程序。

- 层次感: CGI 接口的简明性使它有着非常好的层次感。用户可以使用 CGI 来完成像给适当的电子邮件地址发送调查内容这样简单的事情, 或者像创建一个连接到许多 Oracle 数据库的联机购物中心这样复杂的事情。

除前面这些优点之外, CGI 还提供了一个坚实的基础, 用户可以将它应用到其他后端技术中(参阅第 14 章以了解有关这个议题的更多内容)。

1.5 CGI 工作原理举例

在开始编写 CGI 脚本之前, 必须理解 CGI 事务中所发生的步骤。图 1-4 中的图解展示了一个使用 CGI 的事务中所发生的基本步骤, 同时, 接下来的在一个基于 Web 的词典中寻找单词的例子阐明了图解中的每一个步骤。对图 1-4 中的内容进行编号, 这样, 读者就可以遵循例子中的每一步, 并了解其完成 CGI 请求的过程。

- 1) 用户到达含有一个表单的页, 该表单使他能够在联机词典中查找一个单词。

- 2) 用户键入单词“hacker”并单击 Submit 按钮。把从表单中收集到的数据送回到 Web 服务器, 以供用户按 Submit 按钮时予以处理。

- 3) 服务器从浏览器那里接收到请求对单词“hacker”予以定义的表单并将其确认为一个 CGI 请求。通过 CGI 将该请求传送到 Dictionary 应用程序。

- 4) Dictionary CGI 应用程序查找单词“hacker”, 然后用 HTML 格式化定义(或者说, 如果单词“hacker”不在词典中, 就格式化适当的出错代码), 并把动态生成的 Web 页面发送给 Web 服务器。

- 5) Web 服务器把页发送给发出请求的计算机, 在那里浏览器窗口将显示该页面。

以上是 CGI 工作原理的基本描述。另外需要注意的一点是, 不必总是使用一个表单来开始 CGI 请求。比如, 用户的页面上可能有一个链接, 用来请求得到阿拉斯加 Fairbanks 地区的天气情况。那个链接就可能指向一个 CGI 程序, 该程序从天气服务器自动得到最新的天气信息并在 Web 上发布这些信息。另一点是 CGI 程序不必在 HTML 表单中返回信息。CGI 程序可

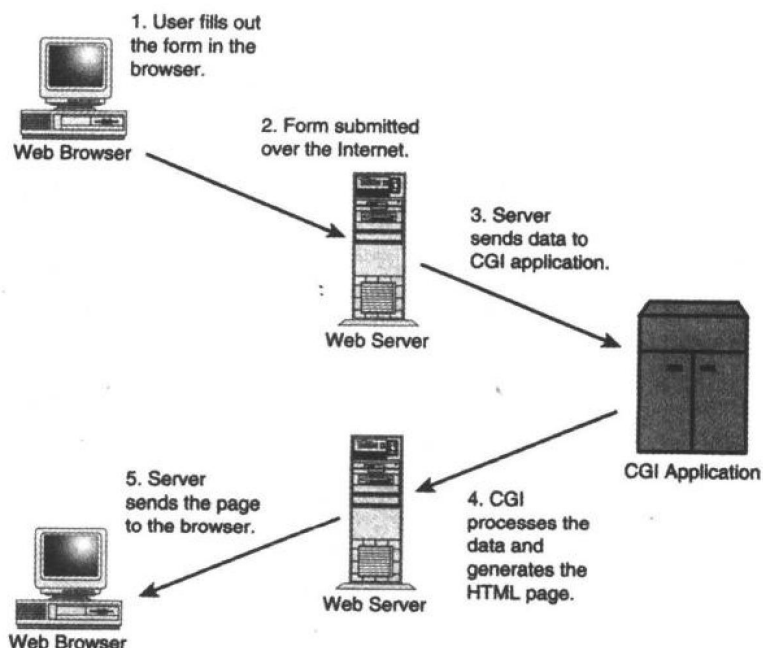


图 1-4 处理 CGI 请求的方法

以很容易地使用 GIF 格式返回一般的 ASCII 文本或图像。预定的用途决定所使用的格式。

1.6 用 UNIX 的方式思维

既然读者对 CGI 的工作原理已有了基本了解,也知道它用在哪些地方,现在就可以开始创建 CGI 应用程序了。UNIX 的熟练用户可以跳过这部分内容,而从未接触 UNIX 的用户就需要花一些时间,以了解 UNIX 平台的基本原理以及它们和 CGI 的关系。CGI 的一个显著特点就是它被内置于每个流行平台(如 UNIX、Windows NT 和 Mac OS))上的大多数 Web 服务器中,所以学过 CGI 之后,就可以随心所欲地创建 Web 应用程序,而不必顾及使用的是哪一个 Web 服务器,至少在未来几年中情况如此。为简明起见,本书中的大多数例子将在 UNIX 平台上编写,不过,本书也将说明将信息应用于其他平台的方法。

作者使用 UNIX,是因为对 Web 服务器而言,它是最流行的平台。另外,CGI 最初也是由基于 UNIX 的 Web 服务器开发而来的,所以在 UNIX 环境的参数中研究 CGI 的工作原理是最为方便的。问题是很多计算机的终端用户对 UNIX 并不十分了解,因为通常情况下它是作为服务器的平台被加以使用,而用户每天都在使用的计算机上却运行 Microsoft Windows 或 Mac OS。

讨论有关 UNIX 的内容时,不由得让人想起《Firefox》(火狐)这部电影。在影片里,影星 Clint Eastwood 扮演一名战斗机飞行员,他被派往俄罗斯去窃取世界上最为先进的战机。这种命名为“Firefox”的飞机,可以发射 Mach 6 型导弹并且拥有最为先进的武器系统,而最为奇妙的是这种飞机由思维控制。Eastwood 的麻烦来了:为了操纵飞机,他必须用俄语思维。

类似地,UNIX 是功能最为强大的计算机环境之一,使用起来极为灵活,并且表现出色。但问题是要想把它的种种特性发挥出来,用户必须用 UNIX 的方式来思维。

大家都知道,UNIX 拥有成百条神秘的命令和数以万计的更为神秘的命令行选项,这些内

容不会在这里讨论。事实上,本章只包括那些能直接应用于 CGI 的 UNIX 概念,它们是标准输入、标准输出以及 UNIX 文件使用权的用法。

1.6.1 标准输入

标准输入,在 UNIX 中被亲切地称作 STDIN,是程序的数据来源。STDIN 通常称为控制台。通俗地讲,标准输入就是从键盘上输入的内容。然而,如果愿意的话,也可以告诉某个程序使用某些别的标准输入。例如,向 UNIX 系统上的另一个用户发送电子邮件,可以只键入 mail 和正确的用户名。如果要向我发送电子邮件消息,可在命令行中键入下列内容:

```
mail rafe
```

然后读者就可以从键盘输入消息,待输入完成,按下 Ctrl + D 键就可将消息发送。假设读者不想靠手来输入上述消息,而是想通过已存放于他的目录下的一个文件来得到一份 Web 站点的列表,这个时候,可以简单地把前述文件作为标准输入,如下所示:

```
mail rafe < coolwebsites.txt
```

命令中小于号的意思是“取出文件的内容,并把它们作为 mail 命令的输入”。这种技术称作重定向(redirection)。

1.6.2 标准输出

标准输出或者 STDOUT 是相对标准输入而言的。在程序执行时,打印出的数据就是 STDOUT。和标准输入相近,标准输出的缺省地址也在控制台(通常是屏幕)。例如,在 UNIX 系统上列出一个目录下的全部内容,要使用 ls 命令。输入 ls 命令后,屏幕上将显示出所有文件的目录清单。这一文件清单就是 ls 命令的标准输出。

如果想将那个清单保存在一个文件中,以备将来之用,就可以像先前更改 mail 命令的标准输入地址那样更改标准输出的地址。把 ls 命令的输出送给文件 currentdirectory.txt 时,可以使用命令:

```
ls > currentdirectory.txt
```

其中,大于号的意思是“把程序的输出送到这个地址”。这条命令将生成新文件“currentdirectory.txt”。如果另一个文件的文件名也是“currentdirectory.txt”,它将被覆盖。

技巧:若想文件不被覆盖,而是把命令的输出附在其后,可以使用两个大于号而不是一个,即“>>”。使用“>>”要当心,如果上述文件不存在,就会有错误信息出现。

1.6.3 文件使用权

编写 CGI 脚本时还需要了解 UNIX 操作系统的另一个方面是文件使用权。文件使用权明确了谁可以读/写或执行某一特定的文件。

注意:文件使用权这个系统是针对 UFS 和 NFS 类型的文件系统而言的。然而,这两种类型的文件系统却是目前为止在使用 UNIX 的计算机中最为常见的。

下面这个详细的目录清单来自一个 UNIX 服务器:

```

-rwxr-xr-x  1 rafec    users    8566 Sep   8   16:00 wt.html
-rw-r--r--  1 rafec    users    8566 Aug  27   10:22 wtnew.html

```

每个文件的访问权都有三个层次：单个用户、用户组及其他人。单个用户是文件的真正拥有者。目录清单中的第三栏包含着文件拥有者的姓名，即 `rafec`。所有用户也是一个组的成员，同时，所有文件被分配给某一特定的组。这两种情况下文件都属于用户组“`users`”。第三类的使用权属于其他人，所谓其他人，就是既不是文件的拥有者也不是用户组成员的那些人。在这种情况下，`rafec` 以及“`users`”组成员以外的任何人都被归入其他人的行列。

真正的文件使用权由目录清单第一栏中经过编码的 10 字符的字符串组成。比如文件 `wt.html` 的使用权看起来不很明了，但如果清楚了各部分的含义，那么它们又是很容易读明白的。图 1-5 对使用权列表的组成进行了解析。

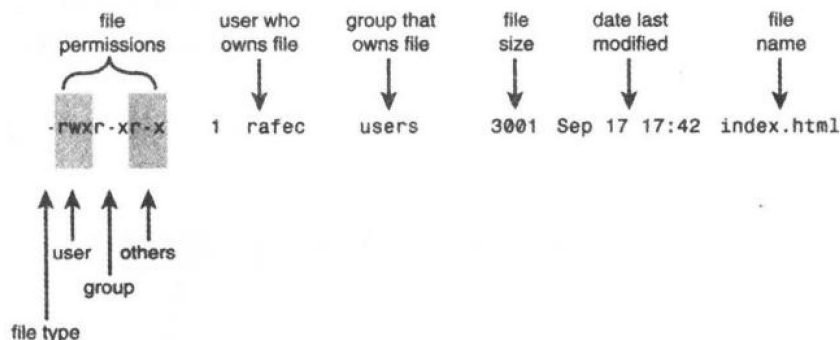


图 1-5 读取 UNIX 文件使用权的方法

从图 1-5 中可以看出，使用权字符串的第一个字符定义了文件类型，通常这个字符是“-”。目录中这个位置上是一个“d”。符号链接中含有“l”作为使用权字符串中首字符，它是简单地作为文件系统中其他某个位置的另一个文件提供指针的文件。

使用权字符串的后九个字符为想获得文件的用户们定义了实际上的使用权。使用权以三个字符为单位，被分成三组字符串。第一组针对那些想拥有文件的用户，第二组针对想拥有文件的用户组，第三组针对其他人。在每一组字符串中，第一个字符代表读，第二个代表写，第三个代表执行。如果字符是“-”，那么特定的用户就不能拥有相应的权利。如果字符是一个字母，那么用户就可以相应地使用文件。顺便指出，定义用户不同使用权的字母分别是“r”代表读，“w”代表写；“x”代表执行。

下面就有一个例子：

```

-rwxr-xr-x  1 rafe     users    1637 Dec 11 1996 everest.html

```

这是个普通文件，文件拥有者(rafec)可以读、写和执行这个文件。用户组的成员可以读和执行这个文件，而其他人则只能读该文件。

这些文件使用权是很重要的，因为要想让文件出现在 Internet 上，他们必须能进入 Web 服务器。以前 Web 服务器经常作为根用户，享有系统中每个文件的读、写和执行权。现在的 Web 服务器则往往作为 `nobody` 用户。`nobody` 意味着几乎没有任何权利，这显然是为了安全起见。所以，你的目录和 HTML 文档必需可供其他人读取，这时 Web 服务器进程才能读取它们。同样的原因，CGI 程序也应该可以被其他人执行，这样，Web 站点的访问者才可以访问它们。

文件使用权不正确是常见的问题,这个问题是人们试着到达他们的 Web 站点并予以运行时经常被忽视的问题。

1.7 关于术语的一些说明

本书使用了大量篇幅介绍 Web 服务器和客户(clients)。所谓客户就是连接到 Web 服务器上的程序。客户程序通常是 Web 浏览器,比如 Netscape Navigator 或 Microsoft Internet Explorer。还有一些其他 Web 客户程序,但在本书当中,客户这个术语总是指 Web 浏览器。

1.7.1 URL 和 URI

为了使用户能够方便地在 Web 上确定文档的位置,World Wide Web Consortium 为指定文档所在位置设计了一个通用寻址规则。这些地址被称为统一资源定位器(Uniform Resource Locators,缩写为 URL)或统一资源标识器(Uniform Resource Identifiers,缩写为 URI)。这两个术语是可以互换使用的,但 URL 最为常用。

URL 最多可以包含 4 个不同的部分。图 1-6 给出了 URL 的一个例子(对组成这个 URL 的每个部分都作了标记)。

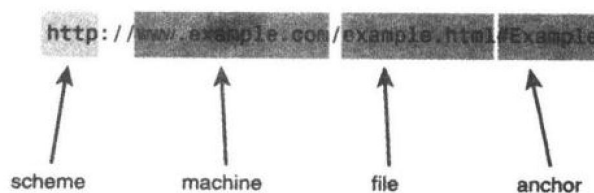


图 1-6 URL 解剖

建立 Web 站点时,理解 URL 的构成方法和 URL 各组成部分的含义是很重要的。在开始编写 CGI 程序之前,也需要弄明白 URL 寻址方法。以下是一个完整 URL 的四个不同部分的介绍。

1.7.2 实体、文档和资源

当 Web 浏览器向 Web 服务器发出一个请求时,服务器通常用某种类型的文档来答复。这个文档可以是 HTML 文档、图像、RealAudio 流或 QuickTime 电影。本书把服务器(有时也包括浏览器)发送的内容称为资源、文档或实体。所有这三个术语都指同一事物,即 Web 服务器发送给浏览器的数据块(data chunk)。

1. 规则(Scheme)

URL 的第一部分叫作规则。规则指的是用来连接 URL 中指定资源的协议。到现在,最为常见的规则是 `http`。然而,也有其他一些规则。比如,`ftp` 用来下载文件;`mailto` 用来发送邮件;`news` 用来打开 Usenet 新闻组。

2. 机器(Machine)

规则之后是资源所在机器的名称或 IP 编号。通常使用机器的名称,比如 `home.netscape.com`、`www.sun.com` 和 `cnn.com`。使用机器名称时,对应于该名称的 IP(Internet Protocol)编号是使用域名服务器(DNS)协议自动形成的。于是,Web 浏览器就根据 IP 编号与服务器相连。与 Internet 相连的每一台计算机都有特定的 IP 编号。