



“九五”国家重点电子出版物规划项目
计算机动画教室系列

编程宝典

丛书

1

自由编程世界奇葩

Perl 5 编程核心技术

北京希望电脑公司 总策划
杜经农等 编 著



本书光盘包含

1. 本书所有例程的代码
2. 与 Perl 相关的部分应用软件



北京希望电子出版社
Beijing Hope Electronic Press
www.bhep.com.cn



“九五”国家重点电子出版物规划项目
计算机动画教室系列

00107521

编程宝典 丛书

1

TP312PE
04

自由编程世界奇葩

Perl 5

编程核心技术

北京希望电脑公司 总策划
杜经农等 编 著



本书光盘包含

1. 本书所有例程的代码
2. 与 Perl 相关的部分应用软件



北航 C0525167



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

Perl 是一门功能强大、易于使用、容易移植的程序设计语言。Perl 用途广泛,可用于 Unix 或 Windows NT 系统管理, Web 开发, Web 服务器管理, 网络程序设计, 跨平台的 GUI 程序设计。尤其在交互式站点的 CGI 程序设计中, Perl 5 是最为方便、有效的设计语言, CGI 编程已成为 Perl 的重要应用之一。

本书从指导程序员实际开发应用程序的角度, 全面系统地介绍了 Perl 5 语言的核心技术, 讨论了它的应用编程方法, 并给出了大量实例辅以说明。全书由 30 章和一个附录组成, 内容包括 Perl 简介, Perl 程序格式, 变量与常量, 标量, 清单、数组和散列, 引用, 操作符与表达式, 上下文, 判断结构, 循环结构, 打印、格式化输出与报表, 命令行输入, 文件与目录操作, 正则表达式, 模块化编程, 面向对象编程原理, 类, DBM 数据库, DBI 接口及 MySQL 数据库, 套接字与 Internet 网络编程, Web 服务器管理编程, Win32 特殊调用, CGI 编程常用技术, SSI 与页面计数器, 基于 Web 的电子邮件, 文学发表站点的实现方案——演示文件上传与论坛技术, 交互式实时页面技术, 调试 Perl 程序, Perl 安全编程等。本书作者集中了 Perl 5 编程经验和方法, 以及学习、实践 Perl 5 语言的规律和历程, 编写了本书, 并提供了个人网站供读者提问和探讨 Perl。

本书内容系统, 结构清晰, 指导性和实用性强。不仅是 Perl 程序设计人员和网络编程人员必备的技术参考书, 而且是高等院校相关专业程序设计课程教学、自学用书。

本书光盘包含了本书所有例程的代码及一些与 Perl 相关的部分应用软件。

系 列 书 : 编程宝典丛书 (1)

书 名 : Perl 5 编程核心技术

文 本 著 作 者 : 杜经农 等编著

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

责 任 编 辑 : 周风明

出版、发行者 : 北京希望电子出版社

地 址 : 北京中关村大街 26 号, 100080

网址: www.bhp.com.cn E-mail: lwm@hope.com.cn

电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

(图书发行和技术支持)

010-62613322-215 (门市) 010-62531267 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心

C D 生 产 者 : 北京中新联光盘有限责任公司

文 本 印 刷 者 : 北京双青印刷厂

开 本 / 规 格 : 787 毫米×1092 毫米 1/16 开本 26.25 印张 613.5 千字

版 次 / 印 次 : 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷

印 数 : 0001-5000 册

本 版 号 : ISBN 7-900049-94-0/TP·94

定 价 : 48.00 元 (1CD, 含配套书)

说明: 凡我社光盘配套图书若有自然破损、缺页、倒页、脱页, 本社负责调换。

前言

Perl 是一门年轻的语言，Perl 之父 Larry Wall 于 1986 年创建了它。随后，在短短的十来年间，它迅速地发展并成熟，成为一门具有功能强大、易于使用、容易移植等众多优点的语言。我不想在这里对 Perl 使用过多的溢美之词，但我相信，一旦你学习并掌握了它，你会像我一样，深深地爱上这门语言，并会乐于使用它进行开发工作。这也是本书被撰写出来的目的——希望有更多的人了解这门语言，并用它来充实你的开发百宝箱。

为什么要学习 Perl？如果你有如下需求之一，就有理由学习它：

1. 编写程序用于进行 Unix 或 Windows NT 系统管理。通常，一个小小的 Perl 程序可以替你完成大量繁琐的手工操作。因此，Perl 出世之初，就成了系统管理员们的最爱。

2. Web 开发。用于产生交互式站点的 CGI 程序中，95%是用 Perl 编写的。目前，Perl 产生了 mod_perl 和 Perlscript 等扩展，以生成更迅速的 CGI 应用。

3. Web 服务器管理。你想编写自动化的程序测试你的 Web 服务器吗？或是需要一个程序自动维护 Web 服务器，并在异常情况下发出警告？用 Perl 吧，比用 C 要节省 50%的时间，但可以和 C 做得一样好。

4. 网络程序设计。Perl 内置了套接字的能力，并且在 Unix 和 Win32 平台上都工作得很好。

5. 跨平台的 GUI 程序。可能吗？这也许是你的第一个反应。但 Perl 确实能做到这种不可思议的事情。你可以在 Windows 中用 Perl 开发一个图形界面程序，它无需改动，在 Unix 下同样可以运行，并具有相同的图形用户界面。

实际上，Perl 已经发展成一门相当通用的语言，从 OpenGL 三维图形编程到串行口驱动程序的开发，它都能做得很好。如果你仅仅只想开发 Windows 下的单机运行的图形用户界面程序，虽然 Perl 可以做到，但显然用 VB 会更好些。不过，在网络迅速发展，操作系统百花齐放的今天，你的编程野心不会仅限于此吧。

掌握一门语言需要付出艰辛劳动，但是也充满了乐趣。如果你抱着“玩”的态度来研究、学习 Perl，你会发现轻轻松松就掌握了它。在本书的每一章，都有大量的例程示范了各种概念和编程原理。学习 Perl，并不需要你掌握全部知识后再来写程序。一旦你掌握了某些概念，就应该有写一些程序来验证它的冲动。本书中的例程不仅仅需要阅读，更需要你实际运行、修改、增强它们。你可以随意使用它们来作试验，这样会增加乐趣、加深理解。

在遇到 Perl 之前，我使用 C++编程。在开始学习 Perl 时，我对它语法中的奇怪符号充满了畏惧，例如一个正则表达式：

```
$_ =~ /([;<*\V`&!\$#\(\)\[\]\{\}\:~"])/;
```

这看上去像是一只猫在键盘上走过后留下的痕迹，但它确实是一条 Perl 语句，并且该语句漂亮地完成了一个模式匹配。如果用 C 编程做同样的事，也许代码将写满一页纸。那么，这些代码是否很难编写呢？事实并非如此，当你学完正则表达式一章，你会发现它很好理解并且容易编写。Perl 中之所以使用了一些不常用的符号，大概与它起源于 Unix 世界有

关。但要记住，键盘上有的符号，我们就有使用它的理由。所以不要害怕这些符号，当你开始编写了一两个 Perl 程序并适应这些符号后，会发现它们和 abc 并没有什么不同。实际上，Perl 借鉴了大部分 C 的语法，但比 C 要好学得多。如果你有 C 语言的经验，可以很快地学会 Perl 语言。如果你不曾学习过 C 语言，也不用失望。本书详细讲解了各种程序设计语言的基本概念，并假设读者没有任何编程语言的经验。

Perl 是免费的，是自由编程世界的一朵奇葩。世界各地有很多 Perl 的爱好者和使用者，他们之中有天赋的人开发了很多用于 Perl 编程的模块，并无偿供大家使用。我从 Internet 上下载了一些模块，并将它们放在了 <http://dujl.uhome.net> 上，在书中谈到了其中一些模块的用法。此外，该网站上还有一些其它的与 Perl 编程相关的工具软件和例程，读者可以在 Internet 上免费下载。

如果在阅读本书时有任何技术问题，可以去 <http://www.perlchina.com> 的技术论坛上提问，我将乐于回答。这个站点由网友 wushan 和我共同维护。此外，我还有一个个人主页 <http://dujl.uhome.net>，上面同样有论坛，还有一些相关的范例程序，有空不妨来坐坐。

在本书的成书过程中，张彦、刘太强、罗世明、黄中胜、吴忠均、杨一帆参与了本书的校对和录排工作，在此表示深切谢意。

好了，现在泡上一杯热咖啡，翻开本书，坐到机器前，让我们开始奇妙的 Perl 之旅吧！

杜经农
2000 年 8 月于华中理工大学
dujl@yahoo.com
<http://dujl.uhome.net>

目 录

第 1 章 Perl 概述 1	8.5 散列上下文.....78
1.1 Perl 的起源与发展..... 1	8.6 Perl 中上下文的确定.....79
1.2 Perl 的原代码..... 2	第 9 章 判断结构 84
1.3 Perl 的特性..... 3	9.1 if 结构.....84
1.4 Perl 能做什么..... 4	9.2 if else 结构.....86
1.5 安装 Perl 5..... 5	9.3 if elsif 结构.....87
1.6 选择合适的编辑器..... 7	9.4 多重条件判断.....89
1.7 Perl 未来的发展: Perl 6 和 Visual Perl..... 7	9.5 unless 判断语句.....91
第 2 章 Perl 程序格式 8	第 10 章 循环结构 93
2.1 Perl 程序预览..... 8	10.1 for 循环语句.....93
2.2 Perl 程序的格式..... 10	10.2 foreach 循环语句.....97
2.3 内嵌的 POD 文档..... 11	10.3 while 与 until 循环.....100
第 3 章 变量与常量 16	10.4 until 循环和 unless 循环.....103
3.1 变量..... 16	10.5 块控制语句.....105
3.2 常量..... 25	10.6 其它流程控制结构.....110
第 4 章 标量 32	第 11 章 打印、格式化输出与报表 114
4.1 标量简介..... 32	11.1 print 语句与一般输出.....114
4.2 标量数据的运算..... 33	11.2 使用 printf 进行格式化输出.....118
第 5 章 清单、数组和散列 45	11.3 输出报表.....121
5.1 清单..... 45	第 12 章 命令行输入 132
5.2 数组..... 46	12.1 命令行特殊变量 ARGV.....132
5.3 散列..... 53	12.2 获取命令行输入.....133
第 6 章 引用 57	12.3 GetOpt 函数.....135
6.1 什么是引用..... 57	12.4 命令行输入的相关模块.....138
6.2 利用引用创建复杂数据结构..... 63	第 13 章 文件与目录操作 142
第 7 章 操作符与表达式 65	13.1 基本文件读写操作.....142
7.1 位操作符..... 65	13.2 文件系统处理.....152
7.2 逻辑操作符..... 69	13.3 目录句柄.....159
7.3 条件操作符?:..... 72	13.4 系统级目录操作.....161
7.4 逗号操作符..... 72	第 14 章 正则表达式 163
7.5 操作符的优先级与结合性..... 72	14.1 正则表达式简介.....163
7.6 表达式..... 74	14.2 与正则表达式有关的函数 和操作符.....164
第 8 章 上下文 76	14.3 正则表达式的组成结构.....168
8.1 什么是上下文..... 76	14.4 正则表达式实例.....175
8.2 标量上下文..... 76	第 15 章 模块化编程 178
8.3 清单上下文..... 77	15.1 子程序.....178
8.4 数组上下文..... 78	15.2 库.....192

15.3 包	195	21.3 站点新文件索引	284
第 16 章 面向对象编程原理	201	第 22 章 Win32 特殊调用	287
16.1 什么是面向对象编程	201	22.1 OLE 调用	287
16.2 对象与类	202	22.2 调用 Windows API 函数	293
16.3 面向对象的基本概念	203	第 24 章 CGI 编程常用技术	319
第 17 章 类	205	24.1 表单与 CGI 程序的交互	319
17.1 如何构建类和对象	205	24.2 使用 cookies 传递页面状态	328
17.2 类方法与构造函数	207	24.3 使用隐含表单传输页面数据	335
17.3 对象方法	210	第 25 章 SSI 与页面计数器	340
17.4 对象属性的存取	212	25.1 服务器端包含 SSI	340
17.5 类的析构函数	216	25.2 页面计数器	343
17.6 类的继承性	217	第 26 章 基于 Web 的电子邮件	348
17.7 重载	220	26.1 在 UNIX 系统上使用 sendmail 发送邮件	348
17.8 类库	225	26.2 在 windows 系统上使用 MAPI 发送邮件	349
17.9 重点回顾	226	26.3 smtp 协议与跨平台的邮件 发送方案	350
第 18 章 DBM 数据库	227	26.4 用 POP3 和 IMAP 协议接 收邮件	367
18.1 建立 DBM 数据库	227	26.5 电子邮件的文件格式与解析	370
18.2 增加和删除 DBM 数据库记录	230	第 27 章 文学发表站点的实现方案— 演示文件上载与论坛技术	376
18.3 搜索数据库记录	236	第 28 章 交互式实时页面技术	385
18.4 定义多字段的 DBM 数据库	237	28.1 使用客户 pull 技术编写 聊天室程序	385
第 19 章 DBI 接口及 mysql 数据库	244	28.2 使用服务器 push 技术 创建动画	389
19.1 DBI 和 DBD 模块	244	28.3 GD 模块与实时图形创建	390
19.2 mysql 数据库	245	第 29 章 调试 Perl 程序	394
19.3 DBI 编程	248	29.1 Perl 编程中常见的错误	394
19.4 DBI 的类方法和句柄对象方法	254	29.2 使用 Perl 内置调试器调试程序	398
19.5 使用 DBI 进行 mysql 数据库处理	259	第 30 章 Perl 安全编程	406
第 20 章 套接字与 Internet 网络编程	269	30.1 设置 setuid 和 setgid 位的程序	406
20.1 套接字简介	269	30.2 变量污染	407
20.2 基于 tcp 协议的客户/服务 器编程	273	30.3 CGI 编程的安全性	408
20.3 使用 HTTP 协议编写 Web 客户程序	277	附 录	411
20.4 TCP 端口扫描	277		
20.5 编写 Web 爬虫	278		
第 21 章 Web 服务器管理编程	282		
21.1 分析服务器日志文件	282		
21.2 服务器文件维护	284		

第 1 章 Perl 概述

Perl 的全名是 Practical Extraction Report Language(实用报表提取语言), 正如这个名字所暗示的那样, Perl 最初的作用是从文本文件中提取信息, 并产生报表。然而, Perl 语言问世后, 马上得到了 Unix 系统管理员们的青睐, 并且有相当多热心的编程高手加入了 Perl 的开发行列, 使得 Perl 迅速发展成熟, 成为一门功能强大、完整的跨平台的通用语言。目前, Perl 还在不断的发展之中, 在 Internet 上, 几乎每天都在出现新的为 Perl 所写的模块, 这些模块能扩展 Perl 的功能, 并大大方便了编程。此外, Perl 的新版本 Perl 6 也正在开发之中, 同时, 微软公司也正在为 Perl 写一个可视化的开发环境 Visual Perl, 并将其包含在其著名的软件开发包 Visual studio 中。

1.1 Perl 的起源与发展

Perl 语言是由 Larry Wall 于 1986 年创建的, 而且至今他还在为 Perl 的发展不懈地努力。为了学好 Perl 语言, 了解一下这门语言的历史和发展是很有必要的。

在开发这门语言时, Larry Wall 是一个程序员, 并正在致力于开发多层次的、安全的广域网系统。他管理着一套系统, 由三台 Vaxe 和三台 sun 机器组成。这套系统位于美国西海岸, 并且和位于东海岸的一套配置非常相似的系统通过加密线路连接起来。Larry 的任务是提供对整个系统的支持。

然而, 管理这些系统时, 需要能够从一些有相互联系的出错报告中提取信息, 并产生报表。以跟踪软件开发项目中的错误并解决问题。当时并没有很好的工具用于这项工作, Unix 上有一些开发工具, 如 sed、awk 和 sh 等, 但它们需要配合实用程序才能完成整个工作, 而且还相当不便。于是, Larry 决定自己开发一种语言, 因此 Perl 诞生了。最初, 这门语言以他妻子的名字命名, 称为“Gloria”, 后来慢慢演变成了“Perl”。

最早的 Perl 语言是用来处理文本的, 它仅仅包含当今 Perl 5 语言的一小部分, 如模式匹配技术、文件操作、标量和格式化输出等内容。不过 Perl 已经显示出了它的优点, 开发出的程序运行速度比 sed 和 awk 程序快, 而开发难度又比 C 语言小, 开发周期也更短。它成为一个很受系统管理员们欢迎的语言。

在此后的几年中, Larry Wall 购买了其它人的正则表达式的技术, 在 Perl 语言中使用了这种技术, 并且在 Perl 中还增添了一些其它有用的特性。然后, 他将这些语言及所有原代码放到了 Internet 上。随着 Internet 的快速发展, Perl 也同时发展壮大起来, 系统管理员们发现由于 Perl 具有强大的文本操作能力, 非常适合 Web 编程。于是, Perl 发展成为主流的 CGI 编程语言, 并且成为 CGI 编程事实上的标准。同时, 由于 Perl 语言公布原代码, 越来越多对其感兴趣的编程高手们积极为其开发其它功能和模块, 如今 Perl 的能力已不仅局限于文本操作了, 而是成为了一种通用的语言。几乎所有用 C 语言能够做的事情, 使用 Perl 同样能够完成。包括 modem 驱动程序或 OpenGL 三维图形编程。

现在 Perl 具有另外一个优秀的特性就是其跨平台特性。Perl 解释器本身能运行在几十

种不同的系统上。由于它是解释性的语言，因此同一个原代码可以在不同类型的机器和操作系统上运行，而不会由于二进制可执行程序格式的不同而不能运行。世界各地为 Perl 开发模块的人来自不同的研究领域，使用不同的操作系统，它们在各自有研究的领域开发出了专业的模块。这大大扩展了 Perl 在不同系统上的能力，甚至有时使人感到惊讶，例如，你可以在 Windows 平台上开发出可以运行在 Unix 的 X Window 下的 GUI 程序，或者写一个程序，可以处理苹果机上的特殊文件。

Perl 这门新语言，目前已经走向成熟，并正得到世界范围内的广泛应用。将来，它必然有广阔的发展前景。

1.2 Perl 的原代码

到 Perl 5 为止，Perl 的原代码都是用 C 语言写成的，如果你对它的原代码感兴趣，可以去 CPAN(Perl 综合文档)站点下载一份。

Perl 允许你用 C 语言写它的扩展模块，或在 C 语言程序中使用 Perl 库。当然，这并不是一项简单的工作，需要你对 Perl 语言的原代码及其数据结构有一定的认识。在 Perl 的标准发布版本中，附带了一系列的 C 语言头文件，这些头文件中是对 Perl 数据结构的定义。我们在下面进行简要的介绍：

XSUB.h	定义了 XSUB 接口
av.h	对数组变量的定义
config.h	在 Perl 安装时产生这个文件，包含配置信息
cop.h	全局指针
cv.h	转化结构
dosish.h	为 DOS 系统重新定义 stat, fstat 和 fflush
embed.h	为在 C 语言中嵌入 Perl 而定义
EXTERN.h	全局和扩展变量
gv.h	定义全局指针
handy.h	为在 C 语言中插入 Perl 而定义
hv.h	散列变量定义
INTERN.h	Perl 内置变量
keywords.h	Perl 语言的语法命令
mg.h	为使用 MAGIC 结构而定义
op.h	Perl 操作符
patchlevel.h	为当前 Perl 的补丁版本信息而定义
pp.h	为预处理指令而定义
Perl.h	Perl 的主要头文件
Perly.h	为 yacc 解析而定义
regexp.h	正则表达式的定义
scope.h	范围规则的定义
sv.h	标量变量定义

unixish.h 为 Unix 所作的特殊定义

1.3 Perl 的特性

1.3.1 免费和易于得到帮助

Perl 是免费的，它可以从 Internet 上自由下载并免费使用。正如 Linux 一样，它的这种自由的特性吸引了众多的人去使用它并为它开发模块。大多数 Perl 开发者也乐于将自己开发的模块放在 Internet 上，允许其它人自由使用。在 Perl 语言的开发者中，资源共享和乐于助人是两种普遍的美德。在国内，网易的 Perl 论坛上有一些热心人，可以帮你解决 Perl 问题。并且当前，有网友 wushan(wushan@371.net)和我在共同维护一个专业的 Perl 站点 <http://www.Perlchina.com>，你可以在其上得到 Perl 教程、应用范例、模块下载等很多服务。如果在阅读本书时有任何疑问，可以在该站点上提问并将得到我的解答。

1.3.2 解释性语言

Perl 是一门解释性的语言。因此，你可在 Perl 的解释器中直接运行 Perl 的 ASCII 文本程序，就如 BASIC 程序一样。这使你避免了执行一个编译性语言(例如 C 语言)那样繁琐的编译-链接-执行过程。这样可以方便调试，加快程序开发周期，同时，正如我们前面所说的，使 Perl 具有了跨平台运行的特性。我本人就经常在 WIN95 系统上开发 Unix 下运行的 CGI 程序，它们大多数无需任何改动即可在两种平台下运行。

当然，解释性语言也有其不足，那就是运行速度要慢于编译性语言，因为它在运行前要完成语法检查，并由解释器解释执行。这占用了额外的时间。

1.3.3 简单易学

Perl 是比 C 简单的一种语言，它只有三种变量类型：标量、数组和散列。但这三种变量类型非常的强大而好用，可以大大节省你编程的时间。同样的任务，用 Perl 来完成比用 C 要快得多。学习 Perl，你可以很快地写出自己的实用程序。实际上学习 Perl 的最好方法就是一边写程序一边学习，而不必象 C 那样学完 6 个月后才开始写实用程序。

使 Perl 编程变得简单的还有大量的免费模块，热心的自由开发者们开发出的模块可以使你的编程变得很轻松。对于其它商业性的语言来说，要使用其它人开发的模块是要付一大笔钱的。

1.3.4 功能强大

在深入学习 Perl 之后，你会认识到它是一门功能非常强大的语言。在文本操作方面，它是最好的语言，超过了 C、Pascal、BASIC 等任何一种，无论是在编程的简单性，还是运行速度方面都是最棒的。在其它方面，它同样也作得非常棒，例如网络编程、GUI 程序设计、面向对象程序设计等等。它比较适合中等规模的项目，能在最短的开发周期内为你提供最强大的功能。当然，如果要开发大规模的项目，目前还是倾向于用 C/C++。不过在 Perl 将来进一步成熟后，一定也能够适应大型项目的开发。

1.3.5 跨平台特性

我仍然要再次提到 Perl 的跨平台特性，这也许是我喜爱使用 Perl 的最大理由。我在个人的机器上使用 Windows 平台，习惯了它方便好用的图形用户界面。并且不愿意回到 Unix 的字符行命令中去。然而我现在经常要开发 Unix 下的应用，主要是 CGI 程序，Perl 能非常好地适应这一工作，你可以在 WIN32 系统下开发、调试程序，然后毫不费力地将它移植到 Unix 系统下去，当然也可以移到 MAC(苹果机)中。有 20 多种操作系统都可以运行 Perl，其中有一些甚至你都没有听说过的系统。

Perl 并没有嚷嚷“一次编写、到处运行”的口号，但它确实做到了这一点，而且并不比 JAVA 差。

1.3.6 与其它语言的混合编程

Perl 本身是用 C 语言写的，同时它也保留了和 C 语言混合编程的接口，你可以在 Perl 中使用 C 语言写的扩展，也可以在 C 语言中使用 Perl 模块。这将大大提高你的 Perl 或 C 编程能力。

同样，Perl 也可以和 JAVA 进行混合编程。在一个网络项目中，可以方便地利用这两种语言各自的优点来写不同的部分，它们通过套接字通信，可以很好地协同工作。例如，在 WWW 应用中，通常我们在后端使用 Perl 写一个服务器(server)，而在前端使用 Java Applet 与服务器通信。对于那些要快速处理文本的服务器应用而言，使用 Perl 是非常合适的。

1.4 Perl 能做什么

Perl 语言目前得到了广泛的应用，尤其是在网络领域。例如系统管理与维护、Internet 应用编程、Web 站点维护、CGI 开发等方面。下面我们进行一些简要的介绍。

1.4.1 系统管理与维护

Perl 出生之初，就被 Unix 系统管理员们用于系统的维护和管理。例如产生报表、维护文件、备份系统等等。当前，Perl 的系统管理能力大大增强了，在 Internet 上有大量的免费程序，可以用来管理系统用户、压缩备份系统文件、自动维护系统等等。此外，Perl 在 WIN32 平台上正得到越来越广泛的应用，对于管理一个 Windows NT 网络来说，它也是非常合适的。

1.4.2 Internet 应用编程

Perl 内置了强大的套接字能力，可以轻松地完成 c/s(客户/服务器)结构的程序。在 Internet 上，你可以使用 Perl 编写 Internet 服务，例如一个 Web 服务器，当然用得更广泛的还是编写客户程序。Perl 非常适合来做邮件处理。在本书中，我们有一章专门对此进行了讨论。此外，在新闻组、FTP 等的客户程序或网关开发方面，Perl 也可以做得很好。

1.4.3 Web 站点的维护和管理

Web 站点的维护和管理是 Perl 的强项。Perl 可以很轻松地完成诸如日志文件分析、站

点索引、Web 页面检查等事情。由于这些任务都涉及到文本处理，因此使用 Perl 不但容易编程，而且所写的程序处理文本的速度也是很快的。由于是管理 Web 服务器，使用 Perl 完成这类任务的另一个好处是，可以充分利用 CGI 程序的 Web 图形界面，从而使程序的用户接口更加友善好用。

1.4.4 Web 应用(CGI)开发

谈到 Perl，就不能不谈 CGI(公用网关接口)程序。Perl 是 CGI 编程事实上的标准语言，在 Internet 上有成千上万用 Perl 写的免费 CGI 程序，内容无所不包，如 BBS，聊天室，购物车等等。当前有一些新的技术如 ASP 或 PHP，可以写出比 CGI 程序更快的 Web 应用。Perl 在 Web 应用方面也在发展，如可以集成到 apache 服务器中的 mod_Perl 和集成到 IIS 服务器中的 Perlscript 等，它们比传统的 CGI 程序可以提高 2~5 倍的速度。并且，它们可以沿用已有的大部分 CGI 程序和模块。

1.5 安装 Perl 5

1.5.1 如何得到 Perl 5

Perl 5 可以从 CPAN 站点上得到。所谓 CPAN，是 Comprehensive Perl Archive Network(综合 Perl 文档网络)的简写，CPAN 的官方 WWW 站点是 www.cpan.org，上面提供了 Perl 的原代码、预编译的二进制发布以及 Perl 的所有模块和参考文档。在世界各地有很多 CPAN 站点的 FTP 或 Web 镜像站点，以方便你迅速地下载。CPAN 是所有 Perl 编程者的圣地，如果你需要某些方面的模块或程序，可以去 CPAN 寻找；如果你编写了一个新模块并希望贡献给全世界使用，可以将它提交到 CPAN。

如果你想得到已经预先编译好的可以用在 Windows 操作系统上的 Perl 解释器，可以从 ActiveState 公司的 Web 站点 <http://www.activestate.com> 上下载，他们称之为 activerPerl。ActiveState 是由微软资助的公司，致力于研究在 Windows 操作系统上运行的 Perl，并进行了大量的开发工作以使 Perl 能更好地进行 Windows 编程，例如 OLE、注册表操作和 Windows NT 网络管理编程等等广泛的方面。ActiveState 预先编译好了可以运行在 Windows 系统上的 Perl，并提供了一个安装程序界面，使你可以简单轻松地在 Windows 操作系统上安装 Perl。

Perl 当前的版本是 Perl5，5 是其主版本号，实际上还有更细的划分，如 5.004 或 5.005 等，不同的小版本号之间也有一些小的区别，但到目前为止，它们是向下兼容的。在我开始写这本书的时候，Perl 的最后版本是 5.005 版本，但是在六月初，Perl 发布了最新的稳定版本 Perl 5.6。在前言中给出的网站上，读者可下载这两个版本的 Perl，可以任选其中一个安装，本书的所有程序都可以在这两个版本上运行。

当然，安装 Perl 可以有两种方式：下载 Perl 的原代码然后在自己的机器上用 C 编译器编译后安装，这可以适合于任何一种操作系统；此外，你可以安装一个预编译好了的 Perl，例如 Active Perl。Perl 可以在各种不同的系统上安装，如 Unix 及其变种 Linux、Windows、macintosh、OS/2 等等。基于国内的应用现状，我们主要介绍两种最流行的系

统 Windows 及 Unix 上的安装。

1.5.2 在 Windows 系统中安装 Perl 5

如果要在 Windows (包括 95/98 或 NT, Windows 2000 也应该可以) 上安装, 最简捷的方法就是使用 activePerl, 它已为你预先编译好了, 可以像其它 Windows 程序那样简单的安装。在 activestate 的站点上可以得到 activePerl 的 5.005 或 5.6 版本。要安装它, 运行安装程序即可。安装程序将显示一个向导, 帮你完成安装, 基本上你只需要点击“下一步”即可。

如果你想在 Windows 系统上自己编译 Perl 原程序并安装, 也是可以的。但是, 这需要一个 C 语言的编译器。例如: VC++ 或 Borland C++ builder, 这些集成开发环境中都包含了 C 语言的命令行编译器。其中 VC++ 中名为 `nmake.exe`, 而 VC++ 中为 `dmake.exe`。下面以很流行的 VC++ 为例, 来讨论如何编译并安装 Perl:

1. 解压: 将下载的 Perl 源代码解压到硬盘上。在 Perl 目录中将有一个 win32 子目录, 其中含有编译文件 `makefile`。

2. 编译: Perl 需要使用 VC++ 的命令行编译工具 `nmake.exe` 来编译。在使用这个工具之前, 应确保我们正确的设置了使它工作的环境变量。由于通常我们在图形化的集成界面下使用 VC++, 所以通常机器上都没有正确设置环境变量。VC++ 提供了一个批处理文件用于自动设置环境变量, 这个批处理文件位于 VC++ 的 bin 目录下, 名为 `vcvars32.bat`, 我们运行它即可。随后, 可以进行编译了, 启动 DOS 环境, 在 Perl 的 win32 目录中运行命令:

```
nmake
```

这将根据文件 `makefile` 编译 Perl 源代码, 并生成 Perl 解释器的可执行文件。

3. 安装: 最后, 我们需要进行安装工作。在 DOS 的命令行环境下运行命令:

```
nmake install
```

这会将 Perl 安装到缺省目录下, 即 `c:\Perl` 下。

经过上面三个步骤, 你应该正确安装了 Perl, 此后可以正常的使用它了。

1.5.3 在 Unix 上安装 Perl

Unix 类的操作系统有很多变种, 但在 Perl 的源代码发布程序中, 随带了一个工具程序 `configure`, 它可以自动根据运行的系统来帮用户配置 Perl。此外还有一个类似的工具 `Configure` (头一个 C 为大写), 它完成同样的任务, 但在运行时需要用户根据实际情况回答一些问题以帮助配置。通常 `configure` 工作得很好, 但如果它失败了, 请试试运行 `Configure`。要在 Unix 系统上编译 Perl, 需遵循以下步骤:

1. 解包: 从 CPAN 上下载源代码文件 `latest.tar.gz` 后, 需要对其解压, 使用如下命令:

```
gunzip latest.tar.gz
```

```
tar xvf latest.tar
```

2. 配置 Perl: 通常运行 `configure` 即可, 如下所示:

```
sh configure --prefix='/usr/bin/Perl'
```

命令行开关 `prefix` 用于指定 Perl 的安装目录。

3. 编译 Perl: 运行以下命令:

```
make
```

这将完成 Perl 的编译过程, 生成二进制文件。

4. 测试和安装 Perl: 测试 Perl 是否正确安装, 并将编译生成的二进制文件以及库文件拷贝到由 configure 指定的安装 Perl 的目录。使用以下命令:

```
make test
```

```
make install
```

通过上面的步骤, 即在 Unix 系统中正确地安装了 Perl。

1.6 选择合适的编辑器

Perl 程序是普通 ASCII 文本, 通常, 任何可以用来编辑文本的编辑器都可以用来写 Perl 程序。但是, 为了有更高的编程效率, 应该使用合适的编辑器。

在 Unix 系统上, 可以使用 Xemacs, 它是最著名的程序编辑器, 提供了一个图形用户界面来供你编写程序文本。几乎在所有 Unix 类型的操作系统上, 都预先提供了 Xemacs 程序。

在 Windows 系统上的用户就没有这么幸运了, 虽然 notepad 可以用来编写 Perl 程序, 但它是非常不方便的。要编写稍大一些的 Perl 程序, 就不能考虑用它。现在在 Windows 平台上流行的一个著名的程序编辑器是 UltraEdit, 这个编辑器小巧而功能强大, 它认识 Perl 程序, 并能用不同的颜色来显示 Perl 的语句、变量和注释等。它的另外一个有用的功能是在一个单独的窗口中显示 Perl 程序中的子程序(函数)列表, 当双击函数条目时, 可以转到程序中相应的位置。这有些类似于 VC++ 的界面, 对于一个大的含有很多函数的程序来说, 是非常有价值的。

但 UltraEdit 也有不方便之处, 它无法与 Perl 的解释器集成。因此, 你在编写完程序后, 必须在 DOS 命令行窗口中来运行调试程序。通常, 编写、修改、调试程序总是交叉进行的, 这就带来了很大的不便。有些 Perl 编辑器可以解决这个问题, 如 PerlBuild 和 MultiEdit 等等。不过, 它们不是免费程序。在 <http://dujl.uhome.net> 上提供了这些程序的演示版, 可以允许你试用一定天数, 如果你想得到使用版本, 可以向相关的公司购买使用许可。在这些编辑器的帮助文件中, 通常都详细描述了如何购买许可。

1.7 Perl 未来的发展: Perl 6 和 Visual Perl

Perl 是一门有活力的语言, 处在不断的发展完善之中。当前, Perl 开发小组的计划是开发一个全新的 Perl 版本, 即 Perl 6。

Perl 6 将全部使用 C++ 语言重写, 其目的是使 Perl 更加快速、健壮和强大。Perl 6 将采取类似 linux 的开发模式, 由一个包括 Larry Wall 及其他贡献者在内的核心小组主持整个开发过程, 并且允许互联网上的任何人提出有价值的建议和贡献代码。由核心领导小组决定是否采纳。

关于 Perl 在 Windows 世界中的一个重大事件是 Visual Perl, 它是一个图形化的开发集成环境, 将包含在微软即将发布的著名的开发工具包 Visual studio 7.0 中。它提供了一个类似 Web 的界面, 用于使用 Perl 开发、调试 Web 应用程序。可以期望, Visual Perl 推出之后, 在 Windows 上开发 Perl 程序将更加简单、快速。

第 2 章 Perl 程序格式

2.1 Perl 程序预览

在我们实际开始学习 Perl 编程之前，我们不妨来看看一个典型的 Perl 程序，从而得到对 Perl 程序的一个总体印象。这有助于我们今后在各章中的学习，可以站在全局的角度来了解它的具体语法，而不会在学习每一章时，得到支离破碎的印象。下面是一个示例程序，它几乎包含了 Perl 所有的语法特点。

例程 2-1

```
#!/usr/bin/Perl
use strict;
my $say="hello world";
my $timeObject=new Time;
for ( my $i=1;$i<6;$i++) {
my $curTime=$timeObject->getTime();
say($say,$curTime);
sleep 1;
}
#####
sub say
#####
{
my ($content,$time)=@_;
print "the program say: $content at $time";
} ##say
#内嵌的包，实现类 time。
package Time;
#####
sub new #00-8-5 22:25
#####
{
my $class=shift;
my $self={};
bless $self,$class;
return $self;
} ##new
#####
```

```
sub getTime #00-8-5 22:29
```

```
#####
```

```
{
```

```
my $class=shift;
```

```
my($sec,$min,$hour,$day,$month,$year)=localtime(time);
```

```
$year+=1900;
```

```
$month+=1;
```

```
$class->{'curTime'}="$year-$month-$day $hour:$min:$sec \n";
```

```
} ##getTime
```

```
=head1 包名
```

```
本模块实现类 time:
```

```
=head1 使用示范
```

```
use Time;
```

```
$time=new time;
```

```
$curTime=$time->getTime();
```

```
=head1 方法
```

```
=head2 方法 getTime
```

这个方法不接收任何参数，它返回当前的时间，使用的格式是：年-月-日:时:分:秒。

```
=cut
```

以上这个程序每隔一秒钟向屏幕上打印信息 hello world，同时还输出当前的时间。下图是该程序运行后的结果：

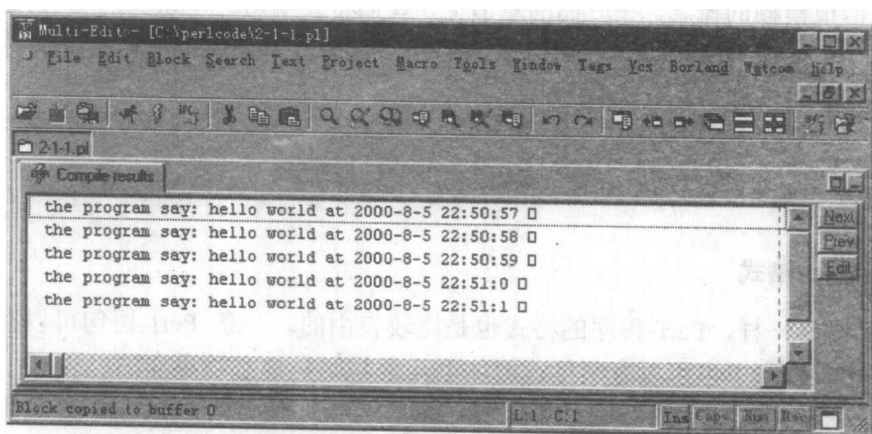


图 2-1

虽然这个程序仅仅完成了一个简单的任务，但它对于一个初学者来说也是非常深奥的，因此读者看不懂是很正常的。下面我们对这个程序的格式作一些简要的解释，具体的内容将在以后各章中学习。

首先，第一行的 `#!/usr/bin/Perl` 用于 Unix 类的操作系统中，它告诉操作系统使用 `usr/bin/Perl` 目录下的解释程序来解释本程序。在 Windows 系统中，该行不起作用，系统会忽略它。而将它看成一个注释，Perl 中的注释是以“#”号开头的。Windows 系统使用

另外的方式来关联应用程序和其文件。可以用资源管理器来关联以.pl 为后缀的程序，使系统总是启动 c:\Perl\bin\Perl.exe 来执行它。

接下来是一条条的 Perl 语句，我们看到，这和其它编程语言基本相似。不过有一点是 Perl 所特有的，即其变量前要使用类型声明符，如上面程序中的“\$”，它指明了标量类型的变量。在后面的章节中我们要学到更多的变量类型。

开头是一个比较特殊的语句，即“use strict;”。use 表示调用一个模块，而 strict 是一个特殊的模块，它被称为 pragma。strict 模块进行一些预处理工作，用于定义一些 Perl 的语法规则，以便除去程序中的错误。因此，看上去好象它改变了 Perl 的语法。

同其它语言一样，Perl 也可以自定义函数(或称为子程序，它们在 Perl 中指同一个意思)。函数用关键字 sub 定义，然后用一对大括号括起函数体。

接下来是一个奇怪的东西：package(包)。包大概也是 Perl 所特有的语言特征，一个包用于定义一个针对其中变量的名字空间。包中的变量只在其内有效，在外部不能直接调用。如果要在外部调用，必须在变量前加上包名。通过使用包，将程序分成了单独的块，彼此不会产生变量名字的冲突，这对构造一个大型项目来说是很有用的。包主要的作用是构造模块和类。在本例中，包 Time 构造了一个简单的类，类是一种特殊的模块，是面向对象的一种语法。在后面我们要详细讨论这一切：包、模块、面向对象与类。

最后，我们将看到一段文档。以=head1 之类的标记开头。Perl 语言中允许嵌入文档，以描述一个程序或模块的用法。Perl 解释器在运行程序时会忽略这些文档，但可以使用其它的专用程序来观看这个文档。这种文档称之为 pod 文档，稍后我们再详细讨论。

好了，我们现在看了一个典型的 Perl 程序，并听到了许多概念。它给我们一个关于 Perl 的比较全面但很模糊的概念。在后面的章节中，我们将要学习这里谈到的所有概念和语法，让一切对你来说变得清晰可用。然后，我们还要用很多篇幅来讨论如何用 Perl 来解决实际问题，如数据库、网络、CGI 编程等等。

2.2 Perl 程序的格式

2.2.1 程序编写格式

如同 C 语言一样，Perl 程序的格式也是比较自由的。一条 Perl 语句可以在任意一行的任意列开始，程序中也可以有空行。但如下几条规则是你必须遵守的：

1. 任意一条可执行的命令必须以分号结束。记住这是针对可执行命令而言的，用于括起程序段的大括号并不是可执行命令，因此其尾部无需加分号。
2. Perl 语言是大小写相关的。因此，\$VAR 和 \$var 不是同一个变量，而 print 和 Print 也不是同一个命令。请时刻注意你的大小写。在 Perl 中，所有的内置函数都是小写的，而所有的缺省句柄都是大写的。
3. 在定义你自己的函数或句柄时，注意其名字不要和 Perl 内置的相同。
4. Perl 语句中命令名和变量之间应该用一个以上的空格隔开。在 Perl 中使用空格的格式基本上也是自由的，一个或多个空格是一回事。

在 Perl 中，通常一行放一条语句。然而，只要你愿意，也可以一行放多条语句，这些