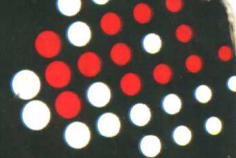


软件开发技术丛书



Perl 5

编程详解

Perl 5 Complete

(美) Edward S. Peschko, Michele DeWolfe 著 康博创作室 译



机械工业出版社
China Machine Press



McGraw-Hill

软件开发技术丛书

Perl 5 编程详解

(美) Edward S. Peschko 著
Michele DeWolfe

康博创作室 译



机械工业出版社
China Machine Press

Perl是一种用于工具构造的强大语言, Perl 5 是它的最新版本。它具有语法简单、用途广泛、跨平台、面向对象、嵌入文档、与C/C++链接、易调试等特点,使它成为当今软件世界里一门重要的语言。

本书以生动而又严谨的论述、丰富的示例,详细讲解了这个语言的基本功能和高级技巧,内容深入浅出、详尽透彻。无论是Perl的初学者还是老手,都能从本书收受益。本书是Perl 5语言最全面的参考书。

Edward S. Peschko & Michele DeWolfe: Perl 5 Complete.

Authorized translation from the English language edition published by The McGraw-Hill Companies, Inc.

Copyright 1998 by The McGraw-Hill companies, Inc.

All rights reserved.

本书中文简体字版由机械工业出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

版权所有,翻印必究。

本书版权登记号:图字:01-98-2590

图书在版编目(CIP)数据。

Perl 5 编程详解 / (美)帕斯科 (Peschko, E. S.), (美)德沃夫 (DeWolfe, M.) 著;康博创作室译. -北京:机械工业出版社, 1999 10

(软件开发技术丛书)

书名原文: Perl 5 Complete

ISBN 7-111-07283-9

I. P... II. ①帕... ②德... ③康... III. Perl 5 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第25590号

机械工业出版社 (北京市西城区百万庄大街2号 邮政编码 100037)

责任编辑:吴怡

三河永和印刷有限公司印刷·新华书店北京发行所发行

1999年10月第1版第1次印刷

787mm × 1092mm 1/16 · 54.25印张

印数: 0 001-5 000册

定价: 79.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

前 言

哈姆雷特：你看远处的云彩是否像骆驼一样？

波洛尼厄斯：啊，它确实像一峰骆驼。

哈姆雷特：我看它倒像一只黄鼠狼。

波洛尼厄斯：它是像一只黄鼠狼。

哈姆雷特：或者它像一条鲸鱼？

波洛尼厄斯：它真像一条鲸鱼！

哈姆雷特：好了，我该到母后那儿去了，再见。

本书是大量劳动的结晶。不仅如此，它还是一次磨难，一次炼狱般的考验。写作本书所花的时间几乎是预期的两倍，正好用了整整一年半才完成。

当尘埃落定，本书结束后，我开始考虑为什么本书是这么一种巨大的劳动。笔者的本意是填补Perl文献中的空白，而这个空白正变得越来越明显，即缺少一本优秀的Perl语言开发人员指南。

在我进一步考虑这个问题后，上面哈姆雷特所说的话浮上了我的脑海。Perl被称作“骆驼”语言，其作者Larry Wall就喜欢说“Perl可能不好看或不好闻，但它能完成任务。”

但将哈姆雷特的话应用到Perl时更形象。Perl不仅是“骆驼”语言，还是可改变的语言，就像哈姆雷特脑海中的骆驼形云彩一样可变。Perl的形状总是在变化，Perl有一个特点，即不会受制于某个特定的问题。你对Perl了解得越多，你用它所能做的事情就越多。

Perl是文本处理语言吗？是的，它是这么设计的。它是网络语言吗？是的，它内置了插口/简单客户-服务器协议。

它是系统管理语言吗？是的，它可管理全球范围的网络。Perl能让你更快地完成简单任务吗？是的，“单行”脚本是很常见的。

Perl是Web脚本编写语言吗？是的，它是因特网上最流行的脚本编写语言。

Perl的功能不胜枚举，例如：

- 用Perl生成代码。
- 用Perl（通过Tk）快速编写GUI。
- 在C和C++程序中进行Perl编程，在Perl程序中进行C/C++编程。
- 使用Perl过滤邮件。
- 测试GUI应用程序。
- 使用Perl进行数据仓库编程。
- 执行对其他语言的源控制。
- 在Web上进行搜索。

Perl是无所不能的。你可以用Perl编写一个有一百万行源代码的项目，也可以编写一行代码完成有意义的事情。你可以编写与其他软件包的接口，如与Mathematica、Microsoft IIS服务器，Oracle、Sybase、Informix的接口，实际上，它可以完成用C“后端”完成的任何事情。

你可以编译你的Perl代码或将Perl代码编译到Java内。总之Perl是一门成熟的经过良好测试

的语言，它几乎可用于任何平台（无须成本），而你获得了一项很难划分其类属的技术，但这种技术将成为你的日常工作中必不可少的部分。

本书尽量向读者介绍Perl语言的方方面面，更重要的是通过大量例子来介绍这门语言。

学习本书不要求对Perl事先进行了解，你只需要熟悉一般的编程概念即可。本书从Perl的基本概念入手，对Perl的细节进行了深入的介绍，对如何调试各种Perl程序以及如何用Perl有效地进行类编程给出了相当完整的阐述。本书尽量通过例子将Perl的核心概念放在一起，为读者实际进行Perl编程提供帮助。如果你需要这样的编程例子，那么本书正适合你。

Perl本身是一门不断发展变化的语言，Perl编译器正在出世，同时还集成了Windows和Unix端口，线程化以及Java/ Perl接口等。笔者无法包罗万象，因此，笔者将本书定位于一种使读者学习Perl并学好Perl的方法，然后将从本书中学到的知识应用到自己的日常工作中。

希望读者喜欢本书，尽管本书很长，但它却很完整。笔者尽量使书中的例子相互关联，而且尽可能使书中的代码在技术上正确。按照笔者的观点，一本技术上不准确的书比无用更糟，因为你没有正确地学习，因此需要花很多时间来弥补由于不正确的学习所带来的损失。因此请读者不要保持沉默，如果你发现了错误请告诉我。如果看到自己认为特别重要的内容被漏掉了，也请告诉我。我的网址为：epeschko@apxtech.com。

英文原书书号为：ISBN-0-07-913698-2

原出版社网址为：www.computing.mcgraw-hill.com

目 录

前言

第1章 设置Perl 5的环境1

1.1 概述1

1.2 Perl 5的安装1

1.3 安装自己的Perl9

1.3.1 在Unix上安装Perl9

1.3.2 在Windows 95 / NT上构造Perl 515

1.3.3 在Macintosh上构造Perl21

1.3.4 在OS / 2上构造Perl23

1.3.5 在VMS上构造Perl26

1.3.6 在MS-DOS上构造Perl27

1.3.7 安装Perl的附加程序包29

1.3.8 安装文档31

1.4 其他支持33

1.4.1 Web 站点33

1.4.2 新闻组33

1.4.3 邮件列表33

1.4.4 专业支持34

1.5 小结34

第2章 30000英尺高度上的Perl: Perl的

概述35

2.1 概述35

2.2 引论35

2.3 运行Perl36

2.3.1 用一般方式运行Perl36

2.3.2 运行Perl的原则37

2.3.3 Perl的开关38

2.3.4 在不同的操作系统上运行Perl38

2.3.5 运行Perl的小结42

2.4 通用的Perl语法42

2.4.1 Perl变量42

2.4.2 其他变量44

2.4.3 常见错误47

2.4.4 在30000英尺的高度上观看Perl的

小结49

2.5 Perl的一些例子49

2.5.1 例1a:从一组ASCII格式平面文件中 访问数据并打印数据50

2.5.2 例1b:从一组Excel格式平面文件中 访问数据并打印数据52

2.5.3 例2:当某个进程完成的时候,向其他 项目成员发送电子邮件54

2.5.4 例3:与Internet服务提供商进行连 接55

2.5.5 例4:在不同的系统上不能提供的功 能: cat57

2.5.6 例5:在不同的系统上不能提供的功 能: grep58

2.5.7 例6:在不同的系统上不能提供的功 能: find60

2.6 小结62

第3章 Perl中的变量63

3.1 概述63

3.2 基本的Perl数据类型63

3.2.1 标量64

3.2.2 操作标量的函数和运算符71

3.2.3 有关标量内容的小结74

3.3 数组和列表74

3.4 哈希变量80

3.5 句柄84

3.5.1 对文件句柄进行操作85

3.5.2 文件句柄小结89

3.5.3 例子89

3.6 Perl变量小结92

第4章 Perl的控制结构和运算符94

4.1 概述94

4.1.1 Perl的本质94

4.1.2 Perl的控制结构96

4.1.3 for控制结构98

4.1.4 foreach控制结构99

4.1.5 if..else..elsif控制结构101

4.1.6 对控制结构的控制102

4.1.7 标记控制结构	106	6.5 带有数组的列表上下文	157
4.1.8 关于Perl控制结构的最后 一些内容	108	6.6 哈希结构怎么样?	158
4.1.9 Perl运算符简介	108	6.7 判断上下文的方法	160
4.1.10 Perl运算符的优先级	109	6.7.1 使用内置函数以确定数据类型	160
4.1.11 使Perl的表达式清晰的技术	110	6.7.2 使用运算符确定数据类型	161
4.1.12 使用优先级表	111	6.7.3 高级上下文	163
4.1.13 Perl语言中常用的运算符	112	6.7.4 利用位置来确定数据类型	164
4.1.14 Perl控制结构和运算符的小结	116	6.7.5 带有内插的上下文规则	165
4.1.15 例子: Perl中常见的表达式	116	6.7.6 上下文和函数调用	166
4.1.16 Perl表达式模式小结	122	6.7.7 数组引用和上下文	167
4.2 小结	122	6.7.8 控制结构和上下文	168
第5章 函数和作用域	124	6.7.9 结论	169
5.1 概述	124	6.7.10 例子	169
5.2 函数	124	6.8 小结	174
5.2.1 语法	125	第7章 引用	175
5.2.2 参数栈	125	7.1 概述	175
5.2.3 参数栈的小结	128	7.2 引论	175
5.2.4 返回值栈	128	7.3 Perl 5的引用: 硬引用和软引用	176
5.2.5 把多个数组或哈希结构传递给 函数	132	7.4 引用和作用域: 垃圾收集	192
5.2.6 Perl函数的注意事项	134	7.5 例子	193
5.2.7 注意事项的小结	137	7.6 小结	200
5.2.8 函数的小结	137	第8章 关于Perl的引用以及常见数据结构 的另外一些内容	202
5.3 Perl作用域的方法	138	8.1 概述	202
5.3.1 作用域语法	139	8.2 数组的数组	203
5.3.2 “my”和词法作用域	139	8.2.1 如何识别数组的数组	203
5.3.3 “local”和动态作用域	143	8.2.2 对数组的数组的直接访问	203
5.3.4 “use strict”	145	8.2.3 在数组的数组中一个常见的错误 概念	205
5.3.5 Perl中作用域规则的小结	146	8.2.4 创建数组的数组	206
5.4 子程序的一些例子	146	8.2.5 数组的数组的访问函数	210
5.4.1 使用递归的子程序的例子	146	8.3 哈希结构数组	213
5.4.2 使用引用的子程序的一些例子	148	8.3.1 匿名引用结构	214
5.4.3 使用wantarray的子程序的例子	149	8.3.2 对哈希结构数组进行访问的小结	216
5.4.4 作用域的例子	151	8.3.3 哈希结构数组的构造函数	216
5.5 小结	152	8.3.4 哈希结构数组的小结	223
第6章 Perl 5中的上下文	154	8.3.5 哈希结构的哈希结构	223
6.1 概述	154	8.3.6 访问哈希结构的哈希结构的方法 小结	225
6.2 数据上下文简介	154	8.3.7 哈希结构的哈希结构的小结	230
6.3 标量上下文	155	8.4 数组的哈希结构	230
6.4 列表上下文	156		

8.5 常见的数据结构的小结	235	10.4.3 内部命名变量	305
8.6 小结	235	10.4.4 内部单字符变量	308
第9章 正则表达式	237	10.4.5 内部变量小结	313
9.1 概述	237	10.5 小结	313
9.2 引论	237	第11章 Perl5 杂项	315
9.3 Perl正则表达式的基础	238	11.1 概述	315
9.3.1 原则1	239	11.2 格式	316
9.3.2 原则2	240	11.2.1 格式的语法	316
9.3.3 原则3	240	11.2.2 格式如何工作	318
9.3.4 原则4	241	11.3 Coderefs	320
9.3.5 原则5	242	11.3.1 Coderefs的格式	320
9.3.6 原则6	247	11.3.2 匿名子程序	321
9.3.7 原则7	256	11.3.3 代码引用小结	324
9.3.8 原则8	257	11.4 Globbing	324
9.3.9 正则表达式原则概括	259	11.4.1 Globbing技巧	325
9.3.10 正则表达式修饰符	260	11.4.2 Globbing 和 Exporter	325
9.3.11 匹配和g运算符	263	11.5 用Perl运行程序	326
9.3.12 修饰符和环境	264	11.5.1 Perl编译步骤	327
9.4 正则表达式的例子	266	11.5.2 关于Perl语法分析的更多内容	328
9.5 小结	273	11.5.3 BEGIN / END的其他用途	330
第10章 Perl的内置函数和变量	274	11.5.4 BEGIN / END小结和流控制	339
10.1 概述	274	11.6 Eval	339
10.2 引论	274	11.6.1 使用eval的原则	340
10.3 Perl的内置函数	275	11.6.2 eval的用法	340
10.3.1 在内置函数后面的原则	276	11.6.3 eval小结	343
10.3.2 主函数	276	11.7 小结	344
10.3.3 用于格式化数据的操作	277	第12章 例子	345
10.3.4 对文件和变量的读写操作	279	12.1 概述	345
10.3.5 文件句柄的读写操作的小结	281	12.2 功能示意图	346
10.3.6 关于变量的操作	282	12.3 grep程序	346
10.3.7 变量操作小结	290	12.3.1 环境grep	347
10.3.8 时间函数	290	12.3.2 cgrepStack.p用法	354
10.3.9 时间函数小结	292	12.3.3 filegrep用法	354
10.3.10 调试函数	292	12.4 文本/文件操纵	358
10.3.11 调试函数小结	294	12.4.1 给文件做索引	358
10.3.12 Perl对操作系统的接口	294	12.4.2 比较和对照目录结构	360
10.3.13 文件运算符	300	12.4.3 删除垃圾文件	364
10.3.14 内部Perl函数小结	301	12.4.4 解开一个简单密码	365
10.4 内部Perl变量	302	12.4.5 匹配Perl性质的数字的正则表 达式	367
10.4.1 内部文件句柄	303	12.5 代码生成器	368
10.4.2 内部令牌	304		

12.5.1 自动化Telnet	369	14.2 Namespaces	451
12.5.2 使ftp自动化	377	14.2.1 名字空间的原则	451
12.6 OLE自动化: 对象链接和嵌入	380	14.2.2 实用的名字空间和包提示	454
12.6.1 OLE简介	381	14.2.3 名字空间小结	455
12.6.2 用OLE对Word进行更多控制	383	14.3 库和关键字require	455
12.6.3 Excel例子: 合并两个电子表格	385	14.3.1 require的用法	455
12.6.4 通过MAPI发送邮件	387	14.3.2 require小结	458
12.7 WinPerl++	389	14.4 带有use的模块	459
12.7.1 Excel	390	14.4.1 use的用法	459
12.7.2 带有Guido的Word	393	14.4.2 常用use指令	461
12.7.3 自动生成模板文件和IDE	393	14.4.3 use的更多例子	463
12.7.4 Guido小结	394	14.4.4 use小结	465
12.8 Lib WWW例子	394	14.5 从磁盘安装模块和库	466
12.8.1 从Web上获取html页	394	14.5.1 库和模块的路径映射:@INC和 %INC	466
12.8.2 从Web表单中获取http标记	395	14.5.2 库和模块路径映射小结	469
12.8.3 分析用户信息在线表单的表单建 造器	396	14.6 建立开发环境	469
12.9 CGI例子	400	14.7 小结	473
12.9.1 可嵌入的Web计数器	400	第15章 Perl中的抽取化和模块编程	474
12.9.2 Meta-Webcrawler	403	15.1 模块化编程概念	474
12.10 错误跟踪器	409	15.1.1 模块化编程的代价	474
12.11 数据库	420	15.1.2 把过程脚本转为模块化脚本	476
12.11.1 数据库监视器: 查看数据库 信息	421	15.1.3 抽取变量处理	484
12.11.2 数据库小结	426	15.1.4 结论例子	490
12.12 PerlTk例子	427	15.1.5 抽取的公用代码的小结	493
12.12.1 小型Tk教程	427	15.1.6 模块化编程的示例	494
12.12.2 PerlTk介绍小结	431	15.2 小结	504
12.12.3 邮件过滤器	431	第16章 对象的语法	506
12.12.4 从Tk应用程序内部运行脚本	435	16.1 概述	506
12.12.5 Tk举例小结	442	16.2 基本对象原则	506
12.13 小结	442	16.2.1 一个简单的例子	507
第13章 面向对象编程简介	443	16.2.2 构造函数	508
13.1 概述	443	16.2.3 对象方法	510
13.2 理解面向对象的编程	444	16.2.4 对象数据	512
13.2.1 抽象	444	16.2.5 对象算法	512
13.2.2 关于学习对象编程的困难	447	16.2.6 析构函数	513
13.2.3 初学者学习面向对象编程指南	449	16.2.7 基本对象原则小结	514
13.3 小结	450	16.3 中等/高级对象原则	514
第14章 库和模块的语法	451	16.3.1 类与对象	515
14.1 概述	451	16.3.2 继承性	518
		16.3.3 继承小结	523

16.3.4 重载.....	524	20.4.2 例2	645
第17章 普通对象	533	20.4.3 例3	655
17.1 概述	533	20.5 小结	660
17.2 普通类和对象	533	第21章 分层和Perl	661
17.2.1 普通类和对象的定义	534	21.1 概述	661
17.2.2 示例概述: 木板游戏Strategem	535	21.2 分层概念	661
17.3 小结	556	21.2.1 在读者看到分层时判别它	662
第18章 将旧代码转变为对象代码	557	21.2.2 分层概念的详细介绍	664
18.1 概述	557	21.2.3 基本的设计样式	677
18.2 设计决定: 模块与对象	557	21.2.4 最后的分层例子	698
18.2.1 模块编程特征	557	21.3 小结	719
18.2.2 面向对象编程及封装	558	第22章 Perl 开发环境	720
18.2.3 在对象与模块之间选择	561	22.1 概述	720
18.3 将程序代码转变成对象	569	22.2 开发程序: Perl式模型	720
18.3.1 例1: ftp和telnet Expect对象	569	22.2.1 调试扩展程序	721
18.3.2 例2: 配置文件	579	22.2.2 编译器和错误检查器	722
18.3.3 重写telnet.p	584	22.2.3 Perl式模型	723
18.4 小结	587	22.3 Perl开发工具	723
第19章 维护代码文档的类	589	22.3.1 缺省的Perl调试器	724
19.1 概述	589	22.3.2 使用调试器的一个例子	741
19.1.1 问题: 解决代码文档恶梦	589	22.3.3 用Devel::Coverage进行范围测试	746
19.1.2 第1步: 查看可用的资源	589	22.3.4 速度调试: 配置程序	748
19.1.3 第2步: 解决问题的提议	594	22.3.5 Perl编译器	764
19.1.4 第3步: 正式设计和伪代码	602	22.3.6 编译器的小结	765
19.1.5 第4步: 全面开发	607	22.4 小结	766
19.1.6 第5步: Pod::Checker文档化	625	第23章 Perl调试技巧	767
19.1.7 第6步: 回归测试	629	23.1 概述	767
19.2 小结	629	23.2 熟悉Perl中的调试编程	767
第20章 继承性和Perl	630	23.3 正确编程的技巧	768
20.1 概述	630	23.3.1 Perl的错误消息	768
20.2 更好的作业工具: 对象图	630	23.3.2 样式技巧	771
20.2.1 ISA	630	23.3.3 Perl安全防护	771
20.2.2 HASA	631	23.3.4 -w标志和use diagnostics	775
20.2.3 对象图的小结	632	23.3.5 Lint模块	779
20.3 继承性概述	632	23.3.6 use strict、-w和use diagnostics 以及Lint的小结	783
20.3.1 继承性的简要回顾	632	23.4 定位错误: 用use carp进行堆栈跟踪	783
20.3.2 继承性的优缺点	634	23.4.1 carp()	784
20.3.3 继承性的优缺点小结	639	23.4.2 cluck()	785
20.4 何时使用继承性或者何时不使用继承 性的例子	639	23.4.3 croak()	785
20.4.1 例1	640	23.4.4 confess()	785

23.4.5 use Carp的小结	786	表格(pss)	826
23.5 运行中调试——动态查找问题	786	25.1 概述	826
23.5.1 %SIG、\$SIG{'_WARN_'}和 \$SIG{'_DIE_'}	786	25.2 为什么设计Perl电子表格	827
23.5.2 \$SIG{'_DIE_'}	787	25.3 高层设计	828
23.5.3 \$SIG{'_WARN_'}	788	25.3.1 单元格	829
23.6 成功的数据调试: Data::Dumper() 和Tie::Watch()	788	25.3.2 电子表格	829
23.6.1 Dumper()和调试对象	789	25.3.3 进行GUI设计	830
23.6.2 Tie::Watch	790	25.3.4 电子表格布局和相应的设计	832
23.7 调试标志	792	25.4 算法:通过utxt处理数据	838
23.7.1 -D的值	792	25.4.1 基本数据	838
23.7.2 -D和调试常规表达式的小结	794	25.4.2 可求值的表达式	838
23.8 用于调试的编程辅助工具	794	25.4.3 内置的电子表格函数	839
23.8.1 例1: 警告指令或实用程序	795	25.4.4 间接引用其他单元格	839
23.8.2 例2: 使用联系创建一个安全的 对象	796	25.4.5 utxt在设置数据和对单元格的引用 方面的基本用法	839
第24章 利用已有的工具创建CGI脚本	799	25.4.6 utxt使用Perl语法的用法	840
24.1 开始: 命令行选项	799	25.4.7 间接引用标志: ==的用法	841
24.2 内部: 输入、输出和数据一致性	808	25.4.8 避免无限的依赖性问题	844
24.2.1 最大限度地利用POST	808	25.4.9 算法小结	845
24.2.2 浏览器特有的特征: Cookie	809	25.5 开发电子表格	845
24.2.3 返回身边的问题: 数据	809	25.6 调整新电子表格	846
24.2.4 通往CGI共存的前几个步骤	811	25.6.1 问题域	847
24.3 又一个反复过程	815	25.6.2 可利用的工具	847
24.4 小结	825	25.6.3 写接口文件	848
第25章 Perl / Tk中的Perl电子		25.6.4 组装在一起	852
		25.6.5 调整的电子表格的小结	857
		25.7 小结	857

第1章 设置 Perl 5 的环境

本章的目的是向读者介绍Perl以及说明如何有效地使用Perl。要有效地使用Perl，在很大程度上取决于是否在系统上正确地安装了Perl。安装Perl仅仅是第一步。用户还需要或应当知道如何发现以下有关内容的信息：

- 根据自己的情况定制Perl。
- 定位有关Perl问题的答案。
- 和专家谈论有关Perl的问题。
- Perl专业支持和培训。

用户还必须能够使用这些文档去解决日常编程过程中所遇到的问题。无论是在商业版本还是在免费版本中，都可利用Perl的某些最佳联机文档。遇到问题时，我们应能轻松地使用它。本章将让读者熟悉这些资源，并且指导读者找到更多信息。

1.1 概述

在本章中，我们将经历大多数人在他们的操作系统上安装Perl时所要经历的每一个步骤。让我们先来对Perl编程做出一个决定：打算使用已经安装好的Perl，还是安装新版本的Perl？其实，每种方案都有优缺点。

我们不妨依次都试一下。首先，我们说明如何查找已经安装在Unix、NT/Win95、Macintosh、OS/2和VMS上的Perl，并且说明在这些已安装好的Perl中查找一些什么内容。其次，我们将在五种操作系统上看一下如何从头安装Perl，并且说明如何测试新版本的Perl以及在安装好的Perl中查找什么内容。第三，我们将看一下如何安装Perl模块以及如何利用人们在网络上已经编好的模块来定制安装过程。这些封装的解决方法包括使用CGI、利用Perl/TK来生成Graphic User Interfaces(图形用户界面)等等。第四，我们将看一下Perl文档，了解如何使用它，以及如何以html、postscript、man和纯文本的形式把它打印出来。最后，我们将看一下Internet上现有支持Perl的系统。

Perl确实是一种富有生命力且不断更新的语言，还没有哪个标准委员会给出该语言最好的规则。使用它的人们决定了它的内容。因此，在Perl领域中总存在着发展、变化和新的东西。进入“Perl流”是有效地学习这种语言的一部分。

1.2 Perl 5的安装

Perl可以安装到绝大多数的平台上，而在这里我们无法述及所有的平台。用户几乎可以让Perl 5运行在每一种操作系统上，像Unix、Windows NT、Windows 95、Win3.1、MS-DOS、VMS、MacOS、plan9、AS/400以及运行MVS(或OS/390)的主机上。在此书出版时，Perl也许还能在VM/ESA上运行。事实上，如果读者是主机程序员并且想知道使用Unix感觉如何，学习Perl是一种理想的方法。同样地，如果读者是Unix程序员，想利用NT来测试一下自己的水平，学习Perl也是一种理想的方法。

Perl所涉及的操作系统很多，我们必须选择一些操作系统讲解。我们将在以下操作系统中

安装Perl:

- 1) Unix
- 2) NT/Windows 95
- 3) OS/2
- 4) VMS
- 5) MacOS
- 6) MS-DOS

注意 我们并没有提及到Perl支持的所有系统。另外支持Perl的操作系统还有: AmigaOS, Atari, VOS, Next, Machten, lynxOS以及一些我们未曾听说过的系统, 如mpe, acorn, aos等等。

如果用户使用的是一种本书没涉及到的操作系统, 那么我强烈地建议用户使用上面列出的操作系统。本书中大多数的例子都可以在这些操作系统上使用, 但我不能保证用于所有操作系统, 这本书只保证用于Windows NT、Windows 95和Unix系统(就像在计算机世界中所保证的任何事情那样)。除了一些例外的情况(已被标明), 所有的例子显然都能运行在Windows和Unix平台上。

开始安装Perl 5

关于如何开始, 有两种基本选择, 可以:

- 1) 查找已安装的Perl。
- 2) 建立Perl, 方法是下载Perl的可执行预建文件或者自己建立一个。

我们将从头复习一些内容, 从对Perl一无所知到能正确地安装Perl。为什么?因为Perl是一种世界语, 这意味着在每处它看起来和感觉起来都非常相似。因此, 使用Perl的一个重要的理由就是熟悉未知环境, 无论是Unix、NT、VMS还是其他任何环境。如果在这些操作系统的任何一种上编译Perl, 那么学习那种操作系统的必要性也就在百分九十以上。我们假定读者不熟悉将要安装Perl的操作系统。如果读者不是新手, 不必阅读下面的文字。

1. 查找已经安装的Perl

假使随处都安装了Perl, 使用预安装的Perl具有几项优点:

- 1) 可维护性——用户不必自己维护Perl。如果遇到什么困难, 可请教系统管理员。
- 2) 集中管理——可以要求安装Perl的人扩充或修改它。
- 3) 可跟踪性——如果很多人都使用Perl, 他们可以使用同一个拷贝, 而不是每人都有一个拷贝。这样如果你正在调试其他人的程序, 可以保证不是因为Perl版本而出现的错误。
- 4) 低的系统开销——用户不必做有关安装的任何事情。

同样预安装也有一些缺点:

1) 无法控制——如果用户想自己更新Perl, 这是不行的。通常Perl位于一处集中的地方, 像Unix上的/usr/local/bin, 并且只能由系统管理员来修改。这就意味着, 当想修改Perl时, 必须征得系统管理员的同意。

2) 无灵活性——用户受制于Perl的版本以及系统管理员安装时设置的选项。

如果正在使用已有的Perl, 也就把维护Perl的责任交给了系统管理员。如果这样做, 也就失去了一些自己安装Perl时所具有的自由。当然, 失去多少自由依赖于安装Perl的操作系统;所使用的系统的范围越广, 从定制安装中获得的益处就越多。

如果读者决定使用已经安装的Perl, 就必须确定好它的位置并且保证它是正确的版本。查

找Perl的目的有两个:一是找出所安装的Perl的版本;二是应当找出Perl安装在什么地方。在系统上查找Perl是一项同系统多少有点关系的任务。让我们熟悉一下每种系统。

(1) 在Unix系统上查找Perl

为了查出在Unix系统上安装的是哪个版本的Perl, 可以使用以下命令:

```
prompt% perl -v
```

命令中的prompt%是命令提示符, perl是Perl可执行文件的最常见的缺省名字。如果该命令失败, 可以使用其他命令:

```
prompt% perl5.00X -v
```

或者

```
prompt% perl5 -v
```

其中的perl5和perl5.00X(X代表数字)是用于perl可执行文件的其他常见的可供选择的名称。此后, 可以使用whence或which, 以便告诉用户在什么位置安装了Perl的可执行文件。以下的命令将给出Perl可执行文件的完整路径:

```
prompt% whence perl
```

如果该命令不起作用, 试一下以下的命令:

```
prompt% which perl
```

因为使用whence还是which取决于系统类型。

只有当Perl位于用户的路径中时——如果系统设置包括了安装Perl的目录——这些命令才会起作用。如果在使用以上这些命令时无法发现该目录, 为了找到Perl, 可以使用一些在Unix平台上极为常用的系统命令。例如, find可容易地找到Perl所在的位置。以下命令:

```
prompt% find/-name "perl*" -print -perm-550
```

将在系统上找出所有以字符串perl*开始并且可被用户执行的文件(-perm 550)。因此, 它可以找出Perl的可执行的候选文件。也可以在某些集中的位置查找Perl。Unix系统上常见的集中位置是/usr/bin, /usr/local/bin和/usr/sbin。如果通过这种方法找到了Perl, 并且假设它位于目录/usr/local/bin中, 为了获得Perl的版本可以使用以下的命令:

```
prompt% /usr/local/bin/perl -v
```

然后把路径变量加到类似下面的命令中:

```
PATH=$PATH:/usr/local/bin
```

这是用于kern shell或bourne shell的语法;该语法允许在整个过程中不必在它前面键入/usr/local/bin就能执行Perl。无论如何, 你的目的是要看到如图1-1所示的窗口。如果读者看到了这个窗口, 就说明做对了。

(2) 在NT/Windows 95上查找Perl

在Windows NT上查找Perl有几种方法。一种是从命令行上查找。打开DOS shell并且键入:

```
C:\>perl -v
```

系统将检查Perl是否安装在集中区域, 并且检查可执行的目录是否加到用户的路径设置上。如果Perl没有安装在常规位置也没有在路径中设置, 可以键入下面的命令:

```
C:\>\perl\bin\perl -v
```

注意存储你所使用的Perl的版本信息, 也即该命令的输出结果。

通过键入以下命令, 可以在整个磁盘驱动器上搜索Perl:

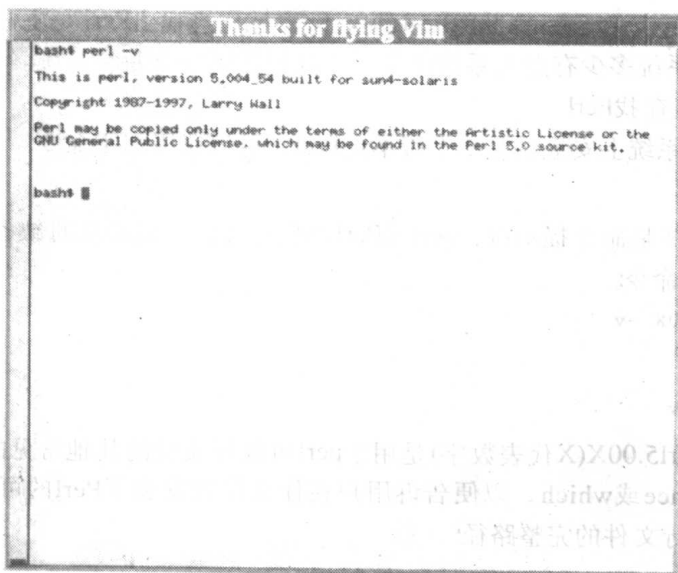


图1-1 在Unix上成功查出Perl版本后的输出结果

```
C:\>dir perl * /s
```

或可以进入Explorer，查找名为perl的文件。如果它不在标准位置，则记下该位置，并通过键入如下命令将其添加到路径中：

```
set PATH = " %PATH%" ; C:\path\to\perl\binary
```

此处，\path\to\perl\binary是Perl二进制文件的名字。（因此，运行Perl脚本时，不必每次都键入Perl的整个路径的名称。）在任何情况下，设置完PATH变量后，C:\>perl -v应当显示类似图1-2的界面。

C:\>perl -v命令指出了读者所使用的Perl的类型

(3) 在Macintosh上查找Perl

不像Windows和Unix，Macintosh具有GUI界面，该界面用于运行Perl。它被称为“MacPerl应用程序” Macintosh 也有一个Perl的命令行版本，该版本带有MPW(专业命令行Mac开发工具)，但最流行的方法是使用MacPerl的GUI。查找MacPerl应用程序，就像在Mac上查找其他任何文件那样。在系统7工具内只需选中find选项，并且查找字符串MacPerl。如果找到了MacPerl，你就会看到如图1-3所示的界面。在Perl世界中，该软件荣获了“最酷图标”的美誉。

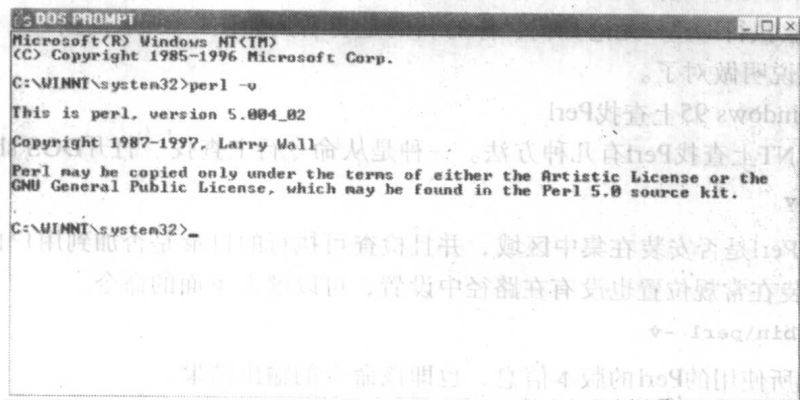


图1-2 在NT环境下成功查找到Perl后的结果

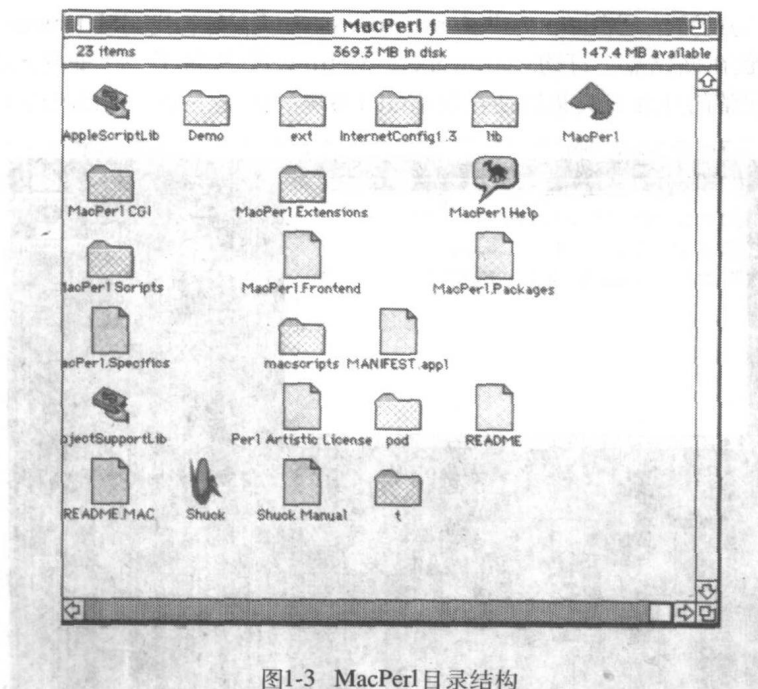


图1-3 MacPerl目录结构

为了找到读者想运行的Perl版本，双击MacPerl图标(有骆驼的那个图标)，并找到Script子菜单。选择one liner并且在图1-4所示的对话框中键入命令。然后，系统会给出图1-5所示的界面。

注意 Perl的Macintosh版本不同于此处讨论的其他版本，因为Macintosh版本目前只达到Perl5.002版本。

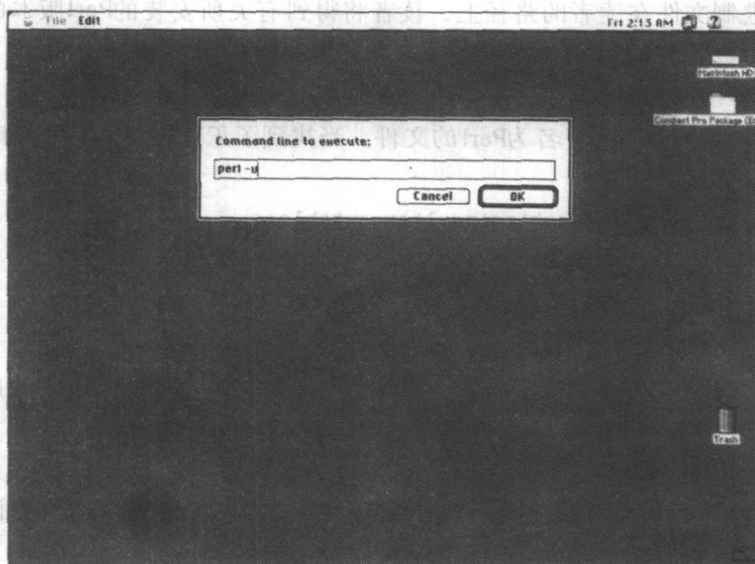


图1-4 获取Perl版本的命令

读者也注意到了，在此处可以找到一个不同的Perl版本，即Macintosh Programming Workshop——MPW版本。这是具有命令行环境的Perl。如果想使用该版本，就必须有MPW。

得到MPW的最容易的方法是在<http://www.metrowerks.com>站点上找到Metrowerks, 然后利用称为CodeWarrior的包在该站点上订购Metrowerks。Metrowerks提供了一个非常像Unix的命令环境, 并带有一套完整的开发工具集以便在没有GUI系统的情况下编写C(以及Perl)程序。

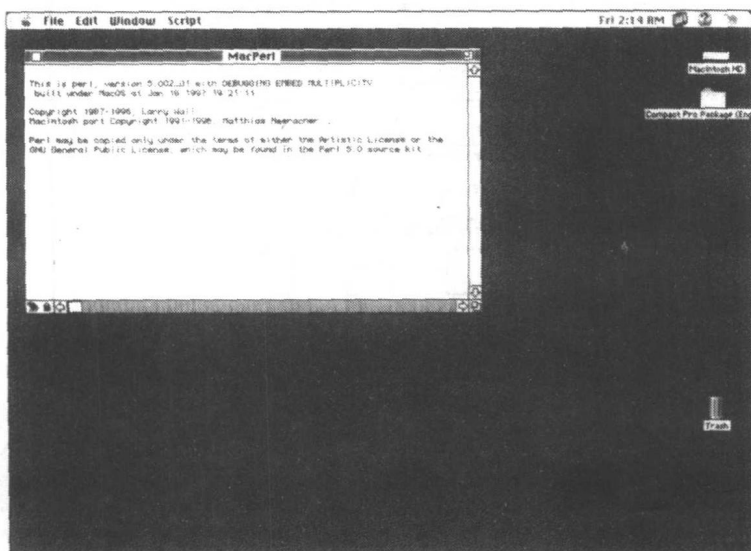


图1-5 Macintosh系统成功找到Perl后的显示结果

(4) 在OS/2上查找Perl

在OS/2上查找Perl与在Unix和Win32上查找Perl非常相似。所有读者需要做的是在'command prompts'文件夹下得到shell, 然后键入:

```
[H:] perl-v
```

如果Perl二进制文件在读者的路径上, 读者将得到有关所安装的Perl版本的信息。就像在Unix或NT系统环境中一样, 如果利用这种方法无法找到Perl, 可以键入find:

```
[H:] find "perl*"
```

该命令将会显示系统上所有名为Perl的文件。当找到了正确的二进制文件时(Perl.exe), 读者就可以键入以下命令:

```
[H:] set PATH="%PATH%";\path\to\perl\executable;
```

该命令把perl.exe设置到读者的系统环境中以便自动找到Perl。键入

```
[H:] perl-v
```

后就应当出现如图1-6所示的界面。

出现此界面说明读者的操作是正确的。查看一下读者是否具有正确的Perl版本号, 然后进行其他操作。

(5) 在VMS上查找Perl

最后, 我们在VMS 6.2及以上版本中查找Perl。如果处在DCL提示下(如果读者是一位Unix用户, VMS中的shell与Unix中的shell是相同的), 键入:

```
$ perl-v
```

就像其他任何的OS那样, 该命令向读者显示Perl的版本。如果想查看一下Perl在磁盘上什么位置, 可以键入以下命令: