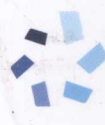


(美) M. Tim Jones 著 张元章 译

 CENGAGE
Learning™

GNU/LINUX环境编程 (第2版)

- 浅显易懂，容易理解的Linux编程基础
- 聚焦Linux编程利器——GNU工具和库
- 全面覆盖各种有用的API
- 配套资源包含本书代码和所有API

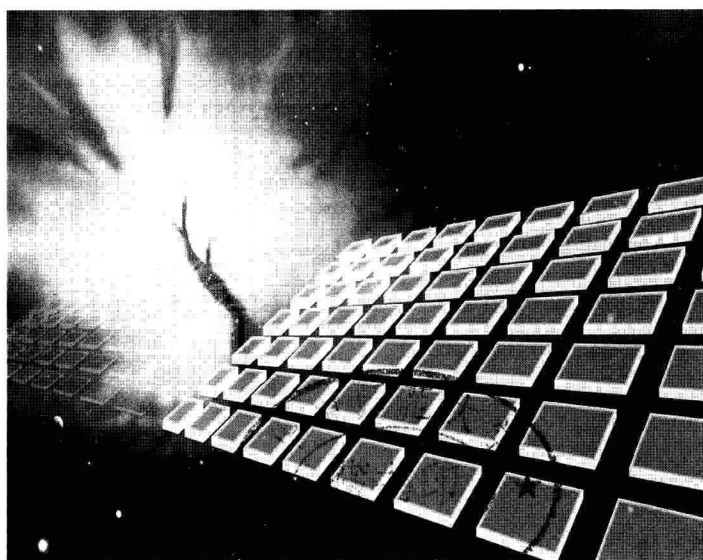


PROGRAMMING SERIES

 CENGAGE
Learning™

清华大学出版社

GNU/LINUX环境编程 (第2版)



内 容 简 介

本书针对 Linux 程序员, 本书详细介绍如何开发高性能、高安全性的应用程序。经过全面修订的第 2 版涵括所有必要的工具和编程技术, 提供丰富示例来说明 GNU/Linux API 的用法。全书共 5 部分 34 章, 主题涉及 GNU 工具, 应用程序开发, shell 与脚本编程, 调试和提高安全性, 虚拟化技术。凡此种种, 吸引着近万名程序员选择本书来学习 GNU/Linux 编程技术。

本书浅显易懂, 从全局介绍 Linux 编程基础, 重点强调 Linux 编程利器的使用, 对 Linux 程序员具有重要的参考和指导作用。

GNU/Linux Application Programming, Second Edition.

M. Tim Jones

Copyright © 2008 by Course Technology, part of Cengage Learning.

Original edition published by Cengage Learning. All Rights reserved. 本书原版由圣智学习出版公司出版。版权所有, 盗印必究。

Tsinghua University Press is authorized by Cengage Learning to publish and distribute exclusively this simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由圣智学习出版公司授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可, 不得以任何方式复制或发行本书的任何部分。

978-7-302-22065-7

Cengage Learning Asia Pte. Ltd.

5 Shenton Way, # 01-01 UIC Building, Singapore 068808

本书封面贴有 Cengage Learning 防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

北京市版权局著作权合同登记号 图字: 01-2009-6812

图书在版编目(CIP)数据

GNU/LINUX 环境编程(第 2 版)/(美)琼斯(Jones, M. T.)著; 张元章译.—北京: 清华大学出版社, 2010.3

书名原文: GNU/Linux Application Programming, Second Edition

ISBN 978-7-302-22065-7

I. G… II. ①琼… ②张… III. Linux 操作系统—程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字(2010)第 026222 号

责任编辑: 文开琪

责任校对: 周剑云

责任印制: 孟凡玉

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京密云胶印厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 34.25 字 数: 829 千字

版 次: 2010 年 3 月第 1 版 印 次: 2010 年 3 月第 1 次印刷

印 数: 1~4000

定 价: 69.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 034684-01



译者序：周虽旧邦，其命维新

张元章

生活在这块土地上的先民们大概不会想象到以后会有这样一个时代，在这个时代中几乎一切都在很快地发生着变化。但先民们确实感觉到，在这个宇宙中，唯一永恒的可能就是变化。

今天，我们都能很明确地体会到自己身处不断地高速度的变化之中。计算机应用程序开发尤其如此，开发工具甚至开发思想都在不断地变化。我们适应着这些变化的同时，也在促成、影响着这些变化。值得欣慰的是，变化的节奏还没有引起大家的不适，而几乎每一次变化都让人类变得更强。

本书展示了用于编写 Linux 应用程序的 GNU 工具和库。重点突出了 Linux 应用程序编程中一些新的变化，如 Ruby 和 Python 不容置疑地兴起；flex 和 bison 的成熟使得构造解析器不再是资深专家的特权；同时本书是帮助你开发 GNU/Linux 操作系统应用程序的利器，即使你原本对 Linux 并不熟悉。本书通过大量的示例演示了各种 API 函数，包括进程管理、共享内存、消息队列、旗语、POSIX、文件操作、套接字等，你可以通过这些示例快速学习 API 函数的使用，快速奠定 GNU/Linux 应用程序编程的基础。

本书在专有名词的使用上，特别地提高了“兼容性”：如“代码硬化”和“防御性编程”，“共享库”和“动态链接库”，这些专有名词在本书中确保不会导致歧义时反复交替使用，表达相同的含义，这是原书本来的特色，以此书入门的读者应该可以容易地适应其他书籍资料中不同的专有名词使用习惯。翻译中保留了这种特色，也交替使用了相同含义的不同中文名词，如“套件”和“软件包”。

最后，我想感谢原作者 M. Tim Jones，是他为我们提供了这样一本出色的教程，本书处处突出“快速”，快速地掌握新工具，快速地面对新变化；同时，也没有忽略对基础理论、基本方法的介绍。这里还要感谢翻译过程中涉及的所有人士，他们是文开琪、史海峰、刘淼、李克武、龙重。

衷心希望读者朋友们能通过本书，体验最新的 GNU/Linux 应用程序编程，加入激动人心的开源社区。

前 言

GNU/Linux 是操作系统中的“瑞士军刀”。从最小的设备(如苹果的 iPod)到最大的设备(如 IBM 的“蓝色基因”巨型机)中都可以看到它的踪影。你会发现 GNU/Linux 运行在很多不同的架构中,从老式的 x86 处理器到 PlayStation 3 使用的 cell 处理器,不一而足。

本书提供 GNU/Linux 操作系统上的应用程序开发的基础知识。无论你是为 iPod 还是 BlueGene 开发应用程序,所需要的编程概念与 API 均可以在本书中找到。

本书内容

本书全面介绍 GNU/Linux 环境下应用程序开发的所有知识。全书五部分,包含 GNU 工具、应用程序开发、shell 与脚本、调试与硬化及一些介绍性的主题,如虚拟化基础。

具体包含以下主题。

- GNU/Linux 架构与虚拟机制。
- GNU 工具,如 GCC、make、automake/autoconf、源代码控制系统、GNU Debugger 以及 GNUplot。
- 应用程序开发基础,如库(静态的和动态的)、文件管理、管道、套接字、编程等。
- GNU/Linux 进程模型(包括线程)和 POSIX IPC 机制(消息序列、旗语及共享内存)。
- shell 与脚本基础,从相应的 GNU/Linux 命令到 Bash、Ruby 以及 Python。sed 和 awk 文本管理,flex 和 bison 语法分析器生成。
- 本书还覆盖调试和硬化技术,包括软件测试工具、覆盖测试和利用 GCov 和 GProf 的剖析及内存调试工具(如 valgrind 等)。

本书配套资源

本书面向的读者

如果你想学习如何开发运行与 GNU/Linux 操作系统有关的应用程序,或者想扩充知识以进入更高深的开发领域,那么本书就是为你而写的,本书适合 GNU/Linux 初级和中级程序开

发人员阅读与参考。书中覆盖相关的工具、API 及开发技术，并通过大量实例来阐明如何使用 GNU/Linux 的 API。

随书资源

随书资源包含本书(GNU/Linux 应用程序编程)所有的应用程序示例。可从 www.wenyuan.com 找到本书后下载，也可通过发送电子邮件到 coo@netease.com 申请和获取。

资源目录

- Source: 本书中所有的示例代码，按章保存。
- Figures: 本书中所有的示意图，按章保存。

系统需求

- Linux，内核版本 2.6(随书资源经过了 Fedora 和 Ubuntu 的测试)
- 奔腾 I 处理器及以上
- 硬盘驱动器。
- 256MB 以上内存
- 为示例代码准备 1MB 硬盘空间

读者指南

本书是为 GNU/Linux 应用程序开发人员写作的。你会注意到本书没有“Linux 内核”或“设备驱动”这样的主题。本书特意如此，因为尽管那些主题本身也很迷人，但掌握那些内容对于开发 GNU/Linux 环境下的应用程序和工具而言却不是完全必需的。

本书分为五个部分，分别集中讨论 GNU/Linux 编程的不同方面。第 I 部分“导论”，向初学者介绍 GNU/Linux。这部分讲述 GNU/Linux 架构，简要介绍了进程模型和使用许可，简介了开源开发及其使用许可。还探究了 Linux 虚拟化，包括 Linux 中的模型和选项。

第 II 部分“GNU 工具”，集中于 GNU/Linux 编程需要的工具。这部分探究了标准的 GNU 编译工具链和 GNU make 自动 build 系统。研究了库的 build 和用法(静态的和动态的)。探究了使用 gcov 和 gprof 进行覆盖测试和剖析，使用 automake 和 autoconf 进行应用程序打包和发布。最后，回顾 Linux 与源码控制相关的常用选项和 Gnuplot 数据可视化。

在介绍 GNU/Linux 架构和必要的应用程序开发工具的基础上，接下来的第 III 部分“应用程序开发”集中关注 GNU/Linux 中最有用的服务。包括管道、套接字编程、文件管理、传统的进程和 POSIX 线程、消息队列、旗语 semaphores 和共享内存管理。

在第 IV 部分“GNU/Linux shell 和脚本”中，我们转而关注使用 shell 和脚本语言的应用程序开发。在 GNU/Linux 上编程需要用到的一些最有用的命令在这里介绍。第 IV 部分还提供一个关于 Bourne-Again shell(bash)的指南。探究了如何使用两种最流行的字符串处理语言(awk 和 sed)来进行文本处理。讨论使用 GNU Flex 和 Bison(lex 和 yacc-兼容分析程序生成器)进行语法分析。这部分还研究了 Ruby 和 Python 脚本。

第 V 部分“调试与测试”从多个不同的方在阐述了调试。我们将研究一些可能有助于自动回归的大型机单元测试。介绍 GNU Debugger 及最常用的命令和技术。最后，探究代码硬化专题及相关的多种调试工具和提升 GNU/Linux 应用程序可靠性和安全性的辅助开发工具。

本书各部分有其内在的线索，但各章也可以根据需要独立阅读。在需要其他章节信息的地方提供了注明。

致谢

1994 年夏天，我第一次接触到“开源”。那时的我刚刚完成一个大型同步通信卫星的操作系统内核开发项目，使用的是 MIL-STD-1750A 微处理器上的 Ada 语言。Ada 语言在技术上十分精确、安全，可读性好。即使以 20 世纪 90 年代早期的标准来看，MIL-STD-1750A 微处理器也是很陈旧的。(这是 20 世纪 70 年代为军用航空电子设备设计的指定集架构，但其简明性仍然是一流的。)

我转而从事伽玛射线暴研究卫星方面的工作，同时从事 1750GALS 项目的验证(validation)工作。Oliver Kellogg 主管的这个项目，包括了针对 1750A 系列的处理器的 Ada 语言的 GCC 编译器、汇编器、连接器和模拟器。我具备 Ada 和 1750A 的相关知识背景，而且伽玛射线暴研究项目当时也进入了试生产阶段，于是我便挪出一些时间进行验证工作。几个月后，我在 comp.compilers 新闻组看到一篇文章，摘录如下：

“1750GALS”，MIL-STD-1750 GCC/汇编器/连接器/模拟器，现在有了欧洲 FTP 站，及一个在美国的 FTP 镜像。

感谢 VTT 公司的 Reikka Ruuska(Pekka.Ruuska@vtt.fi)和 MIT 空间研究所的 M. Tim Jones(mtj@space.mit.edu)，他们的调试报告使这个工具包现在可用。此外，感谢 Tim Jones 架设了美国的 FTP 镜像站。

我无意中变得世界知名且因此而大出风头。当然，这种说法有点夸张，但我帮助那个项目

所花的时间的确既有趣又有价值,我由此进入自由软件(那时自由软件已有 10 年历史)和开放源码(在之后三年,这个术语才被创造出来)的世界。

本书第 2 版不仅仅是数月努力工作的成果,更是根植于 UNIX 和 GNU 工具开发人员数十年不懈的工作。为创建和改进这些工具做出贡献的不计其数的开发人员的名字足够编成一部专著,所以我在这里简单地向(我认为是)四位对 GNU/Linux 操作系统贡献最突出的人致谢。

AT&T 贝尔实验室的 Dennis Ritchie 和 Ken Thompson,他们创建了第一个 UNIX 操作系统(以及其后继变体)和 C 程序语言。

GNU 和自由软件基金的缔造者 Richard Stallman 激发并团结全世界的自由思想者,创建了世界水平的操作系统 GNU/Linux。

最后但绝不意味着贡献最小, Linus Torvalds 提供了 Linux 内核,而且一直是内核源码的把关者和重要贡献者。

特别感谢 Jim Lieb, 他丰富的 UNIX 相关知识及其对本书的详细审阅,使得本书得以很大改进。感谢 Cengage(特别是 Jen Blaney 和 Marta Justak), 他们使第 2 版的出版成为可能。

目 录

第 I 部分 导 论

第 1 章 GNU/Linux 的历史3	开放源码与自由软件..... 17
概述.....3	自由软件项目剖析..... 18
Unix 操作系统的历史.....3	开源证书.....18
AT&T UNIX..... 4	GPL..... 19
BSD..... 4	Qt 公共许可证..... 19
GNU/LINUX 的历史.....4	BSD 许可证..... 20
GNU 和自由软件基金会..... 5	证书小结..... 20
Linux 内核..... 5	开源开发的问题.....20
合作..... 6	可用性/可靠性斜线上升..... 20
Linux 的发行.....7	文档问题..... 21
小结.....7	自我..... 21
参考文献.....7	狂热..... 21
第 2 章 GNU/Linux 系统架构9	小结.....22
概述.....9	参考文献.....22
系统架构概要.....9	资源.....22
Linux 内核的架构.....10	第 4 章 Linux 虚拟化与仿真23
GNU 系统库(glibc).....11	概述.....23
系统调用接口.....11	什么是虚拟化?.....23
内核组件.....12	虚拟化简史.....24
硬件.....15	虚拟化的意义.....26
小结.....16	虚拟化的分类.....27
资源.....16	完全虚拟化.....27
第 3 章 自由软件开发17	准虚拟化.....27
概述.....17	模拟.....28
	操作系统的虚拟化..... 29

硬件辅助虚拟化	30
开源虚拟化解决方案	31
QEMU	31

KVM	34
小结	35

第 II 部分 GNU 工具

第 5 章 GNU 编译工具链	39
概述	39
编译简介	40
GCC 的格式(编译, 汇编和链接)	41
有用的选项	41
编译警告	42
GCC 优化器	44
-O0 优化	45
-O1 优化(-O)	45
-O2 优化	45
-Os 优化	46
-O3 优化	46
架构相关的优化	47
调试选项	48
其他工具	48
小结	49
第 6 章 应用 GNU make 构建软件	50
概述	50
示例项目	50
手工编译	51
buildit 脚本	51
简单的 Makefile 文件	52
Makefile 变量	54
模式匹配规则	57
自动依赖跟踪	58
小结	60
第 7 章 库的构建与使用	61
概述	61

什么是库?	61
生成静态库	62
共享库的生成	68
动态加载库	69
工具	73
file 工具	73
size 命令	73
nm 命令	73
objdump 工具	74
ranlib 工具	76
小结	76
动态库 API	76

第 8 章 用 automake/autoconf 打包	77
概述	77
一个简单的项目	78
Makefile 解决方案	78
自动工具的简单运用	79
automake 工具	81
autoconf 工具	82
configure 脚本	83
生成的 Makefile 文件	84
小结	85
第 9 章 GNU/Linux 的源码控制	86
概述	86
源码控制的定义	86
源码控制范例	87
存储库的架构	87
修订模型	88
有用的源码控制工具	89

CVS	89	用户界面	107
SVN	94	简单绘图	108
Git 源码控制系统	99	绘制文件中的数据	110
小结	104	3D 的函数绘制	111
参考文献	105	3D 等值线绘图	114
资源	105	隐线消除	114
第 10 章 应用 Gnuplot 进行		将图保存为文件	115
数据可视化	106	多图模式	116
概述	106	使用 Gnuplot 的工具	117
Gnuplot	106	小结	117
安装 Gnuplot	107	资源	118

第Ⅲ部分 应用程序开发主题

第 11 章 GNU/Linux 的文件操作	123	系统命令	150
概述	123	mkfifo 命令	150
GNU/Linux 的文件操作	123	小结	151
探究文件操作 API	124	管道编程 API	151
创建一个文件句柄	124	第 13 章 套接字编程简介	152
打开文件	124	概述	152
数据的读写	126	网络的分层模型	152
二进制数据的读写	133	套接字编程的范式	153
基础 API	138	主机	154
小结	140	协议	154
文件操作 API	141	端口	154
第 12 章 管道编程	142	地址	154
概述	142	套接字	155
管道模型	142	客户端/服务器模式	155
管道和命名管道	143	应用程序示例	156
旋风式简介	143	日期查询服务器	157
详细介绍	145	日期查询客户端	160
pipe 函数	145	套接字 API 小结	161
函数 dup 和 dup2	147	创建和清除套接字	161
函数 mkfifo	149	套接字地址	162
		套接字的原语	163

其他“杂项”函数.....	168	API 小结.....	206
其他传输协议.....	169	第 15 章 POSIX 线程(P 线程)编程.....	207
SCTP 的特点.....	169	概述.....	207
SCTP 的特点其他.....	171	什么是线程.....	208
多语言视角.....	171	线程函数基础.....	209
小结.....	173	P 线程 API.....	209
套接字编程 API.....	173	线程基础.....	210
参考文献.....	174	线程管理.....	211
资源.....	174	线程的同步.....	212
第 14 章 GNU/Linux 进程模型.....	175	线程互斥.....	214
概述.....	175	线程条件变量.....	218
GNU/Linux 进程.....	175	构建使用线程的应用程序.....	224
旋风式简介.....	176	小结.....	225
用 fork 创建一个子进程.....	177	参考资料.....	225
与创建者进程同步.....	179	API 小结.....	225
捕获信号.....	180	第 16 章 消息队列 IPC.....	227
发出信号.....	181	概述.....	227
传统的进程 API.....	184	消息队列简介.....	227
fork 函数.....	185	创建消息队列.....	228
wait 函数.....	186	配置一个消息队列.....	229
waitpid 函数.....	187	向一个消息队列中写入消息.....	230
signal 函数.....	188	从消息队列中读取消息.....	231
pause 函数.....	192	移除消息队列.....	232
kill 函数.....	192	消息队列 API.....	233
raise 函数.....	193	msgget 函数.....	233
exec 变体.....	194	msgctl 函数.....	236
alarm 函数.....	197	msgsnd 函数.....	240
exit 函数.....	198	msgrcv 函数.....	241
POSIX 信号.....	199	用户工具.....	243
系统命令.....	202	小结.....	245
ps 命令.....	202	消息队列 API 函数.....	245
top 命令.....	203	第 17 章 旗语同步.....	246
kill 命令.....	203	概述.....	246
小结.....	204	旗语理论.....	246
proc 文件系统.....	204	旗语的类型.....	247
资源.....	206		

GNU/Linux 旗语快速简介	248	小结	290
创建旗语	249	参考文献	290
获取和释放旗语	250	共享内存 API	291
配置旗语	253	第 19 章 高级文件操作	292
移除旗语	254	概述	292
旗语 API	255	测试文件类型	292
semget 函数	256	其他 stat 信息	294
semctl 函数	258	确定当前工作目录	295
semop 函数	264	列举目录	296
用户工具	266	使用 inotify 进行文件事件通知	299
小结	268	通知过程	299
旗语 API	268	从文件系统中移除文件	304
第 18 章 共享内存编程	269	传输数据	305
概述	269	小结	305
快速了解共享内存	270	高级文件操作 API	305
创建共享内存区段	270	第 20 章 其他应用程序开发主题	308
取得共享内存区段的信息	271	概述	308
共享内存区段的挂接和脱离	272	使用 getopt 和 getopt_long 解析	
使用共享内存区段	273	命令行选项	308
移除共享内存区段	275	时间 API	313
共享内存 API	276	用 sysinfo 收集系统信息	315
shmget 函数	276	使用 mmap 进行内存映射	317
shmctl 函数	279	锁定和解锁内存	320
shmat 函数	283	Linux 错误报告	322
shmdt 函数	284	小结	324
使用共享内存区段	285	API 小结	324
用户工具	289		

第IV部分 GNU/Linux 的 shell 与脚本

第 21 章 GNU/Linux 标准命令	331	GNU/Linux 基本命令	335
概述	331	小结	344
重定向	331	第 22 章 Bourne-Again shell(Bash)	345
标准输入/输出/错误	332	概述	345
环境变量	333	预备知识	345
脚本调用	334		

示例脚本.....	346	资源	376
bash 脚本.....	347	第 24 章 使用 awk 进行文本处理	377
变量	347	概述	377
条件结构.....	351	awk 简史	377
条件	351	awk 结构	377
case 结构.....	355	命令行 awk.....	378
循环结构.....	356	脚本 awk	381
while 循环.....	356	其他 awk 样式.....	385
for 循环.....	358	小结	385
输入与输出.....	359	有用的 awk 单行程序.....	386
函数	361	第 25 章 使用 flex 和 bison	
示例脚本.....	362	生成解析器	387
简单的目录档案管理脚本	363	概述	387
查找今天更新/创建的文件的		词法分析和语法处理	387
脚本程序.....	364	词法分析器和解析器的通信.....	389
其他脚本语言.....	366	flex 工具	390
小结	366	bison 工具	393
资源	367	一个简单的语法	393
第 23 章 使用 sed 进行编辑	368	在 bison 中编写语法.....	394
概述	368	连接语法解析器和词法分析器.....	396
剖析一个简单的脚本.....	369	构建一个简单的配置解析器.....	398
sed 空间(缓冲器).....	370	配置文件词法分析器	399
典型的 sed 命令行选项.....	370	全局图像	403
正则表达式	371	小结	405
操作范围.....	372	第 26 章 Ruby 脚本编程	406
基本的 sed 指令.....	372	概述	406
替换(s)	372	Ruby 简介	406
删除(d).....	373	为什么使用 Ruby	407
打印(p).....	373	和其他语言的比较	407
行的添加(a)、插入(i)以及改变(c).....	373	Ruby 快速示例.....	408
退出(q).....	374	语言元素	410
转换(y).....	374	类型和变量	410
行数(=).....	375	控制	411
保持样式空间(h)	375	重复	412
小结	375	Ruby 中的字符串操作.....	413
一些有用的 sed 单行程序.....	376		

关联数组.....	414
类与方法.....	415
高级功能.....	418
动态代码.....	418
异常处理.....	419
自省.....	421
其他功能.....	422
Ruby 作为嵌入式语言.....	422
小结.....	422
资源.....	422
第 27 章 Python 脚本编程.....	423
概述.....	423
Python 简介.....	423
为什么使用 Python.....	424
与其他语言的比较.....	424
Python 快速示例.....	425
语言元素.....	428
类型和变量.....	428
控制.....	430
循环.....	430
Python 中的字符串操作.....	432

关联数组.....	433
类与方法.....	434
高级功能.....	436
动态代码.....	437
函数式编程.....	437
异常处理.....	438
小结.....	440
资源.....	440

第 28 章 GNU/Linux 管理基础.....	441
概述.....	441
Linux 文件系统浏览.....	441
套件管理.....	442
Tar 球发布.....	442
高级套件工具.....	445
内核更新.....	448
获得最新版的内核.....	448
配置内核.....	449
构建内核.....	450
安装内核.....	450
配置启动引导程序(Bootloader).....	451
小结.....	451

第 V 部分 调试与测试

第 29 章 软件单元测试框架.....	455
概述.....	455
单元测试.....	455
单元测试框架.....	457
打造自己的框架.....	457
C 单元测试系统.....	462
嵌入单元测试.....	466
expect 工具.....	469
小结.....	470
资源.....	471
第 30 章 用 GDB 进行调试.....	472

概述.....	472
为 GDB 进行编译.....	472
使用 GDB.....	473
启动 GDB.....	475
查看代码.....	475
使用断点.....	476
逐步运行程序.....	478
检查数据.....	479
改变数据.....	479
检查堆栈.....	480
停止程序.....	480
其他 GDB 调试主题.....	480

多进程应用程序调试.....	480	gcov 可用的选项.....	502
多线程应用程序调试.....	481	注意事项.....	503
调试已有的进程.....	482	小结.....	504
事后分析调试.....	483	参考文献.....	504
小结.....	484	资源.....	504
资源.....	484	第 33 章 用 GNU gprof 进行	
第 31 章 代码硬化.....	485	性能分析.....	505
概述.....	485	概述.....	505
代码硬化技术.....	485	什么是性能分析.....	505
返回值.....	485	什么是 gprof.....	505
细察用户/网络的输入/输出.....	486	准备映像.....	506
使用安全字符串函数.....	486	使用 gprof 工具.....	508
缓冲区溢出.....	486	gprof 可用的选项.....	510
在决定点提供逻辑选择.....	487	注意事项.....	513
自识别结构体.....	488	小结.....	513
报告错误.....	490	参考文献.....	513
降低复杂度, 从而减少潜在的错误.....	491	第 34 章 高级调试主题.....	514
自保护的函数.....	491	概述.....	514
最大调试输出.....	492	内存调试.....	514
内存调试.....	492	Valgrind 工具.....	514
编译器的支持.....	492	Electric Fence.....	517
源码检查工具.....	493	yamd 工具.....	517
代码跟踪.....	493	mtrace 工具.....	520
小结.....	495	交叉引用工具.....	521
资源.....	495	Cscope 工具.....	521
第 32 章 用 GNU gcov 进行覆盖测试.....	496	其他交叉引用工具.....	522
概述.....	496	用 ltrace 跟踪系统调用.....	523
什么是 gcov.....	496	动态挂接 GDB.....	525
准备映像.....	496	小结.....	527
使用 gcov 工具.....	498	资源.....	527
查看分支概率.....	499	附录 缩写与部分缩写词.....	528
不完全覆盖.....	501		



第 I 部分 导 论

- 第 1 章 GNU/Linux 的历史
- 第 2 章 GNU/Linux 系统架构
- 第 3 章 自由软件开发
- 第 4 章 Linux 虚拟化与仿真

本书的第 I 部分探究 GNU/Linux 操作系统及其开发范例的几个相关的介绍性主题：UNIX，GNU 及 GNU/Linux 操作系统简史；GNU/Linux 系统架构概述；对自由软件(以及开源问题)开发模式的讨论；简单概述一下 Linux 虚拟与仿真解决方案。

第 1 章 GNU/Linux 的历史

事实上，GNU/Linux 的历史发端于 1969 年第一个 UNIX 操作系统的开发。本章论述 UNIX 的开发历史以及促成 GNU/Linux 操作系统发布的几个重要开发人员所做的积极努力(和所遭受的挫折)。