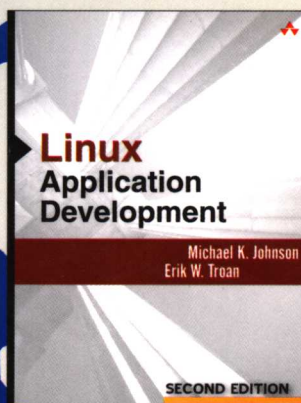


Linux Application Development

Second Edition

[美] Michael K. Johnson Erik W. Troan 著

Linux 应用程序开发 (第2版) (英文版)



典藏原版书苑

Linux 应用程序开发

(第2版) (英文版)

图书在版编目 (CIP) 数据

Linux 应用程序开发: 第2版/(美)约翰逊 (Johnson, M. K.),
(美)特罗安 (Troan, E. W.) 著. —北京: 人民邮电出版社, 2006.7
(典藏原版书苑)

ISBN 7-115-14941-0

I. L... II. ①约... ②特... III. Linux 操作系统—程序设计—英文 IV. TP316.89
中国版本图书馆 CIP 数据核字 (2006) 第 069860 号

版 权 声 明

Original edition, entitled LINUX APPLICATION DEVELOPMENT, 2nd Edition, 0321219147 by JOHNSON, MICHAEL K.; TROAN, ERIK W., published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright © 2005 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2006.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in People's Republic of China excluding Hong Kong, Macau and Taiwan.

仅限于中华人民共和国境内 (不包括中国香港、澳门特别行政区和中国台湾地区) 销售。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

典藏原版书苑

Linux 应用程序开发 (第2版) (英文版)

-
- ◆ 著 [美] Michael K. Johnson Erik W. Troan
责任编辑 李 际
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
印张: 45.25
字数: 1 022 千字 2006年7月第1版
印数: 1—2 500 册 2006年7月北京第1次印刷

著作权合同登记号 图字: 01-2006-4172 号

ISBN 7-115-14941-0/TP · 5517

定价: 75.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

内容提要

本书介绍了在 Linux 环境下开发应用程序需要掌握的知识,对编程实践中经常遇到的问题进行了详细的讲解,并提供了典型实例加以说明。

全书共分为 4 个部分。第一部分介绍初学者需要了解的知识,包括 Linux 的历史、从事自由软件开发需要了解的版权知识和 GPL 等许可证,以及如何获取和使用在线文档系统。第二部分介绍了开发环境和工具,其中包括 Emacs 编辑器、vi 编辑器及 gdb 调试器的使用, gcc 的选项和扩展, GNU C 库的基本知识,内存溢出和泄漏的调试工具,如何创建、使用、管理静态库和动态库,以及如何通过系统调用请求系统服务。第三部分详细介绍了 Linux 系统编程知识,其中包括进程模型、简单文件管理、信号处理、高级文件操作、目录操作、作业控制、终端与伪终端、socket 网络、时间和定时器、随机数、虚拟控制台、Linux 控制台以及如何编写安全的程序。第四部分精选了 Linux 编程中最常用的开发库进行介绍,其中包括字符串匹配、用于终端编程的 S-Lang 库、基于散列的数据库函数库、命令行解析库,如何运行时动态加载共享对象,以及有关用户身份识别和验证的库。

本书内容丰富、实用,适合需要开发 Linux 软件,或者需要把软件从其他平台移植到 Linux 上的程序员阅读。

PREFACE

前 言

本书是为那些需要开发 Linux 软件，或者需要把软件从其他平台移植到 Linux 上的有经验（或者并不是很有经验，但渴望学习）的程序员编写的。我们希望读者在学习 Linux 编程时拥有这本书，并且一直把它放在桌上作为参考。当我们刚刚完成第一版的前 3 章时，我们就已经把草稿作为工作时的参考资料。

第 2 版删掉了上一版中陈旧的内容，增加了新的内容。并且，读者可以通过 <http://ladweb.net/> 浏览和搜索本书的相关内容，从而使本书起到更大的作用。

Linux 与 Unix 的设计相似。本书提供了 Unix 编程基础和风格的背景知识。Linux 与 Unix 没有根本的不同，只是在一些细节上存在差别，这些差别一般来说并不比 Unix 一个版本和另一个版本的差别更大。本书更像是一部从 Linux 的角度出发、具有特定 Linux 信息的 Unix 编程指南。

Linux 也有其独特的扩展，例如它直接访问屏幕的能力（见第 21 章），它也提供一些比其他系统上更常用的特性，例如 `popt` 库（见第 26 章），本书覆盖了许多这些扩展和特性，这样你就可以在编写程序的时候真正充分利用 Linux 的优势。

- 如果你是一位 C 程序员，但是你既不了解 Unix 也不了解 Linux，那么请逐页阅读本书，并且尝试所有的例子，这可使你顺利走上出色程序员的成长之路。在系统相关的文档的帮助下，你可以很容易地过渡到任何一个 Unix 版本上。
- 如果你已经是一位专业的 Unix 程序员，本书将帮助你轻松转向 Linux。我们会尽最大的努力让你很容易找到所需要的精确信息。同时我们仔细而清楚地讲述了那些有经验的程序员有时候也觉得很棘手的话题。例如进程和会话组、任务控制以及 `tty` 处理。
- 如果你已经是一位 Linux 程序员，本书对一些容易混淆的问题的清晰讲解将使你的许多编程任务变得更加容易完成。本书几乎每一章的内容对你来说都是独立的，因为你已经掌握了有关 Linux 的必要基础知识。不管你的编程经验多么丰富，你都会发现本书所提供的资料值得你经常翻阅。

本书与传统的 Unix 编程书籍不同，它忠实地针对一个特定的操作系统。我们不试图去涵盖各种类 Unix 系统的所有差异，因为这对 Linux 程序员、Unix 程序员或者对 Linux 和 Unix 都不熟悉的 C 程序员没有太大的帮助。我们自身的经验说明，只要能在任何一个 Unix 类的系统上学会编程，在其他系统中学习编程将会非常容易。

本书并没有覆盖 Linux 编程的所有细节，没有解释 ANSI/ISO C 所描述的基本接口（因为其他书籍已经做了很好的解释），也没有讲解 Linux 其他可用编程语言的价值以及在各个系统中都相同的图形编程库。相反，我们将为你列出这些领域相关的书籍。简而言之，我们只提供你从一个其他系统，例如 Windows、Macintosh 甚至 DOS 上的 C 程序转为 Linux 程序员所需要的知识。

本书分为 4 个部分：

- 第一部分介绍了 Linux 的基础知识——操作系统、许可证条款以及在线系统文档。

- 第二部分介绍了开发环境最重要的几个方面——编译器、链接器、装载器以及其他系统中用得不多的调试工具。

- 第三部分是本书的核心部分，讲述了应用程序与内核的接口以及与作为内核主要接口的系统库的接口。在这一部分，只有第 19、20 和 21 章是专门针对 Linux 的，多数部分则从 Linux 的角度覆盖了 Unix 编程的一般知识。第 22 章在第 2 版中以新的一章出现，讲述了编写安全程序的基础知识。

- 第四部分描述了一些重要的库，这些库提供了独立于内核的接口，这些库确切地说不是 Linux 所特有的，但是它们中的一些在 Linux 系统中远比在其他系统用得更多。

如果你已经对 Linux 或者 Unix 编程非常熟悉，你可以以任意顺序阅读书中的各章，忽略不感兴趣的部分。如果你对 Linux 和 Unix 都不熟悉，尽管大部分章节都是相互独立的，但建议你首先阅读第 1、2、4、5、9、10、11 和 14 章，这些章节为你提供了阅读其他章节所需的大部分知识。特别是第 10、11 和 14 章，它们组成了 Linux 与 Unix 编程模型的核心。

下面的书籍尽管有或多或少的重复，但作为本书较好的补充，提供了较简单的，较高级或相关的话题。

- *The C Programming Language, second edition* [Kernighan, 1988]。该书简要介绍了 ANSI 标准 C 编程，但是缺乏操作系统的相关参考。它要求读者具备一定的编程知识。或者具备“可以向一个具有丰富编程知识的同事请教”的条件。

- *Practical C Programming* [Owalline, 1993]。该书以按部就班、循序渐进的方式讲解 C 编程的技术和风格，它适合没有编程经验的人阅读。

- *Programming with GNU Software* [Loukides, 1997]。该书介绍了 GNU 编程环境，包含一些介绍 C 编译器、调试工具、make 实用程序以及 RCS 源代码控制系统的章节。

- *Advanced Programming in the UNIX Environment* [Stevens, 1992]。该书覆盖了大多数重要的 Unix 和类 Unix 系统，它们都早于 Linux。它包含与本书的最后两部分相近的内容：系统调用和共享库，它同时提供了许多例子，并且解释了各种 Unix 版本之间的差异。

- *UNIX Network Programming* [Stevens, 2004]。该书详细讲解了网络编程技术，包括在 Linux 上已经不适用（至少在我们写这本书时已经不适用）的陈旧组网类型。在阅读这本书时，应用严格遵守 Berkeley 套接字接口（见第 17 章），以此保持最大的兼容性。如果你需要把 Linux 下的网络程序经过轻微改动后移植到某种 Unix 系统上，这本书也许会非常有用。

- *A Practical Guide to Red Hat Linux 8* [Sobell, 2002]。该书有 1500 多页，包括对使用 Linux、Shell 编程和系统管理的介绍。这本书是关于 Red Hat Linux 的著作，因此它的大多数信息适用于各种 Linux。它还提供了一个对 Linux 系统中许多实用程序的总结性参考。

- *Linux in a Nutshell* [Siever, 2003]。该书内容简短，从 O'Reilly 早期的 nutshell 参考中演绎而来，集中介绍了一个总结性的实用程序参考。

- *Linux Device Drivers, second edition* [Rubini, 1998]。该书为那些从未接触过操作系统代码的人，以及需要编写 Linux 驱动程序的人而作。

读者可以参见第 679 页的参考文献，获得相关主题的更多信息。

本书所有的源代码都来自我们的工作实例，在撰写本书时经过了测试。它们的电子版可以在 <http://labweb.net/> 下载。需要说明的是，对一些简短的代码块我们并没有检查它所有可能的错误，仅仅检查了它在说明系统如何工作方面可能的错误。尽管如此，对于所有出现在本书以及我们网站上的程序，我们尽可能（因为不可能做到完美）对所有可能出现的错误进行了检查。

本书将教你应该使用哪些函数，以及如何将它们组合在一起，我们也鼓励你去学习如何使用参考文档，这些文档大多数都已经包含在了你的系统中。第 3 章将讨论如何在 Linux 系统中寻找在线信息。

Linux 是一个快速发展的操作系统，在阅读本书时，一些情况无疑已经发生了变化（尽管我们希望变化不大）。我们写这本书时所参考的系统是 2.6 版本的 Linux 内核和 2.3 版本的 GNU C 库。

在读者的帮助下，我们将通过 <http://ladweb.net/errata.html> 维护一个勘误和修改列表。

欢迎你把意见发送到 lad-comments@ladweb.net。尽管不能保证逐个回复，但是我们将

认真阅读你的意见。

第 2 版

第 2 版对第 1 版的补充和修改包括：

- 考虑到新的“Single Unix Specification”第 6 款，POSIX 标准的升级版本，全书进行了更新。

- 为了方便找到示例程序，增加了示例源代码表格。

- 为了方便在整个源代码例子中定位，在列出短示例代码片断时加上了行号。

- 第 1 章更新和扩展了 Linux 开发的历史。

- 第 4 章讨论了 `strace` 和 `ltrace` 实用程序。

- 第 6 章作为新出现的一章，讨论了 GNU C 库（`glibc`）以及其所遵守的标准。解释了如何（以及为何）使用测试宏特性。第 6 章还讲述了由系统调用使用的一些基本类型，如何在运行时发现系统的能力。本章还覆盖了 `glibc` 所提供的一些其他系统接口，并且叙述了 `glibc` 所采用的向后兼容的方法。

- 第 7 章对内存调试工具增加了大量的扩展信息，包括 GNU C 库中的新的内存调试手段，新版本的 `mpx` 和新的 `Valgrind` 工具。

- 第 12 章讨论了实施信号和信号上下文。

- 第 13 章讲述了 `poll()` 和 `epoll` 系统调用，作为实现 `select()` 的另外一种推荐方法。

- 第 16 章讨论并且推荐了一种分配虚拟终端（Pseudo TTY）的新机制，讲述了对系统数据文件 `utmp` 和 `wtmp` 的修改。

- 第 17 章在 IPv4 之外讲述了 IPv6，包括既可以用于编写 IPv6 程序也可以用于编写 IPv4 程序的新系统库接口。为了帮助读者维护使用旧接口的程序，或者把旧代码移植到新接口，本章还解释了第 1 版中的相关旧接口。本章还讨论了许多网络服务程序所需的功能，例如非阻塞的 `accept()`。

- 第 22 章作为新的一章，讨论了编写安全程序的基本需求，并且解释了为什么有关安全的考虑不仅适用于系统守护进程和系统常用程序，而且适用于所有程序。

- 第 23 章对使用正则表达式做了进一步讨论，包括一个实现 `grep` 程序简单版本的示例。

- 第 26 章是原篇幅的一倍，讲解了对 `popl` 库的最新改进，并且提供了更好的示例源代码。

- 第 28 章对 Linux-PAM 的实现进行了讲解。
- 第 25 章以 qdbm 库内容代替了原来的 Berkeley db，因为 qdbm 的许可证限制更小。
- 索引和词汇表在范围和质量上得到极大提高。尤其是，索引中的专用词以粗体标记，使之在阅读时更明显。

几乎每一章都有重要的更新。

删减的内容包括：

- 如何通过邮件列表、Web 站点以及新闻组获得 Linux 常用信息，作为一本书中的信息应该在许多年内有用，而某些信息变化太快，因此不适宜放在书中。
- 对 I/O 端口操作的讨论，这部分内容因为与 Linux 设备和电源管理结构冲突，已经过时。
- GNU 通用公共许可证和 GNU 库通用公共许可证。因为这两份许可证一直很重要，它们的重复印刷不会增加公众对它们重要性的意识。此外，自从本书发行第 1 版之后，其他几类许可证也正在变得越来越重要。
- 调试内存用的 Checker 工具已经不再被继续维护，因此第 2 版不再对它进行讨论。

致谢

感谢每一位技术审稿人员，他们付出了时间和细致的思考。他们的建议使本书更加完善。特别要感谢 Linus Torvalds、Alan Cox、Ted Ts'o 以及 Arjan van de Ven，感谢他们对我们的问题的解答。

在支持我们写完第 1 版后，我们的妻子，Kim Johnson 和 Brigid Troan，是如此勇敢和慷慨。她们鼓励我们去写第 2 版。没有她们的帮助，除了单独更新之外就不会有这本书的出现。

Contents

Part 1	Getting Started	1
Chapter 1	History of Linux Development	3
	1.1 A Short History of Free Unix Software	4
	1.2 Development of Linux	6
	1.3 Notional Lineage of Unix Systems	7
	1.4 Linux Lineage	9
Chapter 2	Licenses and Copyright	11
	2.1 Copyright	11
	2.2 Licensing	13
	2.3 Free Software Licenses	14
	2.3.1 The GNU General Public License	15
	2.3.2 The GNU Library General Public License	15
	2.3.3 MIT/X/BSD-Style Licenses	16
	2.3.4 Old BSD-Style Licenses	16
	2.3.5 Artistic License	16
	2.3.6 License Incompatibilities	17
Chapter 3	Online System Documentation	19
	3.1 The man Pages	19
	3.2 The Info Pages	20
	3.3 Other Documentation	21
Part 2	Development Tools and Environment	23
Chapter 4	Development Tools	25
	4.1 Editors	26
	4.1.1 Emacs	27

4.1.2	vi	28
4.2	Make	29
4.2.1	Complex Command Lines	32
4.2.2	Variables	33
4.2.3	Suffix Rules	34
4.3	The GNU Debugger	35
4.4	Tracing Program Actions	40
Chapter 5	gcc Options and Extensions	43
5.1	gcc Options	44
5.2	Header Files	47
5.2.1	long long	47
5.2.2	Inline Functions	47
5.2.3	Alternative Extended Keywords	47
5.2.4	Attributes	48
Chapter 6	The GNU C Library	49
6.1	Feature Selection	49
6.2	POSIX Interfaces	52
6.2.1	POSIX Required Types	52
6.2.2	Discovering Run-Time Capabilities	54
6.2.3	Finding and Setting Basic System Information	55
6.3	Compatibility	57
Chapter 7	Memory Debugging Tools	59
7.1	Buggy Code	59
7.2	Memory-Checking Tools Included in glibc	62
7.2.1	Finding Memory Heap Corruption	62
7.2.2	Using <code>mtrace()</code> to Track Allocations	65
7.3	Finding Memory Leaks with <code>mptr</code>	66
7.4	Investigating Memory Errors with Valgrind	69
7.5	Electric Fence	74
7.5.1	Using Electric Fence	75
7.5.2	Memory Alignment	76
7.5.3	Other Features	77
7.5.4	Limitations	78
7.5.5	Resource Consumption	78
Chapter 8	Creating and Using Libraries	79
8.1	Static Libraries	79
8.2	Shared Libraries	80

8.3	Designing Shared Libraries	81
8.3.1	Managing Compatibility	82
8.3.2	Incompatible Libraries	83
8.3.3	Designing Compatible Libraries	83
8.4	Building Shared Libraries	84
8.5	Installing Shared Libraries	85
8.5.1	Example	86
8.6	Using Shared Libraries	88
8.6.1	Using Noninstalled Libraries	88
8.6.2	Preloading Libraries	89
Chapter 9	Linux System Environment	91
9.1	The Process Environment	91
9.2	Understanding System Calls	92
9.2.1	System Call Limitations	93
9.2.2	System Call Return Codes	94
9.2.3	Using System Calls	96
9.2.4	Common Error Return Codes	97
9.3	Finding Header and Library Files	102
Part 3	System Programming	103
Chapter 10	The Process Model	105
10.1	Defining a Process	105
10.1.1	Complicating Things with Threads	106
10.1.2	The Linux Approach	106
10.2	Process Attributes	107
10.2.1	The pid and Parentage	107
10.2.2	Credentials	108
10.2.3	The filesystem uid	113
10.2.4	User and Group ID Summary	113
10.3	Process Information	115
10.3.1	Program Arguments	115
10.3.2	Resource Usage	118
10.3.3	Establishing Usage Limits	120
10.4	Process Primitives	121
10.4.1	Having Children	122
10.4.2	Watching Your Children Die	123
10.4.3	Running New Programs	125
10.4.4	Faster Process Creation with <code>vfork()</code>	127

10.4.5 Killing Yourself	128
10.4.6 Killing Others	129
10.4.7 Dumping Core	130
10.5 Simple Children	131
10.5.1 Running and Waiting with <code>system()</code>	131
10.5.2 Reading or Writing from a Process	132
10.6 Sessions and Process Groups	134
10.6.1 Sessions	135
10.6.2 Controlling Terminal	136
10.6.3 Process Groups	136
10.6.4 Orphaned Process Groups	138
10.7 Introduction to <code>ladsh</code>	139
10.7.1 Running External Programs with <code>ladsh</code>	140
10.8 Creating Clones	153
 Chapter 11 Simple File Handling	 155
11.1 The File Mode	158
11.1.1 File Access Permissions	159
11.1.2 File Permission Modifiers	161
11.1.3 File Types	162
11.1.4 The Process's <code>umask</code>	163
11.2 Basic File Operations	165
11.2.1 File Descriptors	165
11.2.2 Closing Files	166
11.2.3 Opening Files in the File System	166
11.2.4 Reading, Writing, and Moving Around	168
11.2.5 Partial Reads and Writes	173
11.2.6 Shortening Files	175
11.2.7 Synchronizing Files	175
11.2.8 Other Operations	176
11.3 Querying and Changing Inode Information	177
11.3.1 Finding Inode Information	177
11.3.2 A Simple Example of <code>stat()</code>	178
11.3.3 Easily Determining Access Rights	181
11.3.4 Changing a File's Access Permissions	182
11.3.5 Changing a File's Owner and Group	182
11.3.6 Changing a File's Timestamps	183
11.3.7 Ext3 Extended Attributes	184
11.4 Manipulating Directory Entries	188
11.4.1 Creating Device and Named Pipe Entries	189
11.4.2 Creating Hard Links	191

11.4.3 Using Symbolic Links	192
11.4.4 Removing Files	194
11.4.5 Renaming Files	194
11.5 Manipulating File Descriptors	195
11.5.1 Changing the Access Mode for an Open File	195
11.5.2 Modifying the close-on-exec Flag	196
11.5.3 Duplicating File Descriptors	196
11.6 Creating Unnamed Pipes	198
11.7 Adding Redirection to <code>ladsh</code>	198
11.7.1 The Data Structures	199
11.7.2 Changing the Code	200
 Chapter 12 Signal Processing	 203
12.1 Signal Concepts	204
12.1.1 Life Cycle of a Signal	204
12.1.2 Simple Signals	205
12.1.3 Reliable Signals	207
12.1.4 Signals and System Calls	208
12.2 The Linux (and POSIX) Signal API	209
12.2.1 Sending Signals	209
12.2.2 Using <code>sigset_t</code>	210
12.2.3 Catching Signals	211
12.2.4 Manipulating a Process's Signal Mask	213
12.2.5 Finding the Set of Pending Signals	215
12.2.6 Waiting for Signals	216
12.3 Available Signals	217
12.3.1 Describing Signals	221
12.4 Writing Signal Handlers	222
12.5 Reopening Log Files	224
12.6 Real-Time Signals	227
12.6.1 Signal Queueing and Ordering	228
12.7 Learning About a Signal	231
12.7.1 Getting a Signal's Context	231
12.7.2 Sending Data with a Signal	237
 Chapter 13 Advanced File Handling	 241
13.1 Input and Output Multiplexing	241
13.1.1 Nonblocking I/O	244
13.1.2 Multiplexing with <code>poll()</code>	245
13.1.3 Multiplexing with <code>select()</code>	249
13.1.4 Comparing <code>poll()</code> and <code>select()</code>	253

13.1.5 Multiplexing with <code>epoll</code>	256
13.1.6 Comparing <code>poll()</code> and <code>epoll</code>	263
13.2 Memory Mapping	266
13.2.1 Page Alignment	267
13.2.2 Establishing Memory Mappings	268
13.2.3 Unmapping Regions	273
13.2.4 Syncing Memory Regions to Disk	274
13.2.5 Locking Memory Regions	275
13.3 File Locking	276
13.3.1 Lock Files	276
13.3.2 Record Locking	279
13.3.3 Mandatory Locks	285
13.3.4 Leasing a File	285
13.4 Alternatives to <code>read()</code> and <code>write()</code>	289
13.4.1 Scatter/Gather Reads and Writes	290
13.4.2 Ignoring the File Pointer	291

Chapter 14 Directory Operations **293**

14.1 The Current Working Directory	293
14.1.1 Finding the Current Working Directory	293
14.1.2 The <code>.</code> and <code>..</code> Special Files	295
14.1.3 Changing the Current Directory	295
14.2 Changing the Root Directory	296
14.3 Creating and Removing Directories	297
14.3.1 Creating New Directories	297
14.3.2 Removing Directories	297
14.4 Reading a Directory's Contents	297
14.4.1 Starting Over	299
14.5 File Name Globbing	300
14.5.1 Use a Subprocess	300
14.5.2 Internal Globbing	301
14.6 Adding Directories and Globbing to <code>ls</code>	306
14.6.1 Adding <code>cd</code> and <code>pwd</code>	306
14.6.2 Adding File Name Globbing	307
14.7 Walking File System Trees	311
14.7.1 Using <code>ftw()</code>	311
14.7.2 File Tree Walks with <code>nftw()</code>	313
14.7.3 Implementing <code>find</code>	315
14.8 Directory Change Notification	317

Chapter 15	Job Control	325
15.1	Job Control Basics	325
15.1.1	Restarting Processes	325
15.1.2	Stopping Processes	326
15.1.3	Handling Job Control Signals	327
15.2	Job Control in <code>ladsh</code>	328
Chapter 16	Terminals and Pseudo Terminals	335
16.1	<code>tty</code> Operations	336
16.1.1	Terminal Utility Functions	337
16.1.2	Controlling Terminals	338
16.1.3	Terminal Ownership	339
16.1.4	Recording with <code>utempter</code>	340
16.1.5	Recording by Hand	341
16.2	<code>termios</code> Overview	351
16.3	<code>termios</code> Examples	353
16.3.1	Passwords	353
16.3.2	Serial Communications	355
16.4	<code>termios</code> Debugging	370
16.5	<code>termios</code> Reference	371
16.5.1	Functions	372
16.5.2	Window Sizes	376
16.5.3	Flags	377
16.5.4	Input Flags	378
16.5.5	Output Flags	380
16.5.6	Control Flags	381
16.5.7	Control Characters	383
16.5.8	Local Flags	385
16.5.9	Controlling <code>read()</code>	387
16.6	Pseudo <code>ttys</code>	389
16.6.1	Opening Pseudo <code>ttys</code>	389
16.6.2	Opening Pseudo <code>ttys</code> the Easy Ways	391
16.6.3	Opening Pseudo <code>ttys</code> the Hard Ways	392
16.6.4	Pseudo <code>tty</code> Examples	396
Chapter 17	Networking with Sockets	407
17.1	Protocol Support	407
17.1.1	Nice Networking	408
17.1.2	Real Networking	408
17.1.3	Making Reality Play Nice	409
17.1.4	Addresses	410

17.2 Utility Functions	411
17.3 Basic Socket Operations	412
17.3.1 Creating a Socket	412
17.3.2 Establishing Connections	414
17.3.3 Binding an Address to a Socket	414
17.3.4 Waiting for Connections	415
17.3.5 Connecting to a Server	416
17.3.6 Finding Connection Addresses	417
17.4 Unix Domain Sockets	418
17.4.1 Unix Domain Addresses	418
17.4.2 Waiting for a Connection	419
17.4.3 Connecting to a Server	422
17.4.4 Running the Unix Domain Examples	423
17.4.5 Unnamed Unix Domain Sockets	423
17.4.6 Passing File Descriptors	424
17.5 Networking Machines with TCP/IP	429
17.5.1 Byte Ordering	430
17.5.2 IPv4 Addressing	431
17.5.3 IPv6 Addressing	433
17.5.4 Manipulating IP Addresses	435
17.5.5 Turning Names into Addresses	437
17.5.6 Turning Addresses into Names	449
17.5.7 Listening for TCP Connections	455
17.5.8 TCP Client Applications	457
17.6 Using UDP Datagrams	459
17.6.1 Creating a UDP Socket	460
17.6.2 Sending and Receiving Datagrams	461
17.6.3 A Simple tftp Server	463
17.7 Socket Errors	469
17.8 Legacy Networking Functions	471
17.8.1 IPv4 Address Manipulation	471
17.8.2 Hostname Resolution	473
17.8.3 Legacy Host Information Lookup Example	475
17.8.4 Looking Up Port Numbers	476
Chapter 18 Time	481
18.1 Telling Time and Dates	481
18.1.1 Representing Time	481
18.1.2 Converting, Formatting, and Parsing Times	484
18.1.3 The Limits of Time	489
18.2 Using Timers	490