

经 典 原 版 书 库

UNIX 环境高级编程

(英文版)

Advanced Programming in the UNIX[®] Environment

W. Richard Stevens



ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

(美) W. Richard Stevens 著



机械工业出版社
China Machine Press



Addison-Wesley

经典原版书库

(英文版)

UNIX 环境高级编程

Advanced Programming
in the UNIX Environment

江苏工业学院图书馆
藏书章

(美) W. Richard Stevens 著

English reprint edition copyright © 2002 by PEARSON
EDUCATION NORTH ASIA LIMITED and CHINA MACHINE
PRESS.

Advanced Programming in the UNIX[®] Environment by W. Richard
Stevens, Copyright © 1993.

All rights reserved.

Published by arrangement with Pearson Education, Inc.

本书英文影印版由美国Addison Wesley公司授权机械工业出版社在
中国大陆境内独家出版发行，未经出版者许可，不得以任何方式
抄袭、复制或节录本书中的任何部分。

版权所有，侵权必究。

本书版权登记号：图字：01-2001-3769

图书在版编目（CIP）数据

UNIX环境高级编程（英文版）/（美）史蒂文斯（Stevens, W.
R.）著. -北京：机械工业出版社，2002.1

（经典原版书库）

ISBN 7-111-09508-1

I. U… II. 史… III. UNIX操作系统-程序设计-英文 IV.
TP316.81

中国版本图书馆CIP数据核字（2001）第081805号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：华章

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2002年1月第1版·2002年3月第2次印刷

850mm×1168mm 1/32·24.125印张

印数：2 001-4 000册

定价：39.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭开了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

Preface

Introduction

This book describes the programming interface to the Unix system—the system call interface and many of the functions provided in the standard C library. It is intended for anyone writing programs that run under Unix.

Like most operating systems, Unix provides numerous services to the programs that are running—open a file, read a file, start a new program, allocate a region of memory, get the current time-of-day, and so on. This has been termed the *system call interface*. Additionally, the standard C library provides numerous functions that are used by almost every C program (format a variable's value for output, compare two strings, etc.).

The system call interface and the library routines have traditionally been described in Sections 2 and 3 of the *Unix Programmer's Manual*. This book is not a duplication of these sections. Examples and rationale are missing from the *Unix Programmer's Manual*, and that's what this book provides.

Unix Standards

The proliferation of different versions of Unix during the 1980s has been tempered by the various international standards that were started during the late 1980s. These include the ANSI standard for the C programming language, the IEEE POSIX family (still being developed), and the X/Open portability guide.

This book also describes these standards. But instead of just describing the standards by themselves, we describe them in relation to popular implementations of the standards—System V Release 4 and the forthcoming 4.4BSD. This provides a real-world description, which is often lacking from the standard itself and from books that describe only the standard.

Organization of the Book

This book is divided into six parts:

1. An overview and introduction to basic Unix programming concepts and terminology (Chapter 1), with a discussion of the various Unix standardization efforts and different Unix implementations (Chapter 2).
2. I/O—unbuffered I/O (Chapter 3), properties of files and directories (Chapter 4), the standard I/O library (Chapter 5), and the standard system files (Chapter 6).
3. Processes—the environment of a Unix process (Chapter 7), process control (Chapter 8), the relationships between different processes (Chapter 9), and signals (Chapter 10).
4. More I/O—terminal I/O (Chapter 11), advanced I/O (Chapter 12), and daemon processes (Chapter 13).
5. IPC—Interprocess communication (Chapters 14 and 15).
6. Examples—a database library (Chapter 16), communicating with a Postscript printer (Chapter 17), a modem dialing program (Chapter 18), and using pseudo terminals (Chapter 19).

A reading familiarity with C would be beneficial as would some experience using Unix. No prior programming experience with Unix is assumed. This text is intended for programmers familiar with Unix and programmers familiar with some other operating system who wish to learn the details of the services provided by most Unix systems.

Examples in the Text

This book contains many examples—approximately 10,000 lines of source code. All the examples are in the C programming language. Furthermore, these examples are in ANSI C. You should have a copy of the *Unix Programmer's Manual* for your system handy while reading this book, since reference is made to it for some of the more esoteric and implementation-dependent features.

Almost every function and system call is demonstrated with a small, complete program. This lets us see the arguments and return values and is often easier to comprehend than the use of the function in a much larger program. But since some of the small programs are contrived examples, a few bigger examples are also included (Chapters 16, 17, 18, and 19). These larger examples demonstrate the programming techniques in larger, real-world examples.

All the examples have been included in the text directly from their source files. A machine-readable copy of all the examples is available via anonymous FTP from the Internet host `ftp.uu.net` in the file `published/books/stevens.advprog.tar.Z`. Obtaining the source code allows you to modify the programs from this text and experiment with them on your system.

Acknowledgments

Once again I am indebted to my family for their love, support, and many lost weekends over the past year and a half. Writing a book is, in many ways, a family affair. Thank you Sally, Bill, Ellen, and David.

I am especially grateful to Brian Kernighan for his help in the book. His numerous thorough reviews of the entire manuscript and his gentle prodding for better prose hopefully show in the final result. Steve Rago was also a great resource, both in reviewing the entire manuscript and answering many questions about the details and history of System V. My thanks to the other technical reviewers used by Addison-Wesley, who provided valuable comments on various portions of the manuscript: Maury Bach, Mark Ellis, Jeff Gitlin, Peter Honeyman, John Linderman, Doug McIlroy, Evi Nemeth, Craig Partridge, Dave Presotto, Gary Wilson, and Gary Wright.

Keith Bostic and Kirk McKusick at the U.C. Berkeley CSRG provided an account that was used to test the examples on the latest BSD system. (Many thanks to Peter Salus too.) Sam Nataros and Joachim Sacksen at UHC provided the copy of SVR4 used to test the examples. Trent Hein helped obtain the alpha and beta copies of BSD/386.

Other friends have helped in many small, but significant ways over the past few years: Paul Lucchina, Joe Godsil, Jim Hogue, Ed Tankus, and Gary Wright. My editor at Addison-Wesley, John Wait, has been a great friend through it all. He never complained when the due date slipped and the page count kept increasing. A special thanks to the National Optical Astronomy Observatories (NOAO), especially Sidney Wolff, Richard Wolff, and Steve Grandi, for providing computer time.

Real Unix books are written using troff and this book follows that time-honored tradition. Camera-ready copy of the book was produced by the author using the groff package written by James Clark. Many thanks to James Clark for providing this excellent system and for his rapid response to bug fixes. Perhaps someday I will really understand troff footer traps.

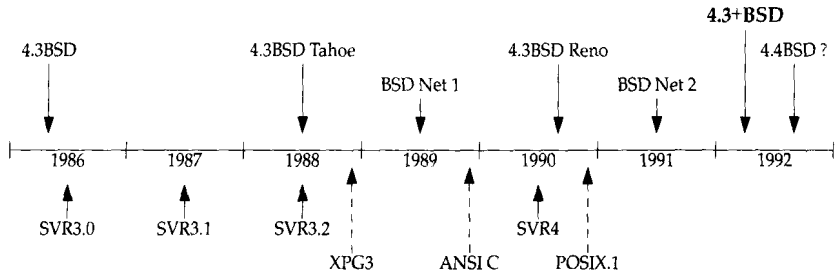
I welcome electronic mail from any readers with comments, suggestions, or bug fixes.

Tucson, Arizona
April 1992

W. Richard Stevens
rstevens@kohala.com
<http://www.kohala.com/~rstevens>

Systems Used to Test the Examples

Unfortunately all operating systems are moving targets. Unix is no exception. The following diagram shows the recent evolution of the various versions of System V and 4.xBSD.



4.xBSD are the various systems from the Computer Systems Research Group at the University of California at Berkeley. This group also distributes the BSD Net 1 and BSD Net 2 releases—publicly available source code from the 4.xBSD systems. SVR x refers to System V Release x from AT&T. XPG3 is the X/Open Portability Guide, Issue 3, and ANSI C is the ANSI standard for the C programming language. POSIX.1 is the IEEE and ISO standard for the interface to a Unix-like system. We'll have more to say about these different standards and the various versions of Unix in Sections 2.2 and 2.3.

In this text we use the term *4.3+BSD* to refer to the Unix system from Berkeley that is somewhere between the BSD Net 2 release and 4.4BSD.

At the time of this writing, 4.4BSD was not released, so the system could not be called 4.4BSD. Nevertheless a simple name was needed to refer to this system and *4.3+BSD* is used throughout the text.

Most of the examples in this text have been run on four different versions of Unix:

1. Unix System V/386 Release 4.0 Version 2.0 (“vanilla SVR4”) from U.H. Corp. (UHC), on an Intel 80386 processor.
2. 4.3+BSD at the Computer Systems Research Group, Computer Science Division, University of California at Berkeley, on a Hewlett Packard workstation.
3. BSD/386 (a derivative of the BSD Net 2 release) from Berkeley Software Design, Inc., on an Intel 80386 processor. This system is almost identical to what we call 4.3+BSD.
4. SunOS 4.1.1 and 4.1.2 (systems with a strong Berkeley heritage but many System V features) from Sun Microsystems, on a SPARCstation SLC.

Numerous timing tests are provided in the text and the systems used for the test are identified.

Contents

Preface

Chapter 1.	Introduction	1
1.1	Introduction	1
1.2	Logging In	1
1.3	Files and Directories	3
1.4	Input and Output	6
1.5	Programs and Processes	9
1.6	ANSI C Features	12
1.7	Error Handling	14
1.8	User Identification	16
1.9	Signals	17
1.10	Unix Time Values	19
1.11	System Calls and Library Functions	20
1.12	Summary	23
Chapter 2.	Unix Standardization and Implementations	25
2.1	Introduction	25
2.2	Unix Standardization	25
2.2.1	ANSI C	25
2.2.2	IEEE POSIX	26
2.2.3	X/Open XPG3	28
2.2.4	FIPS	28
2.3	Unix Implementations	28
2.3.1	System V Release 4	29
2.3.2	4.3+BSD	29

2.4	Relationship of Standards and Implementations	30
2.5	Limits	30
2.5.1	ANSI C Limits	31
2.5.2	POSIX Limits	32
2.5.3	XPG3 Limits	34
2.5.4	sysconf, pathconf, and fpathconf Functions	34
2.5.5	FIPS 151-1 Requirements	39
2.5.6	Summary of Limits	41
2.5.7	Indeterminate Run-Time Limits	41
2.6	Feature Test Macros	44
2.7	Primitive System Data Types	44
2.8	Conflicts Between Standards	45
2.9	Summary	46
Chapter 3.	File I/O	47
3.1	Introduction	47
3.2	File Descriptors	47
3.3	open Function	48
3.4	creat Function	50
3.5	close Function	51
3.6	lseek Function	51
3.7	read Function	54
3.8	write Function	55
3.9	I/O Efficiency	55
3.10	File Sharing	56
3.11	Atomic Operations	60
3.12	dup and dup2 Functions	61
3.13	fcntl Function	63
3.14	ioctl Function	67
3.15	/dev/fd	69
3.16	Summary	70
Chapter 4.	Files and Directories	73
4.1	Introduction	73
4.2	stat, fstat, and lstat Functions	73
4.3	File Types	74
4.4	Set-User-ID and Set-Group-ID	77
4.5	File Access Permissions	78
4.6	Ownership of New Files and Directories	81
4.7	access Function	82
4.8	umask Function	83
4.9	chmod and fchmod Functions	85
4.10	Sticky Bit	88
4.11	chown, fchown, and lchown Functions	89

4.12	File Size	90	
4.13	File Truncation	91	
4.14	Filesystems	92	
4.15	link, unlink, remove, and rename Functions	95	
4.16	Symbolic Links	99	
4.17	symlink and readlink Functions	102	
4.18	File Times	102	
4.19	utime Function	103	
4.20	mkdir and rmdir Functions	106	
4.21	Reading Directories	107	
4.22	chdir, fchdir, and getcwd Functions	112	
4.23	Special Device Files	114	
4.24	sync and fsync Functions	116	
4.25	Summary of File Access Permission Bits	117	
4.26	Summary	118	
Chapter 5.	Standard I/O Library		121
5.1	Introduction	121	
5.2	Streams and FILE Objects	121	
5.3	Standard Input, Standard Output, and Standard Error	122	
5.4	Buffering	122	
5.5	Opening a Stream	125	
5.6	Reading and Writing a Stream	127	
5.7	Line-at-a-Time I/O	130	
5.8	Standard I/O Efficiency	131	
5.9	Binary I/O	133	
5.10	Positioning a Stream	135	
5.11	Formatted I/O	136	
5.12	Implementation Details	138	
5.13	Temporary Files	140	
5.14	Alternatives to Standard I/O	143	
5.15	Summary	143	
Chapter 6.	System Data Files and Information		145
6.1	Introduction	145	
6.2	Password File	145	
6.3	Shadow Passwords	148	
6.4	Group File	149	
6.5	Supplementary Group IDs	150	
6.6	Other Data Files	152	
6.7	Login Accounting	153	
6.8	System Identification	154	
6.9	Time and Date Routines	155	
6.10	Summary	159	

Chapter 7.	The Environment of a Unix Process	161
7.1	Introduction	161
7.2	main Function	161
7.3	Process Termination	162
7.4	Command-Line Arguments	165
7.5	Environment List	166
7.6	Memory Layout of a C Program	167
7.7	Shared Libraries	169
7.8	Memory Allocation	169
7.9	Environment Variables	172
7.10	setjmp and longjmp Functions	174
7.11	getrlimit and setrlimit Functions	180
7.12	Summary	184
Chapter 8.	Process Control	187
8.1	Introduction	187
8.2	Process Identifiers	187
8.3	fork Function	188
8.4	vfork Function	193
8.5	exit Functions	195
8.6	wait and waitpid Functions	197
8.7	wait3 and wait4 Functions	202
8.8	Race Conditions	203
8.9	exec Functions	207
8.10	Changing User IDs and Group IDs	213
8.11	Interpreter Files	217
8.12	system Function	221
8.13	Process Accounting	226
8.14	User Identification	231
8.15	Process Times	232
8.16	Summary	235
Chapter 9.	Process Relationships	237
9.1	Introduction	237
9.2	Terminal Logins	237
9.3	Network Logins	241
9.4	Process Groups	243
9.5	Sessions	244
9.6	Controlling Terminal	246
9.7	tcgetpgrp and tcsetpgrp Functions	247
9.8	Job Control	248
9.9	Shell Execution of Programs	252

9.10	Orphaned Process Groups	256	
9.11	4.3+BSD Implementation	259	
9.12	Summary	261	
Chapter 10.	Signals		263
10.1	Introduction	263	
10.2	Signal Concepts	263	
10.3	signal Function	270	
10.4	Unreliable Signals	274	
10.5	Interrupted System Calls	275	
10.6	Reentrant Functions	278	
10.7	SIGCLD Semantics	279	
10.8	Reliable Signal Terminology and Semantics		282
10.9	kill and raise Functions	283	
10.10	alarm and pause Functions	285	
10.11	Signal Sets	291	
10.12	sigprocmask Function	292	
10.13	sigpending Function	293	
10.14	sigaction Function	296	
10.15	sigsetjmp and siglongjmp Functions		299
10.16	sigsuspend Function	303	
10.17	abort Function	309	
10.18	system Function	310	
10.19	sleep Function	317	
10.20	Job-Control Signals	319	
10.21	Additional Features	320	
10.22	Summary	323	
Chapter 11.	Terminal I/O		325
11.1	Introduction	325	
11.2	Overview	325	
11.3	Special Input Characters	331	
11.4	Getting and Setting Terminal Attributes		335
11.5	Terminal Option Flags	336	
11.6	stty Command	342	
11.7	Baud Rate Functions	343	
11.8	Line Control Functions	344	
11.9	Terminal Identification	345	
11.10	Canonical Mode	349	
11.11	Noncanonical Mode	352	
11.12	Terminal Window Size	358	
11.13	termcap, terminfo, and curses		360
11.14	Summary	360	

Chapter 12.	Advanced I/O	363
12.1	Introduction	363
12.2	Nonblocking I/O	363
12.3	Record Locking	367
12.4	Streams	383
12.5	I/O Multiplexing	394
	12.5.1 select Function	396
	12.5.2 poll Function	400
12.6	Asynchronous I/O	402
	12.6.1 System V Release 4	403
	12.6.2 4.3+BSD	403
12.7	readv and writev Functions	404
12.8	readn and writen Functions	406
12.9	Memory Mapped I/O	407
12.10	Summary	413
Chapter 13.	Daemon Processes	415
13.1	Introduction	415
13.2	Daemon Characteristics	415
13.3	Coding Rules	417
13.4	Error Logging	418
	13.4.1 SVR4 Streams log Driver	419
	13.4.2 4.3+BSD syslog Facility	421
13.5	Client-Server Model	424
13.6	Summary	424
Chapter 14.	Interprocess Communication	427
14.1	Introduction	427
14.2	Pipes	428
14.3	popen and pclose Functions	435
14.4	Coprocesses	441
14.5	FIFOs	445
14.6	System V IPC	449
	14.6.1 Identifiers and Keys	449
	14.6.2 Permission Structure	450
	14.6.3 Configuration Limits	451
	14.6.4 Advantages and Disadvantages	451
14.7	Message Queues	453
14.8	Semaphores	457
14.9	Shared Memory	463
14.10	Client-Server Properties	470
14.11	Summary	472

Chapter 15.	Advanced Interprocess Communication	475
15.1	Introduction	475
15.2	Stream Pipes	475
15.3	Passing File Descriptors	479
15.3.1	System V Release 4	481
15.3.2	4.3BSD	484
15.3.3	4.3+BSD	487
15.4	An Open Server, Version 1	490
15.5	Client-Server Connection Functions	496
15.5.1	System V Release 4	497
15.5.2	4.3+BSD	501
15.6	An Open Server, Version 2	505
15.7	Summary	514
Chapter 16.	A Database Library	515
16.1	Introduction	515
16.2	History	515
16.3	The Library	516
16.4	Implementation Overview	518
16.5	Centralized or Decentralized?	521
16.6	Concurrency	522
16.7	Source Code	524
16.8	Performance	545
16.9	Summary	550
Chapter 17.	Communicating with a PostScript Printer	551
17.1	Introduction	551
17.2	PostScript Communication Dynamics	551
17.3	Printer Spooling	554
17.4	Source Code	556
17.5	Summary	578
Chapter 18.	A Modem Dialer	579
18.1	Introduction	579
18.2	History	579
18.3	Program Design	580
18.4	Data Files	582
18.5	Server Design	584
18.6	Server Source Code	586
18.7	Client Design	615
18.8	Client Source Code	617
18.9	Summary	629