

Contents at a Glance

1	Introduction to UNIX/Linux Shells	1
2	Shell Programming QuickStart	33
3	Regular Expressions and Pattern Matching	67
4	The grep Family	81
5	sed, the Streamlined Editor	125
6	The awk Utility	157
7	The Interactive Bourne Shell	279
8	Programming the Bourne Shell	321
9	The Interactive C and TC Shells	403
10	Programming the C and TC Shells	525
11	The Interactive Korn Shell	583
12	Programming the Korn Shell	653
13	The Interactive Bash Shell	753
14	Programming the Bash Shell	865
15	Debugging Shell Scripts	967
16	The System Administrator and the Shell	1023
A	Useful UNIX/Linux Utilities for Shell Programmers	1055
B	Comparison of the Shells	1103
	Index	1113

Contents

Preface xxxiii

Acknowledgments xxxv

1 Introduction to UNIX/Linux Shells 1

1.1 What Is UNIX? What Is Linux? A Little History 1

1.1.1 A Little Bit About UNIX 1

1.1.2 Why Linux? 2

The Free Software Foundation 3

1.2 Definition and Function of a Shell 3

1.2.1 The UNIX Shells 4

1.2.2 The Linux Shells 5

What Is POSIX? 6

1.3 History of the Shell 6

1.3.1 Uses of the Shell 8

1.3.2 Responsibilities of the Shell 8

1.4 System Startup and the Login Shell 9

1.4.1 Parsing the Command Line 9

1.4.2 Types of Commands 10

1.5 Processes and the Shell 11

1.5.1 What Processes Are Running? 12

The ps Command 12

The pstree/ptree Command 13

1.5.2 What Are System Calls? 14

1.5.3 Creating Processes 14

The fork System Call 14

The wait System Call 15

The exec System Call 15

The exit System Call 15

Killing Processes 17

1.6	The Environment and Inheritance	18
1.6.1	Ownership and Permissions	18
1.6.2	The File Creation Mask	19
1.6.3	Changing Permissions and Ownership	20
	<i>The chmod Command</i>	20
	<i>The chown Command</i>	21
1.6.4	The Working Directory	23
1.6.5	Variables	23
1.6.6	Redirection and Pipes	24
	<i>File Descriptors</i>	24
	<i>Redirection</i>	25
	<i>Pipes</i>	28
1.6.7	The Shell and Signals	30
1.7	Executing Commands from Scripts	32
2	Shell Programming QuickStart	33
2.1	Taking a Peek at Shell Scripts	33
2.2	Sample Scripts: Comparing the Major Shells	33
	<i>Before Getting Started</i>	34
	<i>The Purpose</i>	34
2.3	The C and TC Shell Syntax and Constructs	34
2.3.1	The C/TC Shell Script	39
2.4	The Bourne Shell Syntax and Constructs	41
2.4.1	The Bourne Shell Script	47
2.5	The Korn Shell Constructs	49
2.5.1	The Korn Shell Script	56
2.6	The Bash Shell Constructs	58
2.6.1	The Bash Shell Script	64
3	Regular Expressions and Pattern Matching	67
3.1	Regular Expressions	67
3.1.1	Definition and Example	67
3.1.2	Regular Expression Metacharacters	69
3.2	Combining Regular Expression Metacharacters	75
3.2.1	More Regular Expression Metacharacters	76
4	The grep Family	81
4.1	The grep Command	81
4.1.1	The Meaning of grep	81
4.1.2	How grep Works	82
4.1.3	Metacharacters	83
4.1.4	grep and Exit Status	85
4.2	grep Examples with Regular Expressions	86

4.3	grep with Options	91
4.4	grep with Pipes	94
4.4.1	grep Review	94
4.5	egrep (Extended grep)	95
4.5.1	egrep Examples	96
4.5.2	egrep Review	98
4.6	fgrep (Fixed grep or Fast grep)	99
4.7	Linux and GNU grep	100
4.7.1	Basic and Extended Regular Expressions	100
	<i>The POSIX Class</i>	103
4.8	GNU Basic grep (grep -G) with Regular Expressions	104
4.9	grep -E or egrep (GNU Extended grep)	105
4.9.1	grep -E and egrep Examples	106
4.9.2	Anomalies with Regular and Extended Variants of grep	110
4.10	Fixed grep (grep -F and fgrep)	113
4.11	Recursive grep (rgrep, grep -R)	113
4.12	GNU grep with Options	114
4.13	grep with Options (UNIX and GNU)	116
4.13.1	GNU grep Options Examples	118
	LAB 1: grep Exercise	124

5 sed, the Streamlined Editor 125

5.1	What Is sed?	125
5.2	Versions of sed	125
5.3	How Does sed Work?	126
5.4	Regular Expressions	126
5.5	Addressing	128
5.6	Commands and Options	128
5.6.1	How to Modify a File with sed	130
5.6.2	GNU sed Options	131
5.7	Error Messages and Exit Status	131
5.8	Metacharacters	132
5.9	sed Examples	133
5.9.1	Printing: The p Command	134
5.9.2	Deleting: The d Command	134
5.9.3	Substitution: The s Command	136
5.9.4	Range of Selected Lines: The Comma	139
5.9.5	Multiple Edits: The e Command	141
5.9.6	Reading from Files: The r Command	141
5.9.7	Writing to Files: The w Command	142
5.9.8	Appending: The a Command	143
5.9.9	Inserting: The i Command	144
5.9.10	Changing: The c Command	145

5.9.11	Next: The <code>n</code> Command	145
5.9.12	Transform: The <code>y</code> Command	146
5.9.13	Quit: The <code>q</code> Command	147
5.9.14	Holding and Getting: The <code>h</code> and <code>g</code> Commands	148
5.9.15	Holding and Exchanging: The <code>h</code> and <code>x</code> Commands	151
5.10	sed Scripting	151
5.10.1	sed Script Examples	152
5.10.2	sed Review	154
LAB 2:	sed Exercise	155

6 The awk Utility 157

6.1	What's awk? What's nawk? What's gawk?	157
6.1.1	What Does awk Stand For?	157
6.1.2	Which awk?	157
6.2	awk's Format	158
6.2.1	Input from Files	158
6.2.2	Input from Commands	159
6.3	How awk Works	160
6.4	Formatting Output	161
6.4.1	The print Function	161
	Escape Sequences	162
6.4.2	The OFMT Variable	163
6.4.3	The printf Function	163
6.5	awk Commands from Within a File	166
6.6	Records and Fields	167
6.6.1	Records	167
	The Record Separator	167
	The <code>\$0</code> Variable	167
	The <code>NR</code> Variable	168
6.6.2	Fields	168
6.6.3	Field Separators	169
	The Input Field Separator	169
	Changing the Field Separator at the Command Line	170
	Using More Than One Field Separator	170
	The Output Field Separator	170
6.7	Patterns and Actions	171
6.7.1	Patterns	171
6.7.2	Actions	172
6.8	Regular Expressions	173
6.8.1	Matching on an Entire Line	175
6.8.2	The match Operator	175
	The POSIX Character Class	176
6.9	awk Commands in a Script File	177
6.10	Review	178

6.10.1	Simple Pattern Matching	178
6.10.2	Simple Actions	179
6.10.3	Regular Expressions in Pattern and Action Combinations	182
6.10.4	Input Field Separators	184
6.10.5	awk Scripting	186
LAB 3: awk Exercise		187
6.11	Comparison Expressions	188
6.11.1	Relational and Equality Operators	188
6.11.2	Conditional Expressions	190
6.11.3	Computation	190
6.11.4	Logical Operators and Compound Patterns	191
6.11.5	Range Patterns	192
6.11.6	A Data Validation Program	192
6.12	Review	194
6.12.1	Equality Testing	194
6.12.2	Relational Operators	195
6.12.3	Logical Operators (&&,)	196
6.12.4	Logical NOT Operator (!)	197
6.12.5	Arithmetic Operators	197
6.12.6	Range Operator	200
6.12.7	Conditional Operator	200
6.12.8	Assignment Operators	201
LAB 4: awk Exercise		202
6.13	Variables	203
6.13.1	Numeric and String Constants	203
	<i>Initialization and Type Coercion</i>	203
6.13.2	User-Defined Variables	203
	<i>Increment and Decrement Operators</i>	204
	<i>User-Defined Variables at the Command Line</i>	205
	<i>The -v Option (nawk)</i>	205
	<i>Field Variables</i>	205
	<i>Built-In Variables</i>	206
6.13.3	BEGIN Patterns	208
6.13.4	END Patterns	208
6.14	Redirection and Pipes	209
6.14.1	Output Redirection	209
6.14.2	Input Redirection (getline)	210
	<i>The getline Function</i>	210
6.15	Pipes	212
6.15.1	Closing Files and Pipes	213
	<i>The system Function</i>	213
6.16	Review	214
6.16.1	Increment and Decrement Operators	214
6.16.2	Built-In Variables	215

6.16.3	BEGIN Patterns	219
6.16.4	END Patterns	220
6.16.5	awk Script with BEGIN and END	221
6.16.6	The printf Function	223
6.16.7	Redirection and Pipes	224
6.16.8	Opening and Closing a Pipe	225
LAB 5: nawk Exercise 226		
6.17	Conditional Statements	227
6.17.1	if Statements	227
6.17.2	if/else Statements	228
6.17.3	if/else and else if Statements	229
6.18	Loops	230
6.18.1	while Loop	230
6.18.2	for Loop	231
6.18.3	Loop Control	231
	break and continue Statements	231
6.19	Program Control Statements	232
6.19.1	next Statement	232
6.19.2	exit Statement	232
6.20	Arrays	233
6.20.1	Subscripts for Associative Arrays	233
	Using Variables As Array Indexes	233
	The Special-for Loop	234
	Using Strings As Array Subscripts	235
	Using Field Values As Array Subscripts	236
	Arrays and the split Function	238
	The delete Function	238
	Multidimensional Arrays (nawk)	238
6.20.2	Processing Command Arguments (nawk)	239
	ARGV	239
	ARGC	239
6.21	awk Built-In Functions	242
6.21.1	String Functions	242
	The sub and gsub Functions	242
	The index Function	243
	The length Function	243
	The substr Function	244
	The match Function	244
	The split Function	245
	The sprintf Function	245
6.22	Built-In Arithmetic Functions	246
6.22.1	Integer Function	246
6.22.2	Random Number Generator	247
	The rand Function	247
	The srand Function	247

6.23	User-Defined Functions (<i>nawk</i>)	248
6.24	Review	250
LAB 6:	<i>nawk</i> Exercise	255
6.25	Odds and Ends	256
6.25.1	Fixed Fields	256
	Empty Fields	257
	Numbers with \$, Commas, or Other Characters	257
6.25.2	Multiline Records	258
6.25.3	Generating Form Letters	260
6.25.4	Interaction with the Shell	262
LAB 7:	<i>nawk</i> Exercise	264
6.26	<i>awk</i> Built-In Functions	264
6.26.1	String Functions	265
	The <i>toupper</i> and <i>tolower</i> Functions (<i>gawk</i> only)	268
6.26.2	Time Functions with <i>gawk</i>	269
	The <i>system</i> Function	269
	The <i>strftime</i> Function	269
6.26.3	Command-Line Arguments	271
6.26.4	Reading Input (<i>getline</i>)	272
6.26.5	Control Functions	273
6.26.6	User-Defined Functions	274
6.26.7	<i>awk/gawk</i> Command-Line Options	275

7 The Interactive Bourne Shell 279

7.1	Introduction	279
7.2	The Environment	280
7.2.1	The Initialization Files	280
	The <i>/etc/profile</i> File	280
	The <i>.profile</i> File	282
7.2.2	The Prompts	283
	The Primary Prompt	283
	The Secondary Prompt	284
7.2.3	The Search Path	285
7.2.4	The <i>hash</i> Command	285
7.2.5	The <i>dot</i> Command	286
7.3	The Command Line	287
7.3.1	The Exit Status	287
7.3.2	Multiple Commands at the Command Line	288
	Grouping Commands	288
7.3.3	Conditional Execution of Commands	288
7.3.4	Commands in the Background	289
7.4	Shell Metacharacters (Wildcards)	290
7.5	Filename Substitution	290
7.5.1	The Asterisk	291

7.5.2	The Question Mark	291
7.5.3	The Square Brackets	292
7.5.4	Escaping Metacharacters	293
7.6	Variables	294
7.6.1	Local Variables	294
	Setting Local Variables	294
	The Scope of Local Variables	295
	Setting Read-Only Variables	296
7.6.2	Environment Variables	296
	Setting Environment Variables	297
7.6.3	Listing Set Variables	298
7.6.4	Unsetting Variables	299
7.6.5	Printing the Values of Variables: The echo Command	300
7.6.6	Variable Expansion Modifiers	301
7.6.7	Positional Parameters	303
7.6.8	Other Special Variables	305
7.7	Quoting	306
7.7.1	The Backslash	307
7.7.2	Single Quotes	308
7.7.3	Double Quotes	308
7.8	Command Substitution	309
7.9	An Introduction to Functions	310
7.9.1	Defining Functions	310
7.9.2	Listing and Unsetting Functions	311
7.10	Standard I/O and Redirection	312
7.10.1	The exec Command and Redirection	313
7.11	Pipes	316
7.12	The here document and Input	318
7.12.1	Now What?	320

8 Programming the Bourne Shell 321

8.1	Introduction	321
8.1.1	The Steps in Creating a Shell Script	321
	The First Line	321
	Comments	321
	Executable Statements and Bourne Shell Constructs	322
	Making the Script Executable	322
	A Scripting Session	322
8.2	Reading User Input	323
8.3	Arithmetic	325
8.3.1	Integer Arithmetic and the expr Command	325
8.3.2	Floating-Point Arithmetic	327
8.4	Positional Parameters and Command-Line Arguments	327
8.4.1	The set Command and Positional Parameters	329
8.4.2	How \$* and \$@ Differ	331

8.5	Conditional Constructs and Flow Control	333
8.5.1	Testing Exit Status: The test Command	334
8.5.2	The if Command	335
8.5.3	The exit Command and the ? Variable	337
8.5.4	Checking for Null Values	339
8.5.5	The if/else Command	340
8.5.6	The if/elif/else Command	342
8.5.7	File Testing	344
8.5.8	The null Command	344
8.5.9	The case Command	346
8.5.10	Creating Menus with the here document and case Command	348
8.6	Looping Commands	349
8.6.1	The for Command	349
8.6.2	The \$* and \$@ Variables in Wordlists	352
8.6.3	The while Command	354
8.6.4	The until Command	357
8.6.5	Looping Control Commands	358
	The shift Command	359
	The break Command	361
	The continue Command	362
8.6.6	Nested Loops and Loop Control	364
8.6.7	I/O Redirection and Subshells	365
	Redirecting the Output of a Loop to a File	365
	Piping the Output of a Loop to a UNIX Command	368
8.6.8	Running Loops in the Background	368
8.6.9	The exec Command and Loops	369
8.6.10	IFS and Loops	370
8.7	Functions	372
8.7.1	Unsetting Functions	373
8.7.2	Function Arguments and the Return Value	373
8.7.3	Functions and the dot Command	375
	Storing Functions	375
8.8	Trapping Signals	378
8.8.1	Resetting Signals	379
8.8.2	Ignoring Signals	380
8.8.3	Listing Traps	380
8.8.4	Traps in Functions	382
8.8.5	Debugging	383
8.9	The Command Line	384
8.9.1	Processing Command-Line Options with getopt	384
	getopts Scripts	386
8.9.2	The eval Command and Parsing the Command Line	391
8.10	Shell Invocation Options	392
8.10.1	The set Command and Options	393

8.10.2	Shell Built-In Commands	394
LAB 8:	Bourne Shell—Getting Started	395
LAB 9:	Metacharacters	396
LAB 10:	Redirection	396
LAB 11:	First Script	397
LAB 12:	Command-Line Arguments	397
LAB 13:	Getting User Input	398
LAB 14:	Conditional Statements	399
LAB 15:	Conditionals and File Testing	399
LAB 16:	The case Statement	400
LAB 17:	Loops	400
LAB 18:	Functions	401

9 The Interactive C and TC Shells 403

9.1	Introduction	403
9.1.1	C/TC Shell Startup	403
9.2	The Environment	404
9.2.1	Initialization Files	404
	<i>The .cshrc and .tcshrc Files</i>	405
	<i>The .login File</i>	406
9.2.2	The Search Path	407
9.2.3	The rehash Command	408
9.2.4	The hashstat Command	408
9.2.5	The source Command	408
9.2.6	The Shell Prompts	409
	<i>The Primary Prompt</i>	409
	<i>The Secondary Prompt</i>	409
9.3	The C/TC Shell Command Line	410
9.3.1	The Exit Status	410
9.3.2	Command Grouping	411
9.3.3	Conditional Execution of Commands	412
9.3.4	Commands in the Background	412
9.3.5	Command-Line History	413
	<i>Setting History</i>	414
	<i>Saving History</i>	414
	<i>Displaying History</i>	414
	<i>Re-executing Commands</i>	416
9.4	Aliases	418
9.4.1	Listing Aliases	418
9.4.2	Creating Aliases	419
9.4.3	Deleting Aliases	419
9.4.4	Alias Loop	420
9.5	Manipulating the Directory Stack	420
9.5.1	The pushd and popd Commands	420

9.6	Job Control	423
9.6.1	The Ampersand and Background Jobs	423
9.6.2	The Suspend Key Sequence and Background Jobs	424
9.6.3	The jobs Command	424
9.6.4	The Foreground and Background Commands	425
9.7	Shell Metacharacters	426
9.8	Filename Substitution	427
9.8.1	The Asterisk	427
9.8.2	The Question Mark	428
9.8.3	The Square Brackets	429
9.8.4	The Curly Braces	429
9.8.5	Escaping Metacharacters	430
9.8.6	Tilde Expansion	430
9.8.7	Filename Completion: The filec Variable	431
9.8.8	Turning Off Metacharacters with noglob	431
9.9	Redirection and Pipes	432
9.9.1	Redirecting Input	433
9.9.2	The here document	433
9.9.3	Redirecting Output	435
9.9.4	Appending Output to an Existing File	436
9.9.5	Redirecting Output and Error	436
9.9.6	Separating Output and Errors	438
9.9.7	The noclobber Variable	439
9.10	Variables	440
9.10.1	Curly Braces	441
9.10.2	Local Variables	441
	Setting Local Variables	441
	The set Command	442
	Read-Only Variables (tcsh)	444
	Built-In Local Variables	444
9.10.3	Environment Variables	444
	Printing Environment Variables	445
9.10.4	Arrays	447
	The shift Command and Arrays	448
	Creating an Array from a String	449
9.10.5	Special Variables	450
	Pathname Variable Modifiers	451
9.11	Command Substitution	452
9.11.1	Wordlists and Command Substitution	453
9.12	Quoting	455
9.12.1	The Backslash	455
9.12.2	Single Quotes	456
9.12.3	Double Quotes	457
9.12.4	The Quoting Game	458
	Quoting with the :q Modifier	458

	Quoting with the :x Modifier	460
9.13	New Features of the Interactive TC Shell	460
9.13.1	Versions of tcsh	461
9.13.2	The Shell Prompts	461
	The Primary Prompt	463
	The Secondary Prompt	463
9.14	The TC Shell Command Line	464
9.14.1	The Command Line and Exit Status	464
	Exit Status and the printexitvalue Variable	465
9.14.2	TC Shell Command-Line History	466
	The history Variable	467
	Saving History and the savehist Variable	468
	Displaying History	468
	Accessing Commands from the History File	470
9.14.3	The Built-In Command-Line Editors	474
	The bindkey Built-In Command	475
	The vi Built-In Editor	475
	The emacs Built-In Editor	476
	Binding Keys	477
9.15	TC Shell Command, Filename, and Variable Completion	482
9.15.1	The autolist Variable	482
9.15.2	The fignore Variable	484
9.15.3	The complete Shell Variable	484
9.15.4	Programming Completions	485
	Types of Completions	485
9.16	TC Shell Spelling Correction	490
9.17	TC Shell Aliases	491
9.17.1	Listing Aliases	491
9.17.2	Creating Aliases	492
9.17.3	Deleting Aliases	493
9.17.4	Alias Loop	494
9.17.5	Special tcsh Aliases	494
9.18	TC Shell Job Control	495
9.18.1	The jobs Command and the listjobs Variable	495
9.18.2	Foreground and Background Commands	497
9.18.3	Scheduling Jobs	497
9.19	Printing the Values of Variables in the TC Shell	499
9.19.1	The echo Command	499
9.19.2	The printf Command	500
9.19.3	Curly Braces and Variables	502
9.19.4	Uppercase and Lowercase Modifiers	502
9.20	TC Shell Built-In Commands	503
9.20.1	Special Built-In T/TC Shell Variables	514
9.20.2	TC Shell Command-Line Switches	520
	Lab 19: The TC Shell—Getting Started	522

Lab 20: History	522
Lab 21: Shell Metacharacters	523
Lab 23: Variables and Arrays	524
Lab 22: Redirection	524

10 Programming the C and TC Shells 525

10.1	Introduction	525
10.1.1	The Steps in Creating a Shell Script	525
	<i>The First Line</i>	525
	<i>Comments</i>	526
	<i>Making the Script Executable</i>	526
	<i>An Example Scripting Session</i>	526
10.2	Reading User Input	527
10.2.1	The <code><</code> Variable	527
10.2.2	Creating a Wordlist from the Input String	528
10.3	Arithmetic	529
10.3.1	Arithmetic Operators	529
10.3.2	Floating-Point Arithmetic	530
10.4	Debugging Scripts	531
10.5	Command-Line Arguments	534
10.5.1	Positional Parameters and <code>argv</code>	534
10.6	Conditional Constructs and Flow Control	535
10.6.1	Testing Expressions	536
10.6.2	Precedence and Associativity	537
10.6.3	The <code>if</code> Statement	538
10.6.4	Testing and Unset or Null Variables	539
10.6.5	The <code>if/else</code> Statement	540
10.6.6	Logical Expressions	540
10.6.7	The <code>if</code> Statement and a Single Command	542
10.6.8	The <code>if/else if</code> Statement	542
10.6.9	Exit Status and the Status Variable	543
10.6.10	Exiting from a Shell Script	544
10.6.11	Using an Alias to Create an Error Message	544
10.6.12	Using the Status Variable in a Script	545
10.6.13	Evaluating Commands Within Conditionals	546
10.6.14	The <code>goto</code>	547
10.6.15	File Testing with the C Shell	548
10.6.16	The <code>test</code> Command and File Testing	549
10.6.17	Nesting Conditionals	550
10.6.18	File Testing with the TC Shell	551
10.6.19	The <code>filetest</code> Built-In (<code>tcsh</code>)	554
10.6.20	Additional TC Shell File-Testing Operators	554
10.6.21	The <code>switch</code> Command	557
	<i>Nesting Switches</i>	558

10.6.22	The here document and Menus	560
10.7	Looping Commands	561
10.7.1	The foreach Loop	561
10.7.2	The while Loop	564
10.7.3	The repeat Command	565
10.7.4	Looping Control Commands	566
	<i>The shift Command</i>	566
	<i>The break Command</i>	566
	<i>Nested Loops and the repeat Command</i>	568
	<i>The continue Command</i>	569
10.8	Interrupt Handling	570
10.9	setuid Scripts	571
10.10	Storing Scripts	572
10.11	Built-In Commands	572
Lab 24:	C/TC Shells—Getting Started	577
Lab 25:	Shell Metacharacters	577
Lab 26:	Redirection	578
Lab 27:	First Script	578
Lab 28:	Getting User Input	579
Lab 29:	Command-Line Arguments	580
Lab 30:	Conditionals and File Testing	580
Lab 31:	The switch Statement	581
Lab 32:	Loops	581

11 The Interactive Korn Shell 583

11.1	Introduction	583
11.1.1	Startup	583
11.2	The Environment	584
11.2.1	The Initialization Files	584
	<i>The /etc/profile File</i>	584
	<i>The .profile File</i>	585
	<i>The ENV File</i>	587
	<i>The set -o Options</i>	588
11.2.2	The Prompts	589
	<i>The Primary Prompt</i>	589
	<i>The Secondary Prompt</i>	590
11.2.3	The Search Path	591
11.2.4	The dot Command	591
11.3	The Command Line	592
11.3.1	The Order of Processing Commands	592
11.3.2	The Exit Status	593
11.3.3	Multiple Commands and Command Grouping	594
11.3.4	Conditional Execution of Commands	594
11.3.5	Commands in the Background	595

11.3.6	Command-Line History	595
	<i>The history Command/Redisplay Commands</i>	597
	<i>Re-executing Commands with the r Command</i>	598
11.3.7	Command-Line Editing	599
	<i>The vi Built-In Editor</i>	599
	<i>The emacs Built-In Editor</i>	600
	<i>FCEDIT and Editing Commands</i>	601
11.4	Commenting and Filename Expansion	602
11.5	Aliases	604
11.5.1	Listing Aliases	604
11.5.2	Creating Aliases	605
11.5.3	Deleting Aliases	605
11.5.4	Tracked Aliases	606
11.6	Job Control	606
11.7	Metacharacters	607
11.8	Filename Substitution (Wildcards)	610
11.8.1	The Asterisk	610
11.8.2	The Question Mark	611
11.8.3	The Square Brackets	612
11.8.4	Escaping Metacharacters	613
11.8.5	Tilde and Hyphen Expansion	613
11.8.6	New ksh Metacharacters	614
11.8.7	The noglob Variable	615
11.9	Variables	616
11.9.1	Local Variables	616
	<i>Setting and Referencing Local Variables</i>	616
	<i>The Scope of Local Variables</i>	617
	<i>Setting Read-Only Variables</i>	618
11.9.2	Environment Variables	618
	<i>Setting Environment Variables</i>	619
	<i>Special Environment Variables</i>	620
11.9.3	Listing Set Variables	621
11.9.4	Unsetting Variables	624
11.9.5	Printing the Values of Variables	624
11.9.6	Escape Sequences	626
11.9.7	Variable Expressions and Expansion Modifiers	627
11.9.8	Variable Expansion of Substrings	630
11.9.9	Variable Attributes: The typeset Command	631
11.9.10	Positional Parameters	633
11.9.11	Other Special Variables	635
11.10	Quoting	636
11.10.1	The Backslash	636
11.10.2	Single Quotes	636
11.10.3	Double Quotes	637
11.11	Command Substitution	638

11.12	Functions	639
11.12.1	Defining Functions	640
11.12.2	Functions and Aliases	641
11.12.3	Listing Functions	642
11.12.4	Unsetting Functions	643
11.13	Standard I/O and Redirection	643
11.13.1	The exec Command and Redirection	645
11.13.2	Redirection and the Child Shell	646
11.14	Pipes	646
11.14.1	The here document and Redirecting Input	648
11.15	Timing Commands	650
11.15.1	The time Command	650
11.15.2	The TMOUT Variable	651
12	Programming the Korn Shell	653
12.1	Introduction	653
12.1.1	The Steps in Creating a Shell Script	653
	<i>The First Line</i>	653
	<i>Comments</i>	653
	<i>Executable Statements and Korn Shell Constructs</i>	654
	<i>Naming and Storing Scripts</i>	654
	<i>Making a Script Executable</i>	654
	<i>Using a Script As an Argument to ksh</i>	655
	<i>A Scripting Session</i>	655
12.2	Reading User Input	656
12.2.1	read and File Descriptors	658
12.2.2	Reading Through Files	659
12.3	Arithmetic	660
12.3.1	The Integer Type	660
12.3.2	Using Different Bases	661
12.3.3	Listing Integers	662
12.3.4	Arithmetic Operators and the let Command	662
12.4	Positional Parameters and Command-Line Arguments	665
12.4.1	The set Command and Positional Parameters	665
12.5	Conditional Constructs and Flow Control	668
12.5.1	Testing Exit Status and the \$? Variable	668
12.5.2	The Old test Command	670
12.5.3	The New test Command	671
12.5.4	File Testing with Binary Operators	673
12.5.5	Expression Testing with Logical Operators	673
12.5.6	File Testing with Flags	673
12.5.7	The if Command	675
12.5.8	Using the Old-Style Bourne test	676
12.5.9	Using the New-Style Korn test	677
12.5.10	Using the Old-Style Bourne test with Numbers	678

12.5.11	The let Command and Testing Numbers	678
12.5.12	The if/else Command	679
12.5.13	The if/elif/else Command	679
12.5.14	The exit Command	681
12.5.15	The null Command	683
12.5.16	The case Command	684
	<i>The case Command and the here document</i>	685
12.6	Looping Commands	686
12.6.1	The for Command	686
12.6.2	The \$* and \$@ Variables in Wordlists	689
12.6.3	The while Command	689
12.6.4	The until Command	692
12.6.5	The select Command and Menus	693
12.6.6	Looping Control Commands	697
	<i>The shift Command</i>	697
	<i>The break Command</i>	699
	<i>The continue Command</i>	700
12.6.7	Nested Loops and Loop Control	701
12.6.8	I/O Redirection and Loops	702
	<i>Redirecting the Output of a Loop to a File</i>	702
	<i>Piping the Output of a Loop to a UNIX Command</i>	703
12.6.9	Running Loops in the Background	704
12.6.10	The exec Command and Loops	705
12.6.11	The IFS and Loops	706
12.7	Arrays	707
12.7.1	Creating Arrays with the set Command	709
12.8	Functions	709
12.8.1	Defining Functions	710
12.8.2	Listing and Unsetting Functions	710
12.8.3	Local Variables and the Return Value	711
12.8.4	Exported Functions	712
12.8.5	Function Options and the typeset Command	714
	<i>Autoloaded Functions</i>	714
12.9	Trapping Signals	716
12.9.1	Pseudo or Fake Signals	717
12.9.2	Resetting Signals	718
12.9.3	Ignoring Signals	718
12.9.4	Listing Traps	718
12.9.5	Traps and Functions	721
12.10	Coprocesses	722
12.11	Debugging	725
12.12	The Command Line	728
12.12.1	Processing Command-Line Options with getopt	728
	<i>Sample getopt Scripts</i>	730
12.13	Security	735

12.13.1 Privileged Scripts	735
12.13.2 Restricted Shells	735
12.14 Built-In Commands	735
12.15 Korn Shell Invocation Arguments	740
Lab 33: Korn Shell—Getting Started	741
Lab 34: History	742
Lab 35: Aliases and Functions	743
Lab 36: Shell Metacharacters	744
Lab 37: Tilde Expansion, Quotes, and Command Substitution	744
Lab 38: Redirection	745
Lab 39: Job Control	746
Lab 40: Writing the info Shell Script	746
Lab 41: Variable Expansion of Substrings	747
Lab 42: The <code>lookup</code> Script	747
Lab 43: Using <code>typeset</code>	748
Lab 44: The <code>if/else</code> Construct and the <code>let</code> Command	748
Lab 45: The <code>case</code> Statement	749
Lab 46: The <code>select</code> Loop	749
Lab 47: Autoloading Functions	751

13 The Interactive Bash Shell 753

13.1 Introduction	753
13.1.1 Versions of <code>bash</code>	753
13.1.2 Startup	754
<i>Changing the Shell at the Command Line</i>	755
13.2 The Environment	756
13.2.1 The Initialization Files	756
The <code>/etc/profile</code> File	756
The <code>~/.bash_profile</code> File	758
The <code>BASH_ENV</code> (ENV) Variable	759
The <code>.bashrc</code> File	759
The <code>/etc/bashrc</code> File	761
The <code>~/.profile</code> File	762
The <code>~/.bash_logout</code> File	763
Options to Prevent Startup Files from Being Executed	763
The <code>.inputrc</code> File	763
13.2.2 Setting bash Options with the Built-In <code>set</code> and <code>shopt</code> Commands	763
The <code>set -o</code> Options	763
The <code>shopt</code> Built-In (Version 2.x+)	766
13.2.3 The Prompts	768
The Primary Prompt	768
Setting the Prompt with Special Escape Sequences	769
The Secondary Prompt	770

13.2.4	The Search Path	771
13.2.5	The hash Command	772
13.2.6	The source or dot Command	773
13.3	The Command Line	774
13.3.1	The Order of Processing Commands	774
13.3.2	Built-In Commands and the help Command	775
13.3.3	Changing the Order of Command-Line Processing	776
13.3.4	The Exit Status	777
13.3.5	Multiple Commands at the Command Line	779
13.3.6	Command Grouping	779
13.3.7	Conditional Execution of Commands	779
13.3.8	Commands in the Background	780
13.4	Job Control	781
13.4.1	Job Control Commands and Options	781
	New jobs Options	783
	The disown Built-In	783
13.5	Command-Line Shortcuts	783
13.5.1	Command and Filename Completion	783
13.5.2	History	785
13.5.3	Accessing Commands from the History File	786
	The Arrow Keys	786
	The history Built-In Command	786
	The fc Command	787
	Re-executing History Commands (Bang! Bang!)	790
13.5.4	Command-Line Editing	793
	The vi Built-In Editor	794
	The emacs Built-In Editor	795
	FCEDIT and Editing Commands	796
13.6	Aliases	798
13.6.1	Listing Aliases	798
13.6.2	Creating Aliases	798
13.6.3	Deleting Aliases	799
13.7	Manipulating the Directory Stack	799
13.7.1	The dirs Built-In Command	799
13.7.2	The pushd and popd Commands	800
13.8	Metacharacters (Wildcards)	801
13.9	Filename Substitution (Globbing)	802
13.9.1	The Asterisk	802
13.9.2	The Question Mark	803
13.9.3	The Square Brackets	804
13.9.4	Brace Expansion	805
13.9.5	Escaping Metacharacters	806
13.9.6	Tilde and Hyphen Expansion	806
13.9.7	Controlling Wildcards (Globbing)	807
13.9.8	Extended Filename Globbing (bash 2.x)	808

13.10	Variables	810
13.10.1	Types of Variables	810
13.10.2	Naming Conventions	810
13.10.3	The declare Built-In	810
13.10.4	Local Variables and Scope	811
	Setting Local Variables	811
	Setting Read-Only Variables	813
13.10.5	Environment Variables	814
	Setting Environment Variables	818
13.10.6	Unsetting Variables	820
13.10.7	Printing the Values of Variables	820
	The echo Command	820
	The printf Command	821
13.10.8	Variable Expansion Modifiers (Parameter Expansion)	823
13.10.9	Variable Expansion of Substrings	827
13.10.10	Positional Parameters	828
13.10.11	Other Special Variables	830
13.11	Quoting	831
13.11.1	The Backslash	832
13.11.2	Single Quotes	833
13.11.3	Double Quotes	834
13.12	Command Substitution	835
13.13	Arithmetic Expansion	837
13.14	Order of Expansion	838
13.15	Arrays	838
13.16	Functions	841
13.16.1	Defining Functions	841
13.16.2	Listing and Unsetting Functions	844
13.17	Standard I/O and Redirection	844
13.17.1	The exec Command and Redirection	846
13.18	Pipes	849
13.18.1	The here document and Redirecting Input	851
13.19	Shell Invocation Options	852
13.19.1	The set Command and Options	853
13.19.2	The shopt Command and Options	855
13.20	Shell Built-In Commands	857
Lab 48:	bash Shell—Getting Started	860
Lab 49:	Job Control	860
Lab 50:	Command Completion, History, and Aliases	861
Lab 51:	Shell Metacharacters	861
Lab 52:	Redirection	862
Lab 53:	Variables	862

14 Programming the Bash Shell 865

- 14.1 Introduction 865
 - 14.1.1 The Steps in Creating a Shell Script 865
 - The First Line* 865
 - Comments* 866
 - Executable Statements and bash Shell Constructs* 866
 - Making the Script Executable* 866
 - A Scripting Session* 866
- 14.2 Reading User Input 867
 - 14.2.1 Variables (Review) 867
 - 14.2.2 The read Command 868
- 14.3 Arithmetic 871
 - 14.3.1 Integers (declare and let Commands) 871
 - The declare Command* 871
 - Listing Integers* 872
 - Representing and Using Different Bases* 872
 - The let Command* 873
 - 14.3.2 Floating-Point Arithmetic 874
- 14.4 Positional Parameters and Command-Line Arguments 874
 - 14.4.1 Positional Parameters 874
 - 14.4.2 The set Command and Positional Parameters 876
 - How \$* and \$@ Differ* 878
- 14.5 Conditional Constructs and Flow Control 879
 - 14.5.1 Exit Status 879
 - 14.5.2 The Built-In test and let Commands 880
 - The test Command with Single Brackets* 880
 - The test Command with Double Brackets* 880
 - The let Command and Arithmetic with Double Parentheses* 884
 - 14.5.3 The if Command 886
 - The exit Command and the ? Variable* 889
 - Checking for Null Values* 891
 - Nested if Commands* 891
 - 14.5.4 The if/else Command 891
 - 14.5.5 The if/elif/else Command 893
 - 14.5.6 File Testing 896
 - 14.5.7 The null Command 898
 - 14.5.8 The case Command 900
 - Creating Menus with the here document and case Command* 901
- 14.6 Looping Commands 903
 - 14.6.1 The for Command 903
 - 14.6.2 The \$* and \$@ Variables in Wordlists 906
 - 14.6.3 The while Command 907

14.6.4	The until Command	910
14.6.5	The select Command and Menus	912
14.6.6	Looping Control Commands	917
	The shift Command	917
	The break Command	919
	The continue Command	920
	Nested Loops and Loop Control	921
14.6.7	I/O Redirection and Subshells	923
	Redirecting the Output of a Loop to a File	923
	Piping the Output of a Loop to a UNIX/Linux Command	925
14.6.8	Running Loops in the Background	925
14.6.9	The IFS and Loops	926
14.7	Functions	927
14.7.1	Unsetting Functions	928
14.7.2	Exporting Functions	928
14.7.3	Function Arguments and the Return Value	929
	Arguments	929
	The Built-In local Function	929
	The Built-In return Function	929
14.7.4	Functions and the source (or dot) Command	932
	Storing Functions	932
14.8	Trapping Signals	935
14.8.1	Resetting Signals	937
14.8.2	Ignoring Signals	938
14.8.3	Listing Traps	938
14.8.4	Traps and Functions	940
14.9	Debugging	941
14.10	The Command Line	943
14.10.1	Processing Command-Line Options with getopt	943
	Sample getopt Scripts	944
	Special getopt Variables	947
14.10.2	The eval Command and Parsing the Command Line	949
14.11	bash Options	951
14.11.1	Shell Invocation Options	951
14.11.2	The set Command and Options	953
14.11.3	The shopt Command and Options	955
14.12	Shell Built-In Commands	957
Lab 54:	bash Shell—First Script	959
Lab 55:	Command-Line Arguments	960
Lab 56:	Getting User Input	960
Lab 57:	Conditional Statements	961
Lab 58:	Conditionals and File Testing	962
Lab 59:	The case Statement	962

Lab 60: Loops 964

Lab 61: Functions 965

15 Debugging Shell Scripts 967

15.1 Introduction 967

15.2 Style Issues 967

15.3 Types of Errors 968

15.3.1 Runtime Errors 969

15.3.2 Naming Conventions 969

15.3.3 Insufficient Permissions 970

15.3.4 Path Problems 970

15.3.5 The Shbang Line 972

15.3.6 Sneaky Aliases 973

15.4 Probable Causes for Syntax Errors 975

15.4.1 Undefined and Misspelled Variables 975

15.4.2 Incomplete Programming Statements 976

Missing Keywords 976

Indentation 976

if/endif Errors 977

case and switch Errors 978

Looping Errors 979

Operator Errors 981

Misusing an Operator 982

Quoting Errors 983

Quoting Metacharacters 983

What You Should Know About Quotes 985

Problems with the here document 989

File-Testing Errors 990

15.4.3 Common Error Messages from the Big 5 Shells 991

Common C/TC Shell Error Messages 992

Common Bourne Shell Error Messages 994

Common Korn Shell Error Messages 996

Common Bash Error Messages 999

15.4.4 Logic Errors and Robustness 1001

Logical Operator Errors 1001

Relational Operator Errors 1002

Branching Errors 1003

Exit Status with an if Conditional 1004

Lacking Robustness 1005

15.5 Tracing with Shell Options and the set Command 1009

15.5.1 Debugging Bourne Shell Scripts 1009

Bourne Shell Debugging Options 1009

15.5.2 Debugging C/TC Shell Scripts 1011

15.5.3 Debugging Korn Shell Scripts 1014

15.5.4	Debugging Bash Scripts	1016
	<i>Bash Invocation Options</i>	1017
	<i>The set Command and Options</i>	1018
	<i>The shopt Command and Options</i>	1020
15.6	Summary	1021

16 The System Administrator and the Shell 1023

16.1	Introduction	1023
16.2	The Superuser	1023
16.3	Becoming a Superuser with the su Command	1024
16.3.1	Running Scripts As Root	1027
	<i>Running a Script in the Current Directory</i>	1027
16.3.2	Scripts That Run As Root (setuid Programs)	1029
16.4	Boot Scripts	1030
16.4.1	A Little Terminology	1031
	<i>The Kernel and init</i>	1031
	<i>What Is a Run Level?</i>	1031
	<i>Single-User Mode</i>	1033
	<i>Boot Scripts</i>	1034
	<i>What Is a Daemon?</i>	1035
16.4.2	An Example Boot Script—The cron Utility	1035
	<i>Creating a cron Instruction File (crontab File)</i>	1035
	<i>The cron Boot Script</i>	1037
16.4.3	Writing a Portable Script	1041
16.4.4	User-Specific Initialization Files	1044
	<i>Possible Files for /etc/skel</i>	1045
16.4.5	System-Wide Initialization Files	1046
	<i>/etc/profile</i>	1047
	<i>/etc/bashrc</i>	1049
	<i>/etc/csh.login</i>	1050
	<i>/etc/csh.cshrc</i>	1051
16.5	Summary	1053

A Useful UNIX/Linux Utilities for Shell Programmers 1055

B Comparison of the Shells 1103

B.1	The Shells Compared	1103
B.2	tcsh versus csh	1104
B.3	bash versus sh	1104

Index 1113