

Table of Contents

Preface.....	xiii
1. Beginning bash.....	1
1.1 Why bash?	2
1.2 The bash Shell	3
1.3 Decoding the Prompt	4
1.4 Showing Where You Are	5
1.5 Finding and Running Commands	6
1.6 Getting Information About Files	8
1.7 Showing All Hidden (Dot) Files in the Current Directory	11
1.8 Using Shell Quoting	13
1.9 Using or Replacing Builtins and External Commands	15
1.10 Determining if You Are Running Interactively	17
1.11 Setting bash as Your Default Shell	18
1.12 Keeping bash Updated	20
1.13 Getting bash for Linux	21
1.14 Getting bash for xBSD	22
1.15 Getting bash for macOS	24
1.16 Getting bash for Unix	24
1.17 Getting bash for Windows	26
1.18 Getting bash Without Getting bash	29
1.19 Learning More About bash Documentation	30
2. Standard Output.....	33
2.1 Writing Output to the Terminal/Window	34
2.2 Writing Output but Preserving Spacing	35
2.3 Writing Output with More Formatting Control	36
2.4 Writing Output Without the Newline	38

2.5 Saving Output from a Command	39
2.6 Saving Output to Other Files	40
2.7 Saving Output from the ls Command	41
2.8 Sending Output and Error Messages to Different Files	42
2.9 Sending Output and Error Messages to the Same File	43
2.10 Appending Rather than Clobbering Output	45
2.11 Using Just the Beginning or End of a File	46
2.12 Skipping a Header in a File	46
2.13 Throwing Output Away	47
2.14 Saving or Grouping Output from Several Commands	48
2.15 Connecting Two Programs by Using Output as Input	50
2.16 Saving a Copy of Output Even While Using It as Input	51
2.17 Connecting Two Programs by Using Output as Arguments	53
2.18 Using Multiple Redirects on One Line	55
2.19 Saving Output When Redirect Doesn't Seem to Work	56
2.20 Swapping STDERR and STDOUT	58
2.21 Keeping Files Safe from Accidental Overwriting	59
2.22 Clobbering a File on Purpose	61
3. Standard Input.....	63
3.1 Getting Input from a File	63
3.2 Keeping Your Data with Your Script	64
3.3 Preventing Weird Behavior in a Here-Document	66
3.4 Indenting Here-Documents	67
3.5 Getting User Input	68
3.6 Getting Yes or No Input	70
3.7 Selecting from a List of Options	73
3.8 Prompting for a Password	74
4. Executing Commands.....	77
4.1 Running Any Executable	77
4.2 Running Several Commands in Sequence	80
4.3 Running Several Commands All at Once	81
4.4 Telling Whether a Command Succeeded or Not	82
4.5 Running a Command Only if Another Command Succeeded	84
4.6 Using Fewer if Statements	85
4.7 Running Long Jobs Unattended	87
4.8 Displaying Error Messages When Failures Occur	88
4.9 Running Commands from a Variable	89
4.10 Running All Scripts in a Directory	90

5. Basic Scripting: Shell Variables.....	93
5.1 Documenting Your Script	95
5.2 Embedding Documentation in Shell Scripts	96
5.3 Promoting Script Readability	98
5.4 Separating Variable Names from Surrounding Text	100
5.5 Exporting Variables	101
5.6 Seeing All Variable Values	103
5.7 Using Parameters in a Shell Script	104
5.8 Looping Over Arguments Passed to a Script	106
5.9 Handling Parameters with Spaces	107
5.10 Handling Lists of Parameters with Spaces	109
5.11 Counting Arguments	111
5.12 Consuming Arguments	113
5.13 Getting Default Values	115
5.14 Setting Default Values	116
5.15 Using null as a Valid Default Value	117
5.16 Using More than Just a Constant String for Default	118
5.17 Giving an Error Message for Unset Parameters	120
5.18 Changing Pieces of a String	122
5.19 Getting the Absolute Value of a Number	124
5.20 Using bash for basename	125
5.21 Using bash for dirname	126
5.22 Using Alternate Values for Comma Separated Values	127
5.23 Using Array Variables	128
5.24 Converting Between Upper- and Lowercase	129
5.25 Converting to Camel Case	130
6. Shell Logic and Arithmetic.....	133
6.1 Doing Arithmetic in Your Shell Script	133
6.2 Branching on Conditions	136
6.3 Testing for File Characteristics	140
6.4 Testing for More than One Thing	143
6.5 Testing for String Characteristics	144
6.6 Testing for Equality	145
6.7 Testing with Pattern Matches	147
6.8 Testing with Regular Expressions	149
6.9 Changing Behavior with Redirections	152
6.10 Looping for a While	153
6.11 Looping with a read	155
6.12 Looping with a Count	157
6.13 Looping with Floating-Point Values	159
6.14 Branching Many Ways	160

6.15 Parsing Command-Line Arguments	162
6.16 Creating Simple Menus	165
6.17 Changing the Prompt on Simple Menus	167
6.18 Creating a Simple RPN Calculator	168
6.19 Creating a Command-Line Calculator	171
7. Intermediate Shell Tools I	175
7.1 Sifting Through Files for a String	176
7.2 Getting Just the Filename from a Search	178
7.3 Getting a Simple True/False from a Search	179
7.4 Searching for Text While Ignoring Case	180
7.5 Doing a Search in a Pipeline	181
7.6 Paring Down What the Search Finds	182
7.7 Searching with More Complex Patterns	184
7.8 Searching for an SSN	185
7.9 Grepping Compressed Files	186
7.10 Keeping Some Output, Discarding the Rest	187
7.11 Keeping Only a Portion of a Line of Output	188
7.12 Reversing the Words on Each Line	189
7.13 Summing a List of Numbers	191
7.14 Counting String Values with awk	192
7.15 Counting String Values with bash	194
7.16 Showing Data as a Quick and Easy Histogram	196
7.17 An Easy Histogram with bash	198
7.18 Showing a Paragraph of Text After a Found Phrase	199
8. Intermediate Shell Tools II	201
8.1 Sorting Your Output	201
8.2 Sorting Numbers	202
8.3 Sorting IP Addresses	204
8.4 Cutting Out Parts of Your Output	206
8.5 Removing Duplicate Lines	208
8.6 Compressing Files	209
8.7 Uncompressing Files	211
8.8 Checking a tar Archive for Unique Directories	213
8.9 Translating Characters	214
8.10 Converting Uppercase to Lowercase	215
8.11 Converting DOS Files to Linux Format	216
8.12 Removing Smart Quotes	217
8.13 Counting Lines, Words, or Characters in a File	218
8.14 Rewrapping Paragraphs	219
8.15 Doing More with less	220

9. Finding Files: find, locate, slocate.....	223
9.1 Finding All Your MP3 Files	224
9.2 Handling Filenames Containing Odd Characters	225
9.3 Speeding Up Operations on Found Files	227
9.4 Finding Files Across Symbolic Links	227
9.5 Finding Files Irrespective of Case	228
9.6 Finding Files by Date	229
9.7 Finding Files by Type	230
9.8 Finding Files by Size	231
9.9 Finding Files by Content	232
9.10 Finding Existing Files and Content Fast	233
9.11 Finding a File Using a List of Possible Locations	234
10. Additional Features for Scripting.....	239
10.1 “Daemon-izing” Your Script	239
10.2 Reusing Code with Includes and Sourcing	240
10.3 Using Configuration Files in a Script	242
10.4 Defining Functions	243
10.5 Using Functions: Parameters and Return Values	245
10.6 Trapping Interrupts	248
10.7 Redefining Commands with alias	252
10.8 Avoiding Aliases and Functions	254
10.9 Counting Elapsed Time	256
10.10 Writing Wrappers	257
11. Working with Dates and Times.....	263
11.1 Formatting Dates for Display	264
11.2 Supplying a Default Date	265
11.3 Automating Date Ranges	267
11.4 Converting Dates and Times to Epoch Seconds	270
11.5 Converting Epoch Seconds to Dates and Times	271
11.6 Getting Yesterday or Tomorrow with Perl	272
11.7 Figuring Out Date and Time Arithmetic	273
11.8 Handling Time Zones, Daylight Saving Time, and Leap Years	275
11.9 Using date and cron to Run a Script on the Nth Day	276
11.10 Logging with Dates	278
12. End-User Tasks as Shell Scripts.....	281
12.1 Starting Simple by Printing Dashes	281
12.2 Viewing Photos in an Album	284
12.3 Loading Your MP3 Player	290
12.4 Burning a CD	295

12.5 Comparing Two Documents	298
13. Parsing and Similar Tasks	303
13.1 Parsing Arguments for Your Shell Script	303
13.2 Parsing Arguments with Your Own Error Messages	307
13.3 Parsing Some HTML	309
13.4 Parsing Output into an Array	310
13.5 Parsing Output with a Function Call	312
13.6 Parsing Text with a read Statement	314
13.7 Parsing with read into an Array	315
13.8 Reading an Entire File	315
13.9 Getting Your Plurals Right	317
13.10 Taking It One Character at a Time	319
13.11 Cleaning Up an SVN Source Tree	320
13.12 Setting Up a Database with MySQL	321
13.13 Isolating Specific Fields in Data	323
13.14 Updating Specific Fields in Datafiles	326
13.15 Trimming Whitespace	327
13.16 Compressing Whitespace	331
13.17 Processing Fixed-Length Records	333
13.18 Processing Files with No Line Breaks	335
13.19 Converting a Datafile to CSV	337
13.20 Parsing a CSV Datafile	338
14. Writing Secure Shell Scripts	341
14.1 Avoiding Common Security Problems	343
14.2 Avoiding Interpreter Spoofing	344
14.3 Setting a Secure \$PATH	345
14.4 Clearing All Aliases	347
14.5 Clearing the Command Hash	348
14.6 Preventing Core Dumps	349
14.7 Setting a Secure \$IFS	350
14.8 Setting a Secure umask	350
14.9 Finding World-Writable Directories in Your \$PATH	352
14.10 Adding the Current Directory to the \$PATH	354
14.11 Using Secure Temporary Files	355
14.12 Validating Input	360
14.13 Setting Permissions	362
14.14 Leaking Passwords into the Process List	364
14.15 Writing setuid or setgid Scripts	365
14.16 Restricting Guest Users	366
14.17 Using chroot Jails	368

14.18 Running as a Non-root User	370
14.19 Using sudo More Securely	371
14.20 Using Passwords in Scripts	372
14.21 Using SSH Without a Password	374
14.22 Restricting SSH Commands	383
14.23 Disconnecting Inactive Sessions	386
15. Advanced Scripting.....	387
15.1 Finding bash Portably for #!	388
15.2 Setting a POSIX \$PATH	390
15.3 Developing Portable Shell Scripts	391
15.4 Testing Scripts Using Virtual Machines	393
15.5 Using for Loops Portably	396
15.6 Using echo Portably	397
15.7 Splitting Output Only When Necessary	400
15.8 Viewing Output in Hex	402
15.9 Using bash Net-Redirection	403
15.10 Finding My IP Address	405
15.11 Getting Input from Another Machine	410
15.12 Redirecting Output for the Life of a Script	411
15.13 Working Around “Argument list too long” Errors	412
15.14 Logging to syslog from Your Script	414
15.15 Using logger Correctly	415
15.16 Sending Email from Your Script	416
15.17 Automating a Process Using Phases	421
15.18 Doing Two Things at Once	424
15.19 Running an SSH command on multiple hosts	426
16. Configuring and Customizing bash.....	429
16.1 bash Startup Options	430
16.2 Customizing Your Prompt	430
16.3 A Prompt Before Your Program Runs	439
16.4 Changing Your \$PATH Permanently	441
16.5 Changing Your \$PATH Temporarily	442
16.6 Setting Your \$CDPATH	447
16.7 When Programs Are Not Found	449
16.8 Shortening or Changing Command Names	450
16.9 Adjusting Shell Behavior and Environment	452
16.10 Adjusting readline Behavior Using .inputrc	453
16.11 Keeping a Private Stash of Utilities by Adding ~/bin	455
16.12 Using Secondary Prompts: \$PS2, \$PS3, \$PS4	456
16.13 Synchronizing Shell History Between Sessions	458

16.14 Setting Shell History Options	459
16.15 Creating a Better cd Command	461
16.16 Creating and Changing Into a New Directory in One Step	463
16.17 Getting to the Bottom of Things	465
16.18 Adding New Features to bash Using Loadable Builtins	466
16.19 Improving Programmable Completion	471
16.20 Using Initialization Files Correctly	477
16.21 Creating Self-Contained, Portable rc Files	480
16.22 Getting Started with a Custom Configuration	483
17. Housekeeping and Administrative Tasks.....	499
17.1 Renaming Many Files	499
17.2 Using GNU Texinfo and info on Linux	501
17.3 Unzipping Many ZIP Files	502
17.4 Recovering Disconnected Sessions Using screen	503
17.5 Sharing a Single bash Session	506
17.6 Logging an Entire Session or Batch Job	507
17.7 Clearing the Screen When You Log Out	508
17.8 Capturing File Metadata for Recovery	510
17.9 Creating an Index of Many Files	511
17.10 Using diff and patch	512
17.11 Counting Differences in Files	516
17.12 Removing or Renaming Files Named with Special Characters	518
17.13 Prepending Data to a File	519
17.14 Editing a File in Place	522
17.15 Using sudo on a Group of Commands	524
17.16 Finding Lines That Appear in One File but Not in Another	526
17.17 Keeping the Most Recent N Objects	529
17.18 Writing to a Circular Log	533
17.19 Circular Backups	535
17.20 Grepping ps Output Without Also Getting the grep Process Itself	538
17.21 Finding Out Whether a Process Is Running	539
17.22 Adding a Prefix or Suffix to Output	541
17.23 Numbering Lines	542
17.24 Writing Sequences	545
17.25 Emulating the DOS Pause Command	547
17.26 Commifying Numbers	548
18. Working Faster by Typing Less.....	551
18.1 Moving Quickly Among Arbitrary Directories	552
18.2 Repeating the Last Command	554
18.3 Running Almost the Same Command	555

18.4 Quick Substitution	556
18.5 Reusing Arguments	557
18.6 Finishing Names for You	558
18.7 Playing It Safe	559
18.8 Big Changes, More Lines	561
19. Tips and Traps: Common Goofs for Novices.	563
19.1 Forgetting to Set Execute Permissions	563
19.2 Fixing “No such file or directory” Errors	564
19.3 Forgetting That the Current Directory Is Not in the \$PATH	566
19.4 Naming Your Script “test”	567
19.5 Expecting to Change Exported Variables	568
19.6 Forgetting Quotes Leads to “command not found” on Assignments	570
19.7 Forgetting that Pattern Matching Alphabetizes	571
19.8 Forgetting that Pipelines Make Subshells	572
19.9 Making Your Terminal Sane Again	575
19.10 Deleting Files Using an Empty Variable	576
19.11 Seeing Odd Behavior from printf	577
19.12 Testing bash Script Syntax	579
19.13 Debugging Scripts	580
19.14 Avoiding “command not found” When Using Functions	582
19.15 Confusing Shell Wildcards and Regular Expressions	583
A. Reference Lists.	585
B. Examples Included with bash.	621
C. Command-Line Processing.	631
D. Revision Control.	637
E. Building bash from Source.	667
Index.	673