

Beginning Portable Shell Scripting

From Novice to Professional



Peter Seebach

Beginning Portable Shell Scripting: From Novice to Professional

Copyright © 2008 by Peter Seebach

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-1043-6

ISBN-10 (pbk): 1-4302-1043-5

ISBN-13 (electronic): 978-1-4302-1044-3

ISBN-10 (electronic): 1-4302-1044-3

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Frank Pohlmann

Technical Reviewer: Gary V. Vaughan

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Richard Dal Porto

Copy Editor: Kim Benbow

Associate Production Director: Kari Brooks-Copony

Production Editor: Katie Stence

Compositor: Linda Weidemann, Wolf Creek Press

Proofreader: Dan Shaw

Indexer: Broccoli Information Management

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>. You may need to answer questions pertaining to this book in order to successfully download the code.

For Nora, who believed.

Contents at a Glance

| | |
|---|------|
| About the Author | xv |
| About the Technical Reviewer | xvii |
| Acknowledgments | xix |
| CHAPTER 1 Introduction to Shell Scripting | 1 |
| CHAPTER 2 Patterns and Regular Expressions | 21 |
| CHAPTER 3 Basic Shell Scripting | 43 |
| CHAPTER 4 Core Shell Features Explained | 69 |
| CHAPTER 5 Shells Within Shells | 89 |
| CHAPTER 6 Invocation and Execution | 117 |
| CHAPTER 7 Shell Language Portability | 145 |
| CHAPTER 8 Utility Portability | 175 |
| CHAPTER 9 Bringing It All Together | 205 |
| CHAPTER 10 Shell Script Design | 219 |
| CHAPTER 11 Mixing and Matching | 237 |
| APPENDIX A The Shell Command Language | 255 |
| APPENDIX B The <i>sh</i> Utility | 297 |
| APPENDIX C Regular Expressions | 319 |
| INDEX | 339 |

Contents

| | |
|---|-----------|
| About the Author | xv |
| About the Technical Reviewer | xvii |
| Acknowledgments | xix |
| CHAPTER 1 Introduction to Shell Scripting | 1 |
| About This Book | 1 |
| Conventions | 2 |
| What Shell Scripting Is | 3 |
| What Shell Scripting Isn't | 5 |
| Why Shell? | 6 |
| The Bourne Shell Family | 7 |
| Why Portable? | 8 |
| Why Not? | 9 |
| Beyond Portability: Cleanliness and Good Living | 10 |
| What's in This Book | 11 |
| Introducing the Shell | 11 |
| Interactive and Noninteractive Usage | 11 |
| Simple Commands | 12 |
| Introducing Variables | 14 |
| Introducing Quoting | 15 |
| The printf Command | 18 |
| What's Next? | 20 |
| CHAPTER 2 Patterns and Regular Expressions | 21 |
| Shell Patterns | 21 |
| Pattern-Matching Basics | 22 |
| Character Classes | 24 |
| Using Shell Patterns | 26 |
| Pathname Expansion | 27 |
| Differences from Shell Patterns | 27 |
| Using Globs | 30 |

| | |
|--|-----------|
| Regular Expressions | 31 |
| Basic Regular Expressions | 32 |
| Backreferences | 33 |
| Extended Regular Expressions. | 34 |
| Common Extensions | 36 |
| Replacements | 36 |
| Using Regular Expressions | 37 |
| Replacing Patterns with Regular Expressions. | 39 |
| Common Pitfalls of Regular Expressions | 40 |
| What's Next? | 41 |
| | |
| CHAPTER 3 Basic Shell Scripting | 43 |
| Introducing Control Structures | 43 |
| What Is Truth? | 44 |
| Introducing Conditional Execution. | 47 |
| Introducing Iteration. | 52 |
| Thinking About Control Structures | 55 |
| Introducing Redirection. | 56 |
| Understanding File Descriptors | 57 |
| Redirection Using <code>exec</code> | 61 |
| Introducing Here Documents | 64 |
| Redirection and Loops | 66 |
| What's Next? | 68 |
| | |
| CHAPTER 4 Core Shell Features Explained | 69 |
| Parsing. | 70 |
| Tokens | 70 |
| Words and Keywords. | 72 |
| Command Lists. | 73 |
| Shell Quoting | 77 |
| Escaping Characters with a Backslash | 77 |
| Escaping Characters with Single Quotes | 79 |
| Escaping Characters with Double Quotes | 79 |
| Quoting Examples | 79 |

| | |
|--|------------|
| Substitution and Expansion | 80 |
| Substitution and Field Splitting | 80 |
| Understanding Parameter Substitution | 81 |
| Tilde Expansion | 85 |
| Globbing | 86 |
| What's Next? | 87 |
| | |
| CHAPTER 5 Shells Within Shells | 89 |
| Understanding Processes | 89 |
| Variables and the Environment | 90 |
| Manipulating the Environment | 91 |
| Temporary Changes | 92 |
| Exploring Subshells | 94 |
| Subshells and External Shells | 94 |
| Command Substitution | 95 |
| Implicit and Explicit Subshells | 97 |
| Modifying the State of the Shell | 99 |
| Shell Builtins | 99 |
| Shell Functions | 99 |
| The eval Command | 102 |
| The dot (.) Command | 108 |
| Using Shells Within Shells | 109 |
| When to Use an External Shell | 109 |
| When to Use eval or dot (.) | 111 |
| When to Use Subshells | 111 |
| When to Use Command Substitution | 112 |
| Combinations | 113 |
| What's Next? | 115 |
| | |
| CHAPTER 6 Invocation and Execution | 117 |
| Shell Invocation | 117 |
| How UNIX Runs Scripts | 117 |
| Shell Options | 119 |
| Using Positional Parameters | 120 |
| Manipulating Parameters for Fun and Profit | 122 |
| Shell Startup and Interactive Sessions | 128 |

| | |
|---|------------|
| Execution | 129 |
| More on Jobs and Tasks | 129 |
| Understanding Runtime Behavior | 137 |
| Debugging Tools | 142 |
| What's Next? | 143 |
| | |
| CHAPTER 7 Shell Language Portability | 145 |
| More on Portability | 145 |
| Standardization | 146 |
| Bugs | 147 |
| Portability Issues: Welcome to the Club | 147 |
| Common Extensions and Omissions | 148 |
| Other Kinds of Expansion and Substitution | 148 |
| Syntax Extensions | 154 |
| Common Omissions | 158 |
| Common Shells and Their Features | 158 |
| Almquist Shell | 158 |
| Bourne-Again Shell | 159 |
| Debian Almquist Shell | 160 |
| Korn Shell | 161 |
| Public Domain Korn Shell | 162 |
| Solaris /usr/xpg4/bin/sh | 163 |
| SVR2 Bourne Shell | 164 |
| SVR4 Bourne Shell | 164 |
| Traditional Shell | 165 |
| Z Shell | 166 |
| Execution Preambles | 167 |
| Setting Options and Variables | 168 |
| Picking a Better Shell | 169 |
| Self-Modifying Code | 170 |
| Emulating Features | 172 |
| What's Next? | 174 |

| | | |
|------------------|---|-----|
| CHAPTER 8 | Utility Portability | 175 |
| | Common Variations | 175 |
| | Days of Yore: System V and BSD | 176 |
| | GNU Arrives | 177 |
| | Standardization | 178 |
| | busybox | 178 |
| | Shell Builtins | 179 |
| | Avoiding Unnecessary Dependencies | 180 |
| | Relying on Extensions Considered Harmful | 181 |
| | Test with More Than One Shell on More Than One System | 182 |
| | Document Your Assumptions | 182 |
| | Common Utility Issues | 183 |
| | Public Enemy #1: echo | 183 |
| | Multiple Different Versions | 184 |
| | Archive Utilities | 185 |
| | Block Sizes | 188 |
| | Other Common Problems | 188 |
| | What to Do When Something Is Unavailable | 200 |
| | Roll Your Own | 200 |
| | Add a Missing Utility | 201 |
| | Use Something Else | 201 |
| | Demand a Real System | 201 |
| | A Few Examples | 202 |
| | What's Next? | 203 |
| | | |
| CHAPTER 9 | Bringing It All Together | 205 |
| | Robustness | 205 |
| | Handling Failure | 205 |
| | Temporary Files and Cleanup | 208 |
| | Handling Interrupts | 210 |
| | Startup Files and Environment Variables | 211 |
| | Documentation and Comments | 212 |
| | Degrade Gracefully | 214 |
| | Specify, and Test For, Requirements | 216 |
| | Scripts That Write Scripts | 216 |
| | Building a Script for a Specific Target | 217 |
| | Mixing with Other Languages | 218 |
| | What's Next? | 218 |

| | | |
|-------------------|--|-----|
| CHAPTER 10 | Shell Script Design | 219 |
| | Do One Thing Well | 219 |
| | Separate Functionality | 220 |
| | Isolate Dependencies | 220 |
| | Be Cooperative | 221 |
| | Filters, File Manipulation, and Program Manipulation | 221 |
| | Command-Line Options and Arguments | 224 |
| | Designing Options | 226 |
| | Options and Inputs | 228 |
| | Set Reasonable Limits | 229 |
| | Define Your Functional Scope | 230 |
| | Define Your Target Scope | 230 |
| | Case Study: pids | 232 |
| | What's Next? | 235 |
| | | |
| CHAPTER 11 | Mixing and Matching | 237 |
| | Mixing Quoting Rules | 237 |
| | Embedding Shell Scripts in Code | 239 |
| | Shell and make | 239 |
| | Shell and C | 241 |
| | Embedding Code in Shell Scripts | 242 |
| | Shell and sed | 242 |
| | Shell and awk | 246 |
| | Utilities and Languages | 252 |
| | What's Next? | 253 |
| | | |
| APPENDIX A | The Shell Command Language | 255 |
| | 2. Shell Command Language | 255 |
| | 2.1 Shell Introduction | 255 |
| | 2.2 Quoting | 256 |
| | 2.3 Token Recognition | 258 |
| | 2.4 Reserved Words | 260 |
| | 2.5 Parameters and Variables | 260 |
| | 2.6 Word Expansions | 264 |
| | 2.7 Redirection | 271 |
| | 2.8 Exit Status and Errors | 275 |
| | 2.9 Shell Commands | 276 |

| | |
|---|------------|
| 2.10 Shell Grammar | 285 |
| 2.11 Signals and Error Handling | 292 |
| 2.12 Shell Execution Environment | 293 |
| 2.13 Pattern Matching Notation | 294 |
| 2.14 Special Built-In Utilities | 296 |
| APPENDIX B The <i>sh</i> Utility | 297 |
| Name | 297 |
| Synopsis | 297 |
| Description | 297 |
| Options | 298 |
| Operands | 298 |
| Stdin | 299 |
| Input Files | 300 |
| Environment Variables | 300 |
| Asynchronous Events | 303 |
| Stdout | 303 |
| Stderr | 303 |
| Output Files | 303 |
| Extended Description | 303 |
| Command History List | 303 |
| Command Line Editing | 303 |
| Command Line Editing (vi-mode) | 304 |
| vi Line Editing Insert Mode | 304 |
| vi Line Editing Command Mode | 305 |
| Exit Status | 314 |
| Consequences Of Errors | 314 |
| Application Usage | 314 |
| Rationale | 316 |
| Future Directions | 318 |
| See Also | 318 |
| Change History | 318 |
| Issue 5 | 318 |
| Issue 6 | 318 |

| | | |
|--------------------|---|-----|
| APPENDIX C | Regular Expressions | 319 |
| | 9. Regular Expressions | 319 |
| | 9.1 Regular Expression Definitions | 320 |
| | 9.2 Regular Expression General Requirements | 321 |
| | 9.3 Basic Regular Expressions | 322 |
| | 9.4 Extended Regular Expressions | 327 |
| | 9.5 Regular Expression Grammar | 330 |
| INDEX | | 339 |

About the Author

■ **PETER SEEBACH** is a programmer who writes or, possibly, a writer who programs. He enjoys writing on topics from C standardization to operating system internals; he programs in C, Ruby, Lua, and shell by preference and several other languages when absolutely necessary. He lives in Northfield, Minnesota, and owns cats (who cannot program and do not write).

About the Technical Reviewer

■ **GARY V. VAUGHAN**, in his own words:

```
: ${This='sed -n'} #ightly:  
$This 1s,^,not\ ,p<<rose  
obfuscated!  
With red pen wielded  
every page will be a  
rose
```

```
: A perpetual traveler with no  
time `for sight_seeing in earnest. I  
do sleep ${when-1} # am  
done`; juggling=100 pet=projects...
```

```
# but somehow,  
in_the_end () { I=always return "to mending libtool"; }
```

(You get extra credit if you can predict what this would do if you ran it.)

Acknowledgments

The idea for this book came from Frank Pohlmann, who also edited it and provided a great deal of guidance in making sense of my disjointed ramblings on the topic. Gary V. Vaughan's technical advice and broad experience were invaluable throughout. I am particularly indebted to Sven Mascheck's excellent pages of information about historical shells, as well as the guide to shell portability included in the autoconf documentation. Many other developers contributed tidbits, interesting trivia, or feedback on proposed code; there are too many to list, I'm afraid. This book (and everything else I do) would not have been possible without support from my beloved spouse, Jesse.

