CONTENTS

FOREWORD
by Greg Wilson xv

PREFACE xvii

1 A REGULAR EXPRESSION MATCHER
by Brian Kernighan 1
The Practice of Programming 2
Implementation 3
Discussion 4
Alternatives 5
Building on It 6
Conclusion 8

2 SUBVERSION'S DELTA EDITOR: INTERFACE AS ONTOLOGY
by Karl Foigel 11
Version Control and Tree Transformation 12
Expressing Tree Differences 16
The Delta Editor Interface 17
But Is It Art? 23
Abstraction As a Spectator Sport 25
Conclusions 27

3 THE MOST BEAUTIFUL CODE I NEVER WROTE
by Jon Bentley 29
The Most Beautiful Code I Ever Wrote 30
More and More with Less and Less 31
Perspective 36
What Is Writing? 38
Conclusion 39
Acknowledgments 40

4 FINDING THINGS
by Tim Bray 41
On Time 41
Problem: Weblog Data 42
Problem: Who Fetched What, When? 50
Search in the Large 55
Conclusion 57
Functions
Array and Object Literals
Things to Do and Think About

10 THE QUEST FOR AN ACCELERATED POPULATION COUNT
by Henry S. Warren, Jr.

Basic Methods
Divide and Conquer
Other Methods
Sum and Difference of Population Counts of Two Words
Comparing the Population Counts of Two Words
Counting the 1-Bits in an Array
Applications

11 SECURE COMMUNICATION: THE TECHNOLOGY OF FREEDOM
by Ashish Gulhati

The Heart of the Start
Untangling the Complexity of Secure Messaging
Usability Is the Key
The Foundation
The Test Suite
The Functioning Prototype
Clean Up, Plug In, Rock On...
Hacking in the Himalayas
The Invisible Hand Moves
Speed Does Matter
Communications Privacy for Individual Rights
Hacking the Civilization

12 GROWING BEAUTIFUL CODE IN BIOPERL
by Lincoln Stein

BioPerl and the Bio::Graphics Module
The Bio::Graphics Design Process
Extending Bio::Graphics
Conclusions and Lessons Learned

13 THE DESIGN OF THE GENE SORTER
by Jim Kent

The User Interface of the Gene Sorter
Maintaining a Dialog with the User over the Web
A Little Polymorphism Can Go a Long Way
Filtering Down to Just the Relevant Genes
Theory of Beautiful Code in the Large
Conclusion
14 HOW ELEGANT CODE EVOLVES WITH HARDWARE:
THE CASE OF GAUSSIAN ELIMINATION
by Jack Dongarra and Piotr Luszczek
The Effects of Computer Architectures on Matrix Algorithms
A Decompositional Approach
A Simple Version
LINPACK's DGEFA Subroutine
LAPACK DGETRF
Recursive LU
ScaLAPACK PDGETRF
Multithreading for Multi-Core Systems
A Word About the Error Analysis and Operation Count
Future Directions for Research
Further Reading

15 THE LONG-TERM BENEFITS OF BEAUTIFUL DESIGN
by Adam Kolawa
My Idea of Beautiful Code
Introducing the CERN Library
Outer Beauty
Inner Beauty
Conclusion

16 THE LINUX KERNEL DRIVER MODEL: THE BENEFITS
OF WORKING TOGETHER
by Greg Kroah-Hartman
Humble Beginnings
Reduced to Even Smaller Bits
Scaling Up to Thousands of Devices
Small Objects Loosely Joined

17 ANOTHER LEVEL OF INDIRECTION
by Diomidis Spinellis
From Code to Pointers
From Function Arguments to Argument Pointers
From Filesystems to Filesystem Layers
From Code to a Domain-Specific Language
Multiplexing and Demultiplexing
Layers Forever?
18 PYTHON'S DICTIONARY IMPLEMENTATION: BEING ALL THINGS TO ALL PEOPLE
by Andrew Kuchling
Inside the Dictionary 295
Special Accommodations 296
Collisions 298
Resizing 299
Iterations and Dynamic Changes 300
Conclusion 301
Acknowledgments 301

19 MULTIDIMENSIONAL ITERATORS IN NUMPY
by Travis E. Oliphant
Key Challenges in N-Dimensional Array Operations 304
Memory Models for an N-Dimensional Array 305
NumPy Iterator Origins 307
Iterator Design 307
Iterator Interface 313
Iterator Use 314
Conclusion 318

20 A HIGHLY RELIABLE ENTERPRISE SYSTEM FOR NASA'S MARS ROVER MISSION
by Ronald Mak
The Mission and the Collaborative Information Portal 320
Mission Needs 321
System Architecture 322
Case Study: The Streamer Service 325
Reliability 328
Robustness 336
Conclusion 338

21 ERP5: DESIGNING FOR MAXIMUM ADAPTABILITY
by Rogerio Atem de Carvalho and Rafael Monnerat
General Goals of ERP 340
ERP5 340
The Underlying Zope Platform 342
ERP5 Project Concepts 346
Coding the ERP5 Project 347
Conclusion 351
<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>A SPOONFUL OF SEWAGE</td>
<td>by Bryan Cantrill</td>
</tr>
<tr>
<td>23</td>
<td>DISTRIBUTED PROGRAMMING WITH MAPREDUCE</td>
<td>by Jeffrey Dean and Sanjay Ghemawat</td>
</tr>
<tr>
<td></td>
<td>A Motivating Example</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The MapReduce Programming Model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other MapReduce Examples</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A Distributed MapReduce Implementation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extensions to the Model</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Further Reading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acknowledgments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Appendix: Word Count Solution</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>BEAUTIFUL CONCURRENCY</td>
<td>by Simon Peyton Jones</td>
</tr>
<tr>
<td></td>
<td>A Simple Example: Bank Accounts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software Transactional Memory</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The Santa Claus Problem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reflections on Haskell</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acknowledgments</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>SYNTACTIC ABSTRACTION: THE SYNTAX-CASE EXPANDER</td>
<td>by R. Kent Dybuij</td>
</tr>
<tr>
<td></td>
<td>Brief Introduction to syntax-case</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expansion Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>LABOR-SAVING ARCHITECTURE: AN OBJECT-ORIENTED FRAMEWORK FOR NETWORKED SOFTWARE</td>
<td>by William R. Otte and Douglas C. Schmidt</td>
</tr>
<tr>
<td></td>
<td>Sample Application: Logging Service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Object-Oriented Design of the Logging Server Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementing Sequential Logging Servers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implementing Concurrent Logging Servers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>INTEGRATING BUSINESS PARTNERS THE RESTFUL WAY</td>
<td>by Andrew Patzer</td>
</tr>
<tr>
<td></td>
<td>Project Background</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exposing Services to External Clients</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Routing the Service Using the Factory Pattern</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exchanging Data Using E-Business Protocols</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Conclusion</td>
<td></td>
</tr>
</tbody>
</table>