

A close-up, front-facing view of a motorcycle helmet with a clear visor, set against a background of bare trees.

MARK DALRYMPLE, AARON HILLEGASS & JEREMY SHERMAN

**ADVANCED MAC OS X PROGRAMMING**  
THE BIG NERD RANCH GUIDE



# **ADVANCED MAC OS X PROGRAMMING**

## **THE BIG NERD RANCH GUIDE**

**MARK DALRYMPLE**

by Mark Dalrymple

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission

The 10-gallon hat with propeller logo is a trademark of Big Nerd Ranch, Inc.

The authors and publisher have taken care in writing and printing this book but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

# Dedication

For Zoe. May she grow up to be as geeky as her Weird Uncle Bork.

*This page intentionally left blank*

# Acknowledgments

This book is based upon the experiences teaching a five-day class at The Big Nerd Ranch *Advanced Mac OS X Bootcamp*. The patience and curiosity of my students has made this a complete and comprehensible introduction to the plumbing that makes Mac OS X a robust and high-performance system.

Special thanks go to Jeremy Sherman. Jeremy stepped up and taught one of the Advanced Mac OS X classes when I was unable to so. Along the way he made numerous improvements to this book's course materials, helping to modernize and robusticize the code. Jeremy is also responsible for the excellent under-the-hoods look at Blocks.

Incredible thanks go to Aaron Hillegass, my co-author on the first two editions of this book. A few moons ago Aaron took a chance on this random guy sitting in the back of a Cocoa class. Over the years I've learned from Aaron more than I imagined possible about writing, teaching, and treating others well.

Susan Loper, the tireless editor, performed acts of magic on the purple prose she was given.

In preparing this book, many people reviewed the drafts and brought errors to our attention. The most astonishing quantity of corrections over the years came from Bill Monk. It would be impossible to overstate Bill's contributions. Other technical reviewers who submitted errors or offered suggestions include John Vink, Juan Pablo Claude, Carl-Johan Kihlborn, Mike Morton, Ajeya Vempati, Ericnie Ridd, Michael Simmons, David Southwick, Jeremy Wyld, Richard Wolf, Tom Van Leeuwen, Tomasz Zarzycki, James Spencer, Greg Miller, Anne Halsall, Roy Lovejoy, Jonathan Saggau, and Rob Rix. They made this book better with their useful corrections and suggestions. Any errors that remain in this book are completely my fault.

I would also like to thank my wife, my “Spousal Overunit” Charlotte DeVere, for her patience, support, and understanding.

*This page intentionally left blank*

# Table of Contents

Foreword .....	
Introduction .....	
Mac OS X: Built to Evolve .....	
This Book .....	
Typographical Conventions .....	
Online Materials .....	
1. C and Objective-C .....	
C .....	
The Compiler pipeline .....	
The C preprocessor .....	
Const and volatile variables .....	
Variable argument lists .....	
Bitwise operations .....	
Objective-C .....	
C callbacks in Objective-C .....	
Objective-C 2.0 .....	
Exercises .....	
2. The Compiler .....	
Handy Flags .....	
Debugging .....	
Warnings .....	
Seeing Preprocessor Output .....	
Seeing the Generated Assembly Code .....	
Compiler Optimization .....	
GCC Extensions .....	
Name Mangling .....	
Testing the compiler version .....	
The Optimizer .....	
Vectorization .....	
Even More Compiler Flags .....	
64-Bit Computing .....	
The 64-bit programming model .....	
New Objective-C runtime .....	
Universally Fat Binaries .....	
Fat binaries from the command line .....	
Fat binaries in Xcode .....	
Fat binary considerations .....	



When To Copy .....	
Blocks in Collections .....	
Block Retain Cycles .....	
New API Using Blocks .....	
For the More Curious: Blocks Internals .....	
Implementation .....	
Debugging .....	
Dumping runtime information .....	
Evolving the implementation .....	
Compiler-generated names .....	
Exercises .....	
4. Command-Line Programs .....	
The Basic Program .....	
Inside the Central Loop .....	
Changing Behavior By Name .....	
Looking at the Environment .....	
Parsing the Command Line .....	
getopt_long() .....	
User Defaults .....	
5. Exceptions, Error Handling, and Signals .....	
errno .....	
setjmp, longjmp .....	
Signals .....	
Handling a signal .....	
Blocking signals .....	
Signal issues .....	
Exception Handling in Cocoa .....	
Classic exception handling .....	
Native exception handling .....	
Subclassing NSApplication to catch exceptions .....	
64-bit Objective-C runtime .....	
NSError .....	
Logging .....	
syslog() .....	
ASL .....	
For the More Curious: Assertions .....	
Static assertions .....	
AssertMacros.h .....	
Exercises .....	

	dlopen() .....
	dlsym() .....
	BundlePrinter .....
	For the More Curious: libtool .....
	For the More Curious: otool .....
	For the More Curious: Runtime Environment Variables .....
	Exercises .....
7. Memory .....	
	Virtual Memory .....
	Program Memory Model .....
	Memory Lifetime .....
	Dynamic Memory Allocation .....
	malloc() .....
	free() .....
	realloc() .....
	calloc() .....
	alloca() .....
	Memory Ownership Issues .....
	Nodepools .....
	Debugging Memory Problems .....
	Common API issues .....
	Memory corruption .....
	Memory leaks .....
	Other Tools .....
	ps .....
	Resource limits .....
	Miscellaneous tools .....
	vm_stat .....
	Objective-C Garbage Collection .....
	How to use it .....
	How it works .....
	Strong and weak references .....
	Finalize methods .....
	Non Objective-C objects .....
	External reference counts .....
	The "new" collection classes .....
	GC and threads .....
	Debugging .....
	Exercises .....

8. Debugging With GDB .....	
What Is a Debugger? .....	
Using GDB from the Command Line .....	
A sample GDB session .....	
GDB Specifics .....	
Help .....	
Stack Traces .....	
Program Listings .....	
Breakpoints .....	