

Book Review: *FreeBSD Device Drivers: A Guide for the Intrepid* by Joseph Kong

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
	2012-06-25		WB

Contents

1	Introduction	1
2	Basics	1
3	Case Studies	1
4	Advanced Subjects	1
5	Random Thoughts	1
6	Conclusion	2

© 2012 Warren Block

Last updated 2012-06-25

Available in [HTML](#) or [PDF](#). Links to all my articles [here](#). Created with [AsciiDoc](#).

Low-level descriptions of writing FreeBSD device drivers are rare. This book is unique in that category.

Disclaimer: At the time I obtained it, this book was not yet for sale. Joseph Kong gave me a review copy at BSDCan 2012 with the stipulation that I would write a review of it. This is that review.

1 Introduction

Device drivers on FreeBSD seem mysterious, and the way to remove that mystery is to delve into the subject and see how they work. *FreeBSD Device Drivers* does just that. The emphasis is on examining existing drivers as examples; much of the book is source code, with callouts identifying sections of interest. It's a lot like having a very patient teacher explain how drivers work on FreeBSD.

2 Basics

The first five chapters are about the basics. Kernel modules, memory allocation, communicating with devices, locking, and sleeping. Examples start small and build up. Simplified device drivers are shown as examples. The writing is clear and uncomplicated.

3 Case Studies

Starting with chapter six, real drivers are examined to show actual usage of the concepts introduced in earlier chapters. First up is *nmdm(4)*, a virtual null modem driver.

Subsequent chapters introduce new subjects and then a follow-up case study looks at their actual use. *Newbus and Resource Allocation* and *Interrupt Handling* chapters are presented, and then a case study examines the *lpt(4)* parallel port driver.

Managing and Using Resources talks about I/O ports and memory, streams, and memory barriers, then there is a case study chapter on *ipmi(4)*, the "Intelligent Platform Management Interface" driver that allows monitoring and control of motherboard hardware.

4 Advanced Subjects

Chapters on DMA, storage drivers, and CAM are followed by chapters on the more complex drivers: a chapter on USB drivers, and two chapters on network drivers. While these last three chapters don't have separate case study chapters, they do use actual drivers as examples.

5 Random Thoughts

- The typography in the code listings in this book is pretty, and very readable. The callouts are a little weird, being on the left side of the code instead of the right, but it works.
- The Giant lock is mentioned, but not defined.
- If there is even one typo in this book, I have not found it, and typos usually jump out at me. For any book, that's surprising. For a technical book, it's amazing.
- There is a conclusion to chapter 17, but not an overall conclusion to the book. Who Did It is fairly obvious by the last chapter, but it would be nice to know for sure. A short conclusion could say "now that you know this, you could do these things with it, and here is where to get more information." There is a bibliography, which lists the references used throughout the book.

6 Conclusion

So now the main question: who needs this book? People interested in writing or modifying FreeBSD device drivers, certainly. Also those who want to understand how FreeBSD operates at the lower levels, or expand their skill set to drivers, or even just anyone with an interest in device drivers.

A good case can be made that application programmers will also benefit. Knowing what the drivers do can help guide the writing of higher-level code.

I like this book. It fills a definite need, and does it well.

FreeBSD Device Drivers: A Guide for the Intrepid by Joseph Kong, is published by [No Starch Press](#).