

# Book Reviews

MARK LAMOURINE AND RIK FARROW

## **Becoming a Better Programmer**

Pete Goodliffe

2014, O'Reilly Media Inc., 2014; 432 pages

ISBN 978-1-491-90553-1

*Reviewed by Mark Lamourine*

Peter Goodliffe subtitles his book "A handbook for people who care about code." It's obvious that he is passionate about writing with a clear and concise style. The book is a collection of tips and advice that I might give to new coders, and there's certainly nothing I would disagree with (although I looked for a section on suppressing the urge to cleverness and didn't find it).

He doesn't stop at code, and this is also good. He goes on to talk about the process and the personal side of a life of software development. The final section is more about interacting with the people who surround us than about any technical skills.

While I didn't find anything I disagree with, I also didn't find anything really actionable for me. This is a really good book to help someone who's just started coding get a perspective on the process as a whole. It's very easy for a junior person to focus on the language features and the rest of the tech. Goodliffe reminds them that there's more to the software development life than the editor and the compiler.

## **Ethernet: The Definitive Guide, 2nd ed.**

Charles E. Spurgeon and Joann Zimmerman

Copyright 2014 O'Reilly Media, 2014; 484 pages

ISBN 978-1-449-36184-6

*Reviewed by Mark Lamourine*

I'm actually not sure to whom I'd offer this book; part history, part technical reference, part deployment guide, it's hard to categorize. I guess I'll have to talk through it and see.

Someone just beginning to work with Ethernet, either physically or by programming network protocol interactions, would be interested in the first section. Here, the authors tell the story of the initial development and then the evolution of Ethernet specifications and implementations. They spend a fair amount of time on CSMA/CD, which, while interesting, is unlikely to be found in production these days (I hope). More common will be the full-duplex twisted pair that follows, and the auto-negotiation protocols. The authors also cover the evolution of Layer 2 signaling, which introduces and makes up an Ethernet frame on the wire.

The second section is definitely for the reader who expects to handle Ethernet cabling and interface devices. In this section

Spurgeon and Zimmerman detail each of the existing Ethernet specs, although only the IEEE specs are still relevant. Again, the history is a useful base for understanding the current state. They also talk about the 10BASE40G and 100G specs that are not yet in production and which certainly will be limited to datacenters or to short-haul links between datacenters in an organization.

In the third section, they talk about the issues you'd face if you were building out an office or cube space as well as the structured cabling and termination within a datacenter.

The authors move on to a treatment of the networking hardware that binds the physical and logical networks together and then close with a section on network troubleshooting concepts and tips.

I enjoyed reading this book and would recommend it to anyone who expected to start supporting a datacenter network or a large desktop network space. Although I'd be hesitant to hire someone to do this work for me if I knew their only source of knowledge was a chapter from a book, this is certainly a good guide for someone learning under supervision. It also contains good information for someone trying to evaluate a set of proposals.

In the end, I'd say this is a good general reference. It has relevance for people working in or with Ethernet networks at any scale. I don't do that kind of work anymore, but I might still keep this book handy.

## **The Book of PF, 3rd ed.**

Peter N. M. Hansteen

No Starch Press, 2014; 221 pages

ISBN 978-1-59327-589-1

*Reviewed by Rik Farrow*

PF is the packet filtering language used by OpenBSD, as well as FreeBSD, NetBSD, and Dragonfly BSD. I have the second edition of this book, but software and operating systems continue to evolve. There are now more shortcuts in the PF rules, including helpful ones, like including passing of packets in the same statement containing a redirect rule. ALTQ, the BSD traffic queueing system, has been replaced with a new traffic shaping system. Bridging has been added, allowing you to build firewalls with no IP addresses.

Hansteen has an easy-to-read style, and I can say his writing has improved over time. His explanations of example firewall rule sets are clearer than I recall from the second edition.

I've used PF for my home firewall for many years and plan to build a new firewall appliance (using an APU1D4), which will

also use PF. While some people might ask why not use Linux, PF has the advantages of a single rule set for both IPv4 and IPv6, the BSD license, and not having the same IP stack as the majority of servers and home routers. Diversity is good for security, and PF has a nicer syntax than Linux IP tables. Hansteen does a great job of explaining all you can do with PF.

### Alan Turing: The Enigma

Andrew Hodges

Princeton Press, 1983; 736 pages

ISBN 978-0-691-16472-4

*Reviewed by Rik Farrow*

I got a surprise in the mail while recovering from working at LISA14: a new, paperback edition of the biography of Alan Turing. I've never read biographies, but I found myself with some time on my hands when I couldn't read anything deeply technical, while also being curious about Alan Turing.

Hodges spends the first hundred pages moving Turing to the point of creating his seminal research paper, "On Computable Numbers." Just at that point, Hodges appeared to take a tremendous detour, by describing in detail some points of mathematical philosophy that Turing found important around 1938. I almost stopped reading, but after 10 pages, I discovered that Hilbert's conjectures were actually key to Turing's idea for what we today call a Turing machine.

Hodges actually adds a lot of context about the events surrounding Turing's short professional life. You've likely heard about Bletchley Park, bombs, and the German Enigma, and how important cracking German encrypted communications was to ending the war with Germany. Hodges makes these ideas concrete by explaining that England could not have survived without importing thousands of tons of goods each month, goods which had previously come from Europe. If U-boats had succeeded in sinking 50% of the cargo ships connecting America with England, England would not have had enough supplies to feed its people, much less continue fighting an air war and preparing to invade Europe. When decryption of German naval communications failed because of changes in the German system, U-boats had reached the level of starving out England. The tide of war hung on the success or failure of cryptanalysis.

While Turing did start out as an Oxford fellow, he quickly became interested in applied mathematics. He spent time at the Princeton Institute for Advanced Study, and he and von Neumann read each other's papers. Turing developed the idea of using tubes for memory and logic circuits, although others did the electronics design. And after the war, Turing helped to design the first real electronic computers.

Hodges doesn't skimp on the more troubling side of Turing. Turing was a homosexual in England at a time where just being

homosexual (not practicing) was illegal. Hodges deftly handles how difficult being attracted to men, and having to hide this, was for an intellectual who bristled at any untruths. Turing didn't suffer fools lightly, and this also led to many problems with military and institutional hierarchies. Hodges explains that Turing couldn't understand why anyone would ever avoid telling the truth, even while he himself spent most of his life hiding a basic truth about himself.

In the end, I found the context of Turing's story as important as the telling of his life through letters, papers, and the other paper trails that people leave behind. The "halting problem" went from some ideas I had about "undecidability" to the actual solution of Hilbert's conjecture. I learned about many of the early design decisions that have shaped the field of computer architecture, all by plowing through what I first thought were meanderings. If you want to better understand the context of computer science today, reading Hodges' book can certainly help you.

## USENIX Board of Directors

Communicate directly with the USENIX Board of Directors by writing to [board@usenix.org](mailto:board@usenix.org).

### PRESIDENT

Brian Noble, *University of Michigan*  
[noble@usenix.org](mailto:noble@usenix.org)

### VICE PRESIDENT

John Arrasjid, *EMC*  
[johna@usenix.org](mailto:johna@usenix.org)

### SECRETARY

Carolyn Rowland  
[carolyn@usenix.org](mailto:carolyn@usenix.org)

### TREASURER

Kurt Opsahl, *Electronic Frontier Foundation*

### DIRECTORS

Cat Allman, *Google*

David N. Blank-Edelman, *Northeastern University*

Daniel V. Klein, *Google*

Hakim Weatherspoon, *Cornell University*

### EXECUTIVE DIRECTOR

Casey Henderson  
[casey@usenix.org](mailto:casey@usenix.org)