The Unix Programming Environment

When he came across a classic programmers' book, **Ben Everard** had to take a look.

We came across this classic computing book in a second hand sale and couldn't resist looking at it to see if we could still use it with a modern Linux system.

The Unix Programming Environment isn't a book about programming in general; it's about how to make your programs run well on Unix, and how to use the Unix tools to make your life easier. It starts with the basics like logging in: "Be sure that the switches are set appropriately on your device: upper and lower case, full duplex, and other settings that the local experts advise, such as the speed or baud rate".

We've been a little unfair with that previous quote – it was specifically picked to make the book sound old fashioned. In truth, surprisingly little has changed. The filesystem is a little different (home directories are no longer in **/usr** for instance), and the **#** character isn't used as a backspace any more, but on the whole, the text is very relevant to a modern Linux environment. Sometimes, the methods it recommends are a little antiquated, even though they still work. Of course, you can still edit text files in **ed**, though we don't actually know anyone who does.

The book goes on to focus on using the shell and the tools that were relevant to programmers in the 80s (**sed**, **awk**, **grep** etc). These are still widely used and work in much the same way today. In many cases, the modern GNU versions contain more options than the original Unix tools did, but the core functionality is the same. This is especially true of Bash when compared with the original shell. There are a few cases, such as **mail** and **news**, where the commands do still exist, but are rarely installed as usage patterns have made them mostly obsolete.

Of course, Unix isn't about using individual commands – it's about linking them together. This principal lies at the heart of Unix, and so the chapter that focuses on joining commands together is still perfectly relevant. In fact, there's almost nothing to show that this section of the book wasn't written recently.

It then goes on to show you how to write code that interacts with the system, and the examples are written to make them easy to use in the middle of pipes. This includes using standard in and out, using files, and other system calls. Probably the most surprising part of *The Unix Programming Environment* is that the C code that it gives in the examples still works on our modern Linux machine.
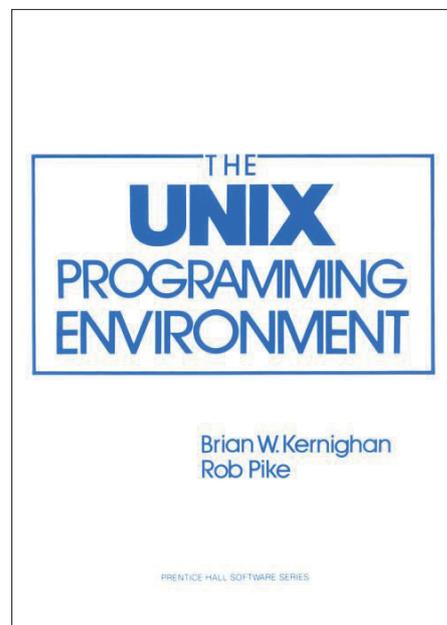
Well, it almost works. We typed in some of the examples and found that most needed to include more header files than were necessary in 1984 (this produces the message "Warning: incompatible implicit declaration of built-in function"). What's more, the design of these programs – taking input from **stdin** and outputting to **stdout** and **stderr** – hasn't changed, at least for command line tools (the book doesn't cover any graphical tools). Again, there are some new parts of Linux (**/sys** and **/proc** for example) that aren't covered, but this isn't that important for most programs.

Practicals from the past

The *Unix Programming Environment* builds up into a big project at the end, which is writing and entire programming language (yes, really) by combining C with programmer's tools such as **yacc**, **lex** and **make**. This was, for us, the best part of the book. Kernighan and Pike are two of the most famous programmers of all time, and in this chapter, they take the reader step-by-step through their process of designing and building a non-trivial program. For us, the whole read was worth it for this alone.

It should come as no surprise that this book isn't really suitable to a newcomer to the world of Linux programming. There's just too much modern stuff that's missed out.

This book is a companion to *The Unix Programmer's Manual* and *The C Programming Language*.

However, for someone with a little experience in Linux and who wants to know more about it, this is still a good choice, especially for any aspiring programmer. Not everything in it is completely relevant, but it covers the basics extremely well, and as you would expect, the examples are insightful, and completely based in the Unix way of doing things. There is also real geek-fun to be had in trying to spot the things that have changed in the years since the book was first published.

The fact that you can still follow a 30-year-old programming book is testament to the stability of the Unix design that Linux has inherited. We got the book as a historic curiosity, but it turned out to be more useful that we anticipated.

"**For someone with a little experience of Linux who wants to know more about it, this book is still a good choice, especially for an aspiring programmer.**"

LINUX VOICE VERDICT

Author Brian W Kernighan and Rob Pike
Publisher Prentice Hall
ISBN 0-13-937681-X
Price £48.99

This book was published in 1984 – the same year Sinclair released the ZX Spectrum – yet is still applicable to modern Linux systems.

★★★★★

Makers: A New Industrial Revolution

Makers will change the world, but **Ben Everard** isn't sure how.

This book isn't really about makers. Not the average folks who go to hackspaces and make stuff that interests them. It's about individuals and small teams or entrepreneurs making customer-oriented products.

The main premise of *Makers* – that we're entering a third industrial revolution driven by the ease of small companies producing custom products – seems shockingly naïve. According to this book, soon these entrepreneurial makers will take over the manufacturing world because small-scale production is becoming more cost effective.

They won't. It's easier to start a business producing goods than it has ever been, and there will be a growing community of crowdfunded small-team products (Linux Voice is one), but this will always be a small segment of the developed economy.

Makers have started to be more innovative than many large businesses,
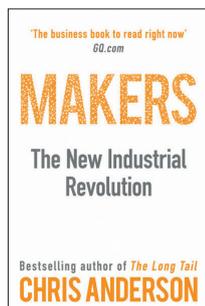
Despite there being plenty of "maker" options available, this book is published by a large publishing firm.

and this will certainly spur on some change. However, this book's vision of a third industrial revolution is unlikely to come to pass.

LINUX VOICE VERDICT
Author Chris Anderson
Publisher Random House
ISBN 978-1-847-94067-4
Price £8.99
A wildly optimistic book that overhypes the role of entrepreneurs in the economy.
★★☆☆☆

Homeland

Andrew Gregory sighs sadly at Orwell's warnings.

There's an early moment in *Homeland*, the sequel to Corey Doctorow's *Little Brother*, when the protagonist just happens to stumble across a game of Dungeons and Dragons played by Will Wheaton, Mitch Kapoor and the founders of the Electronic Frontier Foundation. This episode pulls you out of the story and hits you over the head with the knowledge that the book has Something To Say.

That's a shame, because it really does have something to say. We're being monitored at all times, by companies that are under contract to our governments. The police are increasingly an agent of these companies rather than protectors of the public. And we're carrying mobile telescreeens with us all the time, doing Big Brother's job for him.

As a message, Homeland is powerful, and really crystallises the gut mistrust that many of us feel into something that's easy to understand on a rational level.
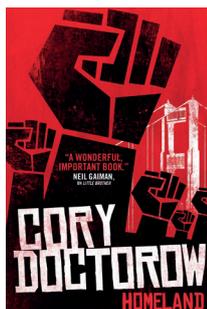
Surveillance, intrusive authorities and an economy hopelessly skewed against the little people. Thank heavens it could never happen here!

As a story, it's a decent page turner and a step up from Little Brother. Best read in a crowed place with lots of witnesses…
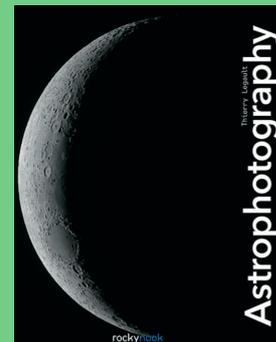
LINUX VOICE VERDICT
Author Corey Doctorow
Publisher Titan Books
ISBN 9781781167489
Price £7.99
Turns a thousand paranoid Reddit and Slashdot comments into a decent thriller and a powerful warning.
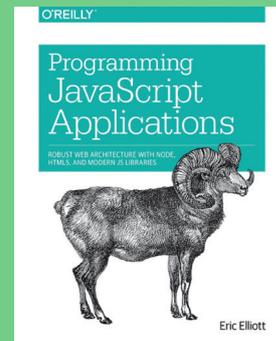★★★★☆

## ALSO RELEASED…

Space goes on forever.

Astrophotography
We're suckers for anything from space, especially photons as they travel from distant suns. What we'd really like to do is capture them. This book looks like it's going to be especially useful, starting as it does with a simple camera on a tripod.

For a quick intro to JavaScript, turn to page 102

Programming JavaScript Apps
We like JavaScript. It's a little like the BASIC of the web, but it's also capable of sprawling great complexity, especially when combined Node.js, HTML 5 and other modern fun stuff. Maybe the age of web and desktop application unification isn't that far off?
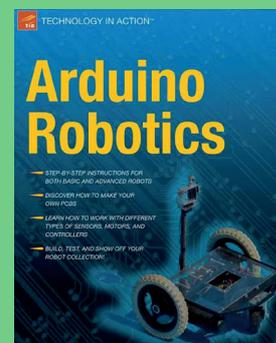
Ya tvoi sluga, Ya tvoi rabotnik

Arduino Robotics
If Ben's awesome Arduino-based Nerf gun targeting system (p90) has whet your appetite for robots and self-defence, this book should enable you to take on Skynet single handed by building your own robot army. It even claims to include a DIY Segway!