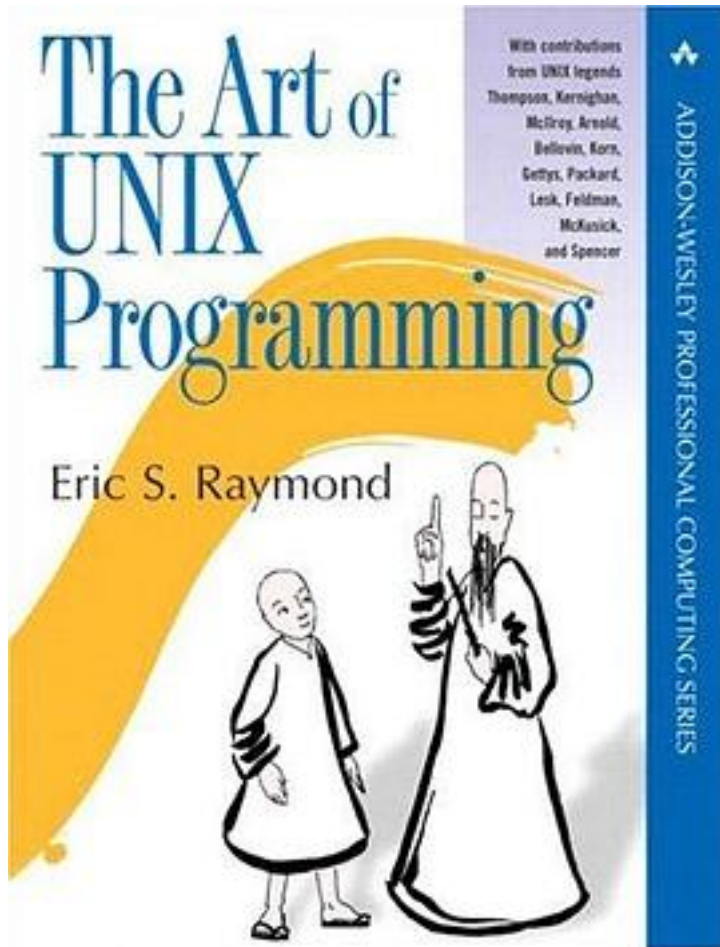


The Art of UNIX Programming



[The Art of UNIX Programming_ 下载链接1](#)

著者:[美] Eric S • Raymond

出版者:Addison-Wesley

出版时间:2003-10-3

装帧:Paperback

isbn:9780131429017

Writing better software: 30 years of UNIX development wisdom In this book, five years in the making, the author encapsulates three decades of unwritten, hard-won software engineering wisdom. Raymond brings together for the first time the philosophy, design patterns, tools, culture, and traditions that make UNIX home to the world's best and most innovative software, and shows how these are carried forward in Linux and today's open-source movement. Using examples from leading open-source projects, he shows UNIX and Linux programmers how to apply this wisdom in building software that's more elegant, more portable, more reusable, and longer-lived.

作者介绍:

目录:

[The Art of UNIX Programming 下载链接1](#)

标签

Unix

Programming

编程

哲学

计算机

Linux

经典

开源

评论

Eric 的大作，绝对经典！

:无

KISS

开源世界的大师的著作，值得细读

获奖了的书，而且看了之后确实觉得不凡

花了好久的路上时间看完，感触很大

K.I.S.S. 程序员必读，有大量关于设计和实现的精妙言论。

其实这并不是一本讲编程的书，叫Unix文化更恰当，唯一不足是不太称得上是艺术，也跟TAOCP太强大有关系。在阅读的过程中自始至终都在与作者产生共鸣，确实是一件很有趣的事情

增广见闻

我已经读过这本书了，可是在豆瓣猜里面还是有它的各种版本出现……无奈，只好把所有版本放进来了

讲了Unix的一些设计理念

read ch16

力荐啊力荐

没有点功底还真不行。unix文化大观园。

内行有门道可看，外行有热闹可看。

精髓在机制而不在策略。

The best things in life are open.

classic

以前读过中文版

读过中文版~~

[The Art of UNIX Programming 下载链接1](#)

书评

贯穿始终的 KISS 原则，很多年前就被谆谆教导过了。它被我无时无刻的都拿出来警告自己的设计过程。读完这本书，让我对 KISS 又有了一次升华。其实，这本书对我几个月来设计游戏服务器架构的影响是满大的。坚定了我每写一个程序做好一件事的决心。让我更确信用多进程的设计取代...

作为一个多年的开源（Open Source）拥趸，像《UNIX编程艺术》（The Art of Unix Programming）这样的好书自然不能错过。大约一周前我无意中在公司书柜中发现了它，立刻开始投入阅读。现在，我已经开始边读第二遍边作读书笔记了。开篇的译序很有趣，第一句话就写道：“...

This book reveals the history, art, culture, philosophy, practices, guideline about programming (with) Unix, from the OS itself to the programming languages. It is not doubt that Unix is one of the most influencing OS ever built. Just as mentioned in the...

所以在这里可以读到正版的：<http://www.faqs.org/docs/artu/>
当然如果不是非要在网上读这样需要思考一下的书不可，买一本还是很值得的。

这本书主要偏向软件工程的角度。在joelonsoftware上有一篇很详细的review（我也是看那篇review才想要读这本书）。其中提到Windows（或者...

本文同步发于我的blog: <http://www.vingel.com>

这本书我已经买了三个星期，一直在看。以前看它的英文版，仅仅走马观花般看了一遍。现在这个中文版，目前为止我已经看了三遍，而每次都会有新的体会。我得到的关于《Unix编程艺术》最深的感想就是：Unix中无所不在的K.I.S.S(Keep...

书籍拿到手比较厚实，感觉很有分量，不过读起来倒不那么难啃。翻译的还是很流畅，整体下来很酣畅，译者说用了1年，看来还是比较恰当的。书中介绍了unix的文化、历史，举了不少案例。ESR身上unix黑客气味贯彻始终本书，在ESR看来unix编程就是个玩的过程。本书中融贯了n...

买这本书主要冲两点，第一是封面上的Software Development Productivity Award标志，第二是作者Eric Raymond，Raymond是开源运动的发起者之一，他的经典文章“大教堂和市集”广为流传。由于没有急用而且是英文，书买来后看了目录就收到书架上。最近又有项目要用linux...

这本书买的比较晚，却是赶在其他书之前最先读完。前前后后估计读了三个月有余，当然因为中途有好看的小说插队，以至于耽搁了不少。这个标题很容易让人以为和那部旷世巨作一样，还好通过评论，已经知道是一种思想的阐述。所以刚才，当我把书放回书架的时候，犹豫了一下，还是...

这本书不是技术类书籍，不是Knuth的The Art of Computer Programming的姊妹篇，而是一本关于黑客文化的书。所以，适合对这些技术有些了解或感兴趣的人，睡前拿来翻一翻。书里没有数学公式，甚至连代码也没有。有的是大段大段的历史和引文，成与败，得与失的比较，最终落到开篇...

这本书是一种智者的言论，作者的经验学识让他所见所想都不同于我们，当一切成为历史，留下的沉淀下来的是这些闪光的思想。但对于性能应该放到最后才进行考虑，有见仁见智的观点，对于性能应该视之为等同于风险来评估预测和管理控制，特别是对于大型软件，开发周期长，人员多，...

500来页的书以五个晚上的时间飙完，本身就说明了其简单。是最近以来看得最爽利的两本软件书之一（另一本是云风的《我的编程感悟》），全书是对“主流”软件工程的反动，但每每契合吾意，往往有醍醐灌顶之感。ESR属于优点和缺点都很明显的作者，还好这次有心理...

如果早三年就去读，我的人生必将不同。如果早五年来读，大概我还读不懂。

写程序做项目，许多事情没有经历过，是很难体会的。（刚写了一大段攻击C++的文字，怕引起圣战，就删了。此处省略300字。）
好吧，总之，不论经验丰富还是初出茅庐，都建议读一读。也建议过一两年再捡...

3.1.3 Cooperating Processes it would not have been trivial without the fundamental unifying notion of the process as an autonomous unit of computation, with process control being programmable.
这句被翻译成了：……进程是自主运算单元的统一性记号…… 你能...

<https://gcd0318.wordpress.com/2016/04/20/%E5%8F%88%E5%88%B7%E4%BA%86%E4%B8%80%E9%81%8Dtaoup/> 2016年04月20日 又刷了一遍taoup Filed under: 感悟 — gcd0318 @ 04:10
十年以来每隔两三年就把这本书拿出来再读一遍，而且一遍比一遍读的快，一方面是我自己的领悟提升了， ...

目前读到了第13章，中文版的。
如果了解Unix的哲学，并且从多个视角去了解Unix，无疑，TAUP是一个很好的起点。
不管是Unix的设计原则，还是Unix的诞生历史，ESR都信手拈来。
每个章节的组织都是松耦合的，这也给了读者极大的阅读自由！

从作者写书到我读这本书已经事隔十年。
对于一个使用linux有2年的人（其实我一直不认为经验年限和对一件东西的了解程度有线性关系），我阅读完之后的感受是：有些观念很深入我心，有一些却不了解，或者根本提不起兴趣看。
还是说说那些深入我心的感悟。 1，机制和策略的分离， ...

<http://herpolhode.com/rob/ugly.pdf> The Good, the Bad, and the Ugly: The Unix! Legacy high-level programming language hierarchical file system uniform, unformatted files (text) separable shell distinct tools pipes regular expressions portability security ...

原著成书于2003年，而且作者说写了5年。从2003到现在2011已经又8年了，许多事情又变化了。。。这本书在学校就读过，但当时也就是读过一遍，略微记住几个名词而已，比如KISS和文本化，现在回想下，这两个概念对自己影响确实蛮深的，当初没有白读。最初阅读时，好多软件都没有接...

内容涵盖从philosophy到 practical issues。其实K.I.S.S.的概念很简单，容易理解，但是就知道K.I.S.S.的概念是远不够的，理解并运用K.I.S.S.才是我们的最终目标。本书包含的很多Case study非常值得一看，里面包含了多年实践的经验，已经对未来软件设计的指导，very nice.

[The Art of UNIX Programming_ 下载链接1](#)