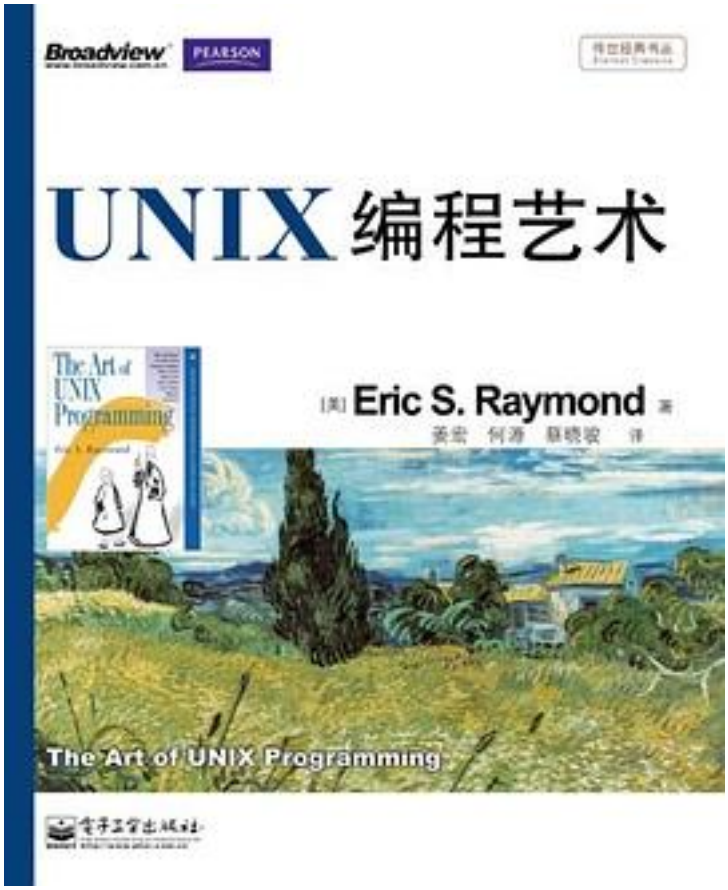
UNIX编程艺术



UNIX编程艺术_下载链接1_

著者:[美] Eric S・Raymond

出版者:电子工业出版社

出版时间:2011-1

装帧:

isbn:9787121123290

本书主要介绍了Unix系统领域中的设计和开发哲学、思想文化体系、原则与经验，由公认的Unix编程大师、开源运动领袖人物之一Eric S.
Raymond倾力多年写作而成。包括Unix设计者在内的多位领域专家也为本书贡献了宝贵的内容。本书内容涉及社群文化、软件开发设计与实现，覆盖面广、内容深邃，完全展现了作者极其深厚的经验积累和领域智慧。

作者介绍:

《The Art of UNIX Programming》，简称TAOUP，作者Eric S. Raymond，简称ESR。这大概是计算机类书籍中很少见的一本课外读物。TCP/IP编程之类典型Unix编程书中讲到的东西在这本书里面找不到，所以书里讲到的当然就是别的书里找不到的东西。读者也许需要有相当的Unix背景、或者长期钻研某个专题，才能体会到作者的弦外之音。ESR作为老牌黑客信手拈来的典故，如果不是在Unix里面长期浸淫，大概很难有所共鸣，所以把这当作Unix的一部坊间史话倒也合适。

目录: 序 xxv

UNIX编程艺术_下载链接1_

标签

UNIX

编程

哲学

计算机

程序设计

经典

计算机科学

Linux

评论

地铁上粗读，这些好听得耳朵都磨出茧子了，其实我更想读 Unix 的设计缺陷=。=

这本书关于模式、规则的理解，达到了一个较高的境界。

这个是新翻译的版本 还是只重新印刷的？见鬼

: TP316.81/1162-1

应当一年一读。

早读早好

读了一半。感觉实在是太老了，有点过时

当成一部历史书来读比较好。了解Unix的文化，能用上就用，不能用上就了解一下，扩展知识面非常不错。

说的是Unix的文化与哲学

很多东西目前还无法理解，工作几年之后会重读的。

部分翻译太屎了

南图 MS没有留下什么东西，需再读

喜欢这本书的前半部分，后面的部分很多软件没有实际用过，感悟不深。

他不是技术手册，他是哲学教材

这本书很好都说明了什么才是Unix风格，什么才是正宗的Unix程序员。

大牛作品,值得一读...说是编程,实际基本没代码在里面,主要介绍编程哲学,不过很受用.

对win系统及尽揶揄之能事 过瘾

在地铁、公交车、床上读完的

学习UNIX的哲学

没啥收获，啰里啰嗦

UNIX编程艺术_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。
如果想了解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.

UNIX编程艺术_下载链接1_