

“Only One Book” 计划（计算机理论类，非TCS）



Snowflyt
20岁，学生，苏州大学软件工程在读

关注他

1,196 人赞同了该文章

“Only One Book”计划，顾名思义，即针对每个细分领域/方向/阶段只推荐一本（个人认为）最好的入门书，免去读者在书海中挑书/选书的烦恼。本文聚焦于计算机理论这一大领域，分别在计算机导论、数据结构与算法、数据库、计算机硬件、计算机组成原理、操作系统、计算机网络等领域各推荐一本最值得阅读的入门书，同时尽量避免不同书之间出现大量重叠内容，并且按照学习顺序由浅到深排列，以形成一套系统的学习路径。

老实说，我本来想将这些内容放在“计算机专业必读哪些经典书籍？”那个提问下的，然而那个提问已经被我拿一个有些跑题的回答占领了，因此就单独开一篇文章了。感兴趣可以去看看，我在那个提问下写了份简易挑书指南：

计算机专业必读哪些经典书籍？
21 赞同 · 0 评论 回答

下面切入正题。

前言

计算机理论知识的学习线性很强，本书单考虑到这一点，尽可能按照学习先后顺序安排每本书的所在位置（如果本系列书单以后还出了其他领域的书单，那就另当别论了），例如计组的书会安排在操作系统前面。同时，我也会给出阅读每本书所需的前置知识，以免有读者在错误的学习阶段阅读了错误的书。

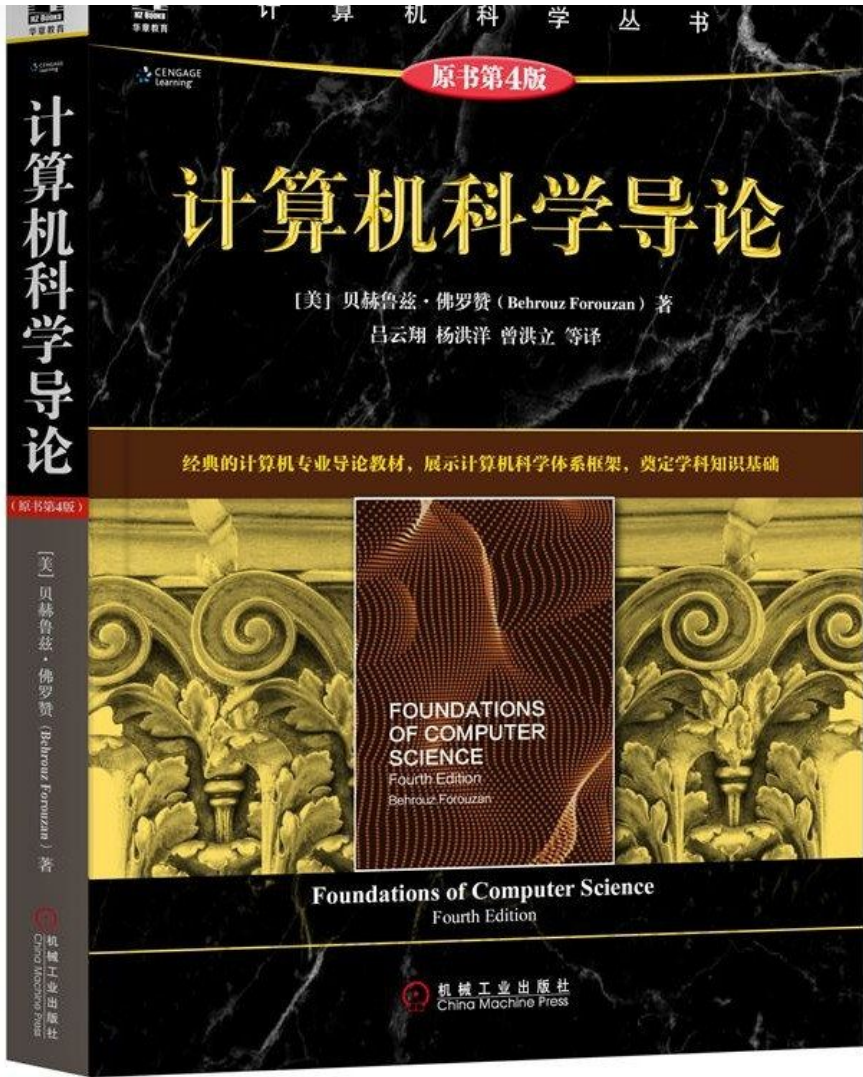
需要在前言指出的一点是，CSAPP（即《深入理解计算机系统》）这本书并未直接包含在本书单中。这确实是一本很好的书，但CSAPP大杂烩式的风格导致它既难以归类到导论性质的书中，又难以归类到任何深入讲解某一领域知识的书中，因此不会直接作为某一细分领域的推荐书籍。如果归类到导论中，CSAPP未免太厚了，如果归类到计组操作系统类的书中，CSAPP在某些层面上又似乎不够深入，因此我并未将其放入该书单。但CSAPP是一本适合你在学习的任何阶段都抽空读上一点的书，读它总会有不同的收获。没把CSAPP放在这里不是说明它不重要。

另外，本书单只包含一些比较重要的理论领域书籍推荐。如果你打算在本书单中看到密码学、体系结构、计算理论甚至组合数学的书籍推荐，那你恐怕就要失望了，因为这些领域相对小众，也并非大部分计算机领域工作者需要深入学习的。我不排除后续会加入这些小众领域的书籍推荐，但暂时这里只包含了（大多数）大学本科阶段作为必修的几个理论领域。

计算机导论



知乎 首发于 “Only One Book”计划（书单）



中文书名：计算机科学导论

英文书名：Foundations of Computer Science

前置知识：无

正如书名所示，这是一本“导论”书。几乎任何学校的计算机科学与技术专业都会在大一开设一门“计算机导论”。简单来说，导论就是大杂烩，把大多数计算机领域都简要给你介绍一遍，但都不深入。然而，**在你有时间的情况下**，学习一下导论是很有意义的。它可以帮助你很好地树立起系统的计算机概念，帮助你在后续各个领域的学习中很轻松地入门，而非在哪门课最初一段时间的学习中感到困惑。

不过，**导论并不是你一定需要先学习的。如果你时间紧急，可以直接跳过导论**，开始数据结构与算法的学习。另外，导论中确实有一些部分不是你必须要阅读的，例如社交媒体导论，在多数情况下就是不需要的。但我还是建议有时间尽可能读一遍，这花不了太多时间，而且会给你带来宝贵的计算机领域“常识”。

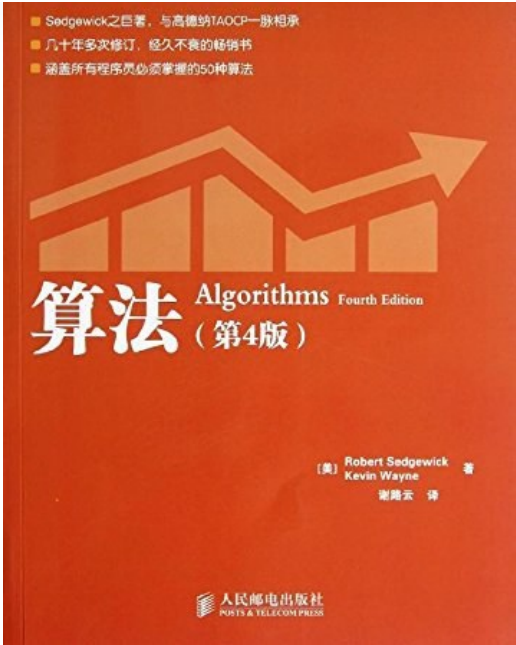
我明白上来就看到一本机工社的黑皮书很容易把人吓退，但这本导论真的是很简单很容易读懂的那种.....这本书翻译还过得去，厚度也适中（不到400页），我想作为一本合格的导论，这厚度与内容量算是底限了。

数据结构与算法



知乎

首发于
“Only One Book”计划（书单）



中文书名：算法

英文书名：Algorithms

前置知识：任何一门编程语言的基础知识

官方网站：[Algorithms, 4th Edition](#)

正如其朴素的书名一样，这是一本算法书——一本完完全全的“算法”书，而不是“算法分析与设计”这些课程的参考书。

说到算法，我相信很多人第一个想到的书是MIT的《算法导论》。不得不承认，直到今天《算法导论》仍是算法分析领域最权威的参考书之一。注意到了吗？我说的是“算法分析”而非“算法”，实际上前者更偏向一门数学课，会涉及较多的时间复杂度分析，而后者则偏向实用。而在当前，除非你需要做一些颇为深入的理论研究，否则学习“算法分析”并不能给你太多实际的帮助。各大公司所谓“算法知识”，要求的自然也是“数据结构与算法”方面的知识。而这里推荐的《算法》被公认为数据结构与算法最好的入门书之一，也有很多人认为这是当前最好的算法入门书。

本书演示算法使用的语言是Java——这可能令一些人感到意外和疑虑。“如果我从未学过Java，是不是说我需要先学点Java才能看这本书？”，我的回答是，这本书几乎不需要你拥有Java基础，也许仅需的一点基础你在类似菜鸟教程这样的网站上简单翻阅前面的一点内容就够了。

为了照顾到使用其他语言的读者，这本书尽可能使用了一个有些特别的封装库，使代码看起来更易懂、更通用，例如使用自定义的标准输入/输出类“StdIn/StdOut”来代替Java中的Scanner与System.out，并且很少使用较为深入的Java特性。我认为只要读者有部分C/C++甚至Python基础，要理解其中的代码都不会有很大难度。即使你不打算学习任何有关Java的知识，直接开始阅读这本书，前面也花了一些篇幅大致介绍了Java语言。如果你打算直接靠这些较少的介绍入门Java可能会稍显吃力，但应该也大致足够。

其实我个人认为可以不考虑安装书中提供的自定义标准库，毕竟对于初学者而言，这可能也是一个劝退要素之一。考虑到Java环境的配置本来就不那么容易，还要引入一个标准库或许更容易让初学者头晕。因此，我建议直接使用Java内置的Scanner和System.out代替书中的StdIn和StdOut，你可以很容易找到如何使用它们。然而，书中使用的绘图库就不太容易使用原生Java直接模拟了，但考虑到绘图本身只是帮助理解，而非绝对必要的，你可以跳过那些要求你使用绘图库完成的习题。

也可能有人会问，“为什么《算法》不像《算法导论》一样使用伪代码描述？”，这是因为本书希望书中的所有代码都是可运行的，读者也可以根据课后习题编写可以实际调试运行的代码。如果你

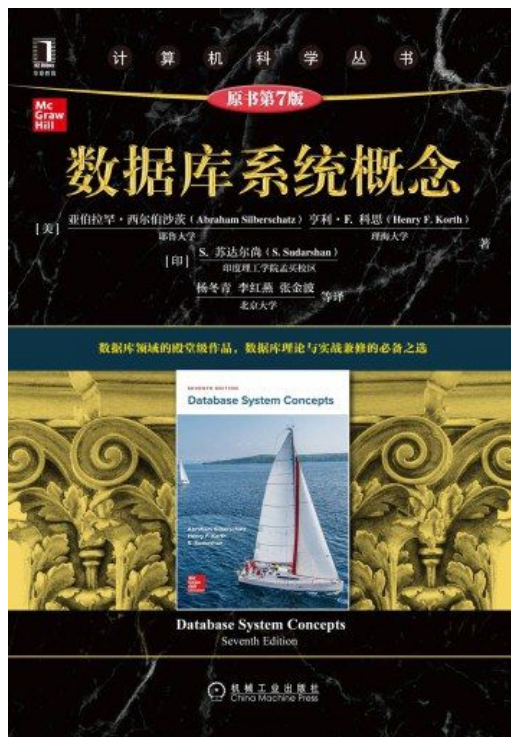


懂，且附以大量用于描述的图片 and 表格，即使你一行实际的Java代码也不曾实际运行过，想明白书中描述的各种数据结构和算法也是非常轻松的。你也可以很容易找到本书使用各种语言重写的代码和课后习题参考。读过这本书的人很多，资料自然也非常多。顺带一提，如果你不打算学习Java，那么第一章中的大部分内容可以跳着读。

另外，需要指出的一点是，算法的学习并非读完一本书就足够了。更重要的是时常去做相关习题，例如在LeetCode上刷算法题。读书只是开了个头，想指望靠读本书就通过各大公司的面试是不可能的。想要熟练掌握算法，不刷题肯定还是不行。

国内邓俊辉老师的《数据结构》似乎也是一本不错的入门书，但我没读过，就不推荐了。感兴趣可以去看看。

数据库



中文书名：数据库系统概念

英文书名：Database System Concepts

前置知识：数据结构与算法

官方网站：[Database System Concepts - 7th edition](#)

数据库知识的学习并没有太多前置知识要求，**将对数据库的学习放在学完数据结构与算法后的任何阶段都是合适的**。有些学习路径会将其安排在学完操作系统之后，这也没什么问题。本书单将数据库安排在算法之后的原因是，下面对计算机硬件知识的学习涉及的算法知识较少，但算法学习非常需要巩固，直接开始学习硬件很容易忘记刚学习的很多算法知识。而数据库这门课程对算法知识的涉及很多（尤其是在文件系统与查询处理部分），学习数据库的过程中也能顺带巩固一下算法知识，因此这么安排。

数据库领域没有太多读物可以推荐，基本上入门数据库都推荐这本《数据库系统概念》，本书单也遵循传统，推荐这本巨著。书中不仅包含了最基本的SQL语句、ER图设计、数据库范式、文件系统、查询处理与查询优化、并发与恢复等一切数据库课程都应该教授的基本内容，也包括了一些实用且前沿的内容，例如关系式数据库、分布式数据库、大数据与数据挖掘。在云原生的时代，《数

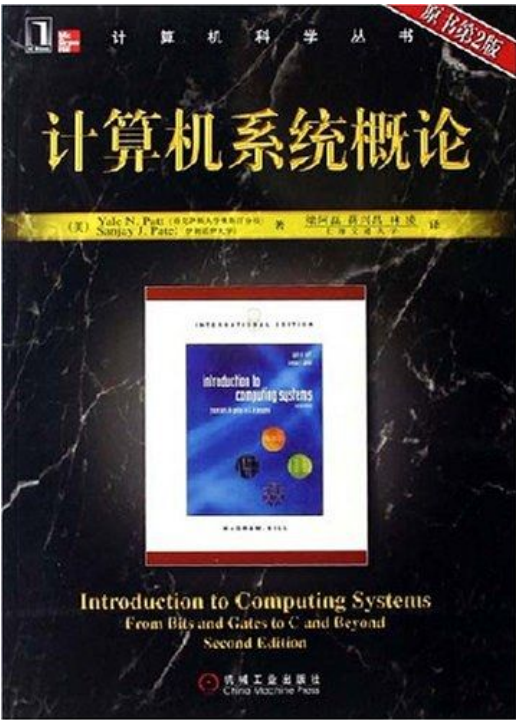


在我个人看来，将对数据库的学习安排在前面还有个好处，就是能够与之后对操作系统知识的学习产生共鸣。在文件系统和索引部分，你可以了解到数据库如何为了提高文件读写性能而绞尽脑汁，而其中的一些思路与操作系统所采用的一些通用解决方法不谋而合。同理，在阅读事务管理部分时，你可以看到数据库如何为了保证原子性而考虑并发控制（锁），而操作系统的并发部分与之非常相似。而如果你第一次接触这些概念是在学习操作系统而非数据库时，也许会因为缺少一些更为实际的例子而难以体会这些概念的重要性。

值得注意的是，《数据库系统概念》这本书中文版将近800页，如果你感到自己很难啃下这么厚一本书，或许可以考虑阅读“本科教学版”，它只有不到500页。“本科教学版”相比原版删去了关系代数、复杂数据类型、大数据与数据分析、RAID、并行与分布式这几个部分，如果你对大数据与分布式数据库不那么感兴趣，“本科教学版”应当也足以提供完全足够的基础知识了。这里当然推荐你尽可能阅读原版，毕竟“本科教学版”删去的东西在如今看来还是相当重要的，然而如果你抱着一些功利性的目的学习数据库，例如打算通过面试或参加考试，不得不承认“本科教学版”也完全足够了。

在本书单中，**这本书的阅读难度可能是最大的那几本之一**，的确要花点力气，要不然这里也不会提出读“本科教学版”这种替代方案。然而**数据库领域确实很难速成**，毕竟一提到数据库一定会提到范式，就会提到函数依赖，然后就会有许多数学推导，即使你不读这本，其他书也一样不容易，那不如读最权威的一本。我的建议是如果你确实读着很有困难，可以考虑先不用和那些公式与算法死磕，集中阅读SQL、E-R模型、关系数据库、存储管理、索引、查询处理、事务管理、并发控制、恢复系统这几章，这些是数据库课程中最核心的内容，也是通常本科阶段一定会教学的内容。至于其他有关分布式、大数据等领域的内容，也许你可以到自己真正需要它们时再读。

数字电路&计算机系统概论



中文书名：计算机系统概论

英文书名：Introduction to Computing Systems: From Bits & Gates to C & Beyond

前置知识：无

提前声明，如果你已经系统学习过**数字电路**，并且已经有了一定的**C语言基础**（指较为熟悉指针，包括函数指针的运用，能够在堆上使用malloc/free释放分配内存，并且对结构体相关的知识有不

有人可能会疑惑我为什么不直接推荐学习计组，而是要插入这样一本看起来有些奇怪的书。我们通常认为学习计组是正式学习计算机理论的开始，但是我既没有选择经典的CSAPP前四章，也没有选择广受好评的《计算机组成与设计 硬件/软件接口》作为开始，而是选择了耶鲁大学的这本《计算机系统概论》。

这并非一本真正的计组教材，而是一本概论性质的书。从副标题“From Bits & Gates to C & Beyond”就很容易看出，**这是一本从二进制和逻辑门开始，一步步介绍计算机中程序运行逻辑的书**。这本书集中于一步步带你了解计算机如何真正运行一段程序，因此许多和主题不那么相关的东西，比如SRAM和DRAM的区分，就不在这本书的讨论范围内了，因此**这不是一本能代替计组教材的书**。

选择这本书作为开始似乎有些多余，然而当我真正翻开这本书时，我才意识到一开始阅读这样一本书是多么重要。我想如果我在大一时就看过这本书，即使只看过一小半，应当也能受益匪浅。

实话实说，如果你打算直接跳过这本书也没事，毕竟下面即将提到的《计算机硬件基础 硬件/软件接口》的附录中也能教会你一些必要的汇编语言/C语言语法，而且也包含了逻辑电路基础（甚至有些verilog）。但这是一本不太厚的小书，在学习路径中插入这样一本书应该不会花你太多时间（毕竟，书中后半段很多关于C的内容你其实可以挑一点看，不用全看）。

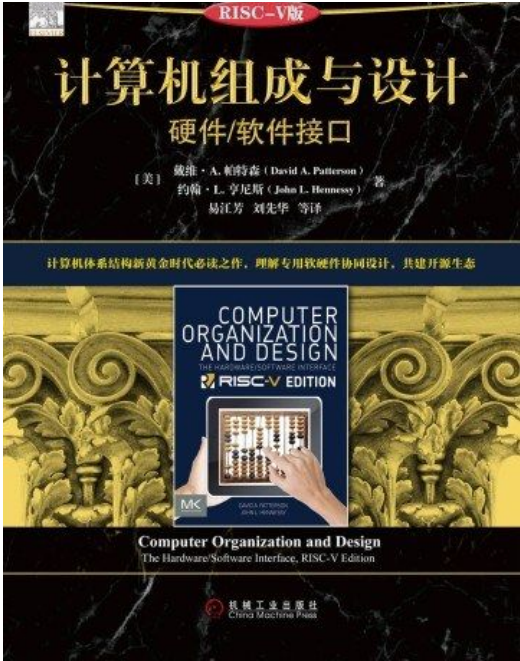
我推荐它的一大原因是因为**这本书几乎没有阅读门槛**，非常适合通过这本书先一步对整个计算机理论体系进行一遍梳理。和计组有关的很多书都会大量涉及汇编或C语言，但却很少有篇幅认真介绍它们，这导致这些书往往需要一定的前置知识才容易看懂。而且，这些相关书籍中对C语言的要求不止局限于能够掌握基本语法，还要求对系统级API调用有一定了解，直接硬读这些书籍中的C语言代码，容易一知半解。而《计算机系统概论》这本书不同，**它在大致介绍完硬件基础后，用了很大篇幅介绍C语言**，更难能可贵的是这部分对C语言的介绍和前面讲解的硬件基础直接关联到一起，这种讲解方式让人很有启发。

你可能会认为需要一本如《C Primer Plus》或《C++ Primer》这样的书帮助你详细地了解C/C++的相关知识。然而如果你未来不打算从事相关工作，读这些书很容易让你过度陷入语言细节。如果你的目的仅仅是使用C/C++辅助理解计算机理论方面的知识，那么这本书应当已经完全足够了。

如果你有能力或者有挑战自己的想法，那么你可以尝试阅读这本书的第三版英文版，第三版出版于2019年，相比起2008年的第二版要新得多。你可以直接在国内买到英文版，也可以很容易地自行找到英文版电子版。不过对于大多数人来说，现在已有中文版的第二版也不算太过时，毕竟理论知识的过时速度非常缓慢（1984年的SICP还被奉为当代经典呢），如果你还是更偏爱中文版，稍旧一些的中文版第二版也可以阅读。

另外，你可能会对本书单中**不包括模拟电路相关知识**感到困惑。然而我实在不认为模拟电路对计算机理论的学习有多么重要，例如我很难想象在什么情况下学习计算机理论需要了解三极管原理，在我看来学习硬件知识止步于明白三极管/MOS管可以起到开关作用就足够了，我们至多在逻辑门层面花些精力，将精力深入到晶体管层面就有些不太划算了。当然，这也仅仅是我的个人见解，如果你有异议，我也尊重你的意见，只是本书单确实就是按照我的想法安排的。

计算机组成原理



中文书名：计算机组成与设计 硬件/软件接口

英文书名：Computer Organization and Design: The Hardware/Software Interface

前置知识：数字电路；数据结构与算法

在阅读完上面一本书后，你应当已经对计算机的硬件基础有一定了解了，现在再正式学习计组应当会轻松不少。事实上，上面一本书涉及的硬件知识已经足以使你直接步入操作系统的学习，但花点时间学习一些更深入的计组知识会更好（当然，如果你很着急，可以直接跳过这本书，但如果你上面一本书也跳过了，这本书就不建议跳过了，你好歹得读一本）。当然，如果你学习计组是为了应付一些更现实的问题，比如本科课程考试或考研，那你就肯定不能跳过这本书了。

在这里我推荐经典的《计算机组成与设计 硬件/软件接口》。在**计算机组成原理**领域，这应当是最经典的作品之一了（实际上，**计算机体系结构**的一些知识也为这本书的厚度做出了很大贡献，这并不是一本纯粹教授计组的书），在国内这更是首选。如果你正在学习计组相关的课程，不妨买一本拿来参考一下，或许比你的老师讲得更好。

我实话实说，这本《硬件/软件接口》并没有达到我理想中最适合学习计算机组成原理的书应该达到的程度，毕竟这其实是一本将体系结构和组成原理混着讲的书，内容安排也稍显枯燥。如果打算将体系结构和组成原理都学好，那这是一本非常合适的入门书，但这里我只想找到一本合适的组成原理书，这好像就显得不那么合适了。然而计组领域我确实找不到更值得推荐的书了，所以暂时将《硬件/软件接口》放在这里。这也是我在之前加入一本概论作为补充的原因。如果你有更好且认为更适合自己的选择，可以考虑将这本书换掉。

值得注意的是，**这本书中一部分内容和上一本书有重叠**。如果你已经阅读了上面一本《计算机系统概论》，那么这本书的不少内容可以略读，你只需要认真阅读那些让你感到陌生的知识。

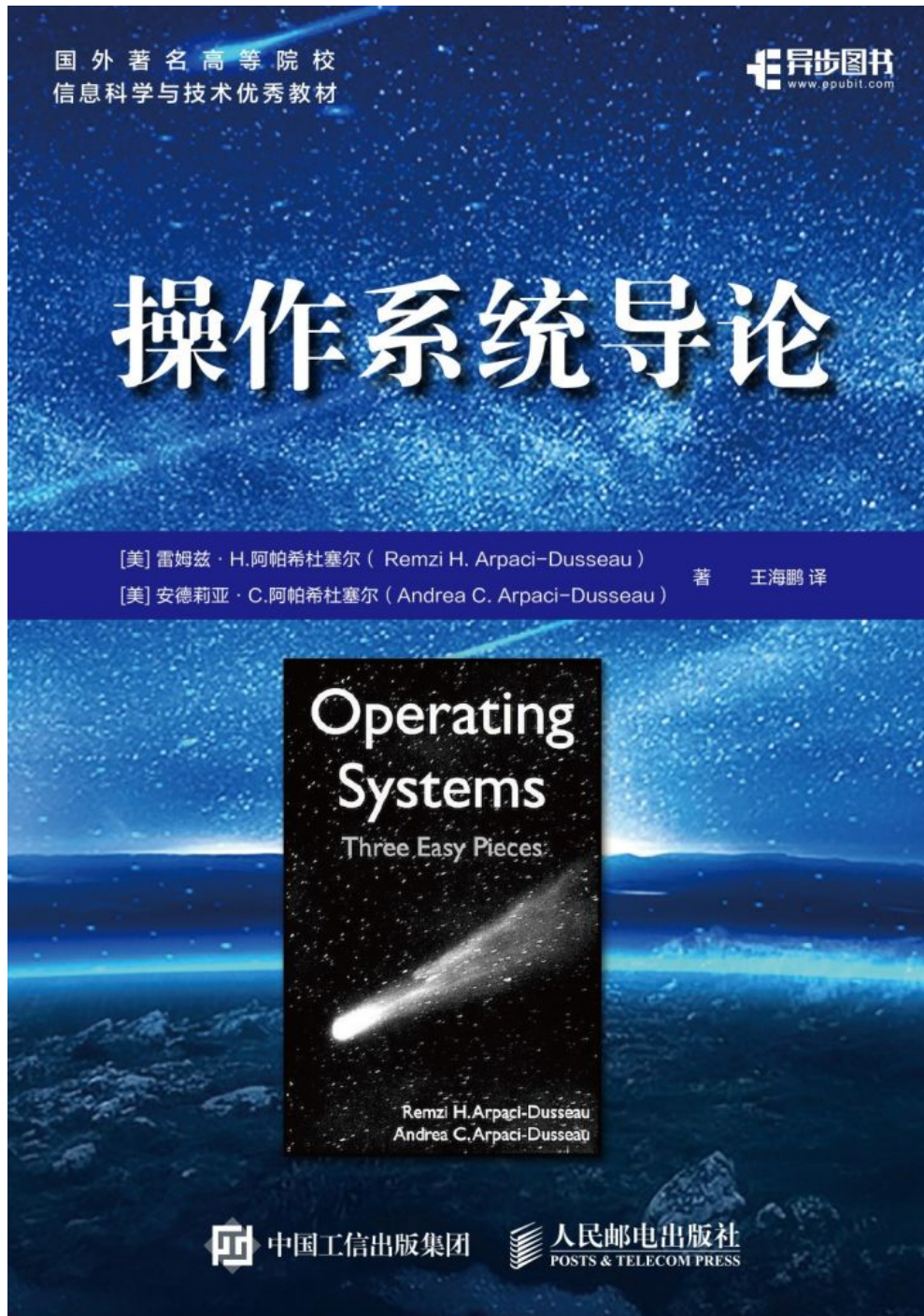
此外，你可以看到这本书有两个版本，除原版外还有个“RISC-V版”。目前来说直接阅读更新的RISC-V版即可。两个版本的区别在于，原版主要采用MIPS指令集（也包含一些x86和ARM指令集）介绍，而RISC-V版显而易见，采用RISC-V指令集介绍，两本书除此之外没有太大差别。相比之下，RISC-V指令集应当更加简单且容易学习，因此更推荐RISC-V版。

你也许注意到也是本书的这两位作者（Patterson和Hennessy）还有另一本更重量级的著作《计算机体系结构：量化研究方法（Computer Architecture: A Quantitative Approach）》，相比这里推荐的这本书，《量化研究方法》是一本更加专业的体系结构领域著作，而且专业到除了专研研究体系结构的研究者外都无需阅读的地步。如果你有志于体系结构领域的研究，可以考虑阅读它



至于大家推荐的CSAPP，我仍考虑将它当作论外处理，建议作为一本“有时间翻翻”的读物，毕竟这本书究竟要归类到哪个领域我至今还难以下定论。有人说CSAPP的前四章也很适合作为计组教材，确实如此，但我想还是在这里放一本更专注于计组的、且更加专业的书更加合适一些。

操作系统



中文书名：操作系统导论

英文书名：Operating Systems: Three Easy Pieces (OS-TEP)

前置知识：数据结构与算法；计算机硬件基本知识；C语言基本知识（至少应当了解至结构体与指针）

官方网站：[Operating Systems: Three Easy Pieces](https://os-tee.org/)



扭，相比之下《操作系统导论》是更为合适的选择（尽管翻译得也比较生硬，但总体可读，至少比《现代操作系统》好上很多）。

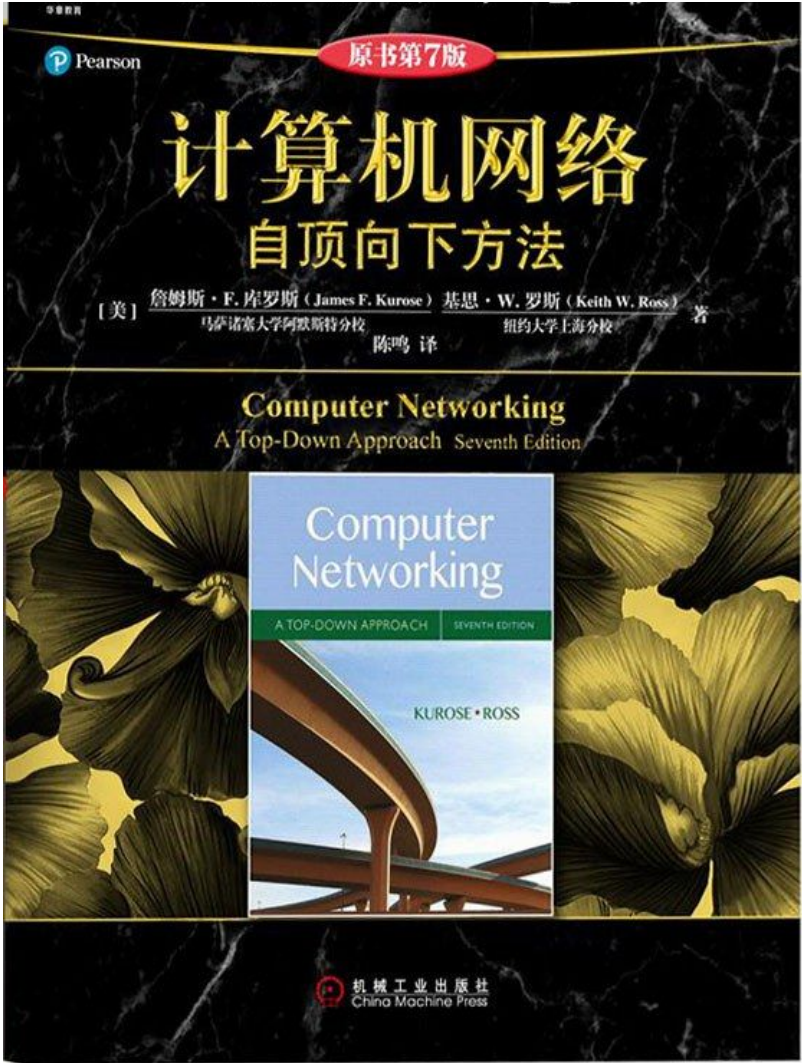
作为本就不容易过时的理论书籍，这本英文原版出版于2018年的书在一众理论书籍中看起来几乎崭新。你可以感受到作者尽可能让学习操作系统变得轻松。书名副标题的“Three Easy Parts”分别指虚拟化、开发与持久性。书中穿插着一些有趣的扩展知识和一些有趣的对话，使阅读体验相对轻松不少。课后作业以给出可运行Python脚本的形式指导读者通过探索式的方法学习，非常生动有趣。除此之外，书后也包含几个大项目可以练手，和大多数西方的专业计算机教材一样，配套资料非常齐全。

与本书单提到的其他书籍不同，这本《操作系统导论》其英文原版是一本完全开源的电子书。由于本书的中文翻译版出版时间是2019年，距今已经有一些时间，所以中文出版物与最新的英文在线电子书已经有了一些微小的区别（然而区别确实很小，如果你不是完美主义者，不用太在意它们之间的微小区别）。为严谨起见，这里贴出本书英文版的官方网站，全书所有章节（的英文版）都可以在官网免费阅读，而且**你可以在官网找到全部配套资料（如作业）**。

当前在线英文版相比中文出版物多出了“Flash-based SSDs（基于Flash的SSD）”一节以及“Security（安全）”部分（除去对话的正文共四节，且安全部分只保留在线电子书中，即使在英文出版物中也不会包含），此外修正了少量打字和排版错误，并且少量修正了部分示意图。如果你打算了解Flash及SSD的工作原理或了解一些加密相关的知识，可以考虑读一读上面提到的新增部分，但通常来说阅读这些内容的必要性不大。另外，本书在微信阅读上有付费电子书可读，这或许可以为你省下一些购买纸质书的费用。

PS：这本书的很大一部分课后作业（编码）需要在UNIX环境下进行，其中绝大部分可以在Linux/Mac OS环境下进行，少部分可以在Windows环境下进行。个人建议，如果使用Windows平台，可以考虑直接使用WSL2进行模拟，现在WSL2支持直接调用Windows本地的VSCode写代码，使用起来应当不很困难，至少比装个Linux/Unix更简单。另外，有一些课后作业（演示）使用Python 2编写，如果你使用Python 3，需要将其中的print语句进行修改才能正常运行。具体来说，就是将 `print 'a'` 改成 `print()`，将 `print ''` 改成 `print()`，`print 'a'`，改成 `print('a', end='')`，基本上这样就可以运行了。

计算机网络



中文书名：计算机网络 自顶向下方法

英文书名：Computer Network: A Top-Down Approach

前置知识：数据结构与算法；一些最基本的Python语法（如果你不想了解，也可以直接跳过书中极少数涉及Python代码的部分，但Python非常简单，学一点基础花不了你半个小时，所以为什么不学一学呢）

计算机网络是个较大的话题，在计算机的许多热门领域（如Web开发）中，计算机网络的重要性似乎也仅次于数据结构与算法。针对不同的需求，对于“计算机网络”这一主题也有许多不同的书籍推荐。然而为贯彻本书单的宗旨，这里还是只主要推荐一本书，即《计算机网络——自顶向下方法》。

本书第八版已出，但尚无中文翻译，如果你对自己的英文水平比较自信（你确实可以自信一些，毕竟这类专业书籍不会涉及太过难以理解的英文句子），可以直接下载[最新版英文电子书](#)阅读。

“自顶向下”不是一个很常见的词语，但即使你从未见过它，也应该能够很容易地从字面意思猜出它的含义——从最顶层一步步讲解到底层知识。很多人第一次接触“自顶向下”这个词语可能就是在看到这本书的标题时。“自顶向下”听起来有些与众不同，毕竟在其他理论的教学上，往往遵从着“自底向上”的教学方式，如经典的“数字电路—计算机硬件—操作系统”路径。然而在计算机网络知识的教学上，“自顶向下”是个很不错的教学方法，读者并不从枯燥的硬件知识学起，而是一开始接触应用层（HTTP），然后一步步学到硬件知识，这可以更直观有趣地学习计算机网络的知识。



可以考虑不读它们。

如果你觉得《自顶向下方法》这本书太过硬核（其实你不该这么觉得，毕竟本书单之前每本书都比它厚或差不多厚），你可以考虑读一读一本小书《网络是怎样连接的》，相比之下这本书更有趣味性一些，且基本也是按自顶向下的方式叙述的（严格来说是按照端到端的方式）。然而既然《网络是怎样连接的》这本书在本书单中被提到了，你也应该清楚这不是什么“通俗科普读物”，其实也是带点硬核的，读起来仍需要一些耐心。

其他

也许有人会发觉这里似乎缺少了一些重要的东西。编译原理去哪了？算法分析去哪了？这些当然是重要的。然而就在我看来，这些并非是每个人都必要学习的。例如编译原理作为本科教学阶段综合难度最大的一门课，却是所有课程中实用度最低的一门，你也几乎很难在编译器开发之外的场景下找到它的应用，因而我不会在这里推荐任何有关的书籍。同理，算法分析甚至计算理论和信息论也是重要的，但我想除了理论计算机科学家，少有人需要掌握这方面的知识，因此我也不会在这些领域给出任何推荐书籍。

离散数学

另一门重要的课是离散数学，然而本书单并未包含离散数学相关的书籍推荐。虽然很多时候讨论经典计算机理论课程只提到计组、计网、操作系统、数据结构、数据库这些课程，然而离散数学的重要性丝毫不亚于这些课程。在西方的传统离散数学教学中，会涉及数理逻辑、集合论、图论相关的知识（在国内，离散数学还通常包含简单抽象代数），而这些知识无论是对于数字电路、数据结构抑或是计算理论的学习都是很有帮助的。

然而本书单并未包含离散数学的原因之一是其中很多内容已经渗入了各类课程中，例如许多经典的算法教材已经包含了本应在离散数学中学习的生成树概念，而数理逻辑的部分内容也常常被数字电路甚至计算机导论所包含，因此已经不很需要单独学习一门离散数学了（其实在西方的教学方式中，很多在国内被认为不属于离散数学的内容，如很大一部分算法知识，都包含在这门课程中，因此在西方离散数学显得很“有用”，而在国内就显得有些太过学术以至于无用了）。另一点是离散数学中大部分内容实际上是计算理论等课程的前置，而这些课程已经在本书单中删去，因此本书单自然也就不包含离散数学了。

如果有读者认为有必要打好一些离散数学的基础，那么可以考虑阅读左孝凌的《离散数学》，这是一本非常简洁的书，几乎用最少的语言较为清晰地描述了离散数学至少该教授的内容，唯一的缺点是出版时间距今已四十多年了，有些陈旧。有人会批评左孝凌这本书不够生动，但能在这么薄一本书中塞下能够让人读懂的解释与合适的习题，其实也实属不易了，不用过度苛求。

另外，有些人可能会推荐Rosen的《离散数学及其应用》，我个人不推荐这本，这本书的厚度远远超出了学习基础理论知识所需要掌握的，更像一本“百科”而非入门书。

SICP?

也许还有人会提到SICP——即《计算机程序的构造和解释》。这是一本非常经典的书，其中也有非常精妙的思想，你或许可以考虑在抽象能力遇到瓶颈时读一读。然而这本书不应该包含在这里提到的计算机理论学习路径中，也绝非必要的。我很难描述这是一本怎么样的书，我只能告诉他人这是一本关于“抽象”这一概念本身的书，至于这本书具体讲了些什么，以我贫乏的表达能力很难总结出来。

总之，SICP应当是一本在你对编程有了一些深入思考后阅读的书。有很多人将这本书抬到了不属于它的地位，经典的就是以“MIT就用SICP作为计算机科学专业的第一门课”举例，然而要知道



我个人认为，**SICP绝对不合作作为一本入门书**。它很重要，但它的重要性非常特别，它既没有被高估也没有被低估，但总有人将它放到不属于它的位置，这很不合适。至少就本书单来说，读者不需要考虑任何有关SICP的事，当前你只需要了解有这样一本书就可以了。

CSAPP仍然很值得读

最后，还记得CSAPP吗？这可能是这里提到的所有书中最值得读的，但没有直接包含在本书单中，因为这本书涉及领域太广，实在是难以归类。显然，书单中没有直接包含CSAPP并不意味着它不值得读，相反，你其实可以在任何时候翻开CSAPP读一读，甚至在已经学完这里提到的所有理论知识后，CSAPP也常常给人新的启发。

本书单的宗旨是总结出严格的“一条”路径。这条路径是循序渐进的、易于理解的，同时也应当是适用于绝大多数人的。本书单必然不可能照顾到少数领域的需求，例如量子计算领域、人工智能领域等。本书单只总结计算机这个大领域中几乎所有人都有必要掌握的知识。如果我像其他书单一样每个领域都推荐好几本书甚至十几本书，那恐怕只能让多数读者感到眼花缭乱。

本文赞数的高速增长实在是让我有些受宠若惊。本书单只是简单地将我自己的学习路径分享出来，希望能对其他同样困惑的人起到一些参考作用。我不得不承认，书单中提到的书中有几本我大概只读了四分之三，并未全部读完，但总体上每本书都是我认真翻阅过的，我不太愿意人云亦云地，大家说哪本书好我就推荐哪本书。而我确实看到了太多这样的书单，这也是我打算自己总结这样一份书单的原因。

评论区有人推荐一些我从未读过的好书，这也是很好的。正如我个人简介里所说，我当前只是一个普通的在读本科生，不可能把这些推荐的优秀好书都读一遍，有人能够补充完善书单，那自然是我乐见的。但我得说一声抱歉，我的精力注定了我不可能把这所有推荐的好书都认真翻一遍，而我确实不愿意把我自己都没读过的书放在书单里，所以它们中的很多或许注定要放在评论区里了。

另外，自然有一批国产书也是相当不错的，特别是近年来我感到国产专业书，无论是理论书还是技术书质量都有明显提高。现在已经不必迷信那些西方经典教材了，国内的几本书，例如邓俊辉的数据结构、陈海波的现代操作系统，其实很适合国内学生使用，它们不会有翻译问题，也不会因为地域差异而水土不服。只是为了贯彻本书单“只给一本书”的宗旨，我还是挑了一些西方的经典书籍放在这里。这其实也有我的一点私心在里面，因为我最初学习算法就是从Sedgewick的算法开始的，它第一次带我走出了面对一堆大黑书看天书一样的困扰，我对这本书还是有些感情的。

最后，作为一名普通本科生，我的见解想必有很大疏漏。如果有读者发现书单中有一些安排不妥当或是描述错误，也欢迎指出，我会尽快修正。

我从没想过我能在知乎上获得千赞，更没想到不是在某个回答中获得千赞，而是在获赞难度更大的知乎文章上获得千赞。我其实不认为这个有些随意的书单能够值得这么高的赞数。毕竟我只是个非常普通的本科生而已，如果有任何一位更有经验的老师或是在行业内深耕多年的工作者来组织类似的书单，应该都能比我做得更好。

我只是简单将我的学习路径分享出来，并且去除了一些我认为不好的、被试过错的参考教材或学习路径，给出了我认为比较适合的一条路子。什么路径最适合自己的肯定是因人而异的，因此我也只能



分享这个书单的初衷也很简单。我大学里最初进入当前专业时，曾在CSDN、知乎等平台上找过很多“计算机专业学生学习路径”，而知乎上这样的推荐更是数不胜数。我看到很多人都在推荐书单，在一个“计算机领域经典书籍推荐”的问题下，我看到有很多高赞回答甚至在一个回答中推荐了几十上百本“经典书籍”。然而这些书籍推荐似乎对我来说有些太多了，有相当多的书单甚至会在一个领域推荐好几本书，并称它们“一样好，各有各的特点”。我理解这是为了尽量客观，但对于最初的我来说，这真的只能徒增困惑。老实说，这些书单反而让人更迷惑了，并没有让我明白怎样形成一条合适的学习路径。

我身处计算机教育并不特别科学的中国，我明白既然进入了这个专业，那就一定要花大力气自己学习。同时，学校的一些理论课程也让我感觉颇为零碎，很多课程即使我期末考试成绩相当不错，我也不认为真的系统学懂了这门课。于是我一直在课余时间尝试着从那些西方的经典教材中获得更系统的认知。我参考了一些西方大学的教材，买了不少书拿来阅读对比。我承认由于时间有限，买来的很多书都只看了一部分，但出现在本书单里的书我不敢说全部一字不落从头到尾读完了，但四分之三以上总是有的。在这些反复的对比参考下，我将我读得最多，也认为最易懂的几本书挑出来，尝试形成一条比较合适的学习路径。

也许是我孤陋寡闻，我并未看到有书单真的能够给出一条具体准确且线性的“读书路径”。我几乎只看到太多书单杂乱无章地推荐一大堆“经典书籍”，有些甚至是极其过时的老书，让我很难不怀疑作者本身是否也没看过这些书。我也看到一些很好的对某领域相关书籍的推荐，但都较为零散，并未考虑形成一个系统的书单。于是在几天前，我尝试将我读过且认为最好的那些书单独挑出来，形成一个我认为相对合适的书单。我希望这个书单是具体的，不会让读者（尤其是选择困难症的读者）感到困惑的，同时也是尽量由浅入深的。于是，这一书单就出现了，我也很自然地想到了“Only One Book”是一个合适的，可以最好地描述本书单初衷的名字。

正如有人在评论区所说的，在某个领域多读几本书，分为入门、进阶和大师，或是找到四五本书相互补充其实更好。但我也很清楚绝大多数人没有这样多的时间能够埋头在某个具体领域的学习中，或许他们的钱包也不支持他们买这样多内容重复度很大的书（例如许多学生）。很多人只是打算系统地学一些理论知识，或许会打算稍微深入点，但再深入点，就很少有人会去做了。他们需要的正是类似这样的具体且线性的书单。我当初没有碰到过类似的书单（正如上面所说的，也许只是我孤陋寡闻），因此也就自己踩坑，尝试总结出一份自认为合理的书单分享出来，希望能帮到和当初的我相似的人。

最后，分享个我觉得很有趣也很简洁的“挑书小技巧”。说来巧合，这还是我在看《Go语言学习指南——惯例模式与编程实践》这本书的前言时看到的：

学无止境，无论刚接触编程语言的新手，还是将 Go 作为第 N 语言的王侯将相，通过阅读本书，都能够以全新的方式思考软件，体会并学到最地道的 Go 语言实践哲学。

第三篇：人生苦短，只读好书

吾真本 Thoughtworks 首席咨询师，《混沌工程：复杂系统韧性实现之道》译者

读书很花时间。由于要运行书中的代码，读编程语言的书就更花时间。对于难以实现时间自由的程序员来说，只有选择读好书，才能让睡眠的时间更多些。

“人生苦短，只读好书。”

我一般用下面的方法来寻找编程语言方面的好书。以 Go 语言为例。先用英文搜索引擎搜索 “books about golang”，然后从搜索结果中挑选一些近几年出版的书，之后再打开 amazon.com 网站，逐一查看这些书的评分星级和书评。如此就能选出好书。



知乎

首发于
"Only One Book"计划 (书单)

Language 的书评中，我得知 Go 能构建静态绑定的可执行文件。这样一来，就不会在运行时出现 DLL 或 .so 版本依赖不匹配的烦恼了。另外，一位读者在 *Network Programming with Go* 一书的书评中提到，她在读该书时会与蓝皮书和 *Learn Go with Tests* 两本书一起读，以便将来读更高阶的 *Distributed Services with Go*。根据她的提示，喜爱编程演练的我不仅找到了适合我的 *Learn Go with Tests* 一书，还从中找到了通过实例学习 Go 语言的网站 <https://gobyexample.com>。

下面是按照上述方法，我找到的 7 本 Go 语言的好书。先按读者评分星级，然后按书评数量，最后按出版日期排序。书评数量越多，说明读者关注度越高。

第 7 名: *Network Programming with Go*，作者 Adam Woodbeck，2021 年 3 月 25 日出版，4.6 星，19 条书评。

第 6 名: *Distributed Services with Go*，作者 Travis Jeffery，2021 年 4 月 6 日出版，4.6 星，22 条书评。

第 5 名: *Cloud Native Go*，作者 Matthew A. Titmus，2021 年 5 月 11 日出版，4.6 星，22 条书评。

第 4 名: *Head First Go*，作者 Jay McGavren，2019 年 5 月 7 日出版，4.6 星，106 条书评。

第 3 名: *Mastering Go, Third Edition*，作者 Mihalis Tsoukalos，2021 年 8 月 31 日出版，4.7 星，9 条书评。

第 2 名: *The Go Programming Language* 蓝皮书，作者 Alan Donovan 与 Brian Kernighan，4.7 星，919 条书评。

第 1 名: *Learning Go* (本书)，作者 Jon Bodner，4.8 星，71 条书评。

我读了本书英文版前 9 章，感觉行文流畅，讲解实用，就像在听一位开发经验丰富的高手在我面前娓娓传授 Go 语言的武功秘籍。作者不仅讲清楚了“是什么”，还会告诉你“为什么”，以及“什么时候该这样用，什么时候该那样用”。

作为曾经翻译过几本英文书的我，浏览本书的中文译稿时，同样感觉十分流畅。我深知译者和编辑的辛苦，所以在此对他们的辛勤付出表示由衷感谢。

编程圈和图书圈的流行趋势就像时尚圈的流行趋势一样，日新月异。几个月之后，上述好书列表必然会发生变化，但我希望上述方法对你有所帮助，也希望你我一起抓紧时间阅读本书，不要辜负了这本排名第一的好书。

希望这能对一些读者有所帮助。

编辑于 2022-07-20 23:18

计算机 理论计算机科学

▲ 赞同 1196 ▼ ● 39 条评论 ↗ 分享 ♥ 喜欢 ★ 收藏 📄 申请转载 ...



评论千万条，友善第一条

39 条评论

默认 时间



火龙果可好吃了 🍉

操作系统方面个人认为operating system concepts才是最好的入门书
07-17 · IP 属地四川

● 回复 👍 11



火龙果可好吃了 🍉 · Snowflyt

是的，这本书最新版很好，并且没有中译本，前一版的中译本已经被很多人拉帖子喷了。操作系统导论的中译本很用心。
07-17 · IP 属地四川

● 回复 👍 4



知乎


首发于

"Only One Book"计划 (书单)

07-17 · IP 属地江苏

回复 1


展开其他 1 条回复 >

- 

人体描边大师

20岁的你, 怎么如此优秀 😊


07-18 · IP 属地美国

回复 4
- 

Mbappe

操作系统其实我推荐陈海波老师的《现代操作系统》, 内容很新, 语言比翻译书更好理解, 而且配套实验指导也做得很好。


07-18 · IP 属地贵州

回复 2
- 

Snowflyt 作者

感谢推荐

07-18 · IP 属地江苏


回复 1
- 

勃拉兔

银杏书更适合入门

07-20 · IP 属地海南

回复 赞



评论千万条, 友善第一条

07-17 · IP 属地北京

回复 3

- 

张一

下了一本第三版 "Introduction to Computing Systems", 翻了一下, 又是教授自己发明了一个玩具ISA, 800页的大书, 里面数字电路的知识对于中国学生也有点多余, 数字电路课程都学过了。个人感觉直接看 "硬件软件接口" 就可以了, RISC-V或者ARM版都可以, 真实世界的ISA, 实用性强多了, 也有更多体系结构方面的知识。

07-18 · IP 属地上海

回复 1
- 

snoopy

这本书的定位是大一第一门课, umich、uiuc、utaustin、中科大、深圳大学等很多学校采用, 对于建立系统观还是有一定价值的, 可以了解一下教材作者的信息。

07-22 · IP 属地山东

回复 赞
- 

Snowflyt 作者

的确如此。但对于没有学过数字电路的自学者来说, 看一下概论应该会有些帮助, 后面半本书关于C/C++的东西也可以适当跳过。第三版确实个人感觉比起第二版增加得有些过分了, 七八百页是太多了些, 不过这也不一定算是大书, 或许翻译成中文也就不到四百页的样子。至于科班生, 大一或大二上学期看看挺合适的, 再往后就没必要了。

07-18 · IP 属地江苏

回复 赞
- 

Dian

作者对 operating system Three Easy Pieces 的介绍帮我解决了作业怎么写的难题

07-17 · IP 属地陕西

回复 2
- 

软软萌萌二次元

one one book, five one book

07-17 · IP 属地美国

回复 1
- 

何不食肉糜

👍厉害, 计算机组成原理那里的书籍推荐可以看出真的是有心人。一些个人经历:
如果没有学过数字电路, 还是不推荐先读《计算机组成与设计:硬件/软件接口》。倒不是说读不懂, 这本书的附录再加上网上找的资料已经够用了。但是书里基本是组成原理和体系结构混杂讲, 很容易在 "我觉得我懂了" "我觉得又不懂了" 来回切换, 定力和心态不够容易读不下去。

07-22 · IP 属地贵州

回复 1
- 

HahaLan97

知乎

首发于

"Only One Book"计划 (书单)

- 07-19 · IP 属地德国
- 

s-quark

“计算机理论”

07-18 · IP 属地美国

回复

1
- 

古德里安

Only One Book你发明的? 计算机领域不适用

07-19 · IP 属地德国

回复

赞
- 

简谐振子

Stanford在94年出过一本Foundations Of Computer Science C Edition, 作者是Aho和Ullman, 那本挺好的。机工翻译的那本也就给外行看看还行。至于SICP就像新手过渡期的装备, 挺鸡肋的。当闲书看不如看看The Little Schemer/ The Reasoned Schemer。如果想学点符号演算可以看CommonLisp: A Gentle Introduction to Symbolic Computation, 但是我也不推荐用Lisp入门FP, 因为它的缺点非常多。

07-20 · IP 属地韩国

回复

赞
- 

埋头养zhu

很系统, 比那种上来就拿大部头堆砌的好多了!

07-19 · IP 属地上海

回复

赞
- 

x796c

帆船书感觉写的有点过于娓娓道来, 对有基础的科班生来说没有必要如此掰开揉碎。我比较推荐奶牛书, 简洁易读且面面俱到。

07-20 · IP 属地陕西

回复

赞
- 

Saterblues

也许一本书在同类型的技术书中是经典, 但是不一定适合初学者了解该领域。最佳的应该是同领域5本书, 2本入门, 2本中级, 1本经典。入门方式多种, 经典唯一

07-19 · IP 属地湖南

回复

赞
- 

Lexy Berry

可以, 作者很专业, 除了操作系统外基本都是本科上课用过的课本, 而且个人感觉OStep确实更棒

07-19 · IP 属地美国

回复

赞
- 

Althusser

黑皮书, 黑皮书, 黑皮书, 还是黑皮书

07-19 · IP 属地广西

回复

赞
- 

Snowflyt

作者

就这些领域不推荐黑皮书推荐什么呢.....我自己也挺不喜欢机工社的黑皮书系列的, 感觉纸张太薄, 翻译也很一般。只能说机工社入局早, 大多数经典的代理出版版权都被占了。这些领域不用黑皮书也可以, 电工社和邮电也有引入一些“通俗版”的入门书, 但总还是不那么深入和系统。至于国内作者写的书, 也有很好的, 但我确实没读过几本, 就不推荐了。

07-20 · IP 属地江苏

回复

赞
- 

方应杭

我比较喜欢 3 本书, 一本入门, 一本进阶, 一本大师。

07-18 · IP 属地湖北

回复

赞
- 

福田

感觉这个是出名又不会读系列

07-18 · IP 属地新疆

回复

赞
- 

郭帅

是你不读吧

07-20 · IP 属地浙江

回复

赞
- 

张一

知乎

首发于
“Only One Book”计划（书单



rushzzh

我在大学的时候这门课也是这么缩的，有点搞

07-24 · IP 属地北京

回复 赞

点击查看全部评论 >

文章被以下专栏收录



“Only One Book”计划（书单
一份简洁的书单，每个细分领域/方向只推荐一本好书

