

FAQ第一期 | 学与不学，知识就在那里，不如就先学好了

2019-07-31 徐文浩

深入浅出计算机组成原理

[进入课程 >](#)



讲述：徐文浩

时长 10:20 大小 9.47M



你好，我是徐文浩。专栏上线三个多月，我们已经进入后半段。

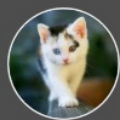
首先，恭喜跟到这里的同学，很快你就可以看到胜利的曙光了。如果你已经掉队了，不要紧，现在继续依然来得及。

其次，非常感谢同学们的积极留言，看到这么多人因为我的文章受到启发、产生思考，我也感到非常开心。因此，我特意把留言区中非常棒的、值得反复阅读和思考的内容，摘录出来，供你反复阅读学习。

有些内容你可能已经非常熟悉了，但是随着工作、学习经验的不同，相信你的理解也会不一样；有些内容可能刚好也是你的疑问，但是你还没发现，这里说不定就帮你解决了。

今天第一期，我们先来聊聊，“学习”这件事。我准备了五个问题，话不多说，一起来看看吧！

Q1：“要不要学”和“学不会怎么办”系列



llody

非科班出生，对于计算机组成原理本就是一片空白。这个领域其实是又爱又怕的地方，想了解，又怕理解不了，浪费时间。

—— 写于 2019年04月26日

引自：深入浅出计算机组成原理

开篇词 | 为什么你需要学习计算机组成原理？

识别二维码打开原文
「极客时间」App



专栏已经更新三个多月了，但是我估计很多人还是停留在前面 3 篇的学习上（我相信，一定有的）。

我观察了一下，其实很多人并不是真的学不会，而是“不敢学”，往往还没开始就被自己给吓到了。很多优秀的人，并非真的智商有多么高，而是他们敢于尝试，敢于突破自己的舒适区。

所以说，学习底层知识或者新知识的第一点，就是要“克服恐惧”，其实大部分东西上手了都不难，都很有意思。就像《冰与火之歌》里面，水舞者教导艾莉亚时的情况一样，“恐惧比利剑”更伤人。破除对于基础知识“难”的迷信，是迈向更高水平必经的一步。

“组成原理可以算是理解计算机运作机制的第一门入门课，这门课的交付目标就是让科班的同学们能够温故而知新，为非科班的同学们打开深入学习计算机核心课程的大门。”这是我在专栏刚上线的时候给一个同学的留言回复，现在拿过来再给你说一遍。

另外，大家在学校里学这些课程的时候，都会遇到一个问题，那就是理论和我们的编程应用实践离得比较远。在这个专栏里，我的目标是让大家能够更“实践”地去学习计算机组成原理。

所以，这门课我的目标就是尽量讲得“理论和实践相结合”，能和你的日常代码工作结合起来。让非科班的同学们也能学习到计算机组成原理的知识，所以在深入讲解知识点之外，我会尽量和你在开发过程中可能遇到的问题放到一块儿，只要跟着课程的节奏走，不会跟不上哦。（跟到这里的同学可以在留言区冒个泡，给跟不上的同学招个手，让他们放心大胆看过来。）

我自己在大学的时候也不是个“好学生”。现在回头看，我自己常常觉得大学的时候没有好好读书，浪费了很多时间。常常想，当时要是做了就好了。当时，不是就是现在么？学或者不学，知识就在那里，不如就先学好了啊。

Q2：“计算机组成原理”和“操作系统”到底有啥不一样？



不专注的linuxer

显然，在硬件和软件之间需要一座桥梁，而“计算机组成原理”就扮演了这样一个角色，它既隔离了软件和硬件，也提供了让软件无需关心硬件，就能直接操作硬件的接口。

这里的计算机组成原理改成操作系统是不是更好，而且感觉这句话说的就是操作系统呀

—— 写于 2019年04月25日

引自：深入浅出计算机组成原理

开篇词 | 为什么你需要学习计算机组成原理？

识别二维码打开原文
「极客时间」App



其实操作系统也是一个“软件”，而开发操作系统，只需要关注到“组成原理”或者“体系结构”就好了，不需要真的了解硬件。操作系统，其实是在“组成原理”所讲的“指令

集”上做一层封装。

体系结构、操作系统、编译原理以及计算机网络，都可以认为是组成原理的后继课程。体系结构不是一个系统软件，它更多地是讲，如何量化地设计和研究体系结构和指令集。操作系统、编译原理和计算机网络都是基于体系结构之上的系统软件。

其实这几门基础学科，都是环环相扣，相互渗透的，每一门课都不可能独立存在。不知道你现在是否明白这几门基础学科的价值呢？

Q3：“图灵机”和“冯·诺依曼机”的区别

课后思考

计算机行业的两大祖师爷之一，除了冯·诺依曼机之外，还有一位就是著名的图灵（Alan Mathison Turing）。对应的，我们现在的计算机也叫图灵机（Turing Machine）。那么图灵机和冯·诺依曼机是两种不同的计算机么？图灵机是一种什么样的计算机抽象呢？

首先，先回答一下这道题本身。有些同学已经回答的不错。我把他们的答案贴在这里。你可以看看跟你想的是不是一样。

Amanda 同学：

两者有交叉但是不同，根据了解整理如下：

图灵机是一种思想模型（计算机的基本理论基础），是一种有穷的、构造性的问题的求解思路，图灵认为凡是能用算法解决的问题也一定能用图灵机解决；

冯·诺依曼提出了“存储程序”的计算机设计思想，并“参照”图灵模型设计了历史上第一台电子计算机，即冯·诺依曼机。

图灵机其实是一个很有意思的话题。我上大学的时候，对应着图灵机也有一门课程，叫作“可计算性理论”，其实就是告诉我们什么样的问题是计算机解决得了的，什么样的问题是它解决不了的。

在我看来，图灵机就是一个抽象的“思维实验”，而冯·诺依曼机就是对应着这个“思维实验”的“物理实现”。如果我们把“图灵机”当成“灵魂”，代表计算机最抽象的本质，那么“冯诺伊曼机”就是“肉体”，代表了计算机最具体的本质。这两者之间颇有理论物理学家和实验物理学家的合作关系的意思，可谓是一个问题的两面。

冯·诺依曼体系结构距今已经几十年了，目前，我们还没有看到真正颠覆性的新的体系结构出现，更多地是针对硬件变化和应用场景变化的优化。但是过去几年随着深度学习、IoT 等的发展，体系结构又有了一波新的大发展，也许未来会有新的变化呢，我们可以拭目以待。

Q4：工作多年，如何保持对知识清晰、准确的认识？



Only now

编码多年，这些知识本科是掌握很扎实的，甚至初参加工作还能说的头头是道，但是现在几乎都模糊了，不知道老师是怎么一直保护知识清晰的？

—— 写于 2019年04月26日

引自：深入浅出计算机组成原理

02 | 给你一张知识地图，计算机组成原理应该这么学

识别二维码打开原文
「极客时间」App



我之前跟很多人聊过，发现工作很多年之后的工程师，在计算机科学的基础知识上，反而比不上很多应届的同学。我总结下来，大概有这么几个因素。

首先，很多工程师只是满足于工作的需求被满足了，没有真的深入去搞清楚一个问题的原理。从网络上搜索一段代码，复制粘贴到自己的程序里，只要能跑就认为问题解决了，并没有深入一行行看明白每行代码到底是做了什么，为什么要这么做。

比如说，我们现在要提升 RPC 和序列化的性能，很多人的做法是，找一个教程用一下 Thrift 这样的开源框架，解决眼下的问题就完事儿。至于，Thrift 是怎么序列化的，每一种里面支持的 RPC 协议是怎么回事儿，完全不清楚。其实这些开源代码并不复杂，稍微花点时间，搞清楚里面的实现细节和原理，你对二进制存储、程序性能、网络性能，就会有一个更深刻的认识，之后遇到类似的问题你就不会再一问三不知，久而久之你的能力就会得到提升。

其次，读书的时候我们认为一个东西掌握扎实了，有时候其实未必。很多人估计都有感受，像计算机这类实践性比较强的专业，书上所学和真正实践中所用完全是两码事。背出计算机的五大组成部分，似乎和我们的实际应用没有联系，但是在实际的系统开发过程中，无论是内存地址转换使用的页表树这样的数据结构，还是各个系统组件间通过总线进行通信的模式，其实都可以和我们自己的应用系统开发里的模式和思路联系起来。

至于究竟该怎么去掌握知识，其实没有什么特别好的方法。我就说说我一般会怎么做，一方面，遇到疑难问题、复杂的系统时，必须要用更底层更本质的理解计算机运作的方式，去处理问题，自然会去回头把这些基础知识捡起来；另一方面，时不时抽点时间回头看看一些“大部头”的教科书，对我自己而言，本身就很有自我满足感，而这种自我满足感也会促使我不断去读它们，从而形成一个良性循环。

Q5：六个最实用的、督促自己学习的办法

看到很多同学在留言里分享了自己学习方法，我看了也非常受益，我把这些方法筛选总结了一下，又结合我自己的学习经验，放在这里分享给你。

1. 好奇心是一个优秀程序员必然要有的特质。多去想想“为什么是这样的”，有助于你更深入地掌握这些知识点。
 2. 先了解知识面，再寻找自己有兴趣的点深入，学习也是个反复迭代的过程。
 3. 带着问题去学习是最快的成长方式之一。彻底搞清楚实际在开发过程中遇到的困难的问题，而不是只满足于功能问题被实现和解决，是提升自己的必经之路。
 4. “教别人”是一种非常高效的学习方式，自己有没有弄清楚，在教别人的过程中，会体会得明明白白。
 5. 每个月给自己投资 100-200 块在专业学习上面，这样花了钱，通过外部约束，也是一个让自己坚持下去的好办法。
 6. 坚持到底就是胜利🏆。把学习和成长变成一种习惯，这个习惯带来的惯性会让你更快地成长。
-

好了，到这里，我们第一期答疑就要结束了。这次我主要和你谈了谈“学习”这个话题，不知道你有什么感受呢？你还想听我和你聊什么专栏之外的话题呢？

欢迎积极留言给我。如果觉得这篇文章对你有帮助，也欢迎你收藏并分享给你的朋友。对了，看到这里的同学，记得在留言区给后面的同学招个手啊：)




深入浅出计算机组成原理

带你掌握计算机体系全貌

徐文浩 bothub 创始人



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 41 | 理解内存（下）：解析TLB和内存保护

下一篇 42 | 总线：计算机内部的高速公路

精选留言 (47)

 写留言



Timmy

2019-07-31

很高兴跟着老师课程保持学习，早上在地铁上学到新的知识都有一种原来如此的感觉，不知不觉到站了还意犹未尽，老师的授课讲解通俗易懂，且开篇带着问题学习事半功倍，还有细致的图解一目了然，使整个学习不那么枯燥老师也是费心了。准备二刷输出总结文档，加油，感谢😊

展开 ∨



3



missingmaria

2019-07-31

35+大龄，外行，纯靠兴趣自学编程，这门课是第一次全程跟上了，觉得老师每一篇的体量安排很合理，上班路上读正好



3



pebble

2019-07-31

订阅了很多专栏，有的掉队了，就这个追剧一样等更新



1



任雪龙

2019-07-31

一只跟着，开心，没掉队，很期待每次更新



1



胖胖胖

2019-07-31

感觉老师的课降低了老师推荐的书的阅读门槛，看文课了再去看计算机系统和软硬件接口几本书会感觉轻松很多



1



度衡

2019-07-31

非科班，每篇都看了，但是大概就掌握了30-50%。比较佩服老师讲解这门课程时问题的解决思路，先看组成要素，从某个元素切入去解决问题，解决问题方案中又存在新问题，然后又去优化解决方案，循序渐进又知其所以然。

展开 ▾



1



gigglesun

2019-07-31

专栏非常的精炼，阅读专栏，提出一些问题，然后通过扩展阅读解决这些问题，挺好



1



小明见万里

2019-08-22



幻月剑

2019-08-21

跟到这里打个卡，工作的原因需要学习其他东西，所以这个优先级不高但是并没有放弃，断断续续的还是在学，就是只是看一遍作用不是特别大，也不牢固，但是呢，看过就比没看过强，学过就比没学过强，继续学习

展开 ▾



allen

2019-08-05

目前看到的质量最高的一门课了



leslie

2019-08-02

其实我是一边跟着老师的课在学计算机组成原理，一边在跟刘超老师的操作系统，还在用剩下的时间去重学算法。

忙不过来：很充实，结合着一起学一起反思还是蛮有收获，相互理解相互对比-蛮痛苦蛮自虐的。这行从程序员-数据库开发-DBA-系统运维：两门课接近的开课时间刚好让我能整体上梳理一下从软件开发-数据库-系统的整体的知识体系，交叉相互理解就能站在更高...

展开 ▾



鱼向北游

2019-08-01

从开始跟到现在 大龄程序员 非科班出身 这些基础课其实以前已经自学过一遍 但是工作这么多年再次学 还是有不同的领悟 感谢老师



Martin □□□□□

2019-08-01

非科班出生,学习到这里,感觉基础真的很重要,特别是老师的一些实践和理论的结合部分都精读了好几遍,过几天回头看的时候每次都会去看下老师的知识地图,希望自己做到心中有图花点时间就能画出整个知识全貌,平常工作其实很少时间看文章,更多的是做了知识储备,感谢老师。

展开 ▾



安迪

2019-08-01

非科班，之前读过csapp，其中比较难理解的地方通过老师的讲解很容易就明白了。学习是不断迭代的过程，不懂的地方多看几遍，反复琢磨就会豁然开朗。再有些不懂的地方可以放一放，说明是由一部分知识欠缺，等哪天突然看到相关的知识就会豁然开朗。
课程学到现在，受益匪浅！

展开 ∨



Feng.X

2019-08-01

非科班出生，有更必读，没有掉队。

老师的专栏写得简明易懂，循序渐进，非常友好。

说起难读的课程，我觉得还属隔壁的趣谈操作系统，那才是艰难，各种数据结构，各种封装调用...

展开 ∨



明翼

2019-08-01

打个卡，大龄程序员跟着老师过一遍，虽然吸收30%.好在没掉队，以后再反复学习吧



jji

2019-07-31

真正深入浅出的一门课程



空知

2019-07-31

Mark~



小海海

2019-07-31

老师我也一直在，从专栏推出就一直跟着，感觉大学都没这么认真学过，配合隔壁的Linux操作系统学习起来更有感觉，也更能把知识点串联起来，不过操作系统的课消化得不好，手动哭脸

展开 ▾



赵阿海

2019-07-31

打个卡。非科班出身，虽然每一节不能说完全掌握，但还是让我有所收获，非常感谢老师的讲解。我觉得每一节后面的扩展阅读是很有用的，准备学完专栏看一下相关书籍。另外，我觉得不应该想着看一遍就全都掌握了，要时不时复习一下前面的内容，养成持续学习的习惯，慢慢来，与时间做朋友，积少成多。

展开 ▾

