

## ## 后台开发应该读的书

根据我的经验来谈一谈，先介绍一下我的情况，坐标上海，后台开发（也带团队了），某知名互联网公司。

目前主流的有C++和JAVA，C++我的经验稍微多一点。我就说说我关于C++方面的学习经验。如果您是学生，临近毕业，没有那么多时间读许多书，可以按下面列举的重要程度来参考。

- 首先，我觉得你应该好好准备算法和数据结构，做到常见的算法和数据结构知识点都能非常熟悉，这样的话你毕业求职的时候可以\*\*轻松拿一些大厂（BAT等）的offer\*\*。我本人非科班出身，一毕业之后各种摸爬滚打。一毕业去\*\*大厂\*\*个人觉得有两个\*\*好处\*\*，第一，你的收入会比一般的小公司高很多，小公司招人要求相对低一些，薪资给的也少很多，它们是实实在在招能干本职工作活儿的人。第二，你的起点也会比一般进入小厂的同学高。我这里并不是歧视小厂，只是说一种普遍的情况。我本人也是从小厂一路过来的。这里我是强调算法和数据结构的重要性。尤其是应届生求职，更应该去好好准备一下这个，因为这个东西是原理性的基础。企业在面试应届生时不会过分要求项目经验和各种操作系统原理、网络通信原理之类的东西，而\*\*唯一能考察一个人的基本功的就是这个了\*\*。我是社招进大厂，基本上算法和数据结构这类问题问的比项目经验本身要多许多。但是社招又不太一样，因为除了要准备算法和数据结构以外，还得准备有项目经验、了解操作系统原理、熟悉网络通信、了解数据库、熟悉要求的各种开源框架和技术等等，实在太多了，即使再怎么准备也不一定能一举拿下。相反，应届生基本上只要好好准备算法和数据结构的东西，大学其他专业课学的不是太差，这基本上就是\*\*进大厂的捷径\*\*。图书方面，你可以使用你们计算机专业的相关教材，也可以使用《数据结构与算法分析:C语言描述》《算法导论》这一类严谨的教材，当然，平心而论我是不敢推荐《算法导论》的，因为这一本书实在是太大部头了，没有好的数学知识，真的很难啃。如果想看一下比较幽默轻松类的书，可以看看程杰的《大话数据结构》。

- 其次，如果你学有余力，可以看看操作系统原理方面的书籍，当然也可以使用你们的教材，我这里推荐一本我看过的吧，Tanenbaum.A.S《现代操作系统》，Tanenbaum是荷兰人，也是Linux之父Linus Torvalds操作系统方面的启蒙老师。当然，我的建议是这种书毕竟流于理论知识，也不一定要看完，但\*\*一定将一些基础概念，如进程线程内存模式等基础概念看懂理解\*\*。\*\*你如果还有时间强烈推荐看看俞甲子的《程序员的自我修养:链接、装载与库》，这本书同时涉及到了Windows和linux两个操作系统平台，用各种辅助工具剖析了程序从源码到二进制文件再到装载到进程地址空间里面的各个细节，甚至连进程地址空间中的堆结构、栈结构也分析得清清楚楚，同时也分析了C Runtime（CRT）、glibc这样的操作系统接口库的原理和执行逻辑，是一本实实在在难得的帮你实战操作系统原理的一本好书。我特别喜欢这个书中序言的一段话：

> “CPU体系结构、汇编、C语言（包括C++）和操作系统，永远都是编程大师们的护身法宝，就如同少林寺的《易筋经》，是最为上乘的武功；学会了《易筋经》，你将无所不能，任你创造武功；学会了编程“易筋经”，大师们可以任意开发操作系统、编译器，甚至是开发一种新的程序设计语言！”

- 再次，你\*\*学这些东西是为了将来实践并有产出的，而落实这个产出的东西就是编程语言\*\*，如果是\*\*入门\*\*，我\*\*首推C/C++\*\*。你只有熟练使用一门编程语言，你才能将你的想法变成现实。注意这里我把C和C++放在一起，但是严格意义上说，C和C++还是有点区别的，但是除了语法上的一些细节差异，基本上可以认为是相通的。个人觉得C语言是所有想成为高手最应该使用的入门语言，不要和我说现在很火的python、go这类语言，“玄都观里桃千树，尽是刘郎去后栽”。我这里也推荐一本C语言方面的图书吧，有兴趣的可以参考一下：《C语言程序设计:现代方法》。至于谭浩强的书就不要提了，还有就是大部头的《C++ Primer》，它虽然是一本好书，但实在是太大部头了。语法层面的东西学会很快，stl库的东西需要实战，也不是翻这类字典一样的书就能很好地掌握的。当然，如果你想掌握好C++，《深度探索C++对象模型》是一定要看的。C++实际编码技巧还有另外一本非常好的书，介绍了常见的C++编码技巧《提高C++性能的编程技术》，建议C++开发的把书中说的技巧全部掌握。

- 接着说，我们再说说网络方面的，首先网络基础方面的书籍，我就没啥推荐了，现在很多计算机学院也开始使用《计算机网络:自顶向下方法》这本不错的教材，如果没有看过的可以看下。当然还是那句话\*\*你一定要看懂而不是看完\*\*。比如三次握手和四次挥手的细节，你一定要很清楚。然后你就可以找一本网络编程的实战书来看下，如果你没有使用任何socket api编程的经验，你可以看看韩国人尹圣雨写的这本《TCP/IP网络编程》，这本书从基础的socket api介绍到比较高级的io复用技术，有非常详细和生动的例子。如果你是初级水平，强烈建议看看这本书。网络编程的细节需要注意的地方实在太多了，这本书上都有介绍。很多人尤其是一些学生，写了一些可以相互聊天的小程序就觉得自己熟悉网络通信了，但是这类程序拿到互联网上或者离开局域网，不是连接出错，就是数据总是收发不全。我当年也是这么过来的，看看这本书，你就能明白许多网络故障的原因。等你有了一定的网络编程以后（熟练使用常见socket API），你可以看看游双的《Linux高性能服务器编程》，这本书给没有基础的人或者基础不扎实的人的感觉是，尤其是书的前三章，这本书怎么这么垃圾，又把网络理论书上面的东西搬过来凑字数，但是如果你有基础再按照书上的步骤在机器上实践一遍，你会发现，真是一本难得的、良心的书，桃李不言下自成蹊吧。如果你掌握了这本说上说的这些知识，你再看陈硕老师的《Linux多线程服务端编程》或者去看像libevent这样的开源网络库，你会进一步的得到提升。这也是我学习网络编程的一些经验和经历吧。注意这里有必要提一下：像UNP、APUE、还有《TCP/IP详解》这一类书，如果你将来不是专门做网络方面的工作或研究，其实是非常不建议抱着他们看的，因为部头太大，其次太多理论和Unix的东西，花的时间产出投入比很低的。

- 接着说，以上说的都是一些基础的东西。其实不管是什么开发，后台开发也不例外，你都是需要基于特定的操作系统的，这里不提Windows系统，单单拿linux操作系统来说，既然你选择做这个方面的开发，你需要熟悉这个操作系统平台提供的一些常用的API函数，网络通信方面上文已经说过，除了网络通信还有如操作文件、操作内存、字符串操作、进程线程系列、线程同步系列（如互斥体、条件变量、信号量）、管道等常用的各种API接口函数。这里的意思是，不是要你背诵记忆每一个接口函数的签名，而是\*\*你要知道何时该用

哪个接口，如何用，有什么注意事项\*\*。我入门的时候看的是Robert Love的《Linux系统编程》，熟悉这个人的应该知道，google的工程师，他还有另外一本非常有名的书《Linux内核设计与实现》。

- \*\*最后，我强调一下，如果你是快毕业的学生，面临着找工作的压力，应该以算法和数据结构为主。如果你是大一大二或研一这个阶段的学生，我上面推荐的书，你还是可以考虑好好咀嚼一下。标准是看懂而不是看完。\*\*

- 再补充一些我觉得要成为高手应该要掌握的东西，先说汇编。虽然第三代第四代语言越来越多，硬件性能越来越好。但是如果你熟练掌握汇编，你就比其他人多很多优势，你会能透彻地知道你写的每一行C/C++代码背后的机器指令的效率。无论是做安全工程还是自己技术提升上都是非常不错的。这里推荐一本王爽老师的《汇编语言(第3版)》，这本书不厚，语言通俗易懂，你也不用刻意去记忆，基本上当小说书看一下就能很快看完了。汇编实战类图书还有另外一本《老"码"识途:从机器码到框架的系统观逆向修炼之路》。我个人是非常喜欢这本书的。当年读这本书的时候，真的有一种“笑看妻子愁何在？漫卷诗书喜欲狂”的感觉。尽管那个时候连女朋友都没有—！

另外补充一些我学生时代看过的书吧，我本人是熟悉Windows和linux两个平台的开发，这也归功于我学生时代看过的一些经典书籍，可能有点跑题了，如果不介意，我可以和你说说：

- 《Windows程序设计》第五版（第六版以后，这个不再是用Windows Native API写C程序了，而是转到C#平台上了），这本书是中国第一代程序的windows启蒙书籍，你所看到的大多数桌面软件，如QQ，的开发者可能都是通过阅读这本书启蒙起来的。
- 《Windows核心编程》，这本书搞Windows开发的一定都知道这本书的分量。
- 《linux内核情景分析》毛德操老师的书，非常的实在，另外他写了一套关于Windows源码分析的书，这本书是基于开源的“Windows” ReactOS，书名叫《Windows内核情景分析》。
- 《编译系统透视：图解编译原理》，编译原理方面的实践书。
- 《编程之美》，关于面试的，主要是一些算法和逻辑思维题实战。
- 《重构：改善既有代码设计》，没有实际写代码经验不推荐看。
- 《程序员的修炼之道—从小工到专家》这本书\*\*特别推荐学生看一下\*\*，能大幅度地提高你实际编码的技巧和编码风格。
- 《代码整洁之道》同上
- 《大话设计模式》
- 《Windows PE文件权威指南》
- 《Java编程思想》
- 《Effective C++》系列
- 《80x86汇编语言程序设计教程》

总的来看，我\*\*学生时代主要是侧重基础知识来读书\*\*的。本科四年、硕士三年，多谢这些书帮助我成长，记得大学毕业的时候，我光读书笔记就有满满十个笔记本。

工作以后，也读了像redis、netty、分布式这一类书。但是那都是工作需要吧。由于我扎实的基础，当然也可能是因为运气成分吧吧，所以得到一些注重扎实的技术基础公司的青睐，给了目前这个阶段看起来还不错的薪资（当然可能还有人比我更厉害，那我这里就贻笑大方了，所以请不喜勿喷）。同时非常感谢我一路上遇到的公司和同事给我的技术上和生活上的帮助。薪资本身不能说明一个人是否成功，我码这么多字，\*\*希望广大的开发者注重基础，勿在浮沙筑高台。尤其是学生，你有大把读书的机会，一定要珍惜大学时光。毕竟工作以后，尤其是毕业后，面临着工作、家庭等各种问题，你可能再也没有心思和完整的时间去学习和提升了。\*\*所以前期的积累很重要，毕竟选择技术这条路，提高技术是升职加薪改善生活水平最直接的方法。最后用我学生时代看到一个技术前辈写的一首诗来结束吧：

- > 仗鼠红尘已是癫，
- >
- > 有网平步上青天。
- >
- > 游星戏斗弄日月，
- >
- > 醉卧云端笑人间。
- >
- > 七载寻梦像扑火，
- >
- > 九州谁共我疯癫？

以上是我的经历，我也曾迷惘和无助过。也有很多朋友找到我，希望我做一些经验分享和职业规划指导，有需要的小伙伴可以加我微信\*\*easy\_coder\*\*。