

## 【第一章：C++后台开发学习路线】第2节：如何拿下BAT：我的学习路线

[试读](#)

来自【《收割BAT：C++校招学习路线总结》】 | 307 浏览 | 0 回复 | 2019-11-28



特立独行MVP



专栏作者

[+关注](#)

### 前言

2019年对我而言必定是人生当中最重要的一年。因为实习的原因，我第一次体验了北京和上海的生活，匆匆且忙碌。这一年生活当中充斥着各种求职咨询，面试，做题，仿佛一夜回到了高三。不过，我是幸运的。从寒假实习开始到暑假实习再到秋招，一路以来我并没有体会到太多的挫折，一路高歌，最终也收获了我自己认为还不错的offer。也许是因为运气，也许是因为坚持，但是硕士这2年来我也经历了种种心酸和波折。我起初走过很多弯路，在无数个深夜里苦恼自己的无知和弱小，曾跌入低谷，连续数天的低沉，最终慢慢调整，告诉自己不能松懈，直到现在...

我希望可以把我从实习到秋招这一路以来的体会和经历分享给大家。作为一个还没有正式进入公司的在校学生，我在技术上还是一个实打实的“菜鸟”，但是作为一个应届求职者，我是“专业的”。本节我将从面经积累，价值观以及学习路线这三个角度介绍我的经验。

### 面经积累

我有一个习惯，就是记录和整理知识。从春招实习开始，我每一次面试都会录音，事后都会复盘。所有的面试记录，面试官问的问题，手写的算法题以及我个人面试的感受我都会记录下来。看过《深入理解计算机系统》的小伙伴都应该知道程序的执行是有空间局部性和时间局部性，其实面试官考察的知识点的范围也是如此。当你多次遇到同一个问题的时候就应该去把它记录下来。与其面对茫茫多的知识，不知从何入手，不如将自己遇到的问题一个一个“缓存”起来，建造一个属于自己的知识“字典”。当你的知识“字典”足够系统和完善，你就足以收获理想的offer。我的积累方案如下：

- 1.面试录音，及时复盘
- 2.事不过三，重复出现的问题单独归纳和整理
- 3.知识串联，通过关联的方式加强对琐碎知识点的记忆

### 价值观

我认为没有最好的学习方案，只有最适合自己的方案。

## 学习路线

实习和秋招的面试过程都是一场又一场的持久战，我在2年前也和和大家一样对面试畏惧，对学习恐慌，非常迷茫。很庆幸自己走过了这2年，在此我会将我自己的学习方法和面试经验总结给大家。整个学习路线分为以下几个部分：编程语言学习，操作系统学习，服务器编程学习，计算机网络学习，分布式数据库学习，其他方面学习。

### 1.编程语言学习

编程语言千万种，到底该学哪一种？

我认为语言不分优劣，只分喜好，只看需求。本硕这几年我学过C/C++,JAVA,Python,Matlab,Golang,Javascript,用来用去还是C++和Golang适合我，但是我也不排斥去使用其他语言。

学习编程语言第一步：上课。如果你是刚入学的小白，那就去好好上课吧，计科老师说的一定比我好。如果你已经错过时机或者没办法天天去教室里上课，那么网课是我最推荐的。

在哪里可以搜索合适的课程：

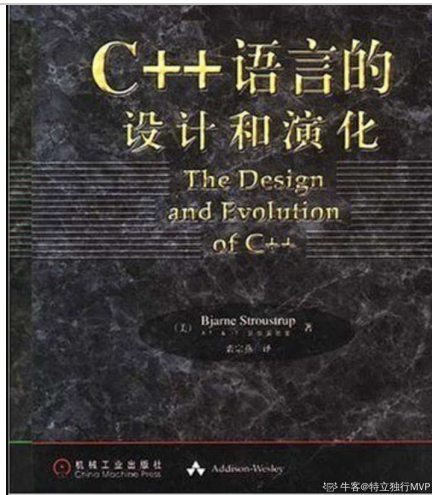
1. Coursera: 我最开始使用的渠道，当时看了很多北大的C++入门课程(比较简单现在只记得有个喜欢扔水瓶的老师)
2. B站: B站是非常好的渠道，如果你知道课程名直接去B站搜吧，肯定有up主上传了，点播放量最多时长最长的总没错，还有弹幕一起学，据说还不错。

我个人主要使用的语言是C++，所以我以C++学习为例：C++不容易学好，如果仅仅是学习语法或者写个算法题那一两周可能就可以上手。但是想用好C++还是需要很长时间的coding和源码阅读积累。我个人的方案是先看书，再实践，再看书这样的不断循环。道理大家肯定都懂，但是能够坚持做下来的太少了。

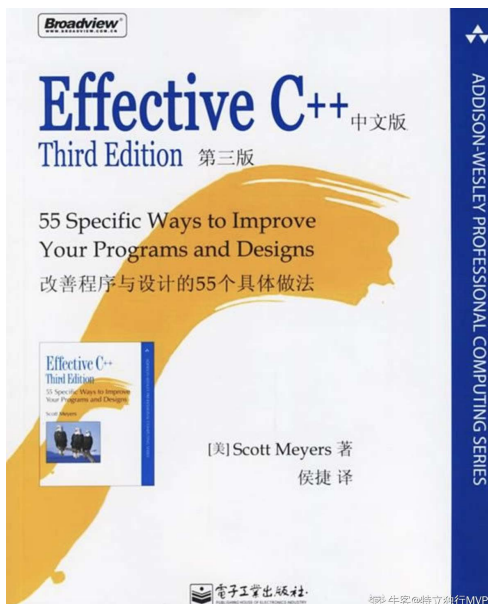
给大家推荐我看过的几本C++相关的书籍：

- 《C++语言的设计与演化》：可以用来理解C++的设计思路，知其所以然

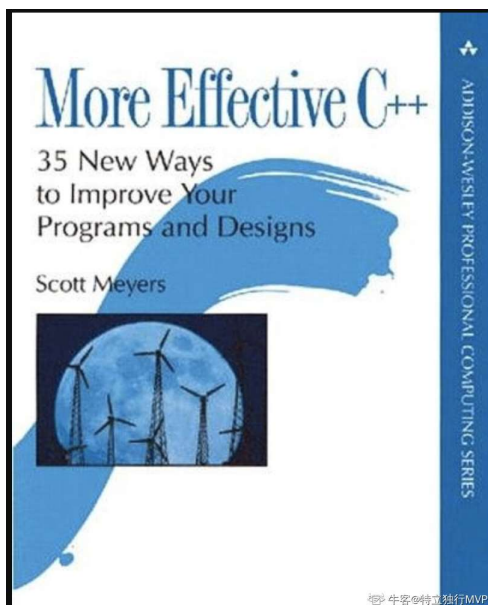
## ≡ 特立独行MVP

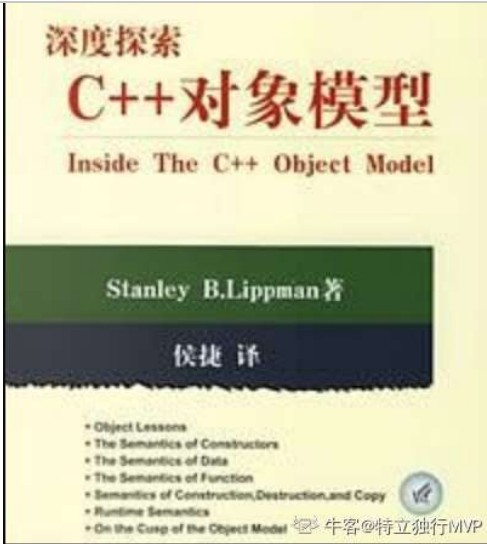


- 《Effective C++》：很多关于C++方面面试问题的出处

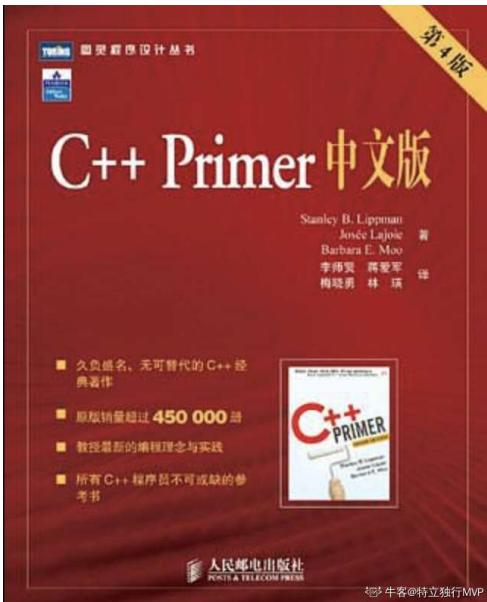


- 《More Effective C++》：《Effective C++》的进阶版

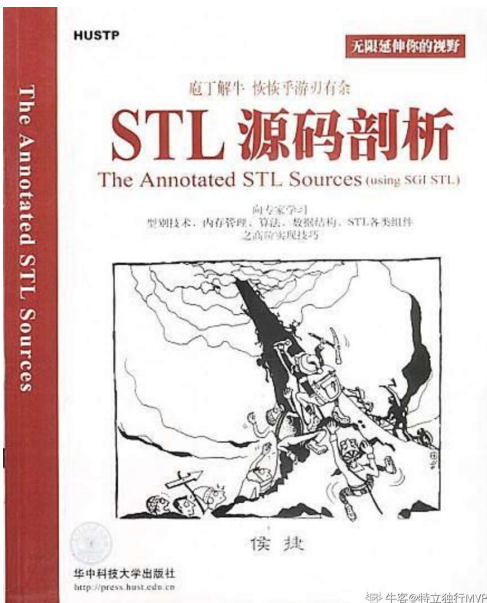




- 《C++ Primer》：很多人推荐看完这本书，但是这本书内容太多了，我建议当个字典，遇到问题去查。



- 《STL源码剖析》：C++代码的典型示例，还可以帮你理解数据结构的标准实现



## C++项目那么多，到底该看哪一个？

## 特立独行MVP

同，那么可以去找一些网络服务器相关的项目去练手，一方面可以加强对Linux操作系统的理解，又能增加网络知识，同时也可以锻炼C++ coding能力，一举多得。那么有哪些值得做的项目呢？

### 1.做别人推荐的好项目：

#### WebServer

我在实习前看的是大佬 (牛客id:健康成长天线宝宝啊) 的个人项目WebServer:<https://github.com/linyacool/WebServer>。基于他的项目和我后续看书的总结，在1-2个月的时间里我也实现了一个web服务器，这个项目是我简历上除了实习项目之外被问的最多的项目。

### 2.看STL源码：

STL源码，有各种C++高级技能的实现。推荐大家去看看STL里的快排实现源码，第一次看到真的很有意思。

### 3.开源项目：

下面是几个还不错的开源项目，如果当前有不错的基础可以去通过源码学习：

**libevent**: Libevent是一个轻量级的开源高性能网络库，采用事件触发，封装了以下三种事件的响应:IO事件,定时器事件,信号事件。

**Memcached**: Memcached 是一个高性能的分布式内存对象缓存系统，用于动态Web应用以减轻数据库负载。它通过在内存中缓存数据和对象来减少读取数据库的次数，从而提高动态数据库驱动网站的速度。

**Redis**: Redis是一个基于纯内存的存储系统，虽然是C写的但是值得一看。

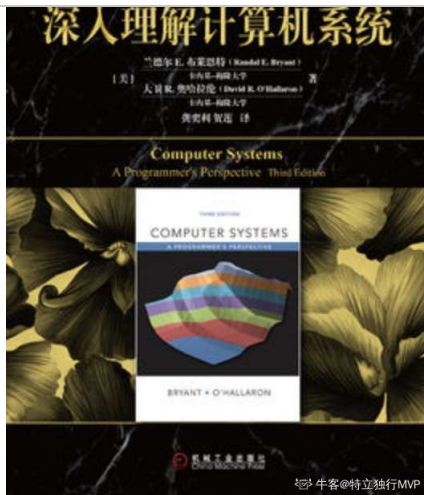
切记，项目当中实现功能只是一方面，测试和文档整理也很重要，项目的每一步都需要文档和数据的支撑。

## 2.操作系统学习

Linux发展到今天，早已超出我们的想象。如果不是想专门做操作系统这块的话，我的学习经验也许可以借鉴。首先我推荐大家看这本《深入理解计算机系统》：



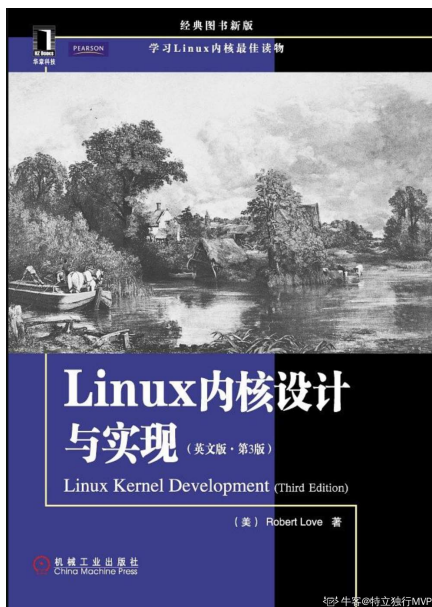
## ≡ 特立独行MVP

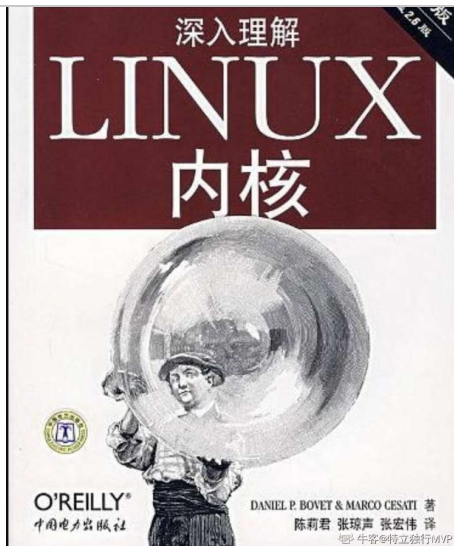


这本书十分经典了，每一次看都会有新的领悟，本书对于计算机体系结构的讲解非常细致，有很多非常有意思的Lab。这本书从程序执行的计算机角度开始，介绍了处理器的体系结构，程序的机器级优化，虚拟存储器，系统级IO，网络等等多个方面。推荐大家和 **CMU 15-213 CSAPP** 这门课一块看。

其次再推荐2本书：

《Linux内核设计与实现》：建议和操作系统原理的课本结合一起看





如果是面对秋招，我总结下来以下这些关键知识点是需要理解的：

1. 进程和线程之间的区别
2. 操作系统的调度算法
3. 进程间通信的方式和区别
4. 虚拟内存机制的作用
5. 缓存的作用以及缓存替换算法
6. 线程的实现方式
7. 虚拟文件系统

### Linux环境如何搭建

学习操作系统当然要配合Linux一起，过去一般建议大家装个双系统比如Ubuntu16.04。如果大家觉得双系统麻烦或者虚拟机慢，那么我还有以下2种方式让大家在几分钟之内就可以获得一个稳定的Linux开发环境：

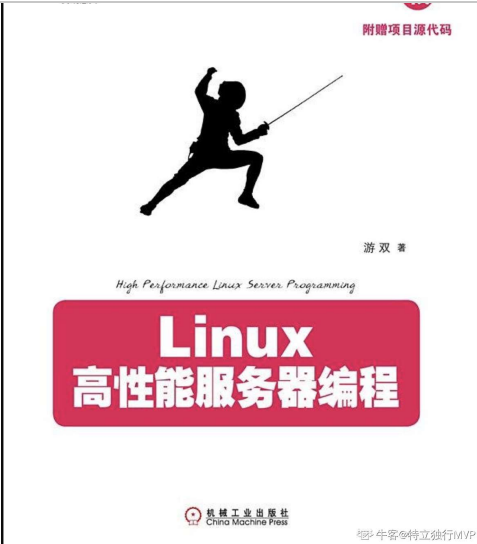
**方法1：** 买阿里云、腾讯云、华为云的服务器。买最便宜的，可以不用关机，远程ssh登录即可，无论手里是什么平台的系统，都可以一秒进入Linux，这也是我常用的方式。

**方法2：** 安装docker，花10min学习docker基本概念和操作，用docker安装Ubuntu16.04的镜像，然后进入容器。这个方法也是跨平台的，非常的简易，我会在后续专栏当中单独介绍docker的原理和使用方式。

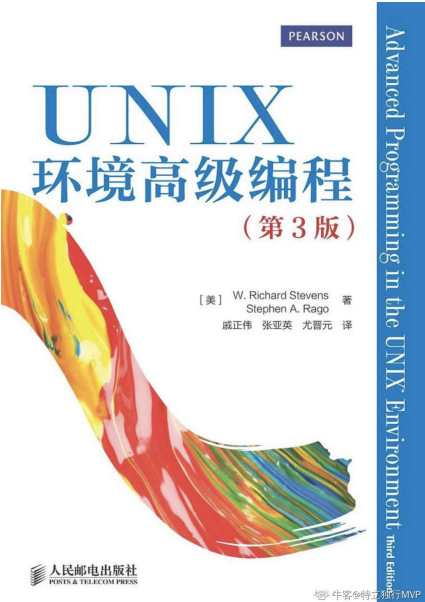
## 3.服务器编程学习

服务器编程其实就是建立在对linux系统调用的理解以及网络协议的理解上的。如果这两块的知识比较好的话就可以动手实现一些简单的服务器，然后对比别人实现的，压测一下性能，这样就可以作为一个个人项目。下面几本书我建议大家可以选看：

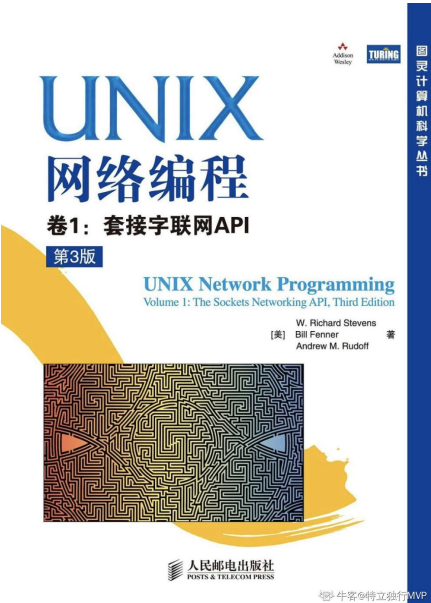
☰ 特立独行MVP



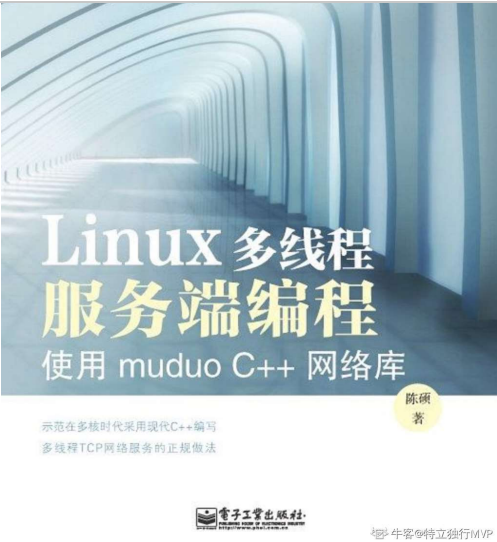
- 《UNIX 环境高级编程》：选看



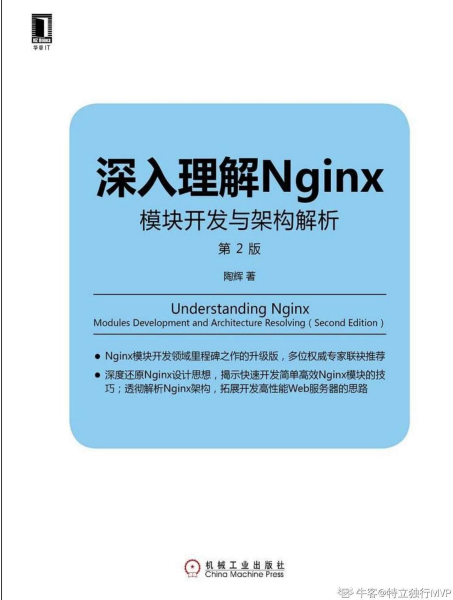
- 《UNIX网络编程卷1》：选看







- 《深入理解Nginx》：选看

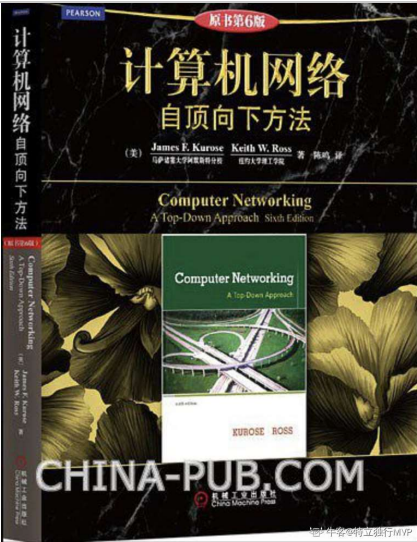


还有很多不错的开源库的实现，比如muduo库。造造轮子也能成为自己的一个项目，推荐大家在有时间的情况下去看看。

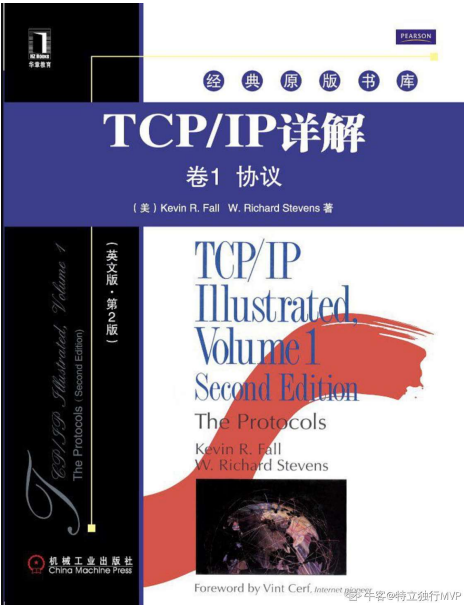
## 4.计算机网络学习

计算机网络这部分，我最开始看的是浙大的计算机网络课程。主要是专注于协议的原理和分层模型理解，因为计算机网络的原理还是比较琐碎的，建议大家多做些笔记和文档，不论对于记忆还是日后的复习都是很有用的。下面4本书值得一看：

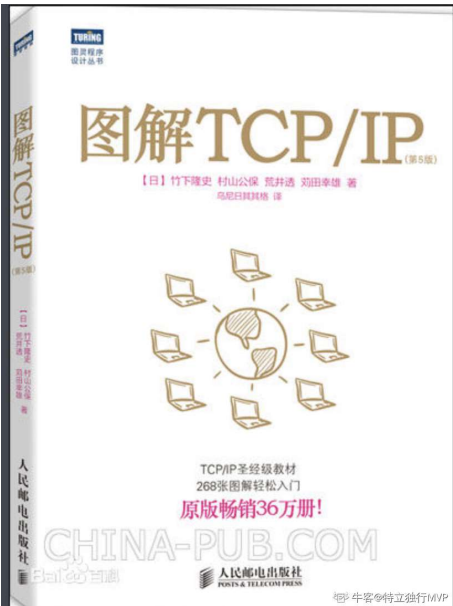
≡ 特立独行MVP



- 《TCP/IP详解卷1》

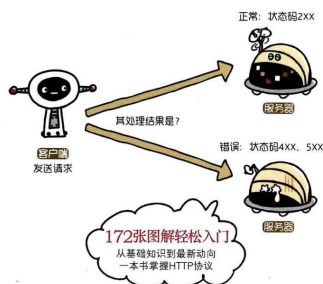


- 《图解TCP/IP》



# 图解HTTP

OWASP [日] 上野 宣 著  
于均良 译



知识星球

人民邮电出版社  
POSTS & TELECOM PRESS

牛客网特立独行MVP

计算机网络主要的常考知识点如下：

## TCP/IP

1. 分层模型
2. 三次握手、四次挥手原理
3. 拥塞控制
4. 流量控制与滑动窗口
5. TCP与UDP比较
6. DDos攻击
7. close\_wait, time\_wait状态原因

## HTTP

1. http/https 1.0、1.1、2.0
2. get/post 以及幂等性
3. http 协议头相关
4. Cookie Session 的比较
5. 长连接与短连接，websocket的对比

## Socket

1. IO模型
2. select, poll, epoll原理和对比

## 5. 分布式数据库学习

## ☰ 特立独行MVP

要。

- MySQL入门建议可以看《MySQL必知必会》。原理比较少，基本都是语法和使用细节，适合先过一遍。关于原理可以看《数据库原理，编程与性能》。
- Redis入门可以看《Redis设计与实现》这本书。非常详细的讲了Redis的底层数据结构，包括其集群部分也做了很好的解释。除此之外可以再去理解一下分布式的Raft算法，Gossip协议等等。

数据库常考知识点：

### MySQL相关

- 1.索引：B树、B+树底层结构，索引失效条件
- 2.sql语法
- 3.关系型数据库三范式
- 4.存储引擎：InnoDB和MyISAM对比
- 5.数据库的锁：行锁，表锁，页级锁，意向锁，读锁，写锁，悲观锁，乐观锁等等
- 6.数据库隔离级别：脏读、不可重复读、幻读
- 7.事务的ACID理论
- 8.查询优化：explain，慢查询，show profile
- 9.分布式：分库分表，读写分离

### Redis相关

- 1.Redis 特点
- 2.Redis 底层数据结构：跳表，字典
- 3.数据淘汰策略
- 4.持久化方式：AOF，RDB
- 5.哨兵模式
- 6.集群同步方式

## 6.其他方面的学习

由于篇幅原因在此我不会把每一个方向和知识点都详细罗列，如果各位有兴趣可以看我在春招实习写的学习路线：

[后台开发/基础架构方向 学习路线](#)

## 写在最后

## 特立独行MVP

实力：建议大家加深自己某一个方面的深度，拓宽技术面的广度，只有深度+广度才能脱颖而出；

面试：我会在后面的专栏当中专门给大家分享我的面试技巧，希望大家可以参考；

运气：作为求职者我们没有足够的信息，无法预知某个公司岗位的难度，不要嫉妒别人的轻松，机会只给有准备的人，努力的人运气一般都不会差；

我目前还在慢慢的职业摸索阶段，也还是一个菜鸟而已，如果有任何错误的地方也欢迎大家指正修改。我的方法和经验不一定适合所有人，如果这篇文章真的可以帮助到未来求职的你也就达到我的目的和初衷。愿大家一起进步。

## 结语

在本节当中我主要介绍了我在求职C++后台开发方向的学习路线，在这中间有很多我一路走来从其他人身上学习到的技巧，也有一些我个人总结的经验，我相信对于不同的人来说没有绝对完美的提升策略但是一定有最合适自己的方案。希望我总结的路线可以给大家带来启发，帮助大家在求职的路上找到属于自己的成长路线。在下一节当中我将重点介绍我在算法方面的提升的策略。

感谢阅读，如果文中有任何错误欢迎你给我留言指出，也欢迎分享给更多的朋友一起阅读。

[举报](#)

收藏 2



赞 2

### 相关专栏



《收割BAT：C++校招学习路线总结》

19篇文章 | 95订阅

已订阅

0条评论

默认排序 ▾




没有回复

请留下你的观点吧~


发布





### 《收割BAT：C++校招学习路线总结》

《收割BAT：C++校招学习路线总结》，专栏共计17节。专栏分为五大主要内容，包括后台开发学习...  
19篇文章 | 95阅读

 牛客博客，记录你的成长

[关于博客](#) | [意见反馈](#) | [免责声明](#) | [牛客网首页](#)