

2015-12-20 23:22:10,953 [myid:] - INFO [main-SendThread(localhost:2181):
ClientCnxn\$SendThread@1032] - Opening socket connection to server
localhost/0:0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SASL
(unknown error)

JLine support is enabled

2015-12-20 Offer之上: 本书教你如何去学习, 而不是仅仅教你如何去拿Offer!
ClientCnxn\$SendThread@1032] - Session established to
localhost/0:0:0:0:0:0:0:1:2181. Session

2015-12-20 23:22:11,672 [myid:] - INFO [main-SendThread(localhost:2181):
ClientCnxn\$SendThread@1299] - Session establishment complete on server
localhost/0:0:0:0:0:0:0:1:2181, sessionId = 151c2115b200, negotiated

timeout

WATCHER

Watched event: state change, new state: null

[zk: localhost/0:0:0:0:0:0:0:1:2181]

程序员炼成记

Chengxuyuan
Lianchengji

从小白到工程师

周明耀◎著

```
[root@localhostconf]# cat zoo1.cfg
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmpfs is a better
# example sake.
dataDir=/var/lib/zookeeper/zoo1
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
```



北京大学出版社
PEKING UNIVERSITY PRESS



Offer之上：本书教你如何去学习，而不是仅仅教你如何拿Offer！
愿本书成为你入行的起点，而不是终点！



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 提 要

本书主要介绍了作为一名软件工程师应具备的能力。内容主要包括Java的基础知识和JVM、死锁、CPP技术、Java8技术、G1 GC的实践、Java的优化方向、代码规范深度解读等深度知识, Spring Boot、Spring Cloud、Spring里的设计模式, 关系型数据库的代表PostgreSQL和NoSQL数据库的代表Cassandra, 分布式技术、消息中间件、大数据框架、搜索引擎、事务、Linux隔离技术、Go语言入门等高端技术。最后一章, 包含了作者多年的经验总结, 列举了可能会遇到的问题, 并提出了解决思路。

本书适合所有软件工程师, 尤其适合工作两年以下的人, 力求覆盖应用软件开发岗位的校招面试范围。

图书在版编目(CIP)数据

程序员炼成记: 从小白到工程师 / 周明耀著. —北京: 北京大学出版社, 2018.10
ISBN 978-7-301-29893-0

I. ①程… II. ①周… III. ①程序设计 IV. ①TP311.1

中国版本图书馆CIP数据核字(2018)第210124号

书 名 程序员炼成记: 从小白到工程师

CHENGXUYUAN LIANCHENG JI; CONG XIAOBAI DAO GONGCHENGSHI

著作责任者 周明耀 著

责任编辑 尹 毅

标准书号 ISBN 978-7-301-29893-0

出版发行 北京大学出版社

地 址 北京市海淀区成府路205号 100871

网 址 <http://www.pup.cn> 新浪微博: @北京大学出版社

电子信箱 pup7@pup.cn

电 话 邮购部 010-62752015 发行部 010-62750672 编辑部 010-62570390

印 刷 者 北京市科星印刷有限责任公司

经 销 者 新华书店

787毫米×1092毫米 16开本 35印张 939千字

2018年10月第1版 2018年10月第1次印刷

印 数 1-4000册

定 价 99.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究

举报电话: 010-62752024 电子信箱: fd@pup.pku.edu.cn

图书如有印装质量问题, 请与出版部联系, 电话: 010-62756370

序

Preface

可能是因为之前报社记者的工作经历,本书作者周工(周明耀)在我周围的同事当中,是那种特别善于总结并将其用文章表达出来的人。也正是这种优势,让他能将工作学习中的经验惠及更大的人群。

这本书是写给从事软件行业的时间在-2(准备选择软件行业的求职者)到+2(软件行业的职场新人)年之间的人的,全书的前半部分内容结合了作者自己在长期的校招,以及后续的新人导师的工作中积累的心得感受,从职业认知和基础知识储备两个方面总结了软件行业对那些准备选择这个行业的求职者的基本要求,并就行业工作特点做了适度展开,这也有利于求职者充分认识软件行业的基本工作状态,结合自身的特点做出更合适的职业选择。

国内软件行业近些年来发展得如火如荼,千百万个年轻人从大学甚至中学时起就憧憬着有一天可以像扎克伯格那样从工程师到CEO,成为人生赢家。那么现实是怎么样的呢?我工作了20多年,负责过各种各样的软件团队,深感哪里都有江湖,在软件的江湖里,有许多的事情要初入职场的你去经历和体悟,但其实你并不孤单。在“预见未来的自己”一章里,你也许会读到一样的困顿和纠结,这时你才发现,大家都是在自己成长的道路上艰难跋涉而已。更为难得的是,书中除了有作为研发经理的作者的経験,还引入了同为我的同事的资深系统架构师季怡的成长感悟,从软件管理通道和软件技术通道两个角度更加全面地为职场新人提供建议和指导。

这本书的题目叫《程序员炼成记:从小白到工程师》,作者的上一本书叫《技术领导力:程序员如何才能带团队》,据悉后续还有更加宏大的计划。让我们一起期待吧!

——海康威视研究院大数据技术部总监 闫春

前言

Preface

7岁那年,当我读完《上下五千年》上、中、下三册全书时,我对自己说:“我想当个作家。”2016年4月我出版了《大话Java性能优化》,4个月后第2次印刷;2017年5月,我出版了《深入理解JVM&G1 GC》,半年后第2次印刷;2018年,我的第三本书《技术领导力:程序员如何才能带团队》面世,这本书在京东创下了连续几个月100%好评的纪录;如今,我的第四本书《程序员炼成记:从小白到工程师》也即将面世。我对自己的每一本书都怀着忐忑、惊喜的心情,就像第一次面对我的女儿。

我一直在考虑如何写这么一本书,它能够反映出我自己是如何学习软件技术、如何去积累的,因为情怀,所以我在做这样的事。以技术管理知识为例,2018年年初,我总结自己和朋友的经验并结合软件工程理论出版了《技术领导力:程序员如何才能带团队》一书,现在我想写一本书分享给工作两年以下的人,使他们知道如何去学习,而不是仅仅教他们如何拿到Offer。当然,也希望这本书能够覆盖应用软件开发岗位的校招面试范围,帮助学生拿到Offer。这也是本书的由来。

本书分为以下8个章节。

第1章 了解这份职业:主要介绍自己对于这份职业的理解,以及这份职业的责任、内容、发展、困难等。

第2章 学习准备:主要介绍软件安装、数据结构和难题解释。

第3章 Java基础知识:主要介绍由各种Java基础知识引申出来的内容。

第4章 Java深度知识:主要介绍JVM、死锁、CPP技术、Java 8技术、G1 GC的实践、Java的优化方向及代码规范深度解读等Java深度知识。

第5章 Spring相关知识:主要介绍热门的Spring Boot、Spring Cloud和Spring中的设计模式。

第6章 数据库知识:主要介绍关系型数据库PostgreSQL和NoSQL数据库Cassandra。



第 7 章 高端技术汇总：主要介绍分布式技术、消息中间件、大数据框架、搜索引擎、事务、Linux 隔离技术及 Go 语言入门等。

第 8 章 预见未来的自己：主要介绍未来可能会遇到的挫折、不愿意面对的领导，以及在工作不同时期会遇到的不同状况，针对这些情况的分析和总结。

本书提供源代码下载，扫描下方二维码可进行下载。



扫描二维码下载源代码

本书的目的并不仅仅是帮助读者拿到 Offer，还包括帮助读者找到学习方法、了解未来的发展方向。最后送大家一句话：不要担心眼前安逸的生活即将结束，而是应该担心真正的生活从未开始。

感谢我的家人，我的妻子美丽、细心、博学，并且还很温柔，我很爱她；我的女儿性格很像我，希望她能够踏踏实实做人，保持创新精神，平平安安、健健康康地生活下去；感谢我的岳父母、我的父母，他们帮我照顾孩子，我才有时间编写此书；感谢浙江省特级教师、杭州高级中学（杭一中）化学老师郑克良，郑老师的一句“永远不要放弃”，推动着我多年不断前进；感谢数学老师张老师在公开场合对我的褒奖，张老师的赞赏对性格内向、内心细腻的我起着重要的作用。感谢我生命中出现的恩师、良友，有你们的存在，让我得以绽放。

本书的稿费将会捐献给杭州慈善总会的小额命名款项——“郑克良先生的学生”，用于感谢郑克良老师在我人生中播种下的“永远不要放弃”的种子。

我相信这本书不是终点，它是“麦克叔叔”一系列技术书籍中的一员，欢迎加入我的朋友圈，可以通过微信号 michael_tec 或者微信公众号“麦克叔叔每晚 10 点说”和我联系。

Write a great book that millions could read is my passion!

目 录

CONTENTS

第1章	了解这份职业	001
1.1	写在前面	002
1.2	入行前	002
1.2.1	对于 ACM 国际大学生程序设计竞赛的理解	002
1.2.2	参加校招	003
1.3	入行后	004
1.3.1	深度思考	004
1.3.2	工作时间	004
1.3.3	公司的选择	005
1.3.4	为什么软件基础设施技术人员话语权不高	005
1.3.5	为什么去做高难度的技术	005
1.3.6	技术人员的上升通道	006
1.3.7	跟进最新技术的重要性	006
1.4	自勉	007
第2章	学习准备	008
2.1	软件安装	009
2.1.1	JDK 安装	009
2.1.2	Eclipse 安装与卸载	013
2.1.3	Eclipse 快捷键介绍	017
2.1.4	虚拟机安装	020
2.2	数据结构	027



2.2.1	算法简介	027
2.2.2	数据类型简介	029
2.2.3	面向对象程序设计	029
2.2.4	算法效能分析	030
2.2.5	线性表	032
2.2.6	链表	033
2.2.7	堆栈	067
2.2.8	算术表达式的求值法	075
2.2.9	队列	078
2.3	难题解释	091
2.3.1	两个数字相加	091
2.3.2	寻找两个数组的中间数	093
2.3.3	查找字符串中最长非重复的子字符串	097
2.3.4	合并两个链表	098
2.3.5	汉诺塔问题	099
2.3.6	迷宫问题	105
2.3.7	八皇后问题	110

第3章 **Java 基础知识** **114**

3.1	switch 关键字	115
3.1.1	Java 6 中的使用方式	115
3.1.2	Java 7 中的使用方式	116
3.1.3	新特性的优缺点	118
3.2	设计模式之单例模式	119
3.2.1	引言	119
3.2.2	详细介绍	119
3.3	设计模式之代理模式	125
3.3.1	引言	125
3.3.2	延迟加载	126
3.4	设计模式之适配器模式	132
3.4.1	引言	132
3.4.2	详细介绍	132
3.4.3	适配器模式在开源项目中的应用	137

3.4.4 适配器模式的使用	145
3.5 字符串操作优化	146
3.5.1 字符串对象	146
3.5.2 SubString 使用技巧	147
3.5.3 切分字符串	148
3.5.4 合并字符串	150
3.6 数据定义和运算逻辑优化	154
3.6.1 使用局部变量	154
3.6.2 位运算代替乘除法	154
3.6.3 替换 switch	155
3.6.4 一维数组代替二维数组	156
3.6.5 提取表达式	158
3.6.6 优化循环	159
3.6.7 布尔运算代替位运算	160
3.6.8 使用 arraycopy()	162
3.7 Java I/O 相关知识	163
3.7.1 Java I/O	163
3.7.2 Java NIO	164
3.7.3 Java AIO	174
3.8 数据复用	178
3.8.1 缓冲区	178
3.8.2 缓存	184
3.8.3 对象复用池	185
3.8.4 计算方式转换	187
3.9 集合类优化	189
3.9.1 集合类之间关系	189
3.9.2 集合接口	190
3.9.3 集合类介绍	192
3.9.4 集合类实践	194
3.10 Java 8 迭代器模型	202
3.10.1 迭代器模式	202
3.10.2 Lambda 表达式	204
3.10.3 Java 8 全新集合遍历方式	204



3.11	Java 9 入门.....	209
3.11.1	模块化编程.....	209
3.11.2	模块化系统目标.....	211
3.11.3	模块化的 JDK.....	212
3.11.4	模块资源介绍.....	212
3.11.5	HelloWorld 案例.....	213
3.12	常见面试题.....	214

第4章

Java 深度知识.....222

4.1	JVM 内存区域.....	223
4.1.1	程序计数器.....	224
4.1.2	虚拟机栈.....	224
4.1.3	本地方法栈.....	228
4.1.4	Java 堆.....	229
4.1.5	方法区.....	234
4.2	JVM 为什么需要 GC.....	235
4.2.1	JVM 发展历史简介.....	235
4.2.2	GC 发展历史简介.....	236
4.2.3	G1 GC 基本思想.....	237
4.2.4	G1 GC 垃圾回收机制.....	237
4.2.5	G1 的区间设计灵感.....	238
4.3	如何使用 SA 工具.....	239
4.4	死锁及处理方式.....	246
4.4.1	死锁描述.....	246
4.4.2	死锁情况诊断.....	251
4.4.3	死锁解决方案.....	254
4.5	JavaCPP 技术.....	256
4.5.1	JavaCPP 示例.....	257
4.5.2	JavaCPP-presets 简介.....	259
4.5.3	JavaCPP-presets 示例.....	261
4.5.4	JavaCPP 性能测试.....	270
4.6	Java 8 解决的若干问题.....	271

4.6.1	HashMap	271
4.6.2	行为参数化	273
4.6.3	读取文件	276
4.6.4	Stream	277
4.7	JDK 8 与 G1 GC 实践	291
4.7.1	基础解释	291
4.7.2	G1 GC 参数讲解	292
4.8	Java 的优化方向	303
4.8.1	Java EE	303
4.8.2	函数式语言	305
4.8.3	VM 启动时间优化	307
4.8.4	JIT 编译器	308
4.9	代码规范深度解读	308
4.9.1	下划线或美元符号	309
4.9.2	拼音与英文混合	309
4.9.3	类命名	309
4.9.4	方法名、参数名和变量名	310
4.9.5	常量命名	312
4.9.6	抽象类的命名	312
4.9.7	避免常量魔法值的使用	312
4.9.8	变量值范围	313
4.9.9	大括号的使用规定	313
4.9.10	单行字符数限制	314
4.9.11	静态变量及方法调用	315
4.9.12	可变参数编程	316
4.9.13	单元测试应该自动执行	318
4.9.14	单元测试应该是独立的	318
4.9.15	BCDE 原则	318
4.9.16	数据类型精度考量	319
4.9.17	使用 Char	321

第5章**Spring 相关知识 323**

5.1	Spring Boot	324
5.1.1	初始 Spring Boot	324



5.1.2	Spring Boot 示例.....	337
5.1.3	Spring Boot 创建 Restful API 示例.....	341
5.1.4	Spring Boot 使用 JavaMailSender 发送邮件.....	344
5.1.5	Spring Boot 1.5.x 新特性.....	347
5.2	Spring Cloud	349
5.2.1	Spring Cloud 简介.....	349
5.2.2	Spring Cloud Eureka.....	350
5.2.3	Spring Cloud Consul.....	353
5.2.4	分布式配置中心.....	354
5.3	Spring 中的设计模式.....	358
5.3.1	解释器设计模式.....	358
5.3.2	构造器设计模式.....	358
5.3.3	工厂方法设计模式.....	362
5.3.4	抽象工厂设计模式.....	364
5.3.5	代理设计模式.....	366
5.3.6	策略设计模式.....	368
5.3.7	模板设计模式.....	370

第6章

数据库知识 374

6.1	关系型数据库和 NoSQL 数据库.....	375
6.1.1	关系型数据库.....	375
6.1.2	NoSQL 数据库.....	378
6.2	PostgreSQL 相关知识.....	380
6.2.1	基本操作.....	380
6.2.2	系统视图表.....	381
6.2.3	索引.....	384
6.2.4	查询计划.....	388
6.3	Cassandra 相关知识.....	393
6.3.1	基本介绍.....	393
6.3.2	数据模型.....	393
6.3.3	关键特性.....	394
6.3.4	访问服务端.....	397
6.3.5	无中心化实现因素.....	403

6.3.6 性能测试工具	408
--------------------	-----

第7章

高端技术汇总	411
--------------	-----

7.1 分布式系统	412
7.1.1 店长负责制	412
7.1.2 订单处理方式	414
7.1.3 员工角色拆分	415
7.1.4 多个任务接收	416
7.1.5 订单处理过程上屏	416
7.1.6 异常数据干扰	417
7.1.7 座位设计模式	418
7.2 选举算法的机制	419
7.2.1 最简单的选举算法	419
7.2.2 拜占庭问题	420
7.2.3 Paxos 算法	422
7.2.4 ZAB 协议	424
7.2.5 ZAB 与 Paxos 的联系及区别	432
7.3 HDFS 中 NameNode 单点失败的改进案例	432
7.4 从星巴克下单过程看事务处理方式	435
7.5 软件工程师需要了解的搜索引擎知识	437
7.6 从 eBay 购物车丢失看处理网络 I/O	443
7.7 Apache Kafka 工作原理及案例介绍	446
7.7.1 消息队列	446
7.7.2 Apache Kafka 专用术语和交互流程	447
7.7.3 利用 Apache Kafka 系统架构的设计思路	448
7.8 Apache ZooKeeper 服务启动源码解释	450
7.8.1 ZooKeeper 服务启动	450
7.8.2 流程及源代码解释	456
7.9 ZooKeeper Watcher 机制	464
7.9.1 集群状态监控示例	464
7.9.2 回调函数	468



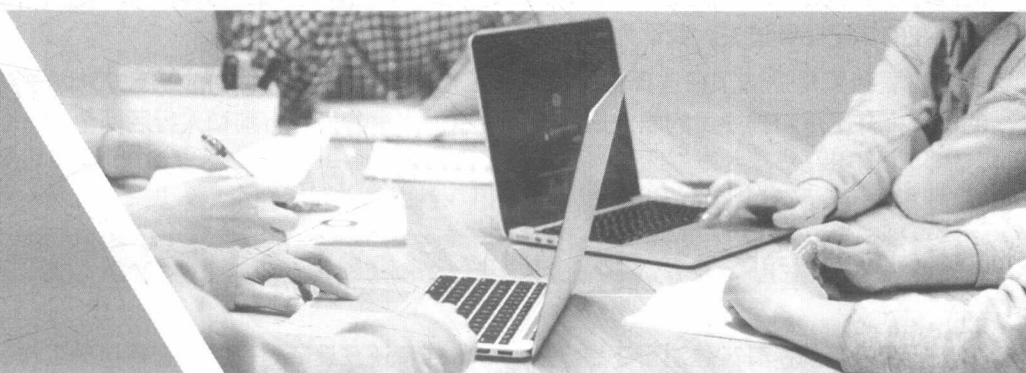
7.9.3 实现原理及源代码解释	469
7.9.4 ZooKeeper Watcher 特性	480
7.10 HBase 数据导入方式	481
7.10.1 启动 HBase	481
7.10.2 向 HBase 导入数据	482
7.11 HBase 优化策略	495
7.11.1 HBase 数据表概述	495
7.11.2 HBase 调用 API 示例	496
7.11.3 HBase 数据表优化	500
7.12 CGroup 技术	506
7.12.1 CGroup 介绍	506
7.12.2 CGroup 部署及应用实例	508
7.13 Go 语言	517
7.13.1 示例程序	517
7.13.2 命名规范	518
7.13.3 变量概述	519
7.13.4 gofmt 工具	520
7.13.5 垃圾回收	520

第8章

预见未来的自己	522
8.1 遇到 Bug 时的态度	523
8.2 平静看待挫折	526
8.3 当遇到了不懂技术的领导	530
8.4 写给未来跳槽的你	533
8.5 技术选型的注意事项	536
8.6 架构师之路	540
8.6.1 软件行业，苦乐自知	540
8.6.2 如何做好一个架构师	541
8.6.3 走出成为架构师的关键道路	543
8.6.4 回顾与总结	545

第1章

了解这份职业



正式了解技术前，先讲一下我和我的朋友们的经历，以及我对程序员这份工作的理解。看完后，你再决定是否要成为一名程序员，或者是否坚持下去。

这也是本章需要解决的问题，它主要包括以下内容。

- » 程序员这份工作涉及哪些方面的知识、前景如何
- » 关于思考、工作时间、公司选择等的个人看法
- » 关于软件人员地位、高难度技术的个人看法
- » 关于技术人员的上升通道、跟进最新技术的重要性的个人看法



1.1 写在前面

某一天我和两位发小聚餐，了解到我们 3 个人都选择了程序员作为自己的职业，刚开始时都是做最基础的工作，往后的十几年开始出现了发展偏差。十四五年过去了，我们都在职业的重要选择点上面临选择，虽然选择不同，幸运的是我们都走在了适合自己的道路上。

我一直认为，如果勉强自己做一件不喜欢的事情，那么很难有动力在今后的工作中继续前进。选择成为一名程序员，大概是在十六七岁时，那时候父母给我买了第一台电脑（一台 586 电脑），在我的内心种下了一颗种子。IT 行业是个知识更新非常快的行业，而我又是一个喜欢尝试新事物的人，心理承受能力也较强，所以程序员是一个不错的选择。此外，IT 行业的机会很多，也适合我。以现在的年纪回顾过去，我认为，自己没有选错。

这里我只是挑选了在选择程序员这份工作时所需面对问题中的一部分大家关注的话题来讲我的理解，并不是对程序员的职业规划进行分享。本章分为入行前和入行后两个部分分别进行介绍。

1.2 入行前

所谓入行前，其实是对于还是学生的你，或者还在迷茫是否需要坚持这份工作，是否应该选择程序员作为工作职业及如何获得这份工作的你。作为一名在这个行业从事多年的人，我觉得真的要想清楚程序员这份工作是否适合你，因为这份职业牵扯的个人精力实在太多，导致自己不可能有太多的个人爱好，也确实没有太多陪伴家人的时间，加班几乎是这个行业的标签。大家都知道，很多程序员会自嘲地称自己为“码农”。目前来看，程序员是最容易逆袭的职业，从收入角度也是最容易逆袭的工作。

入行前，以学生为例，除了正常的上课、实习之外，还可以通过一些竞赛类的准备工作提升自己的技术能力；此外，学生一般也需要通过校招进入技术含量较高的科技公司。针对这两点，我分别谈谈自己的看法。

1.2.1 对于 ACM 国际大学生程序设计竞赛的理解

以个人的经验，参加 ACM 国际大学生程序设计竞赛（ACM-ICPC）的学生遇到的问题有点类似于高中各学科竞赛，需要为了提高竞争力而学习比较深层次的知识，需要做大量的题来积累经验，但致命的是除了顶尖高中竞赛选手和顶尖大学的 ACM-ICPC 爱好者（ACMer）外，其他人学习的知识都不太系统，尤其是数学上的。

大多数人对数学的学习仅仅局限于数据结构、离散数学，可能因为 ACM-ICPC 要有数论题而学习了一些数论和组合数学的基础，具体就不会去深入学习。结果，ACMer 的数学功底并没有因为 ACM-ICPC 的做题训练而提高，对算法的理解甚至可能仅局限于套用算法模板，这对于未来想从事算法研究的人来说，无论是在公司为了某个任务做优化，还是在研究机构发表论文，都会

存在致命的基础短板。

ACM-ICPC 除了算法以外，带来的间接好处是可以提高编程能力，但很明显提升编程能力的主要途径就是编程，至于写什么样的代码、什么语言的代码，区别不是很大。但不可否认，准备 ACM-ICPC 确实是一个可以锻炼写代码能力的机会。

公平地说，ACM-ICPC 依然是我接触的在大学里所有比赛中最公平、最锻炼能力的，它的准备时间长、比赛时间短、评价体系又比较客观。如果想参加比赛，ACM-ICPC 应该是计算机专业的首选，但并不是说 ACM-ICPC 有很大价值，因为 ACM-ICPC 本身就是一种游戏，它可以让人痴迷，可以形成一种容不得别人说 ACM-ICPC 不好的圈子。不管怎么说，可以肯定的是，ACM-ICPC 的经历让学生更容易被公司录用，因为“平均水平”高一些，参加 ACM-ICPC 的人可能更有能力，尤其是编码这种无法体现在简历上的实际能力。

1.2.2 参加校招

连续几年参与了公司组织的校园招聘工作，每次去都会见到大量的学生，我喜欢和他们交流，观察他们的一言一行。为了进一步了解他们的性格特点和综合能力，我每次都会自己准备面试题目，这些题目包括编程基本概念、算法编程、操作系统、数据库编程、开源代码阅读、垃圾回收机制、系统架构描述、实习期经历回顾、人生过程中遇到过的挫折、对于工作氛围的想法、未来的职业发展方向设定等。

这些问题其实大多数都是开放式问题，一些没有固定的答案，另一些甚至是完全开放式的，需要学生提出问题。我的这组题目中，可能只有编程基本概念这一条有固定的标准答案；算法编程、数据库编程都有多种回答方案，只不过每一种回答的运行效率不同，这些都属于半开放式的技术问答；操作系统、开源代码阅读、垃圾回收机制、系统架构描述，这些问题则属于是自己出题的题目，为什么这么说？因为我会根据你所了解的知识一点点地深究下去，一点点往下问，所以这是完全开放式的技术问答；实习期经历回顾、人生过程中遇到过的挫折、对于工作氛围的想法、未来的职业发展方向设定这 4 个问题属于非技术领域的完全开放问答，我之所以提出这些问题，是希望能够更加接近学生的真实生活、内心想法，了解学生的过往经历、内心想法，以及生活环境和周边朋友，这样可以决定是否录用，以及如何更好地发挥他们的能力。

我讲一下自己的应聘面试经历。很多年前，我去参加一家德国企业的面试，总经理是位中国人，50 来岁的老博士，他让我谈谈对公司情况的了解。我已经做足了功课，把他们网站上的英文背出来了，我一边背，他一边睁大了眼睛，扶了扶眼镜，还纠正了我对于创始人德文名字的发音错误，然后和我说：“你有什么要求，现在就可以提”。

除了实际的技术基础能力外，我觉得一名学生还需要具备大的格局，不要局限于眼前利益，不要只考虑自己。推荐大家担任大公司的“校园大使”，这个工作一定要尽自己最大可能做好，积极配合 HR，其实在这个过程中你也在被观察。我有次就遇到了 3 位截然不同的校园大使，第一位校园大使非常认真，两天时间忙上忙下，布置会场、参与宣讲会、电话联系学生、引导学生面试流程等，还找了几位好朋友过来帮忙，忙到自己没有时间参加面试，虽然他存在不太善于沟通的弱点，但是我在结束面试后，单独给他留出了面试时间，并且和 HR 一起邀请他加入；第二位校园大使面试当天就坐下来面试了，技术一般，他自己介绍是校园大使，我正在犹豫时 HR 和我聊起了他，说：