

1. 基础知识：

1. 算法和数据结构

1. 数组、链表、二叉树、队列、栈的各种操作（性能，场景）
2. 二分查找和各种变种的二分查找
3. 各类排序算法以及复杂度分析（快排、归并、堆）
4. 各类算法题（手写）
5. 理解并可以分析时间和空间复杂度。
6. 动态规划（笔试回回有。。）、贪心。
7. 红黑树、AVL 树、Hash 树、Tire 树、B 树、B+树。
8. 图算法（比较少，也就两个最短路径算法理解吧）

2. 计算机网络

1. OSI7 层模型（TCP4 层）
 - 每层的协议
 - url 到页面的过程
2. HTTP
 - http/https 1.0、1.1、2.0
 - get/post 以及幂等性
 - http 协议头相关
 - 网络攻击（CSRF、XSS）
3. TCP/IP
 - 三次握手、四次挥手
 - 拥塞控制（过程、阈值）
 - 流量控制与滑动窗口
 - TCP 与 UDP 比较
 - 子网划分（一般只有笔试有）
 - DDos 攻击
4. (B)IO/NIO/AIO
 - 三者原理，各个语言是怎么实现的
 - Netty
 - Linux 内核 select poll epoll

3. 数据库（最多的还是 mysql，Nosql 有 redis）

1. 索引（包括分类及优化方式，失效条件，底层结构）
2. sql 语法（join, union, 子查询, having, group by）
3. 引擎对比（InnoDB, MyISAM）
4. 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，以及加锁的 select sql 方式）
5. 隔离级别，依次解决的问题（脏读、不可重复读、幻读）
6. 事务的 ACID
7. B 树、B+树
8. 优化（explain, 慢查询, show profile）
9. 数据库的范式。
10. 分库分表，主从复制，读写分离。
11. Nosql 相关（redis 和 memcached 区别之类的，如果你熟悉 redis, redis 还有一堆要问的）

4. 操作系统：

1. 进程通信 IPC（几种方式），与线程区别
2. OS 的几种策略（页面置换，进程调度等，每个里面有几种算法）
3. 互斥与死锁相关的
4. linux 常用命令（问的时候都会给具体某一个场景）
5. Linux 内核相关（select、poll、epoll）

5. 编程语言（这里只说 Java）：

1. 把我之后的面经过一遍，Java 感觉覆盖的就差不多了，不过下面还是分个类。
2. Java 基础（面向对象、四个特性、重载重写、static 和 final 等等很多东西）
3. 集合（HashMap、ConcurrentHashMap、各种 List，最好结合源码看）
4. 并发和多线程（线程池、SYNC 和 Lock 锁机制、线程通信、volatile、ThreadLocal、CyclicBarrier、Atom 包、CountDownLatch、AQS、CAS 原理等等）
5. JVM（内存模型、GC 垃圾回收，包括分代，GC 算法，收集器、类加载和双亲委派、JVM 调优，内存泄漏和内存溢出）
6. IO/NIO 相关
7. 反射和代理、异常、Java8 相关、序列化
8. 设计模式（常用的，jdk 中有的）
9. Web 相关（servlet、cookie/session、Spring<AOP、IOC、MVC、事务、动态代理>、Mybatis、Tomcat、Hibernate 等）
10. 看 jdk 源码

2. 项目经历

- 这个每个人的项目不同，覆盖的技术也不一样，所以不能统一去说。
- 这里的技巧呢，在下面也会详细说明。
- 无非是找到自己项目中的亮点，简历上叙述的简练并且吸引眼球，同时自己要很熟悉这个点（毕竟可以提前准备）
- 最好自己多练，就像有个剧本或者稿子一样，保证面试中可以很熟练通俗地讲出，并且让人听着很舒服。

3. 实习经历

- 这个很抱歉，因为我是找实习的经历，所以也没有实习经历的讲述经验。
- 但我想如果你有实习经历，那面试过程的重点也会在实习做了什么上面，所以大家最好对实习所做的工作做一个总结，并且同样抓出亮点，搞懂内部原理，提前锻炼讲述的过程。

4. 其他扩展技能（这个方方面面太多了，全部掌握基本上不可能，只是作为大家其他时间扩充技能的参考）

- 分布式架构：（了解原理就行，如果真的有实践经验更好）
- CAP 原理和 BASE 理论。
- Nosql 与 KV 存储（redis，hbase，mongodb，memcached 等）
- 服务化理论（包括服务发现、治理等，zookeeper、etcd、springcloud 微服务、）
- 负载均衡（原理、cdn、一致性 hash）

- RPC 框架 (包括整体的一些框架理论, 通信的 netty, 序列化协议 thrift, protobuf 等)
- 消息队列 (原理、kafka, activeMQ, rocketMQ)
- 分布式存储系统 (GFS、HDFS、fastDFS)、存储模型 (skipList、LSM 等)
- 分布式事务、分布式锁等
- 脚本语言：(只是作为横向扩充, 一般问到 linux 也会问问 shell 脚本)
- - python
 - php
 - shell
 - go
 - ...
- 大数据与数据分析：
 - hadoop 生态圈(hive、hbase、hdfs、zookeeper、storm、kafka)
 - spark 体系
 - 语言：python、R、scala
 - 搜索引擎与技术
- 机器学习算法：
 - 模型和算法很多。不细说了, 如果很熟练就去投算法, 国内很多公司都算法岗都很稀缺, 其他岗可以大概了解下理论。
- 其他工具的理论和使用：
 - 这个更多了, 问的多的比如 git、docker、maven/gradle、Jenkins 等等, 自己需要的话选择性地去学。

5. 实力要怎么积累

积累实力最好的方式就是平时在项目中或是学习中, 多学多问, 多思考多钻研。这里就说短期学习的一些方法和路径：

总结下来, 一方面是通过看书、看视频、看面经来不断扩展自己的知识面, 一方面是通过不断的面试积累经验和知识盲区, 在每次总结的过程中积累实力。

• 先说说看书学习这一点

这里应该是有个推荐书籍的环节, 这几个月确实看了不少书, 但是并不是所有都有很高的性价比, 在这边大概列一下, 前后顺序也一定程度上代表了我认为的重要度先后, '/'做分割的是一类的书籍, 一般来说看一个就够了：

书单：

算法与数据结构：

数据结构 (严蔚敏) /大话数据结构 //如果觉得教材无聊就可以看大话系列, 印象中里面还有很多诗

剑指 Offer/程序员面试金典/编程珠玑/编程之美/牛客网+leetcode

程序员笔试面试最优解 (左程云) /不如直接看左神的笔试面试指南视频

Java 的版本 (不是很推荐)：

数据结构与算法经典问题解析 (Java 语言描述)
图解数据结构 (使用 Java)

计算机网络：

计算机网络 (谢希仁)
TCP/IP 详解
HTTP 权威指南
图解 TCP/IP
图解 HTTP

数据库：//数据库主要是多用，书上主要看索引和性能的部分
高性能 MySQL/深入浅出 MySQL

操作系统：

OS 原理：操作系统 (课本，黑色的那个)
Linux：
Linux 私房菜 //鸟哥写的，很全，包括 bash 部分
跟阿铭学 Linux //主要偏重于命令和操作，比较浅显

java：

Java 疯狂讲义/Java 编程思想/Java 核心技术 卷 1
深入理解 Java 虚拟机
并发编程的艺术/多线程编程核心技术
Effective Java
Java 程序员面试笔试宝典 //何昊的那本，个人感觉是突击知识点的神器
Java 程序性能优化
实战 Java 高并发程序设计

Java Web：

Spring 实战/轻量级 JavaEE 企业应用 (红皮，讲 SSH 的) //主要看最后一部分
Spring 的就可以
深入 JavaWeb 技术内幕 (阿里 许令波) //这个讲的还是比较深的
SpringBoot 实战/深入实践 SpringBoot

设计模式：

大话设计模式 //通俗易懂
各类博客的总结

分布式与大数据：

分布式服务框架原理与实践
大型网站技术架构
Hadoop 实战 (hadoop 体系包括得很全)
//还有一本我暂时想不起来名字了

其他：

Git :

Git 权威指南

Git 官方讲解视频 (牛客网有带字幕的)

Redis :

Redis 实战

还买了 docker、springCloud 等等一些工具书, 因为太小众就不列举了

除了上面说的书和视频, 最有用的还是大家分享的各种面经。

面经是个很不错的东西 (嗯, 想看的可以直接翻到最后一节)。记住不要一扫而过 (除了那种岗位不太匹配可以快速抓重点看), 岗位匹配的你可以顺着面经逐条去看, 模拟一次面试过程 (虽然是单向的, 但是你心里应该是知道每个问题你能答道什么层次), 这种不断地模拟可以让你知道盲点或者说弱点在哪, 对于一个你心里没底或者想不太起来的问题一定要当时记下来或者当时就弄懂。

我当时是每天晚上在床上看几篇面经, 然后把存疑的问题 copy 到记事本中, 第二天找个固定的时间短查询解决和总结。长期下来会养成一个很好的习惯, 你的知识点会不断地扩充。