参考面经：

<https://leetcode.cn/circle/discuss/VU7QvY/>

https://leetcode.cn/circle/discuss/t7l1s2/

https://leetcode.cn/circle/discuss/MGZxUb/

# 简历

简历是公司对你的第一印象，决定了你能不能进入后续的笔试面试环节。一般包含个人信息、项目经历、实习经历、专业技能、个人荣誉等方面，其中项目/实习和技能是最为核心的。切记，需要展现你的优势和亮点，隐藏你的劣势和缺陷。

个人信息主要包含姓名、年龄、求职意向、联系方式、学历、照片等。在学历部分，如果成绩亮眼，也可以进行标注。对于项目/实习经历，可以参考Star法则来进行描述，写清楚项目背景，个人负责的内容，采用的方法和技术，取得的成果等等。对于专业技能，首先要真正了解自己的掌握情况，对自己的掌握程度要描述准确，一般用词是了解->熟悉->精通。如果只是学习了基本理论知识，没有很深入，建议用了解；如果理论知识学习比较深入，并且有相关的实践经历，建议用熟悉；而如果在某个技能上可以说是大牛甚至专家级别，才建议用精通。因此，一般大部分时候建议用了解和熟悉来描述专业技能，不要用精通。有个比较著名的笑话可以百度一下，“我精通C++”。除了内容上之外，简历在形式上也应当是简洁美观，逻辑清晰，重点突出的，切忌杂糅冗长排版丑陋，这里推荐一个排版工具超级简历。

  最后，简历不能一招鲜，需要看碟下菜。不能用一份简历投所有岗位，要针对每个具体岗位的JD做相应的修改，哪怕只是改个名字呢：

在每次面试完以后，也可以根据面试反馈做相应的调整。这里也要注意，按照JD进行修改不是照抄JD，**凡是写在简历上的东西，你一定要做到符合描述，切忌夸大其词。**

# **简历**

**简历是块敲门砖，简历写的不够好，面试邀请的都拿不到，也就没有后续的事情了。另外在内推的过程中，看过的简历也有4000多份了， 基本用不了20秒，就能大体判断出一份简历是否合格。那些优秀的简历，都有一些共同点。**

**下面主要分享3个技巧，供大家参考**

**1.根据Jd写简历**

**很多人写简历，都沉浸在个人世界中，与外界“隔绝”。 很多人也都知道简历要个岗位与匹配越好，但是当简历写完，总觉得差一点意思。所以我的建议是根据jd写简历。**

**具体的做法：首先到招聘网站，搜索目标岗位，选择5-6个岗位（选择大公司），然后复制岗位描述 和 任职要求（你会发现这些岗位的描述和要求基本都是大同小异）。这时你会发现，你得到一份这个岗位 在这个行业的专业术语。最后，结合个人情况，讲这些术语，适当性的融合进你的简历！到这里一份与岗位要求匹配的简历就成形了。**

**这样做有以下几个好处：**

**1.提升简历专业性。 每个行业都有自己的“黑话”，看到JVM、MPP就知道是后端研发、看到DAU、用户留存率、KOL就知道是做运营的等等。但是你自己写的话术可能就相对直白、朴素，引用jd中的词汇就会显得更高级一点。每一份招聘岗位的要求都是面试官亲自写的，作为一个岗位面试官，应该都是在这个行业比较资深的，他的话术相对一些初入职场的小白来说，还是会更加高级和专业。**

**2.提高简历通过率。 一个hr会负责多个方向的招聘，太对某些方向的招聘可能熟悉的程度不高，尤其是技术方向，大多数hr都不是技术出身，当你的简历多次出现jd上面的话术的时候，你给他的感觉就三个字：很匹配！ 我面试字节的时候，就是把腾讯 和 字节的岗位描述融合了一下了，当天就约上了面试。**

**3.可复用在个人总结/优势。 简历都会有一个 个人总结 / 个人优势 / 专业技能的模块，你可以引用部分岗位描述到这个模块。然后也可以作为内推的 推荐理由。**

**总之 ，根据Jd写简历，目的是让你的简历和jd更加匹配，越匹配获得面试邀请的机会就越大。hr只关心你的过往经历、专业技能是否能够胜任当前工作。**

**网上很多教你怎么写兴趣爱好，我的建议是，不用写，直接删除这一块，因为hr不关心你喜欢什么（除非你的兴趣爱好与工作相关，能体现你的优势）**

**2 .学会用star法则写简历！**

**这个就不用过多说明，太重要了，建议都用star法则写简历。**

**STAR法则,即为Situation Task Action Result的缩写，具体含义是:**

* **Situation: 事情是在什么情况下发生**
* **Task: 任务**
* **Action: 针对这样的情况分析，你采用了什么行动方式**
* **Result: 结果怎样，在这样的情况下你学习到了什么**

**简而言之，STAR法则，就是一种讲述自己故事的方式，或者说，是一个清晰、条理的作文模板。不管是什么，合理熟练运用此法则，可以轻松的对面试官描述事物的逻辑方式，表现出自己分析阐述问题的清晰性、条理性和逻辑性**

**所以写简历最重要的一点就是 用star法则，说明出项目的背景(s)、你担任的任务(t)，采取什么样的行动(a)，取得了什么样的结果(s)。用star法则写出来简历，一般是差不了的。**

**3.多用数据表达！**

**结合star法则，多用数据量化工作，不要说一大推没用的东西。一定要多用数据突出结果， 数字比文字描述更能吸引hr、面试官的注意，对于一些重要的数字结果可以适当的加粗显示。你用数字去衡量工作成果，就像公司工资衡量你的价值。**

# **面试逻辑安排**

**面试技巧有很多，这里我分享我一直在用的一个方法，或者叫做策略更为合适。**

**我把它叫做 ：三阶段的面试法**

**三阶段的面试法，顾名思义分为三个阶段去面试，每一个阶段要做的事情、花费的时间、要达到的目的都不一样。**

**1.一阶段：**

**面试时间：1-2周**

**面试对象：小厂（建议C轮以上）**

**面试目标： ①找找面试感觉 ②了解现在的行情 ③查漏补缺**

**详细说明：第一阶段 是要找一些小厂面试，主要是找一找面试的感觉，工作一段时间后，刚开始面试难免会有一些不适应，出现紧张的情况。根据自己的情况选择，每天面一家到两家，每次面试都要录音，面完及时总结，晚上统一复盘，针对回答的不好的知识点要重点复习，查漏补缺。**

**这个阶段的公司应该选择比较有前景公司，但不应该是你的理想公司，所以到offer阶段，你可以大胆的要薪资，万一翻了好几倍，你也是可以考虑下的。**

**2.二阶段**

**面试时间：1-2周**

**面试对象： 中厂**

**面试目标： ①加深巩固知识点 ②保offer**

**详细说明：第二阶段 选择一些比较知名的公司，知名公司的面试官的水平都比较专业，在某些点上，会问的比较深。每次面试都是一次复习和学习的过程，问到不会的、理解的不透彻的地方，下来就好好复习加深。**

**有了第一阶段的面试练手，第二阶段的面试节奏可以加快一点，每天选择2-3家公司面试（第一阶段的公司可以适当拒面了）。要争取拿下offer，作为我们的保底offer，也能为第三阶段谈薪提供有用的竞争offer。拿下offer后的公司可以去，但是没必要（除非工资ok）。**

**3.三阶段**

**面试时间：2周+**

**面试对象：大厂 or 你最想去的公司**

**面试目标：冲刺理想offer + 谈薪**

**详细说明：第三阶段，要选择一些顶级大厂 或者你最想取的公司，这些公司你可能没有把握拿下，所以要放在最后来面试。到了这个时间点，你会发现，每一次面试问的问题基本都固化了，同一个问题、同一个项目、同一个知识点，你已经别人说了很多遍，早就烂熟于胸了。再来面这些大厂你就会更加自信。同时，你有第二个阶段的offer，在谈薪也不会吃亏。**

**当然，即使经过三阶段的面试，你可能还是遇到不会的问题，这个没关系，讲好自己会的，不会的大方承认，表明自己的态度，不会的以后工作可以学。（工作时：这个没用过，做不了，你找其他人吧）**

# 笔试面试内容

  笔试大多为计算机基础+算法题，在准备面试的过程中就能准备到。而面试分为技术面和Leader/HR面，一般都包含自我介绍+反问，技术面还会增加计算机基础(OS,NET,数据库，数据结构，算法)+语言基础+项目+手撕算法+场景设计环节，Leader/HR面会比较综合。这里以C++后台开发为例。

  自我介绍主要包含个人信息介绍，项目经历和获奖经历，有些比较亮眼的点可以在介绍的时候突出表现，有可能引起面试官的兴趣。自我介绍一定要提前练熟，做到条理清晰、流畅自然。

## 语言基础

主要考察基础关键字、类与对象、对象内存模型、内存管理、程序编译过程、STL、几个较新的标准等。

https://blog.csdn.net/qq\_45254369/article/details/126023482

https://leetcode.cn/circle/discuss/owG1Fb/

* *C++ 多态及其实现*
* *C++11 的新特性*
* *右值引用*
* *vector 与 list、map 与 unordered\_map 比较*

## 计算机基础

操作系统，操作系统主要考察进程与线程管理、内存管理、IO多路复用、Linux具体使用

*https://leetcode.cn/circle/discuss/zIxrWn/*

* *进程与线程的区别*
* *堆与栈的区别*
* *内核态和用户态的划分与切换*
* *虚拟地址怎么映射到物理地址*
* *简述 IO 多路复用*
* *简述进程通信的各种方法*

计算机网络、计算机网络主要考察应用层、传输层、网络层，

* *简述从输入网址到浏览器显示的过程 https://leetcode.cn/circle/discuss/UrcaDQ/*
* *TCP 为什么是三次握手四次挥手 https://leetcode.cn/circle/discuss/aqTOW4/*
* *TCP 如何保障数据包有效*
* *HTTPS 和 HTTP 的区别*

数据结构，数据结构问的比较多的有哈希表、B+树等

数据库

* 简述关系型数据库与非关系形数据库的区别与联系
* 简述数据库的事务

设计模式

限于篇幅，具体的学习和准备方案这里不做过多赘述，会在后文中贴出很多比较好用的参考链接。这两块虽然被戏称为八股文，但是并不是说背就完事儿，需要有真正的理解，掌握到一定的深度，只有这样，面试的时候你才能讲出自己的东西，从而得到面试官的认可。

  如果面试官对你的项目很感兴趣，一般会问项目问的比较多，对项目中的每个技术点都可能问的很细很深，有可能会针对项目问一些开放扩展性的问题，比如这个项目或者技术点，你认为还有没有能够优化的地方，应该怎么优化？因此对于自己写到简历上的项目，一定要提前做好整理归纳和总结反思，做到烂熟于心，每个技术点的细节都要清楚，而且要能够发散的去思考，对一些技术点进行展望和扩展。

   手撕算法就是现场写算法题。不要一听完题目就写，一定要和面试官多沟通多交流，理解清楚题意。如果遇到问题，就和面试官讲自己的思路，一般面试官都会很乐意指出存在的一些小问题。如果实在不会写，也不要干瞪眼，可以直接和面试官说对这块不太熟悉，可不可以换一个题目。当然，工欲善其事必先利其器，提前进行大量刷题提高算法能力才是根本大法。

   反问环节要体现出你对公司和岗位有热情，感兴趣。

* 可以问岗位的日常工作，需要具备的技能和特质，应聘的整体流程，新人的培养机制等等，
* 不建议问我这次面试能不能过，加班严不严重，给多少钱等问题。

  另外再提一嘴，面试整体表现主要源自于多年的积累，但临时抱佛脚也很有必要。因为很多公司的面试问题都是相对比较固定的，因此新鲜面筋的价值就体现出来了，在面试之前最好是整理一下过往面经中提到的问题，然后进行查缺补漏和针对性的突击，在接下来的面试中遇到相同问题的概率也是比较高的，这样至少心里会有一个准备：

在每轮面试结束之后，要及时的做好整理和总结，尤其是自己回答得不好的问题，一定要花时间去研究和解决，这样当下次面试再遇到的时候就可以迎刃而解：