

06 | 特别放送：北美大厂如何招聘全栈工程师？

2019-09-23 四火

全栈工程师修炼指南

[进入课程 >](#)



讲述：四火

时长 21:56 大小 15.07M



你好，我是四火。

在第一章技术内容的末尾，我们来换换脑子，聊一些略轻松的话题。我曾在开篇词中讲过，全栈工程师的市场需求量很大，今天我就来介绍一下北美大厂，特别是那些大名鼎鼎的互联网巨头们，都是怎样招聘全栈工程师的。

这些大公司在全世界不同的国家内往往都会建立基地聚敛人才，当然包括 Top 2 的互联网超级大国——中国（你可能还不知道，[互联网十大企业中，中国占了四大，美国占了六大](#)）。我想，了解一下他们的做法，对于程序员的你来说，既能拓宽眼界，也能更好地清楚自己在市场上的定位，从而更好地成长，这应当是很有价值的。

招人理念

首先，招聘这个事儿，其重要性毋庸置疑，这几乎是所有的互联网公司都认可的一点。对某些互联网公司来说，例如 Google，则是 “最重要”的事情，连“之一”这两个字都省了。

招到一个优秀的工程师，你的团队和产品，都将获得巨大的收益；而招到一个不合格的工程师，不但会拉低团队的水准，还要花费其他同事大量的时间精力来帮助其成长。因此，招聘可以说是壮大一家公司最快的方法，但同时也是毁掉一家公司最快的方法。于是，面试，对于很多大型互联网公司的工程师来说，就是日常工作的一个重要组成部分。

通常来说，这些公司在招聘的时候，最关心这样两件事情。

一个是非技术能力，很多公司把其中重要的内容归纳成了领导力准则（Leadership Principles），比如 亚马逊的这十几条；另一个则是技术能力，主要包括编程能力、问题解决（Problem Solving）能力和架构、设计能力。其中，不同级别的工程师，对于系统设计等内容的要求不同，但是对于编程能力的要求基本是一样的。例如，初级工程师可能在技术能力上 90% 考察的是编程能力和问题解决能力，而高级工程师这部分往往会掉到 60%，剩下的 30% 考察架构和设计能力。

我们可以把北美和国内不同的工程师岗位考察来做一个比较，它们都立足于基础，但还是有所区别的。国内大厂的面试我认为更具备实战性，即知识性更强，技术面较广，不同用人单位对于不同技术的考察也更具体；但北美（也包括在国内的北美外企）大厂的面试则更偏向于具体技术的问题解决能力、编码能力，以及架构设计能力等等。至于全栈工程师岗位，其实并没有特别显著的特殊性，候选者考察的理念基本是一样的，只是对于问题领域，以及技术栈等方面的考察，会更有针对性。

招聘流程

1. 简历阶段

招聘的流程有时很短，一周内就可以完成所有的事情，有时也会很长，数月、甚至超过一年之久。

大多数程序员还是习惯于使用招聘网站来投递简历，但是也有许多程序员们，在找工作的时间窗口内，是以点对点的方式来找下家的，即答复 E-mail、LinkedIn 上来主动来联系的 Recruiter（招聘人员）或 Manager（经理），主动投递目标公司，或者找朋友推荐，而不是将自己的简历无差别地挂到招聘网站上，这样可以有效避免过多的电话和消息骚扰，还可以针对特定的公司优化简历，做进一步准备。

但无论你是用招聘网站，还是朋友推荐，有一天，有一位自称公司的招聘人员用着客气的语气发来邮件，她介绍了自己的来历，并和你约了时间电话聊天，这就意味着“简历关”已过，进入了互相了解的阶段。

在把公司、团队、项目、薪酬等等这些事情都介绍完之后，如果互相还有进一步的意愿，通常就要进入电话面试环节了。对于全栈工程师这一岗位来说，**有时招聘人员会问一点非常简单的 Web 相关的基础知识，这其实并非为了考察候选人的能力，而是为了过滤掉那些明显不靠谱的候选人**，给后面的面试团队节约时间成本。之后，Recruiter 会根据工作的时间长短给出候选人的最低应聘级别，比如说，已经工作五年的工程师，最低也得是中级工程师的职位了，此时如果候选人达不到要求，那就不要了。

还有公司招聘是统招（General Hire），即招聘完毕之后统一双向选择去哪个小组工作，比如 Facebook，但绝大多数公司还是会采用项目组自己招聘的形式。对于前者，招聘人员会着重介绍公司的文化和公司的发展方向；但对于后者，则可以具体很多，即可以和候选人交流目标团队、项目和技术栈等。

2. 电话（视频）面试

电话面试（Phone Screen）通常有一轮或两轮，每轮 45 分钟到一个小时，有时候也会通过视频面试，当然，对于特别优秀或者他人强力推荐的候选人，甚至可以免试。

电话面试一般由一线工程师来担任面试官，如果是两轮面试，那么第二轮有时也由一线经理来担任面试官。电话面试至少有一轮必须要考编码问题，这个问题一般不会很难，**通常是一个较为简单到中等偏下难度的算法题，但是需要在电话沟通的基础上，通过多人在线文档工具将代码写下来**，面试官可以看到文本上的编码全过程。对于全栈工程师来说，面试官还可能会花几分钟的时间，问几个全栈技术范畴内的问题，但是总的流程是一样的。

电话面试出结果很快，一般 Hiring Manager（招聘经理）和参与面试的工程师一商量，就可以确定要不要进行下一步了。电话面试的通过率对于每个公司都不太一样，但是根据我的观察，一般这个通过率是在 30% ~ 50% 左右。如果电话面试挂掉的话，通常就可以认为候选人距离公司和团队的要求还有较大差距。

3. 现场面试

如果电话面试结果不错，候选人往往会有一个和团队核心成员或是经理见面的机会，比如 OCI（Oracle 的云基础设施部门，它是独立运作的，文化和流程都和传统的 Oracle 有较大

不同），这主要是在进一步的面试前，给候选者和团队继续互相了解和深入的机会。毕竟，面试是双向的。

接下来，招聘人员会和候选人约时间进入 On-site（现场面试）环节，这个环节需要到项目组所在的城市去，且一般需要持续一天的时间。**比较常见的是 5 轮面试的方式，每轮一个小时，中午会和招聘经理或是团队的工程师吃饭**，这顿饭有可能作为一轮面试算入考察过程，也可能不算。像 Facebook 和 Google 都不算，候选人可以轻松地在食堂饱餐一顿；Amazon、OCI 则往往会请候选人选择一份外卖，送来以后和招聘经理边吃边聊，考察非技术能力。

现场面试的招聘经理一般由一线工程师团队的经理担任。从这里你也可以看到，招聘一个人的成本，包括人力、时间、场地、差旅等等开销，是非常高昂的。

在除去吃饭以外的几轮面试中，编码能力是重点，通常最少有三轮是必须要涉及高强度的编码问题，我们把它简称为“主要问题”。这个编码通常在白板上进行。通常**面试官会努力将候选人带入到团队合作解决问题的氛围中，然后给出一个较为模糊的问题，再来一起沟通交流解决问题，最后代码必须落到白板上面**，这个解题过程要占据每轮面试的绝大部分时间。

对于初级工程师以上的职位，还有至少一轮的问题需要重点考察系统设计能力。每轮面试还有十多分钟的项目和问题挖掘时间，这部分的执行相对较为自由，往往是基于候选人的工作经历往深挖项目和技术，越具体越好。有经验的面试官会抓住一两个点往下深挖，挖到非常细节的部分，从而判断候选人是在夸夸其谈，还是一个真正做事的人。对于全栈工程师来说，项目和技术问题可能大量涉及全栈领域，例如以 Web 网站或应用为背景的题目。

这里我要再次强调一下白板代码的重要性，这里的白板代码不仅仅包括最后落笔写代码，还包括写代码前大量的确认、分析、讨论、架构、设计等等过程，这些占据技术面试的“主要问题”，所有的内容都是在白板上进行的，从中可以全方位地考察候选人技术和非技术能力。

就如同那个“没有 jQuery 不会写 JavaScript”这样略带戏谑、但又透露着些许无奈的说法一样，现在很多程序员朋友都忽视了基础能力的修炼，没有了 IDE 就不会写代码了。白板肯定不像 IDE，有方法提示，写错了还可以随意修改，所以你现在明白为什么用白板了吗？那是因为白板要求你的思路非常清晰，代码组织能力也要很高。由于空间的限制，在白板上修改代码总是件不那么容易的事儿。

另外，还要补充一点，上面谈到的每轮面试中，最重要的那道面试题，公司的要求是，**题目一开始必须要足够模糊，从而激发和考察候选人合作解决问题的能力，在沟通中逐步细化问题的时候，必须要达到中等偏上到困难的难度，以保证足够的区分度。**

这样的问题会考察到候选人的多项素质，特别是编程能力和问题解决能力。但也不要觉得，面试官是在刻意为难，这样的题目设计起来其实并不容易，他们要尽量避免使用互联网上很容易找得到的“常见题”，这个过程往往比解题本身要难得多。**题目不得涉及技术本身的奇技淫巧，不得对候选人使用的编程语言有限制，更要避免“知识性问题”。**

综合来看，我觉得是不是主要问知识性问题，是北美软件工程师（包括全栈工程师）的面试，和国内的面试比起来，在技术层面最本质区别。

那什么是“知识性问题”呢？知识性问题，就是那些直接的、较容易通过搜索和文档获取到的知识性内容。比如，Spring 怎样配置 Bean，Tomcat 怎么修改最大连接数等等，这些问题，手册一翻就是分分钟的事情。

但这绝不是说这些知识不实用，面试官通常不问这类问题最大的原因是，**知识性问题的随机性太强**，如果候选人恰巧刚刚遇到过，或者记性不错，就很容易回答上来了。而这些**并不能反映候选者分析问题、思考、判断和权衡的能力**。但话说回来，知识性问题也是考察候选人基础技能的一种方式，有的面试官也会问，但肯定不是每轮面试中占据时间最多的那个“主要问题”。

4. 讨论会

在面试之后，所有的面试官都要写下对于候选人的反馈，这包括候选人的优势和劣势。在很快而来的 Debrief Meeting（讨论会）中，所有的面试官会根据自己的判断评价候选人，分为四档，分别是“强烈建议录用”“建议录用”“建议不录用”和“强烈建议不录用”。

当然，他们可以说服别人，也可以被说服而改变评价。每一个人，都会负责一项技术考察项（比如数据结构和算法）和非技术考察项（比如是否能拥抱变化，逐步改进），这些考察项在不同的公司会有不同，通常来说五轮里面针对算法和数据结构的考察至少有三轮。另外，有一些非技术的“Red Flag”（即所谓的“红线”），是绝对不能触碰的，比如说对于现有的职位或年限说谎。

在一组面试的面试官中，有两个角色值得一提，**一个是招聘经理**，上面已经提到了；**还有一个是技术负责人**（例如在 Amazon 叫做 Bar Raiser，OCI 叫做 Bartender），负责保证招

聘质量，他们拥有一票否决权，也就是说，哪怕其他所有人都同意，但这两人只要有一个不同意就不能通过。

对于其他情况，直接投票且多数获胜。少数情况下，针对某个候选人，讨论会可能会有截然相反的意见，这时候面试官们就会摆出事实依据进行争辩了。这个讨论会的录用结果，还要包括职位级别。另外，有少数公司在这方面采用的方式略有不同，比如微软，候选人的最终录用决策是由一个特殊的“大人物”（叫做 As Appropriate）决定的，这和国内某些互联网公司很像。

除了我刚刚说的讨论会，某些公司为了从更高层面上把控招聘需求和质量，还可能会有额外的环节，比如 Google 和 Facebook 在讨论会以后，Hiring Committee（招聘委员会）拥有下一步的决策权，他们可以对讨论会通过的考生做进一步筛选。总的来说，不同的公司差异较大，但即便是同一家公司，现场面试的总体通过率也非常不确定，并且这个波动较大，高的时候可能达到接近一半，低的时候可能只有十分之一。

如果这一步也过了，就可以根据面试反馈的结果适当调整候选者的职位级别，再往后就是商量并给出 offer，确定入职时间，进行背景调查等等众所周知的步骤了。

进一步思考

众所周知，在一家公司中，软件工程师的未来发展方向，通常包括技术路线和管理路线两个。但是据我了解，大多数程序员还是更钟情于技术路线的，可对于程序员的编程技术，你一定听说过“青春饭”的说法。

我也听到过不少程序员谈论自己的职业现状，表示随着工作经验的增加，公司似乎更爱招刚毕业不久的年轻人，因为他们更有精力，薪水也更低。于是，大家看着互联网大潮是越来越汹涌，可东西却是越来越学不动了，工作也越来越难找了……

这是怎么回事？

首先，我想说的是，找工作总是希望自己是往上走的，薪水越来越高，职位也越来越高，责任也越来越大，高级职位在市场上的需求却越来越小。因此，**从这个角度看，工作本来就是越来越难找的**。因此，如果是这个原因，这未必是一件坏事。

其次，随着年纪的增加，你觉得你的核心竞争力是什么？如果只是重复地写 Spring 的配置，只是照猫画虎写写 CRUD 的样板代码，那么，比你年轻、能吃苦、能加班的，且薪水

更低的程序员当然可以分分钟打败你。因为，你这不是有了工作多年的经验，而是工作一年的经验重复了多遍。从这个角度看，无论这个招聘的职位是不是全栈，学的技术是不是在 Web 方面，道理都是一样的。

你的竞争力，在具备扎实基础的前提下，应该是经验、思路、眼界等等这些东西。技术是相通的，技术本质是不容易改变的，在新技术到来的时候，你有基础，无论是深度还是广度的积累，应该让你学得更快，而不是学得更慢。

我想，今天介绍的关于工程师招聘流程的内容，也恰好反映了上面这两点：

扎实的基础不可或缺，这是前提。老实说，我经常遇到工作了好多年的一线程序员，连一个简单的二叉树广度优先遍历算法都写不出来。

经验、思路、眼界，都有高度，才是更高级别技术职位的要求，这也反映在上述面试系统设计、问题解决等等方面。

这样的招聘方式当然有它的弊端，例如招到的人可能未必对所用技术都熟悉，未必能立马干活，但通过这样的招聘方式，确确实实可以过滤掉那些“一年工作经验重复多遍”的候选人。

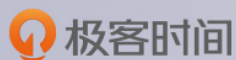
总结思考

作为特别放送，今天的思考题，我想换个形式。如文中介绍的那样，设计一道合理的面试题其实并不容易，需要综合考虑多个因素。下面我列出了几道面试题，假设今天你就是面试官，你能说说它们中哪些适合作为全栈工程师岗位面试的“主要问题”，哪些不适合吗？

- A：设计一个算法，把一个小于一百万的正整数，使用罗马数字来表示；
- B：对一个 Web API，设计一个流量控制系统；
- C：写一个 C++ 算法，实现 atoi 算法，即将字符串转换为数字；
- D：设计一个网约车系统；
- E：完成一个 HTML 页面，能够在网页上显示一个表示当前时间的数字时钟。

最后，我想强调一件事，单个应聘经历永远只能代表单次经验，如果有好的结果，那么恭喜你，但请不要意得志满，这其实并不代表你的整体水准；如果结果不好，也请不要灰心丧气，它并不代表你就真的达不到那家公司的要求。毕竟，招聘也好，面试也罢，其中的随机性太强，冷静、淡定分析自己的情况，再采取合理的措施，才是王道。

今天的特别放送就到这里，希望你在阅读后能有所收获。如果你在应聘和面试方面有什么困惑，或者想分享分享你的面试经历，欢迎在留言区一起讨论。



全栈工程师修炼指南

从全栈入门到技能实战

熊燚

Oracle 首席软件工程师



新版升级：点击「👤 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 05 | 权衡的艺术：漫谈Web API的设计

精选留言 (3)

💬 写留言



leslie

2019-09-23

阐述一下个人观点：目前处在不止只是全栈进一步转型中；既然老师定位的是全栈。那么可能我个人会从觉得这个设计还是明显太片面了。

我认为这5道题目中只有BD是肯定合适：A和C只能保留一个，如果我是出题者我会保留A但是难度上强化；CE的问题就比较典型：C过于单纯的考察单一方式了，E其实应当去考察网络相关的。这样才符合全栈的概念。以下是对于老师给出的5道题目的分析...

展开 ∨



看不懂的你

2019-09-23

有没有什么方式对自我能力进行评判，是看自己能不能回答世面上所有的面试笔试题么？



Geek_0122d3

2019-09-23

四火老师您好，我是明年将要毕业的学生，上周三刚刚参加完亚马逊中国来武汉的现场面试，流程和这篇文章里写的真的很像，编程题占了大比重，不过总共只面了两轮，虽然没什么通过的希望但是趁机会听面试官讲了很多，有谈到亚马逊重视工程师的全栈技能，sde是someone develop everything。

而我个人未来也是想往全栈的方向走，很明显外企这样的氛围更有助于成长为全栈工程...

展开 ∨

