

00 开篇词 为什么大厂面试必考操作系统？

你好，发现求知的乐趣，我是林骐。

我在阿里巴巴做架构的多年时间里，每天都在和复杂的业务场景斗争着，比如如何应对高并发场景？如何解决系统间的数据一致性问题？如何带给用户更快更爽的体验？

在处理一个又一个业务架构、系统架构问题的过程中，我愈发地意识到有一块知识非常重要，也就是我们这门课的主题——操作系统。操作系统可以作为一个完整的知识框架，把复杂如高并发、数据一致性的问题，基础到编程语言、计算框架、业务框架的问题都串联起来。

为什么要学习操作系统？

操作系统 (Operating System) 作为一门计算机专业大学必修课，如今已经成为程序员跳槽、涨薪、过面试的必考内容。像面试中高频的考点，比如 Linux 指令、中断、多线程、并发、性能、内存管理、系统稳定性、文件系统、容器和虚拟化等，知识都来源于操作系统。学了操作系统：

- 你不懂 Java 多线程，也可以回答好 Java 多线程的面试题；
- 你不熟悉 Docker，也可以回答出容器化应该如何做。



任何编程语言学下去都会碰到操作系统知识

任何研发工具学下去也都会碰到操作系统

——林駢

@拉勾教育

操作系统已不仅仅是一门大学的必修课那么简单，更是计算机领域的本源知识，任何编程语言学下去都会碰到操作系统知识，比如 Java 的虚拟机、Go 语言的协程与通道、Node.js 的 I/O 模型等。任何研发工具学下去也都会碰到操作系统，比如：

- MySQL 深入学下去会碰到 InnoDB 文件系统；
- HBase 深入学下去会有 Hadoop 文件系统 (HDFS) ；
- Redis 深入学下去会碰到 Linux 的I/O模型；
- Docker 深入学下去有 Linux 的命名空间等；
- 甚至 Spring 框架，也需要用到线程池和调度算法。

因此，作为面试官，我们需要通过操作系统知识判断求职者的综合能力，你可以将这些语言、开发工具用到什么层次？是能够使用还是理解原理，甚至具备系统改造的能力？

为什么你要学习我这门操作系统课？

互联网领域有一个非常重要的分析方法，那就是一件事情如果可以成功，无非就是合适的人，在合适的时间，做合适的事情。接下来，我将结合课程设计思路和自己的人生经历来分析，你为什么学习我的这门操作系统课。

我们可以先从“事到人”的角度入手。从事情上说，我希望做一门学了就能用的课程，**所以本课程学习目标有两个：一是帮助你顺利通过面试、跳槽涨薪；另一个是帮助你提升应对实际工作场景的能力**，具体包括以下几点。

- **提升学习和理解能力**：比如学习 Redis 可以理解到日志文件系统层面；学习 Java/Python/Node 等语言可以理解到语言最底层。
- **提升应用架构能力**：比如可以将操作系统的微内核架构迁移到自己设计的系统中。
- **提升系统稳定性架构能力**：比如在多线程设计上更出色，可以帮助同事找到设计漏洞。
- **提升运维能力**：做到可以方便地管理集群和分析日志。

下面我再来结合我自身背景和互联网行业特点与你具体聊一聊。

首先，帮你面试涨薪这块我非常有信心，为什么我敢这样说？

我曾在 3 家互联网大厂任职架构师（技术专家），而且是技术委员会成员；另一方面，我做了 10 年的面试官，每年都会到拉勾网筛选简历，保守估计按每年面试 100 个人来算，我至少已经面试过上千人了；还有一方面，我有很多朋友在大厂做技术 Leader，也有很多学生在大厂工作，因此我有一个精准的面试圈子。

我们在面试题、面试技巧方面交流非常多，而且我已经出过几门技术类的在线课程，我的讲课风格也受到了很多学员的认可，他们很喜欢和我交流问题、探讨技术，所以我可以自信地说：我是一个非常懂面试和公司用人标准的老师。

再者，我有丰富的实际应用场景经历。

中国互联网系统最主要的设计约束：并发高、数据量大（毕竟中国互联网是以人口红利起家的）。比较巧的是，海量用户的 C 端场景和大数据商业分析场景，我刚好都负责过。而高并发、大数据中的很多知识，又需要从操作系统中获取，加上我本身操作系统方面的知识也比较扎实，所以在实际场景这块我也有丰富的经验。

接下来再从我的角度来看“现在要不要学操作系统”，我觉得现在的时机刚刚好。

首先，目前是一个在线教育的风口，我结合自身背景以及拉勾网在线招聘求职方向的优势，给你带来一门针对工作场景的就业提升类操作系统课程，符合平台调性。



云原生架构出现之后，越来越强调“谁开发谁运维”
因此业内对操作系统的需求度在提升、要求也在提高

——林駢

@拉勾教育

再者，云原生架构出现之后，越来越强调“谁开发谁运维”，因此业内对操作系统的需求度在提升、要求也在提高，所以我设计这门针对工作场景的就业提升类操作系统课程符合市场需求。

操作系统的需求量也是急剧增长。随着我国用户量和数据量的爆发式增长，也让互联网的运营成本变高。因此，公司很希望自己的员工熟悉操作系统，可以从系统底层去帮助公司节约成本。

此外，随着我国用户量和数据量的爆发式增长，也让互联网的运营成本变高，一个普普通通千万级用户量的网站，就需要每年上亿的运维成本去 Cover。在这种情况下，公司很希望自己的员工熟悉操作系统，可以从系统底层去帮助公司节约成本。所以市场对于操作系统的需求量也是急剧增长的。

课程介绍

接下来我们聊聊课程内容，这门课程对标的是架构师层级的基础能力，看个人的接受程度，学完之后大概会在阿里的 P7 及以上层级。

课程共分为 8 个模块，合计 39 个课时。具体每个模块的介绍，我将在下一课时“课前必读”中详细讲解。在这里，我先跟你分享一下课程的整体设计思路。

这门操作系统课程将帮助你系统地解决面试中可能遇到的计算机原理和操作系统类问题，并以大厂面试题作为切入点，引出很多你在实际工作中会遇到的问题和技術难点。同时，每一个模块聚焦操作系统知识的一个方向，每节课的标题就是这个方向最需要掌握的，也是真实出现过的大厂面试题，同时它也代表着一类知识点。而且，我还将结合实战场景帮助你打牢基础知识，向架构师的方向努力。

说到工作场景，我认为有两个非常重要的问题需要解决。

第 1 个问题是提高大家在实际工作场景中的实战能力。 除了讲解操作系统的知识结构，还会结合以下 6 个场景深入分析：

- 架构师必备的高并发、多线程编程技巧；
- 团队 Leader 如何掌握必备的 Linux 运维技巧；
- 程序语言（Java/Go/Node）的内存分配和回收；
- 数据库底层（MySQL/Hadoop）的文件系统；
- 计算机网络和操作系统 I/O 模型结合有哪些注意事项；
- 如何从操作系统底层理看容器化技术：K8s 和 Docker。

这些都是目前国内大厂研发岗位最为关心的问题，也是大厂面试的热门问题，学会和理解这些问题背后的原理，将对你面试和晋升有很大的帮助。

第 2 个问题是解决面试难题，让你顺利拿到 Offer。 我精选了 80 道左右的大厂面试题：

- 其中 40 道题目会作为课程标题或者课前的提问，在课时中会带着大家分析，以此引出该问题背后的知识点，从点到面一步步把知识点和原理讲清楚。
- 另外 40 道题目将作为课后练习题，帮助你巩固本课时的内容，检测自己的学习成果。此外，这部分问题还会作为精讲习题，在每个模块单独设置一个课时带你梳理解题思路、分析底层原理，结合实战场景帮你分析工作中可能遇到的瓶颈。

学完之后，希望你可以举一反三，这非常有利于提升你的面试成功率。

其实我在做课程设计之前，花了大量的时间进行市场调研，发现目前市场上专门针对互联网方向，帮助你提升系统架构能力、应对日常使用和高并发场景的操作系统课程是缺失的。

这也是我做这门课程的初衷：课程中没有大量的源代码，也没有特别的语言依赖。我会把面试要点和工作场景中的重难点内容都覆盖到，彻彻底底、由内而外地给你讲清楚，帮你应对面试、解决工作难题。

说到这里，我想你可能还有些疑问，比如课程的难度怎么样，适不适合新手学习？或者是课程深度能否满足进阶大厂架构师？

- **第一个问题，适不适合新手学习？**

我的答案是：适合。这门课程对大家的基础知识没做任何假设，比如操作系统唯一依赖的知识——计算机组成原理，我也把它们做成了前置课程，利用模块一的 4 个课时把这块内容讲清楚。

另一方面，我会从设计者角度出发来给你讲问题。比如讲多线程算法的时候，我会先讲遇到了什么困难，再诱导大家思考如何解决这些困难，然后和你一起分析出几个解题方向，最后循序渐进地给出一个合理的方案。所以这门课程非常适合新手学习。

- **第二个问题，课程内容的深度能否满足进阶大厂架构师？**

我的答案依然是：满足，这门课程的深度足够培养一名架构师。课程内容会涉及多线程编程中 AQS 的原理；数据同步中的乐观锁；Java 新生代用了什么算法、老生代用了什么算法；还有比如高并发时有哪些提升性能的方法？这些都是架构师需要掌握的，也是课程的重点。

所以，操作系统这门课程既适合新手入门，也适合有经验的开发人员进阶学习，这并不受经验影响，不同的是你们的学习目标和学习收获。

寄语

最后，我还想和你说点关于职业发展相关的事情。



中国有超过 1000 万程序员，大部分人的年薪小于 30 万
拉开个人薪资和团队整体水平差异的分水岭
根本原因就是计算机基础知识的掌握程度

——林駢
@拉勾教育

中国有超过 1000 万程序员，大部分人的年薪小于 30 万。我观察到一个这样的现象：一方面求职者们抱怨市场竞争激烈，大家争抢一两个岗位；另一方面很多优秀团队的高薪岗位招人难，闲置多个空位，求职者很多但是符合岗位要求的却很少。到底是什么原因造成企业招人难，求职者求职难的情况呢？

其实，**拉开个人薪资和团队整体水平差异的分水岭，根本原因就是计算机基础知识的掌握程度**。基础好的程序员，学习速度快，愿意花时间去积累知识，提高自身能力，因此涨薪快、跳槽更容易；而基础不好的，学习相对较慢，知识输入少，因此涨薪慢、跳槽难。个人能力的高低决定了收入的水平。

这个事情很现实，也很不公平。但是反过来想，为什么基础不好的同学，不花时间精力拿出来去填补自己的知识空缺呢？如果你的操作系统知识还是一盘散沙，那么请你现在就开始行动，跟着我一起重学操作系统，把这块知识捡起来。愿正在看这篇文章的你，能通过自己的努力去到更好的团队，拿更高的薪水，进而得到更广阔的发展空间。

[下一页](#)