

结束语 从学习Redis到向Redis学习

这么快就到课程的尾声了，到了和你说再见的时候了。

在过去的 4 个多月时间里，我们掌握了 Redis 的各种关键技术和核心知识。在课程的最后，我想带你切换一个视角：**如果说我们之前一直在学习 Redis 本身，那么今天，我们来看看能向 Redis 学到什么。**

在聊这个“视角”之前，我想先问你一个问题：你有没有想过，学习技术究竟意味着什么呢？

大多数人都会觉得，就是掌握具体的原理，进行实战，并且学习别人的经验，解决自己在实际工作中的问题。比如说，学习 Redis 时，我们会把它用在缓存、分布式锁、数据集群等业务场景中，这就需要我们掌握关键实践技巧、常见问题和应对方法，这也是我们课程的聚焦点。

但是，我认为，这只是学习技术的第一个层面。当我们对技术的认识和积累达到一定程度后，我们就应该“向技术致敬”。所谓的致敬，就是向技术学习，来解决我们在生活中遇到的问题。这是第二个层面。

这背后的道理其实非常朴素：每一项优秀技术都是一些精华思想的沉淀成果，向技术学习，其实就是向优秀的思想学习。

我一直很崇尚一个理念：一个优秀的计算机系统设计本身就包含了不少人生哲学。所以，接下来，我们就再往前迈一步，从 Redis 设计中总结一些做事方法。

向 Redis 单线程模式学习，专心致志做重要的事

Redis 的最大特点是快，这是 Redis 在设计之初就设立的目标。而能成为某项技术的高手、某个技术方向的大牛，通常是我们给自己设立的目标。Redis 实现“快”这个目标的关键机制就是单线程架构。单线程架构就给我们提供了一个很好的做事方式：**专心致志做一件事，把事情做到极致，是达到目标的核心要素。**

在 Redis 的设计中，主线程专门负责处理请求，而且会以最快的速度完成。对于其他会阻碍这个目标的事情（例如生成快照、删除、AOF 重写等），就想办法用异步的方式，或者

是用后台线程来完成。在给你介绍 6.0 版本时，我还提到，Redis 特意把请求网络包读写和解析也从主线程中剥离出来了，这样主线程就可以更加“专注”地做请求处理了。

我认为，“单线程”思想是非常值得我们品味的。在确定目标以后，我们也可以采用“单线程模式”，把精力集中在核心目标上，竭尽全力做好这件事，同时合理安排自己的时间，主动避开干扰因素。

当我们沉浸在一件事上，并且做到极致时，距离成为大牛，也就不远了。

当然，我们说在一件事上做到极致，并不是说只盯着某一个知识点或某一项技术，而是指在一个技术方向上做到极致。

比如说，Redis 属于键值数据库，我们就可以给自己定个目标：精通主要的键值数据库。因此，我们不仅要扎实地掌握现有技术，还要持续关注最新的技术发展。这就要提到我们可以向 Redis 学习的第二点了：具备可扩展能力。

向 Redis 集群学习可扩展能力

在应用 Redis 时，我们会遇到数据量增长、负载压力增大的情况，但 Redis 都能轻松应对，这就是得益于它的可扩展集群机制：当数据容量增加时，Redis 会增加实例实现扩容；当读压力增加时，Redis 会增加从库，来分担压力。

Redis 的新特性在持续推出，新的存储硬件也在快速地发展，这些最新技术的发展，很可能就会改变 Redis 的关键机制和使用方法。****所以，想要应对复杂的场景变化，我们也要像 Redis 集群一样，具备可扩展能力。***毕竟，技术的迭代速度如此之快，各种需求也越来越复杂。如果只是专注于学习现有的技术知识，或者是基于目前的场景去苦心钻研，很可能会被时代快速地抛弃。

只有紧跟技术发展的步伐，具备解决各种突发问题的能力，才能成为真正的技术大牛。

怎么培养可扩展能力呢？很简单，随时随地记录新鲜的东西。这里的“新鲜”未必是指最新的内容，而是指你不了解的内容。当你的认知范围越来越大，你的可扩展能力自然就会越来越强。

说到这儿，我想跟你分享一个我的小习惯。我有一个小笔记本，会随身携带着，在看文章、参加技术会议，或是和别人聊天时，只要学到了新东西，我就会赶紧记下来，之后再专门找时间去搜索相关的资料，时不时地拿出来回顾一下。这个习惯，让我能够及时地掌握最新的技术，轻松地应对各种变化。

我们做技术的同学，通常习惯于脚踏实地地把事情做好，但是，也千万别忘了，脚踏实地的

同时，也是需要“仰望星空”的。要把学习变成一种习惯，从为了应对问题的被动学习，到为了增强自己的可扩展性而主动学习，这个转变绝对可以让你的技术能力远超过其他人。

当然，Redis 的优秀设计思想还有很多，你还可以自己提炼总结下。我还想再跟你探讨的话题是，我们该怎么把向 Redis 学到的思想真正落地到实践中呢？

其实，道理也很简单：**从做成一件事开始**。在竭尽全力做成事情的过程当中，磨炼自己的专注力，锻炼自己的可扩展能力。

从做成一件事开始

我们常说“不积跬步，无以至千里”，这句话中的“跬步”，我把它解释为做成一件事。我们总是会做很多事，但是，很多时候，能够让我们真正得到提升的是把事做成。

对我来说，创作这门课完全是一次全新的尝试。在写作时，无论是思考内容的结构，确认具体的细节，还是连夜赶稿以保证按时更新，我都感受到了不少压力。但是，现在我回过头来看过去的半年，感到很欣慰，因为这事儿我做成了，而且有很多额外的收获。

其实，做成一件事的目标不分大小。它可以很小，比如学完两节课，也可以很大，比如花 3 个月时间把 Redis 源码读完。

最重要的是，一旦定好目标，我们就要尽全力把这件事做成。我们不可避免地会遇到各种困难，比如临时有其他的工作安排，抽不出时间，或者是遇到了不理解的内容，很难再学进去。但是，这就像爬山，爬到半山腰的时候，往往也是我们最累的时候。

我再跟你分享一下我自己的小故事。

在看 Redis 数据结构的源码时，我觉得非常困难。Redis 的数据类型非常多，每种数据类型还有不同的底层结构实现，而有的数据结构本身就设计得很复杂。

当时我差一点就决定放弃了，但是，我后来憋着一口气，说我一定要把事情做成。冷静下来之后，我进一步细分目标，每周搞定一个结构，先从原理上理解结构的设计，自己在白纸上推演一遍。然后，把每个结构的代码看一遍，同时自己也把关键部分编写一遍。毕竟，我们在看代码的时候，很容易想当然地跳过一些地方，只有自己一行行地去编写时，才会思考得更细致，理解得也更透彻。

攻克了“数据结构”这个难关之后，我发现，后面的就简单多了。甚至在遇到其他困难时，我也不再害怕了。

因为每一次把一件事做成，都会增强我们的自信心，提升我们的能力。随着我们做成的事越

来越多，我们也就越来越接近山顶了，这时，你会真正地体会到“会当凌绝顶，一览众山小”的感觉。

好了，到这里，真的要和你说再见了。“此地一为别，孤蓬万里征”，这是李白送别友人时说的，比较忧伤。古代的通讯和交通没有那么便利，分别之后，好友只能是自己独自奋斗了。

但咱们不是。虽然课程结束了，但是这些内容会持续存在，你可以时不时地复习一下。如果你遇见了什么问题，也欢迎继续给我留言。最后，我给你准备了一份结课问卷，希望你花 1 分钟时间填写一下，聊一聊你对这门课的看法和反馈，就有机会获得“Redis 快捷口令超大鼠标垫”和价值 99 元的极客时间课程阅码。期待你的畅所欲言。

[上一页](#)