春节特别放送（三）| 我的学习资料和工具

2022-02-02 杨胜辉

《网络排查案例课》 课程介绍 >

讲述：杨胜辉

时长 10:46 大小 9.88M

你好，我是胜辉。今天是正月初二，过年好啊，春节玩得还开心吗？

在之前的课程中，我发现有不少同学啊，都在问我学习相关的话题，比如说：

> 能不能介绍下老师是怎么学习的？
>
> 老师你有没有推荐学习 TCP/IP 的书单？
>
> 我是 Java 工程师，怎样可以在网络排查方面快速入门呢？

所以趁着这个机会呢，我想来跟你聊聊“学习”这个老生常谈的话题。

其实，极客时间上的牛人太多了，在学习上呢，可能都比我更有经验，也更有成就，那我这里就抛砖引玉吧，一点小小的总结和回顾。如果能帮到你一点点，让你在这个新春时节里觉得有一丝收获，那我会跟你一样开心。

做笔记：形成理论体系

你有没有发现，当你刚读完一本书的时候，印象还是挺深的，也觉得在这块领域里学到了不少。但是随着时间的推移，对这些知识的印象逐渐模糊，最好笑的是，有时候自己甚至会怀疑：这个技术挺难懂的，也没什么印象了，难道真的是我以前学过的东西吗？

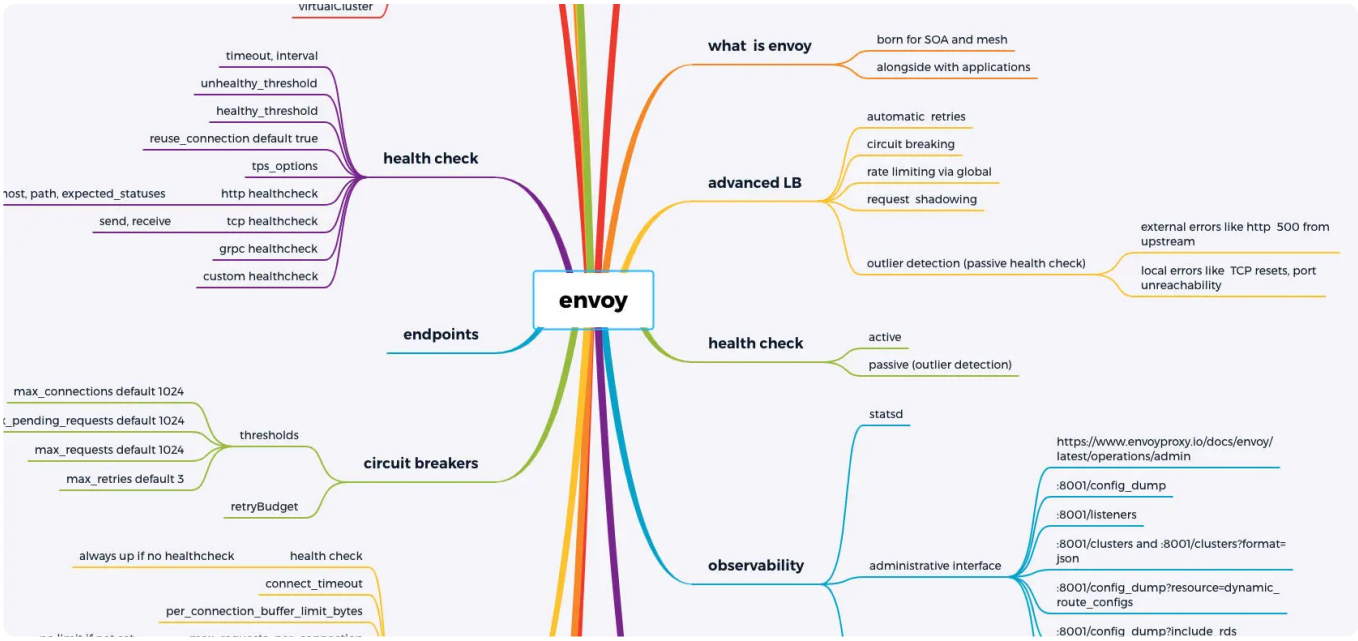
造成这个问题的原因有很多，我觉得其中比较重要的原因，是没有做好笔记。人接收信息的渠道有很多种，比如听说读写，比如实践操作等等。那其中，写应该是比较容易做到的，所以我也挺早就养成了记笔记的习惯。

最早我是在笔记本电脑上直接记录文档，比如用 Windows 的记事本，或者 Word 文档。当然，它们的缺点也显而易见：在这个移动办公的时代，文档不能在手机等移动设备上查看和编辑，真是一大缺憾。

所以这几年，我就以**在线笔记类 App** 为主了，我也用了好几种 App，包括印象笔记、有道云笔记、石墨笔记等等。这些 App 都支持电脑端和手机端，基本上到哪儿都能看笔记或者编辑笔记，十分方便。可以说是让学习深度“侵入”到我的生活中了，但我也乐在其中。看一点笔记，又多懂了一点，以后也可以忘得少一点，这个投入，值得。

就以印象笔记（国外版叫 Evernote）为例，你可以在笔记本、手机、iPad 等终端上安装它，然后多设备同步。出门在外，只要你手机上有印象笔记，就可以继续学习。特别是它的 Markdown 笔记支持代码格式，在阅读一些片段代码的时候很有用。另外，它还有个浏览器插件，当你看到感兴趣的网页时，你就用这个插件把内容给保存为笔记，非常实用。比如像下面这样，我可以把网页中的文稿部分，单独保存为一篇笔记，以后就可以在印象笔记中查看它了：

思维导图也是一类比较有用的学习工具。很多知识未必适合用文档或者表格来做精确的整理，但用思维导图来概括和记录，却更加随性、灵活。我一般在电脑上用 XMind。比如下面就是我学习 Envoy 时候梳理的思维导图：



我觉得，用思维导图的好处之一就是，如果要找某个知识点，很容易从中心点“顺藤摸瓜”找到它。当然好处也不止于此，思维导图的本质，是利用了人的大脑在接收信息方面的特点：**大脑对图形的记忆效率远远超过文字**，通过思维导图，这些知识从文字变成了“树枝图像”，可以更加深刻地印到大脑里。

我在学习一个新领域的知识的时候，经常会新建一个思维导图文件，然后不断往里添加知识的枝干，之后要复习的时候，一眼就能看到这个体系的脉络，确实很有帮助。除了 XMind，国内有个叫 MindMaster 的思维导图产品也不错。

当然，你在工作中同样可以使用思维导图。比如我几年前做的一个比较大的项目，涉及因素众多，一开始我就是用思维导图把各个方面摸清楚的。对着思维导图，我比较容易地抓

到了问题的关键，之后推进起来就比较容易了。推荐你也可以试试。

用正确的方法找学习资料

现在这个时代，知识爆炸乃至泛滥，我们在网上随便一搜，可能正确的和错误的知识各占一半，这种惊人的比例，让我们不得不认真审视学习资料来源的问题。有时候找不到信息还算好的，就怕找到的是错误的信息，浪费时间不说，还让错误的知识占据了大脑，那正确的知识就进不来了。所以，在寻找学习资料方面，我的建议有这么几个：

英文原版

咱们最好加强一下英文，然后阅读原版书籍。这倒不是否认国内翻译者的贡献，而是相对来说，原版的意思会更加贴近原意，然后时效性也更强一点。甚至有些不错的书，未必会有中文译本，特别是一些技术论文，那你要等待中译版几乎是不可能了。你不能总是期望有“野生翻译君”来帮你。

其实，技术类的英文不是太复杂，一旦熟悉了技术名词就会容易许多。再加上我们学习的编程语言也都是英文的，我想你基础再差也不会差到哪里去，所以还是要多一点信心，只要你愿意去尝试，阅读英文技术文章或者书籍都不是难事。从它的收效来说，你的“投资”绝对值得。

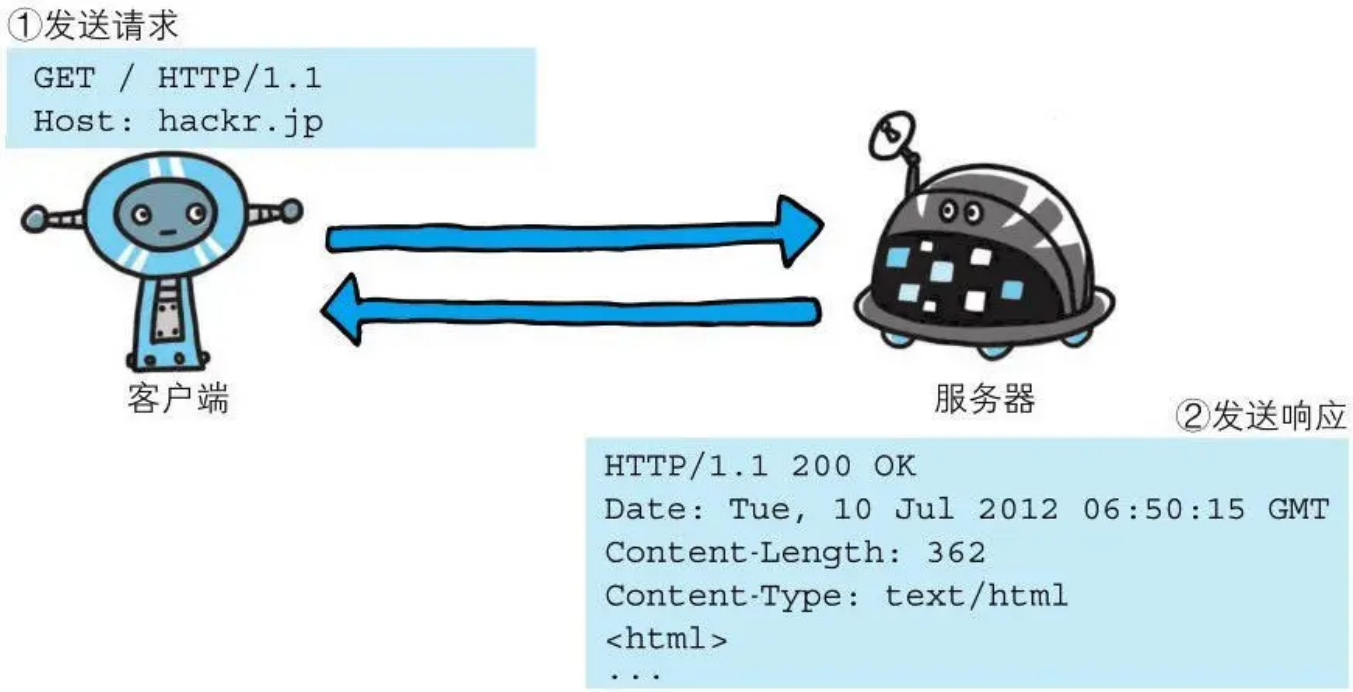
要读经典

现在知识碎片化的趋势越来越明显，不可否认，有时候一小段文字也能给我们不少启迪。不过，要在自己头脑里构建一个扎实并且全面的技术蓝图，还是离不开对经典的研读。

就拿 TCP/IP 来说，我还是推荐经典的 Richard Stevens 等人写就的《TCP/IP 详解》，开发的同学可以看《UNIX 网络编程》。在大多数情况下，读《TCP/IP 详解》第一卷应该就满足日常排查的需求了。在操作系统方面，你也可以读一下《Linux 系统编程》，作者是写过多本技术畅销书的 Robert Love，他本人也还年轻，是一名八零后。

不过，如果你平时开发任务很忙，或者网络基础比较薄弱一点，一下子读大部头确实相对困难，那么你可以选择稍微简单一点的入门书。比如《图解 HTTP》和《图解 TCP/IP》，这两本书的作者都来自日本，用了比较多的示意图来展示知识点。这跟我前面

谈到的思维导图就很类似，图形会给大脑留下更加深刻的印象，所以用这两本来入门也是不错的选择。



要找权威来源

如果我们随便去网上搜索，可能出来很多个人博客，不少内容是抄来抄去，良莠不齐。我建议你直接到比较权威的站点去查看。比如以 HTTP 协议来说，要了解各种 header 和返回码的规范细节，我建议可以去这里看：HTTP | MDN (mozilla.org) 。

要读 RFC

另外一块很大的内容，同时也是很容易把你和其他人拉开差距的地方，就是 RFC 文档。我经常看到，很多人会为了一个技术点争论不休，事实上这个东西就在 RFC 上清清楚楚地写着，只要用心的话，去找到文档仔细读一下就行。也就花一点时间读一下文档，就能收获一个“专业”的对外形象，你应该更有动力了吧？

另外，RFC 文档一般不是太长，内容精练，基本上没什么废话，所以阅读的投入产出比还是挺高的，这也是我推荐你阅读它的另一个很重要的原因。

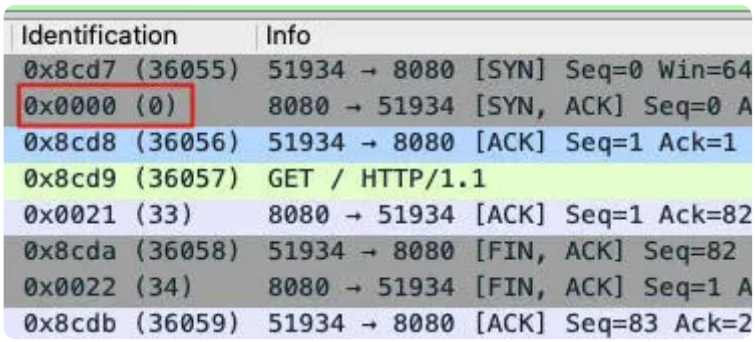
要读源码

最后一块内容，就是内核源码。这是因为，书相对来说更新起来不是那么及时，看源码可以知道最新的行为。另外，还有两个问题你不得不考虑：

有些情况没有在 RFC 里面规定，那这些情况在现实世界里又是如何运作的呢？

RFC 的这些规范，在代码层面是如何具体落实的呢？

比如我们最近遇到的一个小困惑。我们发现，TCP 三次握手里面，第二个报文也就是 SYN+ACK，它的 IP ID 居然是 0。



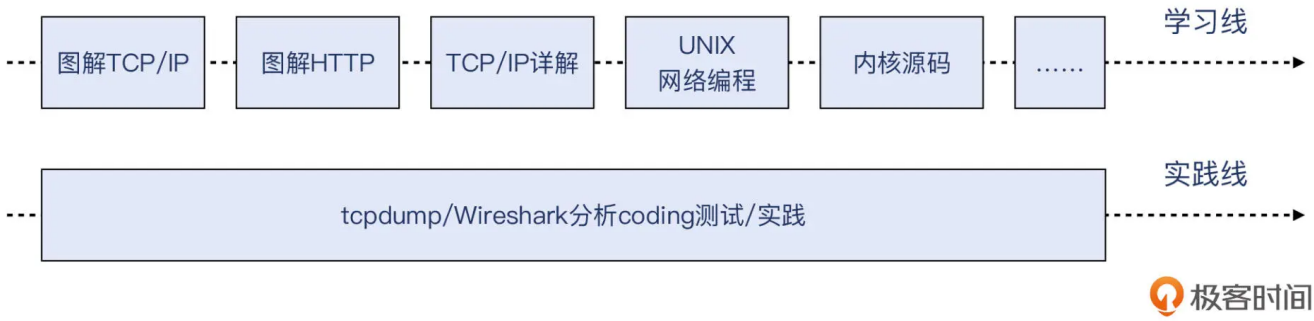
我们就直接查 Linux 内核源代码，确认是对端 Linux 自身的行为了。这样的例子有很多。总之，脱离了代码谈网络，总还是会有盲区。

做中学，学中做

“纸上得来终觉浅”，我们可以在测试环境中，不断把自己学到的知识通过实验进行验证，加深印象。比如，对于 TCP/IP 的各种行为，你就可以一边 curl，一边 tcpdump，观察输出的报文细节，再结合自己学到的理论，看看是否能解释通了？如果还有疑问，那么恭喜你，又可以进步了！人学习的过程，不就是一个不断发现漏洞，不断弥补的过程吗？

如果在实际工作中遇到网络问题，或者你怀疑是网络引起的问题，那么也是很好的机会。你可以参考我在课程里介绍的步骤，比如做抓包，然后 Wireshark 打开做分析。对着看得见的报文来学习网络协议，这个体验确实比单纯看书或者 RFC 有趣得多，也更容易坚持。这也是我的一个小秘诀，分享给你。

另外，你也可以参考这么一条学习和实践路线，来逐步深入掌握 TCP/IP 网络知识：



好了，我的学习资料和学习工具，大概就介绍到这里。欢迎你畅所欲言，在留言区里分享出你认为好用的学习方法，相信我们都可以学到一大筐非常好用的工具和资料，我也很期待！

上一篇 春节特别放送（二）| 聊聊能力陷阱和终身学习

下一篇 春节特别放送（四）| 测一测你的网络排查能力

精选留言 (2)

写留言

宝仔
2022-02-02

老师的技术能力望尘莫及啊，同样一个十年运维老兵，linux内核源码想都不敢想。想问下老师c语言这个技术能力是怎么学的？从老师的专栏中知道老师至少会两门编程语言，我现在其实也是会python，php，现在自己在学go，我学go的目的其实很简单，我想在遇到docker、k8的时候碰到棘手的问题看看源代码

展开

作者回复: 我没有那么厉害：）还是这个感想：每个技术人找到自己的核心竞争力并不断锤炼提升，能保证我们在技术和职业上持续进步。运维这个行业变化很大，传统命令行运维早已是过去式，我们有两个选择：1. 运维开发，做运维自动化平台，让自动化解决运维的难维护、风险大等矛盾，比如把手工SOP逐步转化为自动化工具，不仅解放自己（从日常重复工作中解脱出来），也提升自己（不断提高运维开发能力），最终你就是一个开发，是一个软件架构师，职业天地更加广阔；2. 领域专家，比如网络、操作系统、存储等方面，持续深挖技术底层到别的领域的同事无法企及的深度/高度，把你的技术“招牌”打造到普遍认同的程度。而且，越是底层的技术，越能经受时间的考验，有一种“历久弥新”的魅力，可以让我们的技术生涯长青。
我学习c语言和内核是为了网络这个核心能力服务的，所以我目的很明确，做起来动力也就够。你有心学习go，为了遇到问题时候能看源码，这个也跟我的想法类似，加油：）

4

陈建-binary

2022-02-10

老师好，linux内核网络有什么资料推荐吗，现在在做云，会涉及linux内核参数修改之类，好像没有啥入门头绪。