33 | 特别放送：聊一聊程序员学英语

2019-11-25 四火

全栈工程师修炼指南 进入课程 >

讲述：四火
时长 14:06 大小 9.70M

你好，我是四火。

又到了相对轻松的特别放送时间，这一次，我想聊一聊程序员对于英语的学习。我在专栏最开始的 [学习路径] 中就提到了工程师的一些“基础”能力，比如数据结构和算法，当然也包括英语。当时我说，**对于进阶的工程师来说，英文能力是突破天花板的一个必选项**，而且英文是所有进阶的软件工程师应当强化的能力，对全栈工程师来说更甚。但是我在当时并没有展开论述，为什么在中文技术材料如此丰富的今天，在工作环境是以中文为主的情况下，英语的学习依然那么重要。

为什么英语是必选项？

我记得在 2017 年的时候，就有一股讨论高考是不是应该取消英语的风潮，有不少反对者都说，英语学习了也用不上，可是直到现在，语文、数学和英语，这三门依然是高考中的公共科目。

不知道你还记不记得高中化学课学的，使用氯酸钾在二氧化锰的催化下制氧气，软件技术的职业上升进程就像是氯酸钾分解的过程，而英语就像是软件技术进阶的催化剂，它自己单独未必能给你带来多大的价值，但是掌握了它的软件工程师，视野是完全不一样的。在这里我不想谈论英语学习是否具备一般必要性，但是对于程序员这个特殊的职业来说，我想来谈一谈它重要的原因。

信息获取的最有力工具

其实最重要的原因说起来很简单，它并非是什么需要和世界人民沟通交流这样冠冕堂皇、牵强附会的理由，而是在于程序员这个职业的特殊性。

程序员需要长期地学习，而大多数的学习材料，都是使用英文撰写的。特别是对于基于 Web 的全栈工程师来说更是如此，全栈技术迭代很快，新的技术资料一般都是英文的，英文能力，尤其是英文的阅读能力会很大程度地影响知识获取的速度。

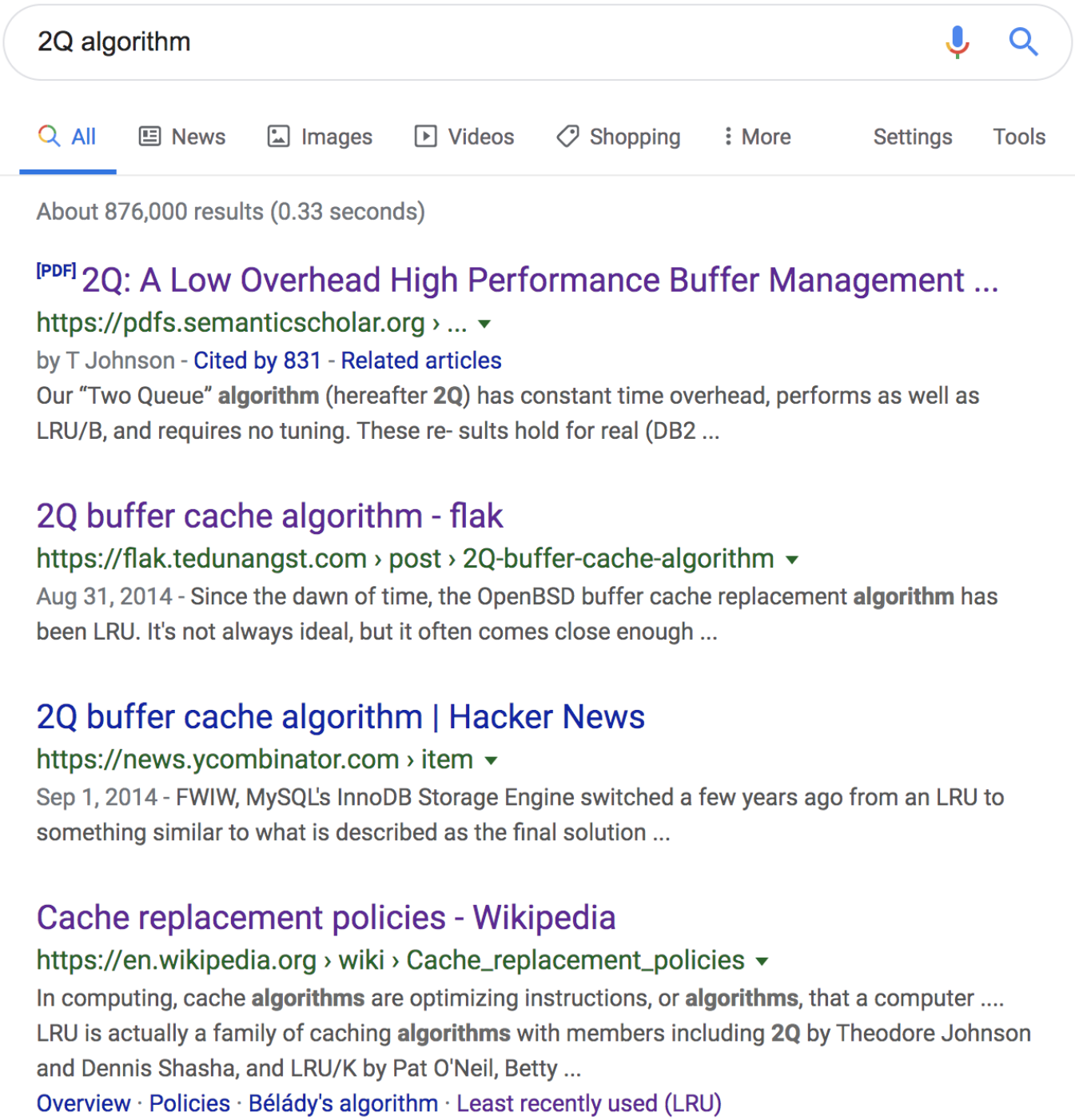
我随便举个例子。

本专栏在介绍缓存的 LRU 算法的时候描述了 LRU 的缺陷，而有一些算法设计出来的目的就是为了解决这个缺陷，2Q 算法就是其中之一。

如果你使用 Google 搜索“2Q 算法”，你会看到类似这样的结果：

基本上结果的相关性不错，但是这 4 条最靠上的结果都无一例外地是一些个人学习和分享的博客，对于 2Q 算法的笔记和理解。拿百度搜索，结果也类似，其中 CSDN 的网站权重要高一些，并且结果页的右边多了一个毫无关联性的搜索热点消息栏。

但是，如果使用英文，搜索“2Q algorithm”，结果页如下：



也一样看看这 4 条最靠上的结果，相关性依然保持的同时，相对有更有价值的内容：

第一条，是 2Q 算法□提出的论文，这显然是对于该算法最权威的材料了，并且标题下面列出了引用数 831，以及论文相关文章的链接；

第 2 条是针对该算法的一个介绍和改进；

第 3 条是 2Q 算法的讨论；

而第 4 条则是维基百科关于缓存替换策略的页面，2Q 算法就是被设计用作缓存替换算法的，这些算法被放在一起横向比较。

如果你进一步点击查看维基百科的这个缓存替换策略的页面，你依然可以发现，英文页面要远比中文页面内容丰富得多：

Cache replacement policies

From Wikipedia, the free encyclopedia

> *This article is about general cache algorithms. For detailed algorithms specific RAM, see CPU cache.*

In computing, **cache algorithms** (also frequently called **cache replacement algo** a hardware-maintained structure can utilize in order to manage a cache of informa memory locations that are faster or computationally cheaper to access than norma the new ones.

Contents [hide]

1 Overview
2 Policies
- 2.1 Bélády's algorithm
- 2.2 First in first out (FIFO)
- 2.3 Last in first out (LIFO)
- 2.4 Least recently used (LRU)
- 2.5 Time aware least recently used (TLRU)
- 2.6 Most recently used (MRU)
- 2.7 Pseudo-LRU (PLRU)
- 2.8 Random replacement (RR)
- 2.9 Segmented LRU (SLRU)
- 2.10 Least-frequently used (LFU)
- 2.11 Least frequent recently used (LFRU)
- 2.12 LFU with dynamic aging (LFUDA)
- 2.13 Low inter-reference recency set (LIRS)
- 2.14 Adaptive replacement cache (ARC)
- 2.15 Clock with adaptive replacement (CAR)
- 2.16 Multi queue (MQ)
- 2.17 Pannier: Container-based caching algorithm for compound objects

你看，英文页面中有这样一系列缓存替换算法的比较，每个算法都有具体说明；但是中文页呢，什么都没有，只有一个简单的介绍。

缓存文件置换机制 [编辑]

维基百科，自由的百科全书

缓存文件置换机制是电脑处理缓存存储器的一种机制。

电脑存储器空间的大小固定，无法容纳服务器上所有的文件，所以当有新的文件要被置换入

缓存文件置换方法有：

- 先进先出算法（FIFO）：最先进入的内容作为替换对象
- 最近最少使用算法（LFU）：最近最少使用的内容作为替换对象
- 最久未使用算法（LRU）：最久没有访问的内容作为替换对象
- 非最近使用算法（NMRU）：在最近没有使用的内容中随机选择一个作为替换对象
- Belady's Min

概述 [编辑]

内存的平均引用时间为：[1]

$$T = m \times T_m + T_h + E$$

其中

T = 内存平均引用时间
m = 未命中率 = 1 – (命中率)
T_m = 未命中时访问主内存需要的时间 (或者在多层缓存中对下级缓存的访问时间)
T_h = 延迟，即命中时引用缓存的时间
E = 各种次级因素, 如多处理器系统中的队列效应

需要强调的是，我不希望被误解，我的意思不是说不能使用中文搜索技术材料，通过这个实际的例子，我是想向你说明，在计算机科学领域，大多我们查询技术资料的时候，使用英文确确实实能带来多得多的好处。

在这个领域里，**学会英文就多了一门获取信息的最佳工具，并且这个工具往往还远远不是别的工具能替代的。**在之前 [第 06 讲] 的特别放送中，我曾经介绍过，如今互联网十大企业，中国占了 4 家，美国占了 6 家，因此如果能够掌握好中文和英文这两门语言，程序员在信息获取上的优势就显而易见了。

随着我们的生活逐渐被微信朋友圈、微博这些社交媒体所“统治”，我们更要注意技术材料的权威性，以避免那些错误的、走样的信息，因此我们需要使用英语去寻找自己需要的资料，并且去官方、可信的渠道去寻找资料。特别是当你可以阅读两份材料，一份英文的原版材料，一份经过了翻译，你会觉得哪一份更可靠呢？

给自己更多的可能性

每过一阵子，互联网就会刮起一阵“无用论”的旋风，除了前面提到的“英语无用论”，还有曾经的“数学无用论”，提出这个说辞的人理由居然是因为“生活中用到的数学”只是买菜那点而已，生活中不需要微积分，不需要线性代数。

有人在回复中讽刺持有这些观点的人“只知道买菜”而已，其实，我倒认为这种观点有它“话糙理不糙”的部分，观点朴素，却是有一定道理的。事实上，倘若你回头看看你所学习的知识和掌握的技能，无论软硬，在你实际的工作中，能使用到的总归只有其中的一小部分。

然而，话要说回来，**我们对大多数技能的学习，是为了给自己的未来带来更大可能性的。**就像软件领域的 Web 全栈技能一样，英语是一个更加广泛的带来更多可能性的技能。你可以参与到更大影响力的项目中去，你对雇主的选择会更加广阔，你可以在世界更多的地方游历。事实上，职业生涯经常会发生“无心插柳柳成荫”的飞跃，我们所做的学习和积累，就是让自己准备好，在机会到来的时候尽量不要失去。

有哪些学习英语的策略？

顺理成章地，从“为什么”到“怎么样”，接下来就该说说程序员学习英语的策略了。这些策略是从我的角度来描述的，坦诚地讲，我并不聪明，学习英语的过程也颇为坎坷，高考的时候英语是拖后腿的科目。

我知道有些朋友英文基础很好，或者英文的学习能力很强，那么我估计就不需要我这些所谓的“经验”了，但是如果你和曾经的我一样，花过不少时间，可英语的学习效果还不好，那么你就可以听听我的介绍了。

首先我想说的是，程序员总是希望英语学习的投资能够尽量得到高回报，因此一般情况下，**我并不赞同所谓的“听说读写”均衡发展的观点。你能做到均衡发展当然好，但是事实往往是残酷的**，如果你还在学校里，那么还好，可是工作以后就不再如此了。也许生活每天都很

忙碌，每天能抽出的时间并不多，更不可能像鸡汤文里说的那样，嘴里含一块小石头去海边练习发音。

听和读重于说和写

英文的“说”和“写”往往并不容易练习，除了立志去英文环境发展的程序员以外，我建议你可以着重关注于“听”和“读”。众所周知语言的学习是需要长期强化的，有了如今便捷的互联网，英文文章、英文电影、英文新闻，只要自己愿意，我认为**“听”和“读”的不断刺激强化已经不成问题了，它们属于信息接收，但是“说”和“写”属于信息表达，后者更需要环境的浸染**，如果仅仅靠一周个把小时的英语角，发几封英文邮件，或者是只进行缺乏互动的练习，进步是很慢的。

就我自己的经历而言，我在大学里面花费了大量的时间去学习英语的说和写，但其实效果并不好。那时候掌握的是语法和一点词汇基础，可以在思考以后说出、写出“正确”的句子，可以用来考试，但没法使用。

这里有两个原因：一个是思维还是中文思维，说的时候需要思考，在大脑中还需要进行从中文到英文的翻译过程；另一个是表达的方式并不是实际的、常用的，而是自己生硬的翻译。因此英语实际应用的提高，基本上都是在工作以后，因为有了实际的需要，以及英语环境，慢慢就给拧到英文思维了，并且逐渐掌握了表达的惯用法和一些技巧，这些都和读书时候学得不一样。

听：多样的口音

我曾经有一个误解，以为英语的学习就要尽量去找那些 VOA、BBC 这样的纯正发音的材料，但实际上根本不用这样。无论是工作还是生活，我们接触的英文都是杂七杂八的，**各种各样的发音，各种各样的表达法。**通过对于不同口音听力的训练，能够让自己对于英语听和理解的能力得到提升。这就有点像打乒乓球，要和各种各样风格的人过招，有直板有横板，有攻有守，自己的功力才能提升。

说：关注内容，而不是发音和语法

上面一条是关于“听”的，这一条则是关于“说”。对于英语能力较初级的朋友来说，发音也好，语法也好，总是有很多的问题，而即便经过了再多的练习，口音和词句的用法往往还是会和“地道”有所区别。

但是，这又有什么关系呢？平时的沟通，还是要把关注点放在内容上面。**无论语法有哪些错误，发音有哪些错误，实际上它们的重要性都远不如把“内容”表达出来高。**练习使用清晰、简洁的逻辑把问题描述清楚，真正达到沟通和交流的目的，日常工作生活中，不会有多少人在意你的发音。

读：从技术材料的检索和阅读开始

英语学习和技术学习，我们当然希望一举两得。我知道在开始的时候，这会比较困难，毕竟谁都有自己的舒适区。事实上，在写这个专栏的过程中，我本来找的扩展阅读材料绝大多数都是英文的，后来在编辑的建议下，才尽可能地把其中我能找得到类似质量和主题的材料换成中文的。

技术材料的阅读可以根据自己的情况循序渐进，但是在开始的时候，你要有个预期，就是阅读英文材料的速度肯定是要慢于中文材料的。作为程序员，我们也可以把自己的英文阅读目标范畴基本定在技术文档上面，从我的经历来看，技术上从中文逐渐适应到英文，还是要比生活上的切换简单得多得多。

写：学习那些文档和邮件中的惯用法

仅仅是靠自己写，没有反馈的话，你不会知道这样写对不对、好不好。很多常见的用法课本里不会写，老师未必教，你可能很明确自己的表达从语法上看是不是正确，但是大家却未必这样使用。因此阅读那些英语母语的程序员写的的技术文档，就是一个很好的积累惯用法的方法。而英文邮件，则是另一个很好的例子，阅读它们，可以积累一些书面上怎么表达的例子，怎么提问，怎么认可，怎么否定，怎么请求帮助，等等。

寻找乐趣

这是最后一点，也是最重要的一点。没有了乐趣，所有的学习都是事倍功半。最理想的状况，应该是乐于做一件事，做完了，还在不知不觉中获得了自己想要的东西。我知道有很多程序员朋友就是因为喜欢编程才逐渐走上了程序员这条路，郭德纲说过：“如果你的工作也是你的爱好，那是老天爷疼你。”英语学习也是如此，如果你在努力的过程中能得到更多的乐趣，那就是绝妙。

拿我自己来说，有两个时期我自己明显感觉英文进步比较大。一个是在读高中的时候，我是《最终幻想》的游戏迷，《最终幻想 VIII》巨长的对白文字，我当时硬是凭借一个文曲星把剧情啃下来了；另一个是工作以后，美剧《Friends》断断续续看了好几遍，从一开始看中

文字幕，到后来看英文字幕，以及再后来大致可以脱离字幕……这其中，兴趣的功劳是第一位的。

好，今天的特别放送就聊到这里。这是我的体会和分享，现在我把话筒给你，不如你也说说你的故事？

上一篇　32 | 和搜索引擎的对话：SEO的原理和基础

精选留言 (1)

写留言

anginiit
2019-11-25

是啊 好多技术疑问没有中文文档 搜到的英文原文是一句一句啃 但还是不太明白 英语学习是必须的 这也是我列入下一年的首要计划

展开