

# 一个平庸程序员的自白



zhuoqun 2015-06-05 02:31:46

Jacob Kaplan-Moss 是著名 Python 框架 Django 的 Co-Creater 和核心开发者，在 Heroku 担任安全部门的 Director，他常年参加 Python 社区的年度聚会 PyCon，并经常上台演讲。在这次 PyCon 2015 上，他声称自己顶多只是一个平庸的程序员，幻灯片上也写着：

"Hi, I'm Jacob, and I'm a mediocre programmer."

这次演讲非常精彩，视频可以在 Youtube 上看到（如果不想看视频，这里也有一篇文字总结：<https://lwn.net/Articles/641779/>）。Jacob 以马拉松为例，试图说明大部分人在大部分领域都是平庸的（Most people are average at most things.），特别好的和特别坏的都是极少数，但在程序员领域，我们听到的大都是某某程序员 rocks 或者 sucks，基本没有中间状态，这是一件非常奇怪的事情。他还提到“真正的程序员”（real programmer）这个概念，也就是所谓的 "10x programmer"，你只有精通某些原理或者长得像 Mark Zuckerberg 才算是一个 real programmer，而一个使用各种云服务以及 API 快速搭建自己应用的人就不算，这非常可笑。顺带一提，Jacob 的 Twitter 个人简介就是 "not a real programmer."

Jacob 认为这种关于“编程天才”的神话非常有害，一方面它把行业门槛设置得特别高，令很多人望而却步，另一方面它也在折磨行业内的人，因为你如果不能 rocks，就会变成 sucks，所以不得不用一切时间来努力学习和工作，导致影响生活。他认为我们应该改变这种态度，编程只是一些技能，并不需要太多天分，它是可以学习的，而且做一个平庸的程序员不丢人，"it's cool to be okay at these skills"。跑步的人有很多种——短跑运动员，纯粹为了锻炼身体的人，参加马拉松的人，等等。为什么编程就一定要做 real programmer，而且做不到就觉得自己不配编程呢？

这个演讲在大会现场赢得很多掌声，在提问环节也可以听到一些怯生生的声音说：“我也是一个平庸的程序员。”不过在其它地方也引起一些争议，有人认为拿跑步和编程对比并不恰当，有人说程序员能力的差距确实大得惊人，有兴趣的人可以去看 Hacker News 上的讨论。

对于我个人来说，Jacob 的演讲触动了我多年来的心结，让我非常感动。我一直认为自己顶多只是一个平庸的程序员，而且无论多么努力学习，总觉得自己不是一个 real programmer，而且好像越走越偏。Jacob 的演讲让我明白，有此苦恼的大概不只我一人，所以这里写一下自己的心路历程，希望对别人有一些启发。

公开承认自己平庸是一件危险的事情，如果不能说明你智商低或者没有天分，至少说明你态度有问题：“平庸就应该去努力练习，而不是写文章为平庸辩解”。而当你说 Jacob 这样的人也承认自己平庸时，就会出现一堆带有“资格论”的反驳：“Jacob 至少写了那么多代码，你怎么能和别人比”“人家那样说是谦虚，你这样就是无耻了”，顿时你就会觉得自己和那些只会从伟大人物身上挑毛病的小人没什么两样。幸运的是，今后我很可能不会以编程为业了，所以这些事情都无所谓了，可以诚恳地说一下自己的经历。

我对 real programmer 的印象始于大学，尽管对于其它专业的人来说我们和修电脑的没什么两样，但专业内部已经隐约有对某种高级事物的崇拜，记得大一时一个同学对我说：“我们现在还学 C，外面早就开始用 Java 了”。那个时候专业内也已经流传了不少程序员的传说，比如某个学长多么厉害，或者微软某个程序员一个人做完了一个项目组的工作。相信很多人都听说过那个故事，大意是一个人电脑坏了，于是找一个电脑高手来修，电脑高手拿来电话，连上电脑，花了整整一夜的时间用 0 和 1 写了一个操作系统，故事的结局是“这个电脑高手去了美国，杳无音信”。当时我听到这个故事的时候半信半疑，居然宁愿相信它是真的！因为如果万一它是真的而我不相信，只能说明我的技术水平或者智商无法理解那个层次的东西，那就太糟糕了。那个时候真的很想变得厉害。

随着对专业的了解，很快我们就开始嘲笑那种“Java 比 C 新，所以比 C 好”的看法，转而去相信“越底层越牛”的论调，

写 C 的比写 Web 的厉害，写汇编的又比写 C 的厉害，为什么呢？因为他们“基础更扎实，更了解原理。”这种论调发展到一个极致就是：学数学或者物理这种“硬科学”的人如果转行写程序，肯定比我们厉害。他们算法一定比我们好，而且就是这帮家发明的计算机啊我靠。有一段时间隐约觉的别的专业的转行写程序都会比我厉害，哪怕是一个钢琴家——因为他手指灵活打字更快。

大学里另外一个让我印象极其深刻的传说就是：高手写程序不用 IDE，都是直接记事本。高手摘花飞叶皆可伤人，谈武器和工具，那就是落了下乘，只有不入流的才整天研究哪个工具更有效率。所以在很长一段时间内我都因为自己没用记事本写程序而怀疑自己的能力，直到后来工作之后我接触到了 Vim，发现它长得和记事本还真的挺像的。

除此之外，还有对经典书籍的顶礼膜拜，MIT 的 Introduction to Algorithms 和 SICP，Knuth 的 The Art of Computer Programming，TCP/IP 详解，以及一些其它的经典操作系统、数据库等方面的书。不啃完这些就说明你天分不够，或者没有热情，还是趁早转行比较好。在我一直很喜欢（现在依然喜欢并推荐）的一篇文章 Teach Yourself Programming in Ten Years 中，作者在结尾推荐了 SICP，他说："The book is challenging and will weed out some people who perhaps could be successful with another approach." 所以尽管我一直没有读完这本书，而且应该也不会去读了，但对它总有一种情结，以至于两年前出国的时候还把它带在身边。

毕业之后开始做职业程序员，每天都是和程序员打交道，关注的也都是科技新闻，那时候硅谷文化也越来越流行，所以更加向往和崇拜 real programmer。Joel Spolsky 写过一篇《飙高音》，其中的两段可以算作是对 real programmer 的最佳诠释：

用许多平庸的程序员取代少数优秀的程序员，这种做法的真正问题在于，不管平庸的程序员工作多长时间，他们做出来的东西，都无法像优秀程序员做得那样好。

.....

一流的歌唱演员不管在什么时候，都可以很轻松地唱出高音，而平庸的歌唱演员就是永远做不到这一点。莫扎特的歌剧《魔笛》中有一段著名的咏叹调"Queen of the Night"，音高必须达到F6（女高音的最高音高）才能唱好这首歌，世界上能达到这个标准的女高音都快要绝迹了，而飙不到著名的F6，你就是不能表演"Queen of the Night"。

简单地说，如果你缺乏天赋，那么你再努力，也做不到 real programmer 可以做到的事情，而一个平庸的程序员就是一个烂程序员。这种断言式的文章一直比较能鼓舞人心，那些认为自己比较聪明的人也很喜欢这种说法，但对于我这种一直怀疑自己智商和能力的人来说，它就像一个阴影：万一我真的没办法飙高音怎么办？

因为这种对 real programmer 的仰视，从工作以来，不对，是从大学以来，有一个想法一直挥之不去，那就是：I don't deserve it.

大学里我也有作品拿过奖，但总觉得自己的代码是一坨屎，不配拿那个奖，如果当时有另外一个 real programmer，肯定轮不到我。

大四找工作的时候，尽管我是通过正规的笔试和面试拿到的 offer，而且被录用的只有我一个本科生，我还是觉得自己不配得到那份工作，因为没写过 10 万行代码，也不是一个 real programmer.

刚工作的时候我每天从早晨 9 点到晚上 9 点都在公司，有时候周末也去，手头工作做完了就看技术文档，无聊了就用打字软件练习正确的指法，以便编程可以快些。即便这样，我还是怀疑自己对编程的热情，觉得自己离一个 real programmer 越来越远，不适合做程序员。

后来去雅虎北研面试，从早晨 8 点面试到 12 点，第二天又跑去美国的同事视频面试。测试其实已经很严格了，而且我平时没有针对性地刷题或做其它应试准备，但很奇怪，我还是觉得自己能力不行，只是侥幸拿到了 offer.

平时帮别人写代码，或者和别的部门合作，第一个念头就是自己的能力不行。尽管最后项目顺利完成，对方也对我评价不错，但我还是觉得自己不配得到那些夸奖，甚至想：如果他们见到 **real programmer**，就不会称赞我的工作。

这种心理还有另外一个负面作用，那就是时时刻刻焦虑，害怕掉队。**Jacob** 在演讲中也说到：

If the only options are to be amazing or terrible, it leads people to believe they must be passionate about their career, that they must think about programming every waking moment of their life. If they take their eye off the ball even for a minute, they will slide right from amazing to terrible again. That leads people to be working crazy hours at work, to be constantly studying programming topics on their own time, and so on.

这段话准确地说出了我以前的某种状态。我对除了编程之外的很多事情都有兴趣，从生活的角度来看，这其实是一件好事，但我却常常因此有负罪感，因为它们耽误了我编程、看文档和学习新技术的时间。那些事情本来是可以带给我快乐的，但每次做好像都有一种偷偷摸摸的感觉，被别人问到也羞于提起。现在回头看，之前有段时间我的工作和生活其实平衡得很好，但当时我只有不断告诉自己“你本来就平庸”，甚至把自己想象成一个自甘堕落的人才能安心享受，实在太扭曲了。

以前我有时还喜欢写点东西，认为自己是在分享，但后来总觉的自己又不是 **real programmer**，分享的东西有什么价值呢？而且写文章会耽误编程的时间，程序员文化又特别推崇“务实”，反对“务虚”——写作就是“务虚”，“只会打嘴炮”在程序员群体里面是一个非常严重的侮辱，记得之前看到过一条招聘信息，明确地说“不欢迎经常写博客的人”。事实上写作不止是表达，它反过来还可以帮助思考，很多想法在下笔之前是不存在的，我从中获益良多。而且，“不愿意表达”和“不会表达”有着本质的区别。以前如果一个猥琐不堪、一句话都说不清楚的人过来朝我扔一句“**Talk is cheap. Show me the code.**”，我立刻就会惭愧不已，甚至认为他或许就是一个 **real programmer**。

以上种种不安和困扰，让我常常觉得自己从性格到业余爱好都不适合做程序员。程序员文化告诉我的是：如果你不是一个 **real programmer** 或者不在成长为一个 **real programmer** 的路上，那么你连一个 **programmer** 都不是。**Hacker News** 上针对 **Jacob** 演讲的讨论中，**panjaro** 说了他自己的经历：

I'm an example how this myth can destroy you. I was programming for 6 years in normal projects and I thought I was not talented. I left job, came to research degree. Now I realized I'm terrible at maths and advanced algorithm techniques. I think about going back but since it has been 3 months I haven't worked in a company, I feel I am a very bad programmer and no one will hire me. I'm depressed and paralyzed. All I do all day is look into HN and read articles, think how all these people are talented and I'm so dumb.

他的描述让我感同身受，当初我也痛恨自己对数学不感兴趣，整天在自习室苦苦复习离散数学和算法，工作之后也常常想要重新回头学习这些“基础知识”。有时候即使你发现在某些工作中算法并不像传说中那么重要，也不能说出来，否则你就是忽视基础，或者你从事的就是程序员当中的低级工作，以至于体会不到算法的重要性。

程序员文化当中“能力决定一切”的态度很多时候都值得商榷，因为除了极少数的计算机科学家，大部分程序员从事的是产品开发工作，而在产品开发中是没办法用一个简单的标准衡量一个程序员的能力的。一个人可能编程能力很强，但他可能执意要实现自己的想法，以至于拖累整个项目。另外，编程能力强的人不一定能从事某些“低级的”、不需要太多智力的工作，那样的工作可能需要大量的耐心，而耐心和传说中的天赋一样，也不是一种人尽可夫的品质。以前看到过一个招聘信息很有意思，里面说希望应聘者“已经结婚，或者至少有过一段长期的恋爱关系”，他们觉得这样的人懂得与别人相处，不太容易是一个鄙视别人工作的 **jerk**。

我并不是否认大牛的存在，**Linus** 和 **John Carmack** 就不用说了，刚毕业在中国雅虎的时候就遇见过一位大牛同事，清

华毕业，基督徒，已婚，业余时间喜欢编程和做饭，经常一个人做完整个团队的工作，打字奇快，每次技术分享都有很多人去听，甚至 HR 都因为好奇过去。听众提问的时候，他能清楚讲解某个不常用命令的某个参数的作用，简直就是活的 **user manual**。他非常乐于助人，经常看到他在 **mail list** 里面回答别人问题，从问题的成因分析到解决方案都一一给出，表达非常清晰。让我比较惊讶的是他对很多琐碎的事情也极有耐心，记得公司想要一些人申请专利，周末就看到他在那里专心写申请。另外一个和我比较要好的同事本来觉得自己还算有天分，比我厉害得多，在大学里真的写过 10 万行代码，但见过大牛之后立刻发现自己其实没天分。那个时候他女朋友在外地，他业余时间也就是看看电影写写代码，不想写的时候就打开大牛的 **Github** 看看，一般都会发现他又有新的 **commit**，然后就有动力继续写了。

问题在于并不是所有人都是大牛，整个行业需要的也不止是大牛。举个例子，很多写 **PHP** 的大概看不起那种“**Wordpress** 工程师”，就是只懂 **Wordpress** 而对 **PHP** 了解不够深入的人。但很多公司确实就需要做一个小网站，只需要你能开发 **Wordpress Theme** 就够了，不需要研究更多东西。帮别人开发 **Wordpress Theme** 完全可以谋生，甚至挣得不少，但他们大概自己也觉得自己不是 **real programmer**，所以很少发声。前一段时间有个应用很火，但他们团队的文章开头就说“我们团队中没有大牛”，好像那是很丢人的事情，一定要大牛组成的团队才配开发出好东西？如果抛开其它不谈，“一个人凭借自己的能力养活自己，他就应该得到尊重”，这话听起来没有任何问题，只是一旦牵扯到代码，好像只研究 **Wordpress** 的工程师就不能见人。

很多公司也迷信只有大神程序员才能做事，所以在招聘的时候会他们说只招聘 **Top 1%** 的程序员，这其实根本不可能，**Joel Spolsky** 早就写文章谈过。你不可能招聘到 **Top 1%**，只可能招聘到应聘者中的 **Top 1%**。我们可以想象一下，最厉害的人可能直接去了 **Facebook**, **Google** 或者硅谷的创业公司，差一点的可能选择百度腾讯等国内巨头，或者非常有前景的创业公司，剩下的人才才是市场中的人，这里面的 **Top 1%** 已经不是全部人中的 **Top 1%** 了。所以看到这种招聘可以评估一下这个公司，如果不是特别牛，完全可以大胆去应聘，因为你在那些愿意应聘这个公司的人当中，可能已经是 **Top 1%** 了。

其实我非常喜欢这个行业的某些理想主义色彩，过去这几年也认识了不少朋友，发现了不少 **thinker**，对我启发很大。可惜我常常感到不舒服，也觉得和这种程序员文化或者 **Geek** 文化格格不入，虽然很努力去尝试过，但发现还是不行。当然，也可能并不是文化问题，而是因为像我和 **panjaro** 这样的人胆小浮躁，太在乎别人看法，容易焦虑和抑郁，这样的人大概在任何行业都是 **loser**。希望其他人可以活得理直气壮一点。

过去常常希望找到人生目标，而当怀疑自己不是 **real programmer** 的时候又庆幸生活是一团乱麻，而人生可能并没有意义，不然的话，可能每时每刻都要生活在一条“鄙视链”当中了。