

程序员成长系列



程序员 成长的烦恼

华中科技大学出版社
<http://www.hustp.com>

吴亮 周金桥 著
李春雷 周礼

程序员成长系列

程序员成长的烦恼

吴亮 周金桥 李春雷 周礼 著

华中科技大学出版社
中国·武汉

前言

这是一本关于程序员奋斗成才的书。四位作者来自不同的技术领域，他们以各自不同的经历，演绎了各自平凡而精彩的人生。

我有幸受邀参与了本书的撰写。作为作者之一，在本书出版之际，我仅从个人的角度谈谈写作的心得，以及此书想要传达的一些理念。

对习惯于编写代码的我们来说，写一部人文类图书，是一个不小的挑战。生活虽很平淡，人生也谈不上成功，但回望一下毕业这么些年的足迹，其间的困惑和迷茫、欢笑与泪水，对于那些有志于在程序员道路上有所作为的大学生，和入行不久正经历如我当年一般困惑的职场同仁，还是有一定益处的。

本书四位作者，来自不同的公司，有着不同的经历，但有一个共同点，就是热爱程序员的工作。当初的我，也是因为热爱这个职业而转行的。工作十年间，遇到过很多挫折，走过很多弯路，始终不变的是对行业的热爱。能把爱好和职业统一起来，是人生的一大幸事。所以，我一直以感恩的心，从事着自己的职业。

曾经有一年时间，由于种种原因，我的生活几乎陷于困顿，似乎看不到希望，当快撑不下去的时候，我对自己说的最多的一句话就是：这是暂时的，终将会过去的。我终于熬过来了。现在回想，那间 200 元租来的民房，一直充满了温馨；蹲在路边吃的那碗热干面，相当美味；邂逅了那个女孩，她的笑容充满阳光。我想说的是，不

管处在人生的什么阶段，也不管是欢笑多还是泪水多，只要心中充满爱与理想，生活处处都有精彩。

是的，对生活和工作充满爱与理想，我们就是这样一群平凡而快乐的程序员。

如果读者能从本书中读到这些并有所感悟，那么我们的写作也就变得更有意义。

此书能够顺利出版，首先要感谢陈禹成编辑和徐定翔编辑。没有陈编辑最初的鼓励与组织，就没有此书的编写；没有徐编辑专业的建议，就没有此书现在的模样。还要感谢曾经的老板与同事，没有你们的严格要求、精诚合作和“勾心斗角”，就没有现在的“我们”，就没有书中那些精彩的故事。

当然，还要感谢作者的家人、同学及朋友，是你们的支持，才让我们有了前进的动力。

最后，请允许我在里这特别感谢一个人：我的二姐——冯广秀。在我女儿小焰焰眼里，你就跟妈妈一样。

李春雷

2011年2月于武汉

目 录

吴亮的故事.....	1
第一章 初识编程.....	3
第二章 我的大学.....	17
第三章 踏入社会.....	38
周金桥的故事.....	63
第一章 高中时代.....	65
第二章 大学时代的学习.....	68
第三章 初入社会.....	77
第四章 程序员之路.....	86
第五章 上海求职记.....	92
第六章 太原求发展.....	98
第七章 ASP.NET 夜话诞生记.....	112
第八章 重回武汉.....	117
李春雷的故事.....	123
第一章 关于理想.....	125
第二章 我的大学.....	127
第三章 工厂岁月.....	133

第四章	艰难入行	141
第五章	Java, 我来了	152
第六章	在项目中快速成长	158
第七章	北漂的日子	176
第八章	重返江城	195
第九章	给进阶中的 Java 程序员的建议	201
周礼的故事		205
第一章	大学那会儿	207
第二章	那些蹉跎的岁月	219
第三章	我的感情生活	233
第四章	新的征程	246

吴亮的故事

想把我的经历写得轻松一些，所以文字就从梦开始吧。

这里的梦不是指睡梦中的梦，而是说梦想。什么样的梦想呢？是儿时关于未来的梦想。为什么不说是理想呢？是因为我想表达两个意思，一是梦想带有一丝感性、一份想象力和一点点童话的味道；二是理想可能包含别人对你的期盼，这份期盼也许是长辈、老师、偶像或者身边的人带给你的，这些也许是现实的和善意的，但这不是你的梦想。

现在，请你闭上眼睛好好想想，把那些不属于你自己的部分从内心暂时拿掉，剩下的部分，就是你自己真正的梦想。梦想很有用，梦想是一切的缘起，如果有可能，请追寻自己心中的梦，在实现之前，不要停下脚步。

第一章 初识编程

追寻心中的梦

先从我最初的梦想讲起，正是这个梦想使得我最终走上了程序员这条道路。

儿时的我是个很爱玩的小孩，对电子游戏没有多少抵抗力。我的父母都是普通的国企职工，家庭条件虽然还可以，但也谈不上富裕。我很少缠着父母买东西，唯一的一次是在一年过生日的时候，我缠着父亲买了一台任天堂的红白机。

虽然我向父母保证了严格限制玩游戏的时间，但我仍然在周末、寒假和暑假里把各种游戏卡带疯狂玩了很多遍。很快，我就发现了自己面临的问题，旧卡带上的游戏玩腻了，能借的卡带和好玩的游戏也越来越少。有时候事情就是这样奇怪，没游戏机时，也不会觉得什么，可当你一旦有了它，麻烦就来了，没有新鲜游戏玩的感觉是那样空虚和无聊。于是我就想，什么时候要是自己能编写游戏玩该多好。

在我开始梦想有一天自己能编写游戏玩的时候，正是 20 世纪 90 年代初。当时个人电脑还未普及，它是大公司和国企里的高级玩意儿，一般人接触不到。

在我还沉迷于红白机游戏的时候，父亲的办公室里就摆着几台电脑，那时电脑给我的印象只是一个键盘接着一个方方的显示器，黑屏幕上滚动着看不懂的英文字母。我想电脑除了打字还不知道能做其他

什么。一天，我在办公室里等待父亲感到无聊的时候，看到父亲一位同事的电脑屏幕上显示出图形界面和中文，因为我在红白机上玩过类似的文字游戏，所以我很快便知道那是一个文字类型的三国志游戏，我第一次知道电脑还可以玩游戏。离开父亲的办公室之后，我仍然在不断地回想刚才看到的情形，我发觉自己对电脑产生了无法抗拒的兴趣。当时我并不知道程序是如何编写和运行的，只是认为电脑的输入终端比红白机要复杂得多。我想，也许用电脑的键盘输入一些“魔法”字符，就真的能实现我的梦想——为自己编写好玩的游戏。

三年后，在我初三的那年秋天，家里有了第一台真正的电脑——486的PC，几个月后我在DOS操作系统上编写了一段用QBasic语言编写的程序。尽管那段程序的功能简单得无法再简单，但它是真正的程序。我终于掌握了和电脑自由沟通的语言。

能做自己喜欢做的事情是幸运的。我之所以走上软件开发的道路，完全是因为儿时那个简单的梦想。虽然不是什么伟大的志向，但重要的是我始终在为这个梦想而努力，至今未曾放弃。

挑选合适的语言

有了电脑以后，我花了几个月时间熟悉操作系统和其他知识。那时的操作系统并不像现在的Windows有友好的界面，记住复杂命令是件比较烦琐的事情，但我却乐在其中。

在熟悉了基本操作之后，我开始研究如何用电脑编写程序——是的，我那时已经知道了程序这个概念。当时，国内流行一种叫学习机的东西，它是一种增强版的游戏机加上一个键盘，在上面可以编写一种简单的Basic代码。

经过摸索和查阅资料之后，我确信我的电脑上带有一种 Basic 语言的实现版本——QBasic。我沉迷于用 QBasic 编写各种程序，起先是一些简单的计算和字符的输入、输出，后来编写了一些文字类型的小游戏，比如猜数字之类，还实现了一个交互菜单。但是，经过一段时间的研究后，我发现 QBasic 的功能很有限。

QBasic 代码不能独立运行，而我想要一种真正能以 exe 文件形式运行的程序（尽管当时我对 exe 的真正原理和 PE（portable executable）头文件等概念一无所知），经过研究之后，我知道了 QBasic 是解释型的，只是一些跑在解析器上的简单指令。另外，QBasic 对底层的支持也很少（现在我知道如何用这样的语言操作寄存器、调用中断以及直接与汇编交互，可当时这些概念对一个初学者来说过于深奥）。经过一段时间的学习之后，我做出了一个决定——放弃 QBasic，学习一种更强大的语言。经过仔细挑选之后，我选择了 C 语言。

有人说程序员选择的第一门编程语言非常重要，甚至能决定程序员未来发展的道路。这种说法虽然太过绝对，但从某种程度上来说也不无道理。编程语言不是程序的全部，但是编程语言能影响程序员解决问题的思考习惯，就像语言可以影响一个民族的文化一样。

现在想起来当时的选择并不多，除了 BASIC、C 和 PASCAL 等语言之外，少有其他语言能够进入我的视野。Java 是后来才流行的语言，而 FORTRAN 语言几乎只在高校的计算机实验室里才有人使用。

我花了大量时间熟悉 C 语言的语法、理解指针等概念，因为我确信它是能够实现我梦想的最合适工具。现在计算机和程序设计如此普及，可选择的余地也大大增加了。许多高校都开设 Java 课程，讲授数据结构和算法，学习算法的朋友还能够选择 Scheme 这样简单的语言（Scheme 是 Lisp 的一种方言；Lisp 是一种善于描述问题本身的优美

语言)。几个月下来，虽然我一直在学习 C 语言的各种特性，但也只弄清楚了其中的一部分。我开始用 C 语言编写一些模块化的程序，每次写完一个程序，总要很费劲地解决由于自己对语法不熟悉而产生的编译错误。我发现自己还缺少一些知识，但是又不知道究竟缺少的是哪一部分。当时的计算机教材很少，我自学的那本书也缺少很多内容，于是我决定自己去找课上。

对于程序员来说，编程语言只是基础得不能再基础的工具，如果你想要成为真正优秀的程序员，还需要掌握语言之外的许多东西。数据结构和算法，是设计和开发复杂软件产品所必须掌握的知识。幸运的是，我很快在课堂上接触到了这些知识。

混入课堂偷学

当我逐渐熟悉 C 语言的时候，升入了省重点高中。我希望高中能有程序设计的相关课程，但可惜的是，学校并没有安排这方面的常规教学课程，只有针对参加信息奥赛班的学生的辅导课。我的一位同班同学是学校信息奥赛班的队员，向他打听到上课时间之后，我开始混入教室去旁听信息奥赛班的课程。当时在信息奥赛班上课的有几所学校的学生，因此相互之间并不认识。而我又是刚升入高中的新人，那些听课的学生以为我是其他学校的，也不会多问。这样我就堂而皇之地偷听完了几乎整个学期的课程。

信息奥赛班的课程主要是用 PASCAL 讲数据结构和算法方面的知识，虽然当时我并不能完全理解那些课程的意义，但是我想那些知识是有用的。学习这些课程后，我发现我以前对数据的理解非常肤浅，我编写的程序几乎很少出现复杂的数据结构，对指针的了解也仅限于基本的用法，学习算法课程使我理解了线性表、二叉树等数据结构，

也让我对 C 语言的指针有了全新的认识。

当时，互联网远没有现在普及。渴望学到更多知识的我，在自己找书本学习的同时，还想出了一个到学校信息奥赛培训班去偷学课程的法子。正是那些课程，使我第一次接触了数据结构和算法，为我将来的程序员之路打下了非常重要的基础。

当我升入高二时，已经能用 C 语言设计和实现稍微复杂一些的程序了。我开始着手编写实用的小工具，还用 Turbo C 图形库编写了一些带图形界面的小软件。

编写带图形界面的程序一直是我努力的方向，因为要实现编写游戏的梦想，这一步是必须的。不过我发现手中的那本教材根本没有提及如何编写这类带图形界面的程序，只有一个章节在介绍中断调用时提到过显示器相关的例子，可我当时完全没有这方面的基础，根本不能理解这章所描述的东西（中断调用，这个概念对于当时的我无疑太过深奥）。我很苦恼，直到有一天，我偶然在书店里发现一本介绍 Turbo C 图形程序设计的书，如获至宝，我现在还清晰地记得当时兴奋的心情。

很长一段时间，我沉迷于使用 Turbo C 图形库编写带图形界面的程序。比如我编写了一个带复杂的多级下拉菜单的学英文的软件，一个支持鼠标、可双人对弈的五子棋游戏（大一时我还给那个五子棋游戏增加了简单的人工智能）和其他一些小程序。我对 Turbo C 的热情持续到高二的寒假，直到父亲帮我升级了一次电脑，操作系统换成了当时流行的 Windows 95，软件更加丰富，娱乐性也更强大了。不过这次升级也带给了我新的问题——发现自己所用的 Turbo C 并不能实现 Windows 下的程序，于是我不得不继续寻找新的答案。

操作系统升级为 Windows 95 之后，编写带有窗体的 Windows 程序就成为我新的追求目标。虽然这个今天看来无比简单的目标，但在当时困扰了我很长一段时间。那时候，计算机书籍很少，几乎没有教材介绍如何编写 Windows 下的程序，于是我不得不暂时放弃编写 Windows 程序的想法，而把兴趣转向另外一个领域。

转战互联网

高二那年，暴雪公司推出了一款即时战略游戏——星际争霸。当时我就读于省重点高中，班上的同学也很爱玩这款游戏。在这群玩星际争霸游戏的同学里，有一位外号叫冰魂的，他不但是星际高手，而且是一位数学方面的奇才，在数学计算上表现了很高的天赋。冰魂接触电脑很早，加上他的天赋，很快就成为我们这群人中的高手。在高二开设的计算机课上，当其他同学还在学习打字时，他用 Basic 编写了五子棋程序。我们经常一起玩星际争霸，也一起探讨程序上的问题，后来还合作写过软件。

高二那年我家里开通了网络。当时的互联网不像现在这么普及，也没有 ADSL 或者小区宽带，更别说 3G 网卡了。当时上网，只能选择 33.6 KB 的 Modern（俗称猫），不论上传还是下载，速度比现在以兆字节计算的网路速度慢得多。当时互联网上的内容也少，形式较单调，远没有现在这么好的互动性。即使如此，上网也是一件令我很高兴的事情，因为我发现，用电脑可以迅速地获取信息，与陌生人交流。

那些和我们一起打星际争霸的同学，因为经常去网吧，也学会了上网。慢慢地一些同学家里也有了电脑，于是除了学校和网吧，我们有了一个新的交流场所——互联网。

当时我们在互联网上的交流没有现在方便，既没有免费邮箱，又不流行像 QQ 这样的 IM 工具，更没有 SNS 服务。我们只能通过简单的 Web 页面、留言簿、BBS 和一种称为聊天室的互动网页服务来进行交流。

上网的同学渐渐多起来，冰魂和我就想做一个让同学们交流的平台。当时我学习 C 语言受阻，于是开始把注意力转向网页。通过学习，我知道了在互联网上看到的页面，是用一种被称为 HTML 的标记语言和被称为 CSS 的样式表来实现的。可我对编写标记语言不感兴趣，也没有设计方面的天赋，好在冰魂有不错的美术功底，所以设计和实现页面的工作就交给他去完成。

HTML 和 CSS 是客户端浏览器上的东西，它们只能实现静态的效果。如何让网页变成一个留言簿或聊天室，成为互动和沟通的平台呢？我再次陷入了困境。

这次遇到的困难比较大，我根本不知道如何入手。虽然我会 C 语言，可是 C 语言能在 Web 上跑起来么？而且当时我甚至不懂 Web 服务器的概念，互联网也不像现在有那么多资源和强大的搜索引擎。所以只能靠自己查阅资料。

后来我才明白原来网页是由远程计算机上的 Web 服务器提供的，大多数 Web 服务器支持一种通用的编程接口，叫 CGI。支持 CGI 的服务器可以用一种通用的方式调用服务器端的程序来提供动态内容，而 C 语言恰好能够胜任这项工作。我非常高兴，立即开始测试，可是当我按照查阅的资料进行操作时，发现资料中只笼统地介绍了 IIS Web 服务器，并没有说明具体的服务器配置。我对 IIS 还是一知半解，而且，当时我只懂得用 Turbo C 的环境来编译源代码，对编译选项一无所知。尽管我经过了多次努力，仍然无法在本地的 IIS 服务器上顺利

运行 CGI 程序。

接下来我又发现了一个更棘手的问题——在远程服务器上运行程序。那时我虽然找到了支持 CGI 的免费空间，但是不能在远程空间上编译出可执行文件（远程空间是 UNIX 系统，本地编译的文件是无效的）。于是，一直对用 C 语言能编译出 exe 文件沾沾自喜的我，第一次感觉到编译型语言在 Web 开发中的无奈。

发现用 C 语言编写动态 Web 程序这条路走不了，我只好去寻找另外的路。大约两周之后，我找到了自己想要的答案。

转向互联网，是我在使用 C 语言受挫后的无奈之举。可万万没有想到的是，几年后，我会真正与互联网和 Web 开发结下不解之缘。也许，一切皆在冥冥中自有定数吧。不过当时，我对 Web 开发没有足够的认识，以为还像写 C 程序那样简单，于是在实践中走了不少弯路。

选择脚本语言

我发现用 C 语言编写动态 Web 程序这条路走不通后，在一个偶然的会中，一种脚本语言进入了我的视野。这种脚本语言就是后来很多 Web 开发者和系统管理员喜爱的 Perl 语言。

Perl 语言给了我很大的震撼。我第一次发现还有这样的脚本语言，能够用简洁的表达来实现复杂的工作。同 C 语言相比，Perl 语言简洁得令人难以置信，没有讨厌的指针，只有简单的易于理解的引用，强大的正则表达式让 Perl 语言在文本处理上的表现极其出色，神奇的符号变量和潜规则将代码省略到极致，这些特点使我很快就喜欢上了这门语言。而且，Perl 语言用简单直白的方式提供了 CGI 模块。

在 Perl 语言的帮助下，我终于成功地执行了 CGI 代码，它在本地

服务器和远程服务器上跑得一样好。

经过初步尝试之后，我想用 Perl 语言实现真正实用的 Web 应用。当时互联网上流行一种简单的访问量计数器，原理是记录每次用户请求，统计出访问网站的次数。

计数器是个非常简单的应用，只需要在服务器响应页面请求时更新访问数值，访问数值保存在文件或者数据库字段中。唯一需要注意的是，不管是使用文件还是数据库字段记录数值，在读/写的时候都要考虑加锁，以避免因为读/写冲突而产生脏数据回写，导致数据统计错误。很快，我实现了在服务器上正常运行的计数器，甚至为它增加了 ID 的功能，以便在不同的页面上配置不同 ID 的计数器，让它们互不冲突地统计不同页面的 PV（page view，页面访问次数）。

实现计数器之后，接下来实现的工具是留言簿。在这次尝试中，我开始碰到真正的问题，它不是来自于服务端脚本，而是来自于前端。

在我研究 Perl 语言的时候，冰魂也没闲着，他一个人完成了网站的页面制作——当时我和他合力维护着一个英语学习的个人网站，我用 Perl 语言编写的工具也正好在这个网站上派上用场。

Web 开发与传统的软件开发是有很大不同的，进入 Web 开发领域，意味着你不仅要像传统的程序员一样同数据、应用逻辑打交道，还要同用户端的浏览器打交道，应付各种各样奇怪的问题。HTML、CSS 这些看似简单的技术，其实有其复杂的一面。如果把那些在服务器上处理数据和应用逻辑的部分称为后端领域，那么同浏览器、用户界面和用户行为打交道的工作，就称为前端领域。今天的互联网发展速度如此迅速，各种复杂的用户体验随着技术和产品的成熟都慢慢变成现实，以至于 Web 开发的前端领域变得越来越重要，于是许多公司独立