

【博客藏经阁丛书】

C语言深度解剖

（第3版）

陈正冲 编著
石虎 审阅



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

【博客藏经阁丛书】

C语言深度解剖

（第3版）

陈正冲 编著
石虎 审阅



RFID



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

内 容 简 介

本书由作者结合自身多年嵌入式C语言开发经验和平时讲解C语言的心得体会整理而成,其中有很多作者独特的见解或看法。由于并不是从头到尾讲解C语言的基础知识,所以本书并不适用于C语言零基础的读者,其内容要比一般的C语言图书深得多、细致得多,其中有很多问题是各大公司的面试或笔试题。第3版中新增加了部分C语言知识点的内容。

本书适合广大计算机系学生、初级程序员参考学习,也适合计算机系教师、中高级程序员参考使用。

图书在版编目(CIP)数据

C语言深度解剖 / 陈正冲编著. --3版. --北京:
北京航空航天大学出版社, 2019.6

ISBN 978-7-5124-3015-0

I. ①C… II. ①陈… III. ①C语言—程序设计 IV.
①TP312.8

中国版本图书馆CIP数据核字(2019)第110609号

版权所有,侵权必究。

C语言深度解剖(第3版)

陈正冲 编著

石 虎 审阅

责任编辑 胡晓柏 张 楠

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(邮编100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: emsbook@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1000 1/16 印张:12.75 字数:257千字

2019年9月第3版 2019年9月第1次印刷 印数:4000册

ISBN 978-7-5124-3015-0 定价:39.00元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024



陈正冲

湖南沅江人，毕业于中国科学技术大学。具有丰富的嵌入式软件开发与管理经验，对于软件工程方法论和研发管理有深刻理解。每年都多次举行软件工程技术和研发管理方面培训，讲课深入、透彻、幽默，深受各方好评。目前主要从事研发管理工作。

作者博客

http://blog.csdn.net/dissection_c

石虎

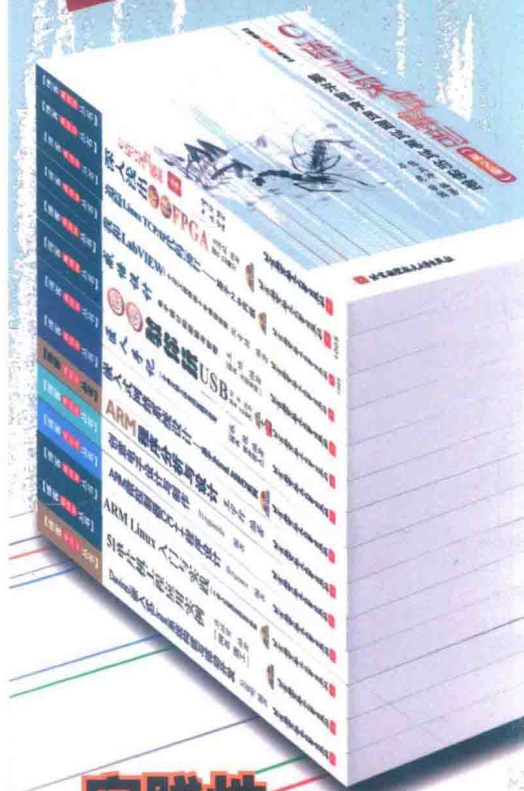
目前为大连交通大学计算机系讲师，主讲C/C++程序设计。具有非常丰富的软件开发和管理经验，参与过多项国家级科研项目。

博客藏经阁丛书

—— 技术博客也能集结成书

藏
经
阁

中国人气最旺的单片机与嵌入式系统博客
拥有忠诚的网络读者和亲密无间的合作伙伴
拥有每天上万次网页展示和上千个IP点击



实践性
应用性
原创性.....

让读者耳目一新，在轻松的交流
过程中获得共鸣

丛书编辑热线：010-82317035
E-mail: hxbpress@buaacm.com.cn

导 读

请先翻到本书的附录 1,按照要求测试一下自己的 C 语言基础。在没有任何提示的情况下,如果能得满分,那你可以扔掉本书了,因为你的水平已经大大超过了作者;如果能得 80 分以上,说明你的 C 语言基础还不错,学习本书可能会比较轻松;如果得分在 50 分以下,也不要气馁,努力学习就行了;如果不小心得了 10 分以下,那就得给自己敲敲警钟了;如果不幸得了 0 分,那实在是不应该,因为毕竟很多题是很简单的。

如果确定要阅读本书,我建议先仔细阅读前言,因为这里回答了你为什么需要本书,为什么 C 语言基础掌握得不太扎实,怎么样才能夯实自己的基础。

而后,建议你按本书的章节排列顺序阅读,因为很多时候前面的内容是后面内容的基础,如果前面内容掌握不扎实,会影响到后面内容的学习效果。

最后,作者认为本书的内容不是看一遍就能完全掌握的,一本好书一定值得多次阅读,而且每读一遍都会有不同的收获。作者本人也相信,你一定会爱上本书的。

知识的增长总是会有一个“把薄书读厚,把厚书读薄”的厚积薄发的过程。作者个人是这么理解这句话的:俗话说,写在纸上的都是垃圾。此话虽然偏激,但也有一定的道理。真正有用的知识很多时候是无法用文字完全表达和承载的,文字能表达的往往只是其中的一部分,甚至是很少的一部分。这就需要我们根据这些有限的文字去理解、挖掘作者所要表达的内容和思想,去揣摩文字的深层含义,并且根据这些基础去举一反三,扩展自己的知识,然后慢慢地把这些文字的营养吸收到自己的知识体系里,这样你所学到的知识就会远远超过原来书本那些文字所表达的内容了,这就是所谓的“把薄书读厚”,即“厚积”。当你读完一本书,并将书本的知识完全吸收消化之后,这些知识就已经完全融入到你自己的知识体系里面去了,并且得到了升华,而后,你需要将融合了自己思想和知识的这些内容的精华慢慢地提炼出来,并慢慢地形成自己独立的思想体系。这个提炼的过程其实就是一个“把厚书读薄”的过程,这个所谓的“厚书”就不仅仅是一本厚的写着文字的书了,而是包含了你自己思想的一部真正的厚书,这个提炼精华的过程也就是所谓的“薄发”,而后真正能影响到你的恰恰就是你自己提炼



出来的这些精华。只要是学习,总需要经历这样的一个过程才能真正学到知识,一个真正懂得学习并且学到知识的人应该能明显地感觉到自己独立的思想体系正在慢慢形成,这才是学习的魅力所在。

最后,以王国维的《人生三境界》与大家共勉。

古今之成大事业、大学问者,必经过三种之境界:“昨夜西风凋碧树。独上高楼,望尽天涯路。”此第一境也。

“衣带渐宽终不悔,为伊消得人憔悴。”此第二境也。

“众里寻他千百度,蓦然回首,那人却在灯火阑珊处。”此第三境也。

此等语皆非大词人不能道。然遽以此意解释诸词,恐为晏欧诸公所不许也。

第3版前言

本书第2版出版到现在已有7年,7年来我收到了很多读者的来信,也大概看了一下网上的一些评论。尤其是京东和当当网上读者购买本书后的评论给了我很多的启示(好评率都在97%以上),在此再次感谢读者们的厚爱。本书初稿其实是在11年前写的,也就是我本科毕业第3年的时候写的。那时候年轻气盛,不知天高地厚,说话写书的语气有点盛气凌人,锋芒毕露,也因此让部分读者觉得不太舒服,在此深表歉意。如果是现在再来写这本书,可能会更加平和一些。不过,“人不轻狂枉少年”,既然轻狂过,那就留下这年轻的痕迹吧,因为我也的确没有时间去重写一遍了。北京航空航天大学出版社的胡主任已经无数次催促我修订第3版,因工作实在繁忙,很难静下心来再增加些内容。不过最近这些年,自己在成长的过程中收获了很多,也有很多的感悟。那我想第3版除了增加部分实际工作中容易出bug的知识点之外,在前言中就谈一些这方面的内容吧,也许能帮读者扫除些许迷雾,让读者未来的路更加清晰一些。这也许比增加一个C语言的知识点更加重要。

不少读者给我来信请教如何学习C语言或者如何成为一名优秀的程序员,也有很多读者不是学计算机相关专业的,但想从事软件开发,却不知道如何学习。其实是否科班出身并不重要,我现在团队里90%以上都不是计算机专业的,但是都干得很不错,尤其是很多数学、物理相关专业的小伙伴比计算机相关专业的小伙伴的技术基础更扎实。逻辑思维是成为优秀程序员的必要的基础之一,这也是为什么一些学数学、物理等专业的也可以成为优秀的程序员的原因。很多人以为掌握一门编程语言便可成为优秀的程序员,所以把精力都放在编程语言本身上,这便是坐井观天了。软件开发不仅仅是学一门语言就够的,编程语言只是工具,是承载软件设计思想的工具。更重要的是软件工程的方法论和其背后的思想,这才是真正需要长时间培养的。对于非科班的程序员朋友,如果想真正提高自身的基本功,建议学一下离散数学、数据结构,这对理解软件设计和逻辑思维训练很有帮助。如果有时间就把数学分析、高等代数等数学系的基本课程都学一遍,这比掌握一门编程语言更重要,这才是超越普通程序员的基础。永远不要舍本求末,否则你就会后劲不足,技术瓶颈很快就会到来。当然,真正理解一门编程语言是成为优秀程序员必不可少的,你需要理解语言的本质,其优

点和缺陷,以及如何避免掉坑。试想一下如果让你来优化 C 语言,你会怎么去优化,或者让你来设计一门编程语言,你会怎么设计。如果悟不透这点,就远远谈不上精通一门语言,当然我也远远算不上精通 C。如果要精通 C,推荐读者读《The New C Standard》,1612 页,网上可找到电子版。这本书没有出版,因为读者群太少,出版社本计划出版的,后来取消了,作者就放到网上供人自由下载。我曾想过翻译这本书,但后来感觉自己水平不够,精力也不够,而且英语对于软件工程师来说就应该是母语,这本书的读者一般都是高手(没一定水平根本看不懂),而高手一般喜欢读原版。所以我就放弃了翻译这本书的想法。不过这本书非常值得一读,可以让你真正理解一门语言。

对于嵌入式开发来说,懂 C,懂一类芯片,懂一个 RTOS 就差不多可以算入门了。我说的懂是真正意义上的懂,比如你把我的书看 10 遍,确实搞懂了每一个问题,理解了我的每一个提问的思维方式,且能举一反三,那就算是基本懂 C 了。以我面人无数的经验来看,中国的嵌入式程序员 90% 以上谈不上懂 C。检验懂不懂的唯一标准就是能不能把不懂的人教懂。我们去理解一个知识点,往往会有这样的体会:第一阶段,感觉这个点比较简单,学了之后感觉懂了;第二阶段,发现原来懂的地方好像又不懂了,感觉一会儿懂一会儿又不懂,自己也不知道自己懂还是不懂;第三阶段,迷糊的点都不迷糊了,能给别人讲解清楚了。这三个阶段就是:看山是山、看水是水;看山不是山、看水不是水;看山还是山、看水还是水。也就是我在第 1 版提到的“把薄书读厚,把厚书读薄”的过程。没有经历这样的一个过程,往往说明没有真正理解一个知识点。不要轻易认为自己读懂了一本经典,反复的悟才是最重要的,经典不是用来读的,是用来悟的。把薄书读厚,把厚书读薄,悟出来并融入你的血液才算是你自己的。我附录所列的参考书目都值得读者认真读上 10 遍,经典的书每读一遍都会有不同的收获。

同时,人的发展要注意三个层次的自我培养:第一层次,知识和技能的培养,比如懂 C、懂 RTOS、懂芯片。第二层次,方法论和思维方式的培养,比如学离散数学、数据结构、数学分析、CMMI、软件设计和架构思想等。第三层次,价值观和职业素养的培养。我的价值观是 5 个字“诚,勤,敬,信,恒”,也是我给我团队传递的价值观。价值观是决定人的思想和行为以及团队氛围的主要因素。一个积极正面的价值观会引导你走向成功。所以,稍微解释一下我的价值观:

诚:“诚者,物之终始,不诚无物”——《中庸》。

对人,我们要做到不欺人,不欺己,不欺心。对事,我们要做到虔诚,诚心实意的做好每一件事,不管有没有人监管都做到一样。这便是最重要,也最难做到的不欺骗自己的本心。

勤:“人生在勤,不索何获”——张衡。“大直若屈,大巧若拙”——《老子》。



勤奋才是成功之根本。任何时候都不要想着随随便便就成功，偶然的成功如同撞树上的兔子，遇到了那是运气，而不能期望永远有兔子撞树上。知识是死的，人是活的，只要勤奋就没有搞不懂的知识。我们大多数人都是做基本工作，不是做爱因斯坦那样开天辟地的创造性工作。所以大多数人拼的是勤奋而不是智商。另一方面，即便是高智商，没有勤奋也将一事无成。

敬：“在貌为恭，在心为敬”——《礼记·曲礼》。

我们要做到敬人，敬己，敬业。尊敬你周围的每一个人，尊敬自己的人格和价值，尊敬所从事的事业，认真做好每一个细节。

信：“信，言合于意也”——《墨子》。

我们要做到信人，信己，立言，立信。要取得别人的信任，首先要信任别人。对自己要有信心，要有永不言输的勇气。行业的圈子往往很小，自己的口碑非常重要。要通过一件件的小事来树立自己的信誉和口碑。

恒：“人而无恒，不可以作巫医”——《论语·子路》。

曾国藩的《家训喻纪泽》里有一句话：“尔之短处，在言语欠钝讷，举止欠端重，看书不能深入，而作文不能峥嵘。若能从此三事上下一番苦工，进之以猛，持之以恒，不过一二年，自尔精进而不觉。”三天打鱼两天晒网是不行的，持之以恒，由量变到质变，才能让你更上一层楼。

还有一点要注意的是，不要把知识等同于能力。知识只有经过实践才能真正转化成能力，多啃硬骨头才能真正提升自己能力。程序员就要有那种迎难而上，永不服输的拼劲。有100分能力就要给自己120分担子，扛过去你的能力就长到120分了。人不扛重担是永远成长不起来的，古之成大事者，无不是在紧要关头扛起重担，成就了万世之功。

最后要重点强调的是，不要投机取巧。很多聪明的人成就反而一般，其根本原因就是太聪明了，做事总想着取巧，舍不得下苦功夫、笨功夫。殊不知，大直若屈，大巧若拙，舍得下苦功夫、笨功夫才是根本的成功之道。比如郭靖，比如曾国藩都是典型的例子。这个世界最可怕的就是比你聪明的人比你更努力，好在他们是你的朋友而非敌人。但我们总归要尽量做到见贤思齐。

陈正冲

2019年6月18日

第 2 版前言

本书的电子版初稿到今天保守估计下载量已超过 20 万次。其中被一位网友转载在百度文库的某单个链接,下载量就达 15 万次,浏览量达 30 多万次,总评分在 4.5 分以上(由于前段时间百度侵权案,百度已删除绝大部分存在于百度文库的本书电子版链接)。至于其他链接或其他网站的下载量从数百到数万不等,对于本书的评价也基本在 4.5 分以上。在大量网友的支持下,在北京航空航天大学出版社嵌入式系统事业部主任胡晓柏先生的大力帮助下,本书于 2010 年 7 月得以正式出版。近日收到胡编辑的 E-mail,告知首次印刷的书已快售罄,马上需要重印,问我是否有勘误或改版计划。经深思熟虑后,决定对本书进行修订。

从本书电子版初稿发布到 2010 年 7 月正式出版,得到了很多网友的帮助,收到了上万封读者来信或留言,帮我修正了很多错误。这些读者中有大学教授,有在读大学生和研究生,有企业工程师和管理人员,也有培训学校讲师,更有一些对编程感兴趣的初中生和高中生。这些读者的指教使我受益匪浅,也使我非常感动,但由于个人学识和能力所限,仍有部分错误或表述不严谨的地方给读者带来了些许困惑。在此,作者表示十分抱歉。于是,在这次的修订中,作者已尽最大努力修正读者所指出的错误,同时为了满足部分在企业工作的工程师的要求,新增加了部分编程规范的内容。即便如此,一些错误和不妥之处仍然在所难免,希望读者朋友不吝赐教。

对于读者的来信绝大部分都已认真回复,少部分来信由于作者本人工作实在太忙,可能没有及时回复,还请读者朋友海涵。同时,从来信中也发现了部分读者过于浮躁的心态。部分读者在并没有真正读懂本书内容的时候,或是没有读完本书的时候,就发 E-mail 或留言问我一些问题,而这些问题其实只要真正读懂了本书就能完全明白的,更何况部分内容在书中已给出了明确答案。也有些读者,缺乏主动思考问题和举一反三的意识,反复询问一些在读懂本书后本应该能理解的问题。遇到问题不是自己先尝试深入研究和解决,而是等待别人给予明确答案,这种心态非常不利于深入学习。如此种种,在作者看来都是由浮躁的心态所引起的。作者在本书中无数次强调,要踏实,多动手,多动脑,一步一个



脚印。希望读者朋友引以为戒,有则改之,无则加勉。

另外,作者有必要再次强调一下:本书绝非读一次就能真正读懂的,对于平时编程经验不多的读者,我保证你读10遍本书,仍然有收获。曾经有位读者来信说到,他把电子版初稿认真读了2遍,自以为基本内容都掌握得挺好了,但是在买了出版后的图书后,做了一下附录1的测试题,只得了30多分,而这些题全部是本书中仔细讲解过的问题。所以作者一直在强调,不要浮躁,要踏实,而作者本人也曾经把参考文献里的绝大部分书读了5遍以上,部分书甚至读了不止10遍。作者本人的水平自然与这些经典书籍的作者有很大的差距,但还是试图融各家之长,加以作者自己的理解和特色的表述方式,把相对枯燥的内容讲得通俗易懂。这也是很多大学老师推荐学生朋友阅读本书的原因之一。当然,我也了解到很多大学老师、培训讲师甚至企业管理人员,已经将本书作为培训教材或参考书目。在此,能为国内的计算机教育有些许贡献,作者深感欣慰。

最后,再次感谢在本书编写过程中所有帮助过作者的家人、领导、同学、网友们,感谢石虎、李婷婷、尹伟红、唐彦邦、周文、熊军、李勇、楚艳秋、陈虎、于婧哲、曾纯、于勇、王继红等为本书所做的工作。

陈正冲
2012年5月

第 1 版前言

我面试过很多人,包括应届本科、硕士和工作多年的程序员,在问到 C 语言相关问题的时候,总是没几个人能完全答上我的问题。甚至一些工作多年,简历上写着“最得意的语言是 C 语言”“对 C 有很深的研究”“精通 C 语言”的人也不完全能答对我的问题,更有甚者我问的问题一个都答不上。

我也给很多程序员和计算机系毕业的学生讲解过《高级 C 语言程序设计》。每期开课前,我总会问学生:你感觉 C 语言学得怎么样? 难吗? 指针明白吗? 数组呢? 内存管理呢?

往往学生回答说:感觉还可以,C 语言不难,指针很明白,数组很简单,内存管理也不难。

一般我会再问一个问题:通过这个班的学习,你想达到什么程度?

很多学生回答:精通 C 语言。

我告诉他们:我很无奈,也很无语,因为我完全在和一群业余者或者是 C 语言爱好者在对话。你们浪费了大学学习计算机的时间,念了几年大学,连 C 语言的门都没摸着。

现在大多数学校计算机系都开了 C、C++、Java、C# 等语言,好像什么都学了,但是什么都不会,更可悲的是有些大学居然取消了 C 语言课程,认为其过时了。我个人的观点是“十鸟在林,不如一鸟在手”,真正把 C 语言整明白了再学别的语言也很简单,如果 C 语言都没整明白,别的语言学得再好也是花架子,因为你并不了解底层是怎么回事。当然我也从来不认为一个没学过汇编的人能真正掌握 C 语言的真谛。

我个人一直认为,普通人用 C 语言在 3 年之下,一般来说,还没掌握 C 语言;5 年之下,一般来说还没熟悉 C 语言;10 年之下,谈不上精通。所以,我告诉我的学生:听完我的课,远达不到精通的目标,熟悉也达不到,掌握也达不到。

那能达到什么目标? ——领你们进入 C 语言的大门。入门之后的造化如何在于你们自己。不过我可以告诉你们一条不是捷径的捷径:把一个键盘的



F10 或 F11 按坏(或别的单步调试快捷键),当然不能是垃圾键盘。

往往讲到这里,学生眼里总是透露着疑虑:C语言有这么难吗?

我的回答是:学起来不难,但要真正用明白很难。

学生说:以前大学老师讲C语言,我学得很好。老师讲的都能听懂,考试也很好。平时练习感觉自己还不错,工作也很轻松找到了。

我告诉学生:听明白、看明白不代表你懂了,你懂了不代表你会用了,你会用了不代表你能用明白,你能用明白不代表你真正懂了!什么时候表明你真正懂了呢?你站在我这来,把问题给下面的同学讲明白,学生都听明白了,说明你真正懂了;否则,你就没真正懂,这是检验懂与没懂的唯一标准。

冰山大家都没见过,但总听过或是电影里看过吧?如果你连《泰坦尼克号》都没看过,那你也算个人物(开个玩笑)。《泰坦尼克号》里的冰山给泰坦尼克造成了巨大的损失。你们都是理工科的,应该明白冰山在水面上的部分只是整个冰山的1/10。我现在就告诉你们,C语言就是这座冰山。你们现在仅仅是摸到了水面上的部分,甚至根本不知道水面下的部分。我希望通过我的讲解,让你们摸到水面下的部分,让你们知道C语言到底是什么样子。

从现在开始,除非在特殊情况下,不允许用 printf 这个函数。为什么呢?很多学生写完代码,直接用 printf 打印出来,发现结果不对,然后就举手问我:老师,我的结果为什么不对啊?连调试的意识都没有!大多数学生根本就不会调试,不会看变量的值、内存的值;只知道 printf 出来结果不对,却不知道为什么不,怎么解决。这种情况还算好的。往往很多时候 printf 出来的结果是对的,然后呢,学生也理所当然地认为程序没有问题。是这样吗?往往不是,书中会有相应的举例进行说明。请读者牢记一点:结果对,并不代表程序真正没有问题。所以,以后尽量不要用 printf 函数,而要去查看变量的值、内存的值。另外,在我们目前的编译器里,变量的值、内存的值对了就代表你的程序没问题吗?也不是,对于这一点,书中也会有相应的举例进行说明。

这个时候呢,学生往往会莫名其妙:大学里我们老师都教我们怎么用 printf,告诉我们要经常用 printf,怎么这位老师的说法与大学中所学的的截然相反呢?我个人认为,这也恰恰是大学教育中存在的不足之处。很多大学老师缺乏真正使用C语言编程的实战经验,不调试代码,不按 F10 或 F11(或别的单步调试快捷键),水平永远也无法提上来。所以,要想学好一门编程语言,最好的办法就是多调试。有兴趣的读者可以去一个软件公司转转,如果发现键盘上的 F10 或 F11 铮亮铮亮,那么毫无疑问此机的主人曾经或现在是开发人员(这里仅指写代码的,不上升到架构设计类的开发人员),否则,必是非开发人员。

非常有必要申明的是,本人并非什么学者或是专家,但本人是数学系毕业,所以对理论方面比较擅长。讲解的时候会举很多例子来尽量使学生明白这个知识点,至于这些例子是否恰当则是见仁见智的问题了。但是一条,长期的数学训练使得本人思维比较严谨,讲解一些知识点尤其是一些概念性、原理性的东西时会抠得很细、很严,这一点相信读者会体会得到的。本书是我平时讲解C语言的一些心得和经验,其中有很多我个人的见解或看法,经过多期培训班的实践,发现这样的讲解比较透彻,深受学员欢迎,也有业余班的学生甚至辞掉本职工作来听我的课。

当然,本书中关于C语言这么多经验和心得的积累并非我一人之力。借用一句名言:我只不过是站在巨人的肩膀上而已。给学生做培训的时候我试图融各家之长,加上我个人的见解传授给学生。我参考比较多的书有:Kernighan和Ritchie的《The C Programming Language》;Linden的《Expert C Programming》;Andrew和Koenig《C Traps and Pitfalls》; Steve Maguire的《Write Clean Code》; Steve McConnell的《Code Complete. Second Edition》;林锐的《高质量程序设计指南——C++/C语言》。这些书都是经典之作,但却都有着各自的不足之处。读者往往需要同时阅读这些书才能深刻地掌握某一知识点。这些书饱含着作者的智慧,每读一遍都有不同的收获,我希望读者能读上十遍。另外,在编写本书时也参考了网上一些无名高手的文章,这些高手的文章见解深刻,使我受益匪浅,在此深表感谢,如果不是他们,恐怕我的C语言水平也仅仅是入门而已。

学习C语言,上面提到的这几本书如果读者真正啃透了,水平不会差到哪儿。与其说本书是我授课的经验与心得,不如说本书是我对这些大师们智慧的解读。本书并不是从头到尾讲解C语言的基础知识,所以,本书并不适用于C语言零基础的人。本书的知识要比一般的C语言书所讲的深得多,其中有很多问题是各大公司的面试或笔试题,所以本书的读者应该是中国广大的计算机系的学生和初级程序员。如果对于书中的内容读者能真正明白80%,作为一个应届毕业生,恐怕没有一家大公司会拒绝你。当然,书内很多知识也值得计算机系教师或是中高级程序员参考,尤其书内的一些例子或比方,如果能被广大教师用于课堂,我想对学生来说是件非常好的事情。

有人说电影是一门遗憾的艺术,因为在编辑完成之后总能或多或少地发现一些本来可以做得更好的缺陷。讲课同样也如此,每次讲完课之后总能发现自己某些地方或是没有讲到,或是没能讲透彻,或是忘了举一个轻浅的例子等。整理本书的过程也是如此,为了尽量精炼,总是犹豫一些东西的去留。

在编写本书的过程中得到了很多领导、同学、网友的帮助,感谢他们抽出宝贵的时间审阅本书初稿并提出各种宝贵的修改意见,是他们使本书变得更加完善;感谢我的家人,没有他们的鼓励与支持,没有他们对我生活上的细心照料,我

也抽不出这么多时间来完成这本书。他们是:石虎、李婷婷、尹伟红、唐彦邦、周文、熊军、李勇、楚艳秋、陈虎、于婧哲、曾纯、于勇、王继红。最后要感谢北航出版社的各位工作人员,特别是嵌入式系统事业部主任胡晓柏先生,没有他的帮助,本书不能这么快地得以出版和发行。

限于作者水平,对于书中存在的遗漏甚至错误,希望各位读者能予以指教。有兴趣的朋友,可发送电子邮件到:dissection_c@163.com,与作者进一步交流;同时作者专门为本书开了个博客,以方便和读者交流,博客地址是:http://blog.csdn.net/dissection_c。

陈正冲
2010年3月

目 录

第 1 章 关键字	1
1.1 最宽宏大量的关键字——auto	3
1.2 最快的关键字——register	3
1.2.1 皇帝身边的小太监——寄存器	3
1.2.2 使用 register 修饰符的注意点	4
1.3 最名不符实的关键字——static	4
1.3.1 修饰变量	4
1.3.2 修饰函数	5
1.4 基本数据类型——short、int、long、char、float、double	5
1.4.1 数据类型与“模子”	6
1.4.2 变量的命名规则	6
1.5 最冤枉的关键字——sizeof	12
1.5.1 常年被人误认为函数	12
1.5.2 sizeof(int) * p 表示什么意思	13
1.6 signed、unsigned 关键字	13
1.7 if、else 组合	17
1.7.1 bool 变量与“零值”进行比较	17
1.7.2 float 变量与“零值”进行比较	18
1.7.3 指针变量与“零值”进行比较	19
1.7.4 else 到底与哪个 if 配对呢	20
1.7.5 if 语句后面的分号	21
1.7.6 使用 if 语句的其他注意事项	22
1.8 switch、case 组合	23
1.8.1 不要拿青龙偃月刀去削苹果	23