

<<品悟C>>

图书基本信息

书名：<<品悟C>>

13位ISBN编号：9787302296263

10位ISBN编号：730229626X

出版时间：2012-10

出版时间：清华大学出版社

作者：薛非

页数：502

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

编者序——敢于向权威挑战经常听到我的作者朋友说类似这样的话：“看我的代码，多帅气！”

何谓帅气、漂亮的代码？

在我看来，先不谈代码丑俊，首先代码必须没有残疾。

在健康的代码基础上，才能谈网络间流传的各种称号：漂亮代码、优质代码、帅气代码、性感代码...这就是本书要讲的内容——摒弃一些错误的观念，重新认识C语言的严谨与华丽。

计算机语言是人类操控计算机硬件的主要手段，是语言创造者智慧的浓缩。

计算机语言在中国的普及，是伴随着中国经济腾飞的这几十年。

无数中国的程序员伴随着很多经典教材顺利入门并成长了起来，随着互联网的发展，更多的朋友在网上发表自己的看法，所以互联网也成为了大家获取编程技术的重要渠道。

但是，金无足赤，随着C语言标准的不断完善，C开发理念的不断进步，市面上的一些教材和网上一些说法确实存在一些值得商榷的问题，或者谬误，因为这些资料传播甚广，有些反而成为了“标准”，这种现象有时候可能会误导初学者，让初学者无法从一开始就建立相对准确的开发观念。

本书就是这样一本敢于向“权威”挑战的书，通过作者多年的积累，以及众多热心网友的鼎力支持，对一些C语言中经常出现的问题进行讨论。

在中国的教育体制下，不可否认，很多创造力被限制甚至扼杀，很多路被强制标注为正确途径，很多人的学习兴趣在一开始就被磨灭.....

但是很高兴能出现这么一本书，在阅读过程中，可以体会一众先锋程序员对开发观念的重视，对细节的精益求精，对逻辑的反复推敲，甚至上升到——对真理的求索。

我通常拿到一本书，会先浏览一下，把我认为重要的文字用特殊形式标出来，比如“提示”、“注意”、“TIPS”之类的样式以及文字加粗、代码加黑之类的重点标注，但是本书我在做出努力之后，发现无法进行这样的处理，因为似乎每句话都是重点，每段代码都是精心挑选的，无论是正确代码还是错误代码。

最后一点不得不提一下，作者的文笔非常犀利，对一些问题的批判毫不留情，可能是作者本人以及相关参与编写的网友受到过太多问题资料的影响，以至于走了不少弯路，所以面对问题资料，个别时候会流露出来一种愤青一般的愤怒。

这可以理解，而且我在编辑加工过程中也尽量保持了文风，但是还是希望作者和所有读者做一名淡定的程序员。

如果本书冒犯了某些朋友，请不要大动肝火，可以发邮件与作者讨论，毕竟从一个角度看问题，存在片面的可能性。

大家讨论技术，以代码会友。

权威之所以是权威，必然有其可取之处，对于权威，我们除了要有挑战的勇气，更多的是要理性的尊重，毕竟“权威”带领太多人进入C语言门槛。

没有BUG的代码是不存在，同样没有问题的资料也是不存在的，希望大家在学习过程中，多思考，多交流，早日树立端正的开发思路。

在此感谢包括作者在内的热心人，有监督才会有活力，有挑战才会有发展！

内容概要

这是迄今为止国内仅有的一部全面且集中地讨论C程序设计中常见错误的书籍。其主旨在于帮助读者从错误中更准确地理解C语言，并在实践中更好地运用C语言。

全书分为上下两部：上部“形而下学”讨论C代码中常见的错误和编程恶习；下部“形而上学”讨论的是对C语言本身认识方面的常见误区。这些问题特别具有普遍性，很多错误不仅仅存在于初学者之中，即使是职业程序员多半也会在本书中找到自己的错误。因此，对于任何想纠正自己对C语言的错误认识以及想不断提高C语言水平的读者来说，本书都提供了一种独到且重要的视角，是一本不容错过的、具有重要参考价值的技术资料。

本书并非那种长篇大论环环相扣的书籍，它的各个主题独立成篇，使人阅读起来轻松愉快。读这本书并不需要正襟危坐在书房里，它可以随时随地地拿过来翻阅。

本书可作为各高校C语言或相关课程的教学参考书，也适合那些具有一定经验的C语言程序员阅读使用。

作者简介

薛非

貌不出众，所以照片还是免了吧。
编辑非要也不给，有码无码都不行。

没有什么丰功伟绩，所以也不敢自吹自擂。
不过我坚信，书籍本身是作者最好的名片，作者的形象决定于读者的阅读感受。

所以，这里的空白还是留给读者的想象吧。

书籍目录

目录

上部 形而下学——代码错误及陋习

第1章 低级错误

问题1 C啊，多少C++假汝之名而行——C、C++不分

问题2 环节缺失——关于C程序开发过程

问题3 错误的“标点”——混用中英文字符

问题4 错误的“单词”

问题5 “合并单词”产生的问题

问题6 形形色色的“分号”误用

问题7 “一仆二主”——两个main()函数

问题8 忘记“&”

问题9 who ' s who

问题10 同形异质——为符号常量赋值

问题11 词不达意

问题12 使用switch语句的常见错误

章节摘录

版权页：这样，这两个副效应产生的时间如果因为不同编译器的不同安排，就会产生无论是整个表达式的值，还是变量“a”最终的值，在不同的编译方式下产生各不相同的互相矛盾的结果。

然而每一种编译“安排”，只要是在“；”之前完成了副效应，却都没有违背C语言的要求。

这也就是说，在不违背C语言原则的情况下，表达式“ $a += (a -= (a * a))$ ”可以有多种解释方式，得到并不唯一的结果。

这叫做“二义性”或“多义性”。

程序设计语言和程序是不可以有二义性的，这一点和我们平时使用的自然语言截然不同。

为此，C语言特别规定：在两个相邻的序点之间，同一个数据对象中保存的值最多只可以通过表达式求值改变一次。

表达式“ $a += (a -= (a * a))$ ”的内部并没有序点，因此必定在其前面和后面拥有这“两个相邻的序点”。

然而在“ $a += (a -= (a * a))$ ”中，“ $+=$ ”、“ $-=$ ”运算的副效应又都是改变同一个数据对象“a”的值，这明显违反了“同一个数据对象中保存的值最多只可以通过表达式求值改变一次”这个C语言对表达式的基本要求，因此这个表达式是一个错误的表达式。

但是这种错误不同于语法错误（违背C语言硬性限制（constraint）的错误），编译器会承认这样的表达式不违背C语言的限制，因而可以进行编译，并且可能不会把这个错误作为语法错误报告给编程者。

最多，好的编译器可能会给出一个“警告”——它嗅出了这里疑似有错。

媒体关注与评论

从“错误”中学到的东西，往往比从“正确”中学到的多。
——源自ChinaUnix论坛著名技术长贴《以其昏昏使人昭昭》、ChinaUnix论坛和博客园技术博客《C博客》系列

编辑推荐

《品悟C:抛弃C程序设计中的谬误与恶习》可作为各高校C语言或相关课程的教学参考书，也适合那些具有一定经验的C语言程序员阅读使用。

名人推荐

把良好的编程风格教给那些之前曾经接触过BASIC的学生几乎是不可能的。
作为可能的程序员，他们已精神残废，无重塑的可能了。

——E. W. Dijkstra没有一种编程语言能阻止程序员写出糟糕的程序来，不管这种语言结构有多良好。

——Larry Flon计算机比人类历史上的任何发明都更快速地导致你犯更多的错误——可能除了手枪和龙舌兰酒。

——Mitch Radcliffe

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>