

《C陷阱与缺陷》

图书基本信息

书名：《C陷阱与缺陷》

13位ISBN编号：9787115106230

10位ISBN编号：7115106231

出版时间：2003-12-1

出版社：人民邮电出版社

作者：Andrew Koenig

页数：192

译者：高巍

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《C陷阱与缺陷》

内容概要

作者以自己1985年在Bell实验室时发表的一篇论文为基础，结合自己的工作经验扩展成为这本对C程序员具有珍贵价值的经典著作。写作本书的出发点不是要批判C语言，而是要帮助C程序员绕过编程过程中的陷阱和障碍。
作者以自己1985年在Bell实验室时发表的一篇论文为基础，结合自己的工作经验扩展成为这本对C程序员具有珍贵价值的经典著作。写作本书的出发点不是要批判C语言，而是要帮助C程序员绕过编程过程

书籍目录

第零章 导读

第一章 词法“陷阱”

第二章 语法“陷阱”

第三章 语义“陷阱”

第四章 连接

第五章 库函数

第六章 预处理器

第七章 可移植性缺陷

第八章 建议与答案

《C陷阱与缺陷》

精彩短评

1、 这本书中介绍的绝大部分内容都偏旧了，或者说对于我这样的编码1年以上的人来说，有点过时了，跟准确的说，书中介绍的东西，我基本上都知道了。不过该书很适合刚学习C语言的新手看看，对于新手了解使用C语言过程中可能存在的“陷阱”很有帮助。

一本书总是或多或少地带给自己以前不知道的一些东东，详细的这些，我都写在了blog中，地址是<http://www.cnblogs.com/cnpirate/>，呵呵

2、 老外写的书真是很牛逼，看着让人欲罢不能。

3、 家里躺着呢，呵呵

4、 大学时最推崇的C语言3本神书之一

5、 对代码来说大多数情况下可读性还是优先于效率的。写出很trick的代码要么是NB，要么是ZB，要么就是NC...

6、 学C语言必读。

7、 读个，封面不一样啊

8、 C语言经典书籍

9、 经验和思考均有的书

世俗的说也是面试的好书

10、 老师上课很无聊，所以就利用上课时间把这本书看完了。推荐学完C基本内容的同学看看这本书，收获会很大，也会避开很多编程是的bug。

11、

12、 这本书的主题是C中的各类细节，平时往往不会注意。

1. 注释的嵌套性，怎么判断编译器是否允许嵌套注释。

2. 运算符优先级，结合顺序，编译器贪婪

3. assert的实现方法，避免if

4. 少敲一个分号导致各种血案

5. 注意函数、if等的作用域。

6. 注意字符集，防止移植性问题

7. 变长参数，stdarg

13、 说一般应该是我能给出的最高褒奖

14、 多看几遍

15、 这书要是影印就好了

16、 非常好的书，读来有拨云见月的感觉，基本解决了我目前对指针的概念与用法的困惑。

17、 书中的内容感觉有一部分还是比较不错的，我觉得我如果碰到了那些问题，肯定栽了。但是这个书中的有些内容确实该去掉的，因为C标准都出了很多年了。

但是感觉书的内容并不怎么配得上各种超高评价。

或许是我的C语言水平比较差吧。

不过这本书还是值得多读几遍的。

18、 当年刚毕业时看过，很有收获。

19、 那还是觉得适合这个：C语言程序设计现代方法

20、 感觉还不错，就是稍微短了一点儿

21、 大概十年前读过这本书，当时还在学校读研究生。内容不深也不浅，难度刚刚好。印象最深的是那个把常数0强制转换为一个指向入口地址为0的函数的指针，我特地用Keil C编了一段程序在80C51单片机上验证，果然可以，佩服至极。

22、 大师风范

23、 这本适合入门的看么？

24、 旧了一点，但很经典，收获不少

《C陷阱与缺陷》

25、 这是一本需要放在案头时时翻阅的书，甚至如果不显得过分的话，每次写C代码之前都应该翻阅一下目录，回忆一下大致的内容。

这本书从词法、语法、语义、链接（中文版译为“连接”，但据我所读过的相关书籍，全部译的是“链接”）、库、预处理器、可移植性等层次和方面介绍了C语言编程中可能遇到的问题。

其中涉及到一些ANSI C之前的内容，这对于一些读者（包括我）可能来说是比较陌生的，如果有志于维护使用K&R C编写的古董程序或者想写出几乎所有C编译器都能搞定的代码的，可以细细琢磨琢磨，其他人亦可以借此训练训练思维，或者说测试测试自己的功底。

这本书里还有一些部分非常底层，会涉及到大端法、小端法的问题，也会涉及到数据的位模式的问题。这个是需要一定的基础的，我这方面的知识应该是来源于CSAPP（不好意思，不算植入广告吧）的。

对于其中记录的缺陷和陷阱，切不可走马观花。我觉得读书、尤其是这类总结了前人犯过的错误和经验的书籍，必须有三问：一问原来为何出错，二问解决办法为何正确，三问如何避免出错。一般的书籍都解释了原来为何出错，那么读者就应该想想如何才能发现这类错误。

关于思考如何避免出错，我举两个例子：

一个是此书2.6节讲到了“悬挂”else引发的种种问题，如果大家写的代码足够多并且以多写一对花括号为耻的话，肯定已经遇到了这个问题。如果其中涉及到宏，则问题隐藏得会更深。那么如何避免出错呢？首先，前面提到宏会使问题得到隐藏，那么“小心、谨慎”这种说法是绝对行不通的，甚至使用一些自动indent的工具也不能发现问题。那么我们就应该想其他的办法。而我能想到的就是所有的if从句和else从句都用花括号包围起来。这当然得先树立正确的荣辱观，即不以多写了“没必要”的花括号为耻。

二是练习1-2中隐含的一个问题：既然编译器不支持嵌套注释，那如果我在注释代码的时候，这段代码中有注释呢？难道编译器报错之后我就把代码段中的注释去掉？那我如果只是临时注释掉以供调试或其他用途，不久还会使用呢？根据《代码大全》第二版（这个也不是广告），我们可以利用预处理器来解决这个问题：

```
#if 0
    /* statements to comment out */
#endif
```

以上便是我的一点读书感受和方法论。

26、 成名作啊，的确不错的书

不过由于之前Expert C Programming看的太认真，看C Traps and Piffalls会快很多

CTP这本书算是作者自己在编程过程中的一些notes吧，其实也就是偶们自己写C的时候很容易犯的一些错误~

27、 c开发者必读，少犯已知的错误。

28、 我的c语言入门书

29、 小而精悍

30、 现阶段木有C专家那本书作用大诶~

31、 有一定的基础，开始做编程的应该挺适合阅读这本。

32、 尺子？干嘛用的？

33、 c专家编程里有讲...

《C陷阱与缺陷》

34、C语言的坑是那么有意思！

35、 前天晚上才拿到这本书，很薄的一本，但是人邮的书就是贵

昨天上午看了两章，晚上看了第三章，今天上午把剩下的看完

唯一的感受就是很爽，这本书写的东西大多数都知道，但也有第一次看到的，比如 `i[arr]` 等价于 `arr[i]`，从指针的角度来解释一目了然，但是过去自己从未想到过这个问题，也没有试验过

写程序最重要的是思考和总结，而不是写的多少，写的再多只不过是垃圾代码

本书最后一章介绍的可移植性，妙哉！考虑边界情况，无符号整数与有符号整数的区别，从C的角度思考问题，就值得研读的就是 Computer Systems

曾读过 Andrew Koenig 的 Accelerated C++、C++ 沉思录，可能是由于翻译的缘故，唯这本书读的最为顺畅

36、他那个年代，C里面还没有const

37、读完了,看来得读第二遍!

38、这个排版确实很恶心人。。。

39、绝对让你有一种似曾相识的感觉

40、必读，写得比肩简练，但都切中要害

41、很不错的书，必备的参考书

42、非常经典的一本书，让我获益匪浅，深入理解c的一些不容易理解的地方。看过两遍，面试前必看

43、 书中讲的全是使用C语言中容易忽视的问题，可是我记得书中好象没有谈及 `char **p` 和 `const char **p` 不兼容，但 `char *p` 和 `const char *p` 却是兼容的问题..... 微失望.....

44、感觉这本书不在看多块，得多翻翻。最有意思的应该是些小细节

45、2011年再读一遍。

46、好像有些收获，忘了。。。

47、#define MACRO monster

48、大一读的，感觉不记得了.....重读~

49、书很棒，可以一口气读完

50、如果什么都不懂就不适合了，因为这本书讲的是最佳实践

51、 刚看完这本书，此书所称归纳了C语言中最晦暗不明的部分，回忆起来几乎都是我之前的编程中全部都出现过，如复杂声明、野指针、边界差一错误、链接器问题、宏定义、移植性问题等。但这些问题在一般的C语言指导书籍上都是很少提到和解释的。当时只得在网上去寻求答案。如果之前看了这本书，相信可以节约一些时间，对于问题的根源也会了解得更加系统。

这本的历史其实蛮久的，部分篇章在ANSI C之前就存在了，因此对于一些ANSI C之前的细节也进行了探讨。经常从底层和编译器的角度进行了问题的阐述，让我了解这些问题为何会出现，以及出现的缘由。

作者的另外一本书《Accelerated C++》也很好，抓紧看完再写书评.....

52、Koenig的书都很不错，喜欢他的风格.特别人家还是夫妻档啊

53、嗯，我貌似做了蛮多笔记，包括orz酱紫的符号- -||。但是，亮点是，这本书是我从图书馆借的...

...

54、呵呵,是啊.

那两个牛牛,又是谁能与其相提并论的呢:)

55、 这本书是我的第二本C语言书，大学学的谭老的书不算。

也许是《C语言程序设计现代方法》一书过于详尽的原因，而且毕竟也有着一些编程基础，在读这本书的时候我读的很快。

《C陷阱与缺陷》

这不是讲C语言基础的书，而是从一些C语言的特性入手，讲C语言编程时容易出现的错误，以及一些最佳实践。

虽然对我的帮助不是特别大，但是我还是推荐初学者把这本书作为学习C语言的第二本书！

56、划线 <<

57、很多内容都太老了，在现代计算机不实用。

58、花了两天时间翻完一遍。

对一个栗子印象深刻：

“0123456789”[n]居然也是一个合法的数组形式。--!!!

printf("%c"，“0123456789”[0])将打印字符‘0’

依次类推，printf("%c"，“0123456789”[1])将打印字符‘1’，printf("%c"，“0123456789”[2])将打印字符‘2’.....

“0123456789”[2])将打印字符‘2’.....

用于解决某些机器的字符集中数字不是顺序排列的问题。

(《K&R》也有提到某些架构的机器上不宜用c+‘0’这样方法来求c的数字表示，具体哪些变态的机器会采取这种策略就不得而知了)

因为一个字符串常量可以用来表示一个字符数组，所以在数组名出现的地方都可以用字符串常量来替换。

59、学c语言必须要看的,虽然里面的有些东西有些过时,又看了一遍

60、太注重技巧了

61、结尾的访谈很有价值

62、昏..K&R的第二版是在ANSI C之后出版的啦..

63、涵盖了一些常见的错误陷阱

64、不要等到笔试面试才想起来这本书，这些都是coding过程中慢慢吃透的，唯手熟耳。

65、这个可是Andrew Koenig同志扬名已久的一本书,可真的是扬名太久了,这本书居然是出版在ANSI C制订以前,真是汗阿.以至于书中居然还在考虑如何兼容K&R风格的函数定义形式.

这本书倒真的是一些很琐碎的事情,比如说dangling else,switch中漏掉的break.

收获还是一些.

最明显的是,我发现其实运算符的优先级也不是那么麻烦,花点心思记下来其实也挺好的:)

66、编程就好比练功，如果学习.net,mfc,vb等具体的语言和工具是外功(招式)，对基础的学习就是内功，只注重招式而内功不扎实是不可能成为高手的。很多人会认为《射雕英雄传》中马玉道长什么都没有教郭靖，马道长教的表面看来是马步冲权实则都是内功心法，郭靖拜师洪七之后开始练习降龙十八掌凭借的就是这深厚的内功，吞食蝮蛇宝血又加上练习了周博通传授的九阴真经和外加功夫双手互博技之后，终于练就行走江湖的武功，由此可见马玉道长传授给了郭靖的是最基础的，也是最重要的观念，编程也好比盖高楼，根基没打好早晚有一天会垮掉的，而且盖得越高，损失也越惨重。这些底层知识和课本不是没有用也不是高深的不能学，而是我们必须掌握的基础。

阅读经典的书籍就是帮助我们打好基础的最好方法，毫无疑问本书就是这样的经典书籍。本书分章讨论社区：<http://bbs.theithome.com/>，来享受经典的魅力吧！！

67、呵呵，K&R两个版本都是在ANSI C正式出来之前出版的。

68、一个下午刚刚看完老罗的《我的奋斗》坐在电脑前和女朋友说着要看书，否则会被她嘲笑书架上那2/3的书还没看完，又买了好多。

于是8点半左右的时候，站在书架前挑选了这本比较薄的先来欺负。

虽然之前电子版也已经看了10来页，但是这次还是从序言开始看起，还对序言里面不懂的单词对照着后面的翻译去猜测。

认认真真的从第一章看到第七章，从企图到网上寻找习题答案到突然想起来找找书中是否已经有

《C陷阱与缺陷》

答案，从20：30看到00：30，真的可以说是一气呵成。

从大学毕业以后很多年都没有在捡起C相关的东西，买这本书的动机，多少是想了解如何预防一些普遍存在的陷阱。不过这么一遍阅读下来，目的性的收获不多，毕竟，好多都是c特有的陷阱，当然对c++也有些醒世的作用，对于其他就不适用了。

标榜着非初级的读物，其实还是相当浅显易懂的，当然前提是当年那些概念就已经非常清楚。浅显不代表没有拍案叫绝的冲动。那些陷阱被揭穿的时候一样会恍然大悟。只是庆幸的，如果自己做，已经从所谓coding style上面杜绝了大多数陷阱的发生。

另外一个巨大的收获却不在陷阱本身，而是c语言的实现。一方面是第一次意识到以前调用的“函数”其实却是宏定义，另外一方面，几乎每一章在阅读时都以编译原理的角度来思考问题的根源，会考虑如果自己来设计这种语言怎么办——其实这也是不少习题试图引导读者思考的方向。

69、《C陷阱与缺陷》，《C专家编程》，《你必须知道的495个C语言问题》，对我而言C语法三剑客。

70、我在图书馆看书时候都带一把尺子orz...

71、c,C++程序员必备

72、作者太牛了，薄薄的一本书，全是干货！

73、书中所写的陷阱。。。程序写多的都会遇到过。。在这本书里倒是没学到多少。。。而且书太老了。。

74、深入C

75、甩课本不知道几条街，指针讲得很细。

76、作者选材很好，文字也很简洁。不过可惜书的内容有老。里面的C不知道是哪个版本跟目前学的的TC3.0或者VC6.0不太一样

77、很好的书，不是初级读物，但也不过于艰深，而且厚度刚刚好，让人在热情高涨阶段读完，不致于到了一半时间就气馁。

但令人感到遗憾的是，排版是太差了！代码缩进极其混乱，许多代码看上去参差不齐，该对齐的地方没对齐，对齐的长短不一，不该换行的地方换行，这种情况比比皆是，版能排成这个样子，显得极其草率。

编者为什么不去随便找一本外版计算机书看一看（这本书本来就是翻译的），随便哪个出版社，O'REILLY, PRENTICE HALL, Addison Wesley, 看看人家的代码是怎样排的，只需要看一看，比较一下，马上就能发现问题，为什么不肯这么做呢？

78、书不错，内容有点老

79、woo,你在都江堰还好?

80、好书，也提出了好多没注意过的陷阱。但是总觉得有很多内容有些过时。

81、太过古老，很多内容都不适用于现代编译器了

82、有几个小知识点以前没注意过，当再学习了。

精彩书评

- 1、书中讲的全是使用C语言中容易忽视的问题，可是我记得书中好象没有谈及 `char **p` 和 `const char **p` 不兼容，但 `char *p` 和 `const char *p` 却是兼容的问题..... 微失望.....
- 2、刚看完这本书，此书所称归纳了C语言中最晦暗不明的部分，回忆起来几乎都是我之前的编程中全部都出现过，如复杂声明、野指针、边界差一错误、链接器问题、宏定义、移植性问题等。但这些问题在一般的C语言指导书籍上都是很少提到和解释的。当时只得在网上去寻求答案。如果之前看了这本书，相信可以节约一些时间，对于问题的根源也会了解得更加系统。这本的历史其实蛮久的，部分篇章在ANSI C之前就存在了，因此对于一些ANSI C之前的细节也进行了探讨。经常从底层和编译器的角度进行了问题的阐述，让我了解这些问题为何会出现，以及出现的缘由。作者的另外一本书《Accelerated C++》也很好，抓紧看完再写书评.....
- 3、这个可是Andrew Koenig同志扬名已久的一本书.可真的是扬名太久了,这本书居然是出版在ANSI C制订以前,真是汗阿.以至于书中居然还在考虑如何兼容K&R风格的函数定义形式.这本书倒真的是一些很琐碎的事情,比如说dangling else,switch中漏掉的break.收获还是一些.最明显的是,我发现其实运算符的优先级也不是那么麻烦,花点心思记下来其实也挺好的:)
- 4、一个下午刚刚看完老罗的《我的奋斗》坐在电脑前和女朋友说着要看书，否则会被她嘲笑书架上那2/3的书还没看完，又买了好多。于是8点半左右的时候，站在书架前挑选了这本比较薄的先来欺负。虽然之前电子版也已经看了10来页，但是这次还是从序言开始看起，还对序言里面不懂的单词对照着后面的翻译去猜测。认认真真的从第一章看到第七章，从企图到网上寻找习题答案到突然想起来找找书中是否已经有答案，从20:30看到00:30，真的可以说是一气呵成。从大学毕业以后很多年都没有在捡起C相关的东西，买这本书的动机，多少是想了解如何预防一些普遍存在的陷阱。不过这么一遍阅读下来，目的性的收获不多，毕竟，好多都是c特有的陷阱，当然对c++也有些醒世的作用，对于其他就不适用了。标榜着非初级的读物，其实还是相当浅显易懂的，当然前提是当年那些概念就已经非常清楚。浅显不代表没有拍案叫绝的冲动。那些陷阱被揭穿的时候一样会恍然大悟。只是庆幸的，如果自己做，已经从所谓coding style上面杜绝了大多数陷阱的发生。另外一个巨大的收获却不在陷阱本身，而是c语言的实现。一方面是第一次意识到以前调用的“函数”其实却是宏定义，另外一方面，几乎每一章在阅读时都以编译原理的角度来思考问题的根源，会考虑如果自己来设计这种语言怎么办——其实这也是不少习题试图引导读者思考的方向。

章节试读

1、《C陷阱与缺陷》的笔记-全章节

这本书不错的，在论坛上写听别人说，后来在毕业时看了看，在一次面试时用到了，做完笔试，面试后，面试官又给了我一张纸条，他说你把这试卷做一下，我倒，以为什么高难度的题了。一看都是些陷阱，考得更为基础，基中有60%就是这书上的知识点，所以哥就进了！

2、《C陷阱与缺陷》的笔记-第11页

0开头的数字被认为是八进制，这个知识点在写javascript时被坑过，c语言中倒是不常见。可能跟网页出现0开头的数字几率大有关。

c语言中单引号被认为是字符，双引号代表的字符串则是一个指向无名数组起始字符的指针。python倒是木有这种差别，源码中介绍过python string的实现，如果是单个字符会被指向字符，并且会用上计数器、缓存池，如果不是单个字符则指向字符串。

3、《C陷阱与缺陷》的笔记-第48页

边界判断容易出现off-by-one error，避免越界的通用原则是1、分析最简单特例，得到结果外推，2、仔细计算。

避免错误的编程技巧在于：用第一个入界点和第一个出界点表示取值范围，这是不对称的表达式（左闭空间，右开空间）。例如 $i \geq 0; i < 10$ 。

`while (--n >= 0)` 是一种n次迭代的方法。它比 `n-- > 0` 速度更快。

4、《C陷阱与缺陷》的笔记-第1页

阿

《C陷阱与缺陷》

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:www.tushu000.com