# 从缺陷中学习C/C++

Lessons Learned from C/C++ Defects

刘新浙 刘玲 王超 李敬娜 等 编著

这是一本
在研究大量C/C++程序Bug基础上集结而成的书！

这是一本
集众多一线C/C++编程人员智慧的书！

这是一本
让您学好C/C++，绕过编程陷阱和障碍的必备案头书！

人民邮电出版社
POSTS & TELECOM PRESS

# 从缺陷中学习C/C++

Lessons Learned from C/C++ Defects

刘新浙 刘玲 王超 李敬娜 等 编著



人民邮电出版社
北京

**图书在版编目（CIP）数据**

从缺陷中学习C/C++ / 刘新浙等编著. -- 北京 : 人民邮电出版社, 2013.9
ISBN 978-7-115-32159-6

Ⅰ. ①从… Ⅱ. ①刘… Ⅲ. ①C语言－程序设计
Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第151124号

## 内 容 提 要

C/C++是广泛用于系统和应用软件开发的语言，也是使用最为广泛的编程语言。C/C++易学难用，尤其C++，经过20多年的发展，已经变得非常复杂，给编程人员带来了很大的挑战。那么如何在工程项目中用好C/C++语言、如何绕过Bug构建稳定可靠的生产系统、如何以最快速度全面了解C/C++编程中的陷阱和难点，编写出健壮可靠的代码呢？本书将通过102个案例，帮助程序员尽快从新手成长为专家。案例涵盖基础问题、编译问题、库函数问题、文件处理、类和对象、内存使用、多线程问题、性能问题等。读者每掌握一个案例就掌握了一个或几个知识点，就能避免一类问题。由于是从大量编程中总结出来的具体Bug案例中学习，这种学习方式更直接，让人印象更深刻。本书将为你成为C和C++高手、编写出完美的程序助一臂之力。

本书适合程序员、测试人员以及C和C++初学者使用，也可以作为各大专院校和培训学校的教学用书。



◆ 编　　著　刘新浙　刘　玲　王　超　李敬娜等
责任编辑　张　涛
责任印制　程彦红　焦志炜
◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
邮编　100061　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷
◆ 开本：720×960　1/16
印张：11.25
字数：176千字　2013年9月第1版
印数：1－3 500册　2013年9月北京第1次印刷

定价：39.00元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

# 本书作者

**主要编著人员：**

刘新浙、刘玲、王超、李敬娜、李爱华、陈足先。

**其他编著人员：**

谭淑丹、曹恒智、付英。

**素材提供者：**

刘新浙、刘玲、刘晓俊、王艳、李爱华、黄元君、谭淑丹、杨晓霞、王竟时、王超、李颖、曹恒智、李敬娜、陈足先、刘云卿、刘霏暄、郜翔、陶旭颖。

# 致　谢

本书在编写过程中得到了很多人的帮助。

除本书作者外，原淘宝北京搜索与广告算法测试小组的很多同事都参与了本书的审查工作，提出了文字等方面的一些改进建议，在此一并表示深深的感谢。在本书编写过程中，淘宝公司技术研发部尤其搜索技术部的一些同事，也给予了一些中肯的意见和反馈，在此表示衷心的感谢。

另外，要特别感谢淘宝公司北京测试部门负责人刘立川先生，没有他的鼓励和督促，作者可能就不会在两年多的写作中坚持下来，这本书可能就不会出版面世。

在本书即将出版时，我们非常荣幸地邀请到了著名计算机专家潘爱民老师为本书做推荐序。潘老师在百忙之中抽出时间通读本书，并作序，还给予了非常专业的修改意见，这让本书受益良多。

最后，感谢人民邮电出版社的编辑，在本书出版过程中出谋划策，给予了很专业的建议，并促成了本书的最终顺利出版。

编者

# 序

C++是易学难用的语言。经过20多年的发展，C++已经变得极为复杂，很多语言特性看似优美，但给工程上带来了很大的挑战。因此，如何在工程项目中用好C++语言，并不像通过一本教程来学会写C++程序那么简单，这是一个不断积累经验和吸取教训的过程。工程上推荐使用的往往是C++大量语言特性的一个子集，原因是为了避免在工程项目中埋下各种陷阱。

譬如，Google是一家重度使用C++的公司，它有许多开源软件（如Android）使用了C++语言。Google公开了一份“Google C++ Style Guide——Google C++编程风格指南”，阐明了在Google的项目中如何有效地使用C++，应遵守哪些规范，以避免各种可能的陷阱。Google的这份指南值得工程线上的C++程序员认真阅读。阿里巴巴也是一家重度使用C++的公司，内部有大量的工程用到了C++语言。虽然阿里巴巴集团没有统一的C++编程指导，但各个团队都在摸索并制定出符合自己需要的编程规范，处于快速积累经验的过程中。很高兴淘宝广告技术部的测试团队很早就意识到了C++语言在工程中的诸多陷阱，并有意识地将其收集和整理出来，这就是你看到的这本书。

《从缺陷中学习C/C++》这本书收集并整理了102个实例，这些实例都来自于工程一线实践，虽然大多数看起来像是初学者犯的低级错误，但实质上有一定的代表性，有的错误根源是对C++机制的不理解或者编译过程中的副作用，或者是对C++标准库的实现依赖。通过阅读这些实例，你可以对C++有更细致的理解。

当刘新浙邀请我为本书作序时，我欣然答应，并有幸成为这本书的第一个读者。当时觉得难能可贵的一点是，这样一本书并非是深入研究C++工程多年的C++高手

在指点技术，而是一群懂C++的测试工程师在跟开发工程师探讨如何避免C++语言本身的各种缺陷。这个角度提炼出来的知识，真正是千金难买，因为他们是从“地雷阵”上走过来分享教训的。

我认真阅读了所有的实例，颇有收获，也愿意推荐给每一位在工程线上从事C++编程工作的工程师。若能够结合本书与“Google C++ Style Guide——Google C++编程风格指南”一起阅读，我相信一定会有相得益彰之成效。

潘爱民

于杭州

# 前　言

这是一本在研究大量 C/C++程序 Bug 基础上集结而成的书！

这是一本汇集众多一线 C/C++编程人员智慧的书！

这是一本让您学好 C/C++，绕过编程陷阱和障碍的必备案头书！

## 为什么写这样一本书

• 在不同的项目或产品中，不同的开发人员重复着同样的 Bug，甚至同一个人重复相同的 Bug。如果将时间周期拉得更长一些看：一个程序员，从刚毕业参加工作到具备丰富编程经验，从一个新手到成为专家，在这个过程中，每个人都在重复着前人走过的弯路，重复着同样的编程错误。测试人员在日常工作中积累了大量验证 Bug 方面的经验，这些 Bug 是有价值的，总结出来可以让更多人受益。

• C/C++是软件/互联网行业最常用的编程语言之一，相对其他语言学习难度高，从新手到专家往往需要多年的磨练。另一方面，C/C++开发的系统往往更容易产生严重的生产事故，一旦出现事故，定位问题根源也比较困难。所以，每一个程序员掌握扎实的 C/C++基础知识，对于构建稳定可靠的生产系统非常重要。我们希望通过这本书帮助 C/C++程序员以最快速度全面了解 C/C++编程中的陷阱，编写健壮可靠的代码，从而达到提高软件质量、减少生产故障、提高工作效率的目的。

相对其他 C/C++类书籍，本书有以下特点和优势：

• 从具体 Bug 中学习

全书由 102 个案例组成，每个案例分析一个 Bug。读者掌握了一个案例就是掌握了一个知识点，就能避免一类问题。由于是从具体 Bug 案例中学习，这种学习方式更直接，更让人印象深刻。普通的 C/C++编程书，即便看完后，写代码时也无法避免 Bug，这是因为，书虽然看完了，知识也理解了，但你并不知道哪里有陷阱。

- 来源于工程实战的宝贵经验

本书中的所有案例都来自于软件/互联网行业开发生产中遇到的实际问题，都是在前人的错误和弯路中总结出来的实战经验，堪称C/C++编程方面的“干货”。

- 学习起来更有成就感

本书由一个个案例组成，在每个案例中，先给出错误代码示例，然后描述Bug的现象和后果，随后对该Bug进行具体分析，最后给出解决方案及建议。这种案例分析式的组织方式，会引导读者先对案例中提出的问题进行思考，当看到案例分析和解决方案时常常有恍然大悟的感觉，让学习过程变得简单，并充满乐趣。

- 更宽的知识面

一个C/C++程序员即使工作多年，由于受所接触项目和技术方向的限制，视野（C/C++编程中值得注意的知识点）往往是有限的。这本书中的案例收集于大量工程项目，几乎涵盖了C/C++编程中的方方面面，远超出一般程序员所能接触的范围。掌握了这本书中的内容，能避免大多数问题。

## 本书适用范围

- 本书适合已经写过一些C/C++代码、期望尽快积累实战经验的C/C++程序员阅读学习。本书也适合打算提高代码编写和代码阅读分析能力的软件测试人员，书中的每个案例都可应用于白盒测试中。

## 本书代码运行环境

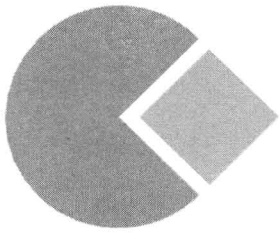
- 本书中的所有案例代码都是针对Linux C/C++环境，在Redhat Linux环境下编译（GCC，G++）测试通过。代码可以通过人民邮电出版社网站（www.ptpress.com.cn）下载。

最后，由于本书作者在C/C++编程方面的经验和技能有限，书中可能会有一些描述不够清晰或不正确的地方，读者如有发现请及时告诉我们，我们会再进行修正。我们的联系邮箱是cppbugbook@gmail.com，编辑联系邮箱是zhangtao@ptpress.com.cn。在此表示感谢！

期望这本书能真真切切地帮助到他人。

编者

# 目　录



# 第1章　基础问题

本章从基础问题讲起，涉及运算符优先级、宏定义、类型转换（显式和隐式）、指针、数组等C/C++编程的一些基本概念和知识点。这些问题虽然简单，但如果不引起重视，也会给编程带来很多麻烦。

## 1.1 运算符优先级引起的问题

- 代码示例

```
//To get 2*n+1
int func ( int n )
{
    return n << 1 + 1;
}
```

- 现象&后果

上述代码中的函数func本意是期望计算2*n+1，但程序实际运行结果是4*n。

- Bug分析

这段代码使用左移1位来代替乘以2的运算，是很好的方法，但是编程者弄错了运算符“<<”和“+”的优先级。C/C++语言规定运算符“+”的优先级高于运算符“<<”，因此，上述语句“return n<<1+1”等同于“return n<<(1+1)”，所以，会先进行加法运算，再进行左移运算，得到结果4*n。

修正的方法是在表达式中添加必要的括号。

- 正确代码

```
//To get 2*n+1
int func ( int n )
{
    return (n << 1) + 1;
}
```

## 1.2 不加括号的宏定义引起的错误

- 代码示例

```
#define PERIMETER(X,Y) 2*X+2*Y
int main(void)
{
    int lenth = 5;
    int width = 2;
    int high = 8;
    int result = 0;
    result = PERIMETER(lenth, width) * high;
    printf("result=%d \n", result);
}
```

- 现象&后果

上述代码是期望通过一个宏先计算一个矩形的周长，然后再乘以高，结果应该为 112，但实际计算结果为 42，与预期不符。

- Bug 分析

上述代码中，语句“result = PERIMETER(lenth, width) * high”本意是希望先进行宏定义部分的运算，然后再乘以变量 high，但是，由于宏替换是在预编译阶段进行的，宏替换本身只是文本替换，上述语句在宏替换后变成了“result = 2 * lenth + 2 * width * high”，按照运算符优先级规则会先做乘法运算再做加法运算，故导致运算结果不符合预期。因此，用于表达式的宏，在宏定义时给整体语句加上小括号是比较保险的做法。