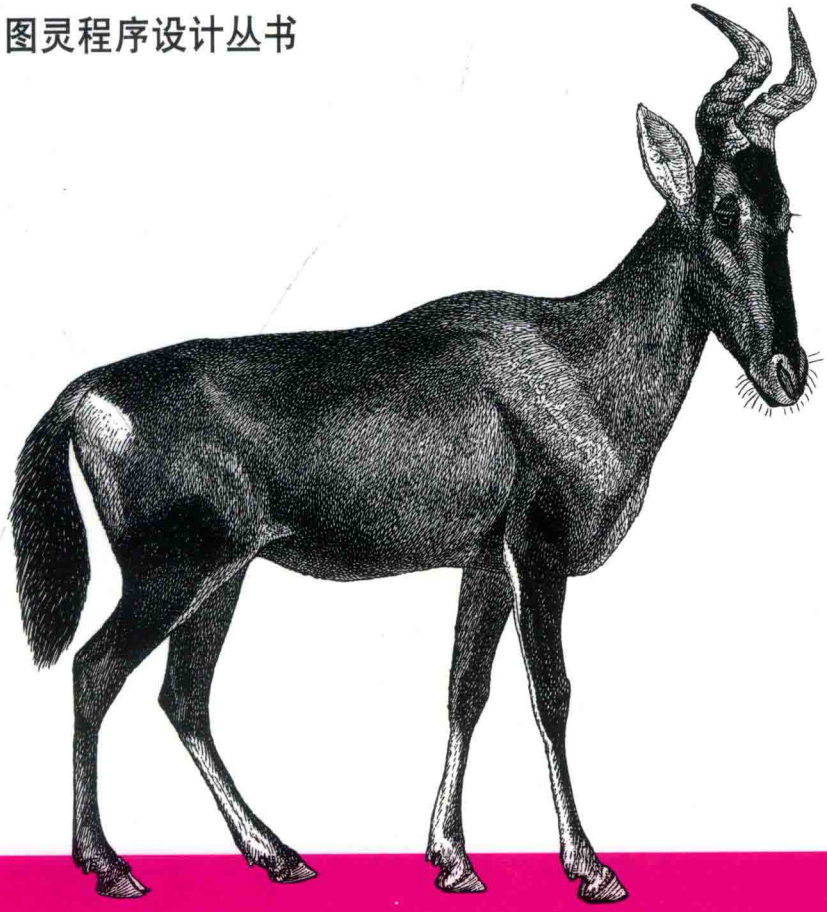
O'REILLY®

TURING 图灵程序设计丛书



C++性能优化指南

Optimized C++: Proven Techniques for Heightened Performance

精选编程中频繁使用和能够带来显著性能提升效果的技术

[美] Kurt Guntheroth 著

杨文轩 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

C++性能优化指南

Optimized C++

[美] Kurt Guntheroth 著
杨文轩 译

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®
O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目（CIP）数据

C++性能优化指南 / （美）柯尔特·甘瑟尔罗斯（Kurt Guntheroth）著 ; 杨文轩译. -- 北京 : 人民邮电出版社, 2018.1
（图灵程序设计丛书）
ISBN 978-7-115-47139-0

Ⅰ. ①C… Ⅱ. ①柯… ②杨… Ⅲ. ①C语言－程序设计－指南 Ⅳ. ①TP312.8-62

中国版本图书馆CIP数据核字(2017)第300400号

内 容 提 要

本书是一本C++代码优化指南。作者精选了他在近30年编程生涯中最频繁使用的技术和能够带来最大性能提升效果的技术，旨在让读者在提升C++程序的同时，思考优化软件之美。书中主要内容有：代码优化的意义和总原则，与优化相关的计算机硬件背景知识，性能分析方法及工具，优化字符串的使用，算法、动态分配内存、热点语句、查找与排序等等的优化方法。

本书适合所有C++程序员，也可供其他语言的程序员优化代码时作为参考。

◆ 著　[美] Kurt Guntheroth
译　杨文轩
责任编辑　朱　巍
责任印制　彭志环

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
大厂聚鑫印刷有限责任公司印刷

◆ 开本：800×1000　1/16
印张：19
字数：449千字　2018年1月第1版
印数：1－3 500册　2018年1月河北第1次印刷
著作权合同登记号　图字：01-2017-4807号

定价：89.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字20170147号

版权声明

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了*Make*杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过图书出版、在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——*Wired*

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

每一本书在出版时，作者都会感谢其伴侣。我知道这种方式平淡无奇，但我还是要感谢我的妻子 Renee Ostler，是她让本书得以出版。她为我腾出了家里的空间，并让我能够腾出几个月的时间专心写作。尽管她不太了解 C++ 性能优化，但她会一直陪伴我到很晚，问我关于这方面的问题，只是为了表示她的支持。她认为这本书对她很重要，因为它对我很重要。有妻若此，夫复何求。

前言

读者朋友，你好！我的名字叫 Kurt，是一名狂热的编程爱好者。

我编写软件已经超过 35 年了。我从未就职于微软、谷歌、Facebook、苹果或者其他知名公司。但是在这些年里，除了几个短暂的假期外，我每天都在写代码。最近 20 年里，我几乎只编写 C++ 程序，并与其他睿智的开发人员讨论 C++。所以我有资格写一本关于优化 C++ 代码的书。我还发表过许多文章，包括规范、评论、手册、学习笔记以及博客文章（http://oldhandsblog.blogspot.com）等。有时令我吃惊的是，我共事过的睿智能干的程序员中，只有半数能将两个符合英文语法的句子合并在一起。

我最喜欢的名言之一出自艾萨克·牛顿爵士的一封信。他在信中写道："我之所以看得更远，是因为我站在巨人的肩上。"现在，我也站在巨人的肩上了，特别是阅读过他们的著作：有优雅的小书，如 Brian Kernighan 和 Dennis Ritchie 合著的《C 程序设计语言》；有充满智慧且走在技术前沿的书籍，如 Scott Meyers 的 *Effective C++* 系列；有充满挑战又能扩展思维的书籍，如 Andrei Alexandrescu 的《C++ 设计新思维》；有科学严谨且讲解准确的书籍，如 Bjarne Stroustrup 和 Margaret Ellis 合著的 *The Annotated C++ Reference Manual*。在我职业生涯的大部分时间，我都从没想过有一天可以自己写一本书。但是突然有一天，我发现我需要写这本书。

那么为什么我要写一本关于 C++ 性能优化的书呢？

在 21 世纪初期，C++ 曾一度受到诟病。C 语言的支持者指出 C++ 程序的性能不如以 C 语言编写的相同代码。拥有巨额营销预算的著名企业吹嘘它们自己的面向对象语言，宣称 C++ 语言难以使用，而它们的工具才是未来。各大高校也决定教授 Java 语言，因为它有一套免费的工具链。由于以上种种原因，大公司投资大笔金钱使用 Java、C# 或是 PHP 来编写网站和操作系统。C++ 看起来正在逐渐衰落。对于任何相信 C++ 语言是强大且有用的工具的人而言，那是一段困难的时期。

就在这时，一件有趣的事情发生了。处理器核心的处理速度停止增长，但是工作负荷在持续加大。于是，那些大公司又开始重新雇用 C++ 程序员去解决它们的扩容问题。用 C++ 从头开始重新编写代码的成本变得比在数据中心消耗的电费要便宜。突然之间，C++ 再度流行起来了。

C++ 与 2016 年年初那些高市场份额的编程语言[1]相比非常突出的一点是，它为开发人员提供了一连串的可选实现方式，从全自动、自动支持到精准手动控制。C++ 赋予了开发人员掌控性能权衡的力量，这种掌控让性能优化成为可能。

市面上介绍如何优化 C++ 代码的书并不多。其中之一是由 Dov Bulka 与 David Mayhew 精心研究所著，不过现在看来有些过时的《提高 C++ 性能的编程技术》。这两位作者似乎与我有相似的职业经历，也发现了很多相同的优化原则。若想看看其他人怎么看待本书中提出的问题，推荐从《提高 C++ 性能的编程技术》开始。另外，Scott Meyers 等人也广泛地讨论过如何避免使用复制构造函数。

需要掌握的关于代码优化的知识太多了，足够写出 10 本书。在本书中，我精选出了自己在工作中频繁使用的技术和能够带来最大性能提升效果的技术进行讲解。奋战在性能优化前线的读者可能会有疑问，为什么本书中没有介绍任何有助于他们解决问题的对策。对于这些疑问，我只能说："对不起，内容太多，篇幅有限。"

欢迎大家将勘误、评论和最喜欢的优化策略发送至 antelope_book@guntheroth.com。

我热爱软件开发。我享受永不停歇地练习每一种新的循环方式和接口。 编写代码是一门科学，也是一门写诗艺术；它是一项冷僻的技术，一种内在的艺术形式，以至于除了少数同行外几乎没有人懂得欣赏它。优雅的函数中蕴藏着美感，被广泛使用的强大的惯用法中蕴藏着智慧。但令人遗憾的是，每一部史诗般的软件诗篇（如 Stepanov 的标准模板库），都对应着 10 000 行单调、庞大、枯燥的代码。

本书的根本目标是帮助读者思考优化软件之美。请记住这一点并践行之。请看得更远一些！

为本书中的代码致歉

虽然我已经编写和优化 C++ 代码超过 20 年了，但是本书中所出现的大多数代码都是特意为本书编写的。就像所有新编写的代码一样，这些代码也会有缺陷。为此，我深表歉意。

多年以来，我一直为 Windows、Linux 和各种嵌入式系统进行开发，而本书中展示的代码则是我在 Windows 上编写的。因此，毫无疑问，本书中的代码和内容都更加偏向于 Windows。那些使用 Windows 操作系统中的 Visual Studio 讲解的优化 C++ 代码的技术，同样也适用于 Linux、Mac OS X 或者其他 C++ 环境。不过，不同优化方式的正确使用时机则取决于编译器和标准库的实现以及代码是在哪种处理器上测试的。优化是一门实验科学。盲目地信任优化建议往往会失望。

我知道在不同的编译器之间、Unix 系统和嵌入式系统之间存在的兼容性是非常难以应对的。如果书中的代码无法在你最喜爱的系统上编译通过，我感到非常抱歉。本书并不会讲解跨系统的兼容性，我个人倾向于展示简单的代码。

我并不喜欢下面这种大括号缩进风格：

```
if (bool_condition) {
    controlled_statement();
}
```

注 1：可以从 http://www.tiobe.com/tiobe-index/ 查询各编程语言的市场份额。——译者注

不过，这种风格的优点是可以在一页中放入更多的代码行，因此，我选择在本书中使用这种风格。

示例代码的使用

可从以下地址下载本书示例代码、解决方案示例等附带资料：guntheroth.com。

本书是要帮你完成工作的，所以书中的示例代码通常可以直接拿去用在你的程序或文档中。除非你使用了很大一部分代码，否则无需联系我们获得许可。比如，用本书的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN，比如“*Optimized C++* by Kurt Guntheroth(O'Reilly). Copyright 2016 Kurt Guntheroth, 978-1-491-92206-4”。

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 permissions@oreilly.com 与我们联系。

排版约定

本书使用了下列排版约定。

- 黑体
  表示新术语。
- 等宽字体（`constant width`）
  表示程序片段，以及正文中出现的变量、函数名、环境变量、语句和关键字等。

目录