臻完善

软件调试与优化典型应用

石磊玉　编著

CD-ROM

包含书中调试基础、调试技术和缺陷管理的所有代码

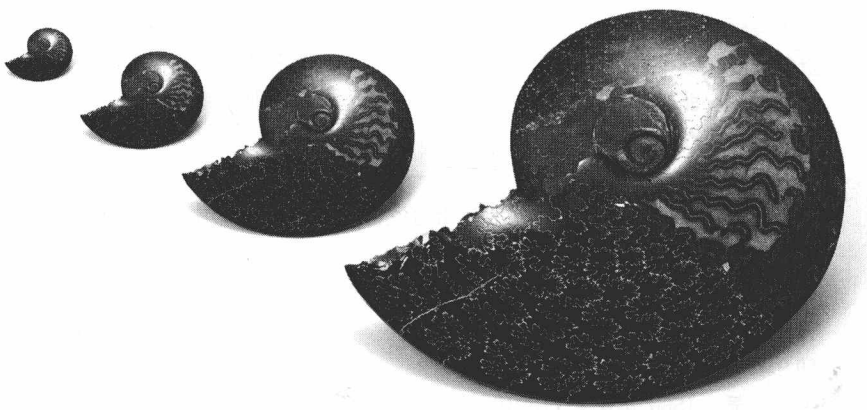
展示凝练代码、规范编码习惯

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

日臻完善

软件调试与优化典型应用

石磊玉　编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书在介绍相关内容时，通过实例阐述使晦涩的理论知识变得生动易懂。实例中的代码都是在 Visual C++ 6.0 开发环境中编写的，但调试理念和思想与软件开发环境无关，只是调试工具的操作细节不同。

本书以软件开发过程的全局来介绍软件调试，涉及编码过程的规范、测试用例的编写、事后调试、缺陷管理等多个开发环节的内容，介绍了各种常用开发技术的调试方法，包括静态链接库调试、动态链接库调试、多线程调试、内存泄漏、内核对象泄漏等。

本书主要面向软件程序编码人员、程序测试人员等软件领域的技术从业人员，同时软件开发管理人员也可从本书中了解到软件缺陷管理方面的内容。

图书在版编目（CIP）数据

日臻完善：软件调试与优化典型应用/石磊玉编著.
北京：中国铁道出版社，2009.11
ISBN 978-7-113-10719-2

Ⅰ.日… Ⅱ.石… Ⅲ.软件－调试 Ⅳ.TP311.5

中国版本图书馆 CIP 数据核字（2009）第 205324 号

书　　名：日臻完善——软件调试与优化典型应用
作　　者：石磊玉　编著

策划编辑：荆　波　　**编辑部电话：**（010）63583215
责任编辑：苏　茜　　**封面制作：**白　雪
特邀编辑：张丽花　　**责任印制：**李　佳
封面设计：付　巍

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号　　邮政编码：100054）
印　　刷：三河市华丰印刷厂
版　　次：2010 年 1 月第 1 版　　2010 年 1 月第 1 次印刷
开　　本：787mm×960mm　1/16　　**印张：**25　　**字数：**393 千
印　　数：3 000 册
书　　号：ISBN 978-7-113-10719-2/TP・3669
定　　价：59.00 元（附赠光盘）

FOREWORD

前 言

为什么写这本书

可能你还没有注意到，与程序编码相比，程序调试在开发中的比重更大，成本更高。有开发经验的人都知道，对于一个要实现的逻辑，要一口气顺利写完代码而且运行正确几乎是不可能的，其中出现的语法错误可以通过编译器的帮助来解决，这些错误相对简单，对于潜伏在代码中的逻辑错误，在运行时才会暴露出来，如导致程序崩溃或者给出错误的结果等，而且它们可能在发布时都没有被发现，成千上万的用户在使用时偶尔发现，这类错误会影响软件的市场。

调试程序是一件烦琐的事情，掌握必要的调试手段可以提高程序开发效率。对于一个程序开发者，除了掌握软件开发技术之外，还要掌握调试技术，没有掌握调试技术就如同一个人没有了免疫系统一样。能够巧妙、熟练地使用各种调试手段可以让程序开发工作变得轻松。程序调试存在于编码、测试、维护等诸多环节，一个合格的开发者应该能够游刃有余地处理各个环节出现的错误。

调试技术更多的是一种经验，需要通过不断积累才能逐渐掌握，很多开发者都是在长时间的调试后才慢慢熟练起来。要是有一本综合介绍调试技术的书籍，就可以大大减少初学者入门的时间。基于这个想法，本书精选了使用 Visual C++开发所必须的调试技术，从软件开发的多个方面来介绍相关的调试方法，如 Visual C++工具的调试功能、共享库的调试、多线程程序的调试、内存泄漏的调试、异常处理、SQL 调试等，技术针对性非常强。读者通过阅读此书，能够迅速掌握各种程序调试手段。

全书学习地图

本书总结了作者软件开发中的调试经验，将这些经验整理成册，是为了帮助读者轻松掌握调试技术。本书着眼于软件工程，系统介绍了各种调试技术、避免错误的方法、缺陷管理等多个方面的内容。

本书的读者对象为已经学习过 C/C++编程语言，掌握了基本的 C++语法，有一定程序开发基础，并且可以使用 Visual C++进行简单程序开发的人员。

本书以软件工程中的各个开发环节为主线，介绍了编码、测试、维护、管理等

多个环节的调试技术。首先介绍开发中的编码规范，这是软件工程对开发人员最基本的要求。通过对编码规范的介绍，读者可以约束自己的行为，编写优良的代码。久而久之，就会养成一种良好的习惯，不会觉得规范是一种约束，相反会觉得遵守规范会让自己的开发工作变得轻松有序。然后介绍开发人员使用最多的 Visual C++ 开发工具的调试功能，熟练掌握这些功能是开发人员最基本的要求。本书中介绍的各种调试技术都是基于这两点的，针对开发中常见的调试技术分别进行详尽阐述，这些调试技术在软件开发的许多环节都非常实用。本书的最后一章介绍了缺陷管理相关的内容，科学的缺陷管理可以更好地积累团队的开发经验，从而避免在后续开发中走弯路。

软件工程涉及多个开发者，开发过程中的任何操作都有章可循才能保证工程的顺利进行。编码规范和良好的编码习惯贯穿于本书始终。

本书特色

本书将对读者有用的调试技术进行分类，查看本书目录就可以清晰地把握这一点，它有利于读者深入研究各个门类。如果读者当前只关心程序内存泄漏的问题，想要深入了解导致内存泄漏的原因以及调试手段，只需要查看本书第 8 章的内容即可。本书针对各种类别程序的开发将调试技术进行分类，如库调试、多线程调试、SQL 调试、内存调试、内核对象调试、异常处理、事后调试等。

本书站在软件工程的高度，剖析各个环节的调试技术。从编码规范、开发环节中的调试技术、测试环节中的调试技术、维护环节中的调试技术、软件缺陷管理等多个角度对软件调试进行了详细的介绍，并且与软件开发过程一一对应，便于读者从软件工程的角度把握软件调试，从而在更高的层次上把握全局。

本书注重理论结合实际。在介绍各种技术的同时，给出了针对性很强的示例，而且尽量做到图文并茂。本书中不乏凝练的代码、图表，通过这些读者可以轻松地理解作者意图，把握本书精髓。

本书不仅针对各种错误给出了调试方法，同时给出了预防措施，防患于未然。帮助开发者在开发中尽量少犯错误，养成良好的编码习惯。

读者定位

如果你致力于成为一名合格的软件开发者，阅读本书将对你非常有用。

事实上，软件领域的从业人员都可以成为本书的读者。程序编码人员可以通过阅读本书缩短调试时间，让开发工作变得更加轻松愉快。程序测试人员在掌握了调

试技术后，可以为编码人员提供更详尽的信息，缩短错误排查时间，加快开发进度。程序维护人员经常与用户打交道，大多数情况下，用户不是软件开发人员，在软件出错时，他们不能提供专业的信息，这时候就需要一定的技巧来搜集错误，本书也涉及了这方面的内容。软件管理人员可以通过本书了解软件缺陷管理的相关内容。

全书结构安排

本书从层次上可以分为 3 大部分，分别是调试基础、调试技术、缺陷管理。

第 1 章～第 3 章属于调试基础部分。从编码层次上介绍简单实用的代码编写规范，介绍 PCLint 静态代码检查工具的使用，还讲述了重构的思想和实施方法，同时重点总结了与代码测试相关的内容。第 3 章详细阐述了 Visual C++工具的调试功能，这些功能非常强大，熟练使用它们可以大大减少开发人员调试的时间。

第 4 章～第 11 章属于调试技术部分。这些章节根据程序开发中常用技术或常见问题的调试方法进行分类说明，如多线程程序调试、内存问题调试等。每个章节各有所指，针对性极强，如“事后调试”一章，几乎搜集了当前所有常用的事后调试方法。这部分每个章节可以独立地作为专题进行学习，读者可以很容易通过章节名称知道该章的意图，在阅读此书时，可以针对自己感兴趣的知识点进行深入研究。

第 12 章属于缺陷管理部分。软件缺陷管理是软件工程中很关键的一个环节，合理的缺陷管理可以确保项目推进时不再出现相同的缺陷，在出现类似缺陷时，参照以前的处理方案可以很快地解决问题。是否有科学的软件缺陷管理是软件开发团队成熟性的一个重要标志。

作者团队

本书由石磊玉编著，同时特别感谢赵婷娟老师为本书的顺利完成所付出的努力。

感谢

一本书的出版，从选题申报到出版，要经历很多环节，在此感谢中国铁道出版社的编辑们，他们不辞辛苦，为本书的出版做了大量工作。

读者服务

读者在学习过程中，如果遇到与本书相关的问题，可以登录到 www.rzchina.net 技术支持网站，在网上注册后便可提交问题，我们的老师会在线回答读者提出的各种技术问题。

编者

2009 年 8 月

CONTENTS

目 录

第1章 绪 论

提到冒泡排序，你可能会觉得实现起来非常简单，但是如果要求将冒泡排序的实现一口气编写完成，在没有调试的情况下，没有谁敢保证刚刚编写的代码是绝对正确的。

其实调试本身就是开发过程不可缺少的一部分，通过调试可以消灭语法错误，也可以配合一定的开发手段确保功能的正确性，这些是调试的常规功能。

很多情况下，调试可以解决那些“绝对不可能出错但是确实出了错”的问题，这些问题往往使开发人员觉得棘手，它消耗的开发时间经常远远超过编写代码本身的时间。

稍有程序开发经验的人都知道，调试程序是一个烦琐又复杂的过程，软件系统的规模越强大，就越是如此。如果开发人员能够更好地掌握调试方法，就能提高调试技能，从而提高程序开发效率，这一点在软件开发人员中已达成共识。

本书就是基于这个共识，搜集和整理那些零散的调试方法。

1.1 不可避免的调试工作

说到软件开发，很多人大脑中的第一反应就是编码，其实软件开发不仅仅只有编码，调试也占据了开发人员很大一部分精力。

开发人员“简单”的工作中，80%的时间都是在编码和调试。调试的对象是bug，bug 就是编码过程的伴生品，“凡是软件必有 bug”，很多项目的延期以及程序员的加班也是由于调试棘手的 bug 导致的。不仅在软件开发过程中需要调试，在软件开发的测试、维护阶段都需要调试。

在编码阶段，需要依靠调试器帮助我们解决语法错误，在自测时需要调试手段解决语义错误，测试阶段需要不同的调试方法解决出现的 bug，维护阶段需要与用户（非开发人员）交互完成调试任务。

1.1.1 不存在完美的程序

世界上不存在没有 bug 的程序，软件系统越是庞大，出现 bug 的可能性越大。

人是善于创造错误的，人创造错误的能力远远高于解决错误的能力。人的思维能力是有限的，系统越复杂，人的思维驾驭这些复杂逻辑就越难，在设计和实现上也越容易出现漏洞。

正因为如此，在人类创建大型系统时，总是先将比较庞大的系统分解为若干个相对比较独立的小系统，这些小系统在功能定义上是比较完备和独立的，将复杂的逻辑分解为一系列简单的逻辑，使问题变得简单，减少出错的可能性。

软件系统是开发者的智力产品，是迄今逻辑最复杂的工作，要想开发出不含错误的程序真的是很难。加之软件开发进度压力，开发人员总是在高压下进行开发，一些错误更是很难避免的。

承认错误不可避免并不是要放纵自己随意发挥，而是要更加严格地要求开发人员遵守开发规范，在开发代码过程中要有忧患意识，不能因为赶进度而放松代码质量要求，成为后期维护的大患。

1.1.2 系统经常不按期望运行

经常会出现这样的情况，用户向你报告了一个错误，同时提供了出错的描述：在单击某个按钮或执行某个操作时程序偶尔会崩溃掉！这个时候，开发人员将面临两个问题：一是搜集崩溃前系统的状态，二是分析崩溃产生的原因并修订软件。

软件崩溃总是让用户觉得厌烦，软件崩溃会影响软件的市场，这是公司老板向员工施加压力的直接原因，来自项目开发进度的压力是增加出错的原因之一，因为这种压力会影响开发人员的心情。