

## <<软件调试的艺术>>

### 图书基本信息

书名：<<软件调试的艺术>>

13位ISBN编号：9787115213969

10位ISBN编号：7115213968

出版时间：2009-11

出版单位：人民邮电出版社

作者：Norman Matloff, Peter Jay Salzman

页数：216

字数：337000

译者：张云

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>

## <<软件调试的艺术>>

### 前言

“嘿，还真不错呢！”

”我们的学生Andrew首次认真使用调试工具后惊叹道。

Andrew三年前在大一上编程课时就学过了调试工具，不过那时他只是将调试工具当做应付期末考试的内容来打发的。

现在他已经大四了，教授强烈要求他停止采用输出语句进行调试的方法，改为使用正式调试工具。

让Andrew喜出望外的是，他发现使用恰当的工具可以大大缩短调试时间。

如今，在学生及已参加工作的程序员中，还有不少“Andrew”，但愿本书能帮助“Andrew们”发现调试工具的好处。

但是，我们更希望本书能帮助那些已经使用了调试工具，但还无法确定在什么情况下该做什么事的程序员做出适当的判断。

本书也适用于打算进一步学习调试工具及其幕后原理的人。

本书的编辑曾说过，很多调试知识以前都只是在圈子里口口相传，没有正式编著成书。

本书将改变这一状况。

本书回答了下列问题。

如何调试线程代码？

为什么有时候断点最终出现的位置与原来设置的位置略有偏差？

GDB的until命令为什么有时会跳到意想不到的地方？

如何巧妙使用DDD和Eclipse？

在当今普遍使用GUI的时代，像GDB这样的基于文本的应用程序还有价值吗？

为什么当错误代码超出数组边界时不发生段错误？

为什么要将我们的一个示例数据结构命名为nsp？

## <<软件调试的艺术>>

### 内容概要

调试对于软件的成败至关重要，正确使用恰当的调试工具可以提高发现和改正错误的效率。本书详细介绍了3种调试器，GDB用于逐行跟踪程序、设置断点、检查变量以及查看特定时间程序的执行情况，DDD是流行的GDB的GUI前端，而Eclipse提供完整的集成开发环境。书中不但配合实例讨论了如何管理内存、理解转储内存、跟踪程序找出错误等内容，更涵盖了其他同类书忽略的主题，例如线程、客户/服务器、GUI和并行程序，以及如何躲开常见的调试陷阱。本书适合各层次软件开发人员、管理人员和测试人员阅读。

## <<软件调试的艺术>>

### 作者简介

马特洛夫 (Norman Matloff) 美国加州大学戴维斯分校计算机科学教授, 研究领域涉及并行处理(尤其是软件分布式共享内存)、计算机通信网络、数据安全及数学和应用统计等。他编写了很多开源软件包。创作的在线教程也深受欢迎。

## &lt;&lt;软件调试的艺术&gt;&gt;

## 书籍目录

第1章 预备知识 1.1 本书使用的调试工具 1.2 编程语言 1.3 调试的原则 1.3.1 调试的本质：确认原则 1.3.2 调试工具对于确认原则的价值所在 1.3.3 其他调试原则 1.4 对比基于文本的调试工具与基于GUI的调试工具，两者之间的折中方案 1.4.1 简要比较界面 1.4.2 折中方法 1.5 主要调试器操作 1.5.1 单步调试源代码 1.5.2 检查变量 1.5.3 在GDB、DDD和Eclipse中设置监视点以应对变量值的改变 1.5.4 上下移动调用栈 1.6 联机帮助 1.7 初涉调试会话 1.7.1 GDB方法 1.7.2 同样的会话在DDD中的情况 1.7.3 Eclipse中的会话 1.8 启动文件的使用第2章 停下来环顾程序 2.1 暂停机制 2.2 断点概述 2.3 跟踪断点 2.3.1 GDB中的断点列表 2.3.2 DDD中的断点列表 2.3.3 Eclipse中的断点列表 2.4 设置断点 2.4.1 在GDB中设置断点 2.4.2 在DDD中设置断点 2.4.3 在Eclipse中设置断点 2.5 展开GDB示例 2.6 断点的持久性 2.7 删除和禁用断点 2.7.1 在GDB中删除断点 2.7.2 在GDB中禁用断点 2.7.3 在DDD中删除和禁用断点 2.7.4 在Eclipse中删除和禁用断点 2.7.5 在DDD中“移动”断点 2.7.6 DDD中的Undo/Redo断点动作 2.8 进一步介绍浏览断点属性 2.8.1 GDB 2.8.2 DDD 2.8.3 Eclipse 2.9 恢复执行 2.9.1 在GDB中 2.9.2 在DDD中 2.9.3 在Eclipse中 2.10 条件断点 2.10.1 GDB 2.10.2 DDD 2.10.3 Eclipse 2.11 断点命令列表 2.12 监视点 2.12.1 设置监视点 2.12.2 表达式第3章 检查和设置变量 .....第4章 程序崩溃处理第5章 多活动上下文中的调试第6章 特殊主题第7章 其他工具第8章 对其他语言使用GDB/DDD/Eclipse

## &lt;&lt;软件调试的艺术&gt;&gt;

## 章节摘录

插图：1.3.1 调试的本质：确认原则下面的规则是调试的本质。

· 确认的基本原则修正充满错误的程序，就是逐个确认，你自认为正确的许多事情所对应的代码确实是正确的。

当你发现其中某个假设不成立时，就表示已经找到了关于程序错误所在位置（可能并不是准确的位置）的线索。

换一种表达方式来说：惊讶是好事！

当你认为关于程序的某件事情是正确的，而在确认它的过程中却失败了，你就会感到惊讶。

但这种惊讶是好事，因为这种发现会引导你找到程序错误所在的位置。

1.3.2 调试工具对于确认原则的价值所在传统的调试技术只是向程序中添加跟踪代码以在程序执行时输出变量的值，例如使用printf（）或COllt语句输出变量的值。

你可能会问：“这样操作够吗？”

为什么要使用GDB、DDD或Eclipse这样的调试工具？

”首先，这种方法要求有策略地持续添加跟踪代码，重新编译程序，运行程序并分析跟踪代码的输出，在修正程序错误之后删除跟踪代码，并且针对发现的每个新的程序错误重复上述这些步骤。

这种工作过程非常耗时，并且容易令人疲劳。

最为重要的是，这些操作将你的注意力从实际任务转移开，并且降低了集中精力查找程序错误所需的推理过程的能力。

相反，通过使用DDD和Eclipse这样的图形调试工具，只需要将鼠标指针移动到代码显示中的变量实例上方就可以检查该变量的值，并且此时会显示该变量的当前值。

为什么还要在整夜的调试中使用printf（）语句来输出变量的值，使自己更加疲劳，等待更长的时间呢？

放松心情，使用调试工具可以减少所花费的时间和需要忍受的厌烦感。

## <<软件调试的艺术>>

### 媒体关注与评论

“Linux/Unix平台的C程序员将从本书中获益最多，其他平台上的开发者也能从中学到宝贵的调试技术和秘诀。

”——D. Spinellis, Jolt奖获奖图书《高质量程序设计艺术》作者“本书是一部软件调试杰作，可以大大提升你的调试技术水平……所有编程入门课程都应该讲授其中的核心内容。

我强烈推荐本书！

”——Amazon读者评论

## <<软件调试的艺术>>

### 编辑推荐

《软件调试的艺术》：调试对软件开发至关重要。然而，即使对于有经验的程序员，调试也并非易事。

《软件调试的艺术》是一部优秀的软件调试入门著作，对业界多年口口相传的调试经验做了很好的总结。

书中通过丰富的C/C++实例，生动阐述了适用于各种平台与编程语言的通用调试原则、基础知识、经验技巧和相关的背景知识，详细讲解了Linux/Unix平台最受欢迎的三个调试工具——GDB、DDD和Eclipse，并讨论了网络、多线程、GUI和多核程序的调试等高级主题。

不仅如此，书中还讲述了如何在调试中运用vim、gcc、errno和lint等工具，以及Java、Python、Perl和汇编等程序的调试。



<<软件调试的艺术>>

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>