TURING 图灵程序设计丛书 数据库系列

Apress®

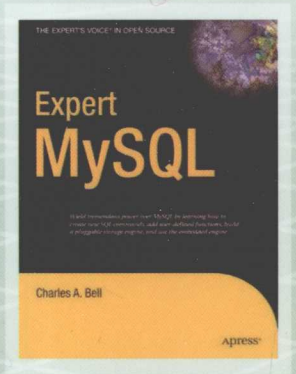
Expert MySQL

深入理解MySQL

[美] Charles A. Bell 著

杨涛 王建桥
杨晓云 韩兰 等译

- MySQL核心开发人员力作
- 带你深入MySQL源代码和底层架构
- 身临其境，透彻掌握数据库理论与实践



人民邮电出版社
POSTS & TELECOM PRESS

Expert MySQL

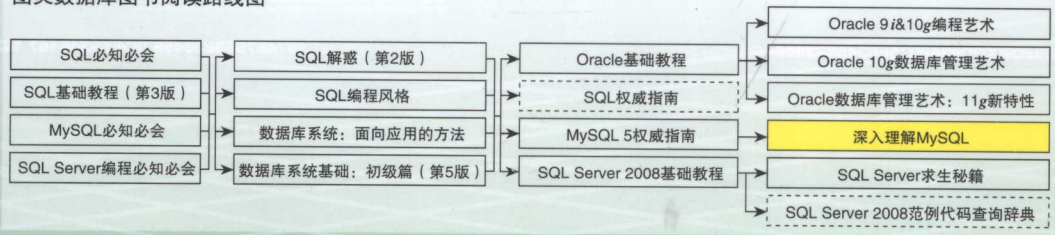
深入理解MySQL

MySQL是目前最流行的开源数据库，经过多年发展，日趋成熟，已经能够和主流的商业数据库相抗衡。

本书结合MySQL源代码深入讲解了MySQL数据库的核心知识。全书分为三个部分，从介绍数据库基础知识开始，逐步深入到存储引擎，最后介绍了查询优化器等数据库内部结构。第三部分还提供了一些有关数据库的实验，以便读者亲自动手来构建一个实验性质的数据库，从而加深对数据库内部结构的了解。作者很好地兼顾了理论与实践，使本书不仅适合数据库开发和管理人员阅读参考，也可以用于高校数据库相关课程的教学。在学习完本书后，你不仅将对MySQL有更加深入的理解，也会对数据库理论有全新的认识，成为一个数据库方面的行家里手。

Charles A. Bell MySQL核心开发人员，目前是Sun公司高级软件工程师；同时也是弗吉尼亚联邦大学的客座教授，主要是为研究生讲授计算机科学课程。他主要从事新兴技术的研究，研究方向包括数据库系统、版本系统、语义网和敏捷软件开发等。

图灵数据库图书阅读路线图



Apress®

本书相关信息请访问：图灵网站 http://www.turingbook.com
读者/作者热线：(010)51095186
反馈/投稿/推荐信箱：contact@turingbook.com

分类建议 计算机/数据库/MySQL

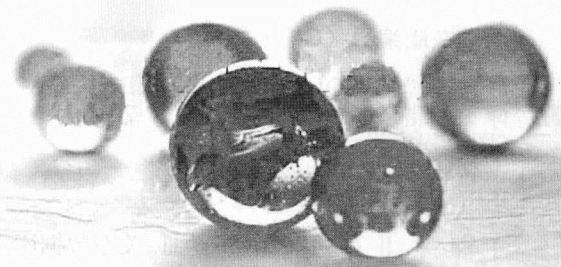
人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-18910-3

ISBN 978-7-115-18910-3/TP

定价：65.00元

TURING 图灵程序设计丛书 数据库系列



Expert MySQL
深入理解MySQL

[美] Charles A. Bell 著
杨涛 王建桥 杨晓云 韩兰 等译

人民邮电出版社
北 京

图书在版编目（CIP）数据

深入理解MySQL /（美）贝尔（Bell, C. A.）著；杨涛等译. —北京：人民邮电出版社，2010.1
（图灵程序设计丛书）
书名原文：Expert MySQL
ISBN 978-7-115-18910-3

Ⅰ.深… Ⅱ.①贝…②杨… Ⅲ.关系数据库-数据库管理系统，MySQL Ⅳ.TP311.138

中国版本图书馆CIP数据核字（2008）第150271号

内 容 简 介

本书深入源代码，剖析了MySQL数据库系统的架构，并提供了分析、集成和修改MySQL源代码的专家级建议。本书分三个部分：第一部分介绍开发和修改开源系统的概念，提供探讨更高级数据库概念所需的工具和资源；第二部分讨论MySQL系统，阐明如何修改MySQL源码，如何将MySQL系统作为嵌入式数据库系统；第三部分更深入地探讨了MySQL系统，讲述数据库工作的内部机理。

本书面向MySQL数据库开发人员。

图灵程序设计丛书

深入理解MySQL

◆ 著 [美] Charles A. Bell
译 杨 涛 王建桥 杨晓云 韩 兰 等
责任编辑 傅志红
执行编辑 傅尔也

◆ 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 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京顺义振华印刷厂印刷

◆ 开本：800×1000 1/16
印张：30
字数：803千字 2010年1月第1版
印数：1－3 000册 2010年1月北京第1次印刷

著作权合同登记号 图字：01-2007-4260号

ISBN 978-7-115-18910-3/TP

定价：65.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

MySQL已被公认为是世界上最流行的开源数据库产品和行业内增长最快的数据库系统之一。来自MySQL AB公司[①]的统计报告显示，MySQL的安装数量已超过800万，每天的下载量接近5万人次。MySQL正迅速成为系统集成商首选的数据库系统。据*SD Times*报上的一篇文章报道，对900多位读者进行的调查表明，MySQL在“装机量最大的数据库”榜上排名第三（www.mysql.com/why-mysql/marketshare/）。

本书对数据库系统的一些高级问题进行了探讨，对MySQL的体系结构进行了剖析，还为分析、集成和修改MySQL源代码使之用于企业级环境提供了专家级建议。在如何修改MySQL系统来满足系统集成商和教育科研机构的独特需求方面，本书提出了独到的见解。

本书结构

本书分为三个部分，每个部分对一组彼此相关的问题进行探讨，内容从MySQL和开源运动的发展，到扩展和定制MySQL系统，甚至还讲述了如何建立一个实验性查询优化器和执行引擎来替代MySQL查询引擎等。

第一部分

本书的第一部分对开发和修改各种开源系统所涉及的基本概念进行了介绍。这一部分为探讨本书后面介绍的更高级的数据库概念提供了必需的工具和资源。

与本书的其他章相比，第1章对技术性问题的探讨相对较少，多是些叙述性的内容。这一章的目的主要是让大家了解开源系统集成商都有哪些权益和责任。这一章突出介绍了MySQL的快速成长及其在开源和数据库系统市场中的重要性。此外，这一章还清晰地勾勒出了开源运动发展的脉络。

第2章对什么是数据库系统和怎样构造数据库系统等基础知识进行了介绍。对MySQL系统的剖析充分展示了现代关系数据库系统的关键组件。

第3章对MySQL软件的源代码以及如何获得和构建一个MySQL系统做了全面的介绍。主要内容包括MySQL源代码的内部机制以及编码指导原则和如何维护源代码的最佳实践。

第4章介绍了生成高质量MySQL系统扩展的一个关键方面。这一章讲解了软件测试技术以及测试大型软件系统常用的实践方法，采用几个具体示例展示了几种已被广泛接受的测试MySQL系统的方法。

---

① 2008年1月，MySQL AB公司被Sun公司收购。2009年4月，Sun公司被Oracle公司收购。——编者注

第二部分

第二部分采用实际操作的方法来研究MySQL系统。这一部分介绍如何修改MySQL代码，以及如何把MySQL系统用作嵌入式数据库系统。还通过各种示例和项目向读者演示如何调试源代码，如何修改SQL命令来扩展这种语言，以及如何创建定制的存储引擎。

第5章介绍了一些调试技巧和技术，有助于保证开发工作更容易，减少不必要的错误和麻烦。在介绍各种调试技术的时候，还对它们的优缺点进行了分析和说明。

第6章指导读者掌握如何把MySQL系统嵌入企业级应用程序。这一章的示例项目将帮助读者运用学到的技巧来进行系统集成。

第7章是本书探讨MySQL代码修改问题的第一章。这一章演示了几种只需修改少量的MySQL代码就可以达到目的的技术。重点探讨MySQL的插件式存储引擎的能力，并通过有关的示例和项目构建一个示范性的存储引擎。

第8章介绍了最流行的MySQL代码修改技术。向读者展示了如何修改SQL命令以及如何建立定制的SQL命令。这一章给出了几个例子说明如何修改SQL命令以添加新参数、新函数和新命令。

第三部分

第三部分深入MySQL系统的内部去探查这个系统的工作原理。首先介绍了一些高级的数据库技术，精辟阐述了有关理论和实践，使读者能够运用所学到的知识去解决与数据库系统有关的更为复杂的问题。这一部分还给出了一些例子，介绍如何实现内部查询表示，如何实现新的查询优化器，以及如何实现新的查询执行机制。并对有关的示例和项目作了详细的讨论。第10～12章演示了如何改变MySQL系统的内部结构，以实现新的查询处理机制。这几章为如何建立和修改大型系统提供了独到的见解。

第9章介绍一些高级的数据库技术并对MySQL体系结构进行深入分析。主要内容包括查询执行、多用户问题以及编程时的注意事项等。

第10章讨论MySQL的内部查询表示，介绍了一个新的示例查询表示。主要讨论了如何通过修改MySQL源代码来实现新的查询表示。

第11章探讨了MySQL内部查询优化器，介绍一个示例性的新的查询优化器，这个查询优化器使用了第10章实现的新的查询表示。读者可以学会如何通过修改MySQL源代码来实现一种新的查询优化器。

第12章把前几章介绍的技术结合起来，指导读者修改MySQL系统来实现一种新的查询处理引擎技术。

附录

本书的附录列出了一份MySQL、数据库系统和开源软件的资源清单。

将本书作为讲授数据库系统内部结构的教材

介绍关系数据库理论和实践的优秀教材有很多。但是，适用于课堂教学和实验环境的资料并不多见，能帮助学生钻研数据库系统内部工作原理的资源就更少了。本书为那些通过实际动手实验来充实

其数据库课程内容的教师提供了一个机会。在课堂上使用本书的方式有三种。

首先，本书可以用来增加本科生或研究生的数据库初级课程的深度。本书的第一部分和第二部分对数据库系统的一些特殊主题进行了深入的讲解。推荐将第2、3、4章和第6章的内容作为授课主题，这几章的主题可当作对更为传统的数据库理论或数据库系统课程的补充。学生动手实践和课堂项目可以从第6章和第8章节选。

本书的第一部分和第二部分内容可用来开设一门本科生和研究生的高级数据库课程，这两个部分里的每一章都适用于课堂教学，可在8～12周讲完，多出来的授课时间可以用来讨论物理存储层的实现问题或加深对存储引擎的理解。学期项目可以以第7章为基础，让学生自行构建一个存储引擎。

面向高年级本科生或研究生的数据库系统高级专题课程可以使用本书作为基本教材，并把本书的前9章内容当作课堂教学的基础。学期项目可以借鉴本书第三部分内容，让学生为一个实验性数据库平台实现它还缺少的功能，包括语言理论、查询优化器、查询执行算法的应用。

开始行动吧

本书充分考虑了各类读者的需求。不论是与数据库系统已经打过多年交道，还是只听了一门数据库理论基础课，甚至只读过Apress公司出版的某本优秀的MySQL书，相信你都可以从本书学到许多东西。如果你想了解像MySQL这样的数据库系统是如何运转的，你甚至可以从源代码入手！

致谢

Apress出版公司到处都是充满天赋而又精明强干的专业人士，我要感谢他们当中的许多人。本书的编辑Jason Gilmore和项目经理Tracy Brown Collins都有着极大的耐心和非凡的见地。正是因为他们的努力，本书才能如期完成，让我言而有信。我还要感谢本书的生产编辑Katie Stence和文字编辑Liz Welch，他们让本书的印刷效果看起来相当不错。非常感谢两位！

我还要特别感谢以下几位技术审稿人：L. M. Parker和Mikael Ronström，正是他们毫不松懈地严格把关才保证了本书的质量；还有Michael Kruckenberg，他保证了本书编程示例的正确性，他对MySQL独特的见解和丰富的经验让我非常佩服。可以说，我曾与精英中的精英一起合作。

最后，我还要感谢我妻子Annette无尽的耐心和理解。

目　录

第一部分　MySQL 开发入门

Part 1 第一部分

MySQL 开发入门

这一部分介绍了一些与开源系统的开发和修改有关的基本概念。第1章介绍开源系统集成商有哪些权益和责任，重点讲述了MySQL的快速成长及其在开源和数据库系统市场中的重要性。第2章对什么是数据库系统以及如何构造数据库系统等基础知识进行介绍。第3章对本书所涉及的MySQL源代码以及如何获得和建立MySQL系统作了全面介绍。第4章介绍为MySQL系统生成高质量扩展的一个关键部分。读者在这一部分将学到软件测试技术以及一些测试大型软件系统的常用方法。

本部分内容

第1章 MySQL与开源运动

开源系统正迅速成为改变软件前景的一支重要力量。开源厂商在软件开发和技术支持方面展现出来的高品质，已成为全世界的信息技术专家所关注和议论的焦点，有许多开源厂商的产品和服务都达到了世界一流的水准。大公司在关注着，因为开源软件能够让它们第一次有机会摆脱商业软件厂商的专利产品；小企业也在关注，因为开源软件可以大幅降低它们在信息系统方面的成本开支；开发人员也在关注，因为开源软件让他们有了更大的挑选余地。在现如今的因特网上，有许许多多的网站都是在Linux、Apache HTTP服务器、BIND、Sendmail、OpenSSL、MySQL和其他一些开源软件的基础上搭建起来的。

促使人们选用开源软件最常见的商业理由是成本。开源软件在从根本上减少软件产品的总体拥有成本（Total Cost of Ownership，TCO）的同时，还提供了一种可行的商业模式，让形形色色的企业去构建和完善它们的市场。具体到开源数据库系统，这一点体现得尤其明显。商业化专利数据库系统少说也要几千美元才能买到，如果加上技术支持方面的费用，总成本往往很容易就达到数万甚至数十万美元。

在过去，有许多人认为开源软件只限于爱好者或黑客们使用，而这些人的目的不过是扰乱大型商业软件公司的市场而已。应该承认，有些开发人员觉得自己就是站在微软巨人面前的大卫[①]，但开源社区与这些人完全是两回事。开源社区不以彻底取代某种商业专利软件为目标，他们对“开源”的解释是“向人们提供另外一种选择”。本章将向你展示，开源软件不仅让人们在商业软件之外多了一种选择，还在软件的开发和营销方面掀起了一场世界性的革命。

注解 在本书中，黑客这个词的含义同理查德·斯托曼（Richard Stallman）对黑客的定义：“热爱编程并为自己的聪明才智感到骄傲的人。”这与报章杂志上面提到的专门偷盗信用卡和损坏他人计算机系统的坏人有着本质的区别。

下面一节是为那些对开源软件或MySQL基本哲理还不甚明了的人准备的。如果你已经很熟悉开源软件的基本哲理，就可以直接跳到1.2节。

1.1 什么是开源软件

开源软件是人们有意识地抵制软件专利思维的产物。理查德·斯托曼在就职于MIT（Massa-

① 圣经中，牧羊人大卫杀死了巨人哥利亚。这里借指某些开发人员认为他们是微软权威技术方面的克星。——编者注

chusetts Institute of Technology，麻省理工学院）的人工智能实验室时，曾于20世纪70年代发起过一场代码共享运动。斯托曼当时的出发点是希望常用的代码能够为全世界的程序员所共享，而这意味着开发人员可以在全世界的范围内彼此合作。这一理念在斯托曼和他的圈内朋友中贯彻得很好——当然，这一切都发生在软件业集体决定软件属于专利并禁止与潜在的竞争对手进行共享之前。在那之后，MIT的许多研究人员因为各种原因离开MIT进入了这些软件公司。最终，这个原本合作无间的小集体消亡了。

幸好，斯托曼没有随波逐流，他留在MIT并创立了GNU（GNU's Not Unix）项目和自由软件基金会（Free Software Foundation，FSF）。GNU项目的目标是推出一个没有专利束缚的Unix风格的操作系统。这个系统是自由的，并对所有人开放（包括对源代码的访问）。在这里，“自由”的概念是不禁止任何人使用和修改这个系统。

斯托曼的目标是重新组建一支能够像他们当初在MIT那样合作的开发人员社区。可是，斯托曼预见到这个系统需要一种版权许可证机制来保证一定程度上的自由。（据说，斯托曼用“copyleft”而不是“copyright”来表达自己对这种版权意义的理解，它的含义是保证软件能够自由交流而不是受到限制。）斯托曼起草了GPL（GNU Public License，GNU公共许可证）。GPL是一份闪耀着智慧光芒的法律许可文书，它允许代码能够不受限制地进行复制和修改，但要求衍生出来的软件产品（以及被修改过的副本）在发行时必须遵守与原始版本完全相同的许可证，不得增加任何限制性条款。本质上来讲，GPL把专利因素全部排除在外，收到了利用版权法来抵制版权的效果。

遗憾的是，斯托曼的GNU项目最终还是没有完全成功，但它的几个阶段性成果已经成为许多开源系统的基本元素，其中最为成功的包括GNU的C语言编译器（GCC）和GNU文本编辑器（Emacs）。虽然GNU操作系统没能完成，但与斯托曼志同道合的人越来越多。李纳斯·托沃兹（Linus Torvalds）在斯托曼及其追随者的前期工作基础上终于填补了这一空白，Linux操作系统在1991年诞生了。Linux已发展成为一种自由的Unix风格的操作系统，斯托曼的心愿总算实现了。如今，Linux已是世界上最流行、最成功的开源操作系统了。

Linux为什么如此流行

Linux是一种建立在开源模型上的Unix风格的操作系统。任何人都可以免费使用、传播和修改。Linux采用了一种非常保守的设计方案，精简了其内核部分的功能，因此很容易演化和完善。自1991年问世以来，Linux在全世界范围内得到了开发人员的广泛支持，人们一直在改善它的性能和可靠性。有些人甚至宣称Linux是发展最成熟的操作系统。正式发布以来，Linux已在全世界的服务器和工作站市场上赢得了巨大的份额。如今，人们提到Linux操作系统的时候都会把它当作最成功的开源代码成果来谈论。

自由软件运动在其发展过程中也曾遇到过问题。这里所说的“自由”（free）指的是人们可以自由地使用、修改和传播有关软件，不是“免费”或“没有成本”的意思（请大家体会“自由发言”和“免费啤酒”的区别）。为了解决这个问题，那些先驱们组建了OSI（Open Source Initiative，开源倡议），并决定采用“开源”这个短语来描述GPL许可证所蕴涵和维护的“自由”含义。[①]对此感兴趣的读者

① 需要注意的是斯托曼本人并不认可开源软件等同于自由软件。——编者注

可以到www.opensource.org网站上去看一看。

OSI组织的努力改变了自由软件运动。软件开发人员开始认识到自由软件并非不需要支付成本，而开放软件也只有懂得与他人合作的团体和个人才能合法使用。随着因特网的迅速普及，合作性社区已经成为了一个世界性的开发人员社区。这个世界性的开发人员社区确保斯托曼的初衷能够得以延续。

因此，开源软件是这样一种软件，它用许可证制度确保开发人员在参与某个合作性社区（该社区的根本目的是让高质量的软件能够繁荣发展）的开发时，有权自由地使用它并对其进行复制、修改和传播。“开源”并不意味着零成本，它实际上意味着任何人都能参与某个软件的开发，因而可以无须支付任何费用使用那个软件。事实上，许多开源系统都依托于某个组织，由该组织发布，而该组织会通过为相关软件提供收费的技术支持服务来维持运转。这种模式下，使用该软件的组织可以消除启动成本，大幅减少软件维护费用来降低信息技术方面的成本。

回想当初，斯托曼坚信软件厂商应该通过售后技术服务而不是专利权来谋取利润。为此，他与其他人通过自身的努力创建了一个软件乌托邦。现在，所有的开源系统都与斯托曼等人的基础性工作有着千丝万缕的联系。斯托曼等人的愿望，有好几个都已实现。GNU/Linux（本书后面将只写作“Linux”）的发展催生出了许多通过销售定制的Linux发行版本、并向Linux提供技术支持服务而取得成功（并且赢利）的公司，Red Hat和Slackware就是其中的代表。另一个例子是MySQL，它现在已是最为成功的开源数据库系统。

在当今世界，软件乌托邦能否从人们理想中的概念变成现实还是个未知数，但人们已经可以下载一整套系统软件和工具软件让一台个人电脑或商用电脑运转起来，而无需为软件本身付费。从操作系统到诸如数据库和Web服务器之类的服务器系统，再到办公软件，几乎所有门类的软件都有相应的免费版本可供任何人下载和使用。

1.1.1　为什么要使用开源软件

人们总会问到这样一个问题：为什么说使用开源软件是一个好主意？如果你想一劳永逸地堵住那些商业专利软件支持者的嘴，就必须为这个问题准备一个无可辩驳的答案。选用开源软件最重要的理由包括以下几点。

- 开源软件只需支付很少的费用，甚至无需支付任何费用就可以使用。这对非营利性的团体、大学和公共机构尤其重要——它们的预算总是在缩减，每年都需要用尽可能少的钱办尽可能多的事。
- 开源软件允许你根据自己的特定需求对之进行修改。
- 开源软件所遵守的许可证制度比商业软件许可证更加灵活。
- 开源软件往往比同类的商业专利软件更健壮（或者说经受过更多的测试）。
- 开源软件往往比同类的商业专利软件更加可靠和安全。

在决定选用开源软件的时候，强制或要求你给出以上这些理由的情况往往不多见，更常见的情况是你会遭到反对。我的意思是，商业专利软件的支持者（或者说开源运动的反对者）会驳斥这些言论，发表有关“为什么你不应当在开发中使用开源软件”的言论。我们先从商业专利软件的立场去看看不使用开源软件的一些较常见的理由，然后再从开源软件的立场去驳斥他们。

1. 论点1：商业专利软件可以激发更大的创造力

这一结论的主要论据是：绝大多数企业级的商业专利软件都提供API（Application Programming Interface，应用编程接口）以便开发人员对其功能进行扩展，使其更加灵活且更便于开发人员发挥创造力。

这句话有一部分是对的。API确实能方便开发人员对软件进行扩展，但它们往往也会阻止程序员给原始软件添加新的功能。这类API通常会把开发人员限制在一个沙箱环境里，这进一步限制了开发人员的创造力。

注解 *创建沙箱往往是为了限制程序员影响核心系统的能力。这么做的主要理由与安全性有关。API越开放，心怀叵测的开发人员就越有可能利用自己编写的恶意代码破坏系统或其中的数据。*

开源软件也支持并提供API，但更重要的是开源软件让开发人员能够直接查看核心系统源代码。事实上，他们不仅可以看到源代码，更可以自由地修改它（这在开源阵营里是一种受到鼓励的行为）！只要它不具备你需要的重要特性，或者你需要系统能够读写某种特定的格式，你就可以亲自动手去修改核心系统。从这一点看，开源软件要比商业专利软件更能激发开发人员的创造力。

2. 论点2：商业专利软件比开源软件更安全

这一结论的主要论据是：在当今这个以因特网为纽带紧密联系的社会里，企业在信息系统安全性方面的要求要比以往任何时候都迫切。商业专利软件生来就更加安全，因为销售这些软件的公司已经投入了较大的力量去保证自己的产品可以经受住数字侵略者的攻击。

尽管这句话很可能被贴在商业软件公司会议室的墙上，作为公司的口号，但这个目标的实现情况不见得像这些公司的广告里所吹嘘的那么好。就拿微软公司的服务器版Windows操作系统来说吧。有关统计数字表明，Windows操作系统的服务器版本在安全性方面比不上Linux。虽然微软已经建立了一个成功而高效的补丁系统来保证Windows用户免遭已知攻击手段的伤害，但为Windows打补丁已成为服务器日常维护工作的一部分，这一事实已足以让我们怀疑微软产品的安全性达不到可以让用户免遭攻击的水平。（有些人为此给出了这样一个说法：只要微软存在，就会有数字侵略者。）

Linux比Windows更安全的主要原因是参与Linux开发活动的开发人员遍布世界各地，他们一起努力让这个系统不会受到恶意攻击的伤害（术语称之为加固）。也就是说，来自世界各地的开发人员在共同加固着Linux操作系统。参与解决某个问题的开发人员越多，解决这个问题的办法就会越高明。这样一来，在Linux里发现的新漏洞会被非常迅速地堵上，从而把数字侵略者拒之门外。

与Linux的情况相比，微软公司负责加固Windows的程序员人数肯定要少得多，解决问题的思路当然也就少得多。这样一来，对Windows进行加固所需要的时间就会比Linux长得多。虽说不是所有的开源软件都能像Linux这样享受到这种“优待”，但它确实表明开源系统有能力从容地面对各种数字化威胁，并比同类商业专利软件更加安全。

3. 论点3：商业专利软件比开源软件经受过更多的测试

这一结论的主要论据是：软件公司卖的就是软件，而它们卖出的产品必须维持在一个高质量的水平才能吸引顾客来购买。开源软件没有这方面的压力，因而其测试环节不会像商业专利软件那么严格。

应该承认，这个论据非常有说服力。事实上，它说到了每一位信息技术采购人员的心坎里——这些人总认为花钱买来的东西要比不花钱就能得到的软件更加可靠且缺陷更少。可惜，这些人没有注意

到开源软件的一个重要概念。

开源软件是由来自世界各地的开发人员共同开发的，其中有许多人在扮演着“缺陷侦探”（测试人员）的角色，他们会为自己发现和报告的每一个缺陷感到自豪。有时候，开源软件公司会向发现bug的开发人员提供一些奖励。事实上，MySQL AB公司为那些在MySQL数据库系统里发现bug的开发人员支付的奖励相当可观。在我撰写本书的时候，任何人在MySQL AB公司的软件产品里发现bug，都可以从该公司获得一台苹果公司的iPod nano音乐播放器作为奖励。别说我没有告诉过你哟！

软件公司会招聘一些软件测试人员是事实，这些人也应该是他们这行里的佼佼者，但商业专利软件几乎都有一个既紧迫又不可更改的交付期限。设置这类期限的理由通常是为了保证重大战略版本的发布时机，或者是因为市场竞争的需要。在许多时候，这类期限会迫使软件厂商放松甚至放弃软件开发流程中的某些环节，而那往往会是整个流程中的最后一个环节——测试。稍微思考一下就会知道，测试人员接触软件的时间（测试时间）越少，他们能够找出的缺陷也就越少。

开源软件公司可以从世界各地的开发人员那里获得帮助和支持，而这意味着会有更多的人对它们的软件做更频繁的测试——换句话说，开源软件的测试强度要高于商业专利软件。

4. 论点4：商业专利软件比开源软件具备更多元的特性和更完备的功能

这一结论的主要论据是：商业专利数据库系统都是些非常完善和复杂的服务器系统，而开源系统的规模和复杂程度难以承担企业级海量数据的处理工作。

这句话对于一些模仿同类商业系统模仿得还不错的开源系统来说是正确的，但这句话绝不适用于像MySQL这样的数据库系统。与那些商业专利数据库系统相比，MySQL的早期版本是缺少了一些功能，但从5.0版开始，MySQL已经具备了在商业专利数据库系统里能够找到的所有高级功能了。

不仅如此，已经有许多事实可以证明，MySQL能够提供大企业在处理关键数据时所要求的可靠性、高性能和可扩展性。如果你想找一个典型，来证明开源系统也能像最好的商业专利数据库系统那样向用户提供全套功能的话，MySQL会是一个最佳的例子。

5. 论点5：商业专利软件公司的售后服务更好——因为它们有专人负责

这一结论的主要论据是：在购买软件系统的时候，卖方都会承诺该软件的开发商会帮助用户解决与该软件有关的问题；而开源系统根本不被谁所“拥有”，所以在遇到困难的时候很难找到帮手。

绝大多数开源软件都是由散布在世界各地的开发人员合作完成的。但目前的创业潮流是在坚持开源精神的基础上创办一家公司来有偿提供有关软件的技术支持和服务。事实上，绝大多数重要的开源产品都是在这种模式下推广的。比如说，MySQL AB公司拥有其MySQL产品的源代码。（MySQL的开源许可证的完整内容可以在网页www.mysql.com/company/legal/licensing/opensource-license.htm上查到。）MySQL AB公司对外提供多种技术支持服务，其中包括一周全天候服务、最快30分钟响应等。

开源软件的开发人员对疑难问题作出响应的速度往往要比商业软件的开发人员快很多。事实上，想直接与某个商业软件的开发人员对话几乎是不可能的事情。以微软公司为例，虽然该公司建立的规模庞大的技术支持机制能够满足几乎所有用户的需求，但如果你想与某位微软产品的开发人员直接沟通，必须先找对门路才行，这意味着你必须不厌其烦地与该公司各级技术支持部门联系，即使这样也不能保证一定能找到你想找的那位开发人员。

开源软件的开发人员以因特网作为他们的基本联络方式。因为他们几乎随时都在网上，所以他们很容易看到你在某个论坛或新闻组里发布的求助帖子。此外，像MySQL AB这样的开源公司还会主动地巡视自己的用户论坛，迅速对用户遇到的问题作出响应。