<<Git版本控制管理>>

图书基本信息

书名：<<Git版本控制管理>>

13位ISBN编号：9787564122607

10位ISBN编号：7564122609

出版时间：2010-6

出版时间：东南大学出版社

作者：罗力格

页数：310

前言

AudienceWhile some familiarity with revision control systems will be good background material,a reader who is not familiar with any other system will still be able to learn enoughabout basic Git operations to be productive in a short while. More advanced readersshould be able to gain insight into some of Git's internal design and thus master someof its more powerful techniques.The main intended audience for this book should be familiar and comfortable with theUnix shell, basic shell commands, and general programming concepts.Assumed FrameworkAlmost all examples and discussions in this book assume the reader has a Unix-likesystem with a command-line interface. The author developed these examples on Debian and Ubuntu Linux environments. The examples should work under other environments, such as Mac OS X or Solaris, but the reader can expect slight variations.A few examples require root access on machines where system operations are needed.Naturally, in such situations you should have a clear understanding of the responsibilities of root access. Book Layout and OmissionsThis book is organized as a progressive series of topics, each designed to build uponconcepts introduced earlier. The first 10 chapters focus on concepts and operationsthat pertain to one repository. They form the foundation for more complex operationson multiple repositories covered in the final six chapters.If you already have Git installed or have even used it briefly, you may not need theintroductory and installation information in the first two chapters, nor even the quicktour presented in the third chapter.

内容概要

Git可以支持几乎无数种开发与合作的方法。
它最早由LinilusTorvalds创建，用于管理Linux内核开发，现已成为分布式版本控制的主流工具。
但是Git的灵活性也意味着某些用户无法发挥它的最大价值。
《Git版本控制管理》提供了最高效的方法教程，其友好又严谨的建议有助于你随心操控Git的各项功能。

作者简介

作者：（美国）罗力格（Jon Loeliger）罗力格，是一位自由职业的软件工程师，致力于Linux、U-Boot和Git等开源项目。
他曾在Linux World等诸多会议上公开讲授Git，还为《Linux Magazine》撰写过数篇关于Git的文章。

书籍目录

章节摘录

插图：It's important to see Git as something more than a version control system: Git is acontent tracking system. This distinction, however subtle, guides much of the design ofGit and is perhaps the key reason Git can perform internal data manipulations withrelative ease. Yet this is also perhaps one of the most difficult concepts for new usersof Git to grasp, so some exposition is worthwhile.Git's content tracking is manifested in two critical ways that differ fundamentally fromalmost all other* revision control systems.First, Git's object store is based on the hashed computation of the contents of its objects,not on the file or directory names from the user's original file layout. Thus, when Gitplaces a file into the object store, it does so based on the hash of the data and not onthe name of the file. In fact, Git does not track file or directory names, which are asso-ciated with files in secondary ways. Again, Git tracks content instead of files.If two separate files located in two different directories have exactly the same content,Git stores a sole copy of that content as a blob within the object store. Git computesthe hash code of each file according solely to its content, determines that the files havethe same SHA1 values and thus the same content, and places the blob object in theobject store indexed by that SHA1 value. Both files in the project, regardless of wherethey are located in the user's directory structure, use that same object for content.If one of those files changes, Git computes a new SHA1 for it, determines that it is nowa different blob object, and adds the new blob to the object store. The original blobremains in the object store for the unchanged file to use.Second, Git's internal database efficiently stores every version of every file——not theirdifferences——as files go from one revision to the next. Because Git uses the hash of afile's complete content as the name for that file, it must operate on each complete copyof the file. It cannot base its work or its object store entries on only part of the file'scontent, nor on the differences between two revisions of that file.

媒体关注与评论

这是一本应该随身携带的书。
——Don Marti 编辑、记者以及会议主席

编辑推荐

《Git版本控制管理(影印版)》你将会：学习如何在多种真实开发环境中使用Git洞察Git的常用案例、初始任务和基本功能理解如何在集中和分布式版本控制中使用Git使用Git管理补丁、差异、合并和冲突获得诸如重新定义分支(rebasing)、钩子(hook)以及处理子模块(子项目)等的高级技巧学习如何结合使用Git与subversion

版权说明