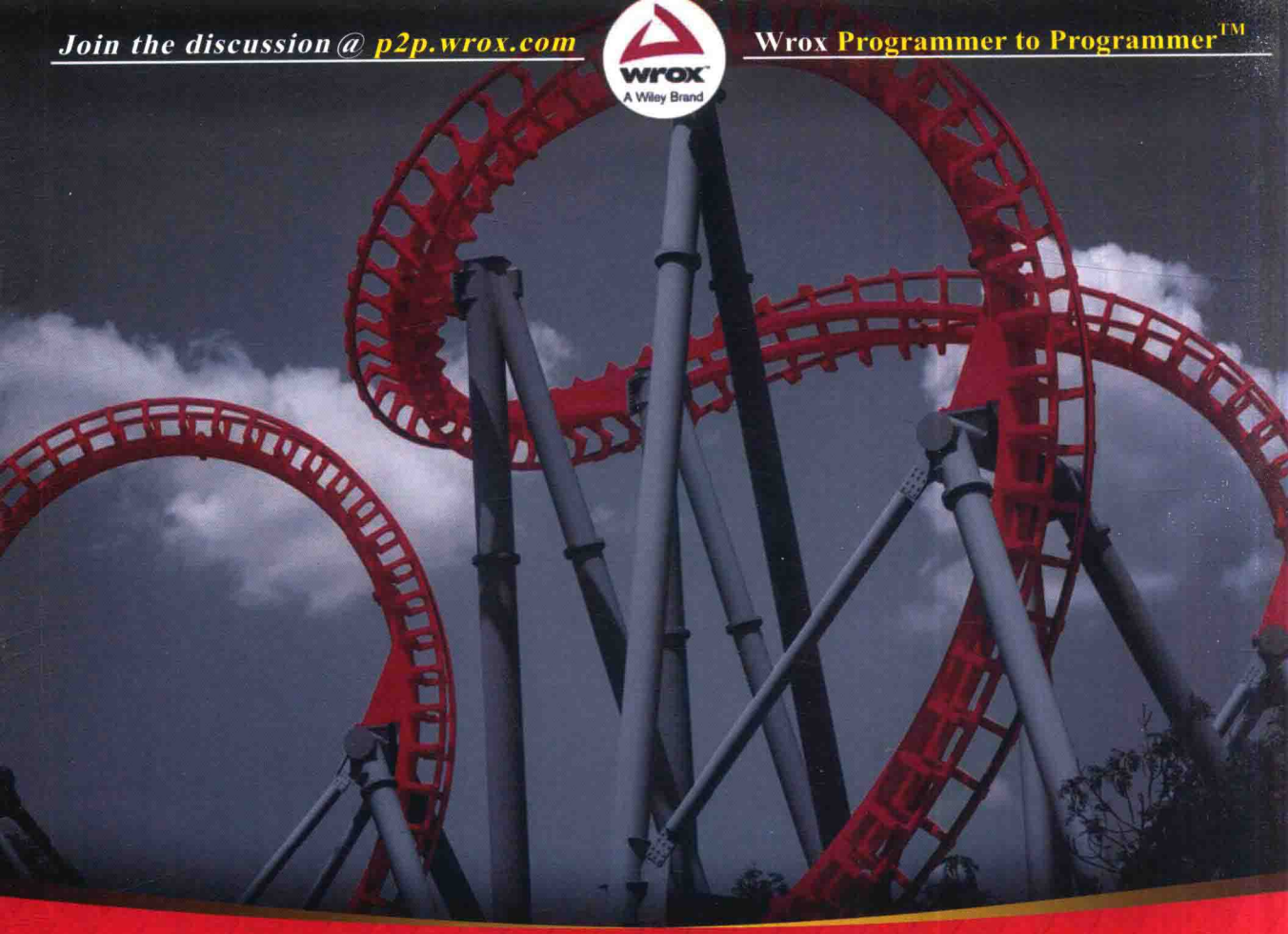
wrox
A Wiley Brand



Professional Git

Git软件开发实战

[美] Brent Laster 著
蒲 成 译

清华大学出版社

Git 软件开发实战

[美] Brent Laster 著
蒲 成 译

清华大学出版社
北 京

Brent Laster
Professional Git
EISBN：978-1-119-28497-0

北京市版权局著作权合同登记号 图字：01-2017-1721

图书在版编目(CIP)数据

Git 软件开发实战 / (美)布伦特 • 勒思特尔(Brent Laster) 著；蒲成 译. —北京：清华大学出版社，2017
书名原文：Professional Git
ISBN 978-7-302-47991-8

Ⅰ. ①G…　Ⅱ. ①布… ②蒲…　Ⅲ. ①软件工具—程序设计　Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 207690 号

责任编辑：王　军　李维杰
装帧设计：孔祥峰
责任校对：成凤进
责任印制：王静怡

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市君旺印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：26.5　　**字　　数**：678 千字
版　　次：2017 年 10 月第 1 版　　**印　　次**：2017 年 10 月第 1 次印刷
印　　数：1~3000
定　　价：79.80 元

产品编号：074900-01

译 者 序

凡是从事软件设计、开发、管理的人，必定都或多或少接触过源管理系统。市面上有很多源管理系统可供大家选择，比如Visual SourceSafe (VSS)、Source Anywhere、Concurrent Version System (CVS)、Subversion(SVN)、Mercurial(Hg)、Git等。可以说，现代软件系统的构建已经离不开这些源管理系统了。无论是从团队协作还是从个人开发来说，学习其中一款或多款源管理系统的使用方法都是现实工作中需要完成的任务。

作为目前最受欢迎的源管理系统之一，Git在全球范围内都得到了极为广泛的应用，相信就算是没有使用过Git的相关从业人员，也一定听说过它。尤其对于开源社区来说，Git几乎就是必不可少的存在。

Git最明显的特性就是分布式，并且它是免费、开源的。Git可以在不连接服务器端的情况下让用户在本地进行任意的内容版本控制，仅在必要的时候才建立与服务器的连接，这就使得源代码的发布和通信极其方便。Git的速度很快，这对于大型项目来说至关重要，并且其合并追踪能力非常出色。

本书从入门介绍开始，深入浅出地讲解了Git应用的完整生命周期。如果读者之前仅仅停留在Git使用层面，那么阅读完本书之后，相信会对Git的处理逻辑有一个全新的认识，并且对于过去使用过程中一知半解的问题和疑惑，在阅读完本书之后，相信你也将豁然开朗。

本书提供简单、清晰的图示来帮助厘清关键的概念和工作流处理机制。此外，除了对常用的基础命令和处理的介绍之外，本书的许多章节还提供高级主题方面的内容。这些内容提供额外的说明来阐释如何使用Git的一些较不为人知的特性以及如何应用这些特性，从而让Git能够发挥出最大的潜力。

在此要特别感谢清华大学出版社的编辑们，在本书翻译过程中他们提供了颇有助益的帮助，没有其热情付出，本书将难以付梓。

本书全部章节由蒲成翻译，参与翻译的还有何东武、李鹏、李文强、林超、刘洋洋、茆永锋、潘丽臣、王滨、陈世佳、申成龙、王佳、赵栋、潘勇、负书谦、杨达辉、赵永兰、郑斌、杨晔。

由于译者水平有限，难免会出现一些错误或翻译不准确的地方，如果有读者能够指出并勘正，译者将不胜感激。

译 者

作 者 简 介

Brent Laster是一位高级经理、软件开发人员，他供职于位于北卡罗来纳州卡瑞的SAS的研发部门。他管理参与发布工程设计流程和内部工具的几个小组。他还充当使用开源技术的资源的角色，并且负责进行Git、Gerrit、Gradle和Jenkins这样的技术培训课程，同时面向美国和美国之外的国家。

除了企业培训之外，Brent还为各种技术会议发起并提供专题研讨会。在像Rich Web Experience/Continuous Delivery Experience、ÜberConf、OSCON这样的会议上，他提供了关于开源技术(以及如何应用它们)的专题研讨会和咨询性研讨。他还是*No Fluff Just Stuff*这样杂志的出版物的撰稿者。Brent时不时地进行在线网络培训。

Brent的热情在于传授知识，并且是以让所有人都能理解的方式来讲解这些难以理解的概念。他参与技术培训已经超过25年了，并且在持续地寻求向其他人展示如何才能使用技术来简化和自动化工作的方法。

可以在Brent的LinkedIn页面http://linkedin.com/in/BrentLaster上或者通过Twitter账号@BrentCLaster来了解更多与他及其工作有关的内容。

技术编辑简介

Chaim Krause是来自位于堪萨斯州利文沃斯的U. S. Army的模拟专家。尽管他持有芝加哥大学政治学学士学位，但在计算机、编程和电子学方面，Chaim是一位自学成才者。他在一台TRS-80 Model I Level I上用BASIC语言编写了他的首款计算机游戏并且将该程序存储在一盒卡式磁带上。无线电这一业务爱好将他引入了电子学，而Arduino开发板和Raspberry Pi为他提供了将计算机技术、编程和电子学融合成一个爱好的媒介。

在空闲时间，他喜欢玩电脑游戏并且偶尔也开发自己的游戏。他最近忙于高尔夫运动以便花费更多的时间与其重要的另一半Ivana待在一起。

技术审校者简介

Philippe Charriere是GitHub的一位解决方案工程师。他也是与编程有关的许多技术活动的演讲者，其中包括JDD(波兰)、Devoxx France、SoftShake(日内瓦)以及Voxxed Days Luxembourg。他很早之前就开始用TI-99/4A(德州仪器)来编程了。在空余时间，他在开发IOT(物联网)解决方案。

致　谢

在编写本书的过程中，我一直对于发现有这么多具有奉献精神的人参与制作这类书籍而感到惊讶。John Wiley & Sons的团队非常优秀，对于他们的付出，我再怎么感谢都不为过。

首先，要感谢Jim Minatel承接了这个项目并且洞察到一本关于Git的新书的潜力。我很感激他对于这一项目的远见以及他的实践指导。还要感谢Adaobi Obi Tulton，她是一名让人惊讶的项目编辑，在我仔细推敲看似浩繁的细节时，她提供了非常好的指导方向以及建议，并且从没犯过错。本书能够成书，离不开她的执着管理和协助。感谢文字编辑Marylouise Wiack，是她让我所写的内容变得具有可读性并且变得清晰；感谢制作编辑Barath Kumar Rajasekaran；还要感谢校对员Nancy Bell，是她将所有内容拼接在一起从而创造出一本最终的、优质的书籍。

由衷感谢技术编辑Chaim Krause，他所付出的所有精力和时间确保了我的阐释、示例和实验课程是正确并且合理的。我要感谢他对于细节的密切关注，以及他基于自身经验所提的建议，正因如此才让本书变得对读者更为有用。还要感谢Philippe Charrière，谢谢他在短时间内就自愿成为本书的另一名技术审稿人。我要感谢他作为经验丰富的专家所提供的精力投入、意见和建议。

感谢SAS的管理层支持我多年来为整个公司的员工创建和提供企业培训课程的举措。我尤其要感谢Glenn Musial、Cyndi Schnupper和Andy Diggelmann，他们给予我鼓励和积极的反馈；感谢BarbaraMiller花费了数年时间来管理培训课程的日程规划和材料准备等所有这些后勤工作，因为这些课程已被扩展到如此之多的区域和国家；感谢我的同事Lee Greene，他帮助参与课程的讲解以及材料的评审。

在会议方面，非常感谢No Fluff Just Stuff系列会议的发起者和组织者Jay Zimmerman，他为我提供了在其遍及全国的活动中演讲的机会。能够参与到此类高质量的技术活动中来是非常棒的，这为所有人提供了有意义的信息和培训。

还要感谢出席我其中一堂培训课程或研讨会并且提出过问题或提供过反馈的每一个人。这对我而言是无价的，能够让内容变得更好并且更适用。

我在这里要匿名感谢一下我的四年级助教King女士，她在多年前告诉我应该成为一名作家。我不确定这本书就是她或我当时所认为的要写的那类书籍，不过我还是把这本书算在其中。

最后，最应该感谢的人必然是我的妻子Anne-Marie以及我的孩子们。本书编写了很长时间，大多数都是在夜间和周末编写的，这使得我没什么时间陪伴她们。无论如何，她们从未吝啬给予我鼓励的话语。Anne-Marie，你一直都是让我产生灵感的人、我的朋友以及这个项目期间我最大的支持者，尽管我知道它对于你来说有多么陌生。感谢你让我的每一天都很愉悦，并且让我们的生活充满惊喜，还要感谢你分享我的梦想，最重要的是，感谢你分享我的人生。

前　言

欢迎阅读《Git软件开发实战》。如果你的工作或兴趣涉及设计、创建或测试软件，或者涉及管理软件开发生命周期的任意部分，那么有可能你已经听说过Git了，并且在某种程度上，已经尝试过使用和理解它了。本书将帮助你达成该目标。简言之，《Git软件开发实战》旨在帮助你理解和使用Git来完成工作，无论该工作是个人项目还是专业需求。在这一过程中，它还会让Git变成你专业技能的一部分。在本书所有内容中，我已经提供了理解Git所需要知道的背景和概念，同时你将学习如何与之交互。

本部分将为你提供本书的简单介绍。它将阐释本书与其他关于Git的书籍相比的独特之处、本书的目标读者、本书的整体结构和内容，以及它为你提供的其中一些价值。

我鼓励你花几分钟时间阅读本前言。然后，你就能够以你自己的节奏来深入探究这些内容，并且通过文本内容和所包含的实践实验课程来提升你的技能以及理解Git。或者，如果你想要快速查看关于内容范围的额外信息，可以浏览目录。

感谢阅读《Git软件开发实战》。

本书独特之处

尽管市面上已经有了许多关于Git的书籍，但其中大多数都有针对性地将提供该应用程序的技术用途作为其主要且唯一的目标。《Git软件开发实战》会为你提供这方面的内容，但它也会为你提供对于你可能已经知晓的概念方面的Git理解。另外，大多数书籍都没有提供整合它们所描述的概念的实践方法。当你具有实际的示例可以着手处理以便能够消化这些概念并且以你自己的节奏掌握它们时，这样的学习才最有效。《Git软件开发实战》包含了连接实验课程，可以通过它们来吸收你刚刚阅读到的知识。

我已经纳入简单、清晰的图例来帮助你可视化关键的理念和工作流。我还在许多章节结尾处引入了“高级主题”小节。这些章节的内容提供了额外的说明来阐释如何使用Git的一些较不为人知的特性以及如何超越标准的Git特性来获得额外的价值。

如果不理解Git，那么很容易就会得到从另一个源管理系统迁移到Git的糟糕体验。为了更为有效，你需要理解Git模型和工作流。你还应该知道，在进行迁移时要注意些什么，以及为何很重要的是，不单单要考虑命令和工作流，还要考虑其基础仓库的结构和范围。我会在《Git软件开发实战》中介绍这方面的所有内容。

本书目标读者

这本书是以我多年来就Git对人们进行培训的经验为基础的；这些人的工作多种多样并且来自许多不同的背景环境——开发人员、测试人员、项目经理、团队管理者、文档编写专家等。我已经在行业会议的许多专题研讨会和企业培训会议上提供了本书中所概述的基本资料。我在美国各地以及各个国家都介绍过它们。我在帮助人们充满信心地使用Git方面已经取得了成功。

在本书中我只进行了一项假设前提：就是你已经至少具有了使用一个源管理系统的经验。使用的是哪一个并不重要：CVS、Subversion、Mercury——任何一个都行。我仅仅是在假设你基本理解源管理系统会做些什么以及像签入和签出代码与分支这样的基础概念。除此之外，你不需要任何预先掌握的知识或经验。并且，即便你具有使用Git或另一个系统的丰富经验，也会在这里找到一些有益的内容。实际上，如果你正在阅读这部分内容，那么你很可能可以被归入以下类别之一：

- 你才开始接触 Git 并且知道你需要学习它。
- 已经使用过 Git，但一直在尝试以使用上一个源控制系统的相同方式来使用它。
- 你已经使用过 Git，并且认为你知道“一知半解很危险”。
- 你正在掌握 Git 的用法，但切实希望理解为何它是这样运行的以及如何真正像预期那样使用它。
- 你在与使用 Git 或需要学习它的人一起工作或者管理这些人。鉴于这一关系，你需要了解 Git 并且理解基础概念。
- 你已经听说过 Git 的潜在好处，因而你对它及其能为你和你为之工作的组织做些什么感兴趣。

你可能实际上会认为你自己归属于这些类别中的多个。不过，你大概仅希望能够完成你的工作(无论该工作是个人目标还是专业目标)。这本书正是基于这一前提来编写的。

Git需要转换思维模式。实际上，它需要一系列思维模式的转换。不过，一旦可以将它与已经知晓的一些事情关联上，那么每一种转换都很容易理解。理解每一种转换将反过来使你变得更具生产力并且掌控这一强大工具的特性——这也就是本书的目的所在。

本书结构和内容

本书被组织为一系列从头开始介绍Git的文章，向你介绍在添加新概念之前，要成为专家所需要知道的并且以之为基础的内容。

在前三章中，我介绍了Git的基础概念：它与其他系统有多大区别、围绕它构建的生态系统、其优势和挑战，以及让你可以理解其工作流并且使用它有效管理内容的模型。本部分将让你基本理解Git的理念、目标以及必要的专业术语。

在本书的其余章节中，我将介绍Git的使用和特性，从执行基本操作到创建仓库以及将变更提交到这些仓库中，一直到创建分支、进行合并以及处理公共仓库中的内容。

注意，我没有立即让你使用Git(如果你希望这样做，可以直接跳到第4章，它会快速让你开始亲身实践Git)。不过，我强烈建议阅读前三章。如果你刚开始接触Git(或者已经使用了一

段时间)，那么其背景阅读，尤其是第2和第3章中的背景阅读，将提供理解其余章节内容所需的基础。并且即使你之前使用过Git，阅读这些章节也可以让你弄明白你曾经就Git使用所面临的问题，还会为你提供一个更好的思维模型以便开展工作，并且让你形成理解其中一些更为高级的概念的基础。

读者将获得的价值

在整本书的内容中，你都会找到与高效使用Git所需的命令和工作流有关的示例和指导。每一章都包含了将概念与你已经知晓并且理解的内容关联起来的几种方式。除了文本之外，你还将发现许多图例有助于你可视化理解概念。正如我已经提到过的，这本书还通过散布在各个章节的连接实验课程来增加允许你亲身实践Git的特性。这些实验课程旨在强化前面章节文本内容中提到的概念，并且让你主动参与到学习过程中，使得你可以更好地领会这些概念。为了从本书中获得最大收益，你应该花些时间来完成每一个实验课程——通常仅需要几分钟时间。你将发现，这些简单的步骤会极大地提升你在使用Git时的整体理解和信心。

另外，建议你阅读一下位于一些章节结尾处的“高级主题”部分。你会发现之前可能没有考虑到的利用Git功能的各种方式的说明和理念，或者会弄明白如何使用你一直存有疑虑的特性。

对于后续的实验课程，http://github.com/professional-git网站上为用户提供了具有样本内容的自定义Git仓库。此外，可以在http://github.com/professional-git/hooks上找到最后一章中各个挂钩(hook)的代码的可下载副本。如果GitHub不可用，那么可以在www.wrox.com/go/professionalgit处找到所需的文件。读者也可以通过手机，扫描封底的二维码来下载这些文件。

后续步骤

如果本书听上去正是你所需的，那么我建议持续阅读并且开始融会贯通本书的内容以及进行思维模式的转换，它们将有助于你成功地使用Git。在阅读本书的过程中，你将发现许多理念、见解，还有让你会心一笑的时刻，这些将让你更好地理解Git。具备了这些知识，不久之后你就会成为Git专家。

目　录

第 I 部分　理解 Git 相关概念

第 1 章　什么是 Git……3
1.1　Git 的历史……4
1.2　行业标准工具……4
1.3　Git 生态系统……5
1.3.1　核心的 Git……5
1.3.2　Git 托管站点……6
1.3.3　自托管软件包……7
1.3.4　易用的包……7
1.3.5　插件……9
1.3.6　包含 Git 的工具……9
1.3.7　Git 库……9
1.4　Git 的优势和挑战……10
1.4.1　优势……10
1.4.2　挑战……12
1.5　本章小结……15

第 2 章　关键概念……17
2.1　设计概念：面向用户的……17
2.1.1　集中式模型……17
2.1.2　分布式模型……18
2.2　设计概念：内部的……19
2.2.1　差异增量存储……20
2.2.2　快照存储……20
2.2.3　Git 的存储需求……21
2.3　仓库设计注意事项……22
2.3.1　仓库范围……23
2.3.2　文件范围……24
2.3.3　共享代码……25
2.4　本章小结……26

第 3 章　Git 升级模型……27
3.1　Git 的级别……27
3.1.1　开发-测试-生产和 Git……27
3.1.2　移动内容的核心 Git 命令……34
3.2　本章小结……36
3.3　关于连接实验课程 1：安装 Git……36

连接实验课程 1：安装 Git……37

第 II 部分　使用 Git

第 4 章　配置和设置……43
4.1　在 Git 中执行命令……43
4.1.1　操作数类型……44
4.1.2　高层命令和底层命令的对比……45
4.1.3　指定参数……47
4.1.4　自动完成……47
4.2　配置 Git……48
4.2.1　告知 Git 你的身份……49
4.2.2　配置范围……50
4.2.3　默认的编辑器……53
4.2.4　设置行结束符……54
4.2.5　别名……55
4.2.6　Windows 文件系统缓存……56
4.3　初始化仓库……56
4.3.1　git init……56
4.3.2　git clone……57
4.4　高级主题……58
4.4.1　git init 揭秘……58
4.4.2　进一步深入了解 Git 仓库……59
4.4.3　将 config 命令映射到配置文件……60
4.4.4　创建参数化别名……61
4.5　本章小结……63

第 5 章 变得高效……65
5.1 获得帮助……65
5.2 多仓库模型……67
5.3 添加内容以便追踪——add……69
5.4 完成变更——提交……77
5.4.1 先决条件……78
5.4.2 提交范围……79
5.5 将一切结合在一起……79
5.5.1 修正提交……80
5.5.2 提交的结果……82
5.6 高级主题……84
5.6.1 使用--verbose 选项……85
5.6.2 完整的消息提交过程……85
5.6.3 自动更正和自动执行……86
5.7 本章小结……87
5.8 关于连接实验课程 2：创建和探究 Git 仓库并且管理内容……87

连接实验课程 2：创建和探究 Git 仓库并且管理内容……89

第 6 章 追踪变更……95
6.1 git status……95
6.1.1 具有状态的工作流示例……96
6.1.2 status 命令的简要形式……100
6.2 git diff……102
6.2.1 Git 中的重要符号名称……102
6.2.2 如何思考 Git 进行对比的方法……102
6.2.3 仅显示有差异的文件名称……107
6.2.4 word-diff……107
6.2.5 忽略非关键变更……107
6.2.6 对比两次提交……109
6.2.7 可视化对比……111
6.2.8 其他的对比技巧……114
6.3 本章小结……115
6.4 连接实验课程 3：通过文件状态生命周期追踪内容……116

连接实验课程 3：通过文件状态生命周期追踪内容……117

第 7 章 处理随时间推移而出现的变更以及使用标签……121
7.1 log 命令……121
7.1.1 常用的显示和过滤选项……122
7.1.2 时间限制选项……123
7.1.3 按文件和路径显示历史……124
7.1.4 日志输出格式……125
7.1.5 搜索历史……126
7.2 git blame……127
7.3 可视化地查看历史……130
7.4 标签……131
7.4.1 查看标签详情……132
7.4.2 修改标签……132
7.4.3 简单的标签示例……133
7.5 撤消历史中的变更……134
7.5.1 reset——回滚变更……134
7.5.2 完全重置本地环境……135
7.5.3 revert——消除变更……136
7.6 高级主题……139
7.6.1 签署提交和标签……139
7.6.2 引用日志……141
7.7 本章小结……143
7.8 关于连接实验课程 4：使用 Git 历史、标签和别名……143

连接实验课程 4：使用 Git 历史、标签和别名……145

第 8 章 处理本地分支……149
8.1 什么是分支？……149
8.1.1 来自另一个源管理系统的示例……150
8.1.2 分支的 Git 模型……150
8.1.3 创建一个分支……151
8.1.4 签出一个分支……152
8.1.5 将内容添加到分支……153

8.1.6 一个工作目录——多个分支……154
8.1.7 获得关于分支的信息……157
8.1.8 删除或重命名一个分支……158
8.1.9 使用分支进行开发……161
8.2 高级主题……167
8.2.1 签出非分支提交……168
8.2.2 签出单独的文件……173
8.3 本章小结……174
8.4 连接实验课程 5：处理分支……174

连接实验课程 5：处理分支……175

第 9 章 合并内容……177
9.1 合并的基础……177
9.1.1 merge 命令……178
9.1.2 为合并做准备……178
9.1.3 合并的类型……178
9.1.4 变基——合并历史……182
9.1.5 樱桃拣选……185
9.1.6 樱桃拣选和变基之间的差异……188
9.1.7 合并操作……189
9.2 处理冲突……192
9.2.1 合并处理是一种状态……192
9.2.2 冲突的错误消息……193
9.2.3 终止操作……194
9.2.4 处理冲突——工作流……194
9.2.5 解决选项和策略……198
9.3 可视化合并……203
9.3.1 选择一款合并工具……205
9.3.2 让合并工具可供 Git 使用…206
9.4 高级主题……206
9.4.1 用于冲突标记的可选样式……207
9.4.2 高级变基场景……208
9.4.3 交互式变基……213
9.5 本章小结……220
9.6 连接实验课程 6：合并实践…220

连接实验课程 6：合并实践……221

第 10 章 Git 中的支持文件……223
10.1 Git 属性文件……224
10.1.1 Git 属性文件的作用……224
10.1.2 Git 属性的范围……224
10.1.3 文件格式……226
10.1.4 常见用例……226
10.1.5 获取文件的属性信息……232
10.2 Git 忽略文件……233
10.2.1 Git 忽略的范围……233
10.2.2 文件格式……234
10.2.3 获取文件的忽略信息……236
10.3 本章小结……236

第 11 章 用 Git 做更多的事情……239
11.1 修改本地环境中文件和目录的布局……239
11.1.1 储藏……240
11.1.2 暂存命令……246
11.1.3 mv……246
11.1.4 rm……247
11.2 用于搜索的命令……248
11.2.1 grep……248
11.2.2 Git 日志搜索……251
11.3 为变更处理补丁和存档……253
11.3.1 archive……254
11.3.2 bundle……254
11.3.3 通过电子邮件共享补丁……256
11.3.4 apply……257
11.3.5 am……257
11.3.6 用邮件发送补丁文件……258
11.4 用于清理的命令……260
11.4.1 clean……260
11.4.2 gc……262
11.4.3 notes……263
11.5 高级主题……265
11.5.1 filter-branch……265

11.5.2 rev-list ····· 265
11.5.3 bisect ····· 272
11.5.4 rerere ····· 279
11.6 本章小结 ····· 287
11.7 连接实验课程 7：删除、重命名和储藏 ····· 287

连接实验课程 7：删除、重命名和储藏 ····· 289

第 12 章 理解远程——分支和操作 ··· 293
12.1 远程 ····· 293
12.1.1 远程访问协议 ····· 295
12.1.2 remote 命令 ····· 297
12.1.3 Git 如何与远程环境交互 ····· 299
12.1.4 远程追踪分支 ····· 299
12.1.5 git clone ····· 300
12.1.6 查看关于远程分支的信息 ····· 303
12.1.7 push ····· 307
12.1.8 fetch ····· 315
12.1.9 pull ····· 317
12.2 本章小结 ····· 320
12.3 连接实验课程 8：设置 GitHub 账户并且克隆仓库 ··· 320

连接实验课程 8：设置 GitHub 账户并且克隆仓库 ····· 321

第 13 章 理解远程——变更工作流 ··· 323
13.1 Git 中基本的冲突以及合并解决工作流 ····· 324
13.2 托管仓库 ····· 336
13.2.1 用于使用 Git 进行协作的模型 ····· 336
13.2.2 对所有内容进行汇总 ····· 340
13.3 本章小结 ····· 341
13.4 连接实验课程 9：对远程仓库使用整体工作流 ····· 342

连接实验课程 9：对远程仓库使用整体工作流 ····· 343

第 14 章 处理 Git 中的树和模块 ····· 347
14.1 工作树 ····· 347
14.1.1 添加一棵工作树 ····· 348
14.1.2 列出工作树 ····· 350
14.1.3 精简工作树 ····· 351
14.2 子模块 ····· 352
14.2.1 理解子模块如何工作 ····· 353
14.2.2 添加子模块 ····· 353
14.2.3 判定子模块状态 ····· 355
14.2.4 处理多个子模块 ····· 359
14.2.5 从子模块的远程更新子模块 ····· 359
14.2.6 查看子模块差异 ····· 361
14.2.7 超级项目与子模块对比 ····· 362
14.2.8 子模块的问题 ····· 364
14.2.9 更新子模块引用 ····· 365
14.2.10 在超级项目被更新后更新子模块 ····· 366
14.2.11 推送来自子模块的变更 ····· 368
14.2.12 子模块与合并 ····· 369
14.2.13 注销一个子模块 ····· 370
14.3 子树 ····· 370
14.3.1 添加一个项目作为子树 ····· 371
14.3.2 更新子树 ····· 374
14.3.3 使用子树分隔功能 ····· 374
14.3.4 从分隔内容中创建一个新的项目 ····· 375
14.3.5 子树推送 ····· 376
14.4 本章小结 ····· 376
14.5 连接实验课程 ····· 376
14.5.1 连接实验课程 10：使用工作树 ····· 376

14.5.2 连接实验课程 11：使用子模块 ····· 376
14.5.3 连接实验课程 12：使用子树 ····· 377
连接实验课程 10：使用工作树 ····· 379
连接实验课程 11：使用子模块 ····· 381
连接实验课程 12：使用子树 ····· 385
第 15 章 使用 Git 挂钩程序扩展 Git 功能 ····· 389
15.1 安装挂钩 ····· 389
15.2 更新挂钩 ····· 390
15.3 常用挂钩属性 ····· 391
15.3.1 挂钩域 ····· 391
15.3.2 控制工作流的返回码 ····· 391
15.3.3 工作目录访问 ····· 392
15.3.4 环境变量 ····· 392
15.4 挂钩描述 ····· 392
15.4.1 applypatch-msg ····· 393
15.4.2 pre-applypatch ····· 393
15.4.3 post-applypatch ····· 393
15.4.4 pre-commit ····· 393
15.4.5 prepare-commit-msg ····· 395
15.4.6 commit-message ····· 396
15.4.7 post-commit ····· 398
15.4.8 pre-rebase ····· 399
15.4.9 post-checkout ····· 399
15.4.10 post-merge ····· 399
15.4.11 pre-push ····· 400
15.4.12 pre-receive ····· 400
15.4.13 update ····· 401
15.4.14 post-receive ····· 401
15.4.15 post-update ····· 402
15.5 其他挂钩 ····· 403
15.5.1 push-to-checkout ····· 403
15.5.2 pre-auto-gc ····· 403
15.5.3 post-rewrite ····· 403
15.6 挂钩快速参考 ····· 403
15.7 本章小结 ····· 405

第 I 部分

理解Git相关概念

- 第 1 章：什么是 Git
- 第 2 章：关键概念
- 第 3 章：Git 升级模型
- 连接实验课程 1：安装 Git