

## <<Git权威指南>>

### 图书基本信息

书名：<<Git权威指南>>

13位ISBN编号：9787111349679

10位ISBN编号：7111349679

出版时间：2011-6

出版时间：机械工业出版社华章公司

作者：蒋鑫

页数：595

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>

## &lt;&lt;Git权威指南&gt;&gt;

## 前言

版本控制是管理数据变更的艺术，无论数据变更是来自同一个人，还是来自不同的人（一个团队）。版本控制系统不但要忠实地记录数据的每一次变更，还要能够帮助还原任何一次历史变更，以及实现团队的协同工作等。

Git 就是版本控制系统中的佼佼者。

我对版本控制系统的兴趣源自于我的个人知识管理实践，其核心就是撰写可维护的文档，并保存于版本控制系统中。

可维护文档的格式可以是 DocBook、FreeMind、reStructuredText 等。

我甚至还对 FreeMind 加以改造以便让其文档格式更适合于版本控制系统，这就是我的第一个开源实践：托管于 SourceForge 上的 FreeMind-MMX 项目。

文档书写格式的问题解决之后，就是文档的存储问题了。

通过版本控制系统，很自然地就可以实现对文档历史版本的保存，但是如何避免因为版本控制系统瘫痪而导致数据丢失呢？

Git 用其崭新的分布式的版本控制设计提供了最好的解决方案。

使用 Git，我的知识库不再只有唯一的版本库与之对应，而是可以通过克隆操作分发到不同的磁盘或主机上，克隆的版本库之间通过推送（PUSH）和拉回（PULL）等操作进行同步，数据安全得到了极大的提升。

在版本控制系统的忠实呵护下，我的知识库中关于 Git 的 FreeMind 脑图在日积月累中变得越来越翔实，越来越清晰，最终成为本书的雏形。

版本控制能决定项目的成败，甚至是公司的生死，此言不虚。

我在推广开源项目管理工具和为企业提供咨询服务的过程中看到，有很多团队因为版本控制系统管理的混乱导致项目延期、修正的 Bug 重现、客户的问题不能在代码中定位……无论他们使用的是什么版本控制系统（开源的或是商业的）都是如此。

这是因为传统的集中式版本控制系统不能有效地管理分支和进行分支间合并。

集中管理的版本库只有唯一的分支命名空间，需要专人管理，从而造成分支创建的不自由；分支间的合并要么因为缺乏追踪导致重复合并、引发严重冲突，要么因为版本控制系统本身蹩脚的设计导致分支合并时效率低下和陷阱重重。

Git 凭借其灵活的设计让项目摆脱分支管理的梦魇。

我的公司也经历过代码管理的生死考验。

因为公司的开发模式主要是基于开源软件的二次开发，所以最早在使用 SVN（Subversion）做版本控制时，很自然地使用了 SVN 卖主分支模型来管理代码。

随着增加和修改的代码越来越多，我们开发的软件与开源软件上游的偏离也越来越远，当上游有新版本发布时，最早可能只用几个小时就可以将改动迁移过去，但是如果对上游的改动多达几十甚至上百处时，迁移的过程就会异常痛苦，基本上和重新做一遍差不多。

那时似乎只有一种选择：不再与上游合并，不再追踪上游的改动，而这与公司的价值观“发动全球智慧为客户创造价值”相违背。

迷茫之中，分布式版本控制系统飘然而至，原来版本控制还可以这么做。

我最先尝试的分布式版本控制系统是 Hg（Mercurial），当发现 Hg 和 MQ（Hg 的一个插件）这一对宝贝儿的时候，我如获至宝。

逐渐地，公司的版本库都迁移到了 Hg 上。

但随着新的开发人员的加入，问题又出现了，一个人使用 Hg 和 MQ 很好，但多个人使用时则会出现难以协同的问题。

于是我们大胆地采用了 Git，并在实践中结合 Topgit 等工具进行代码的管理。

再一次，也许是最后一次，我们的代码库迁移到了 Git。

最早认识分布式版本控制，源自于我们看到了众多开源项目的版本控制系统大迁移，这场迁移还在进行中。

## <<Git权威指南>>

MoinMoin 是我们关注的一个开源的维基软件，2006 年，它的代码库从SVN迁移到了Hg。

Mailman 同样是我们关注的一个开源邮件列表软件。

2007 年，它的代码库从SVN迁移到了 Bazaar。

Linux 采用Git作为版本控制系统（一点都不奇怪，因为Git就是 Linus Torvalds 开发的）。

Android 是目前最为流行的开源项目之一，因为潜在市场巨大，已经吸引了越来越多的开发者进入这个市场，而Android就是用Git维护的。

当开源软件纷纷倒向分布式版本控制系统大旗（尤其是Git）的时候，很多商业公司也在行动了，尤其是涉及异地团队协作和Android核心代码定制开发的公司。

对于那些因保守而不敢向Git靠拢的公司，Git也可以派上用场，因为Git可以与现在大多数公司部署的SVN很好地协同，即公司的服务器是SVN，开发者的客户端则使用 Git。

相信随着Git的普及，以及公司在代码管理观念上的改进，会有更多的公司拥抱 Git。

## <<Git权威指南>>

### 内容概要

《Git权威指南》是Git领域的集大成之作，是一本关于Git的百科全书，在广度、深度和实战性上让同类作品望尘莫及。

作者是国内顶尖的版本控制专家和咨询顾问之一，本书得到了Git官方维护者Junio C Hamano和ITeye创始人范凯（Robbin）先生等数位专家的高度认可和极力推荐，权威性毋庸置疑。

全书一共9篇，共41章和4个附录，内容几乎涵盖了Git的所有方面。

第1篇介绍了版本控制工具的演变历史、Git的各种优点，以及它在3种主流操作系统中的安装与配置。

第2篇和第3篇既是本书的基础，又是本书的核心，不仅介绍了Git的操作和使用，而且还讲解了Git的原理。

第2篇详细讲解了个人用户如何使用Git，包括Git初始化、日常操作、暂存区、对象、重置、检出、恢复进度、历史变更、克隆、库管理等；第3篇详细讲解了Git协议和团队如何使用Git，包括Git支持的协议、冲突解决、里程碑、分支、远程版本库和补丁文件交互等。

第4篇全面介绍了Git的协同模型，即它在实际工作中的使用模式，包括各种经典的Git协同模型

、Topgit协同模型、子模组协同模型、子树合并、Android多版本库协同、Git与SVN协同模型等。

第5篇介绍了Git服务器的架设，首先讲解了HTTP协议、Git协议、SSH协议的使用，然后讲解了Gitolite、Gitis、Gerrit等服务器的架设方法，最后还讲解了Git版本库的托管。

第6篇介绍了版本库的迁移，包括如何从CVS、SVN、Hg等版本库迁移到Git，以及Git版本库整理等方面的内容。

第7篇讲解了Git的其他应用，包括etckeeper、Gistore等的安装、配置和使用，以及补丁中的二进制文件和云存储等内容。

第8篇介绍了Git的跨平台操作，以及它的钩子和模板、稀疏检出和浅克隆、嫁接和替换等重要特性。

第9篇是附录，详细给出了Git的命令索引，以及CVS、SVN和Hg与Git的比较与命令对照，方便读者查阅。

## <<Git权威指南>>

### 作者简介

蒋鑫，国内顶尖的版本控制专家和咨询顾问之一，对Subversion和Git等版本控制工具有十分深入的研究，参与了Git以及Gitosis、Gitolite、Repo、Topgit、Gistore等与Git相关的开源软件的开发或创建，在大量实践中积累了丰富的经验。

此外，他还是一位开源软件实践者，作为北京群英汇信息技术有限公司的创始人兼高级顾问，一直从事开源软件的定制以及面向研发团队的项目管理软件的推广和顾问咨询工作，致力于推动开源软件在中国的发展。

## <<Git权威指南>>

### 书籍目录

#### 前言

#### 第1篇 初识Git

第1章 版本控制的前世和今生

第2章 爱上Git的理由

第3章 Git的安装和使用

#### 第2篇 Git独奏

第4章 Git初始化

第5章 Git暂存区

第6章 Git对象

第7章 Git重置

第8章 Git检出

第9章 恢复进度

第10章 Git基本操作

第11章 历史穿梭

第12章 改变历史

第13章 Git克隆

第14章 Git库管理

#### 第3篇 Git和声

第15章 Git协议与工作协同

第16章 冲突解决

第17章 Git里程碑

第18章 Git分支

第19章 远程版本库

第20章 补丁文件交互

#### 第4篇 Git协同模型

第21章 经典Git协同模型

第22章 Topgit协同模型

第23章 子模组协同模型

第24章 子树合并

第25章 Android式多版本库协同

第26章 Git和SVN协同模型

#### 第5篇 搭建Git服务器

第27章 使用http协议

第28章 使用Git协议

第29章 使用SSH协议

第30章 Gitolite服务架设

第31章 Gitis服务架设

第32章 Gerrit代码审核服务器

第33章 Git版本库托管

#### 第6篇 迁移到Git

第34章 CVS版本库到Git的迁移

第35章 更多版本控制系统的迁移

#### 第7篇 Git的其他应用

第36章 etckeeper

第37章 Gistore

## <<Git权威指南>>

第38章 补丁中的二进制文件

第39章 云存储

第8篇 Git杂谈

第40章 跨平台操作

第41章 Git的其他特性

第9篇 附录

附录

## &lt;&lt;Git权威指南&gt;&gt;

## 媒体关注与评论

2009年9月，我出版了一本针对日本读者的Git专著，当Linus收到我赠送的签名本时，他对我说：“除了截图和命令行示例外，其他我什么也看不懂”（Linus不懂日文）。

因为同样的原因，虽然我不能了解蒋鑫这本书的全部内容，但是我可以看出这本书涵盖了非常广泛的主题，并且可以看出蒋鑫对这本书的用心。

我非常高兴能够看到这本书的出版，感谢向世界传播Git。

——Junio C Hamano Git维护者（2005年7月至今）仔细拜读了本书前三篇共20章的内容，感觉这本书极好。

作者在软件版本控制系统方面有超过10年的经验，对版本控制系统有非常深入的认识。

尤为难得的是，本书文笔很流畅，虽然是技术书籍，但是作者娓娓道来，阅读体验很好。

Git的学习门槛较高，包括我们公司在内的很多企业都将版本控制系统转向了Git，强烈推荐大家看一看。

——范凯（Robbin）CSDN平台开发总监/ITeye（[www.iteye.com](http://www.iteye.com)）创始人这是我读过的最好的关于Git的书。

将复杂的Git解释得清晰而透彻绝非易事，蒋鑫做到了，更让人惊喜的是，他还分享了大量的经验总结。

我几年来累积下来的诸多疑惑都在读罢该书后一一得以解开。

如果你正在使用，或者打算使用Git，本书当然是必备的。

你也可以抱着Subversion或CVS不放，不过，如果哪一天有人拿起这本书敲你的头时可别怪我没提醒过你。

——许晓斌（Juven Xu）资深Maven专家/著有畅销书《Maven实战》Git是当前开源社区最流行的版本控制系统，代表了版本控制的未来。

每一位有志于从事软件开发的学习者都应该学习和掌握这一工具，它不但可用于追踪编程者的思考过程，还是打开开源软件世界的金钥匙，更可以通过版本控制掌握与他人协同工作的技能。

本书是一本耐读的原创著作，因为其涵盖的内容之广足以让每一位程序员在成长的不同阶段都可以从中汲取丰富营养。

——冯铃 清华大学计算机科学与技术系“长江学者”兼特聘教授目前市面上关于Git的书不在少数，但迄今为止，国内外还没有一本书能在广度和深度上达到本书的水平，也许在今后相当长的一段时间内也不会有。

如果你是一位软件开发者，强烈建议你学习并使用Git；如果你要学习Git，本书无疑是你最佳的选择。

——徐继哲 哲思社区（<http://www.zeuux.org/>）创始人



## <<Git权威指南>>

### 编辑推荐

《Git权威指南》编辑推荐：Git领域的集大成之作，在广度、深度和实战性上均史无前例；国内顶级Git专家亲自撰写，Git官方维护者等数位专家联袂推荐。

## <<Git权威指南>>

### 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>