Git tutorial

LeeMoo

2015 年 5 月 20 日

Contents

0.0.1

Chapter 1

起步

1.1 关于版本控制

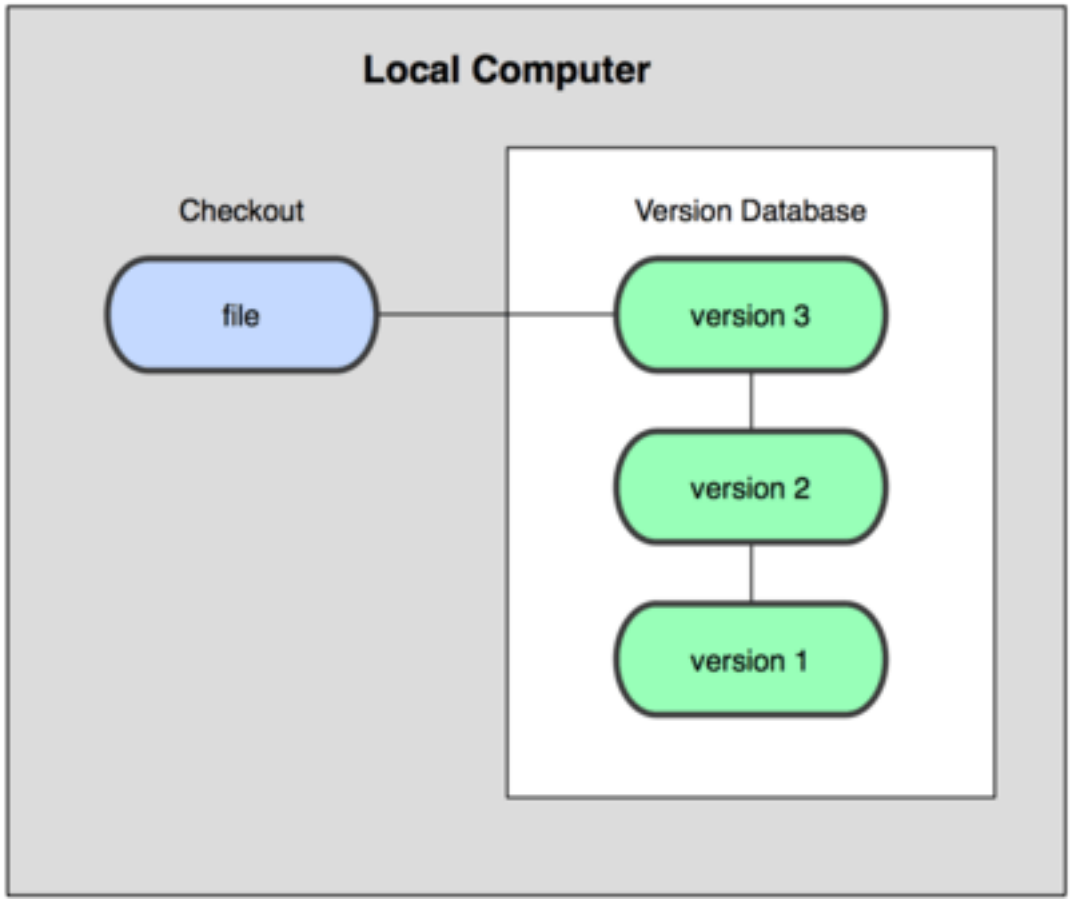
1.1.1 关于版本控制

什么是版本控制? 我为什么要关心它呢? 版本控制是一种记录一个或若干个文件内容变化, 以便将来查阅特定版本修订情况的系统。在配中所展示的例子中, 我们仅对保存着软件源代码的广西文件作版本控制管理, 但实际上, 你可以对任何类型的文件进行版本控制。
如果你是位图形或网页设计师, 可能会需要保存某一幅图片或页面布局文件的所有修订版本 (这或许是你非常渴望拥有的功能)。采用版本控制系统 (VCS) 是个明智的选择。有了它你就可以将某个文件回溯到之前的状态, 你可以比较文件的变化细节, 查出最后是谁修改了哪个地方, 从而找出导致怪异问题的原因, 又是谁在何时报告了某个功能缺陷等等. 使用版本控制系统通常是意味着, 就算你乱来一气把整个项目的文件改的改删的删, 你也照样可以轻松恢复到原先的样子. 但额外增加的工作量却微乎其微。

1.1.2 本地版本控制系统

许多人习惯用复制整个项目目录的方式来保存不同的版本, 或许还会改名加上备份的时间以示区别. 这么做唯一的好处就是简单. 不过坏处也不少: 有时候会混淆所在的工作目录, 一旦弄错文件丢了数据就没法撤销恢复.
为了解决这个问题, 人们很久以前就开发了许多种本地版本控制系统, 大多都是

采用某种简单的数据库来记录文件的历次更新差异 (如图 1-1)。



1.2 Git 简史

1.3 Git 基础

1.4 安装 Git

1.5 初次运行 Git 前的配置

1.6 获取帮助

1.7 小结

Chapter 2

Git 基础

Chapter 3

Git 分支

Chapter 4

服务器上的 Git

Chapter 5

分布式 Git

Chapter 6

Git 工具

Chapter 7

自定义 Git

7.1 配置 Git

7.2 Git 属性

7.3 Git 挂钩

7.4 Git 强制策略实例

7.5 总结

Chapter 8

Git 与其他系统

8.1 Git 与 Subversion

8.2 迁移到 Git

8.3 总结

Chapter 9

git 内部原理