**Git**

# 版本控制器

为什么要用版本控制软件？？

软件有多个版本，不同的版本还需要协同开发，这些问题都需要解决，当然手工可以解决，

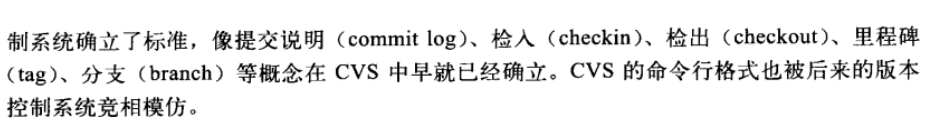
但是手工太慢，管理复杂，容易出错。

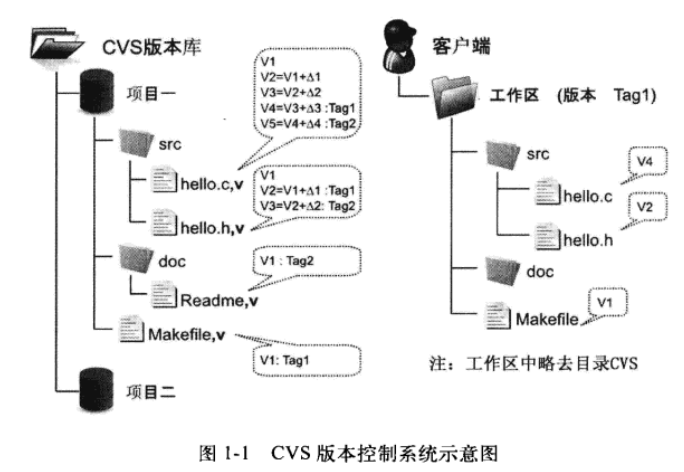
任何事物都是由简单到复杂。。。。。。。。。。。

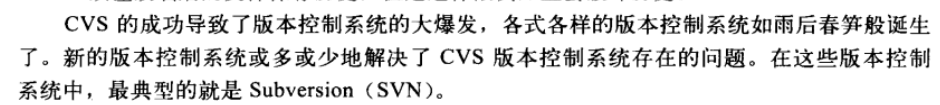
Diff ----🡪 cvs --🡪svn ----🡪 git

## Cvs

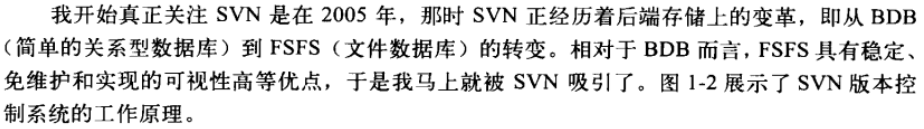
人类第一个正式的版本控制器。

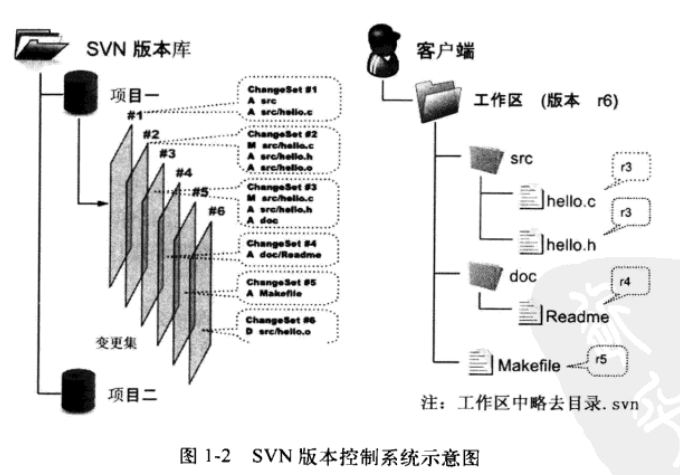


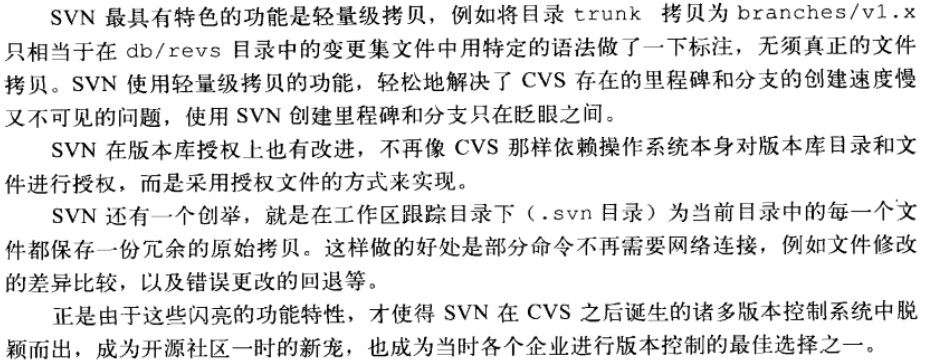


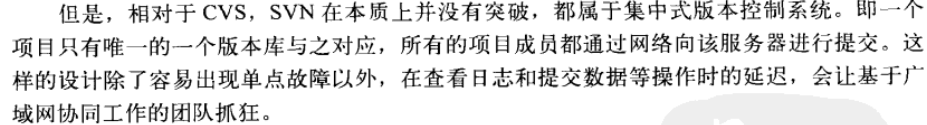


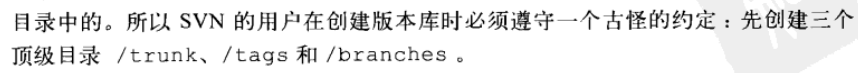
## Svn





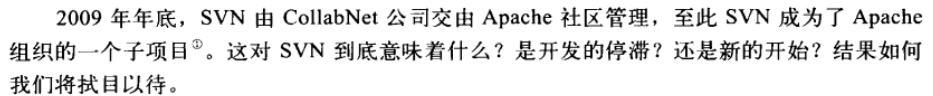




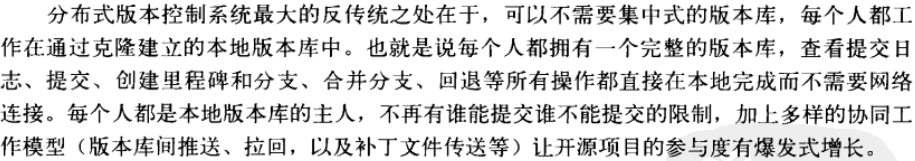


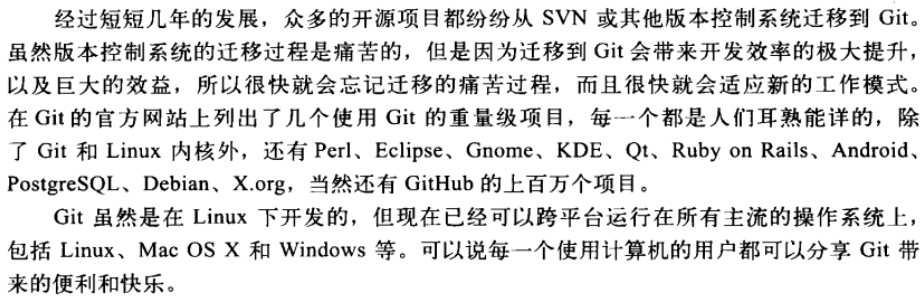
自带权限管理账号

一般用于商业项目开发！！



## Git

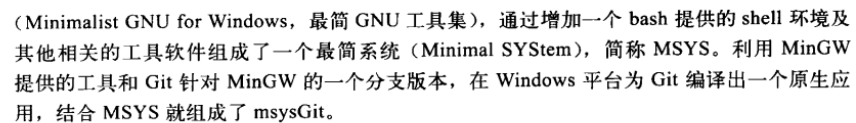




# Windows安装msysgit







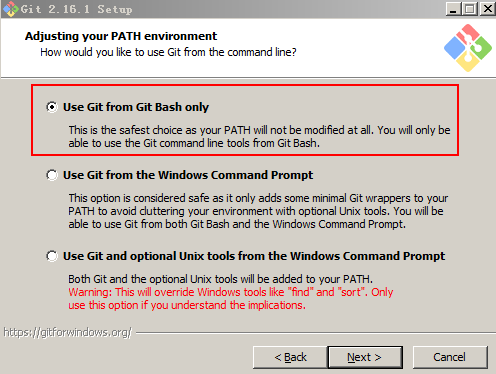
官网：<http://gitforwindows.org/>

官网文档：<https://git-scm.com/docs>

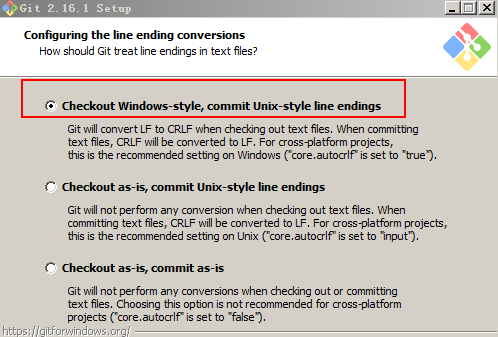
**安装之前科普知识：**

|  |
| --- |
| ****操作系统文件换行符****       首先介绍下，在ASCII中存在这样两个字符**CR**（编码为13）和 **LF**（编码为10），在编程中我们一般称其分别为'\r'和'\n'。他们被用来作为换行标志，但在不同系统中换行标志又不一样。下面是不同操作系统采用不同的换行符：   * **Unix和类Unix（如Linux）**：换行符采用 **\n** * **Windows和MS-DOS**：换行符采用 **\r\n** * **Mac OS X之前的系统**：换行符采用 **\r** * **Mac OS X**：换行符采用 **\n** |

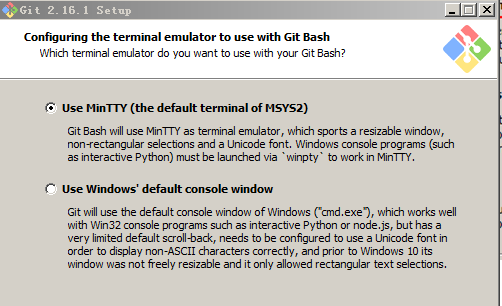
安装步骤比较简单，大部分都是下一步！！



处理不同系统文件换行符问题？？

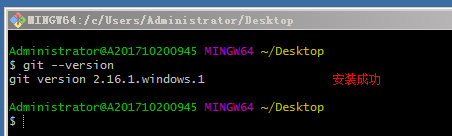


选择运行终端



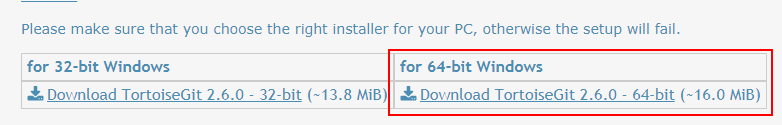
安装后右键菜单会出现的子菜单

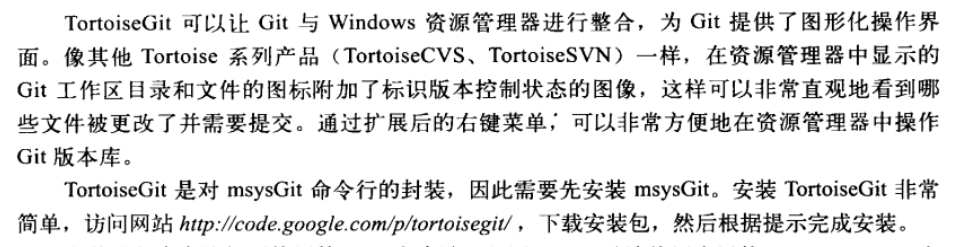
右键菜单打开运行终端，输入以下命令检查安装是否成功！！



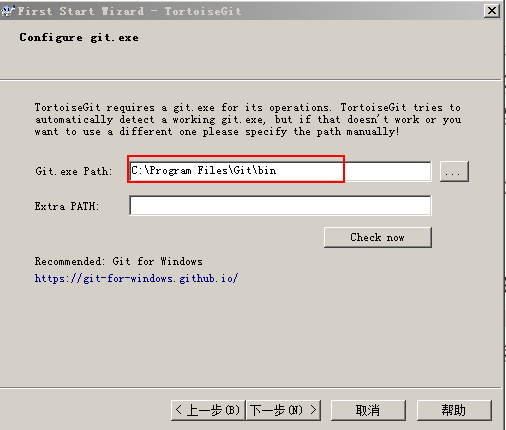
# TortoiseGit安装

<https://tortoisegit.org/>

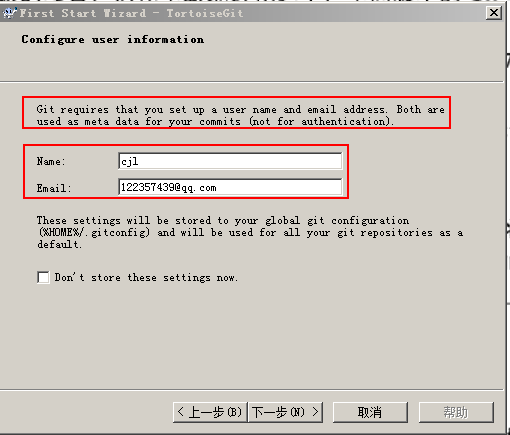




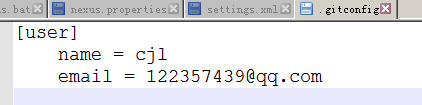
# tortoiseGit配置



初始化git，需要用户名和邮箱，请输入你自己的信息！！



**.gitconfig**



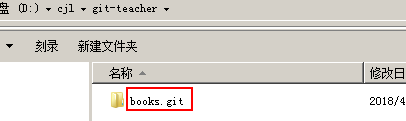
安装成功后，在右键菜单里会有相应的选项：



# Git独奏操作

## 建立本地版本库

Git虽然是分布式的版本库，理论上没有所谓的“集中服务器”概念，但是实际开发中，为了协同开发，共享代码，还是会建立一个逻辑意义上的“服务器”，这个服务器有个约定，名称以**.git**结尾！！！！



**没有工作空间的？！！（作为服务器使用）**





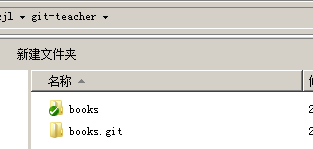
创建后如图目录：

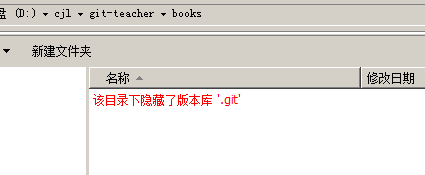


## 创建本地工作区

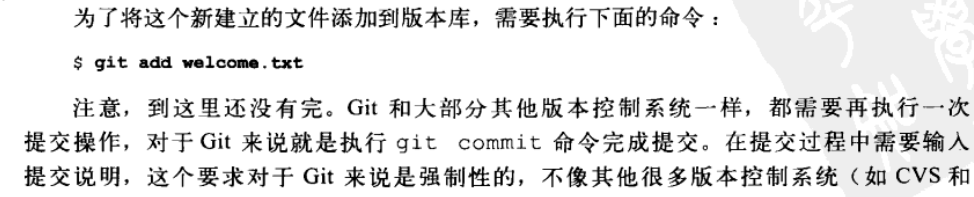


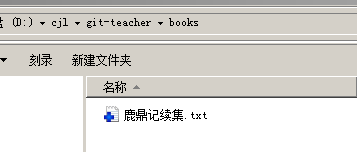
克隆成功后：



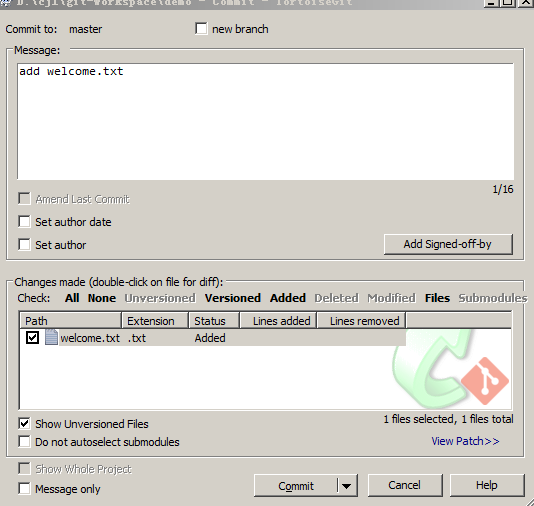


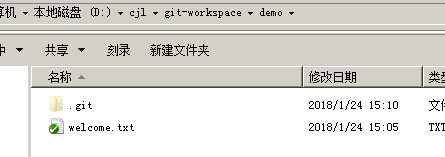
## 新增文件





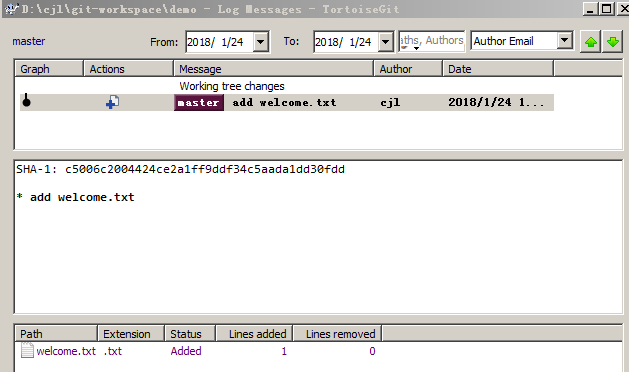
然后提交，或者直接提交



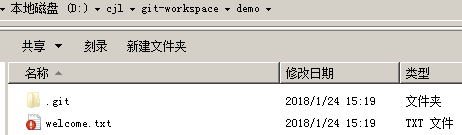


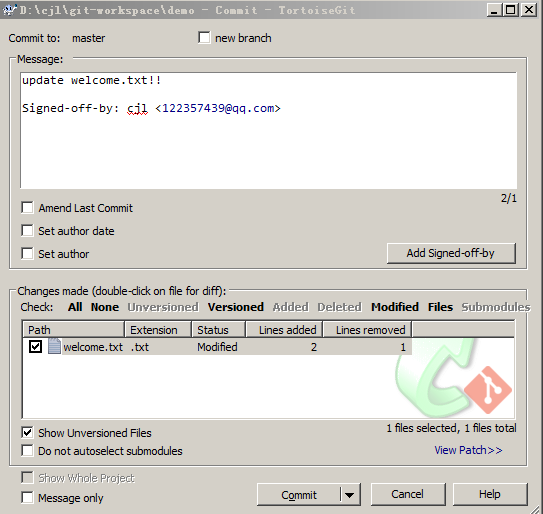
## 查找

可以通过日志命令查看当前版本库的情况



## 修改





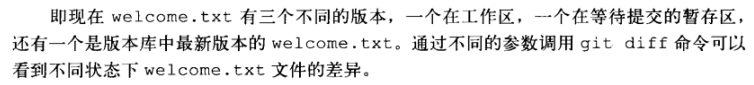
提交修改：

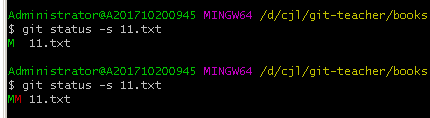


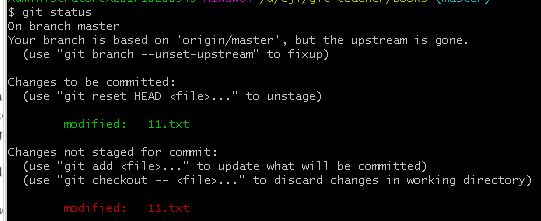
# Git暂存区

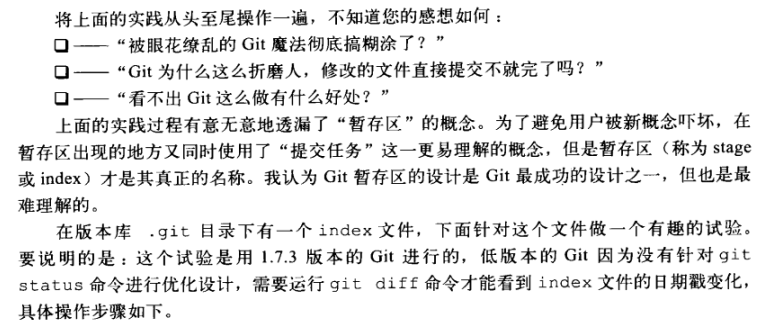
Git最重要的一个概念：暂存区。

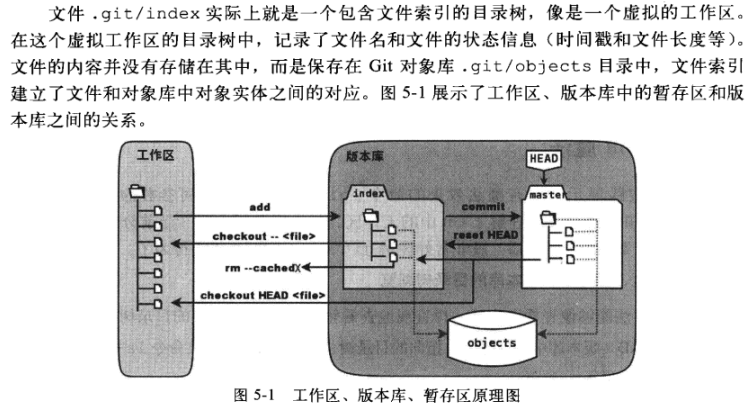
Git中一个文件有三种状态

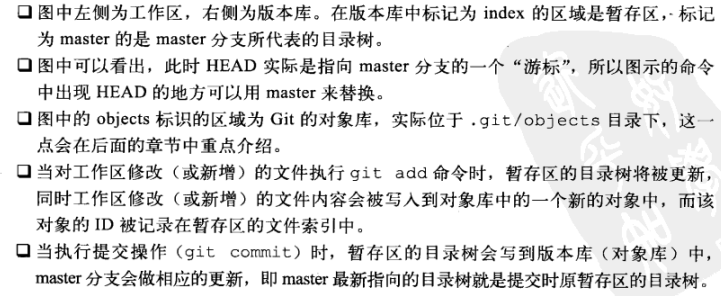


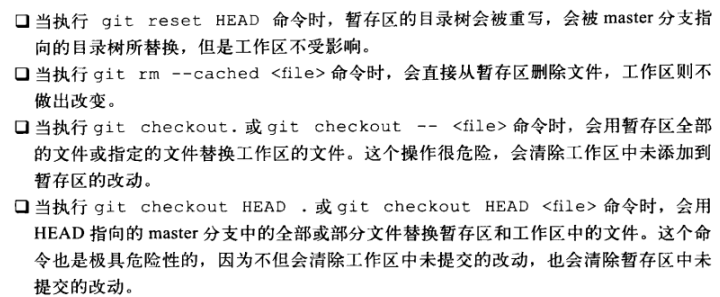








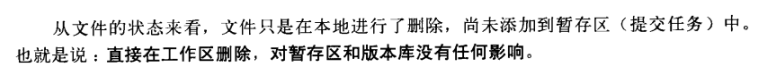


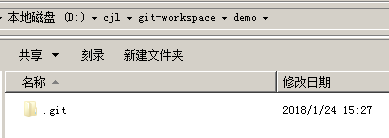


# 删除

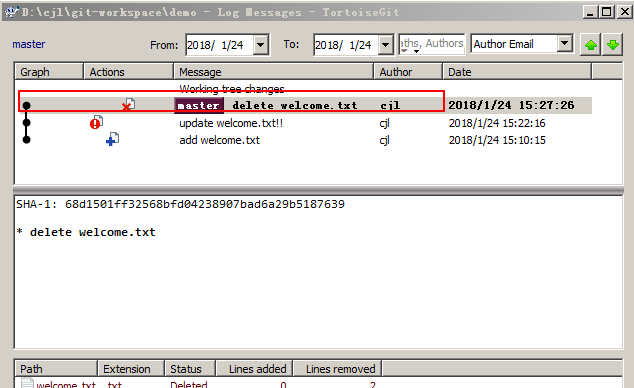
## 删除

直接在本地删除文件，只是删除工作空间的文件而已！！





可以revert来撤销就可以了！！

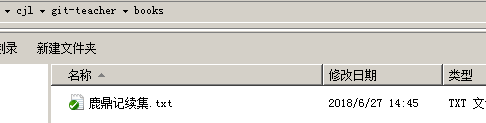


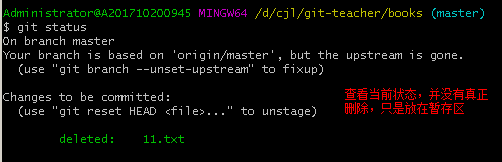




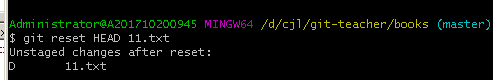
Git 删除

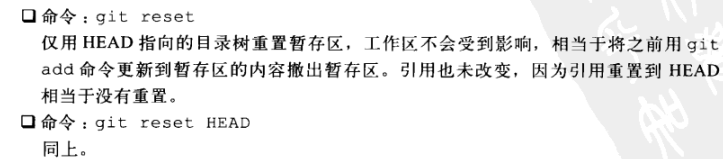






恢复刚删除的文件

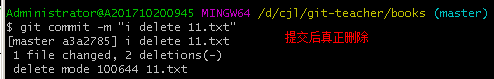




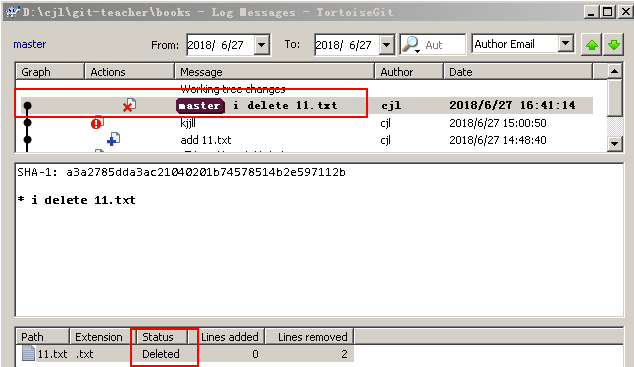


Git Checkout .

要彻底删除需要提交

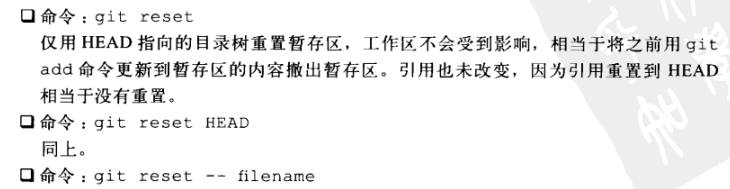


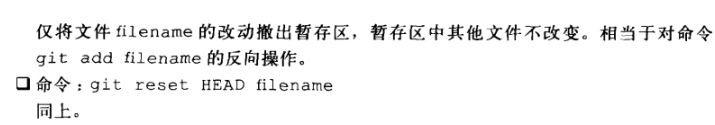
查看



但是在前面的提交版本中，文件依然是存在的！！！

## Reset

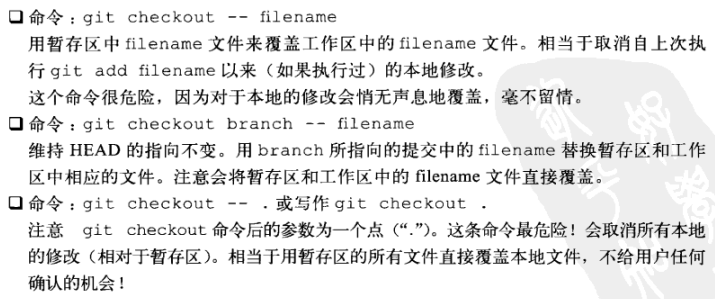




默认不指定撤销类型，就是撤销暂存区操作！！

例如我们利用这个恢复了删除的文件11.txt （只是在暂存区恢复，不影响工作区）

## Checkout



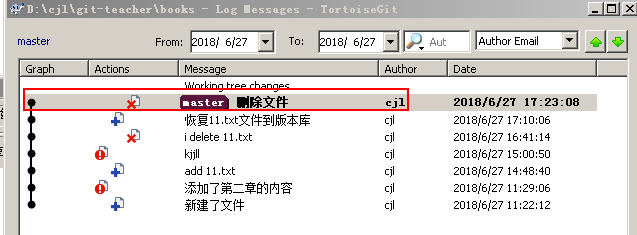
例如上面我们从暂存区恢复11.txt文件到工作区！！

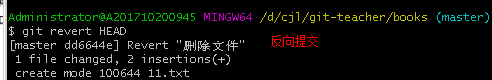
## Revert

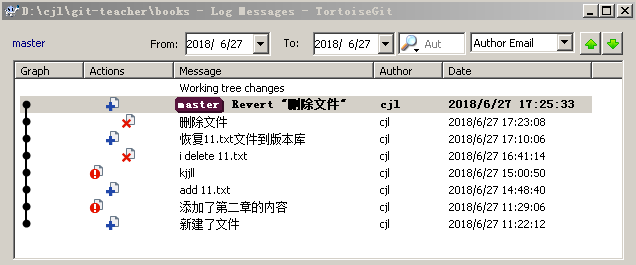
反向提交，可以恢复历史！！

如果只是本地误删文件可以，撤销删除，相对于本地回收站！！并不会执行提交！！

如果是提交删除的话，如下图

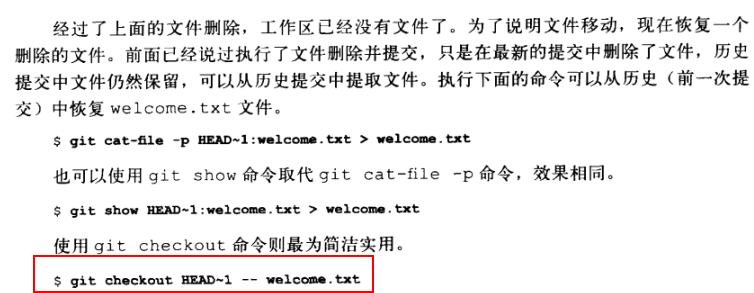




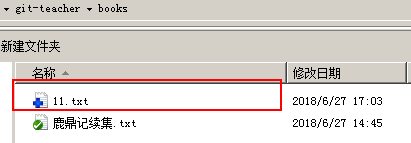


## 恢复删除

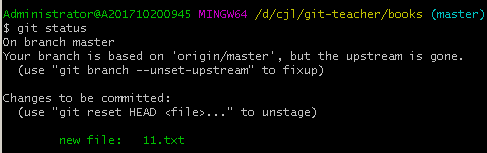
1 提取文件





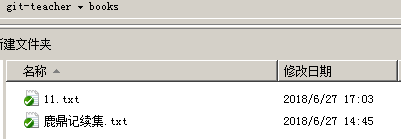


查看状态



2 恢复版本库

执行提交命令就行

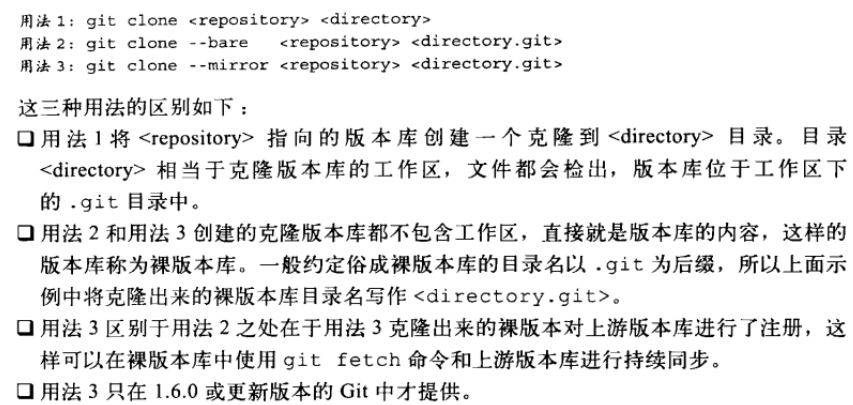




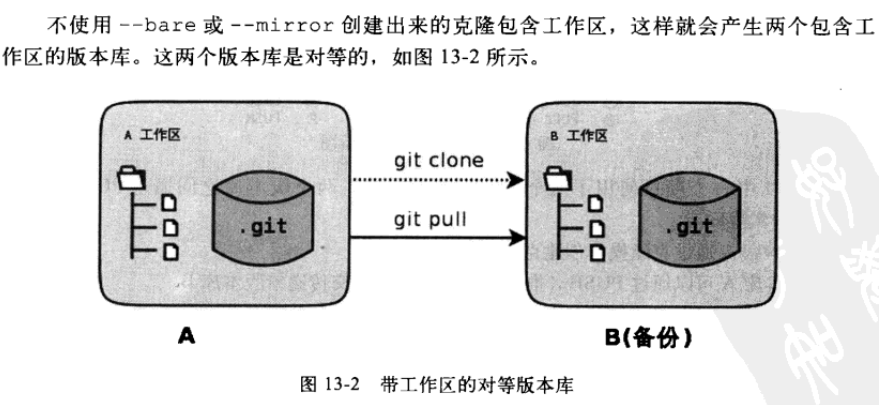


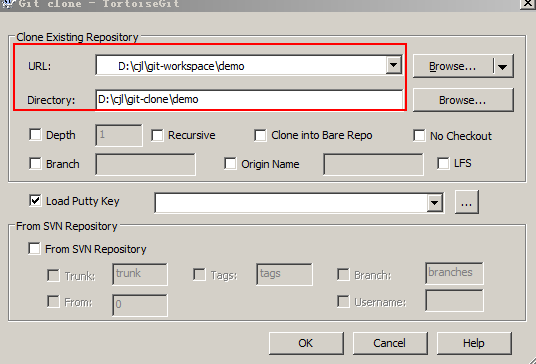


# Git克隆

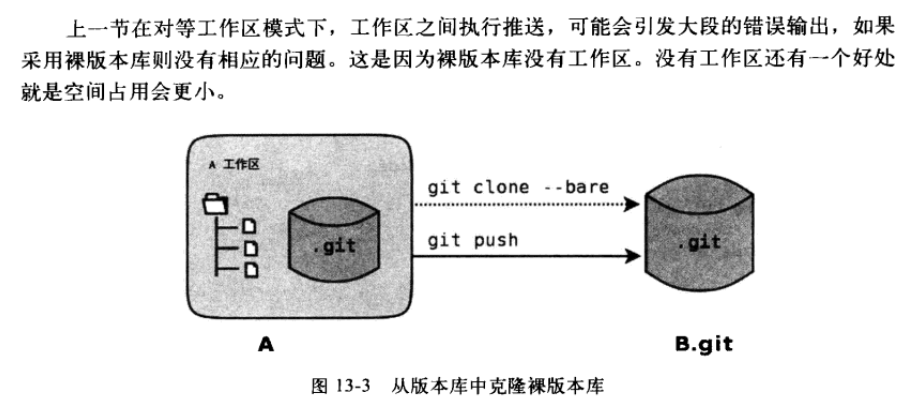


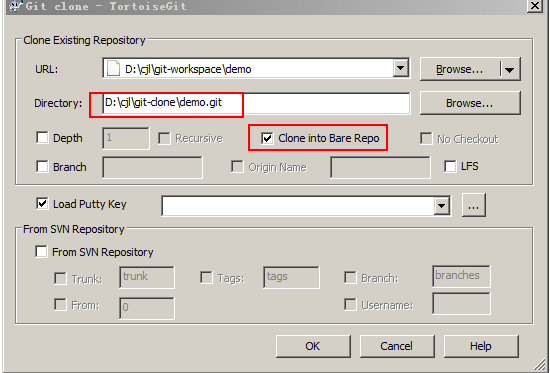
## 对等工作区





## 裸版本库







## 冲突解决

多人开发，对同一份文件修改时，产生不同的结果，就会产生冲突！！

有冲突就使用合并工具，进行沟通，协商，确认一个合并的结果，再次提交，就可以解决！！