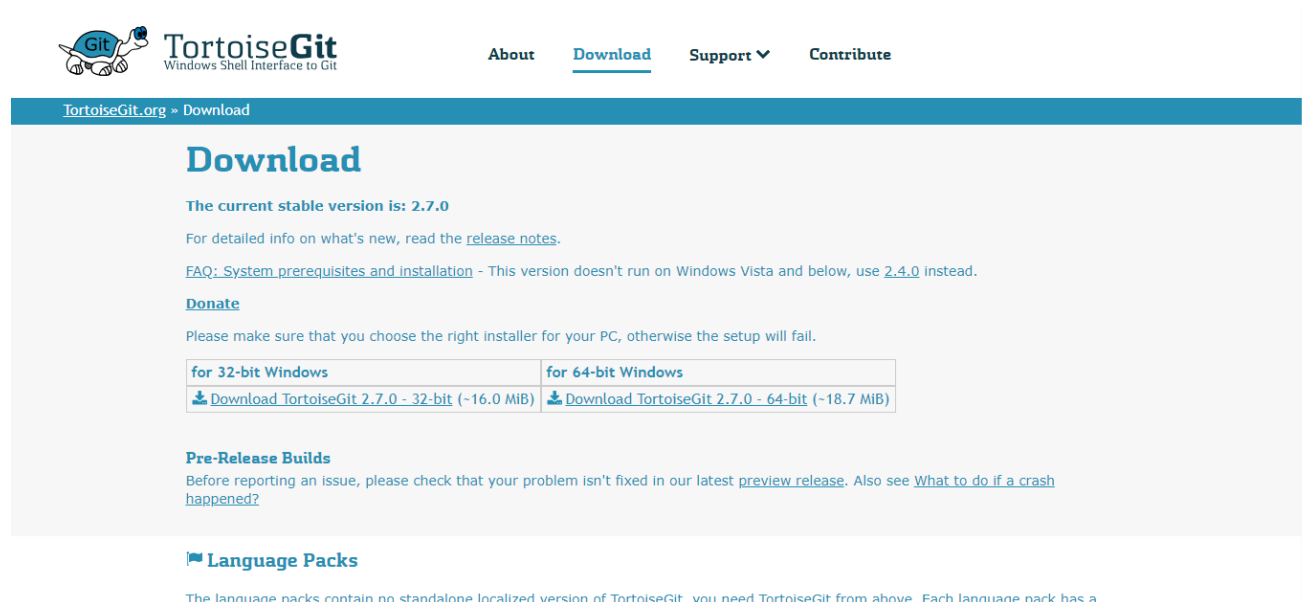


TortoiseGit使用手册

一、下载/安装TortoiseGit

TortoiseGit32/64位最新版及对应的语言包下载地址: <https://tortoisegit.org/download/>



The screenshot shows the TortoiseGit website's download page. At the top, there's a navigation bar with links for 'About', 'Download' (which is underlined), 'Support', and 'Contribute'. Below the navigation bar is a blue header with the text 'TortoiseGit.org > Download'. The main content area has a large 'Download' heading. It states 'The current stable version is: 2.7.0'. Below this, it says 'For detailed info on what's new, read the [release notes](#).' and 'FAQ: [System prerequisites and installation](#) - This version doesn't run on Windows Vista and below, use [2.4.0](#) instead.' There is a 'Donate' link. A note says 'Please make sure that you choose the right installer for your PC, otherwise the setup will fail.' Below this is a table with two columns: 'for 32-bit Windows' and 'for 64-bit Windows'. Each column has a download link: 'Download TortoiseGit 2.7.0 - 32-bit (~16.0 MiB)' and 'Download TortoiseGit 2.7.0 - 64-bit (~18.7 MiB)'. Below the table is a section for 'Pre-Release Builds' with a note about reporting issues and links to 'preview release' and 'What to do if a crash happened?'. At the bottom, there's a section for 'Language Packs' with a note that they don't contain a standalone localized version of TortoiseGit.

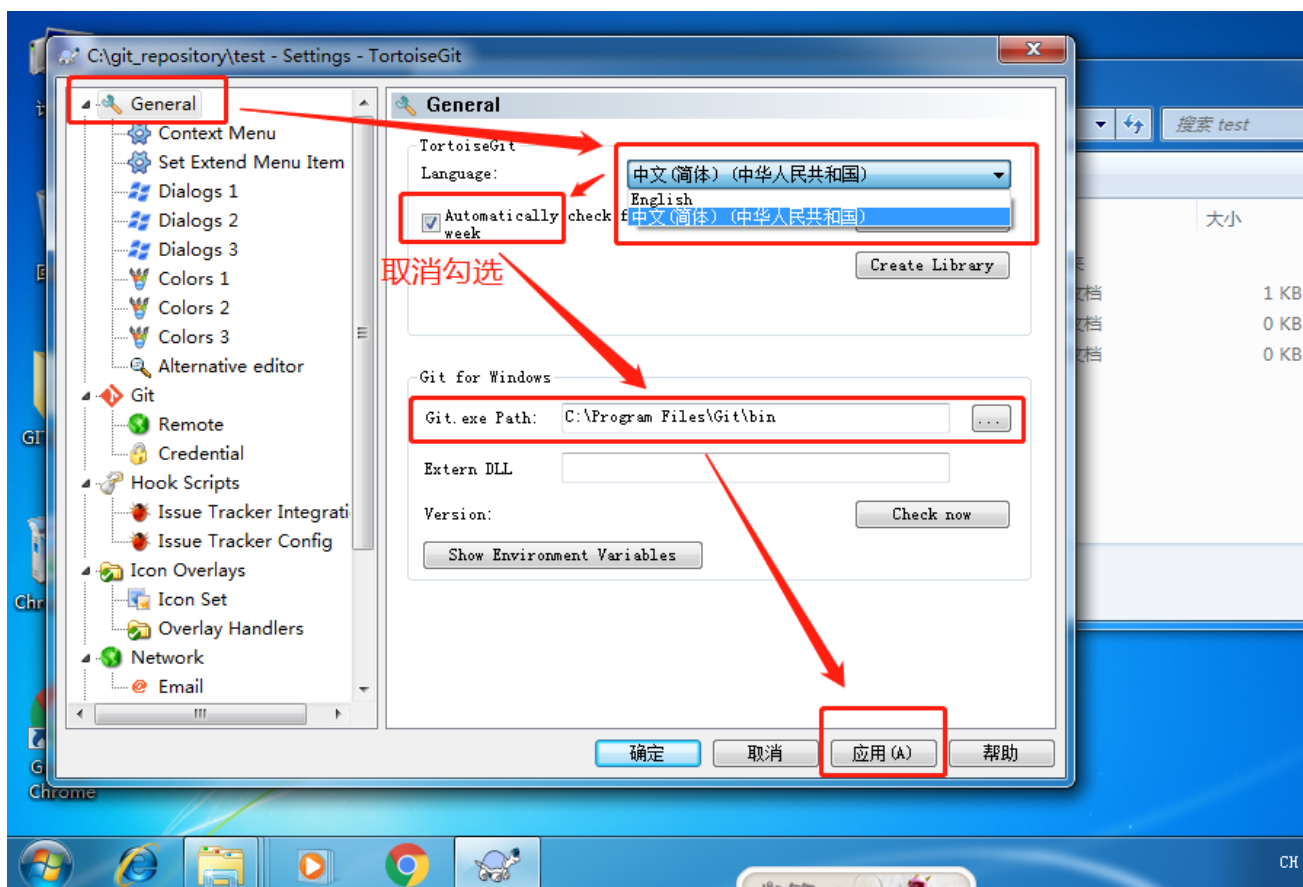
安装的方法, 一直下一步就行, 具体做法省略。

汉化的方法, 安装TortoiseGit-LanguagePack, 一直下一步就行, 具体做法省略。

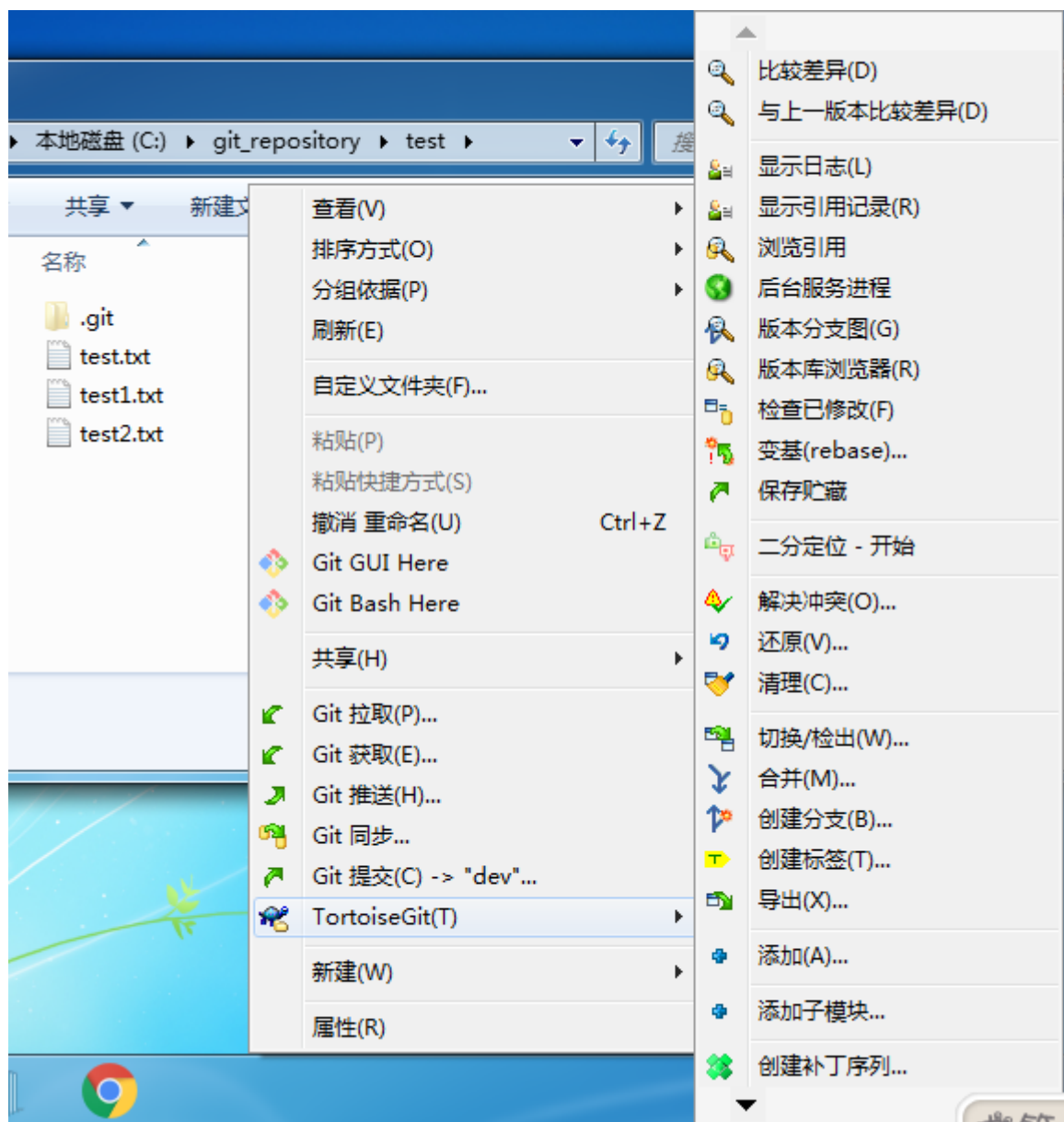
二、配置

首先,请选定一个存放Git项目的目录,这样管理方便. 如: C:\test, 然后在资源管理器中打开.

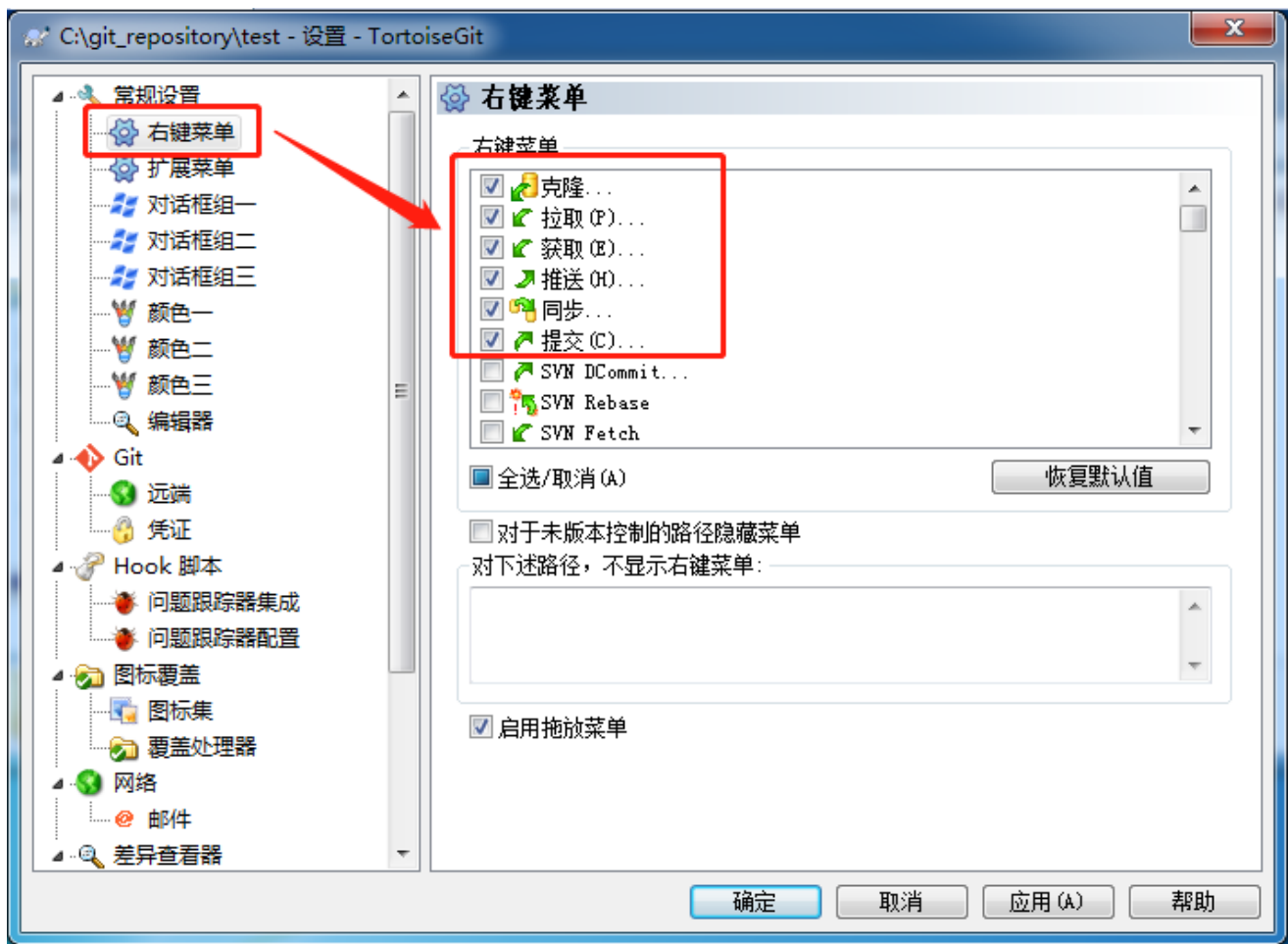
在空白处点击鼠标右键, 可以看到右键菜单中多了几个选项。 选择 --> TortoiseGit --> Settings, 然后就可以看到配置界面, 选中General, Language中选择中文。不勾选自动升级的复选框, 可能还需要指定 Git.exe 文件的路径, 如下图所示:



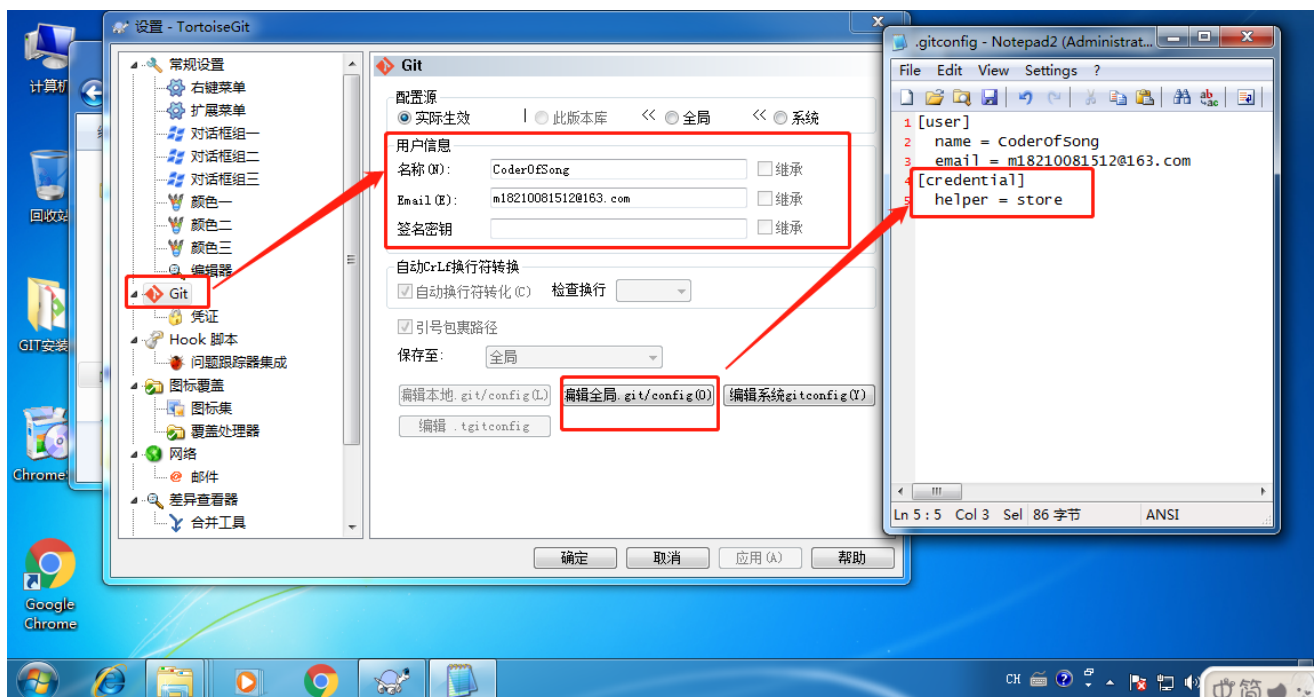
再次点击鼠标右键，可以看到弹出菜单中已经变成中文。原来的 Settings 变成 设置，Clone 变为 克隆。



配置右键菜单。在设置对话框中，點選左边的 **右键菜单(Context Menu)**，设置常用的右键菜单。比较常用的是如下选项：



设置记住密码。密码会明文保存在 C:\Users\song\.git-credentials 这种文件中, 请小心使用。进入设置, 点选左边的Git标签, 可以发现, 右边可以配置用户的名字与Email信息。如下图所示:



因为当前还没有本地项目，所以 `编辑本地 .git/config(L)(Edit local .git/config(L))` 按钮处于灰色不可用状态，如果在某个本地Git项目下打开配置对话框，那么这个按钮就可用，然后就可以编辑此项目的一些属性。点击 `编辑全局 .git/config(O)(Edit global .git/config(O))` 按钮，会使用记事本打开全局配置文件，在全局配置文件中，在后面加上下面的内容：

```
[credential]
    helper = store
```

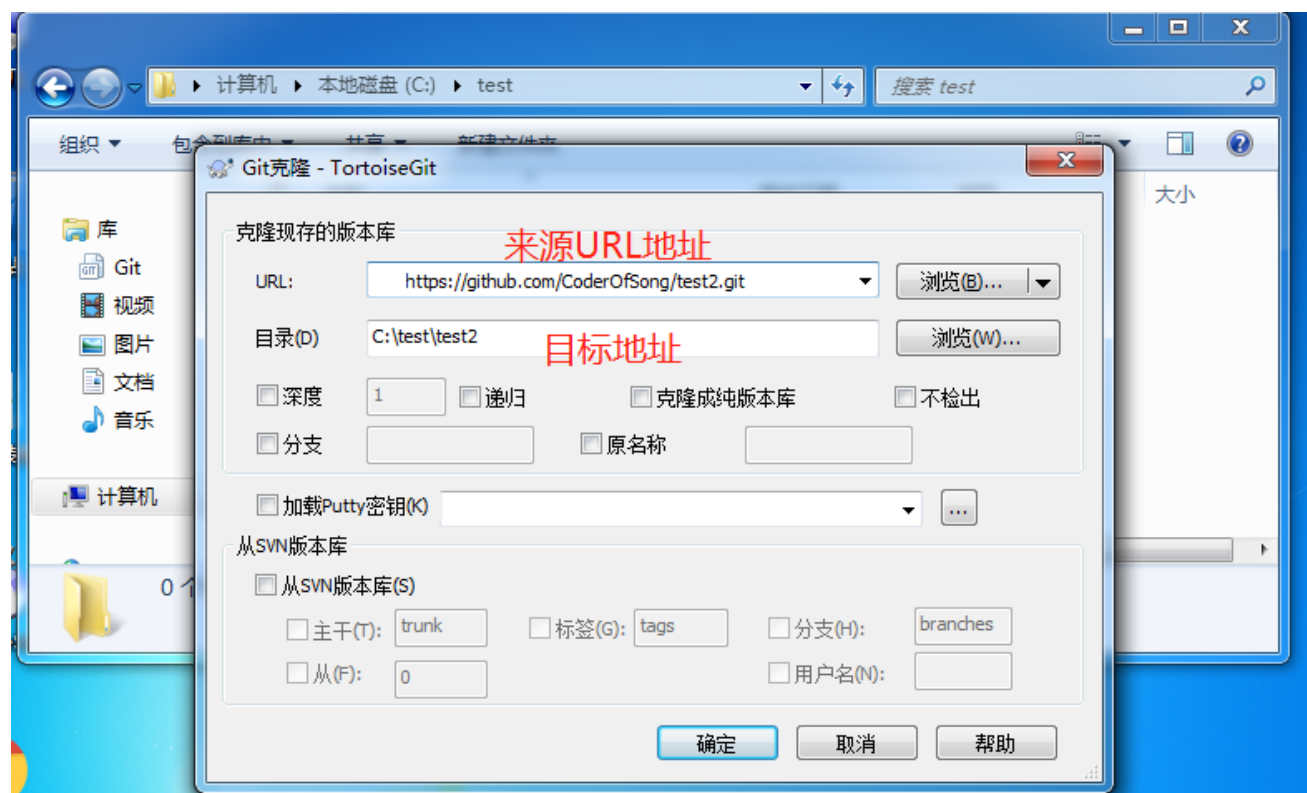
完成后保存，关闭记事本，确定即可。

当你推送项目到GitHub等在线仓库时，会记住你输入的用户名和密码

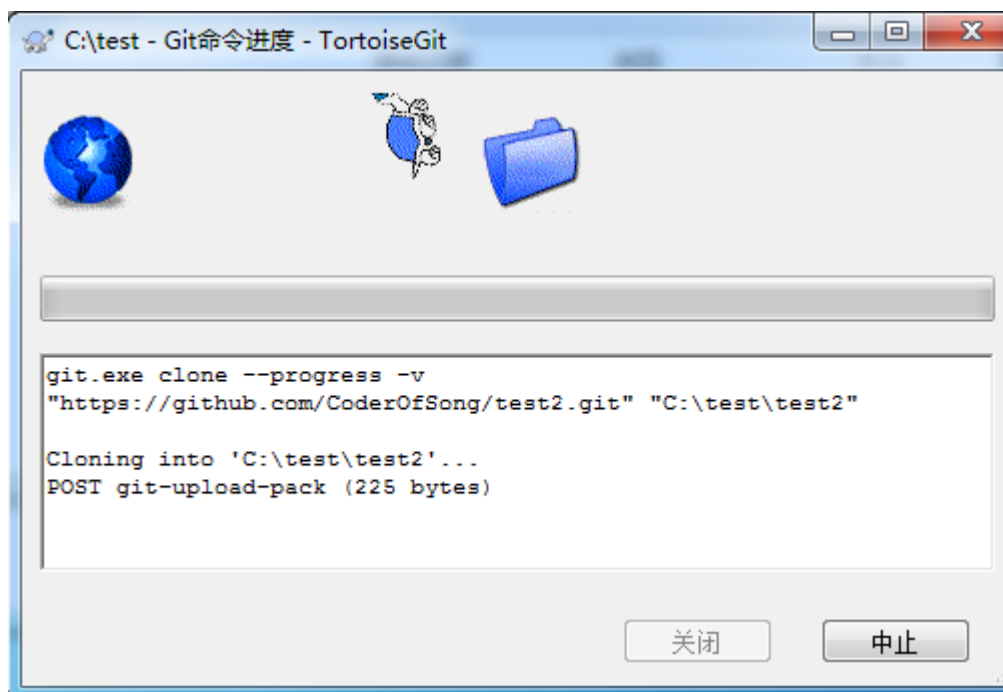
三、使用

1、克隆项目

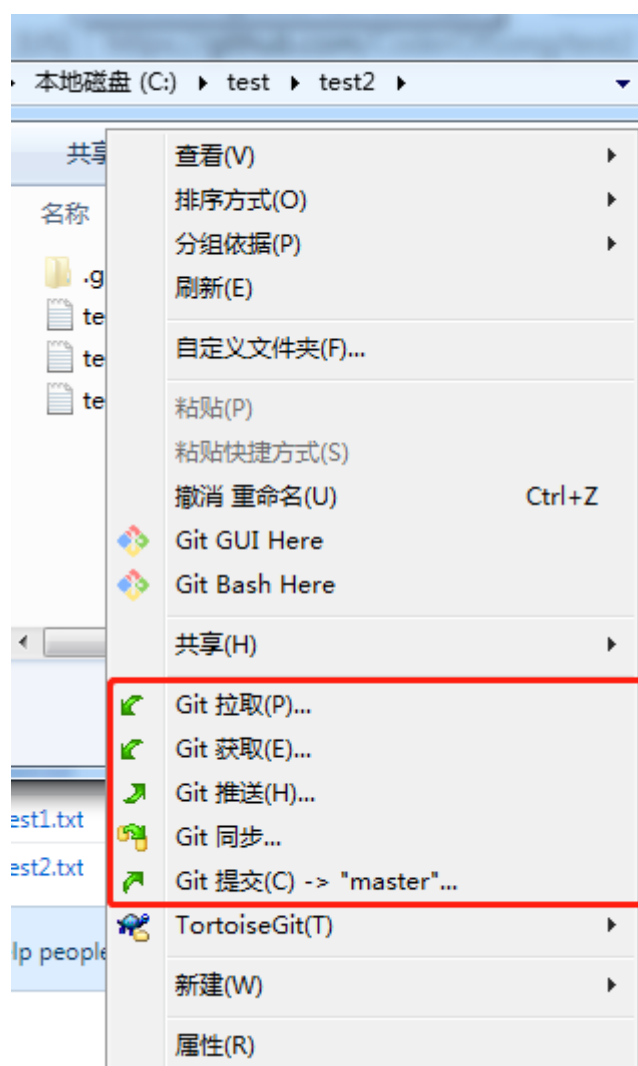
在工作目录下，如C:\test，空白处右键，选择： `Git 克隆(Git clone)`，则会弹出克隆对话框，如下图所示：



在URL中填写项目的访问地址，如：<https://github.com/CoderOfSong/test2.git> 根据项目大小,时间会不一样. 克隆完成后,如果没有错误,会给出提示:



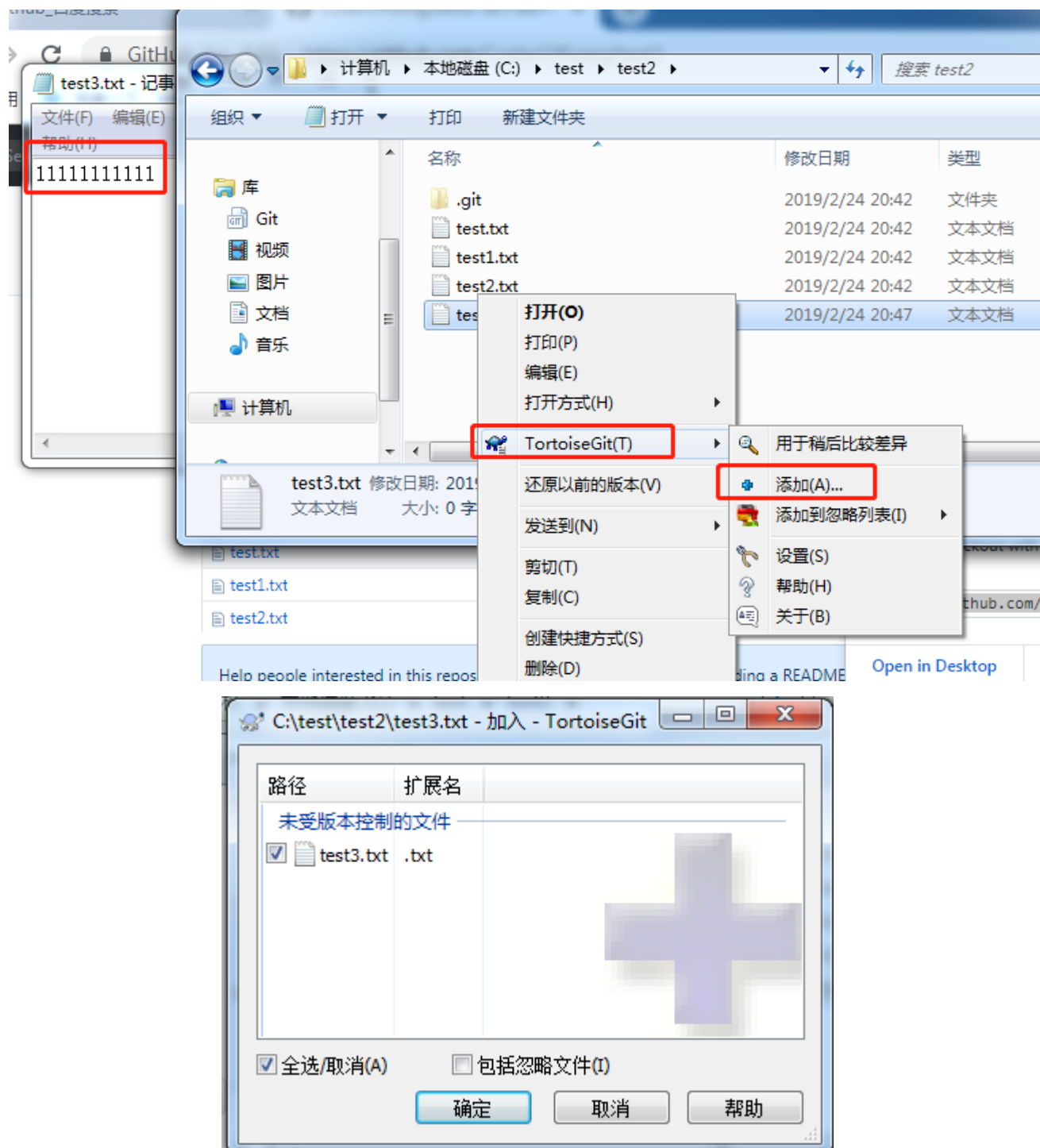
进入克隆下的文件夹中，如C:\test\test2，空白处右键，弹出如下菜单：

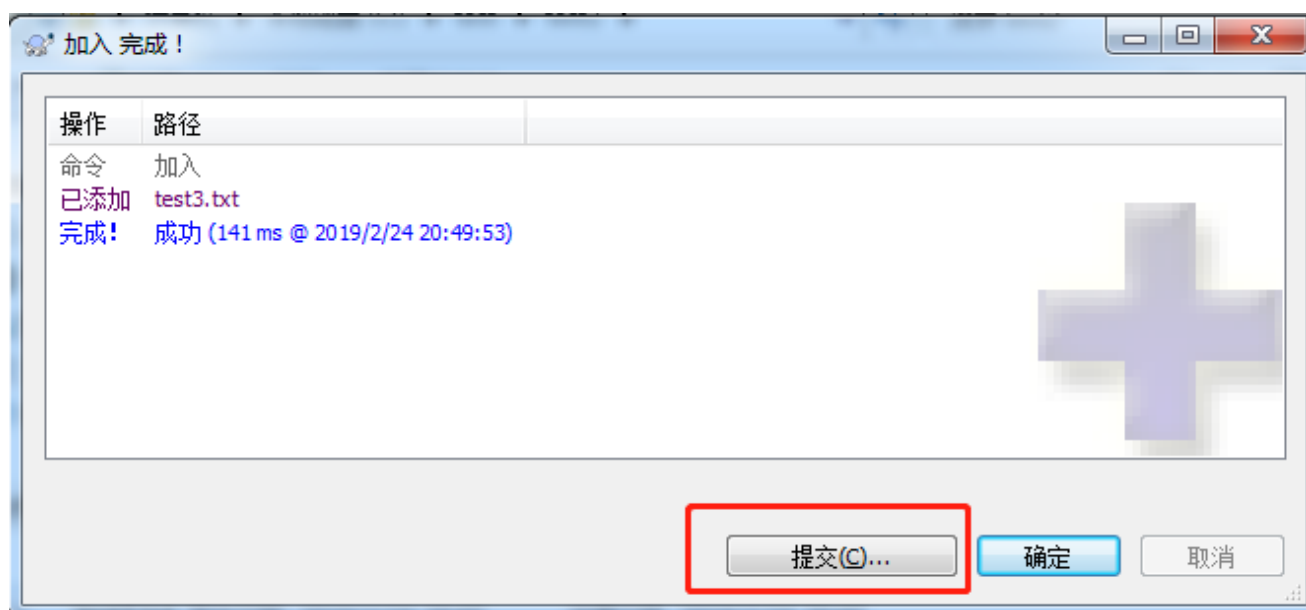


其中，Git 拉取(Git Pull)是从远端拉取最新的代码，Git 获取(Git Fetch)是从远端拉取最新的分支，Git 推送(Git Push)是将本地仓库的代码提交到远端，Git 提交 -> "master" (Git Commit -> "master")，将本地代码提交到本地版本库（默认分支是master）。

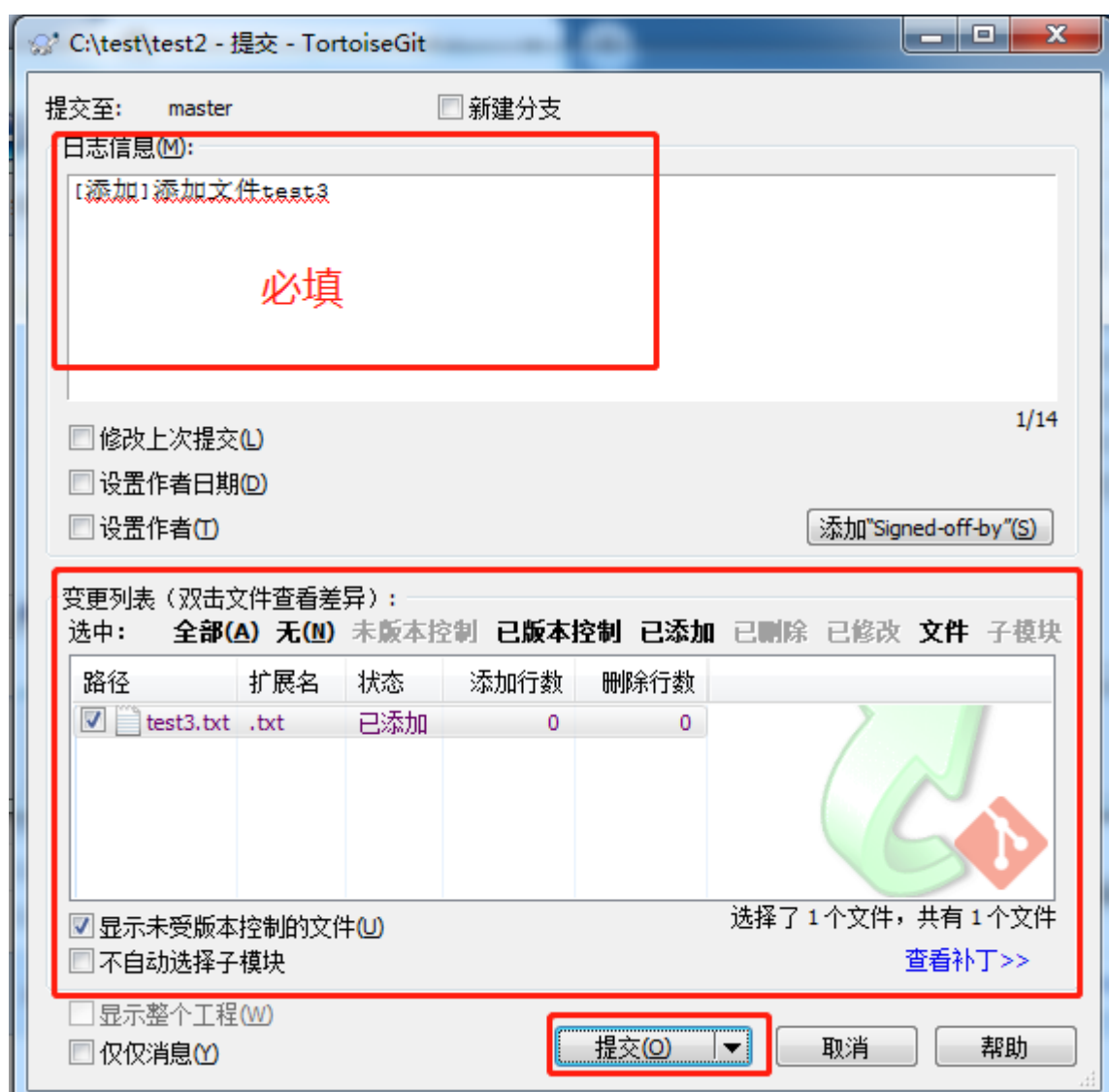
2、提交代码到本地库

创建一个文件，如test3.txt,然后 添加(add) 到暂存区

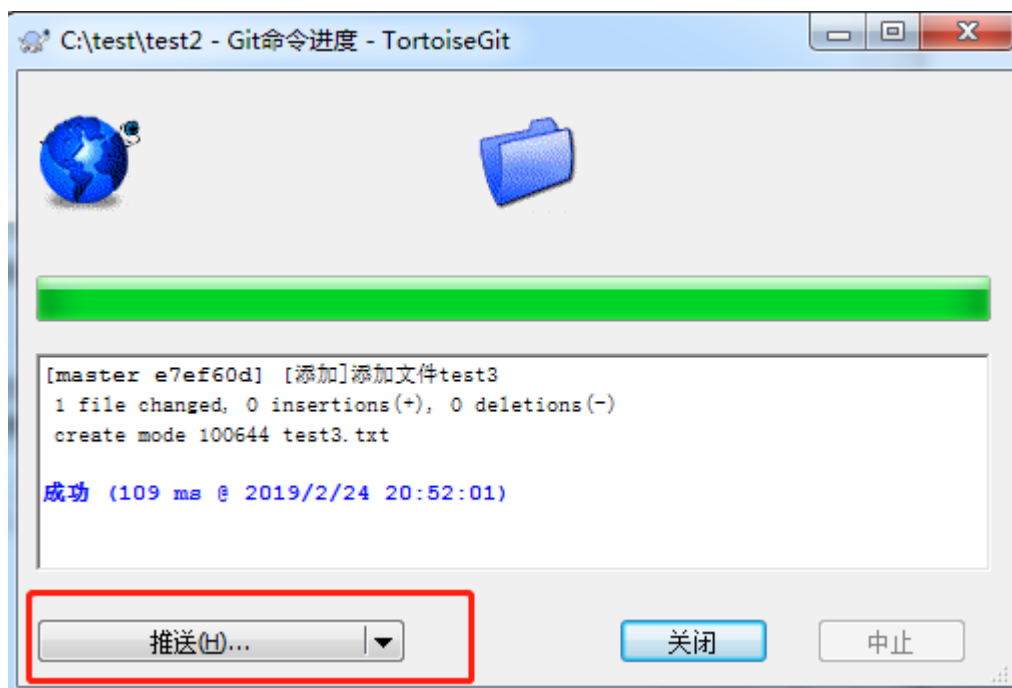




然后提交(commit)到本地版本库(这个操作可以在离线状态操作)，弹出下图：



填写提交备注message（不填写不允许提交），勾选需要提交的文件，点击commit，即可将本地代码提交到本地版本库。出现如下弹框，表示提交成功

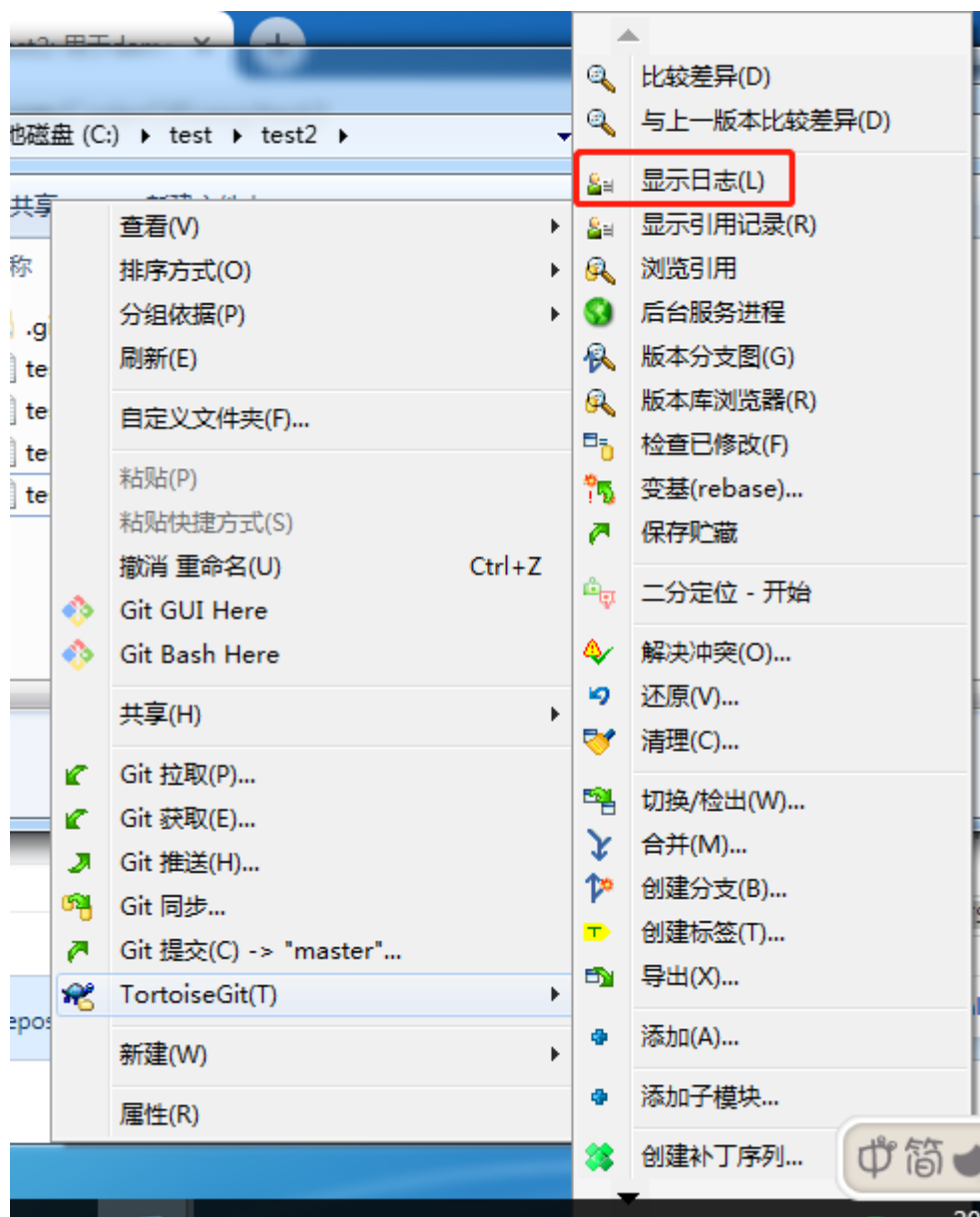


其中，提交时，会发现上图中的Status有几种值：

Unknown: 新增的文件，也不在版本库
Added: 新增的文件，在版本库
Modified: 文件修改，在版本库
Missing: 文件被删除，在版本库

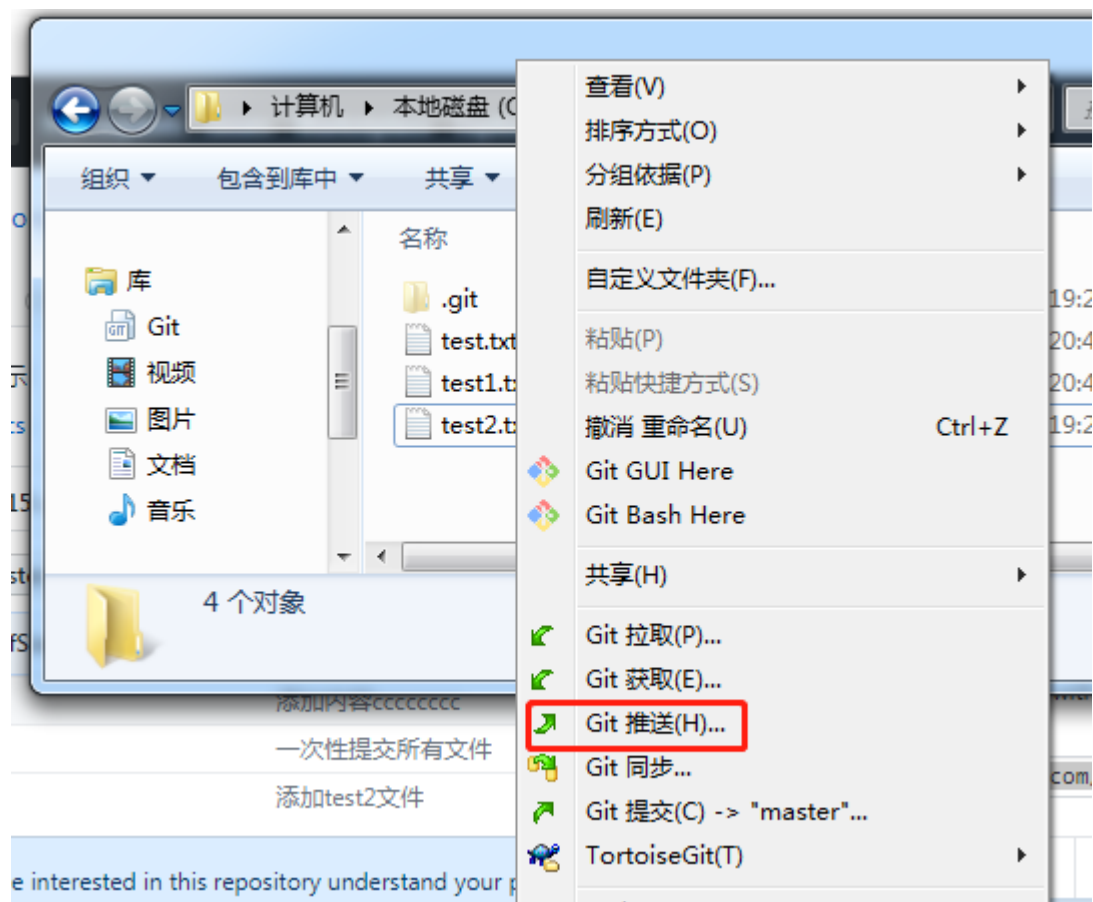
3、查看日志

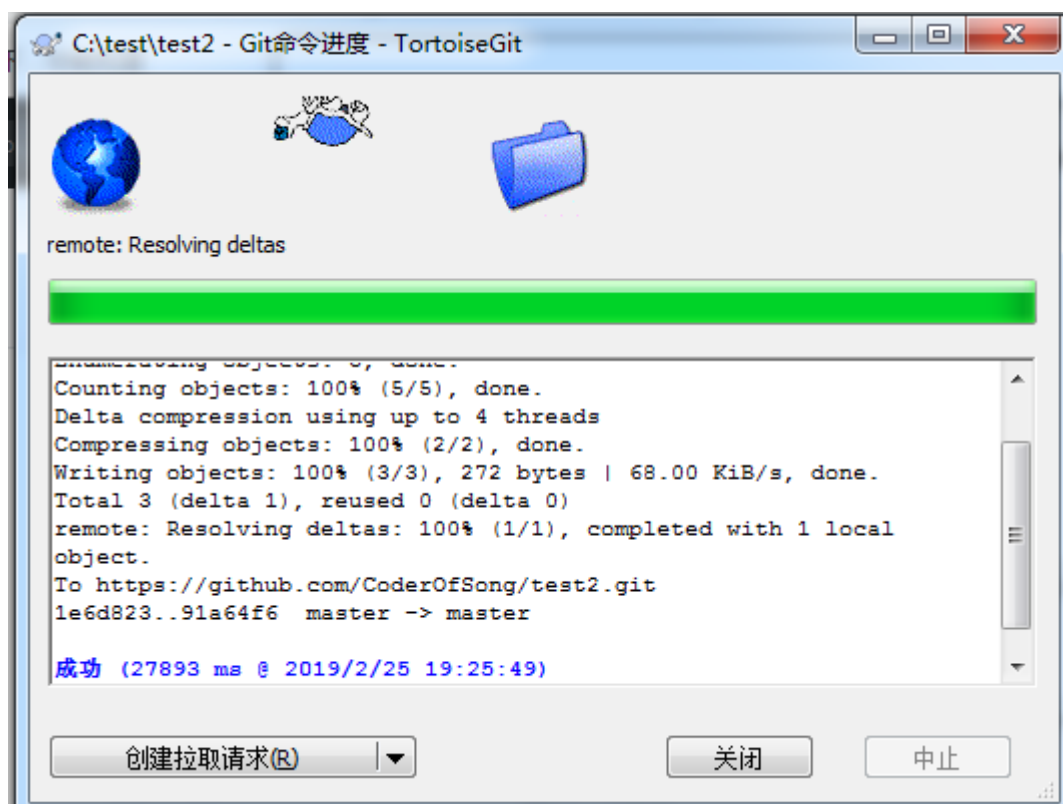
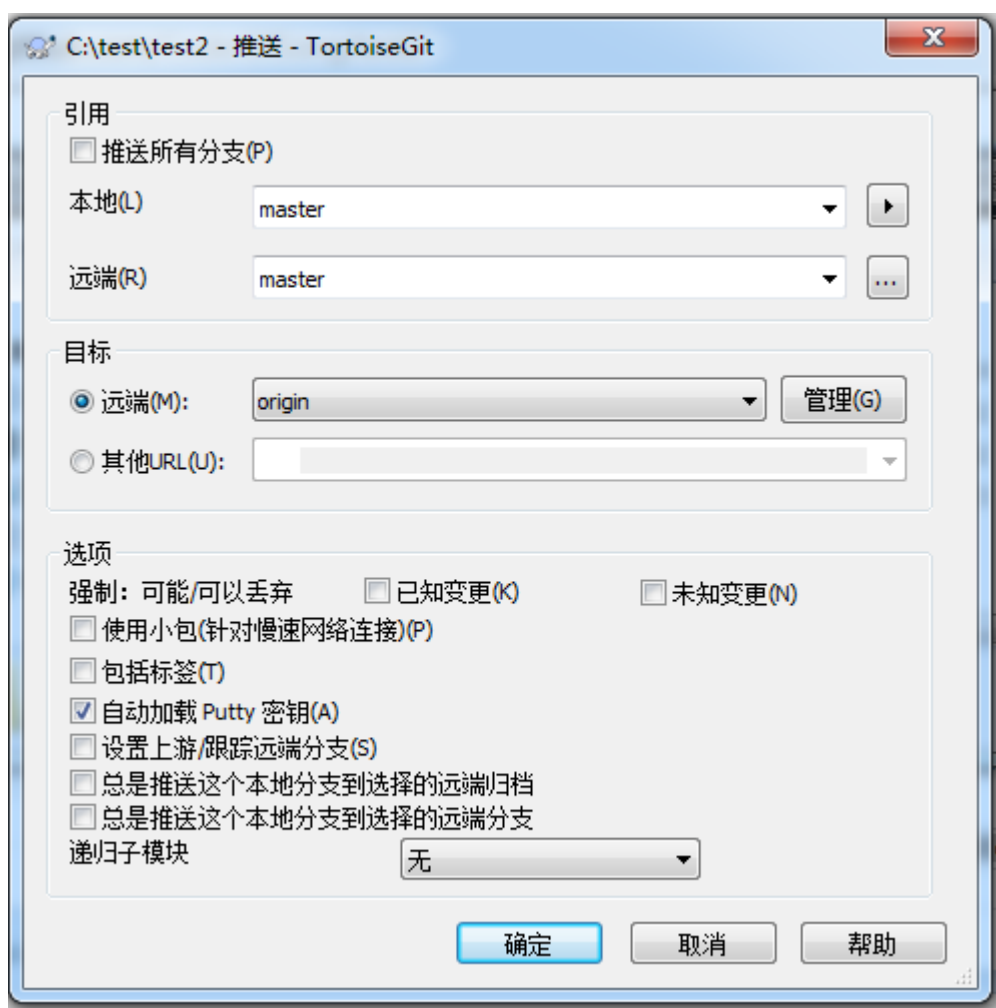
在日志中，可以通过日期、文件名、提交人等等过滤查询。



4、推送到远端

右键空白处，选择Git Push，出现如下弹框：





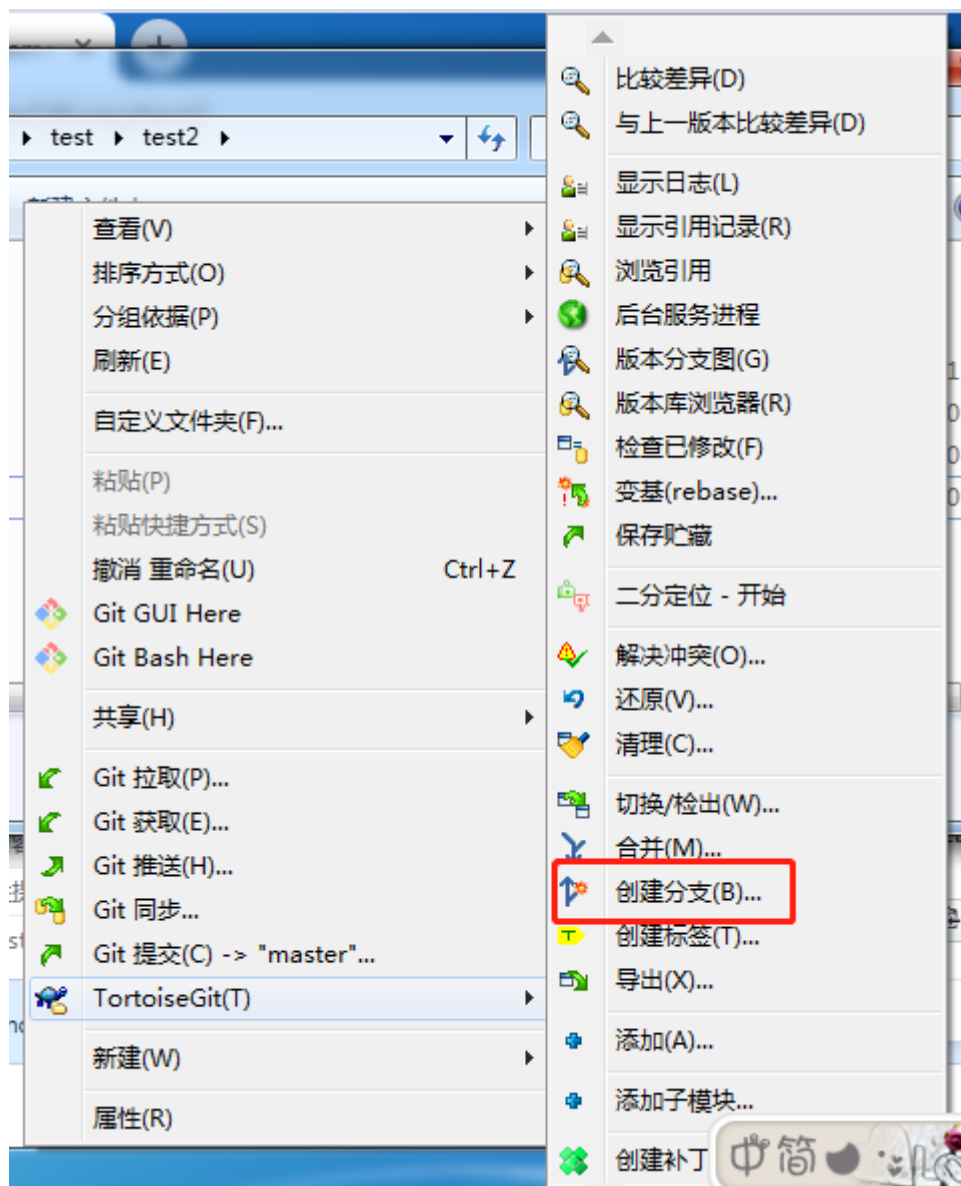
这里可以看到是本地哪个版本库提交到远端。至此，文件的整个提交过程就完成了

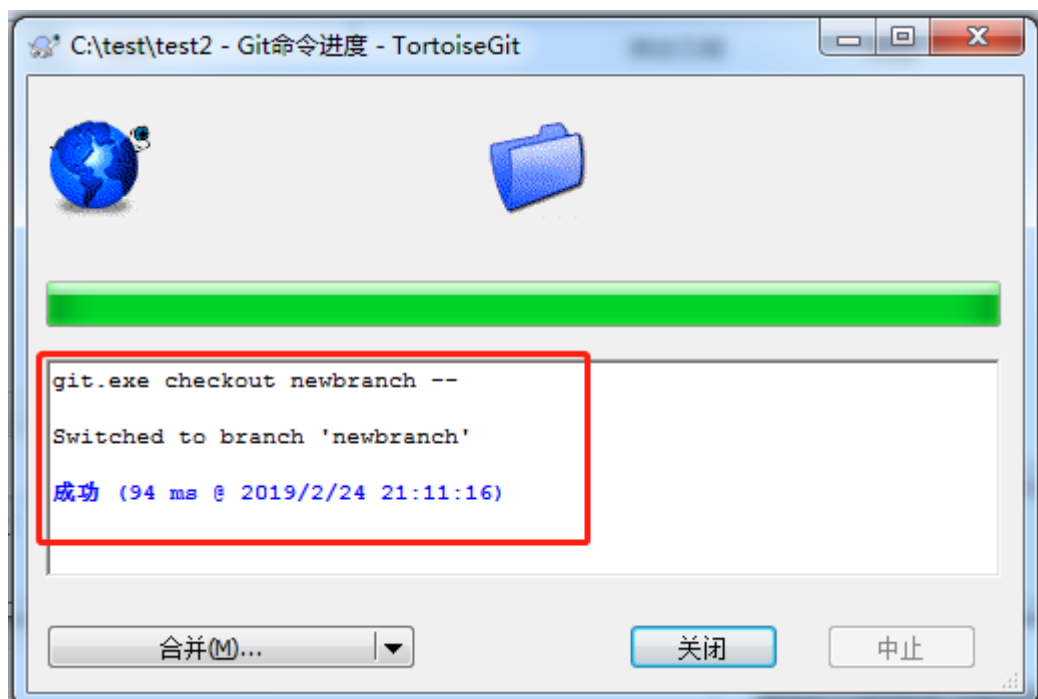
四、分支

发现问题：你代码写了很多，运行OK；但是突然想加个新功能进去，这个功能你也不知道能否正常运行，而且修改过程中，除了新加代码和文件进去，还会修改以前的代码。要是万一失败，修改回来也是一种很麻烦的事情。这时候很多人就用备份方式。来看看git是怎么优雅的处理这个问题的。

git的处理方式：当你想加一个新功能进去的时候，你可以新建一个分支，例如名字叫newbranch，然后在分支中把新功能加上去，如果OK，将代码合并到master分支上，如果新功能失败，切换回master分支上来，在newbranch写的代码，又全看不到了。

1、新建分支

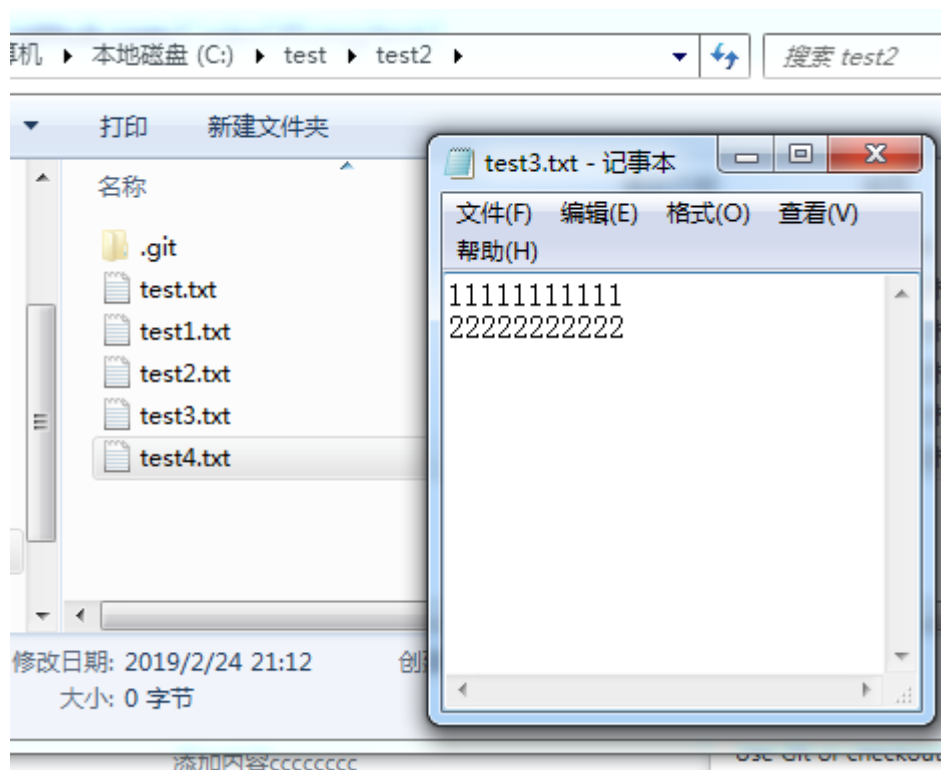




右键，你会发现当前的分支为你新建的newbranch分支了



接下来，就开心新增你的功能，比如我们在test3.txt文件中新增一行文字，同时新增一个文件test4.txt

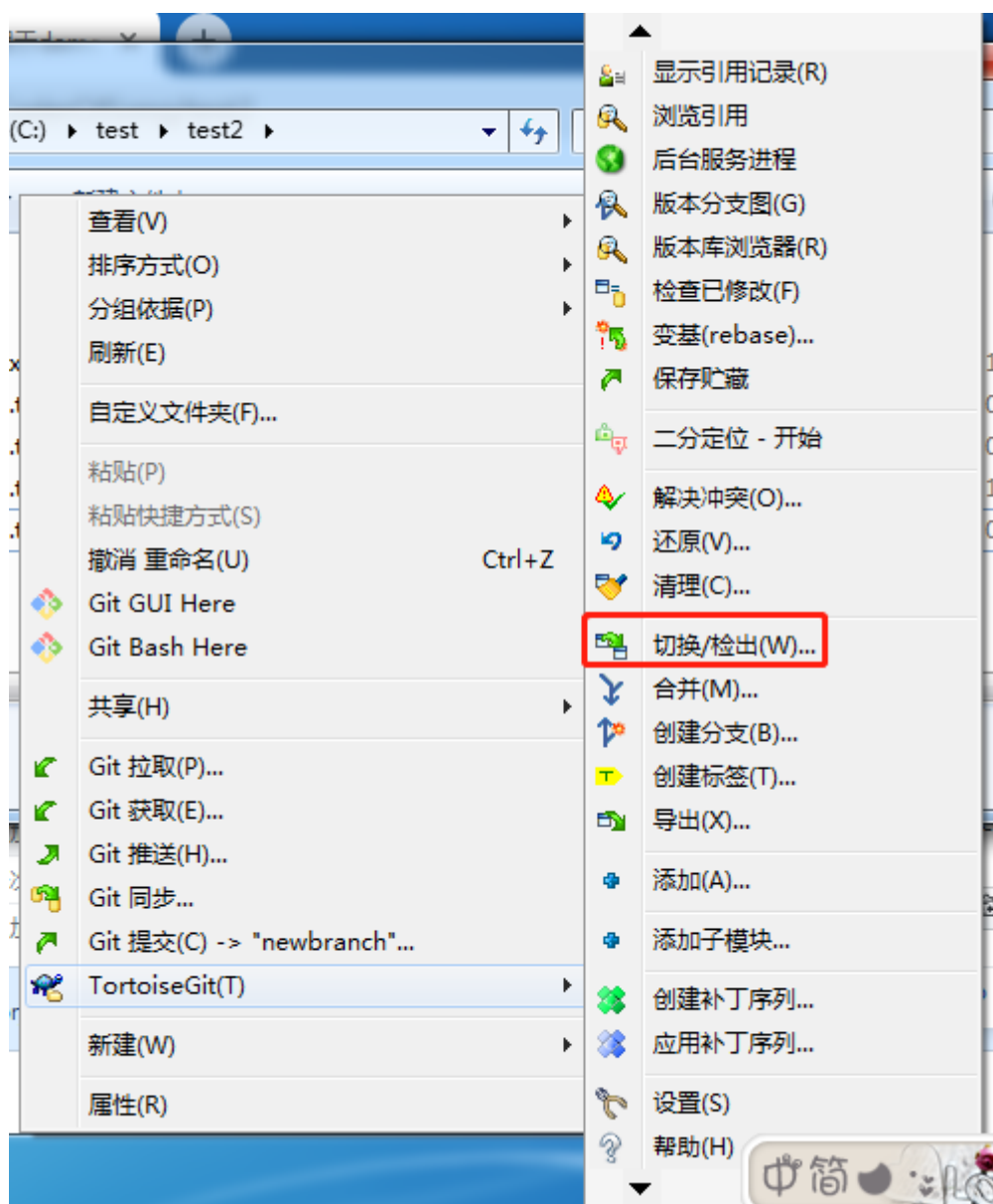


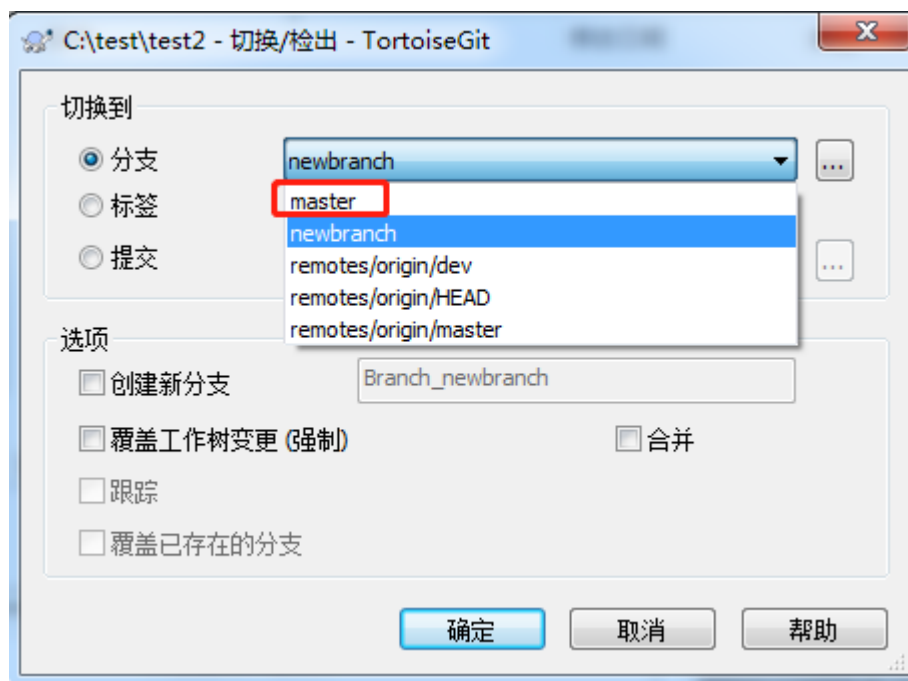
提交我们的代码到newbranch分支的本地仓库，参考之前的3.2

2.、切换分支

突然发现该功能有漏洞，想回到之前的master分支，怎么办呢？

右键-->TortoiseGit-->切换/检出(Switch/Checkout)，选择master即可。如下图：

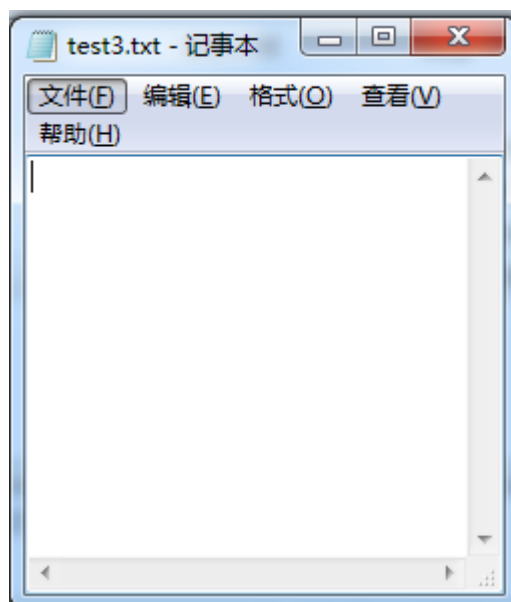




空白处右键，会发现已经切换到master分支了。



此时发现test4.txt并没有带过来，再看看test3.txt文件是否回到了从前呢？

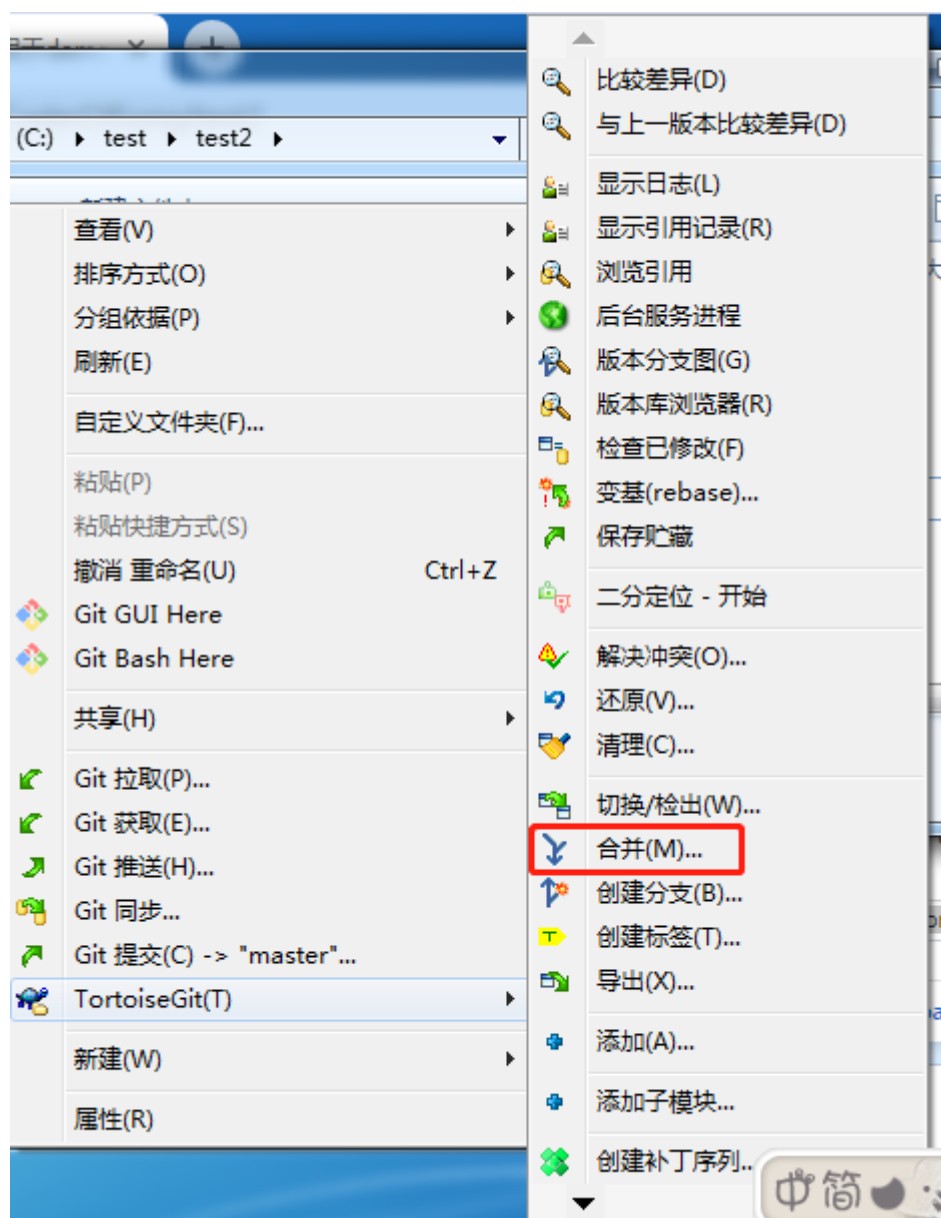


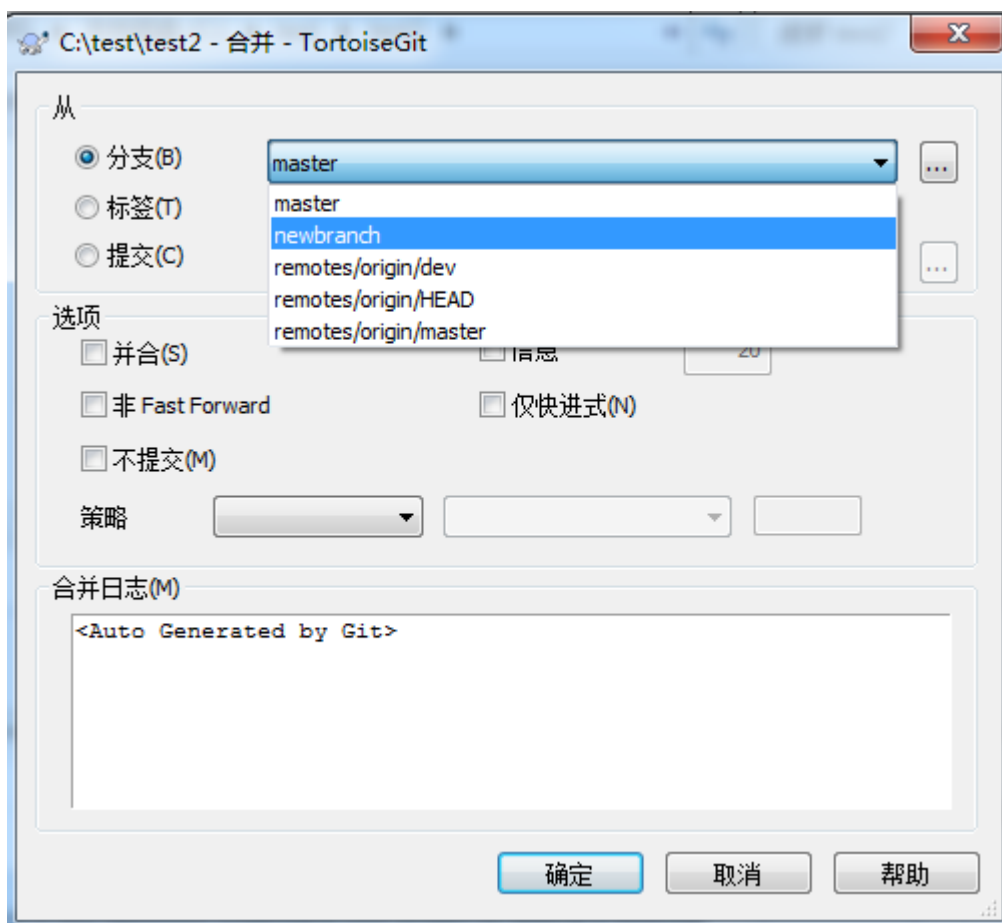
结果显示，无论是修改还是添加，都没有影响master分支上的文件，这就是分支的作用。

3、分支合并

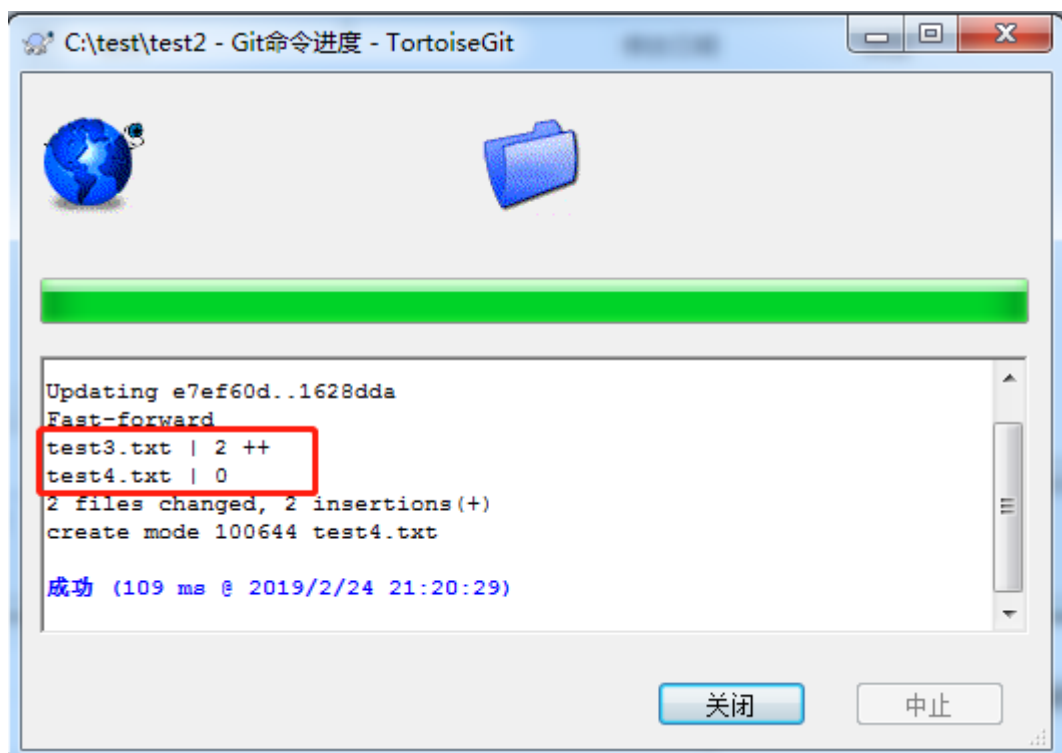
如果此时发现newbranch分支上的功能是有用的，希望能合并到master，又该怎么操作呢？（在合并分支前，一定要确认newbranch分支上的代码全部提交到本地版本库了）

右键-->TortoiseGit-->合并(Merge), 选择被合并的分支, 即newbranch。如下图





点击Ok按钮，会出现下图弹框。弹框中会列出被合并的文件。如下图所示：



我们发现test4.txt文件被合并过来了，打开test1文件，发现内容正好是在newbranch分支上修改的内容。至此，分支合并完成

五、拉取远端内容

此时，我们团队还有另一个成员B，他需要获取我最新修改的内容，该怎么操作呢？

首先B切换到跟我同一分支，然后右键-->Git 拉取(Git Pull)，点击确认即可，就可将我修改的内容拉取到他的本地版本库

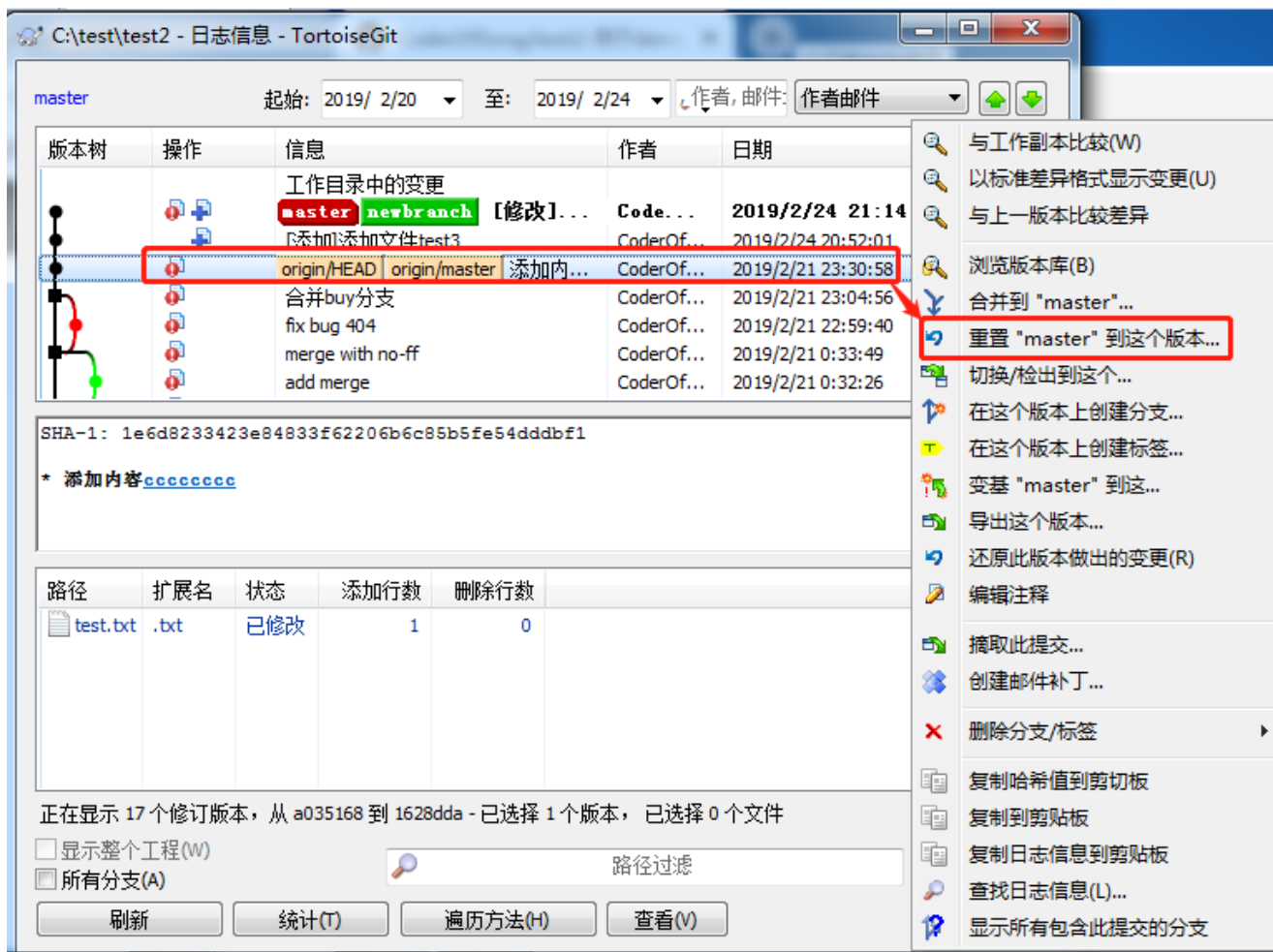


六、版本回滚

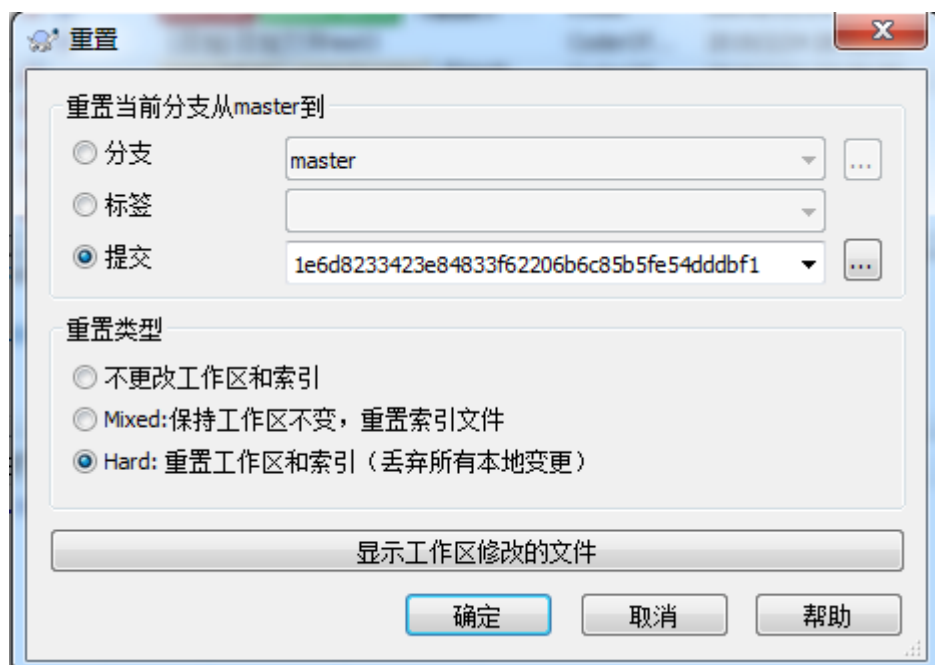
先查看日志，确定想回滚到哪个版本

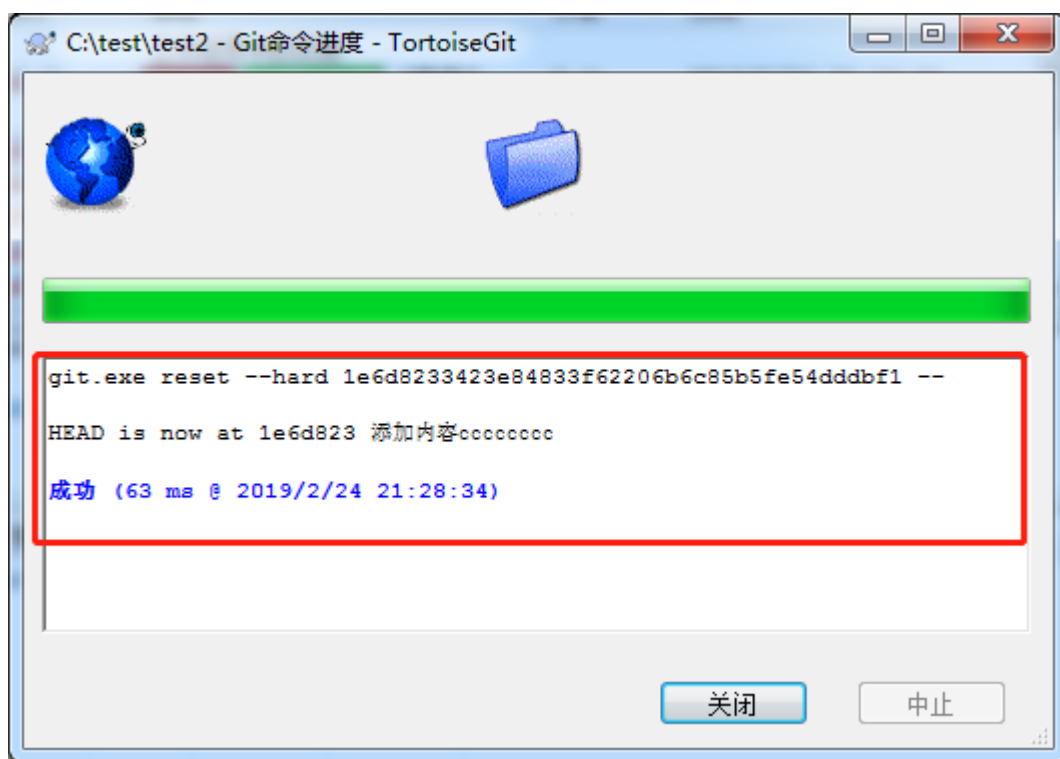


比如想回滚到创建test3.txt文件前, 则点击该步操作所对应的message之前的那一个日志 (add spring demo) 然后右键-->Reset "master" to this..., 表示将当前master分支上的文件回滚到这个版本, 如下图



选择之后，弹出如下弹框，在Reset Type下选择Hard：Reset working.....，点击Ok即可





会发现，此时test3.txt跟test4.txt两个文件都消失了，如下图：

