Git工具的使用

总览：

需要进行下面几步就能把本地项目上传到Github

1、在本地创建一个版本库（即文件夹），通过git init把它变成Git仓库；

2、把项目复制到这个文件夹里面，再通过git add .把项目添加到仓库；

3、再通过git commit -m "注释内容"把项目提交到仓库；

4、在Github上设置好SSH密钥后，新建一个远程仓库，通过git remote add origin github地址 将本地仓库和远程仓库进行关联；

5、最后通过git push (-u) origin master把本地仓库的项目推送到远程仓库（Github）上

**基础命令**

$ git init

命令执行后，该目录下会多出一个.git的目录，这个目录是Git来跟踪管理版本库的，请勿手动修改。但是.git文件夹默认为隐藏属性，因此需要通过设置系统属性，才能看到这个文件夹：具体方法不同版本操作系统设置不一。

版本库(Repository)：工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是stage(暂存区)，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD

touch readme.txt 创建

git add readme.txt 推至缓存区

git commit -m "readme.text提交" 提交至版本库分支并注释（便于日志查看）

git status 查看状态

git diff readme.txt 查看做了哪些更改（未add）

git log 查看更新日志 git log --pretty=oneline 简略版

git reset --hard HEAD^ 回退上一版本 ^^ 两级 git reset --hard 版本号，此句执行对应操作

回退到前100个版本 git reset --hard HEAD~100

git reflog 查看日志与版本号

cat readme.txt 查看文档

git checkout -- readme.txt 撤销 两种情况

rm b.txt 删除操作

git remote add origin +github远程库地址 本地库与远程库关联

git push -u origin main 由于远程库是空的，我们第一次推送master分支时，加上了 –u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令

git push origin master 把本地master分支的最新修改推送到github上

git clone 仓库地址 github远程库-->本地

**创建与合并分支**

**箭头指向代表版本演变**

master分支 主分支

HEAD指向的就是当前分支。

master下，git merge 分支，将分支合并至master上 master-->当前分支，相当于开辟路径

查看分支：git branch

创建分支：git branch name

切换分支：git checkout name

创建+切换分支：git checkout –b name

合并某分支到当前分支：git merge name

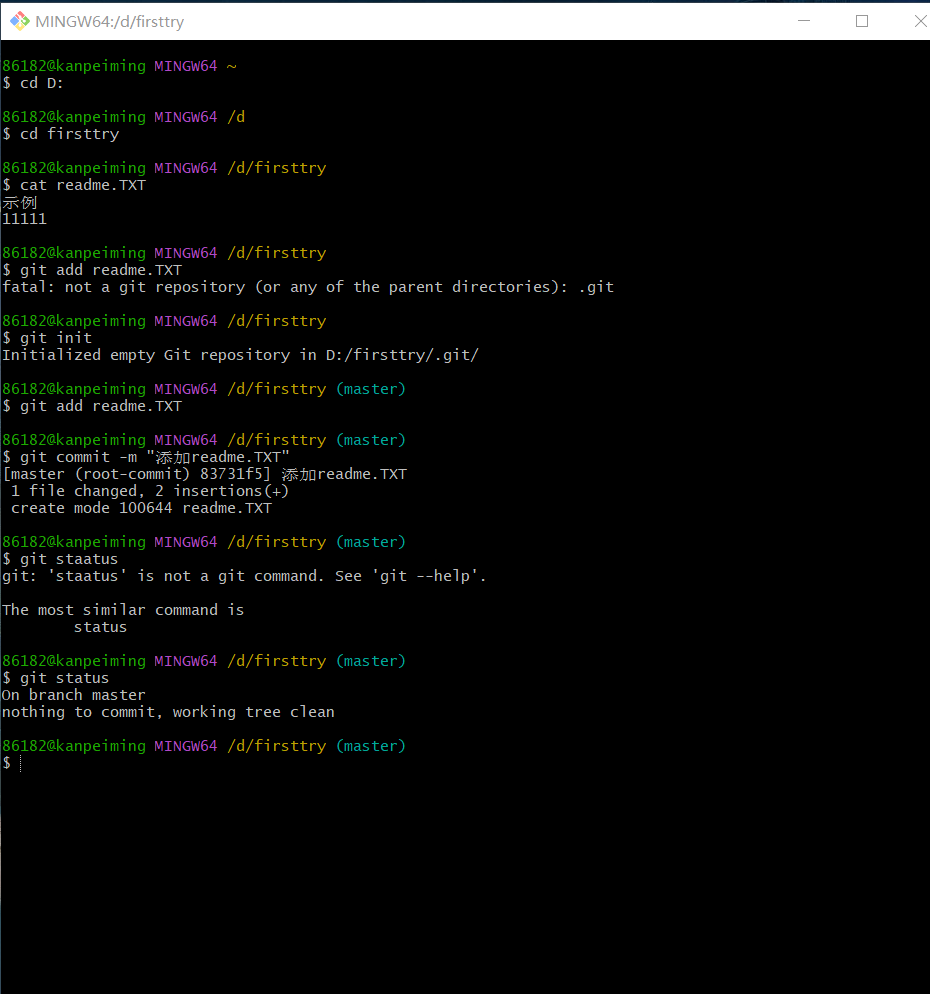
删除分支：git branch –d name

**解决冲突**

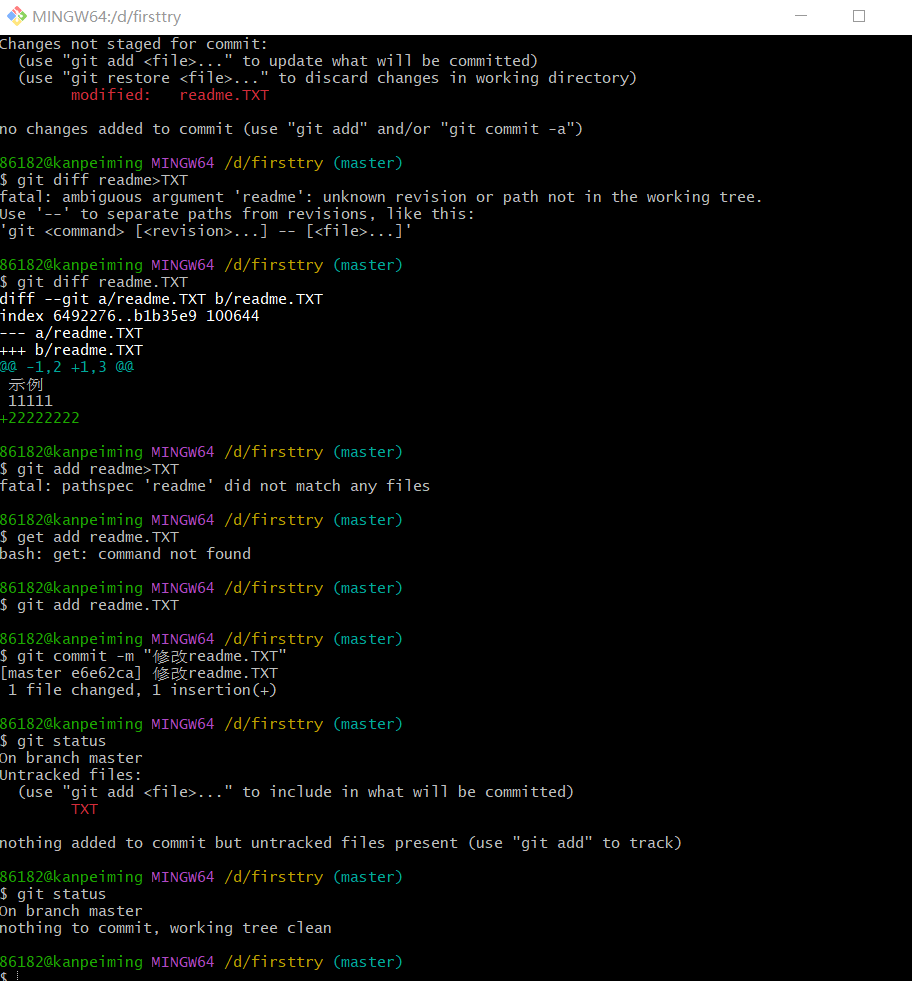
git用<<<<<<<，=======(分割线)，>>>>>>>标记出不同分支的内容，其中<<<HEAD是指主分支修改的内容，>>>>>fenzhi1 是指fenzhi1上修改的内容

**更新远程库、bug分支、分支管理策略（是否留住分支信息）**

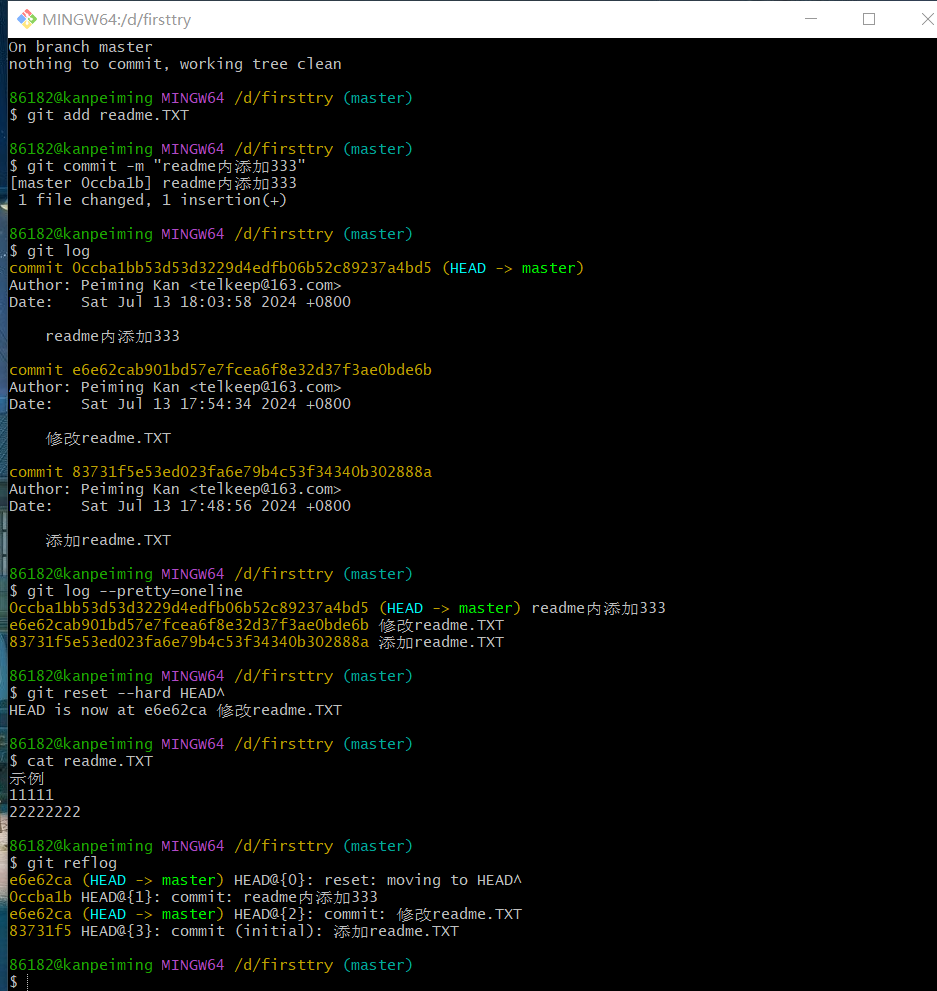
**实验过程**



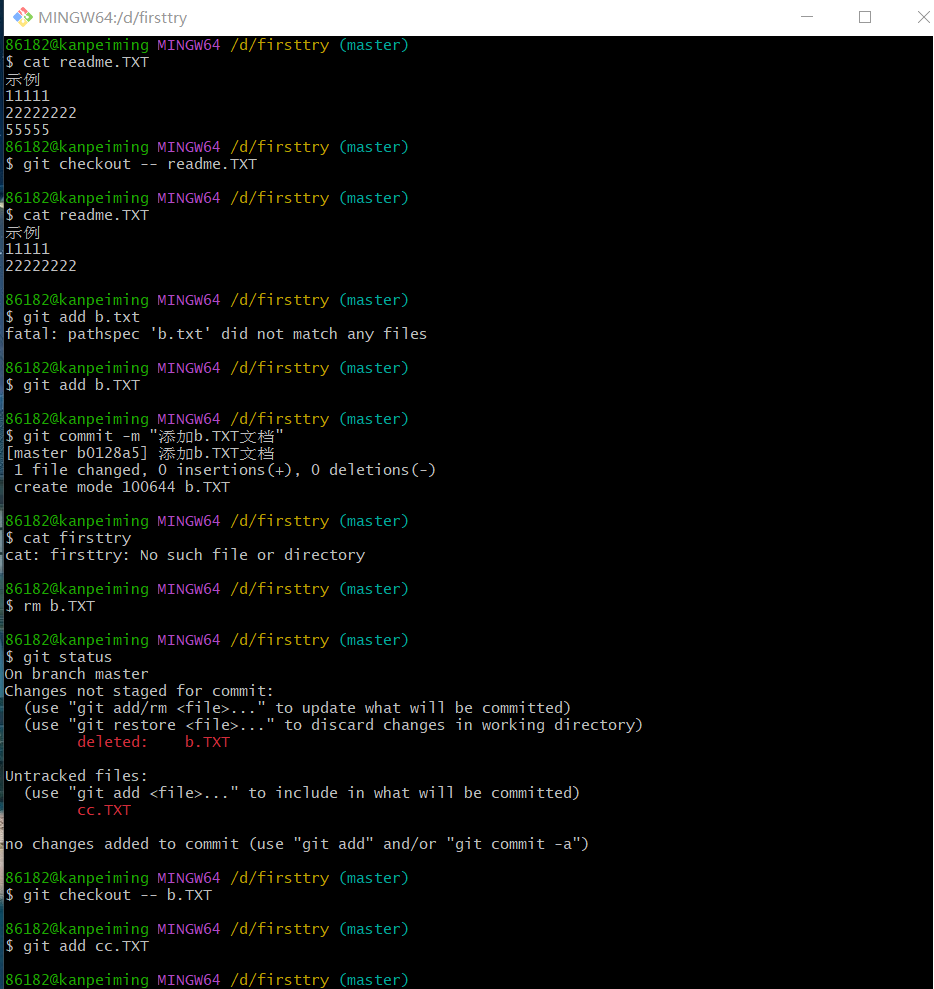
创建firsttry版本库，添加readme.TXT至缓存区，再存入版本库。



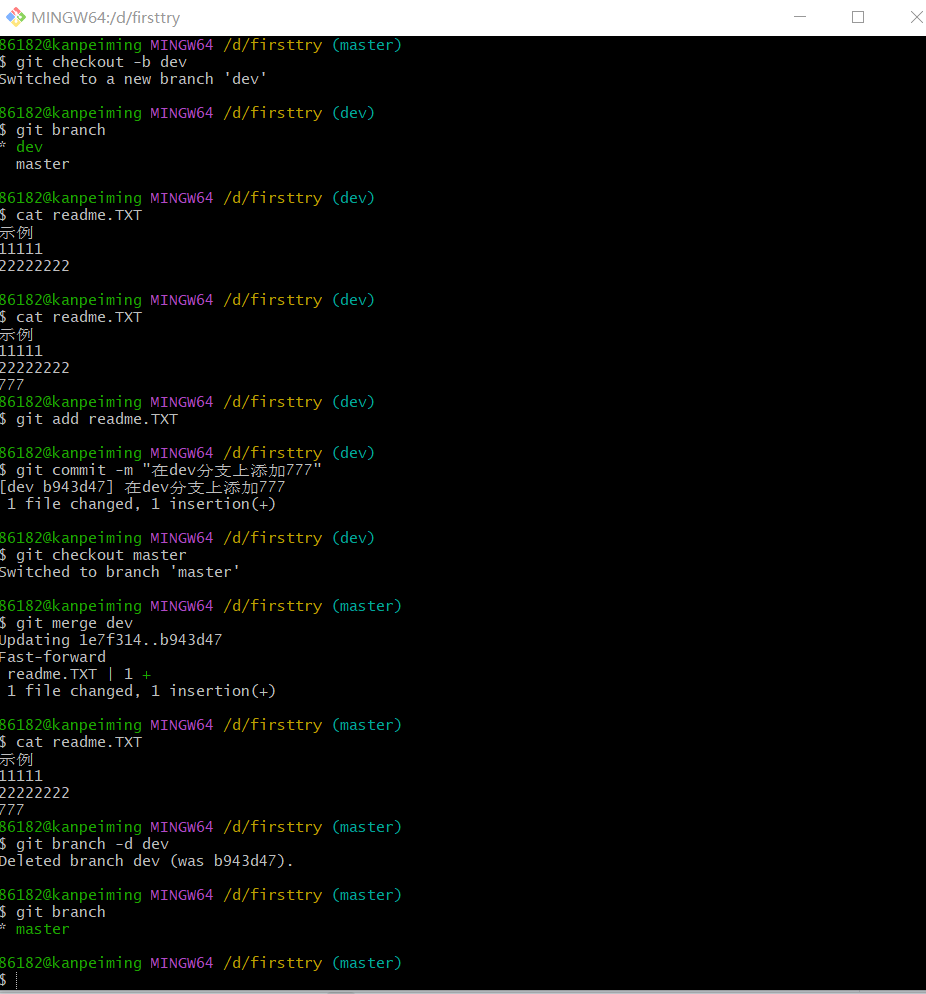
使用cat指令显示add与commit步骤的作用



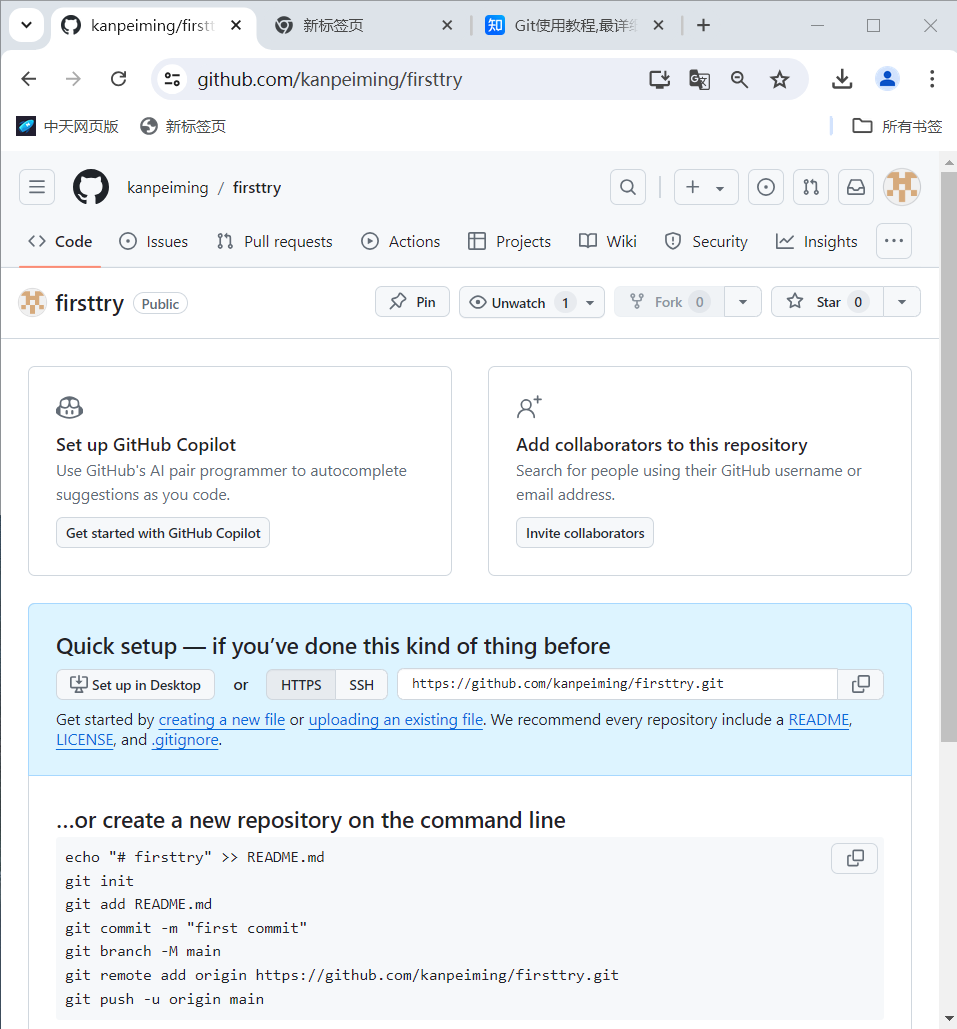
显示日志，版本回退，查看版本号



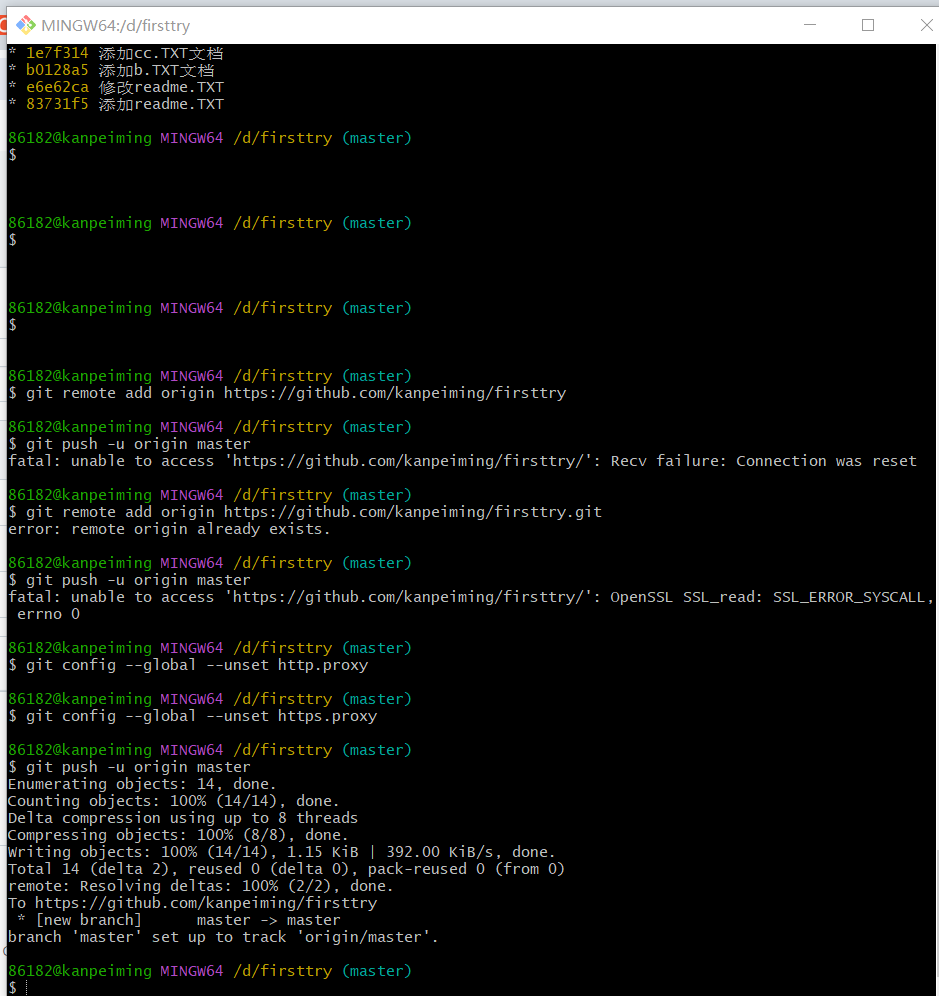
撤销添加操作，添加文件与删除，撤销删除。



创建与合并分支



创建远程库



连接本地库与远程库，将文件上传，如下：

