

《软件工程》

实验报告三

|  |  |
| --- | --- |
| 班 级： | **计163** |
| 学 号： | **10161785** |
| 姓 名： | **林宇翩** |
| 指导教师： | **阮彤** |

信息科学与工程学院

2019年 4月

**实验三 Git实践**

**实验者：林宇翩 实验日期：2019/4/18 实验地点：信息楼215**

1. **Git与SVN**

Git是分布式版本控制系统，而SVN是集中式版本控制系统，主要区别在于代码版本的存放位置不同。SVN的代码版本集中存放于中央服务器，即每次对代码处理都要从中央服务器拿出最新版本处理后再存放回中央服务器，这种方式的操作成本高并且需要联网的工作环境；而Git的代码版本直接存放在每个人的电脑中，即每个人都是“中央服务器”，不需要每次工作都下载/上传最新版本到服务器上，每个人都拥有一个完整的代码版本，相当于对代码进行了备份，而对于集中式版本控制系统，如果中央服务器出现故障，则版本库就会丢失，安全性没有Git高。

1. **Git的优点**

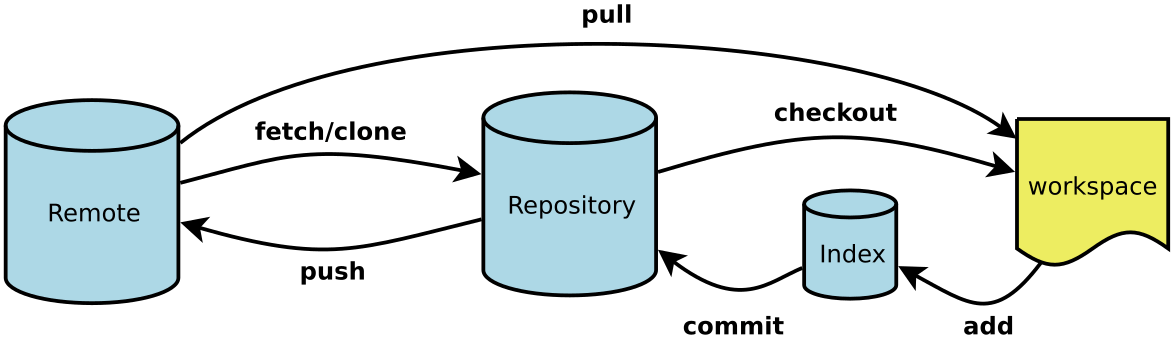
a.能够对文件版本控制，多人协作开发；

b.拥有强大的分支特性，所以能够灵活地以不同的工作流协同开发；

c.分布式版本控制系统，即使协作服务器宕机，也能继续提交代码或文件到本地仓库，当协作服务器恢复正常工作时，再将本地仓库同步到远程仓库；

d.当团队中某个成员完成某个功能时，通过pull request操作来通知其他团队成员，其他团队成员能够review code后再合并代码。

1. **Git的基本工作流**



a.克隆或者新建本地仓库（Repository）

b.在git版本控制的目录下创建或修改文件

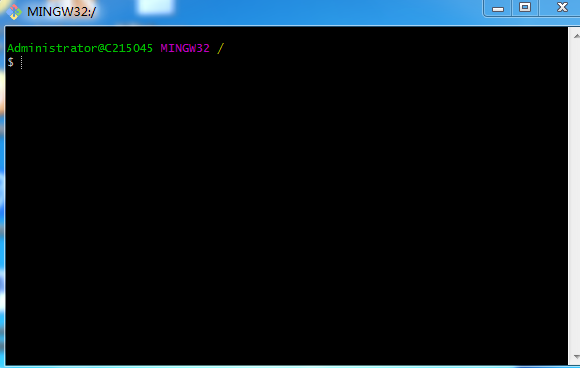
c.使用git add 命令添加新创建或修改的文件到本地缓存区（Index）

d.使用git commit 命令提交到本地仓库（Repository）

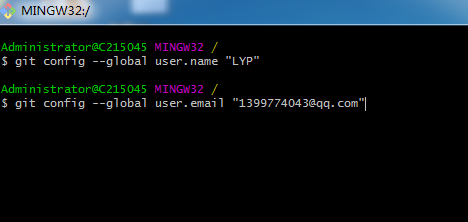
e.使用git push 命令将本地代码库同步到远端仓库(Remote)

1. **Git的安装与使用**

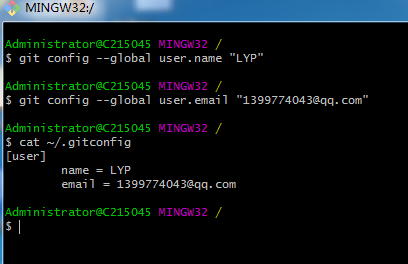
**1).**从官网下载并安装Git，安装成功之后打开”Git Bash”,如下图：



**2).**安装之后，需要使用git config命令和全局参数--global设置user.name和user.email，作为在提交commit时的签名，

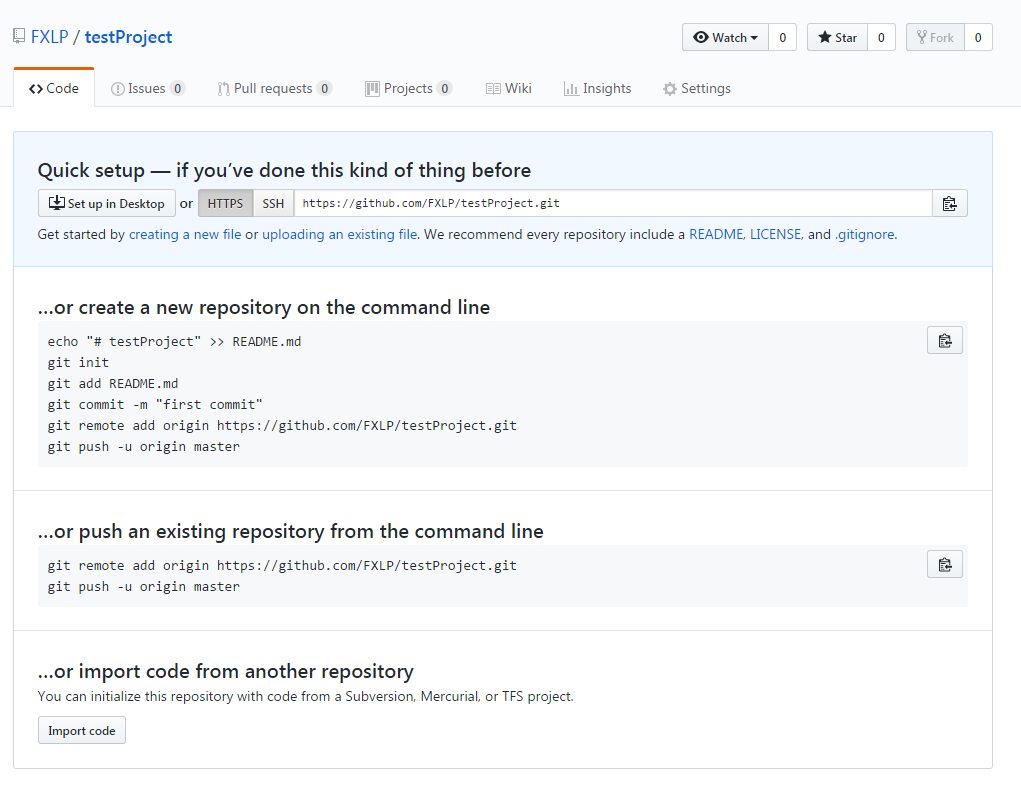


执行上面的命令后，会在目录下建立一个.gitconfig文件，使用vim或cat查看文件：



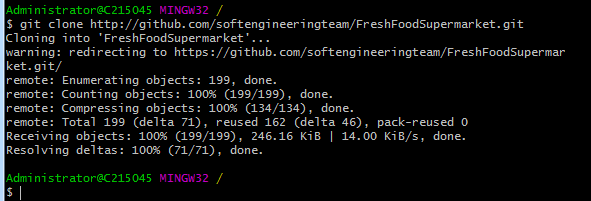
**3).**注册帐号，并新建一个远端仓库”testProject”:

URL:<https://github.com/FXLP/testProject.git>



**4).**使用git clone [远端git仓库]命令克隆目标项目：

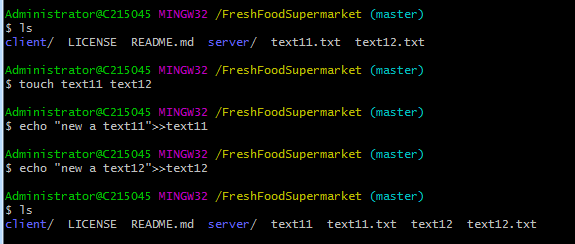
URL：<https://github.com/softengineeringteam/FreshFoodSupermarket.git>



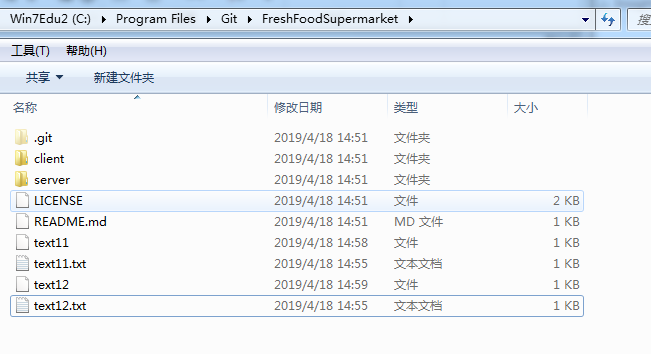
**5).**使用cd 命令进入FreshFoodSupermarket目录，新建text11.txt和text12.txt

修改文件，使用vim编辑内容，也可以直接echo添加内容

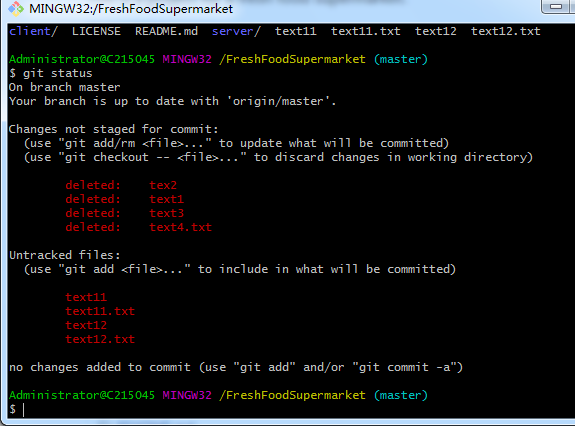




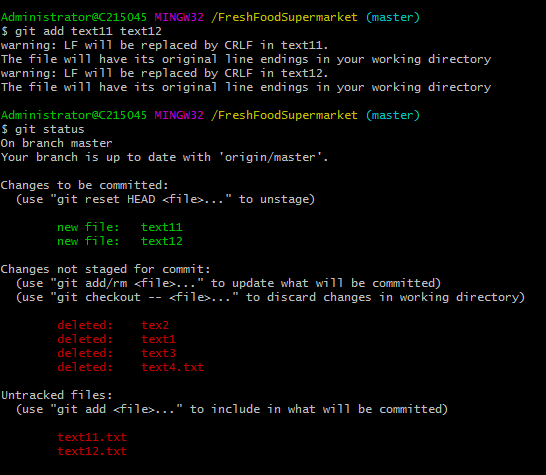
可以看到文件目录下新增了内容：



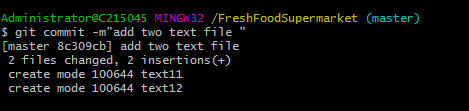
**6).**使用git status命令显示有变更的文件



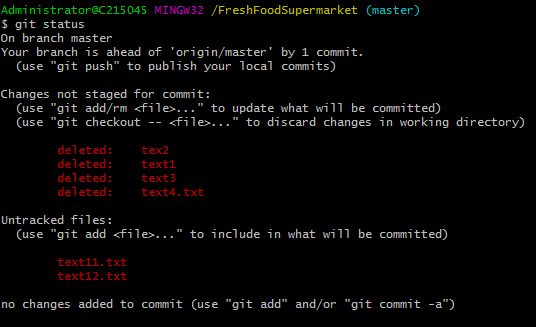
使用git add命令将处于untracked状态的2个文件text11和text12加入到缓存区（Index）。再次执行git status会发现新的变化。



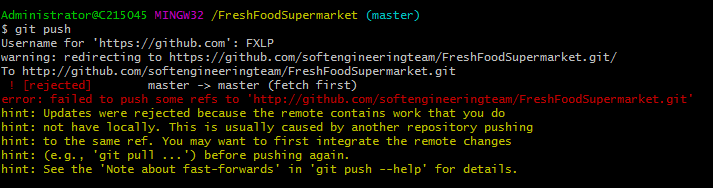
**7).** 现在已经为commit到本地仓库做好了准备。（可以使用git diff--cached命令查看缓存区和上一次commit差异，看哪些文件被修改了。）使用 git commit -m “本次提交内容的摘要” 命令将缓存区文件提交到本地仓库。



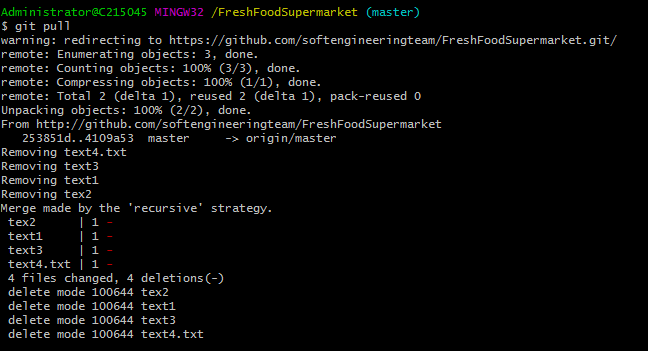
再次执行git status会发现新的变化。



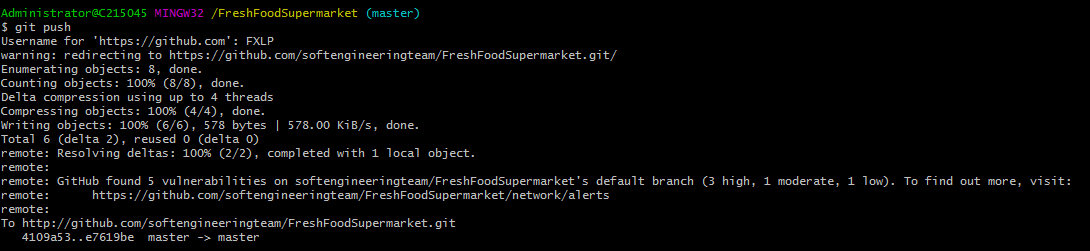
**8).** 使用git push命令将本地仓库的文件更新到远端仓库中。



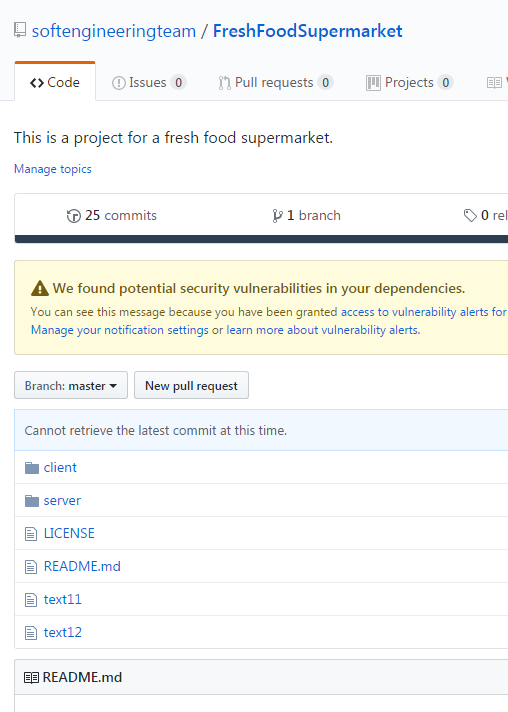
但是提示在push之前需要再pull一次，于是再pull一下



再重新push一遍



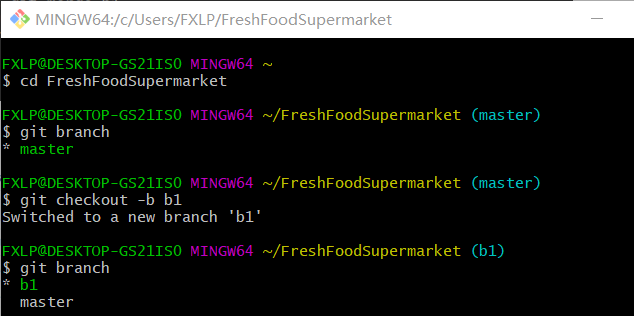
可以看到gitee的项目已新增text11和text12文件：



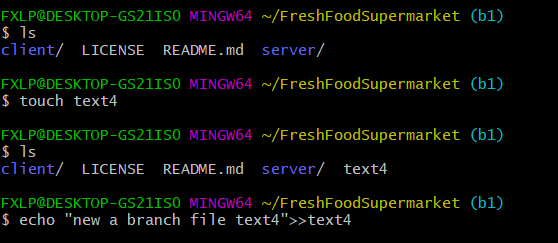
**9).**分支与合并

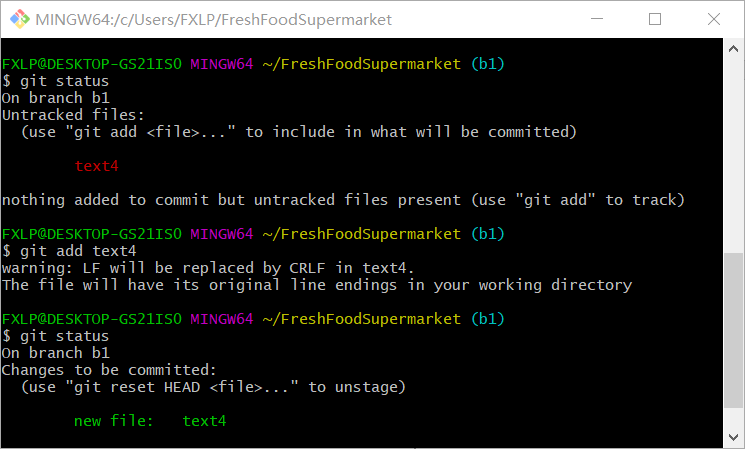
Git的分支可以让你在主线（master 分支）之外进行代码提交，同时又不会影响代码库主线。分支的作用体现在多人协作开发中，比如一个团队开发软件，你负责独立的一个功能需要一个月的时间来完成，你就可以创造一个分支，只有把该功能的代码提交到这个分支，而其它成员仍然可以继续使用主线开发，你每天的提交不会对他们造成任何影响。当你完成功能后，测试通过在把你的功能分支合并到主线上。

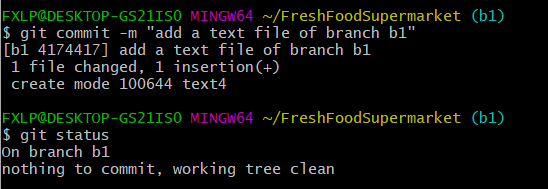
使用git checkout -b [分支名] 命令新建一个分支并跳转到这个分支目录下。使用git branch 查看所有本地分支。



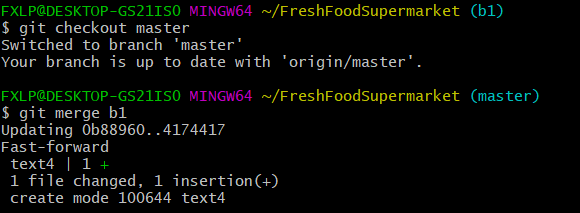
在b1分支下，创建并写入文件text4，并重复上述操作，git add—>git commit，将text4 commit到本地仓库。



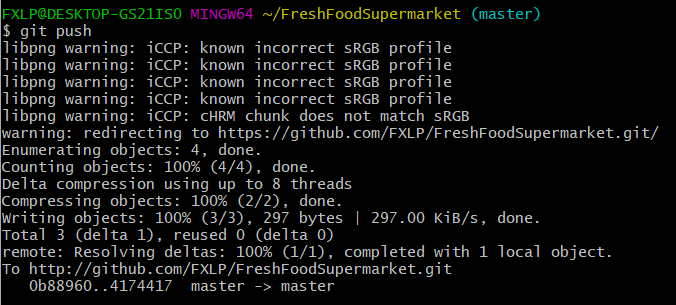




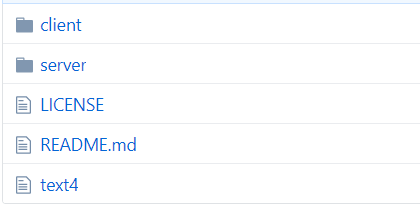
使用git checkout master切换到主线上，即，使合并分支操作的目标分支是master分支，使用git merge [源分支] 命令将分支b1合并到master分支上。



再次push到远端仓库：



查看远程仓库gitee已更新：



1. **收获与总结**

通过本次Git的实践，从安装到真实项目的使用，掌握了Git的理论知识和基本操作，对于版本控制系统和团队合作开发有了更深刻的理解，有助于以后的团队开发和自身代码的高效管理。