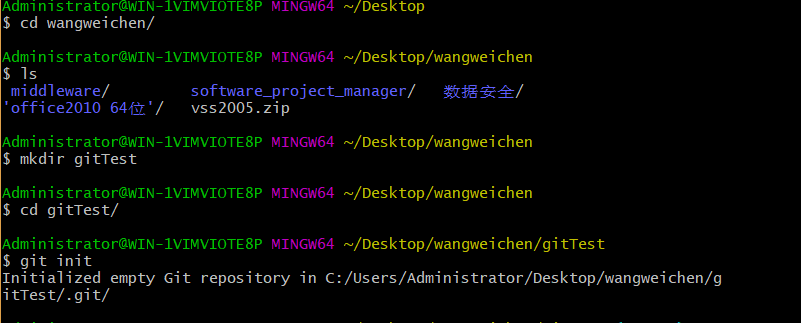
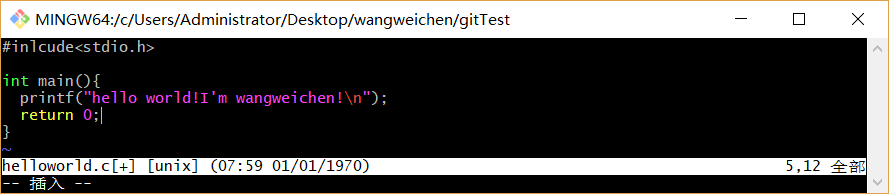
# 实验四 使用GitHub进行版本管理

1、创建版本库

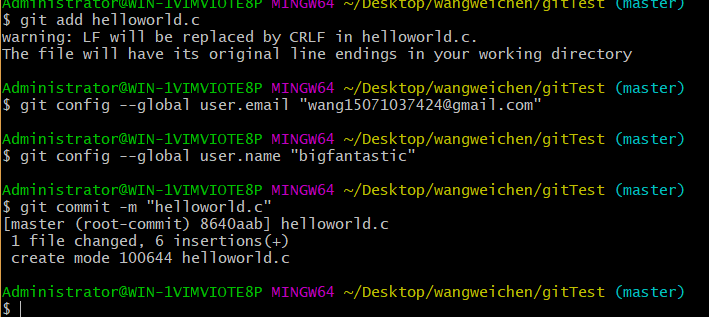
1.1 安装git，去官网直接下，过程略

1.2创建目录，作为仓库，然后再git init初始化仓库，如图：

1.3使用vim新建一个helloworld.c文件

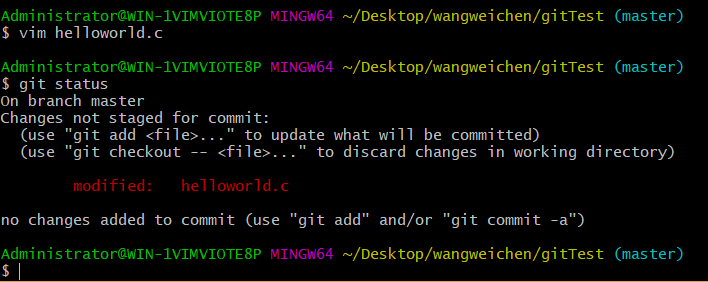


1.4 往代码仓添加刚刚新建的文件，同时先配置好自己的邮件和用户名，以便后面我们commit，push到远程库。然后再进行提交commit。

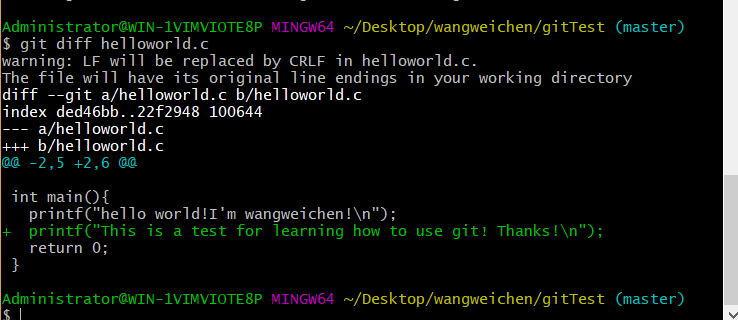


2、提交修改

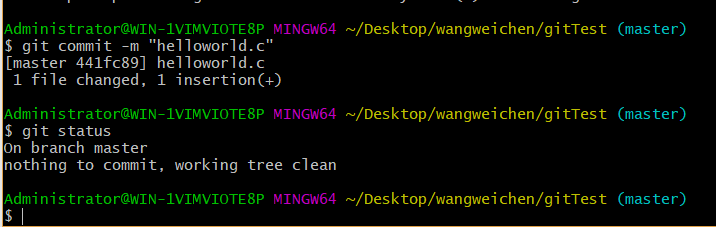
2.1 我们先简单修改一下helloworld.c的代码，然后用git status查看仓库的状态。这提示我们已经进行的修改，但是并没有提交修改。



2.2 使用git diff 来看看我们做了什么修改。

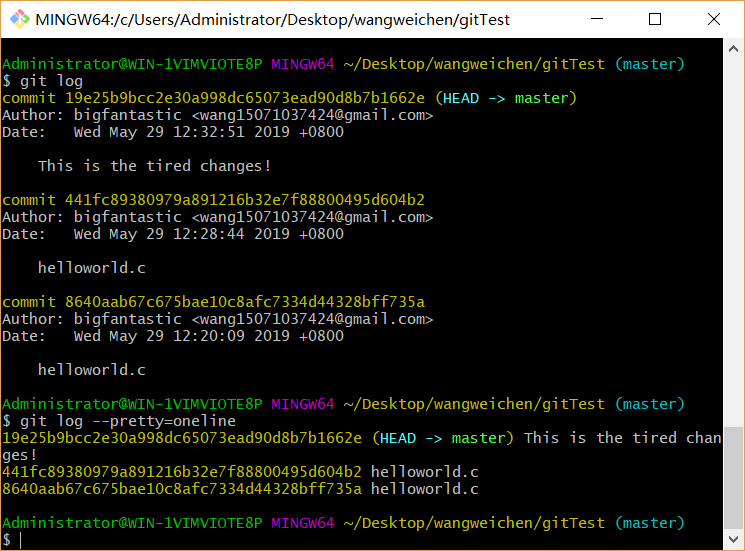


2.3 和第1大步骤中一样，我们提交上去，最后再看看仓库状态。

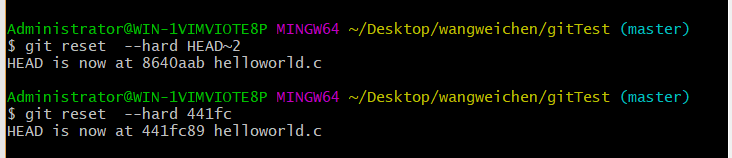


3、版本回退

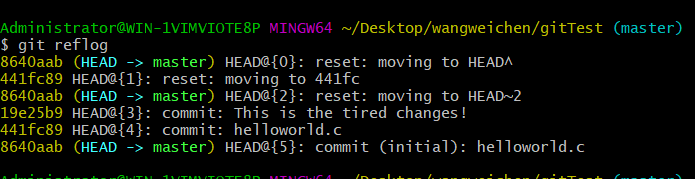
3.1 我们先进行第二次的修改，然后在看看最近到最远的的提交日志。



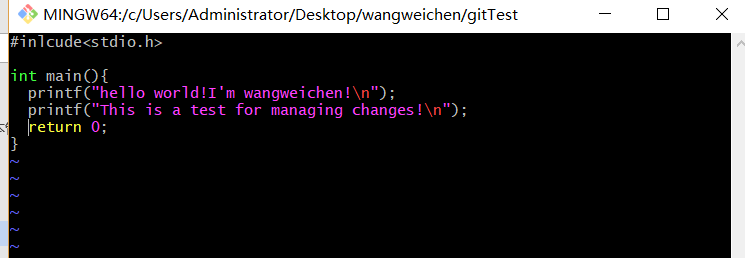
3.2 我们用git reset – hard HEAD~ 来回退到我们想要到的版本。



3.3 我们用git reflog看看我们每一次的命令。



4.管理修改

4.1我们先修改一次然后add ,但是不commit，然后再进行一次修改，直接commit。

4.2 会发现只有我们第一次add上去的提交了，后一次修改没提交，所以我们修改一次add一次再提交就相当于两次修改合并后一起提交了！

5.撤销修改

5.1还没有add也没有commit。直接git checkout –- helloworld.c..(--很重要！不然就是切到了另一个分支)

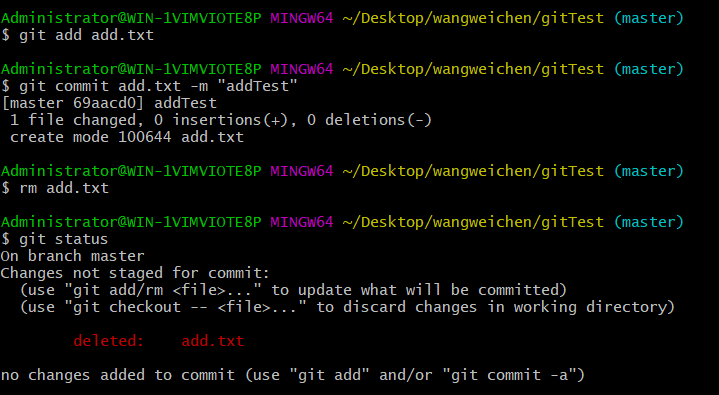
5.2 如果已经add但是没有commit，可以先git stauts查看一下，然后git reset HEAD readme.txt 可以把暂存区的修改撤销掉，重新放回工作区。git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区。当我们用HEAD时，表示最新的版本。

git status 查看一下，现在暂存区是干净的，工作区有修改然后git checkout -- readme.txt

5.3 如果推到了远程版本库，你就GG了。但是如果你的那一分支被弃用了影响就不大。

6.删除文件

6.1 添加一个add.txt,然后删除。会发现工作区和版本库不一致了。



6.2 如果确实是要删除就git rm add.txt然后git commit.

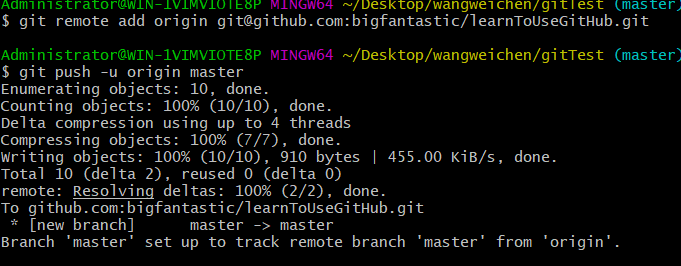
6.3 如果是误删，就git checkout – add.txt

7.添加远程库

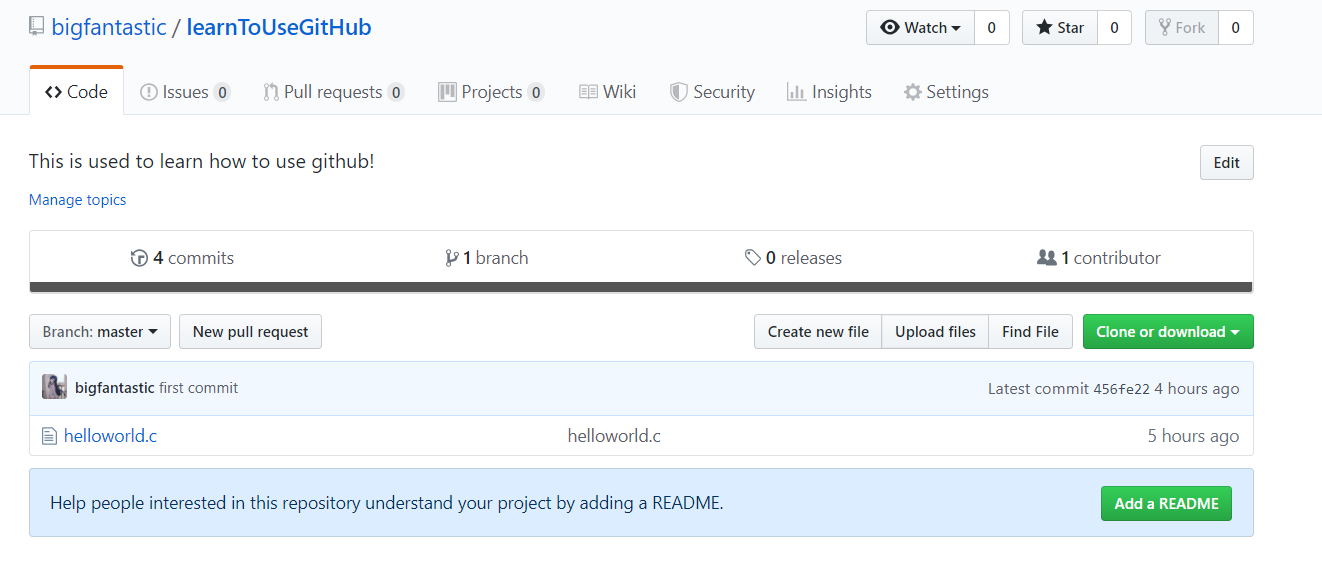
7.1在GitHub用自己账号先新建一个repository，然后

git remote add origin git@github.com:bigfantastic/learnToUseGitHub.git

git push -u origin master

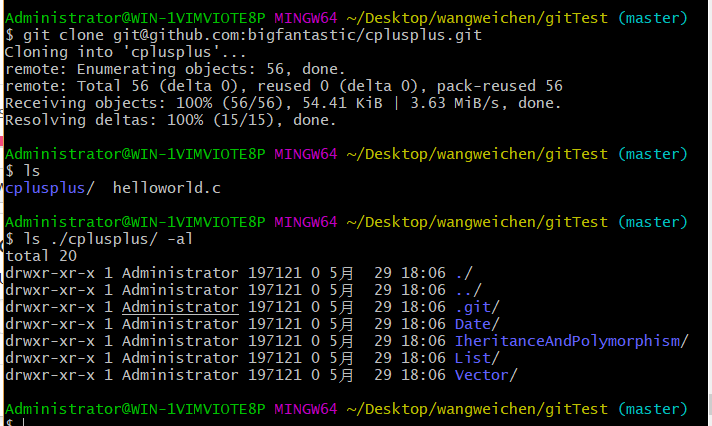


7.2 每次只要有必要都可以push上去，如图刚刚push的结果。



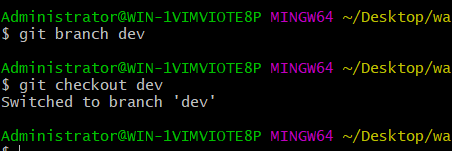
8.从远程库克隆

8.1使用git clone指令克隆下来我以前一个远程库。如图



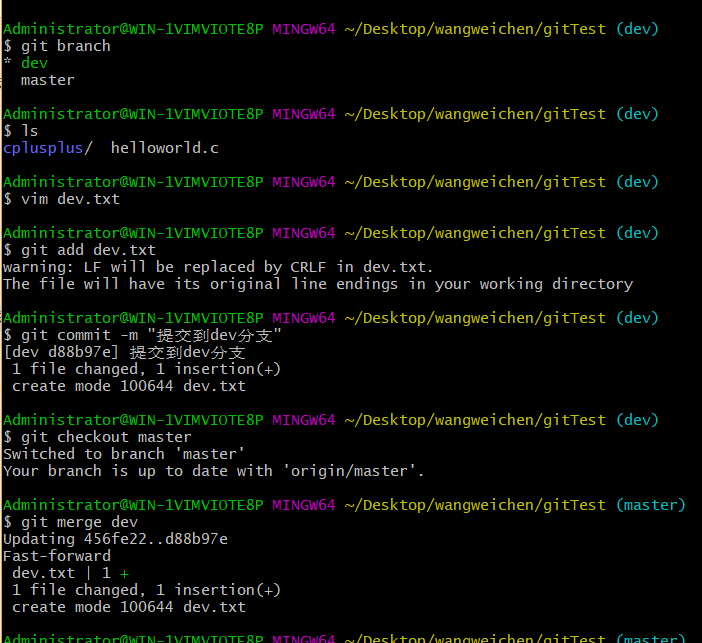
9.分支管理：创建和合并分支

9.1先创建一个新分支dev,然后再切换。

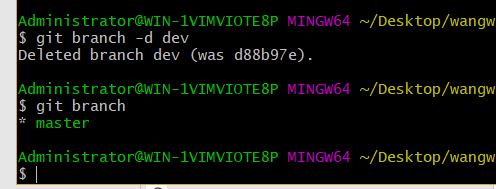


9.2 在新分支中创建一个dev.txt然后提交上去。

9.3切换到master分支，然后将dev分支的修改合并入master，如图



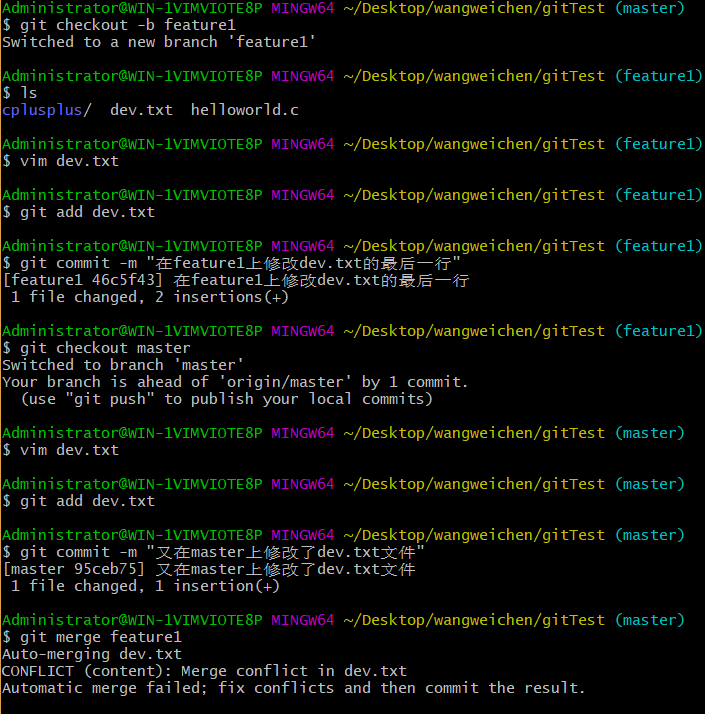
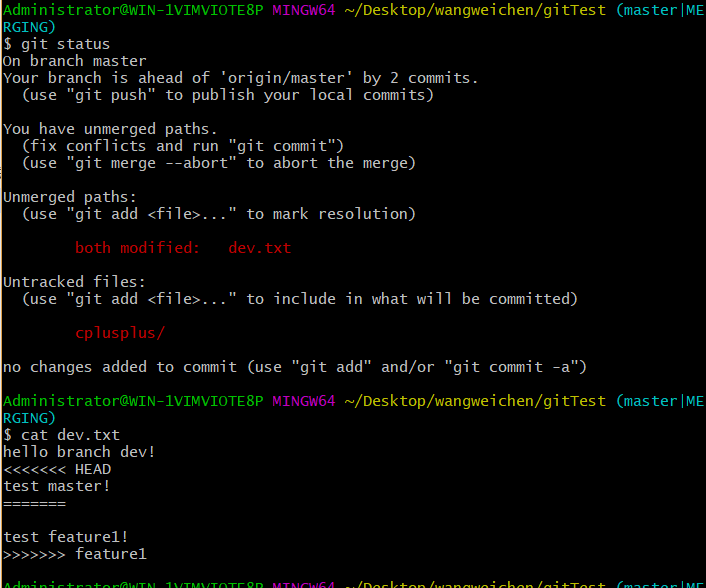
9.4 删除掉dev分支



10.解决冲突

10.1 创建一个新的分支feature1，然后将dev.txt在该分支下进行修改，添加了一段test feature1，然后再提交上去。

10.2切换到master分支然后修改该文件，在master上面提交，然后将分支合并，我们能看到冲突的文件，冲突内容如下;

10.3 然后我们编辑dev.txt的文件，改成我们想要的样子。最后查看分支合并图，如图



以上就是简单的对Git的应用学习！

