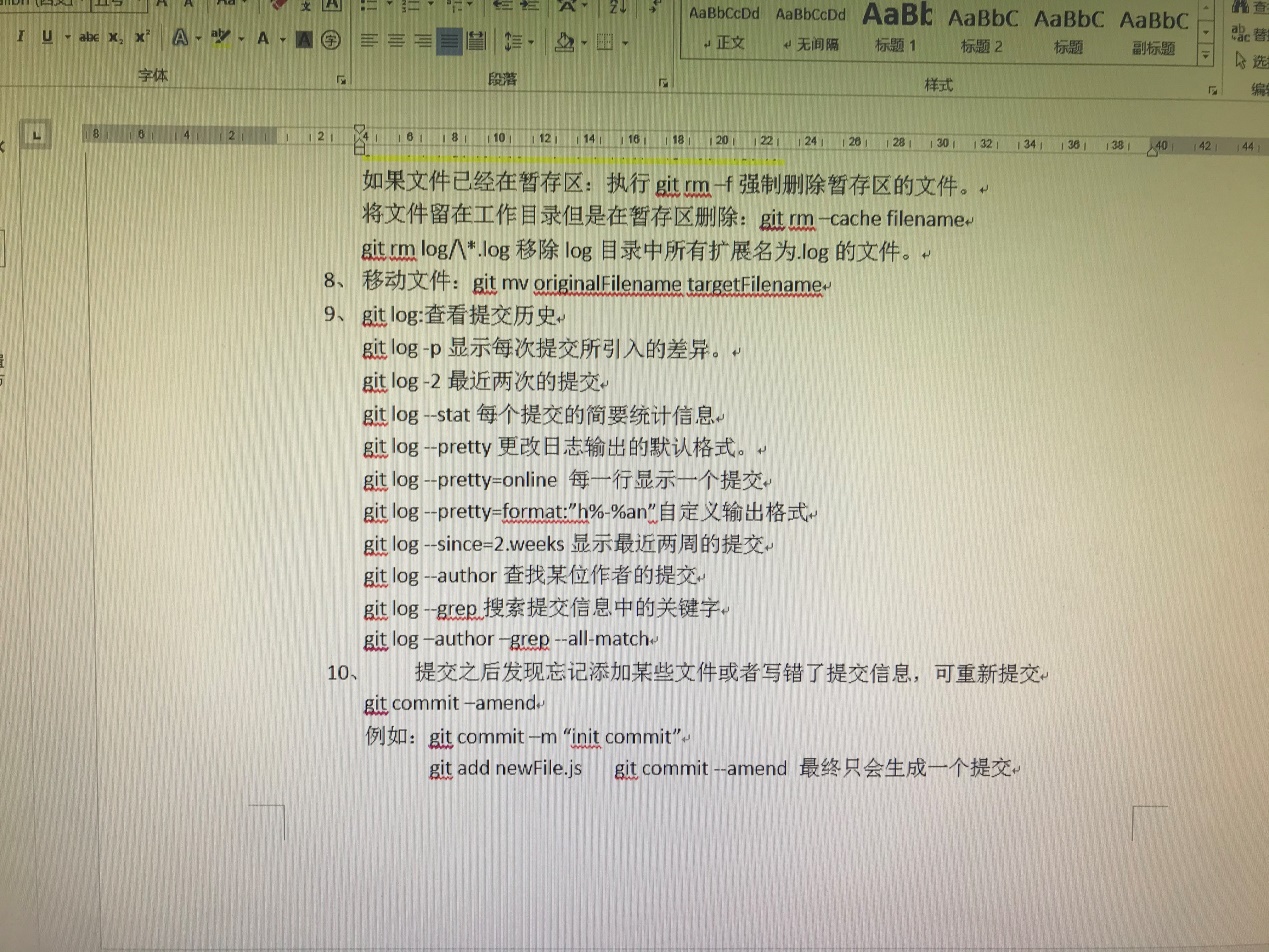
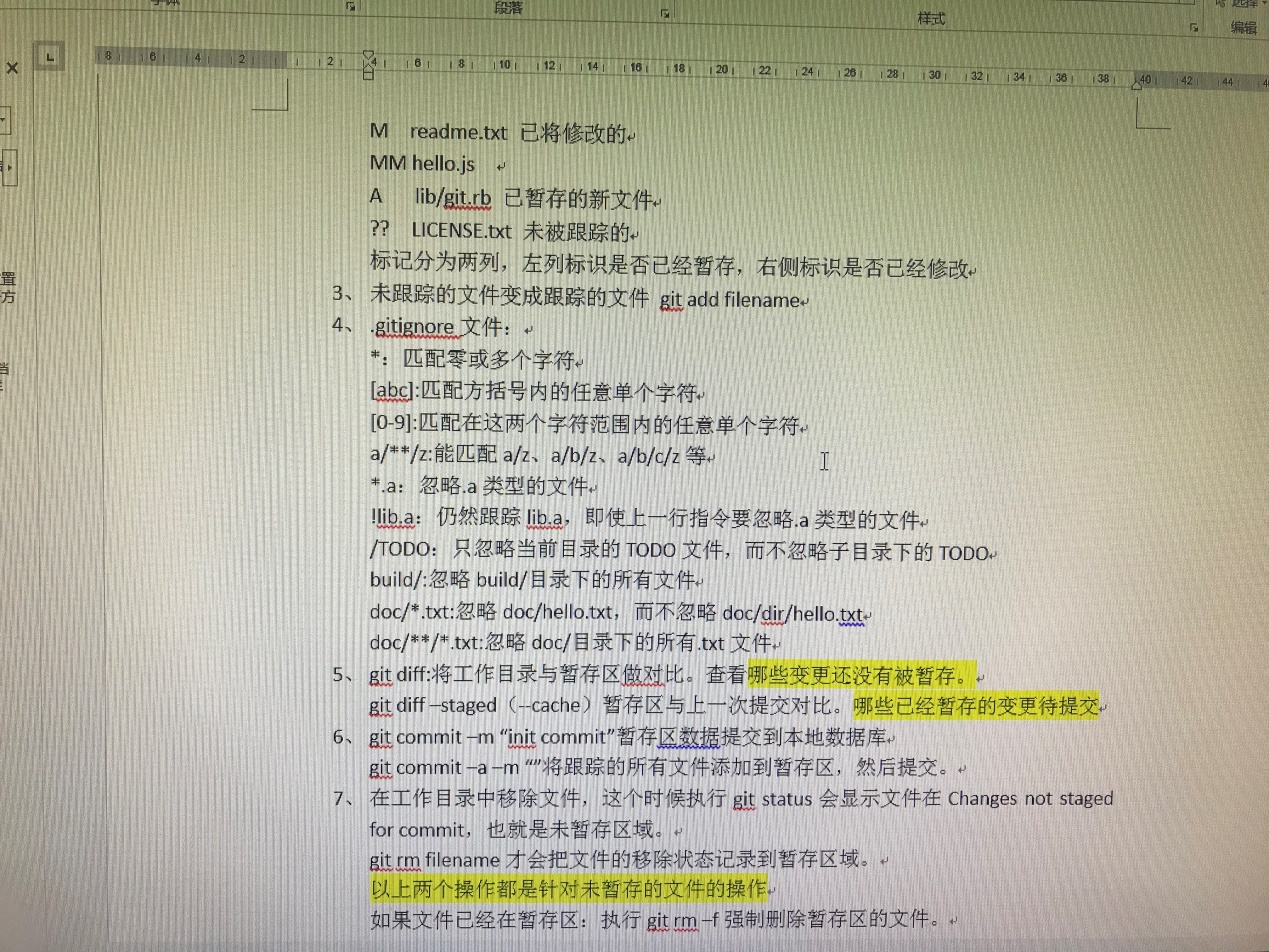
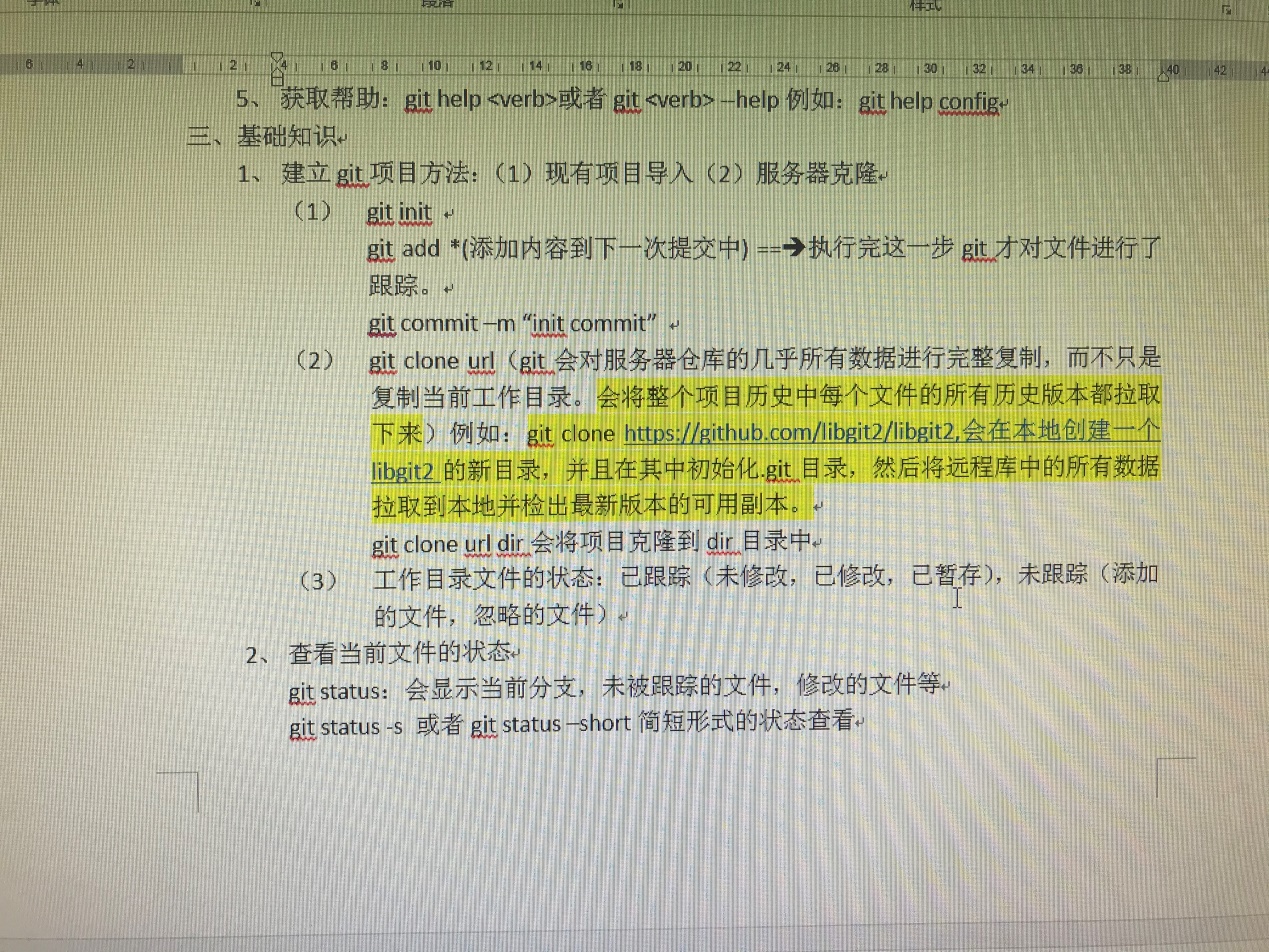
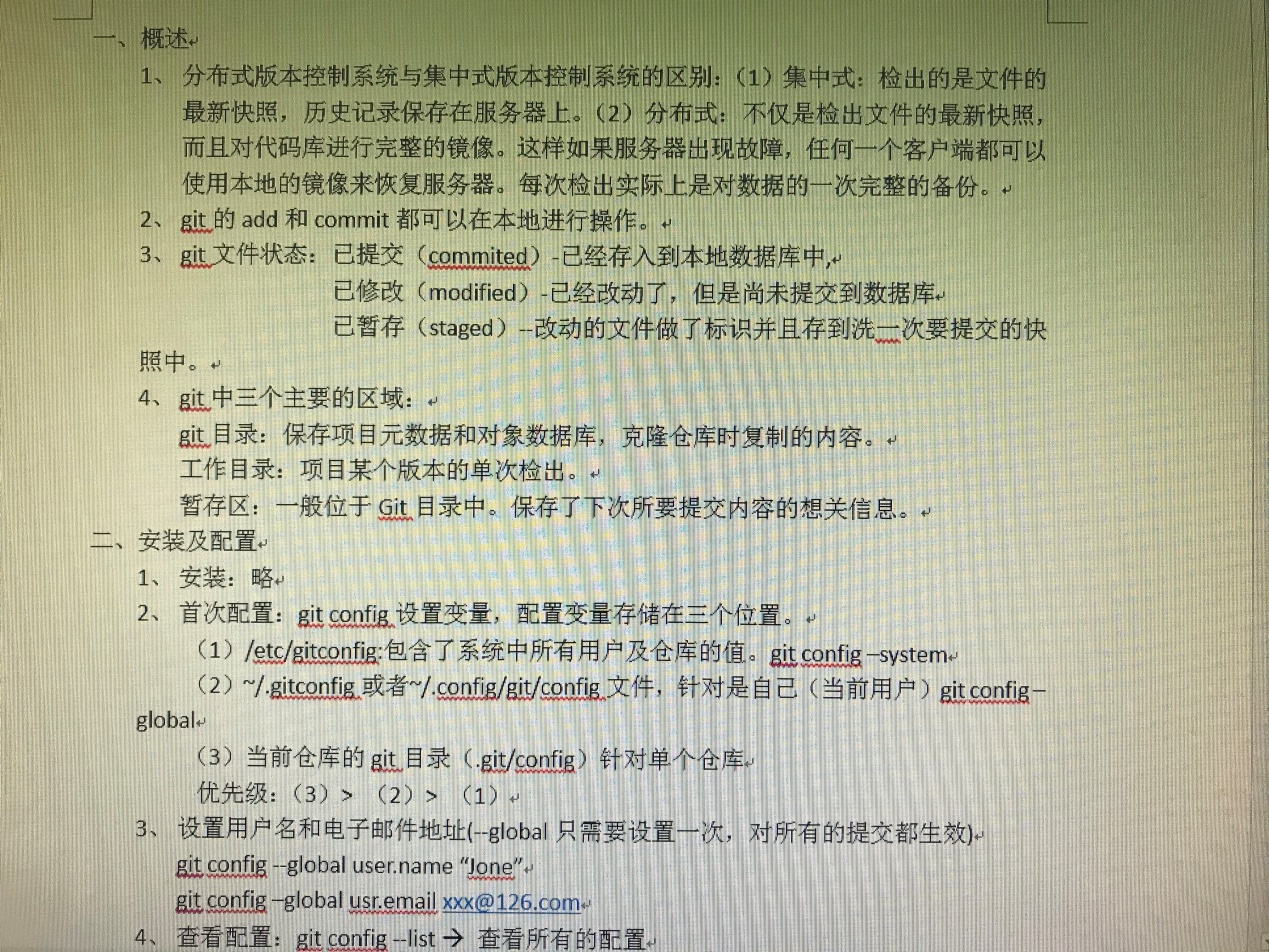
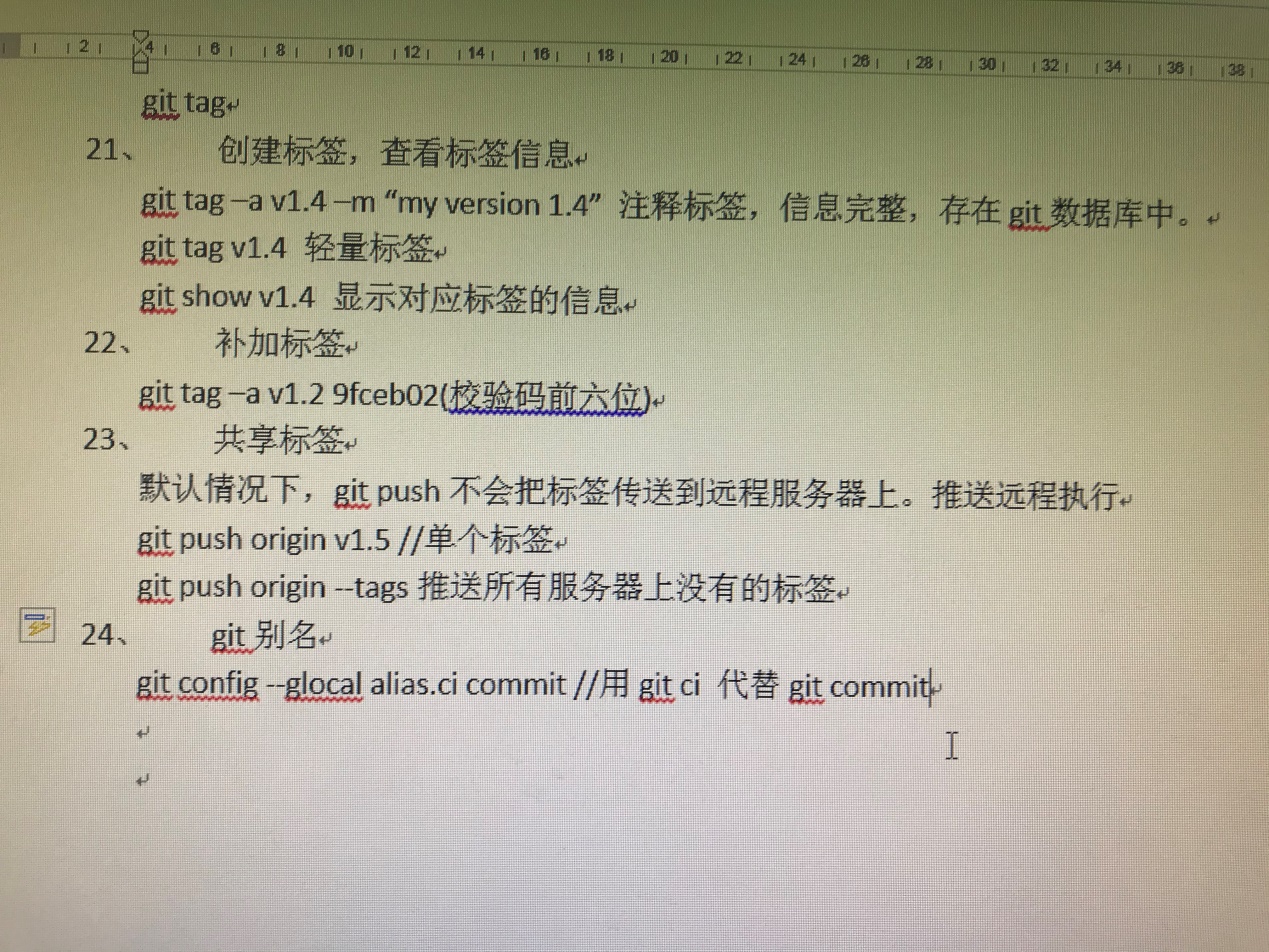
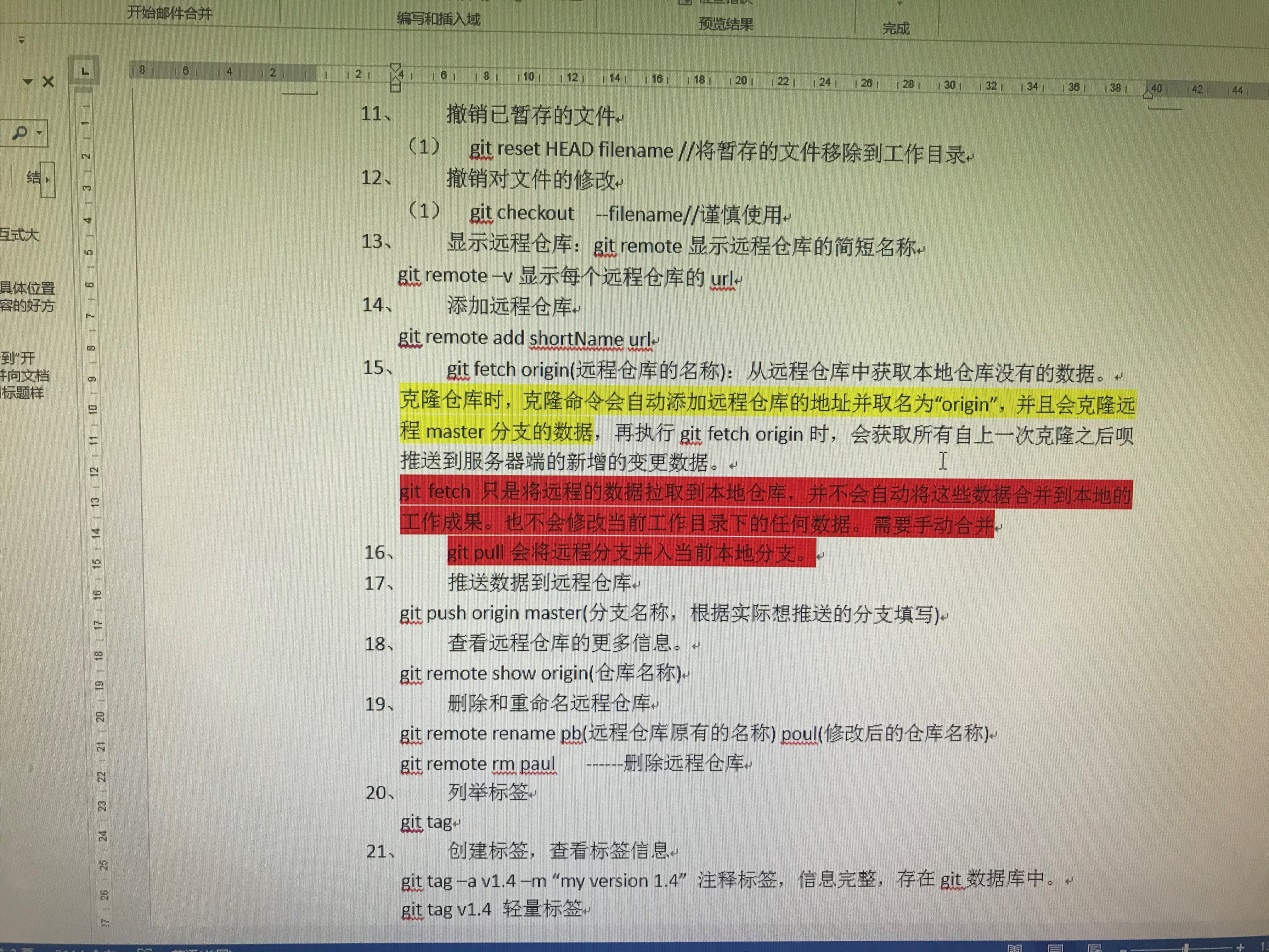
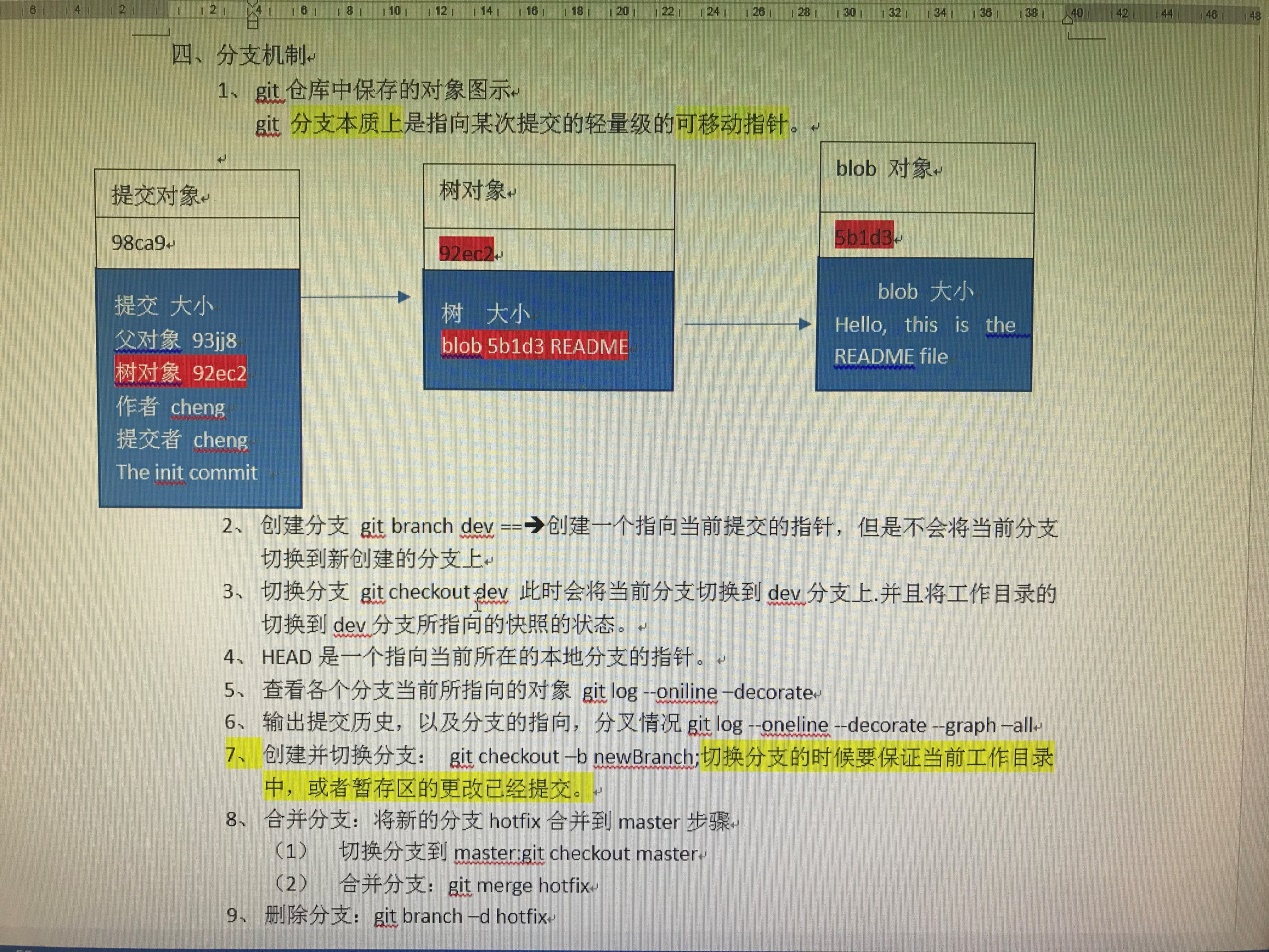
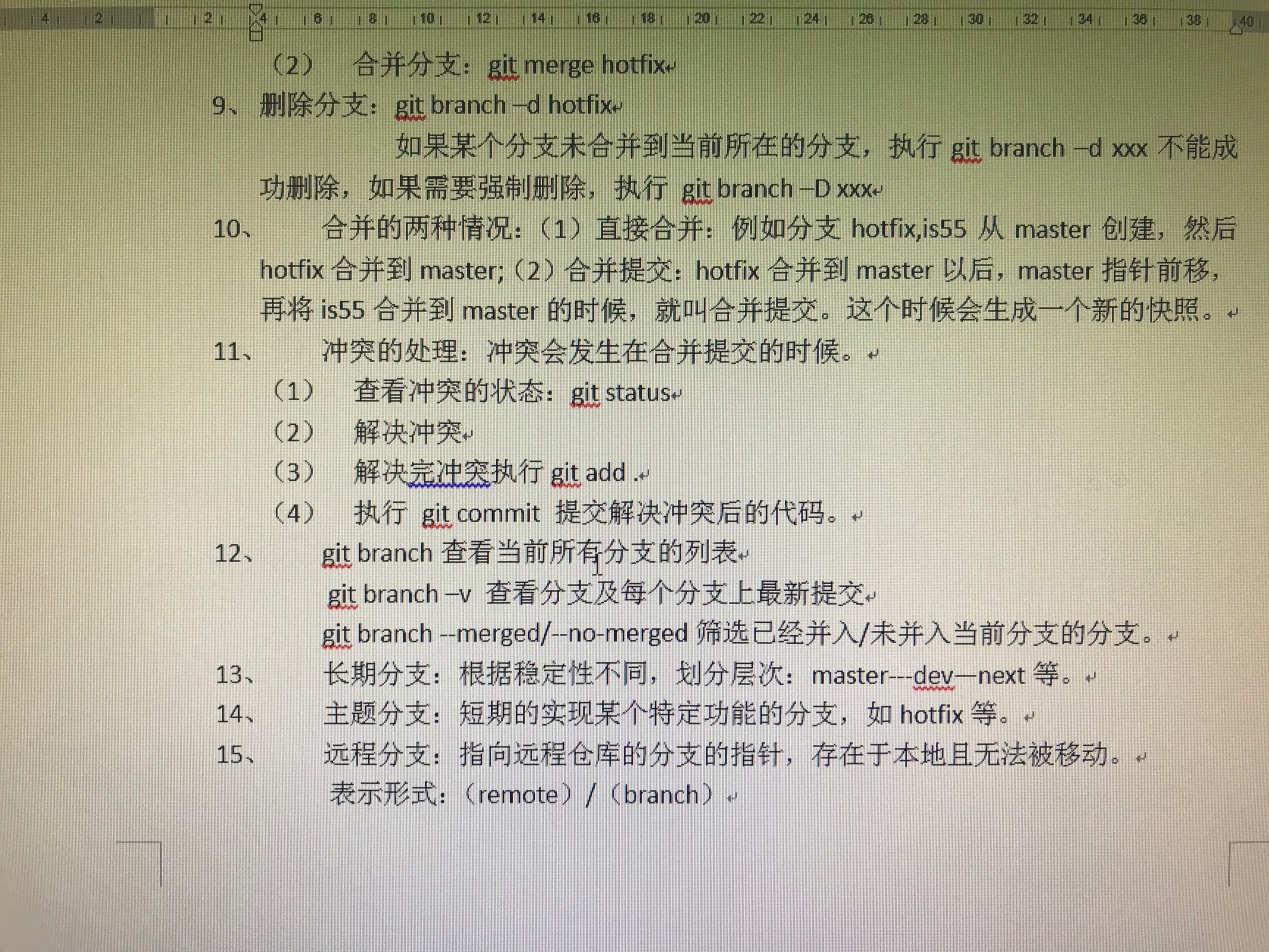
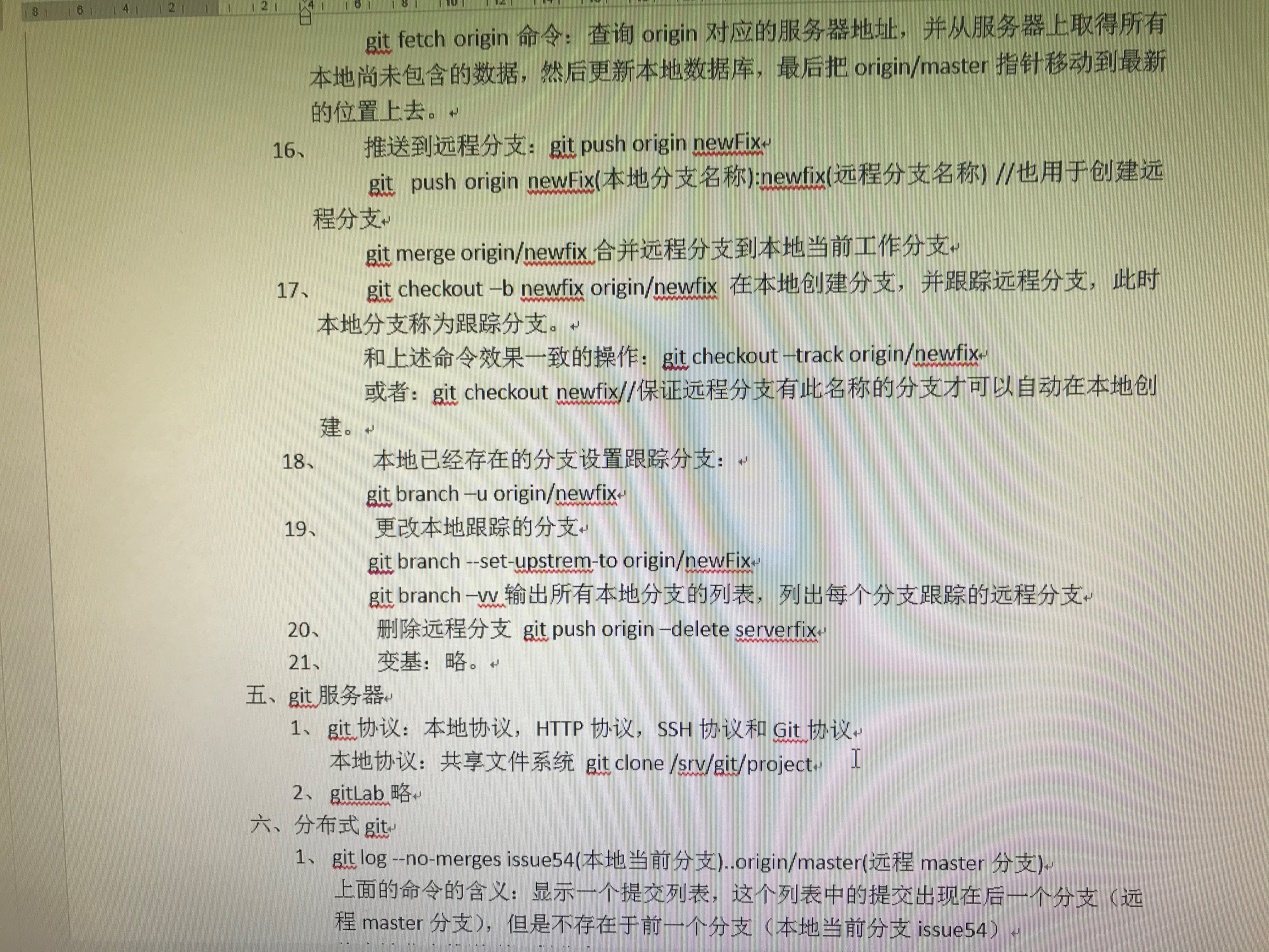
git学习笔记











2、将本地分支推送到远程分支

git push –u origin localBranchName:remoteBranchName

七、git工具

1、查看某次提交

git show 7ab45c

2、git add –i 进入交互式shell模式

3、储存变更：就是有有修改的文件没有提交，处在工作目录或者暂存区内，这个时候 切换分支。

(1)git stash (2)git checkout fixBranch

4、查看储藏 git stash list

5、应用储藏： git apply stash@{3}

原来在暂存区的储藏在应用后，会保存在工作区内，不会重新进入到暂存区，要想重新回到当时的状态，执行git stash apply –index

apply 只会应用储藏的内容，而这些内容依然保存在栈上，要想删除执行git stash drop

也可以执行git stash pop来应用并直接删除这些储藏。

6、git stash --keep-index不储藏已经暂存的内容。

git stash --include-untracked 储藏为跟踪的文件。

git stash --patch 不储藏已经修改过的文件。

7、储藏的分支如何继续开发呢？---在储藏上开分支，继续工作，这样就能恢复储藏的内容

git stash branch newBranch

8、清除工作目录的文件：git clean(慎用，这会清除所有未跟踪的文件，默认删除未被忽略的未跟踪文件)

git stash --all 删除工作目录的所有内容并且将内容保存起来

git clean –f –d 将文件全部删除并且清空所有的子目录

git clean –d –n 查看这个命令会做什么

git clean –n –d –x 删除的时候包含被忽略的文件

git clean –i 进入交互式命令来进行删除操作。

9、git grep 查找工作目录中相应的字符串

git grep -n gmtime\_r查找包含gmtime\_r的文件并且显示出行号

git grep --count hello 查找包含hello的文件并且找出每个匹配文件中有多少处匹配。

git grep -p hello查看所查找的匹配你属于哪个方法或者函数。

git log –S hello --oneline 查找某个字符串是在什么时候出现的。显示此字符串被添加或者删除的时候。

10、git filter-branch --tree-filter ‘rm -f password.txt’ HEAD 在所有的提交上删除密码文件

11、HEAD是当前分支的最后一次提交。

索引：预计的下一次提交，我们称之为暂存区。

12、git reset HEAD filename是将暂存区的修改重置到工作目录（撤销暂存，不会修改文件内容）

git reset 7dadac 将最后一次的提交重置到7dadac

git reset --soft HEAD~将提交重置到最后一次提交的父节点。同时不修改暂存和工作目录。

git reset [--mixed] HEAD~(git reset HEAD~)将提交重置到最后一次提交的父节点。同时更新暂存，不更新工作目录。也就是会退到git add 和git commit之前的状态。

git reset --hard HEAD~撤销了最近的提交，git add git commit命令以及在工作目录所做的全部工作（慎用）

git reset eb34da file.txt将file.txt的提交回退到eb34da，并且将暂存区的内容和eb34da提交中的file.txt的内容一致，工作目录中的内容不会更改。不会更改HEAD的指的位置

git reset --hard eb34da file.txt t将file.txt的提交回退到eb34da，并且将暂存区的内容和eb34da提交中的file.txt的内容一致，工作目录中的内容也更改到和eb34da提交中的file.txt的内容一致。不会更改HEAD的指的位置

13、git checkout 和reset的区别

（1）不加文件路径：git checkout branch切换分支，将HEAD移动到切换到的分支，

git reset branch移动的是HEAD指向的分支。

git checkout branch不会影响工作目录，git reset --hard branch会覆盖工作目录中的内容

（2）加上文件路径

git checkout branch filename 不会修改HEAD，但是会用提交中的文件更新索引，并覆盖工作目录中对应的文件。与git reset --hard branch filename一样的效果。

git reset branch filename只会更新索引不会更新工作目录。

13、合并分支

git merge --abort中止合并

git merge -Xignore-space-change（将单个字符和多个字符视为等同）-Xignore-all-space

whitespace

14、.git目录结构解析

