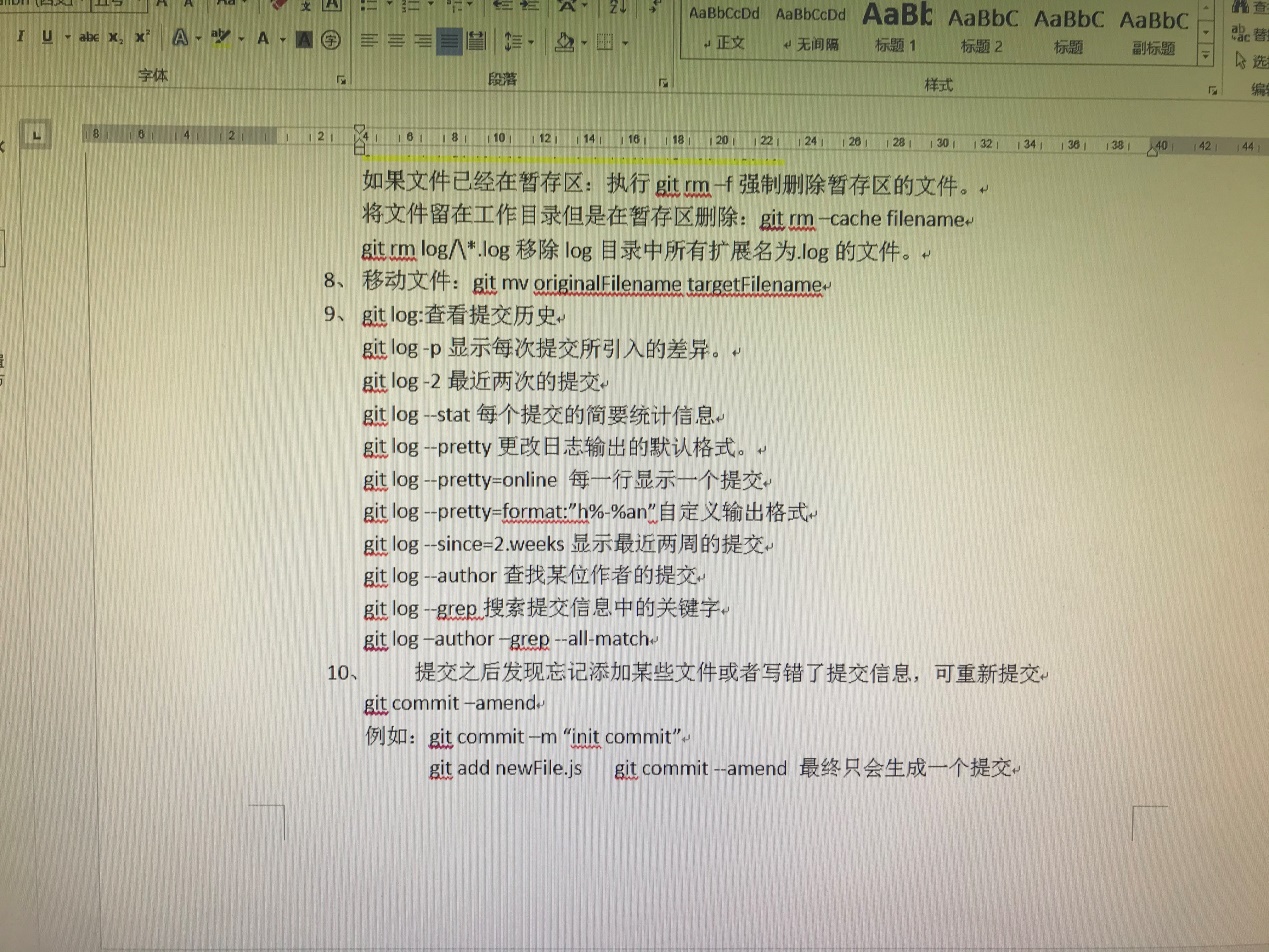
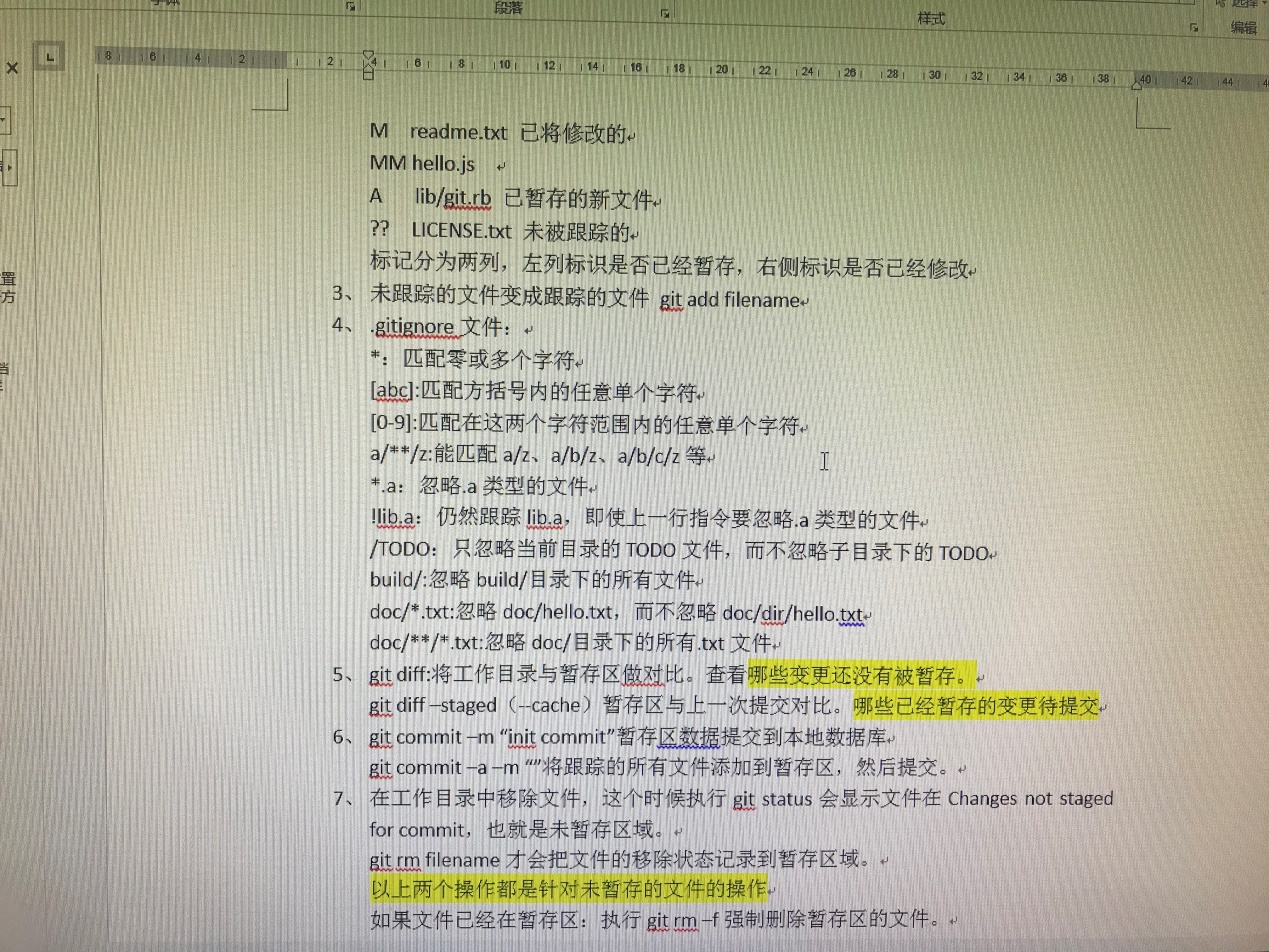
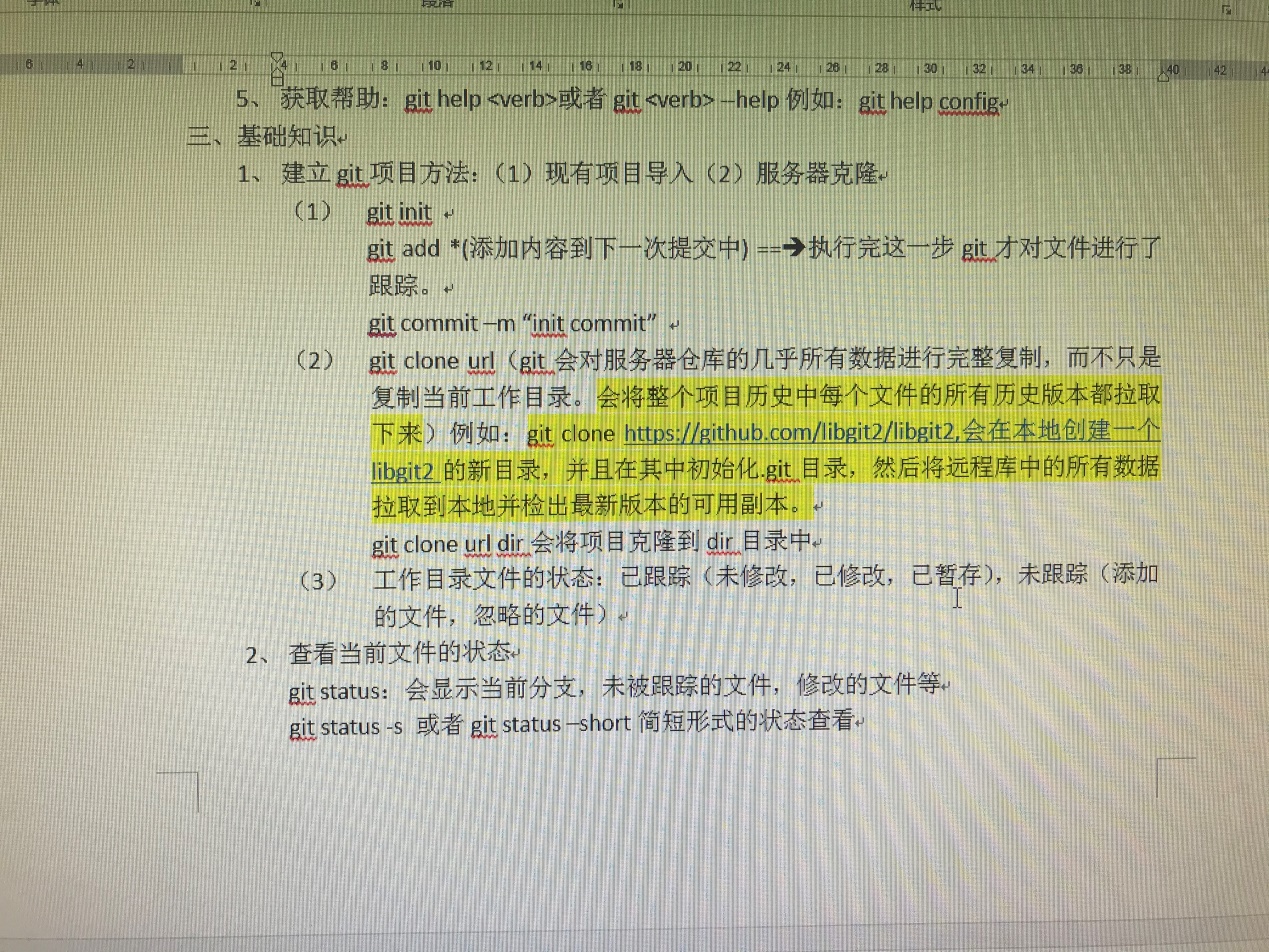
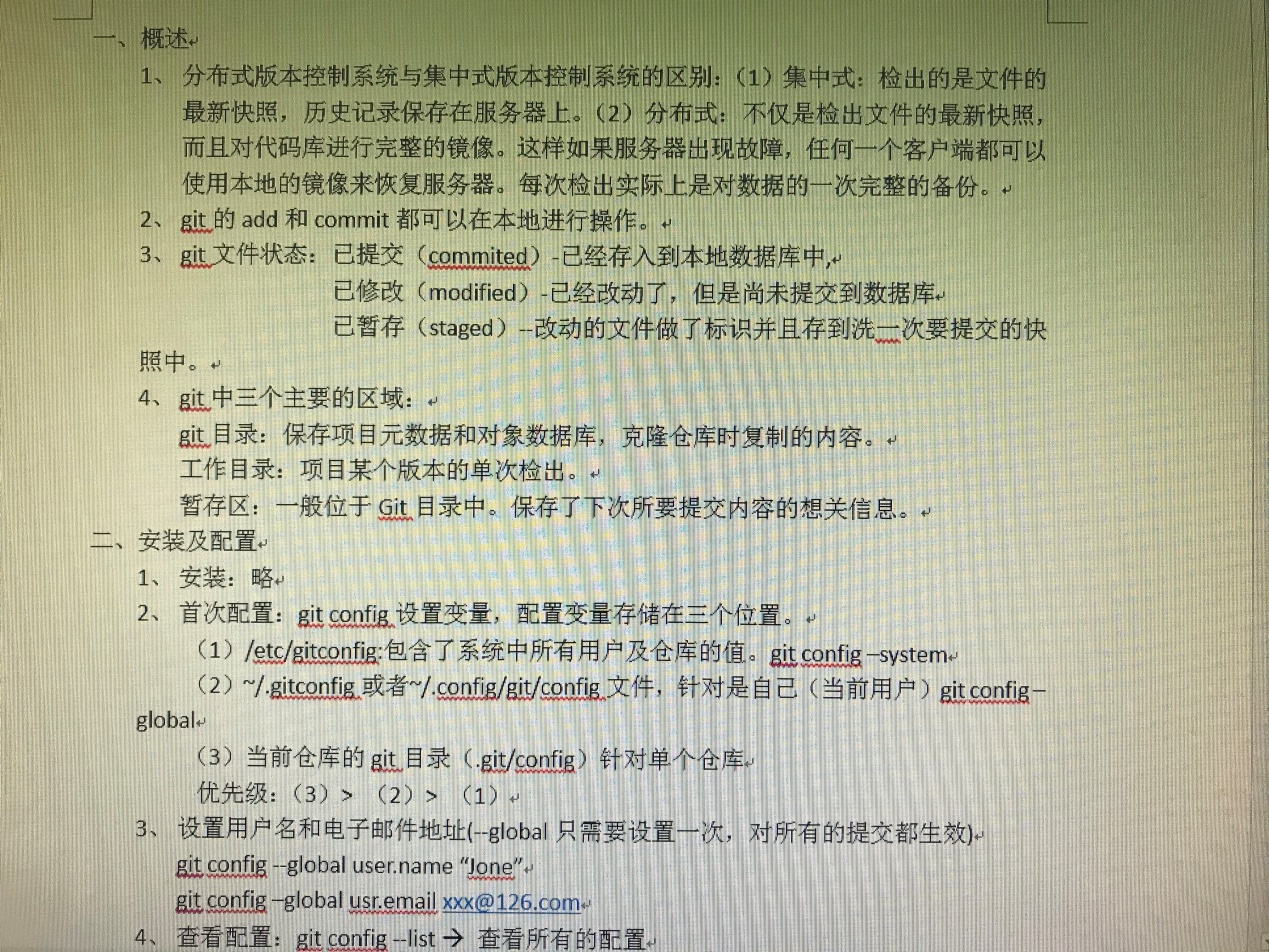
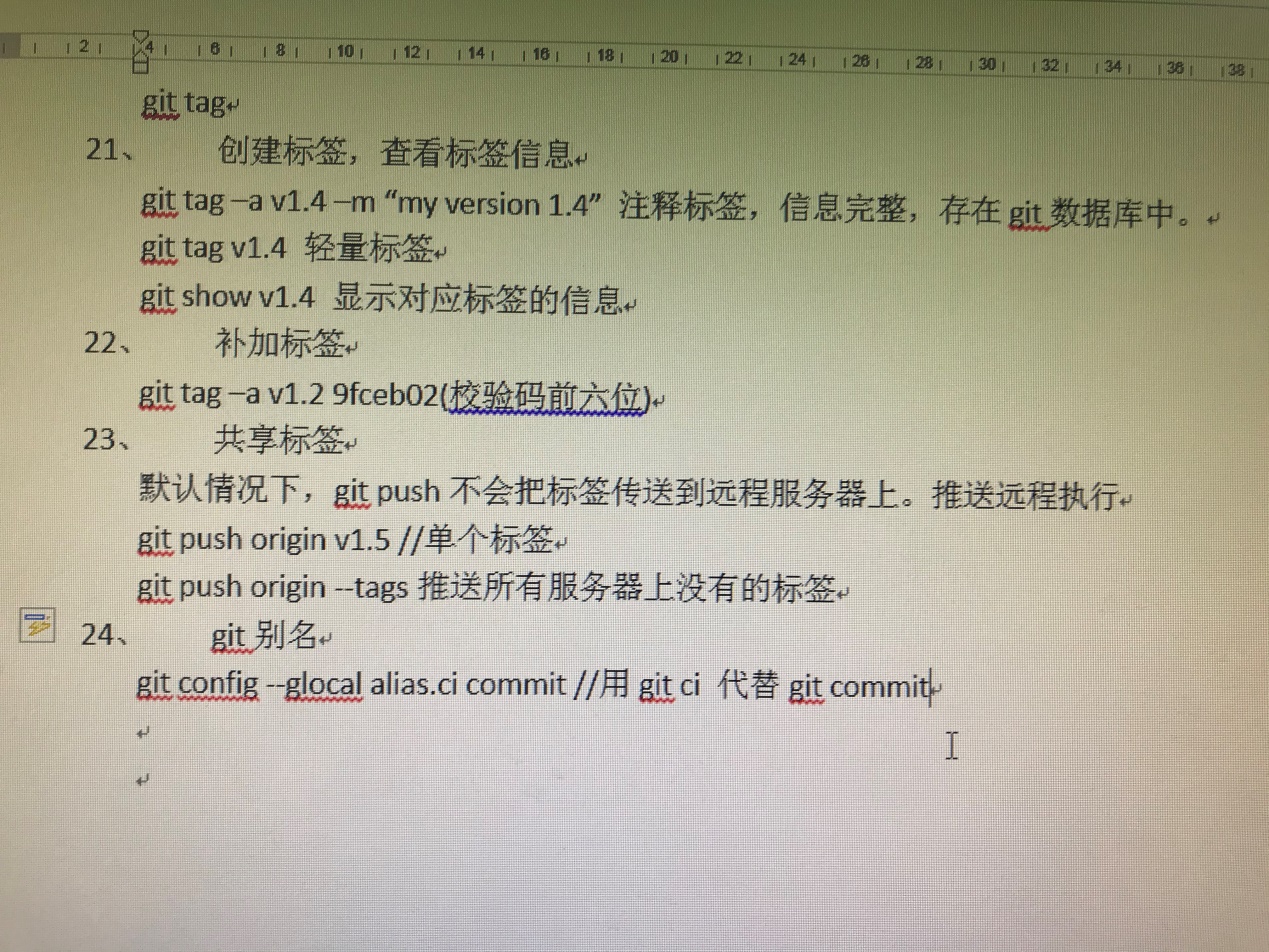
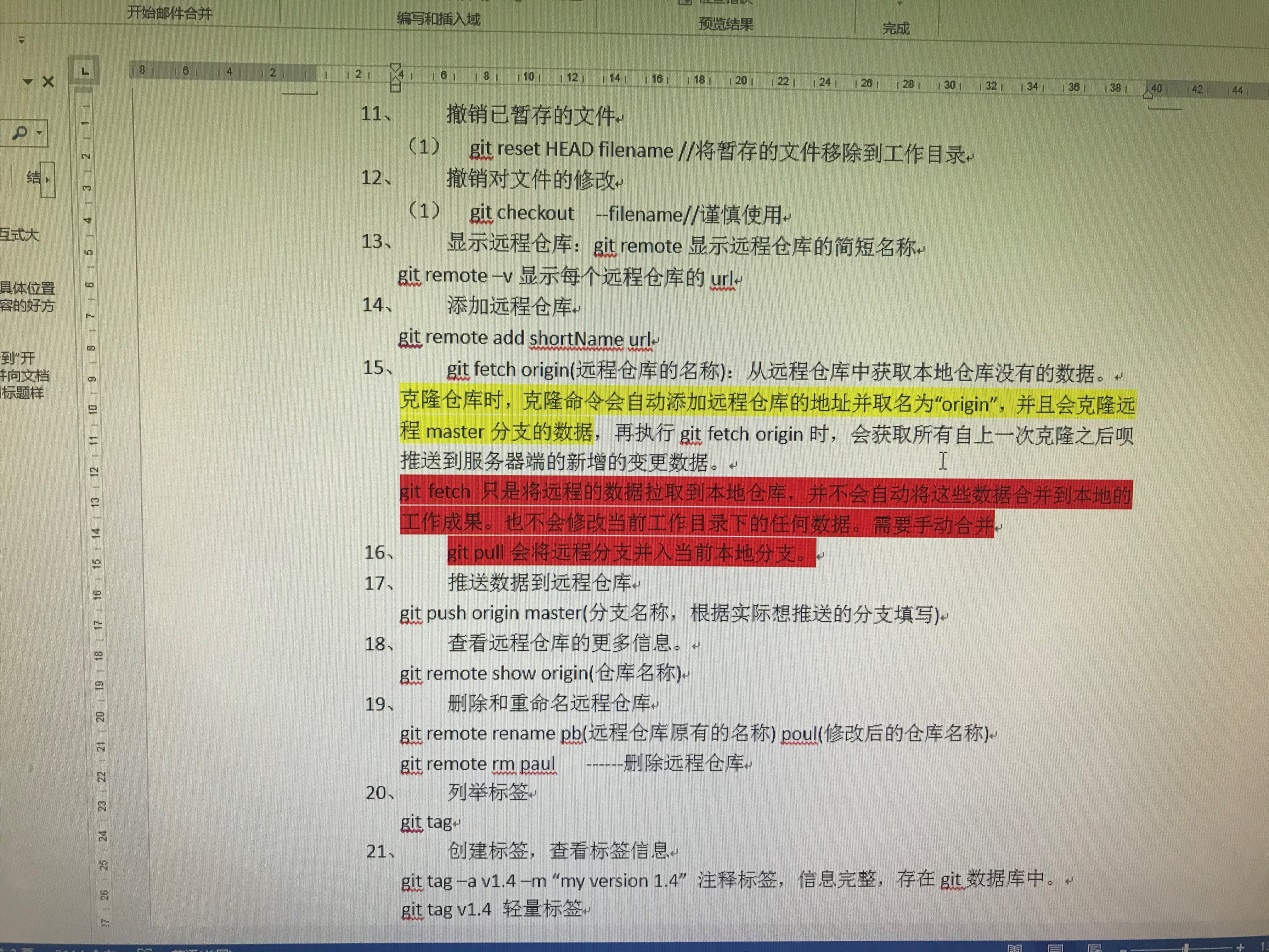
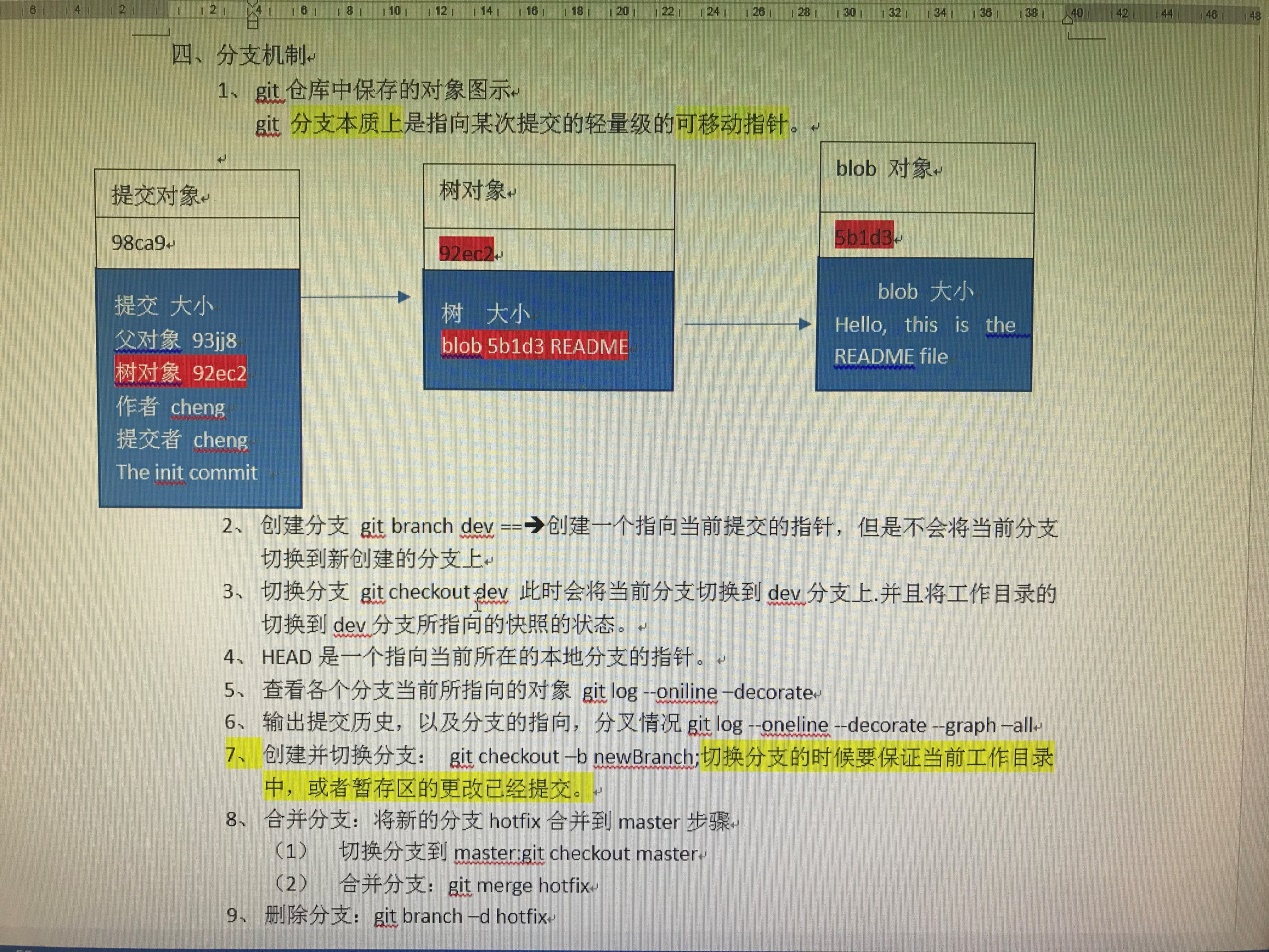
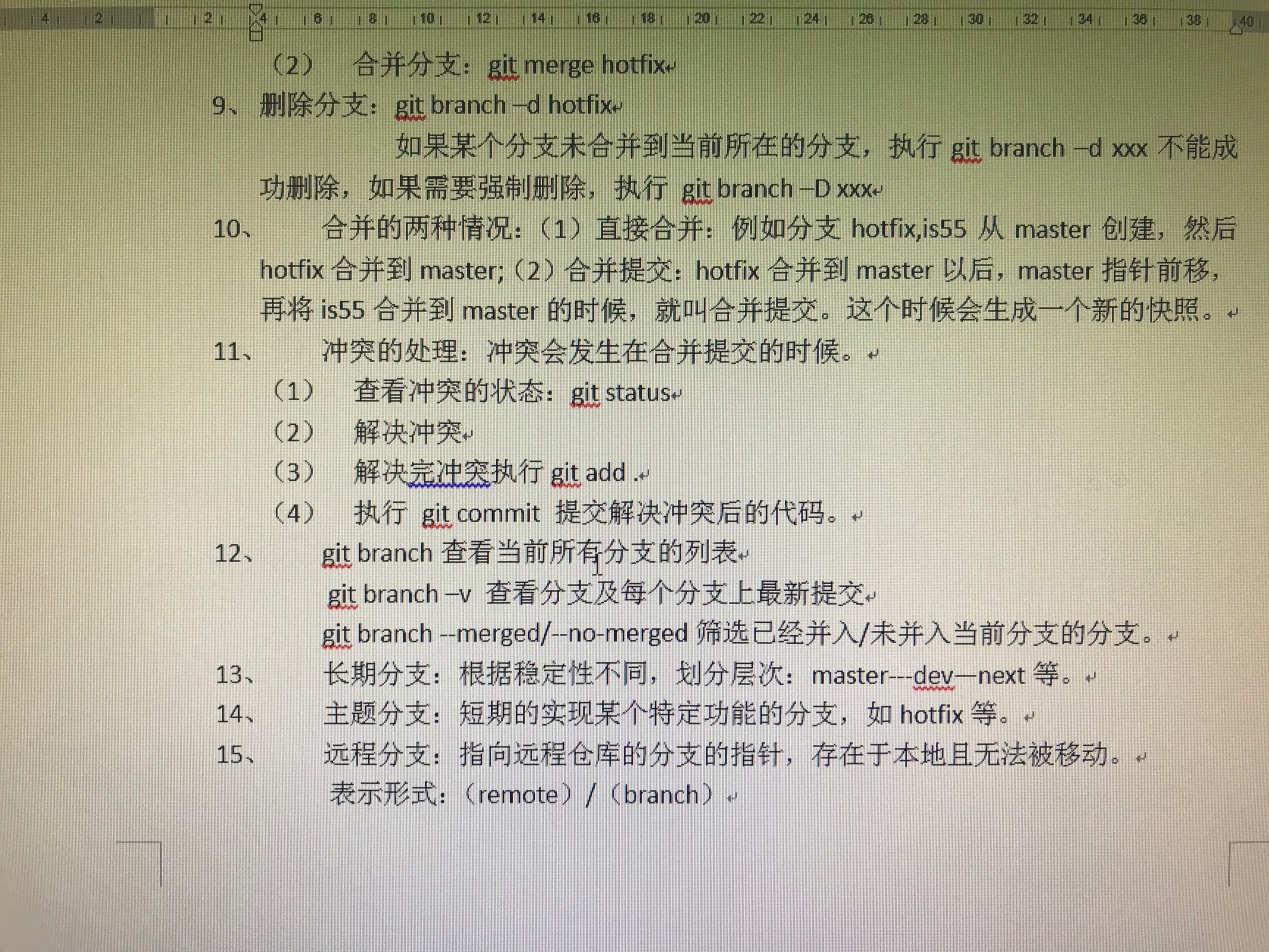
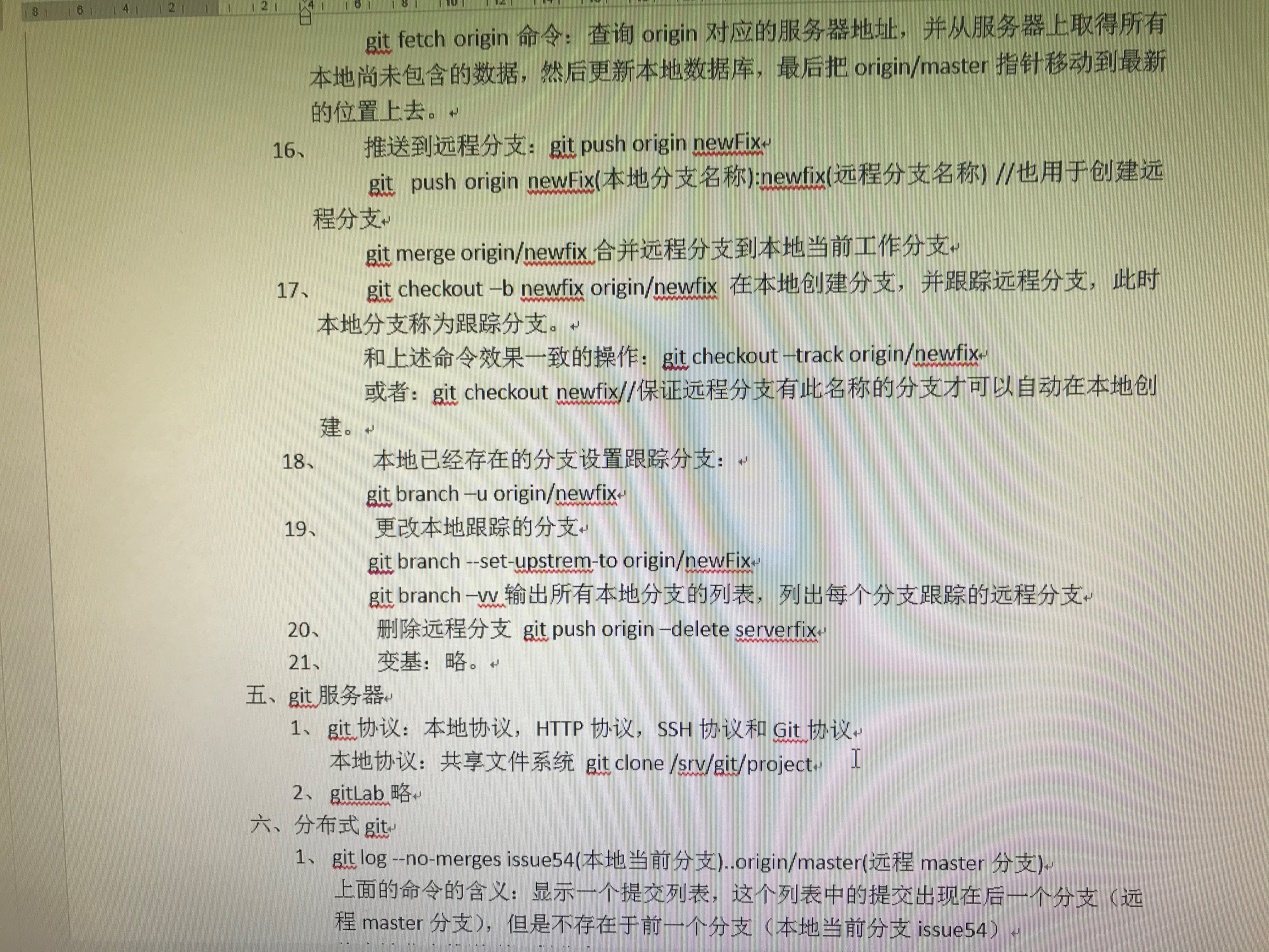
git学习笔记











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 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。